Prediksi Kelulusan Tepat Waktu Menggunakan Pendekatan Pohon Keputusan Algoritma *Decision Tree*

Octarian Prasetya Moerdyanto[1], I Kadek Dwi Nuryana[2]

[1,2] S1 Teknik Informatika, Universitas Negeri Surabaya
[1]octarian.19020@mhs.ac.id
[2]dwinuryana@unesa.ac.id

***Abstrak*— Penting bagi perguruan tinggi untuk mengevaluasi kelulusan mahasiswa tepat waktu sebagai bagian penting dalam proses akreditasi. Namun, tantangan muncul dalam mengidentifikasi mahasiswa yang berpotensi lulus tepat waktu atau tidak. Penelitian ini difokuskan pada prediksi kelulusan mahasiswa tepat waktu dengan menggunakan metode Pendekatan pohon keputusan. Data yang digunakan dalam penelitian ini berasal dari program studi sistem informasi di Universitas Negeri Surabaya, dengan total 312 data yang mencakup angkatan 2016 hingga 2021. Tujuan utama penelitian ini adalah untuk menentukan variabel yang berpengaruh pada kelulusan tepat waktu dan membentuk pola pohon keputusan berdasarkan data tersebut. Harapannya, hasil penelitian ini akan memberikan informasi tentang variabel mana yang signifikan dalam mempengaruhi kelulusan tepat waktu pada mahasiswa. Informasi ini sangat berarti untuk mengantisipasi dan memberikan perhatian khusus pada mahasiswa yang berpotensi tidak lulus tepat waktu. Dengan demikian, perguruan tinggi dapat mengambil langkah yang tepat untuk membantu dan meningkatkan peluang kelulusan mahasiswa dengan cara yang efektif.**

Kata Kunci— **Prediksi kelulusan tepat waktu, pohon keputusan, *Decision Tree*, IPK, pengaruh kelulusan tepat waktu**

I. PENDAHULUAN

Lulusan tepat waktu memiliki peranan krusial dalam proses akreditasi perguruan tinggi. Salah satu elemen penilaian yang signifikan adalah efisiensi pendidikan di institusi tersebut. Dengan demikian, ketika mahasiswa dapat menyelesaikan studi tepat waktu, hal ini akan memberikan kontribusi positif dalam meningkatkan akreditasi perguruan tinggi tersebut. Keuntungan bagi mahasiswa yang lulus tepat waktu adalah mereka tidak perlu lagi mengeluarkan biaya tambahan untuk kuliah dan dapat langsung memasuki dunia kerja setelah kelulusan. Oleh karena itu, pentingnya lulus tepat waktu tidak dapat diabaikan, karena hal ini memiliki dampak langsung pada peningkatan akreditasi institusi perguruan tinggi serta memberikan manfaat nyata bagi mahasiswa [1].

Namun, seringkali waktu kelulusan mahasiswa sulit dideteksi sejak awal, yang dapat menyebabkan keterlambatan dalam pencapaian gelar. Untuk mengatasi tantangan tersebut, diperlukan teknik yang memungkinkan prediksi kelulusan mahasiswa. Salah satu teknik yang sering digunakan adalah data mining, dan metode klasifikasi menjadi metode yang paling umum dipilih untuk melakukan prediksi kelulusan mahasiswa [2].

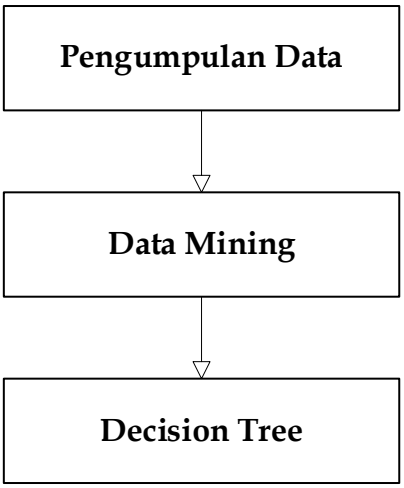
Aspek Menurut David Hand, Heikki Mannila, dan Padhraic Smyth seperti yang dikutip oleh Widodo, Handayanto, dan rekan-rekan, data mining adalah proses analisis data yang bertujuan untuk menemukan pola-pola yang signifikan dan mengungkap informasi baru yang sebelumnya belum diketahui. Metode ini menggunakan teknik terbaru dalam pemahaman data dengan tujuan memberikan manfaat bagi pemilik data tersebut [3].

Nuqson M. juga melakukan penelitian serupa yang mengaplikasikan data mining untuk memperlihatkan informasi mengenai tingkat kelulusan mahasiswa. Dalam penelitian ini, tingkat kelulusan diukur berdasarkan lama studi dan IPK. Algoritma yang digunakan adalah algoritma apriori, dan hasil yang ditampilkan meliputi nilai support dan confidence dari setiap kategori tingkat kelulusan [4].

Penelitian ini menggunakan data prodi sistem informasi, jurusan Teknik Informatika, Universitas Negeri Surabaya, yang terletak di Ketintang Wonokromo Surabaya. Penelitian ini bertujuan untuk melakukan data mining pada riwayat akademik mahasiswa dengan harapan dapat mengantisipasi dan menangani dini kemungkinan mahasiswa yang tidak lulus atau lulus tepat waktu. Dengan memggunakan pendekatan pohon keputusan algoritma *Decision Tree* untuk mengetahui pola keputusan dari data tersebut.

II. METODE PENELITIAN

Metode penelitian merujuk pada serangkaian langkah atau prosedur yang digunakan untuk memperoleh pengetahuan atau ilmu (Suryana, 2010). Definisi lain dari metode penelitian adalah suatu pendekatan yang digunakan untuk menentukan langkah-langkah selanjutnya dan memilih topik permasalahan yang akan menjadi dasar dari suatu penelitian. Berikut adalah langkah-langkah dalam menentukan dan merumuskan topik penelitian.



