Jurnal Riset Matematika (JRM)
e-ISSN 2798-6306 | p-ISSN 2808-313X
https://journals.unisba.ac.id/index.php/JRM

Tersedia secara online di
Unisba Press
https://publikasi.unisba.ac.id/

Machine Learning pada Prediksi Kelulusan Mahasiswa Menggunakan Algoritma *Random Forest*

Maurino Putra, Erwin Harahap*

Prodi Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Islam Bandung, Indonesia.

A R T I C L E I N F O

Article history :
Received : 2/10/2024
Revised : 28/12/2024
Published : 31/12/2024

Volume : 4
No. : 2
Halaman : 127 - 136
Terbitan : **Desember 2024**

Terakreditasi Sinta Peringkat 5 berdasarkan Ristekdikti No. 177/E/KPT/2024

A B S T R A K

Kelulusan tepat waktu adalah indikator penting dalam menilai kualitas perguruan tinggi karena mencerminkan efektivitas proses pembelajaran dan mempengaruhi reputasi serta akreditasi institusi. Penelitian ini bertujuan memprediksi kelulusan mahasiswa menggunakan algoritma *random forest* yang diimplementasikan melalui aplikasi web berbasis Streamlit Python. Data diperoleh dari platform Kaggle dan diolah melalui proses *pre-processing* untuk memastikan kualitas data yang siap digunakan. Data tersebut kemudian dibagi menjadi data *training* dan *testing* untuk membangun model prediksi. Algoritma *random forest* dipilih karena merupakan sebuah metode *ensemble* atau gabungan dari banyak model CART (*Classification and Regression Tree*) sehingga dapat meningkatkan akurasi hasil prediksinya. Hasil penelitian menunjukkan model memiliki akurasi 88%, *precision* 81%, *recall/sensitivity* 97%, dan *specificity* 80% dalam memprediksi kelulusan mahasiswa. Faktor signifikan yang mempengaruhi kelulusan adalah status mahasiswa berdasarkan *variable importance*. Aplikasi web yang dikembangkan memudahkan prediksi status kelulusan mahasiswa, sehingga dapat digunakan sebagai alat bantu bagi institusi pendidikan dalam pengambilan keputusan terkait kelulusan mahasiswa.

Kata Kunci : Prediksi Kelulusan, Random Forest, Streamlit.

A B S T R A C T

On-time graduation is a crucial indicator in assessing the quality of higher education institutions as it reflects the effectiveness of the learning process and impacts the institution's reputation and accreditation. This study aims to predict student graduation using the random forest algorithm, implemented through a web application based on Streamlit Python. The data was obtained from the Kaggle platform and processed through pre-processing to ensure the quality of the data was ready for use. The data was then split into training and testing data to build the predictive model. The random forest algorithm was chosen because it is an ensemble method, combining many CART (Classification and Regression Tree) models, which can improve prediction accuracy. The research results showed that the model has an accuracy of 88%, precision of 81%, recall/sensitivity of 97%, and specificity of 80% in predicting student graduation. The significant factor influencing graduation is the student's status based on variable importance. The developed web application facilitates the prediction of student graduation status, making it a useful tool for educational institutions in making decisions related to student graduation.

Keywords : Graduation Prediction, Random Forest, Streamlit.

Corresponding Author : Email : erwin2h@unisba.ac.id
Indexed : Garuda, Crossref, Google Scholar
DOI : https://doi.org/10.29313/jrm.v4i2.5102

A. Pendahuluan

Kelulusan tepat waktu merupakan salah satu indikator penting dalam menilai kualitas perguruan tinggi. Tingkat kelulusan yang tinggi tidak hanya mencerminkan efektivitas proses pembelajaran tetapi juga mempengaruhi reputasi dan akreditasi perguruan tinggi tersebut [1]. Dalam konteks akreditasi, ketepatan waktu kelulusan mahasiswa menjadi salah satu kriteria penilaian utama, seperti yang dinyatakan oleh Badan Akreditasi Nasional Perguruan Tinggi (BAN-PT) dan berbagai Lembaga Akreditasi Mandiri (LAM) [2].

Mempertahankan eksistensi suatu perguruan tinggi dapat dilakukan dengan berbagai cara, salah satunya dengan mengoptimalkan kinerja mahasiswa agar lulus tepat waktu atau bahkan lebih cepat [16]. Jika banyak mahasiswa menyelesaikan studi dalam jangka waktu yang ditentukan (4 tahun), maka penilaian kriteria lama studi untuk akreditasi program studi akan baik. Ini menjadi salah satu standar penilaian perguruan tinggi dalam mendidik mahasiswa dan menunjang kegiatan akademik [3]. Pengambilan keputusan tidak cukup hanya mengandalkan tindakan perbaikan setelah masalah muncul, tetapi juga diperlukan tindakan preventif dengan melakukan analisis data dan klasifikasi atau prediksi untuk mengetahui pola data dan mengubahnya menjadi informasi yang bermanfaat [17].

