Journal : Jurnal Rekayasa Sistem Informasi dan Teknologi (JRSIT)
Publisher : Yayasan Nuraini Ibrahim Mandiri
ISSN : e-ISSN 3025-888X
Volume / Issue / Year/ Page : Vol. 4 No. 1 (2026): Agustus/ Hal 95-107

PERBANDINGAN DECISION TREE DAN RANDOM FOREST BERBASIS GRID SEARCH HYPERPARAMETER TUNING UNTUK PREDIKSI KELAYAKAN BEASISWA

Bentar Candra Perdana[1*], Nikko Restian Putra Wardoyo[2], Elis Anita farida[3]
[1,2]Universitas PGRI Jombang, Indonesia
[3]Universitas Anwar Medika, Indonesia
ar.bent989@gmail.com[1*],ncool750@gmail.com[2]
elisanitafarida@uam.ac.id[3]
***coresponding author**

Information Artikel

Pengajuan	:	28-07-2026
Peninjauan	:	11-08-2026
Revisi	:	21-08-2026
Diterima	:	25-08-2026
Diterbitkan	:	30-08-2026

Kata Kunci
Pohon Keputusan, Hutan Acak, penyetelan hiperparameter, GridSearchCV, beasiswa

Abstrak

Penentuan kelayakan mahasiswa penerima beasiswa merupakan salah satu proses pengambilan keputusan yang membutuhkan pengolahan data secara sistematis agar bantuan dapat diberikan kepada mahasiswa yang sesuai dengan kriteria. Pemanfaatan data mining dan machine learning dapat digunakan untuk menemukan pola dari data mahasiswa dan membangun model prediksi kelayakan penerima beasiswa. Penelitian ini bertujuan untuk membandingkan kinerja algoritma Decision Tree dan Random Forest dalam memprediksi kelayakan mahasiswa penerima beasiswa serta menganalisis pengaruh hyperparameter tuning terhadap performa kedua algoritma. Dataset yang digunakan terdiri atas karakteristik mahasiswa dan kondisi sosial ekonomi keluarga, meliputi jenis kelamin, alamat, program studi yang dipilih, alat transportasi, penghasilan ayah, dan pekerjaan ayah, dengan status layak atau tidak layak menerima beasiswa sebagai variabel target. Tahapan penelitian menggunakan pendekatan data mining yang meliputi data understanding, data preparation, exploratory data analysis, pembagian data, pemodelan, hyperparameter tuning, dan evaluasi. Empat model dibandingkan, yaitu Decision Tree tanpa tuning, Decision Tree dengan hyperparameter tuning, Random Forest tanpa tuning, dan Random Forest dengan hyperparameter tuning. Proses optimasi dilakukan menggunakan pencarian kombinasi hyperparameter dengan cross-validation. Evaluasi model menggunakan accuracy, precision, recall, dan F1-score, serta analisis confusion matrix. Hasil penelitan membuktikan bahwa kedua Algoritma Random Forest dan Decision Tree dengan hyperparameter tuning dapat meingkatkan akurasinya sehingga memberikan performa terbaik. Akan tetapi algoritma random forest dengan hyperparamater tuning menjadi yang paling tinggi nilai akurasinya yaitu sebesar 87.38% dengan F1-score 84%, Precision 85% dan recall 87% dalam memprediksi kelayakan mahasiswa penerima beasiswa serta mengidentifikasi karakteristik yang paling berkontribusi terhadap hasil klasifikasi

How to Cite: Perdana, B. C., Wardoyo, N. R. P., & Farida, E. A. (2026). Perbandingan decision tree dan random forest berbasis grid search hyperparameter tuning untuk prediksi kelayakan beasiswa. *Jurnal Rekayasa Sistem Informasi dan Teknologi (JRSIT),* 4(1), 95–107. https://doi.org/10.70248/jrsit.v4i1.4552

DOI : https://doi.org/10.70248/jrsit.v4i1.4552
This is an open access article under https://creativecommons.org/licenses/by/4.0/
Copyright© 2026 by Author. Published by Yayasan Nuraini Ibrahim Mandiri

PENDAHULUAN

Beasiswa merupakan salah satu bentuk dukungan pendidikan yang dapat membantu mahasiswa dalam memenuhi kebutuhan selama menempuh pendidikan tinggi. Proses penentuan mahasiswa yang layak menerima beasiswa perlu dilakukan secara tepat karena berkaitan dengan penyaluran bantuan kepada mahasiswa yang memenuhi kriteria yang telah ditentukan. Dalam proses seleksi, berbagai karakteristik mahasiswa dan kondisi sosial ekonomi keluarga dapat menjadi dasar dalam

menentukan tingkat kelayakan penerima bantuan. Oleh karena itu, pemanfaatan data mahasiswa secara sistematis dapat menjadi salah satu pendekatan untuk mendukung proses pengambilan keputusan dalam penentuan penerima beasiswa(Madani et al., 2025).