Gambar 1. Tahapan Penelitian

A. Pengumpulan data

Data yang akan digunakan 312 data mahasiswa prodi sistem informasi jurusan angakatan 2016 sampai 2021, Data tentang mahasiswa tersebut bersumber dari SIM mahasiswa Prodi Sistem Informasi Jurusan Teknik Informatika Universitas Negeri Surabaya. Tidak semua data tentang mahasiswa yang lulus akan digunakan. Data yang akan digunakan adalah Total SKS, Jenis Kelamin, Mengulang, Cuti, Indeks Prestasi Semester 1 sampai Semester 8, Indek Prestasi Komulatif sebagai atribut dan Satus Kelulusan akan digunakan sebagai kelas, gambar dibawah ini adalah dataset

TOTAL SKS	JENIS KELAMIN	MENGULANG	CUTI	IPS 1	IPS 2	IPS 3	IPS 4	IPS 5	IPS 6	IPS 7	IPS 8	IPS 9	IPS 10	IPS 11	IPS 12	IPS 13	IPS 14	IPK	STATUS KELULUSAN
186	LAKI - LAKI	IYA	TIDAK	3,59	3,26	3,43	3,19	3,68	3,31	2	0	0	0	0	0	0	0	2,62	TERLAMBAT
158	LAKI - LAKI	TIDAK	TIDAK	3,46	3,53	3,59	3,82	4	4	2,69	4							3,58	PROSES
152	LAKI - LAKI	TIDAK	TIDAK	3,52	3,77	3,75	3,83	3,44	4	4	4							3,77	PROSES
152	LAKI - LAKI	TIDAK	TIDAK	3,55	3,47	3,65	3,63	3,74	3,75	4								3,68	PROSES
122	PEREMPUAN	TIDAK	TIDAK	3,73	3,55	3,65	3,61	3,51										3,61	
122	PEREMPUAN	TIDAK	TIDAK	3,85	3,8	3,87	3,71	3,89										3,82	
122	PEREMPUAN	TIDAK	TIDAK	3,83	3,77	3,8	3,76	3,36										3,71	
82	LAKI - LAKI	TIDAK	TIDAK	3,6	3,57	3,62	3,53											3,58	
82	PEREMPUAN	TIDAK	TIDAK	3,59	3,64	3,83	3,62											3,67	

Gambar 2. Dataset

B. *Data mining*

1. Pembersihan Data
   Pembersihan data dilakukan untuk menghilangkan data yang tidak relevan atau noise dari dataset. Dalam penelitian ini, proses pembersihan data dilakukan pada aplikasi repositori lulusan. Aplikasi repositori lulusan telah diatur sedemikian rupa sehingga setiap data mengenai mahasiswa yang lulus, termasuk data nama, jenis kelamin, total SKS, pengulangan mata kuliah, cuti, Indeks Prestasi Semester (IPS) 1 sampai 8, dan Indeks Prestasi Kumulatif (IPK), disimpan tanpa nilai kosong dan konsisten.
2. Seleksi Data
   Seleksi data dilakukan untuk memilih data yang relevan dan diperlukan untuk analisis. Pada banyak kasus, tidak semua data dalam database diperlukan. Sebagai contoh, dalam analisis market basket, ketika mempelajari faktor-faktor yang memengaruhi kecenderungan pelanggan membeli produk tertentu, tidak perlu menggunakan nama pelanggan. Cukup menggunakan ID unik pelanggan sudah mencukupi. Oleh karena itu, hanya data yang relevan untuk analisis yang diambil dari database. Pada dataset ini, variabel-nilai dari IPS semester 9 sampai 14 dihapus, karena hanya data IPS semester 1 sampai 8 yang dibutuhkan untuk analisis.
3. Transformasi Data

Attributes Information

Attribute	Data Type	Description
Nama	object	Nama Mahasiswa
TOTAL SKS	int64	Total SKS Mahasiswa
Jenis Kelamin	object	Jenis Kelamin Mahasiswa
Mengulang	object	Mengulang Matakuliah Yang Memungkinkan Menambah Semester
Cuti	object	Status Mahasiswa Pernah Cuti atau Tidak
IPS 1	float64	Indeks Prestasi Semester 1
IPS 2	float64	Indeks Prestasi Semester 2
IPS 3	float64	Indeks Prestasi Semester 3
IPS 4	float64	Indeks Prestasi Semester 4
IPS 5	float64	Indeks Prestasi Semester 5
IPS 6	float64	Indeks Prestasi Semester 6
IPS 7	float64	Indeks Prestasi Semester 7
IPS 8	float64	Indeks Prestasi Semester 8
IPK	float64	Indeks Prestasi Komulatif
Status Kelulusan	float64	Tepat Waktu atau Tidak Tepat Waktu

Gambar 3. Informasi Data

Transformasi data merupakan tahap di mana data diubah menjadi nilai dengan format tertentu. Dalam proses KDD (Knowledge Discovery in Databases), digunakan data yang bersifat diskrit, sehingga data yang bersifat kontinu akan diubah menjadi data diskrit. Selain itu, ada data yang memiliki rentang nilai yang terlalu luas dan dapat mempengaruhi proses KDD, sehingga perlu dikelompokkan menjadi beberapa kelompok kecil. Dalam penelitian ini, dilakukan transformasi tipe data object pada variabel jenis kelamin, matakuliah mengulang, dan cuti menjadi tipe data integer.

C. Algoritma Decision Tree

Decision Tree merupakan salah satu metode pengolahan data yang digunakan untuk meramalkan atau memprediksi hasil masa depan dengan membangun model klasifikasi atau regresi dalam bentuk struktur pohon. Prosesnya melibatkan pemecahan data menjadi himpunan bagian yang lebih kecil, dan secara bertahap membentuk pohon keputusan. Pohon ini terdiri dari node keputusan dan node daun. Setiap node keputusan, seperti "Cuaca/Outlook," memiliki cabang-cabang yang mewakili pilihan seperti "Panas," "Berawan," atau "Hujan."

Selain sebagai alat prediksi, Decision Tree juga berperan dalam eksplorasi data, membantu mengidentifikasi hubungan antara berbagai variabel input dengan variabel target yang diinginkan. Ini adalah langkah penting dalam proses pemodelan, dan pohon keputusan yang dihasilkan bisa

menjadi model akhir atau menjadi bagian dari teknik pemodelan lain.

Keunggulan lain dari metode ini adalah kemampuannya untuk mengeliminasi perhitungan atau data yang tidak diperlukan. Dalam pengujian sampel, hanya kriteria atau kelas tertentu yang digunakan.