Setiap institusi pendidikan tinggi memiliki basis data yang menyimpan informasi akademik dan biodata mahasiswa. Dengan analisis data, pola dan perilaku mahasiswa dapat dipahami untuk meminimalkan keterlambatan kelulusan [4]. *Machine learning* yang andal memiliki potensi untuk membantu institusi dalam pengambilan keputusan kebijakan. Salah satu metode yang digunakan dalam proses optimasi model *machine learning* adalah *hyperparameter tuning*, yang bertujuan untuk menemukan kombinasi parameter terbaik guna meningkatkan performa model [5]. Algoritma *random forest* digunakan dalam penelitian ini karena keunggulannya dalam meningkatkan akurasi prediksi kelulusan [6][18]. Metode ini juga mampu mengidentifikasi variabel penting (*variable importance*) yang mempengaruhi kelulusan mahasiswa serta evaluasi performanya dilakukan melalui *confusion matrix* untuk menilai performa model. Untuk memudahkan penggunaannya, aplikasi web dibuat menggunakan Streamlit yang dapat diakses secara *online* melalui *browser* [7].

Berdasarkan latar belakang yang telah diuraikan, penulis mengidentifikasi beberapa masalah utama yaitu: pertama, faktor-faktor apa saja yang secara signifikan mempengaruhi kelulusan mahasiswa tepat waktu atau terlambat; kedua, bagaimana cara memprediksi kelulusan mahasiswa dengan model yang memiliki performa optimal; dan ketiga, bagaimana cara mengembangkan alat yang efektif dan efisien digunakan untuk memprediksi status kelulusan mahasiswa berdasarkan data akademik dan demografi. Selanjutnya, tujuan dalam penelitian ini yaitu mengidentifikasi dan menganalisis faktor-faktor yang mempengaruhi kelulusan mahasiswa dengan menggunakan nilai *variable importance* dari model *random forest*, menerapkan algoritma *machine learning*, yaitu *random forest*, untuk memprediksi kelulusan mahasiswa dengan melakukan *hyperparameter tuning* untuk mengoptimalkan performa model, serta mengembangkan aplikasi web berbasis Streamlit yang dapat digunakan oleh pengguna untuk memprediksi status kelulusan mahasiswa dengan memasukkan data akademik dan demografi, serta menampilkan hasil prediksi dengan cara yang mudah dipahami.

B. Metode Penelitian

Jenis dan Sumber Data

Penelitian ini menggunakan data sekunder "Kelulusan Mahasiswa" dari Kaggle (https://www.kaggle.com/datasets/hafizhathallah/kelulusan-mahasiswa), yang berisi informasi tentang performa akademik mahasiswa dan faktor-faktor yang mempengaruhi kelulusan. Dataset ini dipilih karena relevansinya dengan topik prediksi kelulusan mahasiswa.

Variabel Penelitian

Penelitian ini melibatkan beberapa variabel. Berikut adalah Tabel 1 yang berisi variabel penelitian beserta definisi operasional masing-masing variabel.

Tabel 1. Definisi Operasional Variabel

No.	Variabel	Jenis Variabel	Definisi Operasional Variabel	Skala
1	Jenis kelamin	Independen	Jenis kelamin mahasiswa, yaitu laki-laki dan perempuan.	Kategorik
2	Status Mahasiswa	Independen	Status pekerjaan mahasiswa, yaitu bekerja atau tidak bekerja.	Kategorik
3	Status Nikah	Independen	Status pernikahan mahasiswa, yaitu menikah atau belum menikah.	Kategorik
4	IPS 1	Independen	Indeks Prestasi Semester mahasiswa pada semester 1.	Numerik
5	IPS 2	Independen	Indeks Prestasi Semester mahasiswa pada semester 2.	Numerik
6	IPS 3	Independen	Indeks Prestasi Semester mahasiswa pada semester 3.	Numerik
7	IPS 4	Independen	Indeks Prestasi Semester mahasiswa pada semester 4.	Numerik
8	IPS 5	Independen	Indeks Prestasi Semester mahasiswa pada semester 5.	Numerik
9	Status kelulusan	Dependen	Status kelulusan mahasiswa, apakah tepat waktu atau terlambat.	Kategorik

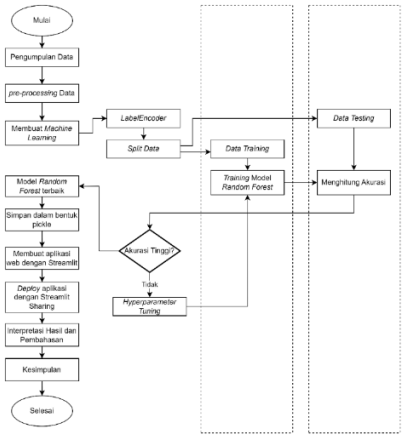
Metode Analisis Data

Dalam penelitian ini, metode analisis data melibatkan beberapa tahapan. Pertama, data sekunder dikumpulkan dari platform Kaggle dengan variabel seperti NAMA, JENIS KELAMIN, STATUS MAHASISWA, UMUR, STATUS NIKAH, IPS 1, IPS 2, IPS 3, IPS 4, IPS 5, IPS 6, IPS 7, IPS 8, IPK, dan STATUS KELULUSAN. Data kemudian diproses untuk mengatasi *missing values* dan duplikasi, serta dibagi menjadi data pelatihan dan data pengujian.

