

PENERAPAN STACKING ENSEMBLE LEARNING UNTUK MENINGKATKAN AKURASI PREDIKSI KETEPATAN WAKTU KELULUSAN MAHASISWA

Diterima Redaksi: 22 September 2025; Revisi Akhir: 1 Desember 2025; Diterbitkan Online: 15 Desember 2025

Hiya Nalatissifa¹⁾, Sri Diantika²⁾, Riki Supriyadi³⁾, Nurlaelatul Maulidah⁴⁾, Ahmad Fauzi⁵⁾

^{1, 2, 4, 5)} Fakultas Teknik dan Informatika, Universitas Bina Sarana Informatika

³⁾ Fakultas Teknologi Informasi, Universitas Nusa Mandiri

^{1, 2, 4, 5)} Jalan Kramat Raya No 98, Kwitang, Kec Senen, Kota Jakarta Pusat 10450

³⁾ Jl. Raya Jatiwaringin No.2, Cipinang Melayu, Makasar, Jakarta Timur, Daerah Khusus Ibukota Jakarta 13620

e-mail: hiya.hys@bsi.ac.id¹⁾, sri.szd@bsi.ac.id²⁾, riki.rsd@nusamandiri.ac.id³⁾, nurlaelatul.nlt@bsi.ac.id⁴⁾, ahmad.fzx@bsi.ac.id⁵⁾

Abstrak: Kepastian waktu lulus mahasiswa adalah salah satu indikator krusial dalam mengevaluasi mutu pendidikan tinggi sekaligus menjadi komponen utama akreditasi program studi maupun institusi. Oleh karena itu, kemampuan untuk memprediksi kelulusan tepat waktu sangat diperlukan guna mendukung intervensi akademik yang tepat sasaran. Tujuan dari penelitian ini yaitu membangun model prediksi ketepatan waktu kelulusan mahasiswa dengan pendekatan Stacking Ensemble Learning. Model yang digunakan terdiri atas Random Forest dan XGBoost sebagai base learners, serta Logistic Regression sebagai meta learner. Dataset yang digunakan berjumlah 1.687 data mahasiswa dengan atribut nilai Indeks Prestasi (IP) dari semester 1 hingga semester 4 serta label kelulusan tepat waktu. Tahapan penelitian meliputi preprocessing data (konversi label, penambahan fitur turunan, normalisasi, dan penyeimbangan kelas dengan SMOTE), pemisahan data latih dan uji dengan rasio 70:30, serta pembangunan model stacking. Evaluasi dilakukan menggunakan metrik Accuracy, Precision, Recall, F1-Score, dan Confusion Matrix. Hasil penelitian menunjukkan bahwa model stacking ensemble mampu mencapai akurasi sebesar 92,27%, precision 94,49%, recall 90,13%, dan F1-Score 92,26%. Hasil ini lebih tinggi dibandingkan penelitian sebelumnya yang menggunakan Random Forest dengan Random Oversampling, yang hanya memperoleh akurasi 90,04% dan F1-Score 88,38%. Dengan demikian, penelitian ini membuktikan bahwa Stacking Ensemble Learning berkinerja lebih baik dalam meningkatkan performa prediksi ketepatan waktu kelulusan mahasiswa dan dapat dijadikan sebagai dasar pengembangan sistem pendukung keputusan akademik.

Kata Kunci— Data Mining, Ketepatan Waktu Kelulusan, Stacking Ensemble Learning

Abstract: The certainty of students' graduation time is one of the crucial indicators in evaluating the quality of higher education and is also a main component of study program and institutional accreditation. Therefore, the ability to predict on-time graduation is essential to support targeted academic interventions. This study aims to develop a predictive model for student on-time graduation using the Stacking Ensemble Learning approach. The model consists of Random Forest and XGBoost as base learners, with Logistic Regression as the meta learner. The dataset used comprises 1,687 student records containing Grade Point Average (GPA) scores from the first to the fourth semester along with the on-time graduation label. The research stages include data preprocessing (label conversion, additional derived features, normalization, and class balancing using SMOTE), splitting the data into training and testing sets with a 70:30 ratio, and building the stacking model. Evaluation was conducted using Accuracy, Precision, Recall, F1-Score, and Confusion Matrix. The results show that the stacking ensemble model achieved an accuracy of 92.27%, precision of 94.49%, recall of 90.13%, and F1-Score of 92.26%. These results outperform the previous study that used Random Forest with Random Oversampling, which only achieved 90.04% accuracy and 88.38% F1-Score. Thus, this study demonstrates that Stacking Ensemble Learning performs better in improving the prediction performance for student on-time graduation and can serve as the basis for developing an academic decision support system.