Namun, Decision Tree juga memiliki kelemahan. Salah satunya adalah risiko tumpang tindih, terutama ketika kelas dan kriteria yang digunakan terlalu rumit, yang bisa memperlambat waktu pengambilan keputusan karena memerlukan kapasitas memori yang lebih besar.

Algoritme merupakan urutan langkah logis yang digunakan untuk memecahkan masalah. Dalam kata lain, masalah harus dipecah menjadi langkah-langkah yang lebih teratur dan terstruktur.

Fungsi pembuatan algoritme ini sangat berarti dalam menyelesaikan masalah pemrograman yang kompleks. Algoritme bisa diterapkan pada program sederhana hingga yang rumit. Salah satu keuntungannya adalah kemampuannya untuk digunakan berulang kali. Algoritme juga membantu membuat kode program menjadi lebih sederhana bagi para pemrogram. Penggunaan algoritme dapat memungkinkan strategi top-down dan divide-and-conquer dalam implementasi program.

Tidak hanya itu, algoritme juga berperan dalam mengurangi kesalahan yang sering muncul dalam program yang kompleks. Dengan menggunakan algoritme yang tepat, program bisa dirancang secara lebih terstruktur dan logis. Jika terjadi kesalahan, mereka dapat dengan cepat diidentifikasi dan diperbaiki.

Decision Tree merupakan algoritme yang mengadopsi struktur pohon untuk melakukan klasifikasi atau regresi terhadap data berdasarkan fitur-fitur yang ada. Tidak ada satu rumus matematis tunggal yang menggambarkan keseluruhan algoritme ini, karena lebih merupakan proses yang melibatkan beberapa langkah penting dalam membangun pohon keputusan.

Konsep inti dalam Decision Tree melibatkan beberapa istilah kunci yang harus dimengerti:

1. Entropi: Merupakan ukuran dari tingkat ketidakaturan atau keacakan dalam data. Dalam konteks klasifikasi, entropi digunakan untuk mengukur tingkat ketidakpastian atau kompleksitas dalam dataset.
2. Information Gain: Digunakan untuk memilih fitur yang paling informatif atau berarti dalam membangun pohon. Information gain mengukur sejauh mana suatu fitur dapat mengurangi ketidakpastian (entropi) dalam data.
3. Gini Impurity: Merupakan ukuran tingkat ketidakmurnian atau impurity dalam data. Gini impurity juga berguna dalam mengukur ketidakpastian dalam klasifikasi.
4. Kriteria Pemisahan: Proses penting dalam memilih fitur dan ambang batas yang optimal untuk membagi data menjadi subset yang lebih kecil saat membangun pohon keputusan.
5. Pemangkasan: Proses menghapus cabang-cabang yang tidak relevan atau berpotensi menyebabkan overfitting pada pohon keputusan. Tujuan pemangkasan adalah untuk menghindari performa yang buruk pada data yang belum pernah dilihat sebelumnya.

Dalam algoritme Decision Tree, terdapat dua konsep yang sangat penting untuk menilai signifikansi suatu fitur dalam membangun pohon keputusan, yaitu Gain (Information Gain) dan Entropy. Kedua nilai ini berperan dalam menentukan fitur mana yang paling optimal untuk melakukan pemisahan data.

1. Information Gain (Gain): Information Gain adalah ukuran dari seberapa banyak informasi baru yang diperoleh atau ketidakpastian yang dikurangi setelah kita memisahkan data berdasarkan fitur tertentu. Gain dihitung dengan membandingkan tingkat ketidakpastian dataset sebelum dan sesudah pemisahan data berdasarkan fitur yang dipertimbangkan.

Rumus Information Gain: $Gain(F) = Entropy(D) - \Sigma((|Di| / |D|) * Entropy(Di))$

Di mana:

- Gain(F) adalah Information Gain untuk fitur F.
- Entropy(D) adalah Entropy dari dataset D sebelum pemisahan.
- Di adalah subset dari dataset D yang dipisahkan berdasarkan nilai fitur F.
- |Di| adalah jumlah sampel dalam subset Di.
- |D| adalah jumlah total sampel dalam dataset D.
- Entropy(Di) adalah Entropy dari subset Di setelah pemisahan.

Semakin tinggi nilai Gain, semakin besar kontribusi fitur F dalam mengurangi ketidakpastian dalam pemisahan data dan semakin optimal fitur tersebut untuk membangun pohon keputusan.

2. Entropy: Entropy digunakan untuk mengukur tingkat ketidakpastian atau ketidakaturan dalam dataset. Dalam konteks klasifikasi, entropy mengukur seberapa bervariasinya kelas (label) dalam dataset. Jika sebuah dataset memiliki entropy yang rendah, maka data tersebut cenderung homogen dalam hal kelasnya. Sebaliknya, jika entropy tinggi, berarti data tersebut memiliki variasi kelas yang lebih tinggi.

Rumus Entropy: $Entropy(D) = -\Sigma((|Ci| / |D|) * \log2(|Ci| / |D|))$

Di mana:

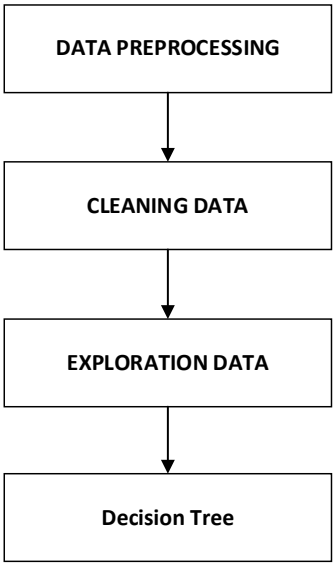
- Entropy(D) adalah Entropy dari dataset D.
- Ci adalah setiap kelas (label) yang ada dalam dataset D.
- |Ci| adalah jumlah sampel yang termasuk dalam kelas Ci.
- |D| adalah jumlah total sampel dalam dataset D.

Semakin tinggi nilai Entropy, semakin besar ketidakpastian dalam dataset dan semakin sulit untuk memisahkan data berdasarkan fitur tertentu.

Penggunaan Gain dan Entropy dalam Decision Tree memainkan peran penting dalam memilih fitur-fitur yang paling informatif dan signifikan dalam membangun pohon keputusan. Dengan demikian, algoritme dapat mencapai pemisahan data yang lebih baik dan akurat dalam tugas klasifikasi atau regresi.