Algoritma *random forest* digunakan untuk memilih variabel penting dan membangun model prediksi [19]. Model ini dievaluasi dengan metrik *accuracy*, *precision*, *recall*/*sensitivity*, dan *specificity* [20]. Aplikasi web dikembangkan dengan Streamlit di Python untuk memudahkan penggunaan model prediksi. Penelitian ini bertujuan menghasilkan model prediksi kelulusan mahasiswa yang akurat dan aplikasi web yang berguna untuk meningkatkan kualitas pendidikan di Indonesia.

Diagram Alir Penelitian

Tahapan yang dilakukan pada penelitian ini dapat digambarkan melalui diagram alir pada Gambar 1 berikut.



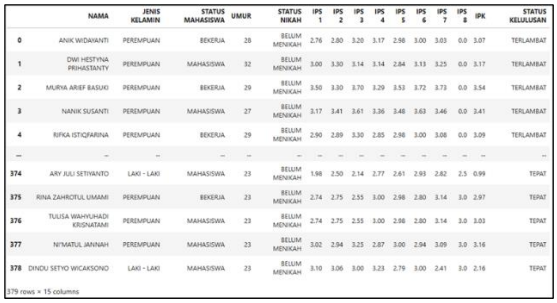
Gambar 1. Diagram Alir Penelitian

C. Hasil dan Pembahasan

Pre-Processing Data

Sebelum tahap pemodelan dilakukan *pre-processing* data. *Pre-processing* data dilakukan untuk memastikan bahwa data siap digunakan dalam model *machine learning*. Namun, Data yang diunduh dari platform Kaggle masih dalam

bentuk file excel yang tidak beraturan (Gambar 2). Oleh karena itu, peneliti melakukan cleaning data dengan membuang variabel atau kolom yang tidak diperlukan, seperti NAMA, UMUR, IPS 6, IPS 7, IPS 8, dan IPK.

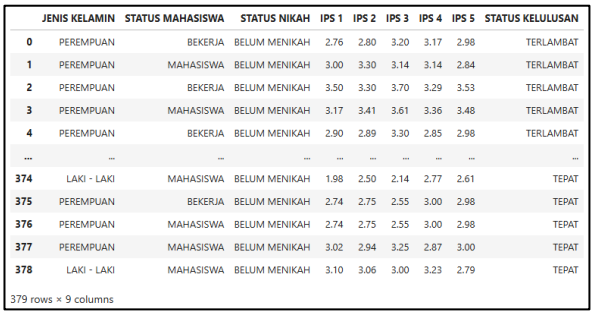


	NAMA	JENIS KELAMIN	STATUS MAHASISWA	UMUR	STATUS NIKAH	IPS 1	IPS 2	IPS 3	IPS 4	IPS 5	IPS 6	IPS 7	IPS 8	IPK	STATUS KELULUSAN
0	ANIK WIDAYANTI	PEREMPUAN	BEKERJA	28	BELUM MENIKAH	2.76	2.80	3.20	3.17	2.98	3.00	3.03	0.0	3.07	TERLAMBAT
1	DWI HESTYNA PRIHASTANTY	PEREMPUAN	MAHASISWA	32	BELUM MENIKAH	3.00	3.30	3.14	3.14	2.84	3.13	3.25	0.0	3.17	TERLAMBAT
2	MURYA ARIEF BASUKI	PEREMPUAN	BEKERJA	29	BELUM MENIKAH	3.50	3.30	3.70	3.29	3.53	3.72	3.73	0.0	3.54	TERLAMBAT
3	NANIK SUSANTI	PEREMPUAN	MAHASISWA	27	BELUM MENIKAH	3.17	3.41	3.61	3.36	3.48	3.63	3.46	0.0	3.41	TERLAMBAT
4	RIFKA ISTIQFARINA	PEREMPUAN	BEKERJA	29	BELUM MENIKAH	2.90	2.89	3.30	2.85	2.98	3.00	3.08	0.0	3.09	TERLAMBAT
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
374	ARY JULI SETIYANTO	LAKI - LAKI	MAHASISWA	23	BELUM MENIKAH	1.98	2.50	2.14	2.77	2.61	2.93	2.82	2.5	0.99	TEPAT
375	RINA ZAHROTUL UMAMI	PEREMPUAN	BEKERJA	23	BELUM MENIKAH	2.74	2.75	2.55	3.00	2.98	2.80	3.14	3.0	2.97	TEPAT
376	TULISA WAHYUHADI KRISNATAMI	PEREMPUAN	MAHASISWA	23	BELUM MENIKAH	2.74	2.75	2.55	3.00	2.98	2.80	3.14	3.0	3.03	TEPAT
377	NI'MATUL JANNAH	PEREMPUAN	MAHASISWA	23	BELUM MENIKAH	3.02	2.94	3.25	2.87	3.00	2.94	3.09	3.0	3.16	TEPAT
378	DINDU SETYO WICAKSONO	LAKI - LAKI	MAHASISWA	23	BELUM MENIKAH	3.10	3.06	3.00	3.23	2.79	3.00	2.41	3.0	2.16	TEPAT

379 rows × 15 columns

Gambar 2. Data Sebelum *Pre-processing*

Penghapusan variabel NAMA, UMUR, IPS 6, IPS 7, IPS 8, dan IPK dalam penelitian ini dilakukan karena alasan ilmiah. NAMA dihapus karena tidak relevan dalam prediksi dan untuk menghindari masalah privasi [8], [9]. UMUR dihapus karena tidak ada keterangan jelas dan usia bukan indikator signifikan dalam hasil akademis [10]. IPS 6, IPS 7, IPS 8, dan IPK dihapus karena data IPS 5 sudah cukup untuk mencegah mahasiswa telat lulus, mengurangi redundansi, dan meningkatkan efisiensi model prediksi [11], [12], [13]. Gambar 3 menunjukkan data setelah penghapusan variabel ini.

