

**KLASIFIKASI MAHASISWA BERPOTENSI PUTUS STUDI
MENGUNAKAN *ALGORITMA DECISION TREE* PADA FAKULTAS
TEKNIK UNISMUH MAKASSAR**

SKRIPSI

Diajukan Sebagai Salah Satu Syarat Untuk Mendapat
Gelar Sarjana Komputer (S.Kom) Program Studi Informatika



YUMI

105841102120

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH MAKASSAR**

2024



UNIVERSITAS MUHAMMADIYAH MAKASSAR

FAKULTAS TEKNIK

GEDUNG MENARA IQRA LT. 3

Jl. Sultan Alauddin No. 259 Telp. (0411) 866 972 Fax (0411) 865 588 Makassar 90221

Website: www.unismuh.ac.id, e_mail: unismuh@gmail.com

Website: <http://teknik.unismuh.makassar.ac.id>

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

PENGESAHAN

Skripsi atas nama Yumi dengan nomor induk Mahasiswa 105 84 11021 20, dinyatakan diterima dan disahkan oleh Panitia Ujian Tugas Akhir/Skripsi sesuai dengan Surat Keputusan Dekan Fakultas Teknik Universitas Muhammadiyah Makassar Nomor : 0010/SK-Y/55202/091004/2024, sebagai salah satu syarat guna memperoleh gelar Sarjana Komputer pada Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar pada hari Senin tanggal 30 Agustus 2024.

Panitia Ujian :

Makassar, 25 Safar 1446 H
30 September 2024 M

1. Pengawas Umum

a. Rektor Universitas Muhammadiyah Makassar

Dr. Ir. H. Abd. Rakhim Nanda, ST., MT., IPU

b. Dekan Fakultas Teknik Universitas Hasanuddin

Prof. Dr. Eng. Muhammad Isran Ramli, ST., MT.

2. Penguji

a. Ketua : Dr. Ir. Zahir Zainuddin, M.Sc.

b. Sekretaris : Desi Anggreani, S.Kom., MT.

3. Anggota

1. Muhyiddin A. M. Hayat, S.Kom., MT.

2. Titin Wahyuni, S.Pd., MT.

3. Fahim Irfhamna Rahman, S.Kom., MT.

Mengetahui :

Pembimbing I

Pembimbing II

Lukman, S.Kom., MT.

Rizki Yusliana Bakti, ST., MT.

Dekan

Dr. Ir. H. Nurmawaty, ST., MT., IPM.
NPM: 795 108



UNIVERSITAS MUHAMMADIYAH MAKASSAR

FAKULTAS TEKNIK

GEDUNG MENARA IQRA LT. 3

Jl. Sultan Alauddin No. 259 Telp. (0411) 866 972 Fax (0411) 865 588 Makassar 90221

Website: www.unismuh.ac.id, e_mail: unismuh@gmail.com

Website: <http://teknik.unismuh.makassar.ac.id>

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

HALAMAN PENGESAHAN

Tugas Akhir ini diajukan untuk memenuhi syarat ujian guna memperoleh gelar Sarjana Komputer (S.Kom) Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar.

Judul Skripsi : **KLASIFIKASI MAHASISWA BERPOTENSI PUTUS STUDI MENGGUNAKAN ALGORITMA DECISION TREE PADA FAKULTAS TEKNIK UNISMUH MAKASSAR**

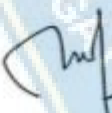
Nama : Yumi
Stanbuk : 105841102120

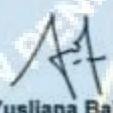
Makassar, 02 September 2024

Telah Diperiksa dan Disetujui
Oleh Dosen Pembimbing:

Pembimbing I

Pembimbing II


Lukman, S.Kom., MT.


Rizki Yusliana Bakti, ST., MT.

Mengetahui,

Ketua Program Studi Informatika


Muhyddin A. M. Hayat, S.Kom., MT

NBM - 1504577

ABSTRAK

Universitas Muhammadiyah Makassar (Unismuh Makassar) menghadapi tantangan signifikan dalam menangani masalah mahasiswa yang berpotensi putus studi, terutama di Fakultas Teknik. Faktor-faktor seperti rendahnya kemampuan akademik, keterbatasan biaya, dan kendala tempat tinggal menjadi pemicu utama masalah ini, yang pada gilirannya dapat menghambat kemajuan perguruan tinggi. Penelitian ini bertujuan untuk mengklasifikasikan mahasiswa yang berpotensi putus studi menggunakan Algoritma Decision Tree C4.5 dan mengevaluasi tingkat akurasi sistem klasifikasi tersebut. Penelitian ini menggunakan metode kuantitatif dengan analisis statistik deskriptif. Dataset yang dianalisis terdiri dari 5657 mahasiswa Fakultas Teknik Unismuh Makassar, dengan atribut-atribut seperti pekerjaan dan penghasilan orang tua/wali, IPK, SKS, dan variabel lainnya yang relevan dengan status akademik mahasiswa. Hasil penelitian menunjukkan bahwa Algoritma Decision Tree C4.5 mampu mengklasifikasikan mahasiswa yang berpotensi putus studi dengan akurasi sebesar 100%. Nilai rata-rata precision dan recall masing-masing adalah 100%, sedangkan nilai rata-rata f1-score mencapai 100%. Temuan ini mengindikasikan bahwa Algoritma Decision Tree C4.5 memiliki performa yang tinggi dan merupakan metode yang efektif dalam mengidentifikasi mahasiswa dengan risiko putus studi, sehingga memungkinkan perguruan tinggi untuk mengambil langkah-langkah preventif yang lebih tepat sasaran.

Kata kunci: putus studi, *Algoritma Decision Tree*, klasifikasi, Fakultas Teknik, Universitas Muhammadiyah Makassar.

ABSTRACT

Muhammadiyah University of Makassar (Unismuh Makassar) faces significant challenges in addressing the issue of students at risk of dropping out, particularly in the Faculty of Engineering. Factors such as low academic performance, financial constraints, and housing issues are the main triggers of this problem, which in turn can hinder the progress of the university. This study aims to classify students at risk of dropping out using the C4.5 Decision Tree Algorithm and evaluate the accuracy level of the classification system. This research employs a quantitative method with descriptive statistical analysis. The analyzed dataset consists of 5657 students from the Faculty of Engineering at Unismuh Makassar, with attributes including the occupations and income of parents/guardians, IPK, SKS, and other variables relevant to students' academic status. The results indicate that the C4.5 Decision Tree Algorithm can classify students at risk of dropping out with an accuracy of 100%. The average precision and recall values are both 100%, while the average F1-score reaches 100%. These findings suggest that the C4.5 Decision Tree Algorithm has high performance and is an effective method for identifying students at risk of dropping out, enabling universities to take more targeted preventive measures.

Keywords: *dropout, Decision Tree Algorithm, classification, Faculty of Engineering, Muhammadiyah University of Makassar.*

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Assalamu'alaikum Warahmatullahi Wabarakatuh

Puji syukur penulis panjatkan kehadiran Allah SWT, atas limpahan rahmat dan hidayah-Nya sehingga Proposal Skripsi yang berjudul “KLASIFIKASI MAHASISWA BERPOTENSI PUTUS STUDI MENGGUNAKAN ALGORITMA DECISION TREE PADA FAKULTAS TEKNIK UNISMUH MAKASSAR” ini dapat di rampungkan.

Proposal skripsi ini di ajukan untuk memenuhi salah satu persyaratan yang harus ditempuh dalam rangka menyelesaikan Studi di Fakultas Teknik Program Studi Informatika Universitas Muhammadiyah Makassar. Dengan selesainya proposal skripsi ini tidaklah berarti bahwa proposal skripsi ini sudah dalam bentuk yang sempurna. Oleh karena itu, kritik dan saran sangat diharapkan dari pembaca demi kesempurnaan proposal skripsi ini.

Penulis menyadari bahwa selama dalam penyusunan proposal skripsi ini banyak pihak yang telah membantu dan memberikan dukungannya. Tak lupa penulis mengucapkan banyak-banyak terima kasih kepada seluruh pihak yang telah membantu dalam penyusunan proposal ini. Oleh karena itu, penghargaan yang setinggi-tingginya kami haturkan kepada:

1. Allah SWT, yang hingga sampai sekarang masih menyetatkan penulis sehingga bisa sampai pada di titik ini.
2. Kedua orang tua kami tercinta dan keluarga, penulis mengucapkan terima kasih atas segala doa dan dukungan baik secara moral maupun materi.
3. Rektor Universitas Muhammadiyah Makassar Bapak Dr. Ir. H. Rakhim Nanda, S.T., M.T.
4. Ibu **Dr.Ir.Hj Nurnawati, S.T., M.T., I.P.M**, selaku Dekan Fakultas Teknik Universitas Muhammadiyah Makassar.

5. Bapak **Muhyiddin AM Hayat, S.Kom., M.T**, selaku Ketua Prodi Informatika.
6. Bapak **Lukman, S.Kom., M.T**, selaku Dosen Pembimbing I Proposal.
7. Ibu **Rizki Yusliana Bakti, S.T., M.T**, selaku Dosen Pembimbing II Proposal.
8. Bapak/Ibu Dosen dan Staf Fakultas Teknik Universitas Muhammadiyah Makassar.
9. Teman-temanku, dengan rasa persaudaraan yang tinggi banyak membantu serta memberi dukungan dalam menyelesaikan tugas proposal skripsi ini.