III. HASIL DAN PEMBAHASAN

Paragraf Terdapat enam tahapan pada penelitian ini untuk membangun sistem pendukung keputusan Prediksi Kelulusan



Mahasiswa menggunakan algoritma XG-BOOST, yaitu meliputi sebagai berikut :

Gambar 4. Tahapan Machin Learning Prediksi Kelulusan

A. Data Preprocessing

Dataset penelitian ini menggunakan dataset kelulusan mahasiswa jurusan sistem informasi Universitas Negeri Surabaya dari angaktan 2016 sampai 2021 berisi 312 data mahasiswa.

Mengimpor pustaka pandas dan menggunakannya untuk membaca file Excel bernama 'datasetsiakad.xlsx'. Mengimpor pustaka seaborn dan matplotlib.pyplot untuk tujuan visualisasi data. Setelah membaca file Excel, akan menampilkan beberapa baris pertama DataFrame untuk memberi pratinjau data df.head().

Nama Mahasiswa	TOTAL SKS	JENIS KELAMIN	MENGULANG	CUTI	IPS 1	IPS 2	IPS 3	IPS 4	...	IPS 7	IPS 8	IPS 9	IPS 10	IPS 11	IPS 12	IPS 13	IPS 14	IPK	STATUS KELULUSAN
ABDA KURNIAWAN PRIYAMBADA	156	Perempuan	IYA	TIDAK	3.43	3.35	3.52	3.74	...	1.5	0.0	0.0	0.0	NaN	NaN	NaN	NaN	3.03	TERLAMBAT
ABDUL AZIZ	152	Laki-Laki	TIDAK	TIDAK	3.12	3.64	3.66	3.70	...	4.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	3.68	PROSES
ACHMAD ALVIN ARDIANSYAH	125	Laki-Laki	TIDAK	TIDAK	3.69	3.76	3.81	3.76	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	3.77	NaN
ACHMAD ASRORI	152	Laki-Laki	TIDAK	TIDAK	3.13	3.71	3.24	3.55	...	4.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	3.48	PROSES
ACHMAD MAULUDI ASROR	149	Laki-Laki	IYA	IYA	3.89	3.54	3.22	3.54	...	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2.93	TERLAMBAT

Gambar 5. 'datasetsiakad.xlsx'

Mengganti nilai string tertentu dengan nilai numerik yang sesuai di kolom 'JENIS KELAMIN', 'MENGULANG', 'CUTI', dan 'STATUS KELULUSAN'.**replace()**

Nama Mahasiswa	TOTAL SKS	JENIS KELAMIN	MENGULANG	CUTI	IPS 1	IPS 2	IPS 3	IPS 4	...	IPS 7	IPS 8	IPS 9	IPS 10	IPS 11	IPS 12	IPS 13	IPS 14	IPK	STATUS KELULUSAN
ABDA KURNIAWAN PRIYAMBADA	156	2	2	1	3.43	3.35	3.52	3.74	...	1.5	0.0	0.0	0.0	NaN	NaN	NaN	NaN	3.03	2.0
ABDUL AZIZ	152	1	1	1	3.12	3.64	3.66	3.70	...	4.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	3.68	1.0
ACHMAD ALVIN ARDIANSYAH	125	1	1	1	3.69	3.76	3.81	3.76	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	3.77	NaN
ACHMAD ASRORI	152	1	1	1	3.13	3.71	3.24	3.55	...	4.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	3.48	1.0
ACHMAD MAULUDI ASROR	149	1	2	2	3.89	3.54	3.22	3.54	...	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2.93	2.0

Gambar 6. *Replace data type*

B. *Cleaning Data*

Pada tahapan ini bertujuan untuk mengubah, menghapus, atau memodifikasi dataset untuk memperbaiki data yang tidak sesuai.

1. Drop Feature

   Menghapus *variable* kosong atau *variable* yang tidak digunakan sebagai berikut :

   Gambar 7. *Drop data*

   Menghilangkan data dari objek DataFrame bernama 'df'. Kolom yang dihapus adalah 'NIM', 'IPS 9', 'IPS 10', 'IPS 11', 'IPS 12', 'IPS 13', dan 'IPS 14'. Dengan menggunakan metode dengan parameter, kolom yang ditentukan dihapus dari DataFrame, menghasilkan

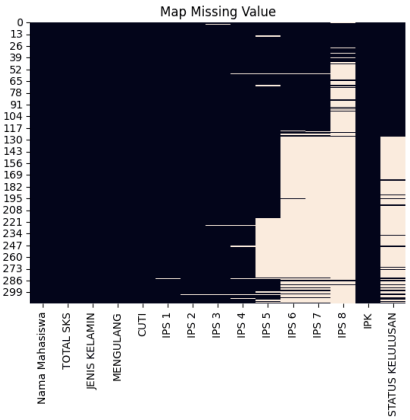
```
df=df.drop(['NIM','IPS 9','IPS 10','IPS 11','IPS 12','IPS 13','IPS 14'], axis=1)

df.head()
```

	Nama Mahasiswa	TOTAL SKS	JENIS KELAMIN	MENGULANG	CUTI	IPS 1	IPS 2	IPS 3	IPS 4	IPS 5	IPS 6	IPS 7	IPS 8	IPK	STATUS KELULUSAN
0	ABDA KURNIAWAN PRIYAMBADA	156	2	2	1	3.43	3.35	3.52	3.74	3.76	3.18	1.5	0.0	3.03	2.0
1	ABDUL AZIZ	152	1	1	1	3.12	3.64	3.66	3.70	3.70	4.00	4.0	NaN	3.68	1.0
2	ACHMAD ALVIN ARDIANSYAH	125	1	1	1	3.69	3.76	3.81	3.76	3.81	NaN	NaN	NaN	3.77	NaN
3	ACHMAD ASRORI	152	1	1	1	3.13	3.71	3.24	3.55	3.76	4.00	4.0	NaN	3.48	1.0
4	ACHMAD MAULUDI ASROR	149	1	2	2	3.89	3.54	3.22	3.54	3.22	3.96	0.0	NaN	2.93	2.0