Keywords— Data Mining, On-Time Graduation, Stacking Ensemble Learning

I. PENDAHULUAN

Ketepatan waktu kelulusan mahasiswa menjadi salah satu indikator penting dalam menilai kualitas pendidikan tinggi [1]. Selain sebagai bentuk keberhasilan mahasiswa dalam menyelesaikan studi, tingkat kelulusan tepat waktu juga menjadi bagian dari indikator akreditasi suatu program studi maupun institusi [2]. Oleh karena itu, kemampuan untuk memprediksi secara dini apakah seorang mahasiswa akan lulus tepat waktu sangat penting untuk mendukung intervensi akademik yang tepat sasaran [3].

Berbagai pendekatan telah dilakukan untuk memprediksi kelulusan mahasiswa, salah satunya adalah dengan memanfaatkan nilai akademik seperti Indeks Prestasi Kumulatif (IPK) dari semester ke semester [4]. Dalam penelitian sebelumnya oleh [5], digunakan algoritma *Random Forest* untuk memprediksi ketepatan waktu lulus dengan pendekatan *Random Oversampling* (ROS) guna mengatasi ketidakseimbangan kelas data. Hasilnya menunjukkan kinerja yang cukup memuaskan dengan tingkat akurasi 90,04%, *precision* 87,05%, dan *recall* 90,04%.

Namun demikian, pendekatan yang hanya menggunakan satu algoritma cenderung memiliki keterbatasan dalam menangkap kompleksitas pola dalam data [6]. Oleh karena itu, penelitian ini mencoba memperkenalkan pendekatan *Stacking Ensemble Learning*, yaitu metode kombinasi dari berbagai model pembelajaran mesin untuk meningkatkan kinerja prediksi [7]. Pada penelitian ini digunakan algoritma *Random Forest*, *XGBoost*, serta *meta-learner Logistic Regression* [8].

Selain pendekatan *ensemble*, penelitian ini juga menambahkan fitur baru seperti rata-rata IPK, pertumbuhan IPK, dan standar deviasi IPK untuk memperkaya informasi yang digunakan dalam proses pelatihan model. Teknik penyeimbangan data menggunakan SMOTE (*Synthetic Minority Oversampling Technique*) juga digunakan untuk menjamin model tidak berpihak kepada kelas yang dominan [9].

Hasil dari penelitian ini menunjukkan bahwa pendekatan *stacking ensemble* mampu meningkatkan performa model dengan akurasi mencapai 92,27%, *precision* 94,49%, *recall* 90,13%, dan *F1-Score* 92,26%. Nilai ini lebih unggul dibandingkan hasil penelitian oleh [5] yang hanya menggunakan *Random Forest*. Oleh karena itu, studi ini berkontribusi pada peningkatan akurasi prediksi ketepatan waktu kelulusan mahasiswa melalui pendekatan yang lebih kompleks dan menyeluruh.

II. TINJAUAN PUSTAKA

Tinjauan Pustaka disusun untuk memberikan dasar teoretis serta memuat penjelasan tentang konsep dasar, teori yang mendukung, serta hasil penelitian sebelumnya, sehingga dapat menjadi acuan dalam merumuskan model yang digunakan

A. Data mining

Proses mengeksplorasi dan menganalisis data dalam volume besar dengan tujuan menemukan pola, tren, atau wawasan yang tidak terlihat dengan memanfaatkan metode statistik, pembelajaran mesin, serta kecerdasan buatan merupakan pengertian dari *Data Mining*. Tujuannya adalah mengubah data yang belum diolah menjadi informasi yang relevan untuk mendukung proses pengambilan Keputusan [10]. Dalam bidang pendidikan, *Data mining* berkembang menjadi cabang khusus yang dikenal dengan *Educational Data mining (EDM)*. EDM digunakan untuk menganalisis data akademik mahasiswa agar institusi dapat memahami perilaku belajar, memprediksi performa, serta meningkatkan kualitas pembelajaran [11]. Beberapa contoh penerapannya adalah:

- 1) *Prediksi ketepatan waktu kelulusan mahasiswa berdasarkan nilai IP semester dan IPK kumulatif.*
- 2) *Analisis risiko drop-out, untuk mengidentifikasi mahasiswa yang berpotensi tidak menyelesaikan studi.*
- 3) *Rekomendasi mata kuliah atau jalur studi sesuai dengan performa akademik mahasiswa.*
- 4) *Evaluasi efektivitas kurikulum dan pengajaran berdasarkan hasil belajar mahasiswa.*

Dalam konteks penelitian ini, *Data mining* dimanfaatkan untuk membangun model prediksi ketepatan waktu kelulusan mahasiswa menggunakan pendekatan *Stacking Ensemble Learning*. Dengan mengombinasikan data akademik dan teknik *machine learning*, diharapkan dapat dihasilkan model prediksi yang lebih akurat dan bermanfaat sebagai dasar sistem pendukung keputusan akademik.