Akhir kata, penulis mengharapkan tugas proposal skripsi ini dapat bermanfaat bagi pembaca. Aamiin.

“Billahi Fii Sabilil Haq Fastabiqul Khaerat”

Wassalamu’alaikum Warahmatullahi Wabarakatuh

Makassar, 16 Mei 2024

Yumi

DAFTAR ISI

ABSTRAK	ii
<i>ABSTRACT</i>	v
KATA PENGANTAR	vi
DAFTAR ISI	vi
DAFTAR GAMBAR	x
DAFTAR TABEL	xi
DAFTAR LAMPIRAN	xii
DAFTAR ISTILAH	xiii
BAB I PENDAHULUAN	1
A. Latar Belakang	1
B. Rumusan Masalah	3
C. Tujuan Penelitian	3
D. Manfaat Penelitian	3
E. Ruang Lingkup Penelitian	4
F. Sistematika Penulisan	4
BAB II TINJAUAN PUSTAKA	6
A. Landasan Teori	6
B. Penelitian Terkait	14
C. Kerangka Berpikir	18
BAB III METODE PENELITIAN	19
A. Tempat dan Waktu Penelitian	19
B. Alat dan Bahan	19
C. Perancangan Sistem	19
D. Teknik Pengujian Sistem	22
E. Teknik Analisis Data	22
BAB IV HASIL DAN PEMBAHASAN	24
A. Pengumpulan Data	24
B. Data Mentah	24
C. Data Preprocessing	27
D. Pembagian Data	43

E. Klasifikasi Menggunakan Algoritma Decision Tree C4.5	44
F. Pengujian Menggunakan <i>Cross Validation</i>	44
G. Evaluasi Hasil Klasifikasi	45
BAB V PENUTUP	50
A. Kesimpulan	50
B. Saran	50
DAFTAR PUSTAKA	51
LAMPIRAN	53



DAFTAR GAMBAR

Gambar 1. Konsep Decision Tree.....	9
Gambar 2. Diagram Kerangka Pikir	18
Gambar 3. Kerangka Penelitian	20
Gambar 4 Flowchart algoritma decision tree.....	21
gambar 5. Data Mentah.....	26
Gambar 6. Hasil Penambahan Atribut	29
Gambar 7. Hasil Codingan Atribut Target.....	31
Gambar 8 Codingan dan Hasil Kategori.....	32
Gambar 9. Hasil Seleksi Atribut.....	33
Gambar 10. Hasil Penghapusan Atribut	34
Gambar 11. Data missing value.....	35
Gambar 12. Codingan dan Hasil Transformasi Data.....	41
Gambar 13. <i>Confussion Matrix</i> Pada <i>Algoritma Decision Tree C4.5</i>	46
Gambar 14. Hasil Pengukuran Kinerja Algoritma Decision Tree C4.5	47
gambar 15. Pohon Keputusan	48

DAFTAR TABEL

Tabel 1. Confusion matrix.....	13
Tabel 2. Penelitian terkait	14
Tabel 3. Kategori pada atribut ipk terakhir	31
Tabel 4. Kategori pada atribut sks lulus.....	31
Tabel 5. Jumlah Data Missing Value	34
Tabel 6. data missing value setelah diubah.....	35
Tabel 7. Jumlah Data Hasil Penghapusan	36
Tabel 8. Pelabelan Pada Atribut ayah_pekerjaan.....	41
Tabel 9. Pelabelan Pada Atribut ayah_penghasilan	41
Tabel 10. Pelabelan Pada Atribut ayah_status	41
Tabel 11. Pelabelan Pada Atribut ibu_pekerjaan	41
Tabel 12. Pelabelan Pada Atribut ibu_penghasilan.....	42
Tabel 13. Pelabelan Pada Atribut ibu_status.....	42
Tabel 14. Pelabelan Pada Atribut wali_pekerjaan	42
Tabel 15. Pelabelan Pada Atribut wali_penghasilan.....	42
Tabel 16. Pelabelan Pada Atribut lulus	42
Tabel 17. Pelabelan Pada Atribut Kategori_IPK	43
Tabel 18. Pelabelan Pada Atribut kategori_SKS	43
Tabel 19. Pelabelan Pada status_putus_studi.....	43
Tabel 20. Pembagian Data	44
Tabel 21. Hasil Akurasi.....	44
Tabel 22. Hasil Pengujian 10-fold cross validation	45
Tabel 23. Confusion Matrix Pada Algoritma Decision Tree C4.5.....	46
Tabel 24. Nilai Gain Pada Masing-Masing Atribut	49

DAFTAR LAMPIRAN

Lampiran 1. Data Mentah	53
Lampiran 3. Transformasi Data	55
Lampiran 4. Codingan Dan Output.....	56
Lampiran 6. Permohonan Penelitian.....	89
Lampiran 7. Pengantar Penelitian	90
Lampiran 8. Bebas Plagiat	91
Lampiran 9. Hasil Plagiat Per BAB	92



DAFTAR ISTILAH

Klasifikasi	Proses pengelompokan atau pengkategorian data berdasarkan karakteristik tertentu.
<i>Algoritma Decision Tree</i>	Metode pembelajaran mesin yang digunakan untuk membangun model klasifikasi atau prediksi dalam bentuk pohon keputusan.
<i>Decision Tree C4.5</i>	Salah satu varian dari algoritma decision tree yang digunakan untuk membentuk model prediksi atau klasifikasi.
Fakultas Teknik	Salah satu fakultas di Universitas Muhammadiyah Makassar yang mencakup berbagai program studi terkait teknik.
Putus Studi	Situasi di mana mahasiswa berhenti atau tidak melanjutkan studinya hingga selesai.
Akurasi	Ukuran kinerja model klasifikasi yang menunjukkan persentase prediksi yang benar.
<i>Precision</i>	Ukuran seberapa banyak hasil prediksi yang relevan dari total hasil yang diprediksi sebagai positif.
<i>Recall</i>	Ukuran seberapa baik model menemukan semua data relevan dari keseluruhan data yang seharusnya diprediksi sebagai positif.
<i>f1-score</i>	Ukuran yang menggabungkan precision dan recall untuk memberikan gambaran keseluruhan tentang kinerja model klasifikasi.
Dataset	Kumpulan data yang digunakan dalam penelitian atau analisis.
Data Mining	Proses penggalian informasi atau pola penting dari kumpulan data yang besar, yang digunakan untuk memecahkan masalah atau membuat prediksi.
<i>Entropi</i>	Ukuran ketidakpastian atau keacakan dalam data yang

digunakan dalam proses pembuatan pohon keputusan untuk memilih atribut terbaik dalam memisahkan kelas.

Gain Nilai yang digunakan untuk mengukur seberapa besar suatu atribut memberikan informasi yang berguna dalam memisahkan kelas data pada pohon keputusan.

Data testing Sub set dari data yang digunakan untuk menguji kinerja model atau sistem. Ini digunakan untuk mengukur seberapa baik model dapat memprediksi data yang tidak pernah dilihat selama training atau pelatihan.

Data training Sub set dari data yang digunakan untuk melatih model atau sistem. Ini adalah data yang digunakan oleh model untuk belajar pola atau hubungan dalam data.

Flowchart Diagram yang menampilkan langkah-langkah dan keputusan untuk melakukan sebuah proses dari suatu program. Setiap langkah digambarkan dalam bentuk diagram dan dihubungkan dengan garis atau arah panah.

Cross Validation Teknik evaluasi kinerja model dengan membagi dataset menjadi beberapa bagian atau "fold". Model dilatih pada sebagian fold dan diuji pada fold yang tersisa, lalu proses ini diulang beberapa kali untuk memastikan model memiliki performa yang baik pada data yang belum dilihat sebelumnya.

K-Fold Cross Validation Metode cross validation yang membagi dataset menjadi k bagian yang hampir sama. Setiap bagian digunakan sekali sebagai data pengujian, sementara sisanya digunakan sebagai data pelatihan. Proses ini diulang sebanyak k kali untuk mendapatkan estimasi kinerja model yang lebih akurat.

Confusion Matrix

Matriks yang digunakan untuk mengevaluasi kinerja model klasifikasi dengan menunjukkan jumlah prediksi yang benar dan salah yang dibuat oleh model. Matriks ini membantu dalam menghitung metrik seperti akurasi, precision, recall, dan f1-score.



BAB I

PENDAHULUAN

A. Latar Belakang

Universitas Muhammadiyah Makassar atau dikenal juga dengan sebutan Unismuh Makassar didirikan pada tanggal 19 Juni 1963 sebagai cabang dari Universitas Muhammadiyah Jakarta. Sejak berdirinya hingga saat ini, Unismuh Makassar telah meluluskan alumni sebanyak lebih dari 15.000 orang. Salah satu fakultas yang ada di Universitas Muhammadiyah Makassar yaitu Fakultas Teknik. Dimana dalam Fakultas Teknik Muhammadiyah Makassar terdiri dari beberapa program studi yaitu Teknik Pengairan, Teknik Elektro, Arsitektur, Informatika dan Perencanaan Wilayah dan Kota (PWK).