   DataFrame yang dimodifikasi dengan kolom tersebut dihapus.drop()axis=1

2. Missing Value

   Missing value dapat divisualisasikan dengan map menggunakan fungsi dari perpustakaan Seaborn. Metode ini mengembalikan DataFrame dengan bentuk yang sama dengan 'df', dengan nilai True di mana ada nilai yang hilang dan nilai False di mana data ada. Heat map untuk memvisualisasikan DataFrame ini, di mana nilai yang hilang disorot.sns.heatmap()df.isnull() sebagai berikut :



Gambar 8. *Map missing value*

Terdapat 1 data atau 0.3% pada IPS 1, 1 data atau 0.3% pada IPS 2, 3 data atau 0.9% pada IPS 3, 6 data atau 1.9% pada IPS 4, 75 data atau 24% pada IPS 5, 179 data atau 57.3% pada IPS 6, 180 data atau 57.6% pada IPS 7, 260 data atau 83% pada IPS 8 dan 162 data atau 51% pada Status Kelulusan.

```
----Persen Missing Value----
Nama Mahasiswa        0.000000
TOTAL SKS             0.000000
JENIS KELAMIN         0.000000
MENGULANG             0.000000
CUTI                  0.000000
IPS 1                 0.320513
IPS 2                 0.320513
IPS 3                 0.961538
IPS 4                 1.923077
IPS 5                24.038462
IPS 6                57.371795
IPS 7                57.692308
IPS 8                83.333333
IPK                   0.000000
STATUS KELULUSAN     51.923077
dtype: float64
```

```
Jumlah Missing Value
Jumlah Data
Nama Mahasiswa         0
TOTAL SKS              0
JENIS KELAMIN          0
MENGULANG              0
CUTI                   0
IPS 1                  1
IPS 2                  1
IPS 3                  3
IPS 4                  6
IPS 5                 75
IPS 6                179
IPS 7                180
IPS 8                260
IPK                    0
STATUS KELULUSAN     162
dtype: int64
```

Gambar 9. *Missing value*

3. *Handling missing value*

Menangani nilai yang hilang pada IP Semester 1 sampai 8 dengan menghitung nilai rata-rata kolom "IPS 1 sampai 8" di DataFrame 'df' lalu mengganti nilai yang hilang (NaN) di kolom tersebut dengan nilai rata-rata terhitung dengan output :

Nama Mahasiswa	TOTAL SKS	JENIS KELAMIN	MENGULANG	CUTI	IPS 1	IPS 2	IPS 3	IPS 4	IPS 5	IPS 6	IPS 7	IPS 8	IPK
ABDA KURNIAWAN PRIYAMBADA	156	2	2	1	3.43	3.35	3.52	3.74	3.760000	3.180000	1.500000	0.0000	3.03
ABDUL AZIZ	152	1	1	1	3.12	3.64	3.66	3.70	3.700000	4.000000	4.000000	0.7025	3.68
ACHMAD ALVIN ARDIANSYAH	125	1	1	1	3.69	3.76	3.81	3.76	3.810000	3.732115	3.031176	0.7025	3.77
ACHMAD ASRORI	152	1	1	1	3.13	3.71	3.24	3.55	3.760000	4.000000	4.000000	0.7025	3.48
ACHMAD MAULUDI ASROR	149	1	2	2	3.89	3.54	3.22	3.54	3.220000	3.960000	0.000000	0.7025	2.93
...	...	...	...	...	...	...	...	...	...	...	...	...	...
FAUZAN ALI GHOFUR	152	1	1	1	3.65	3.66	3.63	3.63	3.660000	4.000000	4.000000	0.7025	3.73
FAZA SALSABILA	128	2	1	1	3.70	3.68	3.81	3.73	3.400000	3.732115	3.031176	0.7025	3.67
FEBRI PUJIANI	125	2	1	1	3.65	3.70	3.78	3.51	3.400000	3.732115	3.031176	0.7025	3.62
FEBRI TRI PRASETYO	128	2	1	1	3.75	3.61	3.88	3.69	3.550000	3.732115	3.031176	0.7025	3.70
FEBRIAN DAFFA EKA PUTRA	82	2	1	1	3.98	3.92	3.88	3.75	3.639524	3.732115	3.031176	0.7025	3.88

Gambar 10. *Handling missing value*

C. *Exploration Data*

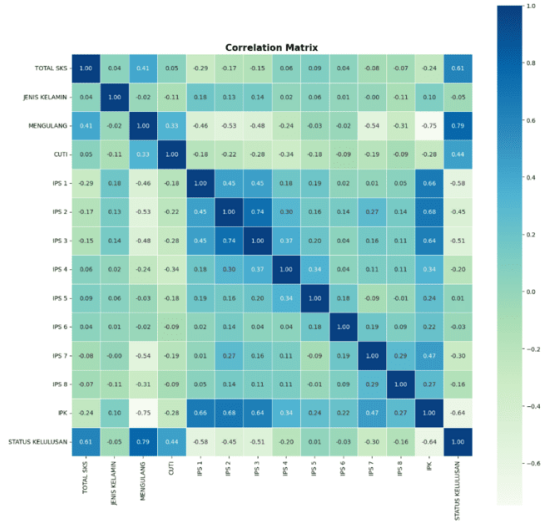
1. *Correlation*

Mencari korelasi antar variable guna untuk mengetahui pengaruh antar variable dengan menggunakan fungsi df.corr()