B. Random Forest Classifier

Merupakan algoritma *machine learning* berbasis *ensemble learning* yang bekerja dengan membuat sejumlah besar pohon Keputusan pada data pelatihan, selanjutnya menggabungkan output prediksi dari setiap pohon dengan metode *voting* mayoritas untuk klasifikasi/rata-rata untuk regresi [12]. Keunggulan utama *Random Forest* adalah:

- 1) Mengurangi *overfitting*, karena prediksi akhir dihasilkan dari gabungan banyak pohon, bukan hanya satu pohon keputusan.
- 2) Mampu menangani data dengan banyak fitur, baik numerik maupun kategorikal.
- 3) Robust terhadap *noise*, karena sifat ensemble yang mengurangi variansi model tunggal.
- 4) Memberikan estimasi pentingnya variabel (*feature importance*), yang membantu dalam interpretasi model.

C. XGBoost Classifier

XGBoost (*Extreme Gradient Boosting*) merupakan algoritma *boosting* yang dikembangkan untuk efisiensi dan tingkat akurasi yang tinggi. Tidak seperti *Random Forest* yang memanfaatkan *voting*, *XGBoost* mengembangkan model secara berurutan, di mana setiap model baru berupaya untuk memperbaiki kesalahan dari model yang sebelumnya [13]. *XGBoost* unggul dalam menangani data tidak seimbang dan memiliki kemampuan regularisasi yang baik [14]. *XGBoost* juga fleksibel untuk berbagai tugas seperti klasifikasi, regresi, maupun ranking, sehingga sering menjadi pilihan utama dalam berbagai penelitian prediktif.

D. Logistic Regression

Algoritma klasifikasi linier yang bekerja dengan menghitung probabilitas berdasarkan kombinasi linier dari input [15]. Keunggulan algoritma ini terletak pada kesederhanaan dan kemudahan interpretasi, karena menghasilkan koefisien yang dapat menjelaskan pengaruh setiap variabel terhadap probabilitas kelas. *Logistic Regression* juga ringan secara komputasi, efektif untuk data yang linier terpisah, dan sering digunakan sebagai baseline model klasifikasi. Dalam konteks ensemble learning, *Logistic Regression* banyak dipilih sebagai *meta learner* karena mampu menggabungkan output dari berbagai model dasar untuk menghasilkan prediksi akhir yang lebih konsisten dan akurat [16].

E. Stacking Ensemble Learning

Stacking Ensemble Learning adalah salah satu teknik ensemble yang mengintegrasikan beberapa model dasar (*base learners*) dengan dukungan model lain sebagai *meta-learner* [17]. Berbeda dengan metode ensemble lain seperti bagging atau boosting yang menggabungkan model sejenis, *stacking* justru memanfaatkan keberagaman algoritma untuk saling melengkapi kelemahan satu sama lain. Dengan cara ini, *stacking* mampu menghasilkan model prediksi yang lebih kuat, stabil, dan akurat.

Proses *stacking* biasanya dilakukan dalam dua lapisan. Pada lapisan pertama, beberapa *base learners* dilatih secara paralel menggunakan dataset yang sama [18]. Hasil prediksi dari beberapa model dasar ini selanjutnya dikumpulkan, baik dalam bentuk probabilitas maupun kelas prediksi. Selanjutnya, hasil tersebut menjadi input bagi lapisan kedua yang berisi *meta-learner*. Model *meta-learner* inilah yang bertugas mengombinasikan prediksi dari *base learners* untuk menghasilkan keputusan akhir.

Kelebihan utama *stacking* terletak pada kemampuannya memanfaatkan keragaman model dasar. Misalnya, algoritma berbasis pohon seperti *Random Forest* atau *XGBoost* dapat menangkap pola non-linear yang kompleks, sementara *Logistic Regression* sebagai *meta-learner* memberikan kombinasi yang lebih sederhana dan interpretatif. Dengan demikian, *stacking* tidak hanya meningkatkan akurasi, tetapi juga mampu menyeimbangkan antara kompleksitas model dan generalisasi hasil prediksi [19].

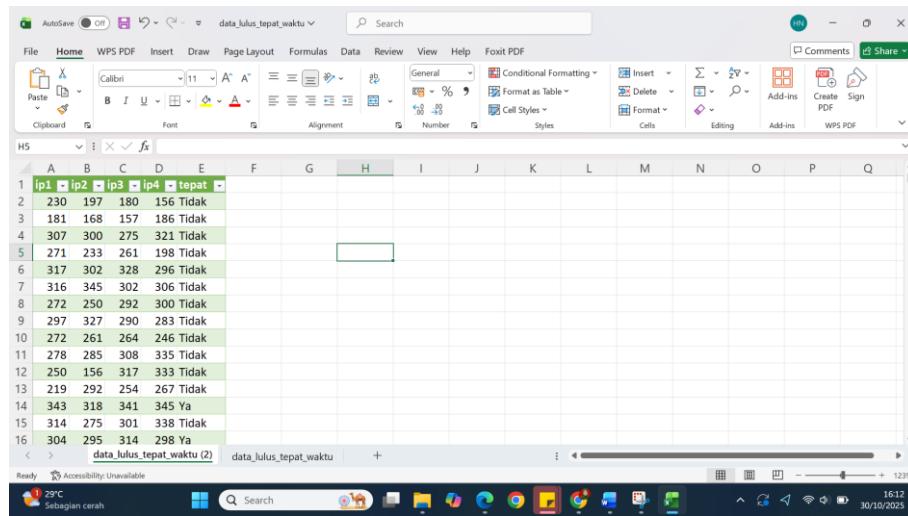
III. METODE PENELITIAN

Tahapan ini mencakup pengumpulan dan persiapan data, *preprocessing*, pemisahan data, pemilihan algoritma, pembangunan model dengan pendekatan *Stacking Ensemble Learning*, serta evaluasi performa model menggunakan berbagai metrik klasifikasi.