Apabila ada beberapa mahasiswa yang terlambat lulus atau tidak tepat pada waktunya sehingga menjadi kendala untuk kemajuan dari perguruan tinggi tersebut. Salah satu masalah yang dihadapi perguruan tinggi adalah terdapat mahasiswa yang Putus Studi. Kasus putus studi juga bisa terjadi karena beberapa faktor seperti rendahnya kemampuan akademik, faktor biaya dan tempat tinggal saat menempuh pendidikan. Putus studi termasuk masalah yang serius dan merupakan situasi yang pernah dihadapi oleh sebagian mahasiswa di perguruan tinggi (Samasil et al., 2022). Untuk mengidentifikasi mahasiswa yang berpotensi putus studi saat ini dilakukan secara manual. Proses ini melibatkan pencacatan nilai, pembuatan tabel, dan perhitungan manual, yang membuatnya rumit dan memakan waktu. Ketidakpraktisan ini menjadi salah satu alasan mengapa sistem otomatis kemudian dikembangkan.

Untuk mencegah putus studi mahasiswa, perguruan tinggi dapat mengambil langkah-langkah *preventif* dan *efektif*. Salah satu strategi yang dapat diterapkan adalah dengan melakukan prediksi mahasiswa yang berpotensi untuk putus studi, dengan demikian, pihak kampus dapat mengambil langkah-langkah pencegahan sejak awal dan mengurangi kemungkinan terjadinya kasus putus studi.

Decision Tree merupakan sebuah *algoritma* yang termasuk pada teknik klasifikasi pada data mining. Klasifikasi adalah proses menemukan pola atau fungsi-fungsi yang mendeskripsikan dan memisahkan kelas data satu dengan lainnya, untuk dapat digunakan memprediksi data yang belum memiliki kelas data tertentu. Data mining merupakan sebuah teknik yang dipergunakan untuk menyelesaikan permasalahan pada proses data yang besar. Proses pada *decision tree* bertujuan untuk mengelompokkan atau mengklasifikasi pada data terhadap kelasnya masing-masing. Pada *decision tree* hasil dari proses yang dilakukan berupa sebuah pohon keputusan dengan rule yang didapatkan dari pohon keputusan tersebut. Pohon keputusan pada *decision tree* terbentuk berdasarkan dengan perhitungan nilai *gain* dan *entropy*. Atribut yang memiliki nilai *gain* tersebut yang didapatkan dari proses perhitungan pada pengolahan data (Arfyanti et al., 2022)

Kelebihan yang dimiliki *Decision Tree* adalah sifatnya yang fleksibel sehingga mampu meningkatkan kualitas keputusan yang dihasilkan, sedangkan kekurangan dari algoritma ini adalah akan terjadi overlap jika menggunakan data yang memiliki kelas dan kriteria dengan jumlah yang sangat banyak (Rahmadeyan and Mustakim 2023).

Penelitian terkait permasalahan putus studi banyak dilakukan para peneliti sebelumnya seperti *Kotsiantis*, *Pierrakeas* dan *Pintelas* menyebutkan bahwa sangat penting bagi dosen untuk mendeteksi mahasiswa yang cenderung putus studi sebelum mereka memasuki pertengahan masa studi. *Gerben W Dekker* menyebutkan bahwa monitoring dan dukungan terhadap mahasiswa di tahun pertama sangat penting dilakukan (Hermanto, 2020).

Berdasarkan latar belakang yang telah diuraikan, maka penulis mengambil judul “Klasifikasi Mahasiswa Berpotensi Putus Studi Menggunakan *Algoritma Decision Tree* Pada Fakultas Teknik Universitas Muhammadiyah Makassar” yang akan memprediksi mahasiswa berpotensi putus studi agar mahasiswa atau dosen

dapat membuat strategi atau merencanakan sesuatu untuk dapat melihat apa yang harus dilakukan agar kedepannya tidak terjadinya putus studi pada mahasiswa.

B. Rumusan Masalah

Berdasarkan latar belakang maka dapat dirumuskan masalah sebagai berikut:

1. Bagaimana penggunaan *Algoritma Decision Tree* untuk mengklasifikasi mahasiswa putus studi.
2. Bagaimana tingkat akurasi dari sistem klasifikasi yang menggunakan *decision tree* dalam mendeteksi mahasiswa berpotensi putus studi?

C. Tujuan Penelitian

Berdasarkan dari rumusan masalah maka tujuan penelitian yaitu:

1. Untuk mengetahui bagaimana penggunaan *Algoritma Decision Tree C4.5* untuk mengklasifikasi mahasiswa yang berpotensi putus studi.
2. Mengetahui tingkat akurasi sistem klasifikasi yang menggunakan *decision tree* dalam mendeteksi mahasiswa berpotensi putus studi.

D. Manfaat Penelitian

Adapun manfaat dari penelitian ini yaitu:

1. Bagi Peneliti
 - a. Membantu peneliti untuk meningkatkan pemahaman dan pengetahuan tentang algoritma *decision tree* dapat diterapkan untuk memprediksi mahasiswa berpotensi putus studi.
 - b. Penelitian ini dapat membantu memecahkan masalah yang berkaitan dengan putus studi mahasiswa.
2. Bagi Dosen
 - a. Manfaat bagi dosen adalah dapat memberikan masukan dan mengawasi mahasiswa yang mengalami kendala pada faktor kelulusan, sehingga dosen dapat membantu mahasiswa tersebut mengatasi masalah dan meningkatkan peluang kelulusan mereka di masa depan.

- b Dosen dapat menggunakan hasil penelitian untuk memperbaiki proses pengawasan akademik, seperti mengidentifikasi mahasiswa yang berpotensi putus studi dan memberikan bantuan untuk meningkatkan performa akademik mereka.

3. Bagi Mahasiswa

- a Penelitian ini dapat bermanfaat bagi mahasiswa dengan membantu mereka memperkirakan dan memprediksi penyebab putus studi di masa depan..
- b Penelitian ini juga membantu mahasiswa untuk merencanakan dan membuat strategi agar tidak putus studi di masa depan.

E. Ruang Lingkup Penelitian

Ruang lingkup penelitian adalah batasan atau ruang lingkup yang dijadikan fokus penelitian. Dengan bantuan daerah penelitian, wilayah atau bidang yang akan dipelajari ditentukan dan batas-batas objek atau fenomena yang diamati ditentukan.

1. Penelitian ini membutuhkan data mengenai mahasiswa, seperti, nim, status orang tua, wali, jenis kelamin, angkatan, IPK, IPS, SKS dan status mahasiswa. Data ini diperoleh dari simak fakultas teknik Universitas Muhammadiyah Makassar dan data yang diambil dimulai dari angkatan 2013-2015.
2. Dalam penelitian ini, akan digunakan *algoritma decision tree C4.5* untuk mengolah data yang telah dikumpulkan.

F. Sistematika Penulisan

Sistematika penulisan dalam penyusunan laporan proposal penelitian ini adalah sebagai berikut:

BAB I PENDAHULUAN

Pada bab ini menguraikan tentang latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, ruang lingkup penelitian dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Pada bab ini menguraikan mengenai studi pustaka terhadap teori-teori yang digunakan dalam penelitian serta aspek-aspek yang mendukung dalam penelitian.

BAB III METODOLOGI PENELITIAN

Bab ini berisi penjelasan tentang langkah-langkah yang digunakan dalam penelitian, dimulai dari pengidentifikasian masalah, pengumpulan data, analisa algoritma dan alat ukur yang digunakan dalam penelitian.

BAB IV HASIL DAN PEMBAHASAN

Bab ini menyajikan temuan utama penelitian dan analisis yang telah dilakukan. Ini merupakan gambaran menyeluruh tentang pencapaian penelitian tersebut.

BAB V PENUTUP

Bab ini menjelaskan temuan penelitian yang berkontribusi pada pemahaman di bidang ini. Meskipun ada batasan, hasil penelitian memiliki implikasi praktis yang dapat diterapkan dalam konteks tertentu dan memperbaiki metode penelitian.

BAB II

TINJAUAN PUSTAKA

A. Landasan Teori

1. Klasifikasi

Klasifikasi adalah proses menemukan kumpulan pola atau fungsi-fungsi yang mendeskripsikan dan memisahkan kelas data satu dengan lainnya, untuk dapat digunakan dalam memprediksi data yang belum memiliki kelas data tertentu (Nasrullah, 2021).

Dalam konteks pendidikan, klasifikasi dapat digunakan untuk mengelompokkan mahasiswa menjadi beberapa kategori berdasarkan perilaku, prestasi akademik dan faktor-faktor lain yang mempengaruhi kemungkinan mereka putus studi (Rahmadeyan and Mustakim 2023).

2. Mahasiswa Putus Studi

Mahasiswa merupakan salah satu aspek penting dalam menentukan keberhasilan penyelenggaraan program studi. Mahasiswa yang selesai tepat waktu merupakan salah satu elemen penilaian akreditasi Universitas. Tetapi tidak semua mahasiswa bisa selesai tepat waktu sesuai jangka waktu yang telah ditentukan, tidak sedikit mahasiswa yang menempuh studi S1 lebih dari batas waktu maksimal yang telah ditentukan atau bahkan sampai terancam putus studi (Paskalis et al., 2019).