Perkembangan *data mining* dan *machine learning* memberikan peluang untuk memanfaatkan data historis dalam membangun model klasifikasi. Model klasifikasi dapat mempelajari pola dari data yang telah memiliki label, kemudian menggunakan pola tersebut untuk memprediksi kelas pada data baru(Gori et al., 2024). Dalam konteks bantuan pendidikan, pendekatan klasifikasi telah digunakan untuk mengidentifikasi kelayakan calon penerima bantuan berdasarkan karakteristik sosial ekonomi. Setyowati *et al.* menggunakan Random Forest untuk mengklasifikasikan kelayakan penerima Kartu Indonesia Pintar (KIP) berdasarkan faktor ekonomi dengan memanfaatkan data SUSENAS dan melakukan optimasi parameter menggunakan *grid search* (Elim & Utami, 2025). Hasil penelitian tersebut menunjukkan bahwa proses optimasi parameter dapat digunakan untuk memperoleh konfigurasi *Random Forest* yang sesuai untuk permasalahan klasifikasi kelayakan bantuan pendidikan (Amanda & Voutama, 2025).
Salah satu algoritma klasifikasi yang dapat digunakan dalam permasalahan tersebut adalah *Decision Tree*. *Decision Tree* membentuk struktur pohon keputusan berdasarkan proses pemisahan data menggunakan atribut tertentu sehingga menghasilkan aturan yang dapat digunakan untuk menentukan kelas suatu data. Salah satu kelebihan *Decision Tree* adalah hasil model relatif mudah diinterpretasikan karena keputusan dapat ditelusuri melalui struktur pohon. (Sari et al., 2026) Penelitian Sari, Hermawan, dan Avianto pada tahun 2026 menerapkan *Decision Tree* untuk klasifikasi kelayakan bantuan pendidikan menggunakan 4.424 data mahasiswa dengan atribut akademik, administratif, dan sosial ekonomi. Penelitian tersebut melakukan pemeriksaan kualitas data, seleksi fitur, pembagian data, serta evaluasi performa model (Hermawan et al., 2020).

Selain *Decision Tree*, *Random Forest* merupakan algoritma yang relevan untuk digunakan sebagai pembanding*. Random Forest* merupakan metode *ensemble* yang membangun sejumlah *Decision Tree* dan menggabungkan hasil dari pohon-pohon tersebut untuk menghasilkan prediksi (Firnanda et al., 2025). Penggunaan banyak pohon memungkinkan model menangkap pola data dari berbagai proses pembentukan pohon. Dalam penelitian mengenai kelayakan KIP di Jawa Barat, menerapkan *Random Forest* dan melakukan *hyperparameter tuning* menggunakan *grid search*. Parameter yang dioptimalkan antara lain jumlah pohon, jumlah variabel yang dipertimbangkan pada setiap pemisahan, dan ukuran simpul terminal (Setyowati et al., 2025).

Kinerja algoritma machine learning tidak hanya dipengaruhi oleh pemilihan algoritma, tetapi juga oleh konfigurasi parameter yang digunakan. Parameter yang nilainya ditentukan sebelum proses pembelajaran model dikenal sebagai *hyperparameter*. Pemilihan konfigurasi yang tepat dapat dilakukan melalui proses *hyperparameter tuning* dengan mengevaluasi berbagai kombinasi parameter menggunakan metode validasi silang (*cross-validation*). Dokumentasi scikit-learn menjelaskan bahwa teknik pencarian parameter dapat digunakan untuk mengevaluasi sejumlah konfigurasi model berdasarkan data yang ditahan pada proses validasi, sehingga konfigurasi yang memberikan skor terbaik dapat dipilih sebagai model akhir (Ramadhan, M. G. I. 2024).

Penerapan *hyperparameter tuning* menjadi semakin relevan ketika model digunakan untuk permasalahan klasifikasi kelayakan bantuan pendidikan(Surip, A., 2021). Penelitian Setyowati *et al.* menunjukkan penerapan *grid search* untuk

menemukan kombinasi parameter *Random Forest* yang optimal pada klasifikasi kelayakan KIP. Dengan menggunakan data sebanyak 13.044 individu, penelitian tersebut memperoleh kombinasi parameter optimal berupa *mtry* = 5, *ntree* = 674, dan *node_size* = 26 dengan nilai *balanced accuracy* sebesar 65,47% (Setyowati et al., 2025). Hasil tersebut menunjukkan bahwa konfigurasi parameter perlu diperhatikan karena karakteristik dataset dapat memengaruhi performa model klasifikasi.

Penelitian mengenai *Decision Tree* juga telah diterapkan pada permasalahan penerima bantuan pendidikan. Santuni *et al.* pada tahun 2025 menggunakan pendekatan pohon keputusan untuk mengklasifikasikan penerima Program Indonesia Pintar berdasarkan kondisi sosial ekonomi orang tua, lokasi tempat tinggal, dan jumlah tanggungan orang tua. Penelitian tersebut menggunakan 299 data siswa dan memperoleh akurasi sebesar 84,27% pada pengujian model (Surip, A., 2021). Hasil tersebut menunjukkan bahwa algoritma berbasis pohon dapat digunakan untuk membantu proses klasifikasi penerima bantuan berdasarkan karakteristik sosial ekonomi.

Meskipun *Decision Tree* dan *Random Forest* telah digunakan dalam berbagai penelitian mengenai klasifikasi penerima bantuan pendidikan, hasil performa suatu algoritma tidak dapat secara langsung digeneralisasikan ke seluruh dataset. Perbedaan jumlah data, karakteristik variabel, distribusi kelas, serta pola hubungan antar variabel dapat menghasilkan performa model yang berbeda. Penelitian terdahulu juga memiliki fokus yang berbeda, misalnya klasifikasi KIP berdasarkan data sosial ekonomi (Madani et al., 2025), klasifikasi bantuan pendidikan berdasarkan atribut akademik dan administratif (Gori et al., 2024), serta klasifikasi PIP berdasarkan kondisi sosial ekonomi orang tua (Amanda & Voutama, 2025). Dengan demikian, diperlukan pengujian pada dataset mahasiswa yang memiliki karakteristik variabel yang berbeda.
Dataset yang digunakan dalam penelitian ini terdiri atas informasi jenis kelamin, alamat, program studi yang dipilih, alat transportasi, penghasilan ayah, pekerjaan ayah, serta status kelayakan menerima beasiswa. Variabel *layak atau tidak layak menerima beasiswa* digunakan sebagai variabel target, sedangkan karakteristik mahasiswa dan kondisi sosial ekonomi keluarga digunakan sebagai variabel prediktor. Variabel nama mahasiswa tidak digunakan sebagai prediktor karena berfungsi sebagai identitas data dan tidak memiliki hubungan substantif yang diperlukan untuk proses prediksi.