A. Dataset

Penelitian ini menggunakan *dataset* dari *Kaggle* dengan judul *On Time Graduation Classification* yang terdiri dari 1687 data yang memuat nilai Indeks Prestasi (IP) mahasiswa dari semester 1 sampai 4. Setiap entri data juga mencakup label apakah mahasiswa lulus tepat waktu (Ya) atau tidak (Tidak). *Dataset* terdiri dari 5 atribut utama:

- 1) *ip1, ip2, ip3, ip4 merupakan nilai IP semester*
- 2) *tepat merupakan label ketepatan waktu kelulusan*



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	ip1	ip2	ip3	ip4	tepat												
2	230	197	180	156	Tidak												
3	181	168	157	186	Tidak												
4	307	300	275	321	Tidak												
5	271	233	261	198	Tidak												
6	317	302	328	296	Tidak												
7	316	345	302	306	Tidak												
8	272	250	292	300	Tidak												
9	297	327	290	283	Tidak												
10	272	261	264	246	Tidak												
11	278	285	300	335	Tidak												
12	250	156	317	333	Tidak												
13	219	292	254	267	Tidak												
14	343	318	341	345	Ya												
15	314	275	301	338	Tidak												
16	304	295	314	298	Ya												

Gambar 1. Dataset

B. Preprocessing Data

Dalam penelitian ini, *preprocessing* dilakukan meliputi beberapa langkah utama, antara lain konversi label, penambahan fitur turunan, normalisasi fitur, dan penyeimbangan kelas data.

1) Konversi Label

Kolom target tepat, yang semula berupa data kategorik berupa string “Ya” (lulus tepat waktu) dan “Tidak” (tidak lulus tepat waktu), perlu dikonversi menjadi bentuk numerik agar dapat diproses oleh algoritma klasifikasi. Ini dilakukan dengan menggunakan fungsi `.map()` dari pandas:

```
# Konversi label ke numerik
df['tepat'] = df['tepat'].map({'Ya': 1, 'Tidak': 0})
```

Gambar 2. Konversi Label ke Numerik

Konversi ini menghasilkan label biner dengan 1 sebagai kelas positif (lulus tepat waktu) dan 0 sebagai kelas negatif (tidak lulus tepat waktu).

2) Penambahan Fitur Turunan

Agar model dapat menangkap lebih banyak pola dari data, ditambahkan tiga fitur turunan berbasis statistik dari nilai IP semester:

- a. *ipk_avg*: rata-rata IP dari semester 1 sampai 4
- b. *ip_growth*: selisih antara IP semester 4 dan IP semester 1, yang mencerminkan peningkatan atau penurunan performa
- c. *ipk_std*: standar deviasi IP antar semester untuk melihat kestabilan nilai IP mahasiswa

```
# Fitur tambahan
df['ipk_avg'] = df[['ip1', 'ip2', 'ip3', 'ip4']].mean(axis=1)
df['ip_growth'] = df['ip4'] - df['ip1']
df['ipk_std'] = df[['ip1', 'ip2', 'ip3', 'ip4']].std(axis=1)
```

Gambar 3. Penambahan Fitur

Fitur-fitur ini memberikan konteks tambahan selain nilai IP individual, seperti kestabilan, tren akademik, dan performa agregat mahasiswa.