Mahasiswa yang tidak dapat menyelesaikan program studi S1 dalam waktu yang ditentukan dapat menghadapi beberapa konsekuensi. Salah satu konsekuensi yang paling umum adalah dinyatakan sebagai putus studi. Putus studi dapat terjadi jika mahasiswa tidak dapat memenuhi standar akademik yang ditetapkan oleh perguruan tinggi atau jika mereka tidak dapat menyelesaikan program dalam waktu yang telah ditentukan, maka mereka dapat dinyatakan tidak lulus dan tidak dapat melanjutkan pendidikan di perguruan tinggi yang bersangkutan.

Didalam Peraturan Akademik Universitas Muhammadiyah Makassar Tahun 2021 Bagian Keenam (Penilaian Akhir Tahun dan Putus Studi Pasal 61) Mahasiswa dinyatakan putus studi jika yang bersangkutan: (1) Mahasiswa program Sarjana (S-1) dinyatakan putus studi apabila apabila pada akhir tahun kedua (semester 4) tidak dapat memperoleh minimal 52 sks, dengan $IPK \leq 2.0$. (2) mahasiswa program S1 dinyatakan putus studi apabila pada masa studi akhir tahun semester 14 memperoleh < 144 sks dengan $IPK < 2,76$ (3) tidak melakukan registrasi/ber KRS 3 (tiga) semester secara berturut-turut.

3. Indeks Prestasi Kumulatif (IPK)

Pendidikan adalah prioritas semua orang saat ini. Pendidikan merupakan salah satu bentuk komunikasi interpersonal yang digunakan dalam bentuk pembelajaran. Di tingkat perkuliahan, dinamakan Nilai Rata-Rata Semester (IPK). IPK adalah penilaian yang diberikan kepada mahasiswa atas hasil kerja kerasnya selama menempuh pendidikan di perguruan tinggi (Hakam et al., 2015).

IPK adalah nilai rata-rata yang diambil dari keseluruhan mata kuliah yang pernah diikuti oleh mahasiswa, setiap perguruan tinggi memiliki penerapan yang berbeda dalam menentukan rentang nilai. IPK adalah akumulasi akhir dari total seluruh nilai yang diperoleh mahasiswa selama belajar di bangku perkuliahan. IPK dihitung berdasarkan nilai mahasiswa pada setiap mata kuliah. Setiap mata kuliah diberikan sejumlah sks (sistem satuan kredit), yang menunjukkan berapa banyak jam kuliah yang harus diikuti mahasiswa selama satu semester. Nilai akhir yang diperoleh mahasiswa pada setiap mata kuliah diubah menjadi bobot numerik sesuai dengan standar yang ditetapkan oleh universitas atau perguruan tinggi.

4. Indeks Prestasi Semester (IPS)

Indeks Prestasi Semester (IPS) adalah hasil penilaian capaian pembelajaran lulusan di setiap semester yang dinyatakan dengan indeks prestasi. IPS yang rendah dapat membatasi jumlah SKS yang dapat diambil oleh mahasiswa di semester berikutnya sehingga mereka tidak dapat mengambil mata kuliah yang lebih banyak (Untung Susilo & Mohammad Arifin, 2020).

5. Sistem Kredit Semester (SKS)

sistem Kredit Semester (SKS) adalah suatu sistem pendidikan yang dirancang untuk mengevaluasi kemampuan belajar mahasiswa. Program SKS ini berfungsi sebagai solusi bagi permasalahan mahasiswa dalam belajar, dengan harapan agar mereka dapat memperoleh pengetahuan secara optimal. SKS memungkinkan mahasiswa untuk memilih mata kuliah dan beban studi yang sesuai dengan kebutuhan mereka, sehingga mereka dapat mengatur waktu belajar dan beban studi yang harus diambil dengan lebih efektif. Dengan demikian, SKS membantu mahasiswa dalam mencapai tujuan pendidikan mereka dengan lebih baik (Mudawarna & Rokhmatin, 2022).

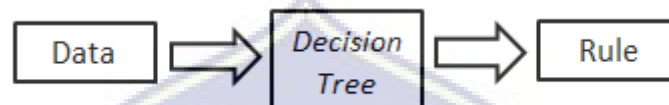
6. Decision Tree

Algoritma *Decision Tree* merupakan algoritma yang umum digunakan untuk pengambilan keputusan. *Decision tree* akan mencari solusi permasalahan dengan menjadikan kriteria sebagai *node* yang saling berhubungan membentuk seperti struktur pohon. *Decision tree* adalah model prediksi terhadap suatu keputusan menggunakan struktur hirarki atau pohon. Setiap pohon memiliki cabang, cabang mewakili suatu atribut yang harus dipenuhi untuk menuju cabang yang selanjutnya hingga berakhir di daun (tidak ada cabang lagi) (Sartika & Sensuse, 2017).

Decision tree merupakan salah satu metode klasifikasi yang kuat dan terkenal. Metode *decision tree* mengubah fakta yang besar menjadi pohon keputusan yang mempresentasikan aturan, aturan tersebut dapat dengan mudah untuk diinterpretasi oleh manusia. *Decision tree* juga berguna untuk mengeksplorasi data, menemukan hubungan tersembunyi antara sejumlah variabel input dengan sebuah variabel target. Kelebihan *decision tree* yaitu daerah pengambilan keputusan yang sebelumnya kompleks dan sangat global, dapat diubah menjadi lebih simpel dan spesifik, eliminasi perhitungan-perhitungan yang tidak diperlukan, karena ketika menggunakan metode *decision tree* maka sampel diuji hanya berdasarkan kriteria atau kelas tertentu, fleksibel untuk memilih fitur dari

node internal yang berbeda, fitur yang terpilih akan membedakan suatu kriteria dibandingkan kriteria yang lain dalam node yang sama (Sang et al., 2021).

Decision tree juga berguna dalam mengeksplorasi data, menemukan hubungan tersembunyi antara sejumlah calon variabel input dengan sebuah variabel target. Konsep dari *decision tree* ialah mengubah data menjadi pohon keputusan (*decision tree*) dan aturan-aturan keputusan (*rule*) (Salmawati, Yuyun, 2021).



Gambar 1. Konsep Decision Tree

Ada beberapa proses dalam *decision tree* diantaranya:

- 1) Mengubah bentuk data (tabel) menjadi model *Tree*.
- 2) Mengubah model *tree* menjadi *rule*.
- 3) Menyederhanakan dan menguji *rule*.
 - Membuat tabel distribusi terpadu dengan menyatakan semua nilai kejadian pada setiap *rule*.
 - Menghitung tingkat independensi antara kriteria pada suatu *rule*, yaitu antara atribut dan target atribut.
 - Mengeliminasi kriteria yang tidak perlu, yaitu yang tingkat inderpendensinya tinggi.

Konsep *decision tree* atau pohon keputusan adalah mengubah data menjadi aturan-aturan keputusan. Manfaat utama penggunaan *decision tree* adalah kemampuannya untuk mem-*break down* proses pengambilan keputusan yang kompleks menjadi lebih *simple* sehingga pengambilan keputusan akan lebih memudahkan solusi dari sebuah permasalahan (Sinaga et al., 2021). *Decision tree* adalah salah satu metode klasifikasi yang paling populer karena mudah diinterpretasikan oleh manusia. *Decision tree* digunakan untuk pengenalan pola dan termasuk dalam pengenalan pola secara *statistic*. *Decision tree* menggunakan

2 perhitungan yang pertama adalah perhitungan *Gain* pada persamaan dan perhitungan *Entropy* pada persamaan (Samasil et al., 2022).

Salmawati, Yuyun, (2021) didalam penelitiannya menyebutkan bahwa banyak algoritma yang dapat digunakan dalam pemebntukan *decision tree* atau pohon keputusan diantaranya yaitu: ID3, CART dan C4.5. algoritma C4.5 merupakan algoritma *Decision tree* yang terkenal dan merupakan pengembangan dari algoritma ID3 oleh seorang peneliti di bidang *machine learning* yaitu J. Ross Quinlan. Algoritma C4.5 memiliki beberapa kelebihan diantaranya mudah dimengerti, fleksibel dan menarik karena dapat divisualisasikan dalam bentuk gambar (pohon keputusan). Ada beberapa tahap dalam membuat sebuah pohon keputusan dengan algoritma C4.5 yaitu:

- 1) Menyiapkan *data training*. *Data training* biasanya dari data histori yang pernah terjadi sebelumnya dan sudah dikelompokkan kedalam kelas-kelas tertentu.
- 2) Menentukan akar pohon. Akar pohon akan diambil dari atribut yang terpilih dengan cara menghitung nilai *Gain* dari masing-masing atribut, nilai *Gain* paling tinggi yang akan menjadi akar pertama. Sebelum menghitung nilai *Gain* dari atribut, hitung dulu nilai *Entropy* yaitu:

$$\text{Entropy (S)} = \sum_{i=1}^n - p_i * \log_2 p_i \dots\dots\dots(1)$$

Keterangan:

S : himpunan kasus

A : fitur

n : jumlah partisi S

pi : Proporsi dari Si terhadap S

\sum : sigma kapital

- 3) Kemudian hitung nilai *Gain* dengan metode *information gain*:

$$\text{Gain (S, A)} = \text{Entropy (S)} - \sum_{i=1}^n * \text{Entropy}(S_i) \dots\dots\dots(2)$$

Keterangan:

S : himpunan

A : atribut

n : jumlah partisi atribut A

$|S_i|$: jumlah kasus pada partikel ke-i

$|S|$: jumlah kasus dalam S

Σ : sigma kapital

4) Ulangi langkah ke-2 hingga atribut semua tupel (*record*) terpartisi

5) Proses partisi pohon keputusan akan berhenti saat:

- Semua tupel dalam *node* N mendapat kelas yang sama
- Tidak ada atribut didalam tupel yang terpartisi lagi
- Tidak ada tupel di dalam cabang yang kosong

Dalam membangun pohon keputusan memiliki beberapa tahapan, yaitu pemilihan atribut sebagai akar; membuat cabang untuk tiap-tiap nilai; membagi kasus dalam cabang; dan mengulangi proses untuk setiap cabang sampai semua kasus pada cabang memiliki kelas yang sama. Algoritma mempunyai input berupa *training samples* dan *samples*. *Training samples* adalah berupa data contoh yang digunakan untuk membangun sebuah *tree* yang telah diuji kebenarannya. Sedangkan *samples* merupakan *field-field* data yang nantinya akan digunakan sebagai parameter dalam melakukan klasifikasi data (Gaol, 2020).

7. Cross Validation

Cross validation adalah sebuah tindakan pembuktian dari sebuah metode atau performa suatu *algoritma*. *Cross validation* merupakan pembuktian dengan membagi data menjadi *data training* dan *data testing* dengan komposisi tertentu. Pembagian paling sering digunakan dalam penelitian klasifikasi data mining ialah dengan membagi data secara acak menjadi 10 bagian. Satu bagian sebagai *data testing* dan 9 bagian sebagai *data training*. Validasi ini disebut juga dengan *k-fold cross validation* dengan $k=10$.

K-fold cross validation adalah salah satu metode yang digunakan untuk mengetahui rata-rata keberhasilan suatu sistem dengan cara melakukan perulangan dengan mengacak atribut masukan sehingga sistem tersebut teruji untuk beberapa atribut input yang acak. Setiap kelas pada kelompok data harus diwakili dalam proporsi yang tepat antara *data training* dan *data testing*. Data dibagi secara acak pada masing-masing kelas dengan perbandingan yang sama. Untuk mengurangi bias yang disebabkan oleh sampel tertentu, seluruh proses *training* dan *testing* diulangi beberapa kali dengan sampel yang berbeda. Tingkat kesalahan pada iterasi yang berbeda akan dihitung rata-ratanya untuk menghasilkan *error rate* secara keseluruhan. Model yang memberikan kesalahan terkecil adalah model yang terbaik (Salmawati, Yuyun, 2021)

8. Confusion matrix

Confusion matrix adalah sebuah hasil evaluasi dari sebuah klasifikasi *data mining* yang diwujudkan dalam sebuah tabel. *Confusion matrix* berisi tentang perhitungan jumlah objek *data testing* yang diprediksikan ke dalam sebuah kelas dengan klasifikasi yang sebenarnya. *Confusion matrix* menghasilkan pengukuran kinerja klasifikasi berupa nilai *accuracy* dan juga menghasilkan nilai *Precision* dan *recall*. Nilai *precision* dan *recall* yang dihasilkan ialah antara 0-1. Semakin tinggi nilai yang didapat, maka semakin baik (Salmawati, Yuyun, 2021).

- a. *Accuracy* adalah presentase ketepatan *record* data yang diidentifikasi benar.
- b. *Recall* adalah proporsi kasus positif yang diidentifikasi dengan benar.
- c. *Precision* adalah proporsi kasus yang diidentifikasi positif dengan benar.

Confusion matrix menggunakan tabel matriks yang mana jika *dataset* hanya terdiri dari dua kelas maka kelas tersebut dianggap sebagai positif negatif.

Tabel 1. Confusion matrix

<i>Correct Classification</i>	<i>Classified as</i>	
	+	-
+	<i>True Positives</i>	<i>False Negative</i>
-	<i>False Positive</i>	<i>True Negative</i>

Perhitungan *performance* dalam *confusion matrix* menggunakan persamaan sebagai berikut:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \dots\dots\dots(1)$$

$$Precision = \frac{TP}{FP+TP} \times 100\% \dots\dots\dots(2)$$

$$Recall = \frac{TP}{FN+TP} \times 100\% \dots\dots\dots(3)$$

Keterangan:

TP = *True Positive* (data positif yang terklasifikasi benar positif)

TN = *True Negative* (data negatif yang terklasifikasi benar negatif)

FP = *False Positive* (data positif yang terklasifikasi negatif)

FN = *False Negative* (data negatif yang terklasifikasi positif)

Selain menggunakan *accuracy*, *precision* dan *recall*, kinerja dari algoritma juga dapat diukur menggunakan *F-Measure (F1-Score)* merupakan *harmonic mean* dari *precision* dan *recall*, yang mana memiliki persamaan sebagai berikut(Salmawati, Yuyun, 2021):

$$F1 - Score = 2 \times \frac{Precision \times recall}{Precision + Recall} \dots\dots\dots(4)$$

B. Penelitian Terkait

Peneliti memberikan banyak inspirasi dan referensi untuk penyusunan proposal skripsi ini dari penelitian sebelumnya. Penelitian yang terkait sebelumnya meliputi:

Tabel 2. Penelitian terkait

Peneliti	Tujuan/Kasus	Metode/ Algoritma	Hasil
(Arfyanti et al., 2022)	Penerapan algoritma decision tree untuk penentuan pola penerima beasiswa KIP kuliah	<i>Algoritma Decision Tree</i>	Pengembangan sistem penentuan penerima beasiswa KIP kuliah yang efektif menggunakan algoritma decision tree, memprediksi kemungkinan penerimaan beasiswa dengan akurasi 85%
(Samasil et al., 2022)	Membandingkan kinerja dari dua algoritma klasifikasi, yaitu <i>Algoritma Naive Bayes</i> dan <i>Algoritma Decision Tree</i> untuk menentukan algoritma yang paling efektif	<i>Algoritma Naive Bayes</i> Dan <i>Decision Tree</i>	Hasil penelitian menunjukkan bahwa <i>algoritma decision tree</i> memiliki kinerja yang lebih baik dalam mengklasifikasi mahasiswa berpotensi drop out dibandingkan <i>algoritma naive bayes</i>
(Salmawati, Yuyun, 2021)	Membandingkan kinerja dari dua algoritma	<i>Algoritma Decision Tree C4.5</i> Dan	<i>Algoritma Decision Tree</i> mencapai tingkat akurasi 96,74% dan <i>f1-score</i> 86,55%

	klasifikasi, yaitu <i>Algoritma Decision Tree C4.5</i> dan <i>Algoritma Naive Bayes</i>	<i>Naive Bayes</i>	dalam mengklasifikasi mahasiswa berpotensi drop out di universitas Jambi, lebih efektif dibandingkan dengan <i>Algoritma Naive Bayes</i> yang mencapai 96,24% dan 82,34
(Hermanto, 2020)	Prediksi kelulusan dan putus studi mahasiswa dengan pendekatan bertingkat pada perguruan tinggi.	<i>Algoritma naive bayes</i> dan <i>decision tree</i>	<i>Algoritma</i> yang terbaik untuk memprediksi kelulusan dan putus studi mahasiswa di perguruan tinggi adalah <i>decision tree</i> dengan nilai akurasi terbaik 99.15 dengan data training sebanyak 30% (390 record)
(Paskalis et al., 2019)	Membangun sebuah model prediksi awal masa studi mahasiswa menggunakan <i>algoritma decision tree C4.5</i>	<i>Algoritma Decision Tree C4.5</i>	Model prediksi awal masa studi mahasiswa yang akurat dengan akurasi 92% menggunakan <i>algoritma decision tree C4.5</i>

Penelitian pertama oleh (Arfyanti et al., 2022) dengan judul “Penerapan *Algoritma Decision Tree* Untuk Penentuan Pola Penerima Beasiswa KIP Kuliah” Tujuan dari penelitian ini yaitu untuk menerapkan *algoritma decision tree* untuk menentukan pola penerima beasiswa KIP kuliah. Hasil penelitian ini memberitahukan bahwa *Algoritma Decision Tree* dapat digunakan untuk

menghasilkan pohon keputusan dari *rule* dalam mengetahui pola pemahaman mahasiswa pada pembelajaran daring.

Penelitian selanjutnya oleh (Samasil et al., 2022) dengan judul “Klasifikasi Mahasiswa Berpotensi *Drop Out* Menggunakan *Algoritma Naive Bayes* Dan *Decision Tree*” Tujuan dari penelitian ini yaitu untuk membandingkan kinerja dari *Algoritma Naive Bayes* Dan *Decision Tree* dalam mengklasifikasi mahasiswa berpotensi *drop out*. Hasil penelitian menunjukkan bahwa *algoritma decision tree* memiliki performa yang lebih tinggi dan efektif dengan nilai akurasi 99,13% dan *precision* dan *recall* sebesar 84,61% dan 100%. *Algoritma decision tree* juga menghasilkan 11 *rule* dengan atribut IPK sebagai akarnya, yang berarti IPK merupakan faktor paling berpengaruh dalam klasifikasi mahasiswa *drop out*.