Penelitian ini memiliki fokus yang lebih spesifik, yaitu membandingkan *Decision Tree* dan *Random Forest* baik sebelum maupun setelah dilakukan hyperparameter tuning pada dataset mahasiswa penerima beasiswa. Dengan pendekatan tersebut, penelitian tidak hanya melihat algoritma mana yang menghasilkan performa terbaik, tetapi juga menganalisis apakah proses *hyperparameter tuning* memberikan peningkatan performa pada masing-masing algoritma. Evaluasi model akan dilakukan menggunakan beberapa metrik klasifikasi, seperti *accuracy*, *precision*, *recall*, dan *F1-score*, sehingga perbandingan tidak hanya bergantung pada satu ukuran performa.

Selain membandingkan performa, penelitian ini juga diarahkan untuk menganalisis karakteristik atau fitur yang berkontribusi terhadap hasil klasifikasi. Analisis tersebut penting karena model prediksi tidak hanya diharapkan mampu menghasilkan keputusan layak atau tidak layak, tetapi juga dapat memberikan informasi mengenai variabel yang paling berperan dalam proses klasifikasi. Dengan demikian, hasil penelitian dapat memberikan perspektif yang lebih komprehensif mengenai penerapan algoritma machine learning dalam mendukung proses seleksi penerima beasiswa.

GridSearchCV dipilih bukan karena dianggap sebagai metode optimasi yang

selalu lebih unggul, tetapi karena karakteristik penelitian memiliki ruang hyperparameter yang terbatas, diskrit, dan telah ditentukan secara teoritis, sehingga exhaustive search terhadap seluruh kombinasi masih feasible. Pendekatan ini memungkinkan evaluasi sistematis, transparan, dan reproducible melalui cross-validation, sedangkan RandomizedSearch dan Bayesian Optimization lebih menguntungkan pada ruang pencarian yang lebih besar atau ketika biaya evaluasi setiap konfigurasi tinggi (Bischl et al. 2023), (Brazdil, P., et al. 2022)

Berdasarkan uraian tersebut, penelitian ini dilakukan dengan judul "Perbandingan Decision Tree dan Random Forest Berbasis Grid Search Hyperparameter Tuning untuk Prediksi Kelayakan Beasiswa". Penelitian ini bertujuan untuk membangun dan membandingkan model *Decision Tree* dan *Random Forest* berbasis Grid Search *hyperparameter tuning* pada kedua algoritma, mengevaluasi perubahan performa sebelum dan sesudah tuning, serta menentukan model yang memberikan performa terbaik dalam memprediksi kelayakan mahasiswa penerima beasiswa.

METODE PENELITIAN

Jenis Penelitian

Penelitian ini merupakan penelitian kuantitatif dengan pendekatan *data mining* dan *machine learning* yang bertujuan membangun model klasifikasi untuk memprediksi kelayakan mahasiswa sebagai penerima beasiswa. Pendekatan yang digunakan adalah *supervised classification* karena dataset penelitian memiliki variabel target berupa status kelayakan mahasiswa menerima beasiswa.

Tempat dan Waktu Penelitian

Penelitian ini dilakukan di lingkungan salah satu perguruan Tinggi Swasta, perguruan tinggi X dengan menggunakan data mahasiswa selama beberapa tahun akademik. Proses penelitian meliputi beberapa tahapan dimulai dari pengumpulan data, prapemrosesan, pelatihan model, pengujian, serta analisis hasil. Adapun waktu yang dilakukan untuk penelitian ini direncanakan berlangsung selama 1 bulan, dimulai dari tahap yaitu tahap persiapan data hingga penyusunan laporan hasil penelitian.

Sumber dan Jenis Data

Data yang digunakan untuk penelitian ini diantaranya adalah atribut seperti, program studi, Jenis kelamin, alamat rumah, penghasilan orang tua (ayah), pekerjaan orang tua (ayah), alat transportasi, NIM, dan layak atau tidak layak. Jenis data ini merupakan data dalam kategori data terstruktur dengan tipe klasifikasi biner, di mana variabel target adalah kelayakan mahasiswa penerima beasiswa. Menurut (Sobari et al., 2025), penggunaan data terstruktur memungkinkan penerapan algoritma klasifikasi seperti Decision Tree dan Random Forest secara efisien untuk menemukan pola-pola prediktif.

Variabel Penelitian

Variabel yang digunakan dalam penelitian ini terdiri dari dua variable yaitu Variabel Independen (X) meliputi program studi, Jenis kelamin, alamat rumah, penghasilan orang tua (ayah), pekerjaan orang tua (ayah), alat transportasi, NIM. Dan Variabel Dependen (Y) Adalah layak atau tidak layak

Menurut (Achsanuddin et al., 2023), variabel independen merupakan faktor yang diduga memengaruhi variabel dependen, sementara variabel dependen adalah hasil atau keluaran yang ingin diprediksi.

Pengumpulan Data

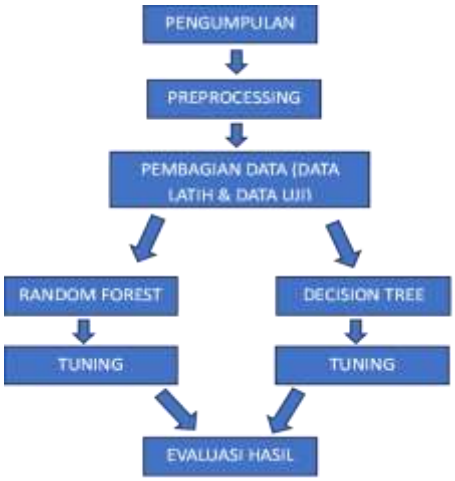
Metode pengumpulan data di penelitian ini dilakukan melalui dokumentasi, yaitu pengambilan data mahasiswa dari Perguruan Tinggi X dan dilakukan wawancara, agar

Journal : Jurnal Rekayasa Sistem Informasi dan Teknologi (JRSIT)
Publisher : Yayasan Nuraini Ibrahim Mandiri
ISSN : e-ISSN 3025-888X
Volume / Issue / Year/ Page : Vol. 4 No. 1 (2026): Agustus/ Hal 95-107

memperoleh informasi tambahan dari pihak pengelola beasiswa terkait apa yang menjadi kriteria dalam penilaian kelayakan.