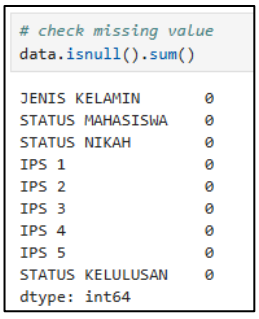


	JENIS KELAMIN	STATUS MAHASISWA	STATUS NIKAH	IPS 1	IPS 2	IPS 3	IPS 4	IPS 5	STATUS KELULUSAN
0	PEREMPUAN	BEKERJA	BELUM MENIKAH	2.76	2.80	3.20	3.17	2.98	TERLAMBAT
1	PEREMPUAN	MAHASISWA	BELUM MENIKAH	3.00	3.30	3.14	3.14	2.84	TERLAMBAT
2	PEREMPUAN	BEKERJA	BELUM MENIKAH	3.50	3.30	3.70	3.29	3.53	TERLAMBAT
3	PEREMPUAN	MAHASISWA	BELUM MENIKAH	3.17	3.41	3.61	3.36	3.48	TERLAMBAT
4	PEREMPUAN	BEKERJA	BELUM MENIKAH	2.90	2.89	3.30	2.85	2.98	TERLAMBAT
…	…	…	…	…	…	…	…	…	…
374	LAKI - LAKI	MAHASISWA	BELUM MENIKAH	1.98	2.50	2.14	2.77	2.61	TEPAT
375	PEREMPUAN	BEKERJA	BELUM MENIKAH	2.74	2.75	2.55	3.00	2.98	TEPAT
376	PEREMPUAN	MAHASISWA	BELUM MENIKAH	2.74	2.75	2.55	3.00	2.98	TEPAT
377	PEREMPUAN	MAHASISWA	BELUM MENIKAH	3.02	2.94	3.25	2.87	3.00	TEPAT
378	LAKI - LAKI	MAHASISWA	BELUM MENIKAH	3.10	3.06	3.00	3.23	2.79	TEPAT

379 rows × 9 columns

Gambar 3. Data Setelah Penghapusan Variabel yang Tidak Diperlukan

Setelah menghapus variabel-variabel yang tidak diperlukan, dilakukan pengecekan ada tidaknya *missing value*. Data yang terdapat *missing value* akan mempengaruhi tingkat akurasi dalam model *machine learning*. Pengecekan *missing value* pada data dapat dilihat pada Gambar 4 berikut.



```
# check missing value
data.isnull().sum()
```

```
JENIS KELAMIN       0
STATUS MAHASISWA    0
STATUS NIKAH        0
IPS 1               0
IPS 2               0
IPS 3               0
IPS 4               0
IPS 5               0
STATUS KELULUSAN    0
dtype: int64
```

Gambar 4. Cek *Missing Value* Data Setiap Variabel Data

Berdasarkan Gambar 4 diketahui bahwa untuk setiap variabel, yaitu JENIS KELAMIN, STATUS MAHASISWA, STATUS NIKAH, IPS 1, IPS 2, IPS 3, IPS 4, IPS 5, dan STATUS KELULUSAN tidak

terdapat *missing value*. Setelah melakukan pengecekan *missing value* selanjutnya melakukan pengecekan terhadap data duplikat.

```
# cek duplikasi pada data
data.duplicated().sum()

1
```

Gambar 5. Cek Data Duplikat

```
# Hapus data duplikat
data = data.drop_duplicates()

# cek duplikasi pada data
data.duplicated().sum()

0
```

Gambar 6. Hapus Data Duplikat

Berdasarkan Gambar 5 diketahui bahwa terdapat satu data yang duplikat. Maka dilakukan penghapusan terhadap data duplikat sebagaimana Gambar 6. Setelah melakukan pengecekan *missing value* dan data duplikat, Proses selanjutnya adalah melakukan transformasi atau *labelling* data pada variabel kategorikal, yaitu JENIS KELAMIN, STATUS MAHASISWA, STATUS NIKAH, dan STATUS KELULUSAN menjadi tipe data numerik. Hal ini dilakukan agar algoritma *random forest* dapat melakukan komputasi terhadap data. *Labelling* dilakukan dengan fungsi *LabelEncoder* yang terdapat pada *module sklearn.preprocessing*. Gambar 7 adalah hasil transformasi atau *labelling* data pada variabel kategorikal.