Penelitian selanjutnya oleh (Arfyanti et al., 2022) dengan judul “Penerapan *Algoritma Decision Tree* Untuk Penentuan Pola Penerima Beasiswa KIP Kuliah” Tujuan dari penelitian ini yaitu untuk menerapkan *algoritma decision tree* untuk menentukan pola penerima beasiswa KIP kuliah. Hasil penelitian ini memberitahukan bahwa *Algoritma Decision Tree* dapat digunakan untuk menghasilkan pohon keputusan dari *rule* dalam mengetahui pola pemahaman mahasiswa pada pembelajaran daring.

Penelitian selanjutnya oleh (Salmawati, Yuyun 2021) dengan judul “Klasifikasi Mahasiswa Berpotensi *Drop Out* Menggunakan *Algoritma Decision Tree C4.5* Dan *Naive Bayes* Di Universitas Jambi” Tujuan dari penelitian ini yaitu untuk membandingkan kinerja dari algoritma *Decision Tree C4.5* dan *Naive bayes* sehingga didapat algoritma paling baik dalam mengklasifikasi mahasiswa berpotensi *drop out*. Penelitian ini memakai metode deskriptif, yaitu penelitian kualitatif menggunakan mendeskripsi insiden-insiden aktual yang terjadi secara keseluruhan. Teknik pengumpulan data yang dipakai merupakan teknik *observasi*, wawancara dan dokumentasi. Hasil penelitian ini memberitahukan bahwa *algoritma Decision Tree C4.5* menghasilkan nilai *accuracy* dan *f1-score* sebesar 96,74% dan 86,55% sedangkan *algoritma Naive Bayes* menghasilkan nilai

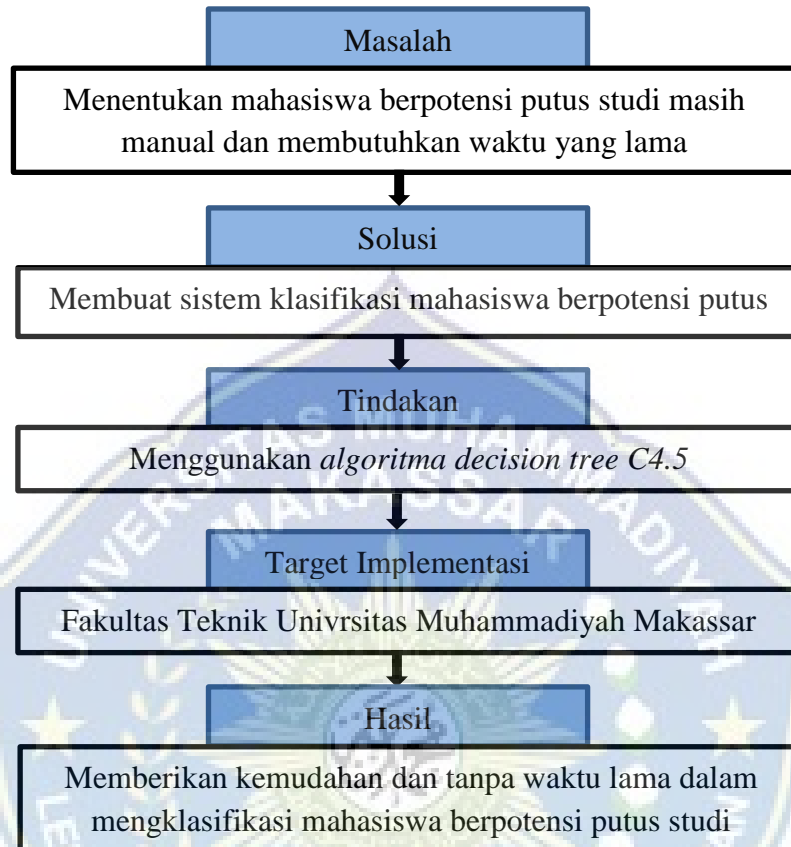
accuracy dan *f1-score* sebesar 96,24% dan 82,34%. Dari hasil penelitian tersebut dapat disimpulkan bahwa *algoritma* yang memiliki performa paling baik dalam mengklasifikasi mahasiswa *drop out* ialah *algoritma Decision Tree C4.5*. hasil penelitian ini diimplementasi menggunakan pohon keputusan dengan SKS Total sebagai root yang berarti SKS Total merupakan faktor paling berpengaruh dalam klasifikasi mahasiswa *drop out*. Rekomendasi untuk penelitian selanjutnya dapat membandingkan lebih dari dua *algoritma* atau menambah variabel lain dalam mengklasifikasi mahasiswa berpotensi *drop out*.

Penelitian selanjutnya oleh (Hermanto, 2020) dengan judul “Prediksi Kelulusan Dan Putus Studi Mahasiswa Dengan Pendekatan Bertingkat Pada Perguruan Tinggi” peneliti menggunakan *algoritma naive bayes* dan *decision tree* untuk memprediksi masa studi mahasiswa. Hasil penelitian menunjukkan bahwa model prediksi yang dibangun menggunakan algoritma naive bayes dan decision tree memiliki akurasi yang tinggi, dengan nilai *precision*, *recall* dan *accuracy* diatas 50% dan *error rate* dibawah 20%.

Penelitian selanjutnya oleh (Paskalis et al., 2019) dengan judul “Model Prediksi Awal Masa Studi Mahasiswa Menggunakan *Algoritma Decision Tree C4.5*” Tujuan dari penelitian ini yaitu untuk membangun sebuah model untuk prediksi awal masa studi mahasiswa pada program studi informatika universitas tanjung pura. Hasil penelitian ini memberitahukan bahwa model prediksi yang dibangun menggunakan *decision tree C4.5* baik dalam melakukan prediksi disebabkan karena hasil perhitungan rata-rata nilai *precision*, *recall* dan *accuracy* diatas 50% dan rata-rata nilai *error rate* berada dibawah 20%. Dari hasil tersebut dapat disimpulkan bahwa *algoritma decision tree C4.5* memiliki performa yang lebih tinggi dan efektif dalam mengklasifikasi masa studi mahasiswa.

C. Kerangka Berpikir

Berikut kerangka pikir dalam pelaksanaan penelitian:



Gambar 2. Diagram Kerangka Pikir

BAB III

METODE PENELITIAN

A. Tempat dan Waktu Penelitian

Penelitian dilakukan di Universitas Muhammadiyah Makassar tepatnya di Fakultas Teknik Universitas Muhammadiyah Makassar yang beralamat di Jl. Sultan Alauddin No 259.

Adapun pelaksanaan penelitian ini dilaksanakan dengan waktu 3 bulan dimulai pada bulan Juni tahun 2024 sampai dengan bulan Agustus tahun 2024

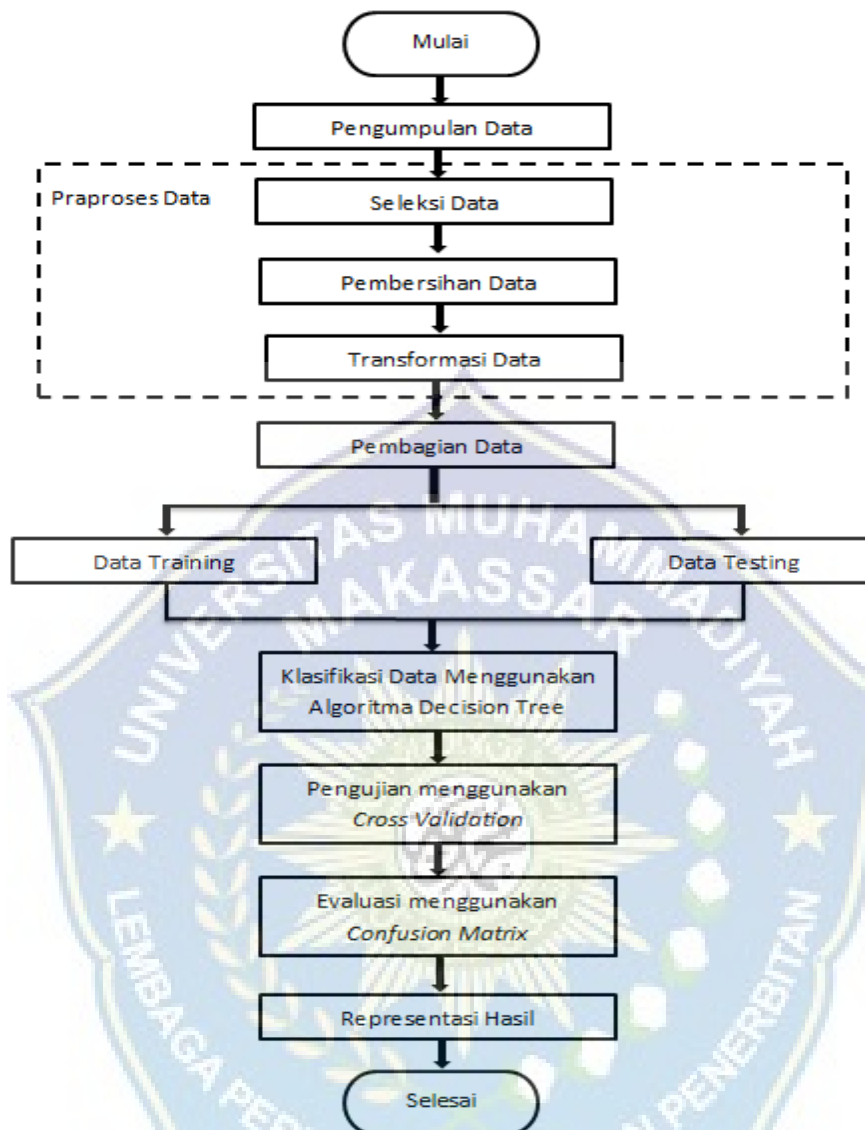
B. Alat dan Bahan

Adapun alat dan bahan yang akan digunakan dalam penelitian ini, yaitu:

1. Kebutuhan Hardware (Perangkat Keras)
 - a. Prosesor Intel(R) Core(TM) i3-1005G1
 - b. Besar Memory RAM 4GB
 - c. Kapasitas SSD 238,46 GB
2. Kebutuhan Software (Perangkat Lunak)
 - a. *Visual Studio Code*
 - b. *Microsoft Word Windows 10*
 - c. *Python sebagai bahasa programming*

C. Perancangan Sistem

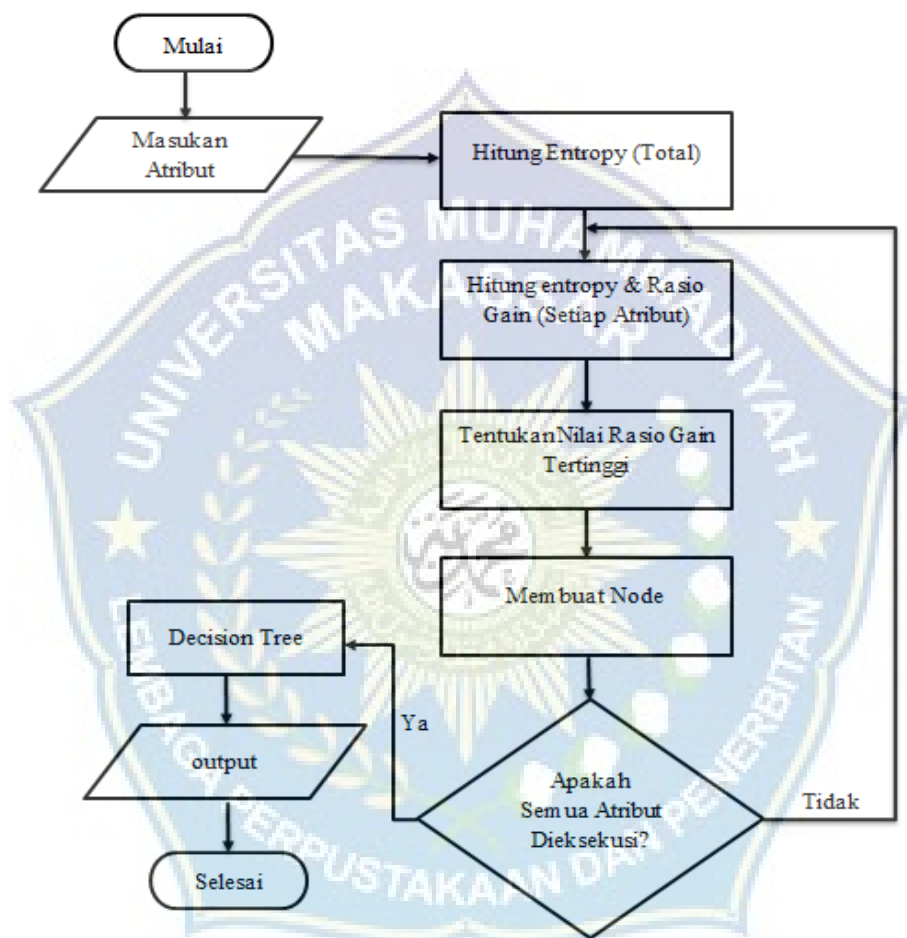
Perancangan sistem sangat penting dalam pembangunan suatu sistem karena menguraikan bagaimana suatu sistem dibangun dari tahap perencanaan hingga tahap pembuatan fungsi- fungsi yang diperlukan untuk pengoperasian sistem. Perancangan sistem juga merupakan langkah kerja dari pembuatan benda uji. Tujuan dari perancangan sistem adalah untuk menentukan apakah sistem yang akan dikembangkan akan menghasilkan hasil yang diinginkan.



Gambar 3. Kerangka Penelitian

Penelitian ini dimulai dari tahap pengumpulan data dari sumber-sumber relevan di lingkungan penelitian, dilanjutkan dengan praproses data yang mencakup seleksi, pembersihan dari nilai yang hilang dan transformasi data agar sesuai dengan kebutuhan *algoritma decision tree*, kemudian dataset dibagi menjadi data *training* (latih) dan data *testing* (uji), dimana data latih digunakan untuk melatih model *decision tree* dan data uji untuk menguji keakuratannya. Setelah itu, proses klasifikasi data dilakukan menggunakan model *decision tree*

yang dibangun dari data latih. Pengujian model dilakukan menggunakan metode *K-fold cross validation* untuk mengevaluasi kinerja dan keandalan model. Hasil evaluasi kinerja model akan dianalisis menggunakan matriks kebingungan (*confusion matrix*). Hasil penelitian akan disajikan melalui aplikasi web sederhana sebagai representasi temuan penelitian.



Gambar 4 Flowchart algoritma decision tree

Proses pembuatan *decision tree* dimulai dengan mengumpulkan dan memasukan data atribut yang relevan untuk klasifikasi, mengetahui entropi total dataset untuk mendapatkan gambaran umum ketidakpastian, menghitung entropi dan rasio gain setiap atribut untuk menentukan atribut paling informatif, memilih

atribut dengan rasio gain tertinggi sebagai node keputusan, membuat node berdasarkan atribut yang dipilih dan membagi dataset kedalam subset berdasarkan nilai atribut, mengevaluasi apakah semua atribut sudah digunakan atau pohon keputusan telah memenuhi kriteria tertentu, jika tidak, kembali ke langkah perhitungan entropi dan rasio gain untuk subset data baru, jika ya, maka pohon keputusan selesai dibangun dan siap digunakan untuk klasifikasi dataset baru berdasarkan atribut yang dimasukkan.

D. Teknik Pengujian Sistem

Teknik pengujian yang digunakan adalah *cross validation* dan *confusion matrix*. *Cross validation* adalah teknik yang umum digunakan untuk melalui kinerja model prediktif dengan membagi data menjadi beberapa subset atau “*fold*”. Teknik yang paling sering digunakan adalah *k-fold cross validation*, dimana dataset dibagi menjadi k subset atau *fold*. Misalnya, dalam *10-fold cross validation*, dataset dibagi menjadi 10 bagian yang hampir sama. Untuk setiap *fold*, model dilatih menggunakan 9 *fold* sebagai data pelatihan dan 1 *fold* sebagai data pengujian. Proses ini diulang sebanyak 10 kali dengan menggunakan *fold* yang berbeda sebagai data pengujian dan rata-rata metrik evaluasi dari eksperimen dihitung untuk mendapatkan estimasi kinerja model yang lebih akurat. Selain itu, *confusion matrix* digunakan untuk mengevaluasi kinerja model klasifikasi dengan menampilkan jumlah prediksi benar dan salah yang dibuat oleh model. Komponen *confusion matrix* meliputi *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), *False Negative* (FN), yang membantu dalam menghitung metrik evaluasi seperti akurasi, *precision*, *recall*, dan *f1-score*.

E. Teknik Analisis Data

Setelah semua data yang diperlukan untuk membahas topik penelitian terkumpul, maka analisis data merupakan salah satu proses penelitian yang dilakukan. Kegiatan analisis data merupakan kegiatan yang tidak dapat diabaikan dalam proses penelitian karena pentingnya ketajaman dan kecermatan dalam penerapan instrumen analisis dalam menentukan keakuratan hasil. Kesalahan dalam menentukan alat analisis dapat berakibat fatal pada kesimpulan yang

dicapai dan akan berdampak lebih besar pada pemanfaatan dan penerapan temuan penelitian.

Pendekatan analisis data secara garis besar diklasifikasikan menjadi dua kategori yaitu, analisis kuantitatif dan kualitatif. Satu-satunya perbedaan antara kedua teknik tersebut adalah jenis data. Analisis yang digunakan untuk data kualitatif adalah analisis kualitatif (tidak dapat diangkakan), sedangkan data yang dapat dikuantifikasi dapat dipelajari secara kuantitatif bahkan kualitatif.