Tahapan Penelitian

Penelitian ini dilakukan dengan tahap sebagai berikut :



Gambar 1. Tahapan Penelitian

Pengumpulan Data

Data mahasiswa diperoleh dan dikumpulkan dari Perguruan Tinggi Swasta X diambil dari beberapa tahun akademik mahasiswa yang ada di Perguruan Tinggi Tersebut.

Prapemrosesan Data (Data Preprocessing)

Pada tahap ini data di dibersihkan atau di *cleaning,* atribut yang tidak perlu atau tidak berpengaruh akan dihapus, data yang dihapus seperti jenis kelamin, Alamat rumah, dan NIM.

Pembagian Data (Data Latih & Data Uji)

Kemudian pada tahan pembagian data, dataset dibagi menjadi dua, yaitu data latih dan data uji. Data latih yaitu sebesar 70% dan data uji sebesar 30%. Pembagian dataset ini dilakukan secara acak agar supaya menjaga data tetap ada keseimbangan kelas.

Pengujian ALgoritma base dan Tuning

Membandingkan dua algoritma klasifikasi, yaitu Decision Tree dan Random Forest. Masing-masing algoritma akan diuji dalam dua kondisi, yaitu menggunakan konfigurasi awal (*baseline*) dan menggunakan konfigurasi hasil hyperparameter tuning.

Hyperparameter Tuning

Hyperparameter tuning dilakukan untuk memperoleh konfigurasi parameter yang optimal pada algoritma *Decision Tree* dan *Random Forest*. Proses optimasi menggunakan *GridSearchCV* dengan 5-fold cross-validation. Metode *GridSearchCV* dipilih karena memungkinkan seluruh kombinasi parameter dalam *search space* yang telah ditentukan untuk dievaluasi secara sistematis. Pada algoritma *Decision Tree* dan *Random Forest*, *search space* meliputi parameter criterion dengan nilai gini dan entropy, max_depth dengan nilai None, 5, 10, dan 15, min_samples_split dengan nilai 2, 5, dan 10, serta min_samples_leaf dengan nilai 1, 2, dan 4. Pemilihan parameter tersebut

didasarkan pada pengaruhnya terhadap kriteria pemisahan, kedalaman pohon, serta jumlah minimum sampel dalam pembentukan node dan leaf. Proses pencarian menggunakan accuracy sebagai fungsi scoring, sehingga kombinasi parameter dengan nilai rata-rata accuracy *cross-validation* tertinggi ditetapkan sebagai konfigurasi terbaik. Hasil tuning menunjukkan bahwa konfigurasi terbaik pada *Decision Tree* adalah criterion=gini, max_depth=5, min_samples_split=2, dan min_samples_leaf=1, sedangkan konfigurasi terbaik pada *Random Forest* adalah criterion=entropy, max_depth=5, min_samples_split=10, dan min_samples_leaf=2. Selanjutnya, masing-masing konfigurasi terbaik digunakan untuk membangun model hasil tuning dan dievaluasi menggunakan data pengujian yang telah dipisahkan sebelumnya.

Tabel 1. Search Space Hyperparameter

Algoritma	Hyperparameter	Nilai yang Diuji
Decision Tree	criterion	gini, entropy
Decision Tree	max_depth	None, 5, 10, 15
Decision Tree	min_samples_split	2, 5, 10
Decision Tree	min_samples_leaf	1, 2, 4
Random Forest	criterion	gini, entropy
Random Forest	max_depth	None, 5, 10, 15
Random Forest	min_samples_split	2, 5, 10
Random Forest	min_samples_leaf	1, 2, 4

Evaluasi Hasil

Pada tahap ini hasil nilai *accuracy*, *Precision*, *F1-Score* dan *Recall* akan dihasilkan dan dilakukan evaluasi.

Alat dan Perangkat yang Digunakan

Perangkat lunak yang digunakan pada penelitian ini menggunakan Bahasa pemrograman Python Libraries: Scikit-learn, Pandas, NumPy, Matplotlib. Tools visualisasi yaitu menggunakan Google Colab.

HASIL DAN PEMBAHASAN

Dalam penelitian ini data yang digunakan terdiri dari 1083 record. Terdiri dari 943 data tidak layak penerima beasiswa dan 142 layak penerima beasiswa. Selanjutnya sebelum dilakukan proses pelatihan model, data melewati tahapan prapemrosesan diantaranya, Peneliti mempersingkat nama atribut biar lebih ringkas dalam pemrosesan data yaitu PS (Program Studi), AT (Alat Transportasi), PA (Penghasilan Orang Tua), PEA (Pekerjaan Orang Tua), dan LT (Layak atau tidak layak).

Tabel 2. Karakteristik dataset sebelum pemrosesan Data

No	Nama	Tipe
1	Nama Sesuai Ijazah	Polynominal
2	Program Studi	Polynominal
3	Jenis Kelamin	Binominal
4	Alamat	Polynominal
5	Kecamatan	Polynominal
6	Alat Transportasi	Polynominal
7	Pekerjaan Ayah	Polynominal
8	Penghasilan Ayah	Polynominal
9	Layak atau Tidak Layak	Binominal

Sebelum masuk di pemodelan atribut yang tidak perlu atau tidak berpengaruh

Journal : Jurnal Rekayasa Sistem Informasi dan Teknologi (JRSIT)
Publisher : Yayasan Nuraini Ibrahim Mandiri
ISSN : e-ISSN 3025-888X
Volume / Issue / Year/ Page : Vol. 4 No. 1 (2026): Agustus/ Hal 95-107

akan dihapus, data yang dihapus seperti nama, Jenis Kelamin, Alamat rumah, kecamatan.