3) Normalisasi Data

Setelah penambahan fitur, seluruh atribut numerik distandarisasi menggunakan metode *StandardScaler* dari *Scikit-learn*. Tujuannya adalah untuk menyesuaikan skala tiap fitur agar memiliki distribusi dengan rata-rata 0 dan deviasi standar 1. Ini sangat krusial, khususnya ketika menerapkan algoritma seperti *Logistic Regression* atau *stacking* model yang sensitif terhadap perbedaan skala [20]:

```
# Standardisasi fitur
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

Gambar 4. Standarisasi Fitur

4) Penyeimbangan Data

Distribusi data awal menunjukkan bahwa kelas “tidak lulus tepat waktu” lebih dominan dibandingkan “lulus tepat waktu”, yang dapat menyebabkan bias pada model. Untuk mengatasi hal ini, digunakan teknik SMOTE yang berfungsi menghasilkan sampel sintetis tambahan dari kelas minoritas berdasarkan tetangga terdekat, sehingga distribusi antara kelas mayoritas dan minoritas menjadi seimbang [21]:

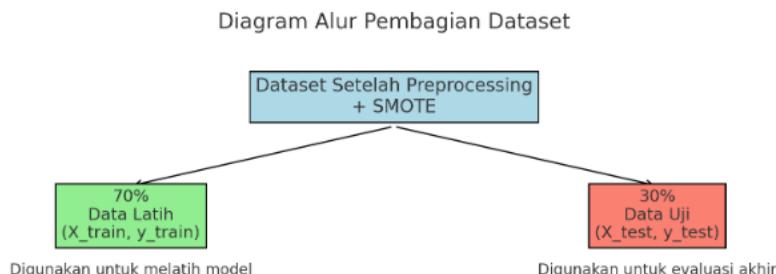
```
# SMOTE
smote = SMOTE(random_state=42)
X_resampled, y_resampled = smote.fit_resample(X_scaled, y)
```

Gambar 5. SMOTE

Dengan proses ini, model dilatih dengan data yang seimbang, mengurangi kemungkinan model hanya “menebak” kelas mayoritas, dan meningkatkan kemampuan model untuk mendeteksi mahasiswa diprediksi akan lulus tepat waktu.

C. Pemisahan Data

Dataset yang telah diproses dibagi antara data pelatihan dan data pengujian dengan proporsi 70:30 menggunakan fungsi *train_test_split*. Data latih digunakan untuk melatih model, sedangkan data uji digunakan untuk menilai kinerja model.



Gambar 6. Train Test Split

Berikut adalah diagram alur yang menggambarkan proses pembagian dataset setelah preprocessing dan SMOTE:

- Dataset awal dibagi menjadi dua bagian menggunakan *train_test_split*.
- 70% data untuk melatih model (X_{train}, y_{train}).
- 30% data untuk menilai kinerja model (X_{test}, y_{test}).

D. Algoritma dan Model

Dalam penelitian ini digunakan pendekatan *Stacking Ensemble Learning* untuk membangun model prediksi ketepatan waktu kelulusan mahasiswa. Pendekatan ini dipilih karena memiliki kemampuan untuk menggabungkan kekuatan beberapa model berbeda (*base learners*) guna meningkatkan performa prediksi [22]. Proses *stacking* melibatkan pelatihan beberapa model dasar secara paralel, dan hasil prediksi mereka dijadikan input bagi model *meta-learner* untuk menghasilkan keputusan akhir.

1) Base Learners

Model dasar (*base learners*) yang digunakan dalam *stacking* terdiri dari dua algoritma yang umum dan terbukti kuat dalam berbagai tugas klasifikasi yaitu *Random Forest Classifier* dan *XGBoost Classifier*. Dalam penelitian ini, digunakan *Random Forest* dengan jumlah pohon (*n_estimators*) sebanyak 100 dan *random_state*=42 untuk replikasi hasil. *XGBoost* digunakan dengan parameter default, tetapi dengan penyesuaian *eval_metric='logloss'* dan *use_label_encoder=False*.

2) Meta Learner

Sebagai *meta-learner*, digunakan *Logistic Regression*, metode ini dipilih karena sifatnya yang sederhana, interpretatif, dan efektif dalam menggabungkan output prediksi dari *base learners* [23]. *Meta-learner* ini mengambil hasil prediksi dari *Random Forest* dan *XGBoost* (baik dalam bentuk probabilitas maupun kelas akhir), kemudian mempelajari hubungan antara prediksi tersebut dengan label sebenarnya untuk memberikan output akhir.

```
base_learners = [
    ('rf', RandomForestClassifier(n_estimators=100, random_state=42)),
    ('xgb', XGBClassifier(use_label_encoder=False, eval_metric='logloss', random_state=42))
]

meta_learner = LogisticRegression()
```

Gambar 7. Base Learners dan Meta Learner

3) Implementasi Stacking

Implementasi *stacking* dilakukan menggunakan *StackingClassifier* dari pustaka *sklearn.ensemble*. Parameter *cv=5* digunakan untuk menerapkan *5-fold cross-validation* saat melatih meta learner, sehingga hasil model menjadi lebih general dan tidak overfit. Selain itu, digunakan parameter *passthrough=True*, yang berarti fitur asli (bukan hanya output prediksi *base learners*) juga dikirimkan ke meta learner, sehingga memperkaya input yang dipelajari oleh *Logistic Regression*. Model ini kemudian dilatih menggunakan *stack_model.fit(X_train, y_train)* dan dievaluasi pada data uji. Pendekatan *stacking* memungkinkan model untuk memperbaiki kelemahan dari masing-masing base learner secara kolektif, sehingga hasil prediksi menjadi lebih akurat dan stabil.