	JENIS KELAMIN	STATUS MAHASISWA	STATUS NIKAH	IPS 1	IPS 2	IPS 3	IPS 4	IPS 5	STATUS KELULUSAN
0	1	0	0	2.76	2.80	3.20	3.17	2.98	1
1	1	1	0	3.00	3.30	3.14	3.14	2.84	1
2	1	0	0	3.50	3.30	3.70	3.29	3.53	1
3	1	1	0	3.17	3.41	3.61	3.36	3.48	1
4	1	0	0	2.90	2.89	3.30	2.85	2.98	1
...	...	...	...	...	...	...	...	...	...
374	0	1	0	1.98	2.50	2.14	2.77	2.61	0
375	1	0	0	2.74	2.75	2.55	3.00	2.98	0
376	1	1	0	2.74	2.75	2.55	3.00	2.98	0
377	1	1	0	3.02	2.94	3.25	2.87	3.00	0
378	0	1	0	3.10	3.06	3.00	3.23	2.79	0

378 rows × 9 columns

Gambar 7. Data Hasil *Labelling*

Pembagian Data *Training* dan Data *Testing*

Sebelum melakukan pemodelan dengan menggunakan klasifikasi *random forest*, langkah pertama yang harus dilakukan adalah membagi data menjadi dua bagian: data *training* dan data *testing*. Langkah ini bertujuan untuk mengukur kinerja model dengan mengevaluasi kesalahan prediksi yang mungkin terjadi. Data *training* digunakan untuk melatih algoritma dan membentuk model, sedangkan data *testing* digunakan untuk menguji keakuratan model yang telah dibuat. Jika performa yang dihasilkan oleh model tersebut tinggi, maka model tersebut dapat diandalkan untuk melakukan prediksi pada data baru. Data *training* dan data *testing* dibagi dengan proporsi 80% untuk data *training* dan 20% untuk data *testing* dari total *dataset*.

Tabel 2. Proporsi Data *Training* dan Data *Testing*

Keterangan	Data *Training*	Data *Testing*	Total
Proporsi	80%	20%	100%
Jumlah	302	76	378

Berdasarkan pada Tabel 2 diketahui bahwa 378 dataset yang ada, pembagian data untuk data *training* sebanyak 302 data dengan persentase masing- masing kelas adalah kelas 0 (TEPAT) adalah sebanyak 180 (59,6%) dan kelas 1 (TERLAMBAT) sebanyak 122 (40,4%). Sedangkan untuk data *testing*, ada sebanyak 76 data dengan persentase masing-masing kelas adalah kelas 0 (TEPAT) adalah sebanyak 41 (53,9%) dan kelas 1 (TERLAMBAT) sebanyak 35 (46,1%). Pembagian data *training* dan data *testing* pada dataset dilakukan secara *random* dengan bantuan *software* Python.

Implementasi *Random Forest* Pada Python

Setelah membagi data menjadi data *training* dan data *testing*, langkah selanjutnya adalah melakukan analisis klasifikasi menggunakan *random forest* pada data sampel *training* yang telah ditentukan. Variabel dependen dalam penelitian ini adalah STATUS KELULUSAN yang akan diprediksi, sementara variabel JENIS KELAMIN, STATUS MAHASISWA, STATUS NIKAH, IPS 1, IPS 2, IPS 3, IPS 4, dan IPS 5 berperan sebagai variabel independen.

Langkah pertama dalam membangun model klasifikasi *random forest* adalah melakukan *hyperparameter tuning* menggunakan fungsi *gridsearchCV* pada *scikit-learn*. Parameter yang digunakan dalam penelitian ini adalah hasil modifikasi dari parameter penelitian sebelumnya [14], [15], [16]. Hasil *hyperparameter tuning* ditampilkan pada Tabel 3.

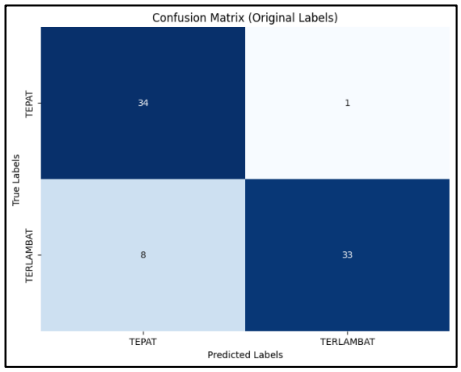
Tabel 3. *Hyperparameter Tuning*

Parameter	Deskripsi	*Grid Search Values*	Parameter Terbaik
n_estimators	Jumlah pohon	100, 200, 500	500
max_depth	Maksimum kedalaman dari pohon	2, 5, 6, 7, 8	5
Criterion	Metrik yang akan digunakan untuk menentukan simpul yang akan dipisah: a. jika *gini*, maka menggunakan *gini indeks*. b. jika *entropy*, maka menggunakan *entropy*	*Entropy*, *Gini*	*Entropy*
min_samples_leaf	Jumlah sampel minimum yang dibutuhkan *leaf node*.	1, 2, 4	4
max_features	Jumlah fitur yang dipertimbangkan saat mencari *split* terbaik: a. Jika auto, maka max_$features = \sqrt{jumlah\ fitur}$ b. Jika sqrt, maka max_$features = \sqrt{jumlah\ fitur}$ c. Jika log 2, maka max_$features = \log_2(jumlah\ fitur)$	*Auto*, *sqrt*, log 2	log2
min_samples_split	Jumlah minimum sampel yang diperlukan untuk memisahkan node internal.	2, 5, 10	10

Berdasarkan Tabel 3, hasil *tuning parameter* diperoleh melalui proses *grid searchCV* dengan pencarian menyeluruh terhadap parameter yang diujikan. Penelitian ini menggunakan 10-*fold cross validation* untuk mengevaluasi kinerja model dengan sepuluh kali pengulangan dalam proses *grid searchCV* untuk setiap parameter. Nilai parameter terbaik dari proses *grid searchCV* akan digunakan dalam penentuan model klasifikasi.