Kemudian tahapan berikutnya Adalah dilakukannya Transformasi data (data transformation) dataset, untuk mengubah format atribut kategorikal menjadi numerik dengan teknik *label encoding* seperti pada gambar berikut.

```
#Merubah data kategori ke Numerik
from sklearn.preprocessing import LabelEncoder
label = LabelEncoder()
dataset['PS'] = label.fit_transform(dataset['PS'])
dataset['AT'] = label.fit_transform(dataset['AT'])
dataset['PA'] = label.fit_transform(dataset['PA'])
dataset['PEA'] = label.fit_transform(dataset['PEA'])
```

Gambar 2. Transformasi Data

```
dataset
```

	PS	AT	PA	PEA	LT
0	8	10	2	3	tidak layak
1	8	10	12	3	tidak layak
2	8	10	5	0	tidak layak
3	4	10	1	3	tidak layak
4	4	10	1	5	tidak layak
...	...	...	...	...	...
1078	17	10	1	2	tidak layak
1079	17	10	1	4	tidak layak
1080	17	10	3	3	tidak layak
1081	17	10	6	3	tidak layak
1082	17	1	7	2	tidak layak

1083 rows × 5 columns

Gambar 3. Hasil Transformasi Data

Gambar 3 menunjukkan hasil dari transformasi data, proses ini dilakukan bertujuan agar data bisa di proses menggunakan *machine learning,* baik algoritma *random forest* maupun algoritma *decision tree.*

Tahapan selanjutnya pembagian data latih dan data uji, dengan Bahasa pemrogaman python. Data dibagi menjadi dua secara random. Untuk data latih 70% dan untuk data uji 30%.

```
#membagi dataset dengan data testing yakni 70% data training dan 20% data testing
X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size=0.3, random_state=100)
```

Gambar 4. Pembagian Data Latih

Proses berikutnya hasil dengan dua algoritma baseline, yaitu algoritma decision tree dan algoritma *random forest*. Untuk hasil akurasi menggunakan algoritma *random forest* didapatkan seperti gambar berikut.

```
import numpy as np
cm = np.array(confusion_matrix(Y_test, y_pred))
print(cm)
#Calsification Report
cr = classification_report(Y_test, y_pred)
print(cr)
```

```
[[ 13  31]
 [ 16 265]]
              precision    recall  f1-score   support

       layak       0.45      0.30      0.36        44
 tidak layak       0.90      0.94      0.92       281

    accuracy                           0.86       325
   macro avg       0.67      0.62      0.64       325
weighted avg       0.83      0.86      0.84       325
```

Gambar 5. Hasil menggunakan Algoritma *Random Forest*

Journal : Jurnal Rekayasa Sistem Informasi dan Teknologi (JRSIT)
Publisher : Yayasan Nuraini Ibrahim Mandiri
ISSN : e-ISSN 3025-888X
Volume / Issue / Year/ Page : Vol. 4 No. 1 (2026): Agustus/ Hal 95-107

Untuk hasil akurasi menggunakan algoritma *Decision Tree* didapatkan seperti gambar berikut.

```
#Menampilkan akurasi dari model dan menampilkan confusion matrix

import numpy as np
print("Accuracy :", metrics.accuracy_score(Y_test,Y_pred))
cm = np.array(confusion_matrix(Y_test, Y_pred))
print(cm)
#Calsification Report
cr = classification_report(Y_test, Y_pred)
print(cr)

Accuracy : 0.8276923076923077
[[ 16  28]
 [ 28 253]]
              precision    recall  f1-score   support

       layak       0.36      0.36      0.36        44
 tidak layak       0.90      0.90      0.90       281

    accuracy                           0.83       325
   macro avg       0.63      0.63      0.63       325
weighted avg       0.83      0.83      0.83       325
```

Gambar 6. Hasil menggunakan Algoritma *Decision Tree*

Tahap selanjutnya yaitu penggunaan hyperparameter Tuning dengan *GridSearchCV* dalam algoritma random Forest. Hasilnya sebagai berikut

```
print("Accuracy :", accuracy_score(Y_test,Y_pred))
cm = np.array(confusion_matrix(Y_test, Y_pred))
print(cm)
cr = classification_report(Y_test, Y_pred)
print(cr)

Accuracy : 0.8738461538461538
[[  8  36]
 [  5 276]]
              precision    recall  f1-score   support

       layak       0.62      0.18      0.28        44
 tidak layak       0.88      0.98      0.93       281

    accuracy                           0.87       325
   macro avg       0.75      0.58      0.61       325
weighted avg       0.85      0.87      0.84       325
```

Gambar 7. Hyperparameter menggunakan Tuning untuk Algoritma Random Forest

```
print("Accuracy :", metrics.accuracy_score(Y_test,Y_pred))
cm = np.array(confusion_matrix(Y_test, Y_pred))
print(cm)
cr = classification_report(Y_test, Y_pred)
print(cr)

Accuracy : 0.8615384615384616
[[  8  36]
 [  9 272]]
              precision    recall  f1-score   support

       layak       0.47      0.18      0.26        44
 tidak layak       0.88      0.97      0.92       281

    accuracy                           0.86       325
   macro avg       0.68      0.57      0.59       325
weighted avg       0.83      0.86      0.83       325
```

Gambar 8. Hyperparameter menggunakan Tuning untuk Algoritma Decision Tree

Tabel 2. Perbandingan algoritma *Random Forest* dan *Decision Tree* base dan menggunakan Hyperparameter menggunakan Tuning