```
stack_model = StackingClassifier(
    estimators=base_learners,
    final_estimator=meta_learner,
    cv=5,
    passthrough=True
)
stack_model.fit(X_train, y_train)
```

Gambar 8. Stacking Classifier

E. Evaluasi Model

Untuk menilai kinerja klasifikasi berdasarkan prediksi yang dikerjakan terhadap data uji dilakukanlah evaluasi model [24]. Pada penelitian ini, evaluasi model menggunakan empat metrik utama, yaitu *Accuracy*, *Precision*, *Recall*, dan *F1-Score*. Akurasi mengukur seberapa banyak prediksi yang benar secara keseluruhan, Presisi menilai ketepatan model dalam memprediksi mahasiswa yang lulus tepat

waktu, *Recall* mengukur kemampuan model dalam menemukan seluruh mahasiswa yang benar-benar lulus tepat waktu, sedangkan *F1-Score* yaitu keseimbangan antara *Precision* dan *Recall*. Selain itu, digunakan *Confusion Matrix* untuk memberikan gambaran detail distribusi prediksi. *Confusion Matrix* membantu memahami jenis kesalahan prediksi yang masih terjadi, sehingga dapat menjadi dasar untuk perbaikan model lebih lanjut [25].

IV. HASIL DAN PEMBAHASAN

Bagian ini menyajikan hasil dari penerapan *Stacking Ensemble Learning* yang dievaluasi dengan metrik klasifikasi dan *Confusion Matrix*, serta dibandingkan dengan penelitian sebelumnya untuk melihat peningkatan performa.

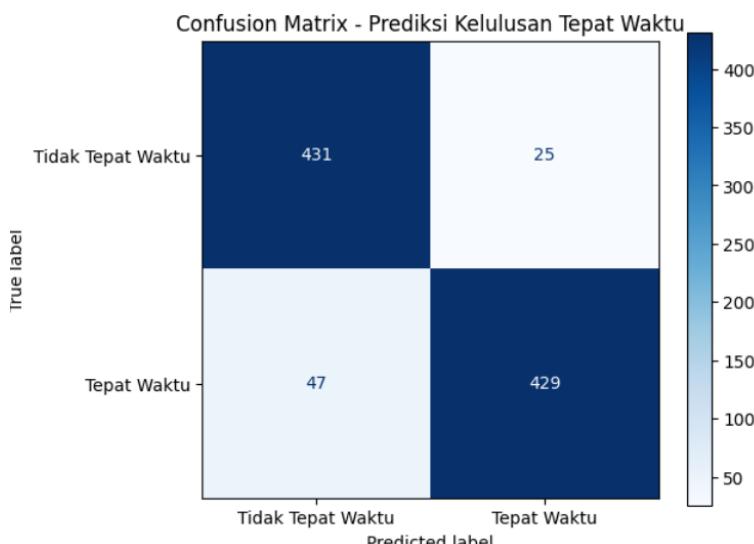
A. Hasil Evaluasi Model

Setelah proses pelatihan dan pengujian, diperoleh hasil evaluasi model sebagai berikut:

```
Hasil Evaluasi Stacking Ensemble:  
Accuracy : 0.9227  
Precision : 0.9449  
Recall : 0.9013  
F1-Score : 0.9226
```

Gambar 9. Hasil Evaluasi Model Stacking Ensemble

Nilai akurasi di atas menunjukkan bahwa model *stacking* mampu memprediksi ketepatan waktu kelulusan mahasiswa dengan sangat baik. Nilai *precision* dan *recall* yang tinggi menunjukkan bahwa model ini tidak hanya akurat, tetapi juga konsisten dalam mengenali mahasiswa yang berpotensi lulus tepat waktu. Di samping itu, hasil prediksi divisualisasikan melalui *Confusion Matrix* untuk memberikan penjelasan lebih rinci tentang distribusi prediksi model:



Gambar 10. Confusion Matrix Stacking Ensemble Model

:

- 1) *True Negative (TN) = 431*
Mahasiswa diprediksi *tidak lulus tepat waktu* dan benar-benar tidak lulus tepat waktu.
- 2) *True Positive (TP) = 429*
Mahasiswa diprediksi *lulus tepat waktu* dan benar-benar lulus tepat waktu.
- 3) *False Positive (FP) = 25*
Mahasiswa diprediksi *lulus tepat waktu*, tetapi sebenarnya tidak.
- 4) *False Negative (FN) = 47*
Mahasiswa diprediksi *tidak lulus tepat waktu*, padahal sebenarnya lulus tepat waktu

Hasil ini menunjukkan bahwa kesalahan prediksi masih lebih banyak terjadi pada *False Negative*, yaitu model gagal mengenali sebagian mahasiswa yang sebenarnya lulus tepat waktu. Namun secara keseluruhan, distribusi prediksi benar jauh lebih dominan dibandingkan kesalahan, sehingga performa model dapat dikatakan sangat baik.

Temuan penelitian ini menunjukkan bahwa model yang dikembangkan berhasil mencapai akurasi sebesar 92,27% dengan F1-Score 92,26%. Pencapaian ini lebih tinggi dibandingkan studi sebelumnya oleh [5] yang menggunakan algoritma *Random Forest* dengan metode *Random Oversampling*, di mana akurasi yang diperoleh sebesar 90,04% dengan *F1-Score* 88,38%. Dengan demikian, pendekatan *stacking ensemble* yang menggabungkan dua algoritma pohon keputusan dengan *meta learner linear* terbukti menghasilkan model yang lebih *robust* dan akurat.

Tabel 1. Perbandingan Hasil Penelitian

Metode	Akurasi	F1-Score
<i>Stacking Ensemble</i>	92,27%	92,26%
<i>Random Forest + Random Oversampling</i>	90,04%	88,38%

B. Analisis Hasil

Penambahan fitur turunan seperti rata-rata IPK, pertumbuhan IPK, dan deviasi antar semester memberikan informasi tambahan yang penting dalam memodelkan performa akademik mahasiswa. Selain itu, penggunaan SMOTE berhasil menangani masalah data tidak seimbang yang umumnya menyebabkan bias prediksi ke kelas mayoritas.

V. KESIMPULAN DAN SARAN

Penelitian ini membuktikan bahwa pendekatan *Stacking Ensemble Learning* memberikan peningkatan signifikan terhadap akurasi prediksi ketepatan waktu kelulusan mahasiswa. Model yang terdiri dari *Random Forest* dan *XGBoost* sebagai *base learners* serta *Logistic Regression* sebagai *meta learner* mampu menghasilkan akurasi sebesar 92,27%, lebih tinggi dibandingkan pendekatan konvensional. Kontribusi dari penelitian ini terletak pada penerapan *ensemble* model yang digabungkan dengan fitur turunan dan penyeimbangan data menggunakan SMOTE, sehingga meningkatkan stabilitas dan akurasi prediksi. Ke depan, penelitian ini dapat ditingkatkan lebih lanjut dengan menyertakan variabel non-akademik seperti keaktifan organisasi, status ekonomi, dan lama studi skripsi untuk meningkatkan kualitas prediksi.

DAFTAR PUSTAKA

- [1] T. H. Hasibuan And D. Mahdiana, “Prediksi Kelulusan Mahasiswa Tepat Waktu Menggunakan Algoritma C4.5 Pada Uin Syarif Hidayatullah Jakarta,” *Skanika*, Vol. 6, No. 1, Pp. 61–74, 2023, Doi: 10.36080/Skanika.V6i1.2976.
- [2] M. F. Alie And I. Saluza, “Perbandingan Model Prediksi Kelulusan Mahasiswa Menggunakan Algoritma Decision Tree Dan Naive Bayes,” Vol. 5, No. 2, Pp. 666–672, 2025.
- [3] A. Hidayahutulloh And D. Prasetyo, “Penerapan Algoritma Decision Tree Untuk Prediksi Kelulusan Mahasiswa Berdasarkan Data Akademik,” *Teknabis Teknol. Bisnis Dan Pendidik.*, Vol. 2, No. 4, Pp. 615–619, 2024.
- [4] A. Sastra, K. Sabri, B. Yanto, F. Asmen, I. Inal, And A. Crisdianto, “Analisis Pengaruh Gaya Hidup Mahasiswa Terhadap Indeks Prestasi Kumulatif (Ipk) Menggunakan Model Regresi,” *Kernel J. Ris. Inov. Bid. Inform. Dan Pendidik. Inform.*, Vol. 4, No. 2, Pp. 89–101, 2023, Doi: 10.31284/J.Kernel.2023.V4i2.7384.