Hasil *hyperparameter tuning* menunjukkan parameter terbaik untuk model random forest dengan kriteria akurasi. 'Criterion' menggunakan 'entropy', 'max_depth' diatur ke 5, 'max_features' menggunakan 'log2', 'min_samples_leaf' diatur ke 4, 'min_samples_split' diatur ke 10, dan 'n_estimators' ditetapkan ke 500. Parameter-parameter ini membantu membentuk model *random forest* yang optimal, mengurangi *overfitting*, dan meningkatkan performa.

Setelah model diperoleh, langkah selanjutnya adalah evaluasi model. Langkah ini bertujuan untuk mengidentifikasi keakuratan model. Ukuran yang digunakan untuk mengevaluasi hasil prediksi model meliputi nilai *accuracy*, *precision*, *recall/sensitivity*, dan *specificity* dengan menggunakan *confusion matrix*.



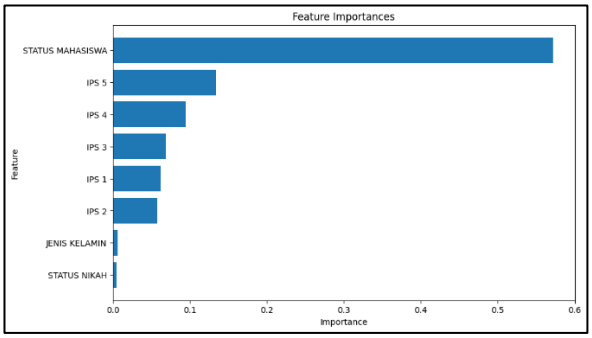
Gambar 8. *Confusion Matrix Random Forest*

Berdasarkan Gambar 8, kelas 0 merepresentasikan "TEPAT" dan kelas 1 merepresentasikan "TERLAMBAT". Dari 76 sampel yang diuji, model menghasilkan akurasi 88%, dengan 67 sampel terklasifikasi dengan benar. Nilai *precision* sebesar 81% menunjukkan proporsi prediksi tepat dari total prediksi positif, sedangkan *recall* sebesar 97% menunjukkan kemampuan model dalam mendeteksi mahasiswa yang lulus tepat waktu. *Specificity* tercatat sebesar 80%, menunjukkan klasifikasi yang benar pada kelas "TERLAMBAT". Metrik evaluasi yang menunjukkan keandalan model *random forest* dalam memprediksi kelulusan mahasiswa disajikan pada Tabel 4.

Tabel 4. Hasil Matriks Evaluasi *Random Forest*

Accuracy	*Precision*	*Recall/Sensitivity*	*Specificity*
88%	81%	97%	80%

Selanjutnya adalah mengukur kepentingan variabel independen terhadap status kelulusan mahasiswa berdasarkan nilai *features/variables importance*.



Gambar 9. *Features Importance*

Gambar 9 menunjukkan seberapa penting variabel dalam pembentukan model dan prediksi. Semakin tinggi nilai *features importance*, semakin besar pengaruh variabel tersebut terhadap hasil prediksi. Variabel STATUS MAHASISWA paling berpengaruh dalam memprediksi STATUS KELULUSAN, sedangkan STATUS NIKAH memiliki pengaruh paling kecil. *Features importance* ditentukan melalui perhitungan *information gain* menggunakan *entropy* atau *gini*. Nilai *features importance* tiap variabel dapat dilihat pada Tabel 5.

Tabel 5. *Features Importance*

No.	Variabel	*Features Importance*
1	STATUS MAHASISWA	0.571698
2	IPS 5	0.133935
3	IPS 4	0.094826

No.	Variabel	*Features Importance*
4	IPS 3	0.069220
5	IPS 1	0.061810
6	IPS 2	0.057035
7	JENIS KELAMIN	0.006479
8	STATUS NIKAH	0.004996

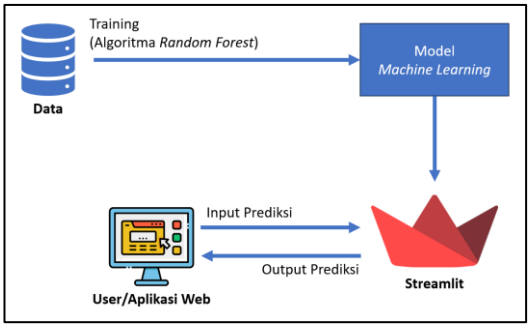
Setelah analisis selesai dilakukan, model yang telah diperoleh kemudian disimpan untuk pengembangan aplikasi web, pada Python disimpan dengan perintah *pickle* sebagaimana Gambar 10. Model ini efektif dan efisien untuk memprediksi suatu data karena penulis hanya memanggil model tanpa melakukan analisis yang sebelumnya dilakukan. Model ini akan digunakan dalam aplikasi Streamlit.

```
# Save the trained model to a file
filename = 'trained_model.pkl'
pickle.dump(rf_model, open(filename, 'wb'))
```