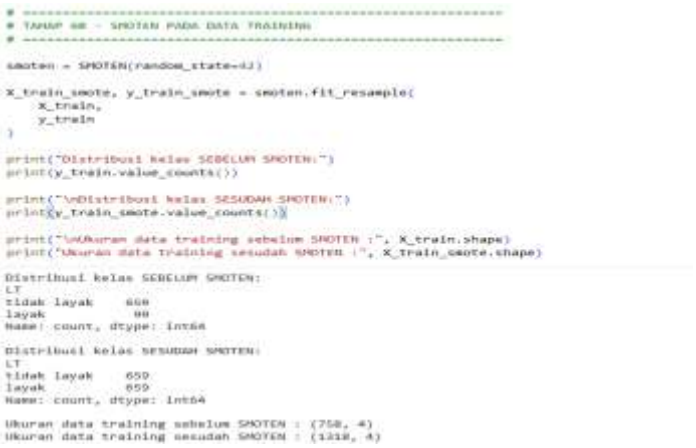
Model / Algoritma	Tuning	Accuracy (%)	Precision	Recall	F1-Score
Random Forest	Base (Default)	85.54% (0.86)	0.83	0.86	0.84
Decision Tree	Base (Default)	82.77% (0.83)	0.83	0.83	0.83
Random Forest	Hyperparameter Tuning	**87.38% (0.87)**	**0.85**	**0.87**	**0.84**
Decision Tree	Hyperparameter Tuning	86.15% (0.86)	0.83	0.86	0.83

Journal : Jurnal Rekayasa Sistem Informasi dan Teknologi (JRSIT)
Publisher : Yayasan Nuraini Ibrahim Mandiri
ISSN : e-ISSN 3025-888X
Volume / Issue / Year/ Page : Vol. 4 No. 1 (2026): Agustus/ Hal 95-107

Berdasarkan tabel 2, algoritma random forest dan decision tree mengalami peningkatan akurasi setelah menggunakan hyperparameter tuning. Algoritma decision tree mengalami kenaikan secara *signifikan* yang awal mulanya mendapatkan hasil akurasi 82.77% menjadi 86.15% setelah menggunakan *hyperparameter tuning.*

Hasil dari gambar 7 pengujian menunjukkan bahwa ketidakseimbangan kelas memberikan pengaruh terhadap kemampuan model dalam mengenali mahasiswa yang termasuk kelas layak. Pada model Random Forest dengan *hyperparameter tuning*, diperoleh *accuracy* sebesar 87,38%, tetapi *recall* kelas layak hanya sebesar 0,18. Berdasarkan *confusion matrix*, dari 44 mahasiswa yang sebenarnya termasuk kelas layak, hanya 8 mahasiswa yang berhasil diklasifikasikan dengan benar, sedangkan 36 mahasiswa diklasifikasikan sebagai tidak layak. Kondisi tersebut menunjukkan bahwa *accuracy* yang tinggi belum dapat menggambarkan kemampuan model dalam mendeteksi kelas minoritas. Permasalahan serupa juga terlihat pada Decision Tree dengan *hyperparameter tuning* yang menghasilkan *recall* kelas layak sebesar 0,18.

Untuk mengatasi permasalahan tersebut, penelitian ini menerapkan *Synthetic Minority Over-sampling Technique for Nominal (SMOTEN)* pada data training. SMOTEN dipilih karena seluruh variabel prediktor yang digunakan dalam penelitian bersifat kategorikal, yaitu program studi, alat transportasi, penghasilan ayah, dan pekerjaan ayah. Setelah penerapan SMOTEN, jumlah data kelas layak pada data training meningkat dari 99 menjadi 659 data, sedangkan kelas tidak layak tetap sebanyak 659 data. Dengan demikian, distribusi kelas pada data training menjadi seimbang, yaitu 50% kelas layak dan 50% kelas tidak layak. Data testing tidak dilakukan oversampling dan tetap mempertahankan distribusi asli untuk memperoleh evaluasi yang merepresentasikan kondisi dataset sebenarnya.



```
# ------------------------------------------------------------
# TAHAP 6B - SMOTEN PADA DATA TRAINING
# ------------------------------------------------------------

smoten = SMOTEN(random_state=42)

X_train_smote, y_train_smote = smoten.fit_resample(
    X_train,
    y_train
)

print("Distribusi kelas SEBELUM SMOTEN:")
print(y_train.value_counts())

print("\nDistribusi kelas SESUDAH SMOTEN:")
print(y_train_smote.value_counts())

print("\nUkuran data training sebelum SMOTEN :", X_train.shape)
print("Ukuran data training sesudah SMOTEN :", X_train_smote.shape)
```

```
Distribusi kelas SEBELUM SMOTEN:
LT
tidak layak    659
layak           99
Name: count, dtype: int64

Distribusi kelas SESUDAH SMOTEN:
LT
tidak layak    659
layak          659
Name: count, dtype: int64

Ukuran data training sebelum SMOTEN : (758, 4)
Ukuran data training sesudah SMOTEN : (1318, 4)
```

Gambar 9. SMOTEN Pada Data Training

Setelah penerapan SMOTEN, distribusi kelas training berubah dari 659 data tidak layak dan 99 data layak menjadi masing-masing 659 data. Setelah itu dilakukan penanganan *class imbalance* menggunakan SMOTEN pada data training. Setelah penerapan SMOTEN, distribusi kelas training berubah dari 659 data tidak layak dan 99 data layak menjadi masing-masing 659 data. Hasil pengujian menunjukkan bahwa Decision Tree + SMOTEN + GridSearchCV menghasilkan *recall* kelas layak sebesar 0,53, meningkat dari 0,18 pada Decision Tree + GridSearchCV. Sementara itu, Random Forest + SMOTEN + GridSearchCV menghasilkan *recall* kelas layak sebesar 0,49, meningkat dari 0,18 pada Random Forest + GridSearchCV. Peningkatan tersebut menunjukkan