- [5] S. Diantika, H. Nalatissifa, N. Maulidah, R. Supriyadi, And A. Fauzi, “Penerapan Teknik Random Oversampling Untuk Memprediksi Ketepatan Waktu Lulus Menggunakan Algoritma Random Forest,” *Comput. Sci.*, Vol. 4, No. 1, Pp. 11–18, 2024, Doi: 10.31294/Coscience.V4i1.1996.
- [6] Abdul Karim, Budianto Bangun, Sugeng Prayetno, And Mohammad Afrendi, “Optimasi Prediksi Harga Sawit Menggunakan Teknik Stacking Algoritma Machine Learning Dan Deep Learning Dengan Smote,” *Build. Informatics, Technol. Sci.*, Vol. 7, No. 1, Pp. 638–645, 2025, Doi:

10.47065/Bits.V7i1.7239.

[7] H. Nalatissifa, "Implementasi Stacking Ensemble Untuk Prediksi," Vol. 5, No. 1, Pp. 1–7, 2025.

[8] N. Herlinawati, "Klasifikasi Penyakit Diabetes Melitus Menggunakan Metode Stacking Ensemble," *J. Device*, Vol. 15, No. 1, Pp. 163–170, 2025.

[9] A. T. Arsanto, A. Faizin, M. Lutfi, And Z. N. Saadah, "Optimization Of The Naive Bayes Algorithm With Smotetomek Combination For Imbalance Class Fraud Detection," *Sistemasi*, Vol. 13, No. 6, P. 2709, 2024, Doi: 10.32520/Stmsi.V13i6.4719.

[10] M. F. Haryanti, A. Fauzi, A. A. Jelita, And A. Setiyowati, "Pengaruh Data Mining , Strategi Perusahaan Terhadap Laporan Kinerja Perusahaan," Vol. 3, No. 1, Pp. 71–90, 2024.

[11] S. Anastassia, A. Kharis, A. Haqqi, And A. Zili, "Learning Analytics Dan Educational Data Mining Pada Data Pendidikan," Vol. 6, Pp. 12–20, 2022.

[12] M. Azwar, L. Widyawati, R. Azhar, K. Kartarina, T. Tanwir, And A. S. Anas, "Deteksi Malware Pada Perangkat Android Menggunakan Ensemble Learning," *Jtim J. Teknol. Inf. Dan Multimed.*, Vol. 7, No. 3, Pp. 408–419, 2025, Doi: 10.35746/Jtim.V7i3.573.

[13] W. Kurniawan And U. Indahyanti, "Prediksi Angka Harapan Hidup Penduduk Menggunakan Metode Xgboost," *Indones. J. Appl. Technol.*, Vol. 1, No. 2, P. 18, 2024, Doi: 10.47134/Ijat.V1i2.3045.

[14] H. L. Wati, N. Anggraeni, S. Kolbiah, U. Hendar, And N. Agustina, "Perbandingan Algoritma Random Forest Dan Naratif : Jurnal Ilmiah Nasional Riset Aplikasi Dan Teknik Informatika," Vol. 07, No. 01, Pp. 64–71, 2025.

[15] A. M. Majid *Et Al.*, "Komparasi Algoritma Klasifikasi Machine Learning Dan Penerapan Metode Ensemble Stacking Untuk Menganalisis Sentimen Kesehatan Mental," Vol. 8, Pp. 293–304, 2025.

[16] A. Priandika And A. R. Isnain, "Application Of Ensemble Learning Technique For Classification Of Anemia Types Penerapan Teknik Ensemble Learning Untuk," Vol. 5, No. July, Pp. 972–980, 2025.

[17] Sudarto And Kusrini, "Klasifikasi Tsunami Gempa Bumi Dengan Teknik Stacking Ensemble Machine Learning," 2023.

[18] A. F. Nugraha, R. Faticha, A. Aziza, And Y. Pristyanto, "Penerapan Metode Stacking Dan Random Forest Untuk Meningkatkan Kinerja Klasifikasi Pada Proses Deteksi Web Phishing," Vol. 7, No. 1, 2022.

[19] M. Hermansyah, A. Saikhu, And B. Amaliah, "Pemodelan Data Radiosonde Menggunakan Stacking Ensemble Untuk Klasifikasi Hujan," Vol. 10, No. 2, Pp. 1678–1687, 2025.

[20] J. T. Santoso, *Learning Python*, Vol. 78, No. 1. 2007. Doi: 10.1016/0019-1035(89)90077-8.

[21] O. Siboro, Y. Pricia Banjarnahor, A. Gultom, N. Antonius Siagian, And P. D. Silitonga, "Penanganan Data Ketidakseimbangan Dalam Pendekatan Smote Guna Meningkatkan Akurasi Algoritma K-Nn 1," *Snistik Semin. Nas. Inov. Sains Teknol. Inf. Komput.*, Vol. 1, No. 2, Pp. 473–478, 2024, [Online]. Available: <Https://Ejournal.Ust.Ac.Id/Index.Php/Snistik/Article/View/3705>

[22] A. K. Putri And Hari Suparwito, "Uji Algoritma Stacking Ensemble Classifier Pada Kemampuan Adaptasi Mahasiswa Baru Dalam Pembelajaran Online," *Konstelasi Konvergensi Teknol. Dan Sist. Inf.*, Vol. 3, No. 1, Pp. 1–12, 2023, Doi: 10.24002/Konstelasi.V3i1.7009.

[23] D. Fabiyanto And Z. Pratama Putra, "Validasi Efektivitas Logistic Regression Untuk Diagnosa Penyakit Jantung Melalui Pendekatan Machine Learning," *J. Ilm. Fifo*, Vol. 16, No. 2, P. 158, 2024, Doi: 10.22441/Fifo.2024.V16i2.006.

[24] M. Fadli And R. A. Saputra, "Klasifikasi Dan Evaluasi Performa Model Random Forest Untuk Prediksi Stroke Classification And Evaluation Of Performance Models Random Forest For Stroke Prediction," *J. Tek.*, Vol. 12, No. 2, Pp. 72–80, 2023, [Online]. Available: <Http://Jurnal.Umt.Ac.Id/Index.Php/Jt/Index>

[25] A. F. Azmi And A. Voutama, "Prediksi Churn Nasabah Bank Menggunakan Klasifikasi Random Forest Dan Decision Tree Dengan Evaluasi Confusion Matrix," *Komputa J. Ilm. Komput. Dan Inform.*, Vol. 13, No. 1, Pp. 111–119, 2024, Doi: 10.34010/Komputa.V13i1.12639.