	TOTAL SKS	JENIS KELAMIN	MENGULANG	CUTI	IPS 1	IPS 2	IPS 3	IPS 4	IPS 5	IPS 6	IPS 7	IPS 8	IPK	STATUS KELULUSAN
TOTAL SKS	1.000000	0.043746	0.382348	0.047596	-0.291991	0.094153	-0.146656	0.061444	0.094457	0.043021	-0.079142	-0.068608	-0.244476	0.596517
JENIS KELAMIN	0.043746	1.000000	-0.001152	-0.113557	0.177880	-0.051077	0.144826	0.016980	0.059189	0.007449	-0.004874	-0.106705	0.100575	-0.066717
MENGULANG	0.382348	-0.001152	1.000000	0.354251	-0.479517	0.093535	-0.502686	-0.381031	-0.210057	-0.094151	-0.495462	-0.282114	-0.741922	0.786373
CUTI	0.047596	-0.113557	0.354251	1.000000	-0.184915	-0.018896	-0.280979	-0.335877	-0.183388	-0.092138	-0.191351	-0.085502	-0.275640	0.426375
IPS 1	-0.291991	0.177880	-0.479517	-0.184915	1.000000	0.002589	0.454909	0.181082	0.190036	0.020757	0.008052	0.047470	0.661782	-0.594720
IPS 2	0.094153	-0.051077	0.093535	-0.018896	0.002589	1.000000	0.009973	0.033889	0.056186	0.016118	-0.143671	-0.095752	-0.088257	0.079873
IPS 3	-0.146656	0.144826	-0.502686	-0.280979	0.454909	0.009973	1.000000	0.373705	0.198439	0.044505	0.156057	0.113730	0.641446	-0.525399
IPS 4	0.061444	0.016980	-0.381031	-0.335877	0.181082	0.033889	0.373705	1.000000	0.341359	0.044651	0.114522	0.114015	0.335455	-0.236841
IPS 5	0.094457	0.059189	-0.210057	-0.183388	0.190036	0.056186	0.198439	0.341359	1.000000	0.182961	-0.089298	-0.009748	0.243310	-0.082695
IPS 6	0.043021	0.007449	-0.094151	-0.092138	0.020757	0.016118	0.044505	0.044651	0.182961	1.000000	0.192868	0.092884	0.219831	-0.099683
IPS 7	-0.079142	-0.004874	-0.495462	-0.191351	0.008052	-0.143671	0.156057	0.114522	-0.089298	0.192868	1.000000	0.289800	0.469658	-0.287082
IPS 8	-0.068608	-0.106705	-0.282114	-0.085502	0.047470	-0.095752	0.113730	0.114015	-0.009748	0.092884	0.289800	1.000000	0.273723	-0.151152
IPK	-0.244476	0.100575	-0.741922	-0.275640	0.661782	-0.088257	0.641446	0.335455	0.243310	0.219831	0.469658	0.273723	1.000000	-0.660772
STATUS KELULUSAN	0.596517	-0.066717	0.786373	0.426375	-0.594720	0.079873	-0.525399	-0.236841	-0.082695	-0.099683	-0.287082	-0.151152	-0.660772	1.000000

Gambar 11. *correlation*

Terdapat tiga variable yang paling berpengaruh terhadap status kelulusan ialah Total SKS, Mengulang, dan Cuti. Untuk mencari korelasi antar variable dapat juga menggunakan correlation matrix dengan menggunakan library matplotlib.pyplot dan seaborn meggunakan metode pearson. Dengan hasil correlation matrix sebagai berikut :



Gambar 12. *Correlation matrix*

Pada gambar matriks korelasi diatas dapat dijelaskan semakin tua warnanya maka tingkat pengaruh antar feature semakin kuat, jika warna semakin muda atau meuju ke warna putih, paka tingkat pengaruh antar feature semakin lemah.

2. *Outlier*

Rumus kuartil outlier digunakan untuk mengidentifikasi adanya data pencilan atau outlier dalam sebuah set data. Kuartil pertama (Q1) dan kuartil ketiga (Q3) merupakan nilai yang membagi data menjadi empat bagian yang sama besar. Rentang kuartil (IQR) adalah selisih antara Q3 dan Q1. Untuk menentukan batas outlier, kita mengambil 1.5 kali IQR di bawah Q1 untuk batas bawah outlier dan 1.5 kali IQR di atas Q3 untuk batas atas outlier. Jika terdapat data di luar batas-batas ini, maka data tersebut dianggap sebagai outlier. Rumus ini membantu dalam mengidentifikasi nilai-nilai yang berbeda secara signifikan dari data lainnya.

Menghitung persentase *outlier* dalam kaitannya dengan jumlah total baris dalam DataFrame asli . Dengan membagi panjang (yang mewakili jumlah baris *outlier*) dengan panjang (yang mewakili jumlah total baris dalam DataFrame) dan kemudian mengalikannya dengan 100, bisa mendapatkan persentase *outlier*.**len (out_all)/len(df)*100df out_all df**

```
print('Persentage Outlier')
len(out_all)/len(df)*100
```

```
Persentage Outlier
41.34615384615385
```

Gambar 13. *Presentage outlier*

Dapat disimpulkan daru fungsi diatas menghasilkan 41.3% *outlier* atau 129 data dari 312 data. *Drop outlier all feature* yaitu menghilangkan 41.3% atau 129 data *outlier* atau data yang menyimpang pada semua *feature* dengan fungsi df.drop.

D. *Decision Tree*

Plotting pohon keputusan dari model Decision Tree berfungsi untuk memvisualisasikan struktur pohon keputusan yang telah dipelajari oleh model. Visualisasi ini membantu dalam pemahaman dan interpretasi model, serta memberikan gambaran yang jelas tentang bagaimana model membuat keputusan berdasarkan fitur-fitur yang ada. Dengan menggunakan plot tree Decision Tree, berfungsi sebagai:

1. Node dan cabang: mengetahui setiap node dalam pohon keputusan dan cabang-cabang yang menghubungkannya. Node mewakili keputusan atau aturan yang diambil berdasarkan fitur-fitur tertentu, sedangkan cabang menggambarkan arah yang diambil oleh aliran keputusan.
2. Fitur-fitur penting: mengetahui fitur-fitur yang paling penting dalam pembuatan keputusan oleh model. Fitur-

fitur tersebut terletak pada node-node penting dalam pohon keputusan, dan mungkin memberikan wawasan

```
from sklearn import tree
decision_gradprama = tree.DecisionTreeClassifier(max_depth = 5,
                                                 min_samples_split = 10,
                                                 min_samples_leaf = 10)
```

tentang atribut-atribut yang memiliki pengaruh besar dalam prediksi.