Journal : Jurnal Rekayasa Sistem Informasi dan Teknologi (JRSIT)
Publisher : Yayasan Nuraini Ibrahim Mandiri
ISSN : e-ISSN 3025-888X
Volume / Issue / Year/ Page : Vol. 4 No. 1 (2026): Agustus/ Hal 95-107

bahwa SMOTEN mampu meningkatkan kemampuan kedua algoritma dalam mengenali kelas minoritas. Seperti yang terlihat pada gambar di bawah ini, gambar 10 dan gambar 11.

```
Hyperparameter terbaik:
{'criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 2, 'min_samples_split': 10}

Best CV Accuracy:
0.8892585551330798

Accuracy Data Testing:
0.8153846153846154

Confusion Matrix:
[[ 23  20]
 [ 40 242]]

Classification Report:
              precision    recall  f1-score   support

       layak       0.37      0.53      0.43        43
 tidak layak       0.92      0.86      0.89       282

    accuracy                           0.82       325
   macro avg       0.64      0.70      0.66       325
weighted avg       0.85      0.82      0.83       325
```

Gambar 10. Hasil *Decision Tree, SMOTEN* dan *Hyperparameter Tuning*

```
Hyperparameter terbaik:
{'criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 10}

Best CV Accuracy:
0.9021747897223182

Accuracy Data Testing:
0.8461538461538461

Confusion Matrix:
[[ 21  22]
 [ 28 254]]

Classification Report:
              precision    recall  f1-score   support

       layak       0.43      0.49      0.46        43
 tidak layak       0.92      0.90      0.91       282

    accuracy                           0.85       325
   macro avg       0.67      0.69      0.68       325
weighted avg       0.86      0.85      0.85       325
```

Gambar 11. Hasil *Random Forest, SMOTEN* dan *Hyperparameter Tuning*

Meskipun demikian, penerapan SMOTEN menyebabkan penurunan *accuracy* keseluruhan. *Accuracy* Decision Tree menurun dari 86,15% menjadi 81,54%, sedangkan Random Forest menurun dari 87,38% menjadi 84,62%. Penurunan *accuracy* tersebut menunjukkan adanya *trade-off* antara kemampuan model dalam mengenali kelas minoritas dan performa klasifikasi secara keseluruhan. Dalam konteks prediksi kelayakan penerima beasiswa, kondisi tersebut perlu dipertimbangkan karena kesalahan dalam mengidentifikasi mahasiswa yang sebenarnya layak dapat menyebabkan mahasiswa yang memenuhi kriteria tidak teridentifikasi oleh model. Oleh karena itu, evaluasi model dalam penelitian ini tidak hanya didasarkan pada *accuracy*, tetapi juga mempertimbangkan *precision*, *recall*, dan *F1-score*, khususnya pada kelas layak sebagai kelas minoritas.

KESIMPULAN

Berdasarkan hasil penelitian ini, bisa di lihat bahwa algoritma *Decision Tree* dan *Random Forest* dapat digunakan untuk melakukan prediksi kelayakan mahasiswa penerima beasiswa berdasarkan karakteristik program studi, alat transportasi, penghasilan ayah, dan pekerjaan ayah. Hasil pengujian awal menunjukkan bahwa *Decision Tree* memperoleh *accuracy* sebesar 82,77%, sedangkan *Random Forest*

memperoleh *accuracy* sebesar 86,00%. Setelah dilakukan *hyperparameter tuning* menggunakan GridSearchCV dengan 5-fold cross-validation, *Decision Tree* memperoleh *accuracy* sebesar 86,15%, sedangkan *Random Forest* memperoleh *accuracy* sebesar 87,38%.

Hasil analisis juga menunjukkan adanya ketidakseimbangan kelas yang cukup tinggi, yaitu 941 mahasiswa (86,89%) berstatus tidak layak dan 142 mahasiswa (13,11%) berstatus layak. Kondisi tersebut menyebabkan model cenderung lebih baik dalam mengenali kelas mayoritas dibandingkan kelas minoritas. Hal ini terlihat pada model *Random Forest* dengan *hyperparameter tuning* yang memperoleh *accuracy* sebesar 87,38%, tetapi hanya menghasilkan *recall* sebesar 0,18 pada kelas layak. Dengan demikian, *accuracy* yang tinggi belum menunjukkan bahwa model mampu mengidentifikasi mahasiswa yang sebenarnya layak menerima beasiswa secara optimal. Untuk mengatasi permasalahan tersebut, penelitian ini menerapkan SMOTEN pada data training karena variabel prediktor yang digunakan bersifat kategorikal. Hasil eksperimen menunjukkan adanya peningkatan kemampuan model dalam mengenali kelas minoritas. *Decision Tree* + *SMOTEN* + *GridSearchCV* menghasilkan *recall* kelas layak sebesar 0,53, meningkat dari 0,18 pada Decision Tree + GridSearchCV. Sementara itu, Random Forest + *SMOTEN* + *GridSearchCV* menghasilkan *recall* kelas layak sebesar 0,49, meningkat dari 0,18 pada *Random Forest* + *GridSearchCV.* Hal tersebut menunjukkan bahwa penerapan *SMOTEN* mampu meningkatkan kemampuan kedua algoritma dalam mendeteksi mahasiswa yang termasuk kelas layak.