Gambar 10. Menyimpan Model yang Telah Dilatih

Rancangan Sistem

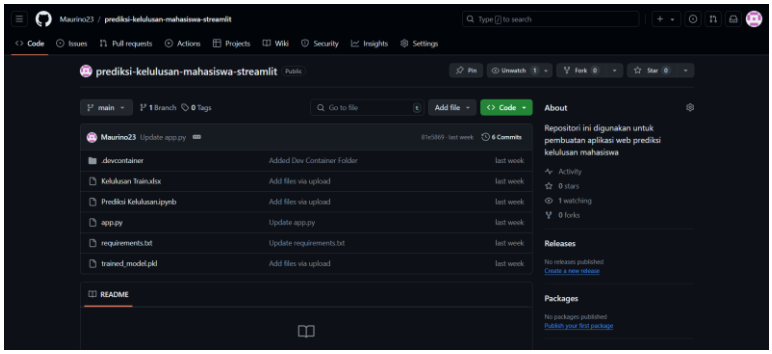
Penelitian ini bertujuan untuk mengembangkan aplikasi web yang dapat memprediksi kelulusan mahasiswa menggunakan algoritma *random forest* dan platform Streamlit. Sistem ini terdiri dari beberapa komponen utama, yaitu: dataset kelulusan mahasiswa, model *machine learning random forest*, dan aplikasi web Streamlit. Gambar 11 adalah arsitektur sistem.



Gambar 11. Arsitektur Sistem

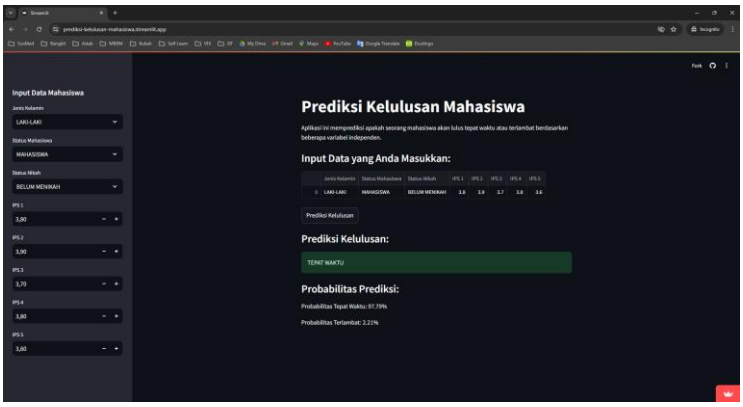
Membangun Aplikasi Web Menggunakan Streamlit

Model terbaik digunakan sebagai prototipe aplikasi untuk memprediksi kelulusan mahasiswa, termasuk probabilitasnya. Prediksi dibuat dengan model *random forest* menggunakan variabel seperti jenis kelamin, status mahasiswa, status nikah, dan IPS 1-5. Aplikasi Streamlit dikembangkan dengan file 'app.py', memuat model, judul, deskripsi, dan input pengguna. Setelah input dimasukkan, tabel data ditampilkan dan hasil prediksi muncul saat tombol ditekan. Aplikasi dijalankan dengan 'streamlit run app.py'. Untuk *deployment*, Streamlit Sharing digunakan dengan mengunggah file ke *repository* GitHub, seperti ditunjukkan pada Gambar 12.



Gambar 12. *Repository* Prediksi-Kelulusan-Mahasiswa-Streamlit Github

Selanjutnya, akun didaftarkan pada Streamlit Sharing, hubungkan dengan GitHub, pilih *repository*, lalu klik "Deploy" untuk memulai deployment. Setelah selesai, URL publik akan diberikan untuk akses aplikasi (https://prediksi-kelulusan-mahasiswa.streamlit.app/). Gambar 13 menampilkan aplikasi web yang berhasil di-*deploy*.



Gambar 13. Tampilan Aplikasi yang Telah Berhasil di-*deploy*

D. Kesimpulan

Berdasarkan pembahasan dalam penelitian ini, peneliti menyimpulkan beberapa hasil penelitian yaitu faktor-faktor yang secara signifikan mempengaruhi kelulusan tepat waktu atau terlambat seorang mahasiswa telah diidentifikasi berdasarkan nilai *variable importance*. Faktor utama adalah status mahasiswa, diikuti oleh IPS semester 5, IPS semester 4, IPS semester 3, IPS semester 1, IPS semester 2, jenis kelamin, dan status pernikahan. Prediksi kelulusan seorang mahasiswa dapat dilakukan dengan algoritma *random forest* yang telah dioptimalkan melalui *hyperparameter tuning*. Model ini menghasilkan akurasi 88%, *precision* 81%, *recall* 97%, dan *specificity* 80%. Serta, Aplikasi web untuk memprediksi kelulusan mahasiswa telah dikembangkan dan dapat diakses melalui https://prediksi-kelulusan-mahasiswa.streamlit.app/. Aplikasi ini memudahkan pengguna dengan menyediakan bagian input, tampilan data, tombol prediksi, hasil prediksi yang diberi warna untuk menunjukkan status kelulusan (hijau untuk "TEPAT WAKTU" dan merah untuk "TERLAMBAT"), dan probabilitas prediksi.

Daftar Pustaka

[1] Mahmud Basuki, "Kontribusi Mahasiswa Dalam Akreditasi Program Studi," NUSANTARA Jurnal Pengabdian Kepada Masyarakat, vol. 3, no. 2, pp. 48–54, Apr. 2023, doi: 10.55606/nusantara.v3i2.1038.