Gambar 14. *Decision Tree*

a. from sklearn import tree: Ini adalah pernyataan untuk mengimpor modul tree dari library sklearn. Modul ini berisi implementasi model Decision Tree.
b. decision_gradprama = tree.DecisionTreeClassifier(max_depth=5, min_samples_split=10, min_samples_leaf=10): Pada baris ini, objek decision_gradprama diinisialisasi sebagai model Decision Tree menggunakan DecisionTreeClassifier().
   - **max_depth**: Parameter ini mengatur kedalaman maksimum pohon keputusan yang akan dibangun. Dalam contoh ini, nilai **max_depth** ditetapkan menjadi 5, sehingga pohon keputusan akan memiliki kedalaman maksimum 5.
   - **min_samples_split**: Parameter ini mengatur jumlah minimum sampel yang diperlukan untuk membagi node internal. Jika jumlah sampel dalam suatu node kurang dari **min_samples_split**, node tersebut tidak akan dibagi. Dalam contoh ini, nilai **min_samples_split** ditetapkan menjadi 10.
   - **min_samples_leaf**: Parameter ini mengatur jumlah minimum sampel yang diperlukan dalam setiap daun (leaf) pohon keputusan. Jika jumlah sampel dalam suatu daun kurang dari **min_samples_leaf**, pohon keputusan akan berhenti membagi node. Dalam contoh ini, nilai **min_samples_leaf** ditetapkan menjadi 10.

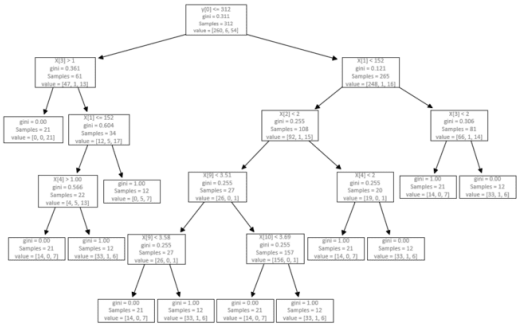
```
tree.plot_tree(decision_gradprama.fit(x_train, y_train))
```

```
[Text(140.32058823529414, 199.32, 'X[5] <= 8.015\ngini = 0.311\nsamples = 320\nvalue = [260, 6, 54]'),
 Text(39.38823529411765, 163.07999999999998, 'X[3] <= 1.75\ngini = 0.467\nsamples = 55\nvalue = [12, 5, 38]'),
 Text(19.694117647058825, 126.83999999999999, 'gini = 0.0\nsamples = 21\nvalue = [0, 0, 21]'),
 Text(59.082352941176474, 126.83999999999999, 'X[0] <= 306.5\ngini = 0.604\nsamples = 34\nvalue = [12, 5, 17]'),
 Text(39.38823529411765, 90.6, 'X[5] <= 7.67\ngini = 0.568\nsamples = 22\nvalue = [4, 5, 13]'),
 Text(19.694117647058825, 54.359999999999985, 'gini = 0.486\nsamples = 12\nvalue = [0, 5, 7]'),
 Text(59.082352941176474, 54.359999999999985, 'gini = 0.48\nsamples = 10\nvalue = [4, 0, 6]'),
 Text(78.77647058823531, 90.6, 'gini = 0.444\nsamples = 12\nvalue = [8, 0, 4]'),
 Text(241.25294117647056, 163.07999999999998, 'X[5] <= 8.645\ngini = 0.121\nsamples = 265\nvalue = [248, 1, 16]'),
 Text(187.09411764705882, 126.83999999999999, 'X[4] <= 3.75\ngini = 0.255\nsamples = 108\nvalue = [92, 1, 15]'),
 Text(137.85882352941178, 90.6, 'X[0] <= 316.5\ngini = 0.306\nsamples = 81\nvalue = [66, 1, 14]'),
 Text(98.47058823529412, 54.359999999999985, 'X[6] <= 0.5\ngini = 0.361\nsamples = 61\nvalue = [47, 1, 13]'),
 Text(78.27647058823531, 18.119999999999976, 'gini = 0.296\nsamples = 40\nvalue = [33, 1, 6]'),
 Text(118.16470588235295, 18.119999999999976, 'gini = 0.444\nsamples = 21\nvalue = [14, 0, 7]'),
 Text(177.24705882352941, 54.359999999999985, 'X[0] <= 318.5\ngini = 0.095\nsamples = 20\nvalue = [19, 0, 1]'),
 Text(157.55294117647056, 18.119999999999976, 'gini = 0.0\nsamples = 10\nvalue = [10, 0, 0]'),
 Text(196.94117647058823, 18.119999999999976, 'gini = 0.18\nsamples = 10\nvalue = [9, 0, 1]'),
 Text(236.32941176470589, 90.6, 'X[3] <= 3.75\ngini = 0.071\nsamples = 27\nvalue = [26, 0, 1]'),
 Text(216.63529411764708, 54.359999999999985, 'gini = 0.0\nsamples = 16\nvalue = [16, 0, 0]'),
 Text(256.0235294117647, 54.359999999999985, 'gini = 0.165\nsamples = 11\nvalue = [10, 0, 1]'),
 Text(295.41176470588234, 126.83999999999999, 'X[5] <= 8.705\ngini = 0.013\nsamples = 157\nvalue = [156, 0, 1]'),
 Text(275.7176470588236, 90.6, 'gini = 0.1\nsamples = 19\nvalue = [18, 0, 1]'),
 Text(315.1058823529412, 90.6, 'gini = 0.0\nsamples = 138\nvalue = [138, 0, 0]')]
```

Gambar 15. *decision_gradprama*

Pada kode di atas, setelah menginisialisasi objek **decision_gradprama** sebagai model *Decision Tree* dan melatihnya dengan data latih (**x_train** dan **y_train**), kita menggunakan **tree.plot_tree()** untuk melakukan plotting pohon keputusan. Beberapa hal yang perlu diperhatikan dalam kode tersebut:

- Mengimpor modul **tree** dan **pyplot** dari library **sklearn** dan **matplotlib** secara berurutan.
- Mengatur ukuran figur menggunakan **plt.figure()** agar plot terlihat dengan baik.
- Fungsi **plot_tree()** digunakan untuk melakukan plotting pohon keputusan. Argumen yang diberikan adalah model Decision Tree yang telah dilatih (**decision_gradprama**), nama fitur (**feature_names**), nama kelas (**class_names**), dan **filled=True** untuk memberi warna pada node berdasarkan kelas mayoritas.