Meskipun penerapan *SMOTEN* menyebabkan penurunan *accuracy* keseluruhan, hasil tersebut menunjukkan adanya *trade-off* antara *accuracy* dan kemampuan mendeteksi kelas minoritas. *Decision Tree* + *SMOTEN* + *GridSearchCV* memperoleh *recall* kelas layak tertinggi sebesar 0,53, sedangkan Random Forest + *SMOTEN* + *GridSearchCV* memperoleh *F1-score* kelas layak tertinggi sebesar 0,46. Oleh karena itu, pemilihan model dalam prediksi kelayakan penerima beasiswa tidak sebaiknya hanya didasarkan pada *accuracy*, tetapi juga mempertimbangkan *recall* dan *F1-score* kelas layak. Berdasarkan tujuan untuk meningkatkan kemampuan identifikasi mahasiswa yang layak menerima beasiswa, *Decision Tree + SMOTEN + GridSearchCV* menjadi model dengan kemampuan deteksi kelas layak terbaik berdasarkan nilai recall.

Pernyataan mengenai Penggunaan Kecerdasan Buatan (AI).

AI digunakan dalam penyusunan naskah ini. Penulis menggunakan perangkat kecerdasan buatan semata-mata untuk membantu perbaikan bahasa, pemeriksaan tata bahasa, serta penyuntingan demi kejelasan dan keterbacaan. Penulis tetap bertanggung jawab penuh atas akurasi, orisinalitas, integritas, dan konten akademis naskah tersebut. Seluruh konten yang dibantu oleh AI telah ditinjau dan diverifikasi oleh penulis sebelum naskah diserahkan.

Konflik kepentingan

Penulis menyatakan tidak ada konflik kepentingan.

DAFTAR PUSTAKA

Achsanuddin, A., Khaerunnisa, F., Violin, V., & Yusuf, M. (2023). Faktor-faktor yang Memengaruhi Produksi Tempe di Kecamatan Burau Kabupaten Luwu Timur. *Kompeten: Jurnal Ilmiah Ekonomi Dan Bisnis*, *2*(2), 568–577.

Amanda, W., & Voutama, A. (2025). Klasifikasi pendapatan menggunakan algoritma random forest: Studi kasus dataset adult income. *Jurnal Ilmiah Informatika Global*, *16*(2), 79–84.

Elim, M. I., & Utami, E. (2025). Performance Comparison of Child Stunting Prediction Support Vector Machine vs Random Forest with Grid Search Optimization. *Jurnal Teknik Informatika (Jutif)*, *6*(5), 5305–5319.

Firnanda, P. A., Shofwatillah, L., Rahma, F., & Fauzi, F. (2025). Analisis Perbandingan Decision Tree dan Random Forest dalam Klasifikasi Penjualan Produk pada Supermarket: Analisis Perbandingan Decision Tree dan Random Forest dalam Klasifikasi Penjualan Produk pada Supermarket. *Emerging Statistics and Data Science Journal*, *3*(1), 445–461.

Gori, T., Sunyoto, A., & Al Fatta, H. (2024). Preprocessing data dan klasifikasi untuk prediksi kinerja akademik siswa. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, *11*(1), 215–224.

Hermawan, L. A., Engkeng, S., & Sekeon, S. S. (2020). HUBUNGAN ANTARA PENGETAHUAN DAN SIKAP PESERTA DIDIK PEREMPUAN DENGAN TINDAKAN PENCEGAHAN KEPUTIHAN DI SMP NEGERI 3 TAHUNA. *KESMAS*, *9*(6).

Madani, M. N., Naila, N. F., & Rasyad, R. Z. (2025). Efektivitas beasiswa pendidikan dalam meningkatkan akses pendidikan tinggi bagi masyarakat berpenghasilan rendah di Kalimantan Timur. *Nusantara Innovation Journal*, *3*(2), 122–130.

N. Santuni, D. Irfan, Y. Hendriyani, and Y. H. (2025). Klasifikasi Penerima Beasiswa Program Indonesia Pintar (PIP) Menggunakan Algoritma Decision Tree. *Jurnal Pendidikan Tambusai, 9*.

Ramadhan, M. G. I. (2024). *Implementasi Algoritma Random Forest Untuk Identifikasi Citra Kayu Makroskopik Dengan Ekstraksi Fitur Gray Level Co-occurrence Matrix (GLCM)*. Universitas Sangga Buana YPKP.

Sari, I. K., Hermawan, A., & Avianto, D. (2026). Klasifikasi Kelayakan Bantuan Pendidikan Menggunakan Metode Decision Tree. *INSOLOGI: Jurnal Sains Dan Teknologi*, *5*(3), 914–925.

Setyowati, S. L., Qalbi, A., Aristawidya, R., Sartono, B., & Firdawanti, A. R. (2025). Optimizing random forest parameters with hyperparameter tuning for classifying school-age KIP eligibility in West Java. *Jambura Journal of Mathematics*, *7*(1), 40–48.

Sobari, S., Purnamasari, A. I., Bahtiar, A., & Kaslani, K. (2025). Meningkatkan model prediksi kelulusan santri tahfidz di Pondok Pesantren Al-Kautsar menggunakan algoritma Random Forest. *Jurnal Informatika Dan Teknik Elektro Terapan*, *13*(1).

Surip, A., Pratama, M. A., Ali, I., Dikananda, A. R., & Purnamasari, A. I. (2021). Penerapan Machine Learning menggunakan algoritma C4. 5 berbasis PSO dalam Menganalisa Data Siswa Putus Sekolah. *Informatics For Educators And Professional: Journal of Informatics*, *5*(2), 147–155.

Bischl, B., Binder, M., Lang, M., Pielok, T., Richter, J., Coors, S., Thomas, J., Ullmann, T., Becker, M., Boulesteix, A.-L., Deng, D., & Lindauer, M. (2023). Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges. *WIREs Data Mining and Knowledge Discovery, 13*(2), e1484.

Brazdil, P., van Rijn, J. N., Soares, C., & Vanschoren, J. (2022). Metalearning for hyperparameter optimization. In P. Brazdil, J. N. van Rijn, C. Soares, & J. Vanschoren, *Metalearning: Applications to data mining* (2nd ed., pp. 103–122). Springer.