[2] D. Handini, "Peralihan Akreditasi Program Studi dari BAN-PT kepada Lima Lembaga Akreditasi Mandiri (LAM) Baru," https://dikti.kemdikbud.go.id/.

[3] A. Azahari, Y. Yulindawati, D. Rosita, and S. Mallala, "Komparasi Data Mining Naive Bayes dan Neural Network memprediksi Masa Studi Mahasiswa S1," Jurnal Teknologi Informasi dan Ilmu Komputer, vol. 7, no. 3, pp. 443–452, May 2020, doi: 10.25126/jtiik.2020732093.

[4] M. Mubarok, M. Muliadi, and R. Herteno, "Hyper-parameter Tuning pada XGBOOST Untuk Prediksi Keberlangsungan Hidup Pasien Gagal Jantung," Kumpulan Jurnal Ilmu Komputer, vol. 9, no. 2, 2022.
[5] I. Mulyahati, "Implementasi Machine Learning Prediksi Harga Sewa Apartemen Menggunakan Algoritma Random Forest Melalui Framework Website Flask Python (Studi Kasus: Apartemen di DKI Jakarta Pada Website mamikos.com)," Universitas Islam Indonesia, Yogyakarta, 2020.
[6] A. Suryadi, E. Harahap, and A. Rachmanto, "Rancang Bangun Sistem Informasi Persediaan Obat Berbasis Web DI Apotek XYZ," Jurnal PETIK, vol. 4, no. 2, Sep. 2018.
[7] M. Doshi and S. K. Chaturvedi, "Correlation Based Feature Selection (CFS) Technique to Predict Student Perfromance," International journal of Computer Networks & Communications, vol. 6, no. 3, pp. 197–206, May 2014, doi: 10.5121/ijcnc.2014.6315.
[8] S. Helal, J. Li, L. Liu, E. Ebrahimie, S. Dawson, and D. J. Murray, "Identifying key factors of student academic performance by subgroup discovery," Int J Data Sci Anal, vol. 7, no. 3, pp. 227–245, Apr. 2019, doi: 10.1007/s41060-018-0141-y.
[9] M. Zaffar, M. A. Hashmani, K. S. Savita, S. S. H. Rizvi, and M. Rehman, "Role of FCBF Feature Selection in Educational Data Mining," Mehran University Research Journal of Engineering and Technology, vol. 39, no. 4, pp. 772–778, Oct. 2020, doi: 10.22581/muet1982.2004.09.
[10] L. Rahman, N. A. Setiawan, and A. E. Permanasari, "Feature selection methods in improving accuracy of classifying students' academic performance," in 2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE), IEEE, Nov. 2017, pp. 267–271. doi: 10.1109/ICITISEE.2017.8285509.
[11] W. Xiao, P. Ji, and J. Hu, "RnkHEU: A Hybrid Feature Selection Method for Predicting Students' Performance," Sci Program, vol. 2021, pp. 1–16, Nov. 2021, doi: 10.1155/2021/1670593.
[12] W. Punlumjeak and N. Rachburee, "A comparative study of feature selection techniques for classify student performance," in 2015 7th International Conference on Information Technology and Electrical Engineering (ICITEE), IEEE, Oct. 2015, pp. 425–429. doi: 10.1109/ICITEED.2015.7408984.
[13] D. Asif, M. Bibi, M. S. Arif, and A. Mukheimer, "Enhancing Heart Disease Prediction through Ensemble Learning Techniques with Hyperparameter Optimization," Algorithms, vol. 16, no. 6, p. 308, Jun. 2023, doi: 10.3390/a16060308.
[14] Y. Hu and M. Sokolova, "Explainable Multi-class Classification of Medical Data," Dec. 2020.
[15] M. A. M. Mellal, Design and Control Advances in Robotics. IGI Global, 2022.
[16] A. Khaerunnisa, "Analisis Tingkat Kelulusan Mahasiswa di Unisba dengan menggunakan Algoritma K-Means Clustering," J. Ris. Mat., pp. 67–76, Jul. 2022, doi: 10.29313/jrm.v2i1.1018.
[17] B. Haya Pangestu, "Data Mining Menggunakan Algoritma Naïve Bayes Classifier Untuk Evaluasi Kinerja Karyawan," J. Ris. Mat., pp. 177–184, Dec. 2023, doi: 10.29313/jrm.v3i2.2837.
[18] S. Zein and G. Gunawan, "Prediksi Hasil FIFA World Cup Qatar 2022 Menggunakan Machine Learning dengan Python," J. Ris. Mat., pp. 153–162, Dec. 2022, doi: 10.29313/jrm.v2i2.1382.
[19] Rahadatul Aisyi and Respitawulan, "Implementasi Pemilihan Siswa Berprestasi Menggunakan Metode Preferences Selection Index," J. Ris. Mat., vol. 1, no. 2, pp. 145–153, Feb. 2022, doi: 10.29313/jrm.v1i2.487.
[20] S. A. Savitri and D. Suhaedi, "Penerapan Inference Fuzzy Mamdani dalam Seleksi Penerima Bantuan Sosial Tunai Kabupaten Belitung Timur," J. Ris. Mat., pp. 163–172, Dec. 2022, doi: 10.29313/jrm.v2i2.1383.