Gambar 16. *Tree decision Tree*

1. Dalam notasi X [indeks array], variabel independen dilambangkan dengan X, dan jika X [1], itu berarti X adalah total SKS. Namun, jika y [indeks array], y adalah variabel dependen, yaitu status kelulusan yang akan diprediksi.
2. Gini (Gini impurity) adalah metrik yang digunakan dalam Decision Tree untuk mengukur ketidakmurnian (impurity) suatu node. Semakin rendah nilai gini, semakin homogen atau murni data pada node tersebut. Gini impurity dihitung berdasarkan distribusi kelas pada node tersebut. Nilai gini berkisar antara 0 dan 1, di mana 0 menunjukkan keadaan yang sangat murni (semua sampel pada node termasuk dalam satu kelas), dan 1 menunjukkan keadaan yang sangat tidak murni (sampel terdistribusi merata di antara kelas-kelas yang ada).
3. Istilah "Sample" mengacu pada jumlah total sampel yang ada pada suatu node dalam Decision Tree. Informasi ini menunjukkan jumlah sampel yang termasuk dalam node tersebut dan digunakan untuk membuat keputusan berdasarkan aturan yang ditentukan oleh pohon keputusan.
4. "Value" menunjukkan distribusi kelas pada suatu node dalam Decision Tree. Informasi ini menunjukkan jumlah sampel dalam setiap kelas pada node tersebut. Misalnya, jika ada 10 sampel pada node dan 8 dari mereka termasuk dalam kelas A dan 2 dalam kelas B, maka nilai "value" akan ditampilkan sebagai [8, 2], menunjukkan jumlah sampel pada masing-masing kelas.

```
print(gs_dt.best_score_)
```

```
0.7595712098009189
```

Gambar 17. *accuracy*

y pred atau variabel prediksi (status kelulusan) memiliki dua scenario percabangan pada 312 dataset memperoleh nilai akurasi 0.75 atau jika

dikalikan dengan 100% ialah 75.95%, dengan scenario :

1. *Indeks* y < 312 akan terdapat seleksi variabel X[3] atau mengulang, jika mahasiswa tersebut mengulang > 1 atau bernilai 2 maka seara otomatis akan tidak lulus tepat waktu, tetapi jika megulang = 1 atau tidak mengulang maka akan berlanjut ke total SKS. Pada X[1] total SKS, jika SKS > 158 maka secara otomatis akan tidak lulus tepat waktu, tetapi jika SKS < 150 akan berlanjut ke cuti. Pada X[4] cuti jika nilai cuti > 1 atau melakukan cuti, maka secara otomatis akan tidak lulus tepat waktu, tetapi jika nilai cuti = 1 atau tidak melakukan cuti, maka mahasiswa tersebut akan lulus tepat waktu.
2. *Indeks* y > 312 atau dari 312 akan terdapat seleksi variabel x[1] total SKS > 152 maka akan berlanjut ke mengulang dan jika pada X[3] atau mengulang memiiki nilai < 2 maka akan lulus tepat waktu, tetapi jika mengulang > 2 maka tidak lulus tepat waktu. Tetapi jika pada X[1] total SKS < 152 akan berlanjut pada percabangan jenis kelamin. Pada X[2] jenis kelamin = 2 akan bercabang ke kanan dengan X[4] cuti, pada cuti memiliki dua hasil, jika nilai cuti = 2 maka tidak lulus tepat waktu, tetapi jika nilai cuti < 2 maka lulus tepat waktu. Jika pada percabangan jenis kelamin < 2 atau laki-laki akan bercabang ke X[9] IPS 5, jika IPS 5 > 3.51 maka akan becabang ke X[8] IPS 4, jika IPS 4 < 3.58 maka akan otomatis tidak lulus tepat waktu, tetapi jika IPS 4 > 3.58 maka akan lulus tepat waktu. Namun jika pada percabangan jenis kelamin = 2 atau perempuan maka akan bercabang pada X[10] IPS 6, jika IPS 6 > 3.69 maka akan lulus tepat waktu, tetapi jika IPS 6 < 3.69 maka akan tidak lulus tepat waktu.

IV. KESIMPULAN

Kesimpulan yang dapat diambil dari penelitian ini menggunakan dataset 312 mahasiswa program studi sistem informasi jurusan teknik informatika di Universitas Surabaya, dengan pelatihan model menggunakan algoritma Decision Tree menghasilkan akurasi model sebesar 75.95%.

Variabel independen yang mempengaruhi Status Kelulusan, berdasarkan matriks korelasi, adalah mengulang matakuliah (korelasi 0.786), total SKS (korelasi 0.610), cuti (korelasi 0.444), IPS 5 (korelasi 0.011), IPS 6 (korelasi -0.029), jenis kelamin (korelasi -0.053), IPS 8 (korelasi -0.156), IPS 4 (korelasi -0.196), IPS 7 (korelasi -0.298), IPS 2 (korelasi -0.453), IPS 3 (korelasi -0.506), IPS 1 (korelasi -0.578), dan IPK (korelasi -0.641).

UCAPAN TERIMA KASIH

Peneliti menyatakan rasa syukur dan terima kasih kepada Tuhan yang telah memberikan pertolongan dan rahmat-Nya, sehingga peneliti dapat menyelesaikan jurnal ini. Peneliti juga mengucapkan terima kasih kepada semua pihak yang telah membantu dan memberikan dukungan sehingga jurnal ini berhasil diselesaikan dengan baik. Semua kontribusi dan bantuan dari berbagai pihak sangat berarti dan sangat dihargai oleh peneliti. Terima kasih atas segala upaya dan peran aktif yang telah diberikan dalam penyelesaian jurnal ini.

REFERENSI

[1] DEPDIKNAS, B. (2007). Buku II: Standar dan Prosedur Akreditasi Institusi Perguruan Tinggi.

[2] Rohmawan, E. P. (2018). Prediksi Kelulusan Mahasiswa Tepat Waktu Menggunakan Metode Desicion Tree Dan Artificial Neural Network. *Jurnal Ilmiah MATRIK*, *20*(1), 21-30.

[3] Hand, D. J. (2007). Principles of data mining. *Drug safety*, *30*, 621-622.

[4] Huda, N. M. (2011). *Aplikasi Data Mining Untuk Menampilkan Informasi Tingkat Kelulusan Mahasiswa (Studi Kasus di Fakultas MIPA Universitas Diponegoro)* (Doctoral dissertation, UNDIP).