Pada penelitian ini, teknik analisis data yang diterapkan adalah analisis data kuantitatif dengan analisis statistik deskriptif. Analisis statistik deskriptif adalah statistik yang digunakan untuk mengevaluasi data dengan meringkas atau menggambarkan data yang diperoleh apa adanya, tanpa membuat kesimpulan atau generalisasi yang terbuka untuk umum. Dalam arti tidak menggali atau menjelaskan keterkaitan, menguji hipotesis, membuat prediksi atau menarik kesimpulan, analisis ini hanya berupa akumulasi data dasar dalam berupa deskripsi.



BAB IV

HASIL DAN PEMBAHASAN

A. Pengumpulan Data

Data yang dikumpulkan merupakan data dari Fakultas Teknik Universitas Muhammadiyah Makassar dengan jumlah data sebesar 1493 data. Data yang diperoleh merupakan data mahasiswa program S1 angkatan 2013-2014. Data dalam penelitian ini merupakan data awal sebelum melakukan praproses data yang mana terdiri dari 18 atribut yaitu : nim, nama, jenis_kelamin, ayah_pekerjaan, ayah_penghasilan, ayah_status, ibu_penghasilan, ibu_pekerjaan, ibu_status, wali_pekerjaan, wali_penghasilan, lulus, semesterTerakhir, kategori_IPK, kategori_SKS, status_putus_studi.

B. Data Mentah

Data yang diolah dalam penelitian ini berasal data yang sudah dikumpulkan sebelumnya dari mahasiswa Fakultas Teknik Universitas Muhammadiyah Makassar, yang terdiri dari angkatan 2013-2015. Data set ini terdiri dari berbagai atribut seperti mahasiswa (nim, kodeProdi, angkatan, nama, jenisKelamin, semesterAwal, tahunAkademikLulus, lulus, khs, masaStudi), ayah (pekerjaan, penghasilan, status), ibu (pekerjaan, penghasilan, status) wali (pekerjaan, penghasilan) khs (tahunAkademik, ips, sksSmt, ipk, sksTotal, statusMahasiswa).

Pengumpulan data mentah dikumpulkan dari server GraphQL secara asinkron untuk beberapa mahasiswa berdasarkan nim mereka. Proses pengumpulan data ini dilakukan dalam beberapa langkah, yang memanfaatkan concurrency untuk mempercepat pengambilan data. Berikut penjelasan langkah-langkah pengumpulan data mentah:

1. Membuat sesi http asinkron

```
async with aiohttp.ClientSession() as session:
```

membuka sesi HTTP asinkron dengan menggunakan pustaka aiohttp. Sesi ini digunakan untuk mengirim beberapa permintaan HTTP secara bersamaan

tanpa menunggu setiap respons secara berurutan, memungkinkan pengambilan data yang lebih cepat.

2. Membuat daftar tugas asinkron

- `tasks = []` inisialisasi daftar kosong yang akan menampung semua tugas asinkron untuk pengambilan data.
- `for nim in nim_list:` digunakan untuk mengiterasi daftar NIM (`nim_list`). Untuk setiap NIM, dibuat tugas baru yang memanggil fungsi `query_graphql(session, nim)`, yang akan mengirim permintaan GraphQL untuk mendapatkan data mahasiswa dengan NIM tersebut.
- Tugas-tugas ini kemudian ditambahkan ke dalam daftar `tasks`.

3. Menjalankan semua tugas secara bersamaan

`responses = await asyncio.gather(*tasks)` menjalankan semua tugas asinkron dalam `tasks` secara bersamaan dan menunggu sampai semua tugas selesai. Hasil dari setiap tugas disimpan dalam daftar `responses`.

4. Memproses respons dan menyimpan data

- `for i, result in enumerate(responses):` mengiterasi daftar `responses`, di mana setiap respons berisi data mentah untuk satu mahasiswa.
- Untuk setiap respons, data mahasiswa diakses melalui `result.get('data', {}).get('mahasiswa', {})`. Jika data mahasiswa ada, informasi tersebut diekstraksi dan disimpan dalam dictionary `row`.
- Data Kartu Hasil Studi (KHS) juga diekstraksi jika tersedia. Setiap catatan KHS dimasukkan ke dalam `row` dengan nama kolom yang menunjukkan urutan semester, seperti `khs1_tahunAkademik`, `khs1_ips`, dll.

5. Menyimpan data ke data list

C. Data Preprocessing

Preprocessing data merupakan proses mempersiapkan data sebelum dilakukannya proses klasifikasi. *Preprocessing* data dalam penelitian ini terdiri dari *data selection*, *data cleaning*, dan *data transformation*.

1. Data Selection

Data selection atau seleksi data merupakan proses pemilihan data yang benar-benar diperlukan dan sesuai untuk proses klasifikasi, yang mana dalam proses ini akan dilakukan beberapa hal diantaranya:

a. Menambahkan Atribut IPK Terakhir, SKS Lulus dan Semester

Pada atribut ipk, sks dan semester data yang dimiliki terdiri dari banyak variasi atribut data sehingga perlu dikelompokkan menjadi satu atribut. Berikut tambahan atribut data yang dibuat untuk mengelompokkan data dari SKS, IPK dan Semester.

1) SKS Lulus

Penambahan atribut SKS Lulus diperoleh dari hasil identifikasi kolom-kolom yang berisi nilai total SKS, kemudian mengisi nilai-nilai kosong dengan nilai total SKS terakhir yang tersedia disetiap baris. Output dari proses ini yaitu nilai total SKS lulus yang terisi untuk setiap mahasiswa. Berikut kode yang dibuat pada *visual studio code*:

```
import pandas as pd
import os

# Baca data dari CSV
file_path = 'data_mahasiswa.csv' # Ganti
dengan path file Anda
df = pd.read_csv(file_path)

# Identifikasi kolom SKS Total per semester
sks_total_columns = [col for col in df.columns
if 'sksTotal' in col]

# Ambil nilai SKS terakhir yang tidak null
df['SKS_LULUS'] =
df[sks_total_columns].ffill(axis=1).iloc[:, -
1]
```

```

# Simpan hasil ke CSV baru
output_file_path = os.path.join(os.getcwd(),
'hasil_sks_lulus.csv')
df.to_csv(output_file_path, index=False)

# Tampilkan beberapa baris pertama untuk
verifikasi
print(df[['SKS_LULUS']].head())

```

2) IPK Terakhir

Penambahan atribut IPK Terakhir diperoleh dari hasil identifikasi kolom-kolom yang berisi nilai IPK, kemudian mengisi nilai-nilai kosong dengan nilai IPK Terakhir yang terisi untuk setiap mahasiswa. Berikut programnya:

```

import pandas as pd

# Baca data dari CSV
file_path = 'hasil_sks_lulus.csv' # Ganti
dengan path file Anda
df = pd.read_csv(file_path)

# Identifikasi kolom IPK per semester
ipk_columns = [col for col in df.columns if
'ipk' in col]

# Ambil nilai IPK terakhir yang tidak null
df['IPK_TERAKHIR'] =
df[ipk_columns].ffill(axis=1).iloc[:, -1]

# Simpan hasil ke CSV baru
output_file_path = os.path.join(os.getcwd(),
'hasil_ipk_lulus.csv')
df.to_csv(output_file_path, index=False)

# Tampilkan beberapa baris pertama untuk
verifikasi
print(df[['IPK_TERAKHIR']].head())

```

3) Semester

Penambahan atribut semester diperoleh dari hasil konversi masa studi mahasiswa yang diberikan dalam format “X Tahun, Y Bulan” menjadi jumlah semester dan menambahkannya sebagai kolom baru dalam

DataFrame. *Output* dari proses ini yaitu data mahasiswa dengan tambahan kolom jumlah semester berdasarkan masa studi. Berikut programnya:

```
import pandas as pd
import re

# Baca data dari CSV
df = pd.read_csv('hasil_ipk_lulus.csv')

# Fungsi untuk mengkonversi masa studi menjadi semester
def
convert_masa_studi_to_semester(masa_studi):
    if isinstance(masa_studi, str): #
Pastikan hanya string yang diproses
        match = re.match(r'(\d+) Tahun, (\d+)
Bulan', masa_studi)
        if match:
            years = int(match.group(1))
            months = int(match.group(2))
            total_months = (years * 12) +
months
            semesters = total_months // 6
            return semesters
        return None

# Terapkan fungsi ke kolom 'masaStudi' dan
simpan hasilnya ke kolom baru 'semester'
df['semester'] =
df['masaStudi'].apply(convert_masa_studi_to_se
mester)

# Simpan kembali data ke CSV
df.to_csv('data_mahasiswa_dengan_semester.csv'
, index=False)
print('Data berhasil disimpan ke
data_mahasiswa_dengan_semester.csv')
```

	SKS_LULUS	IPK_TERAKHIR	semesterTerakhir
0	60.0	2.90	8
1	156.0	3.08	12
2	22.0	2.91	3
3	30.0	2.80	3
4	156.0	3.18	10

Gambar 6. Hasil Penambahan Atribut

b. Menentukan Atribut Target

Dalam penelitian ini akan ditentukan atribut target yaitu status putus studi yang dikelompokkan kedalam dua *class* yaitu Potensial Putus Studi dan Tidak Potensial. Data yang digunakan dalam penelitian ini untuk menentukan atribut status putus studi ialah semua variable yang telah di seleksi yaitu ayah_pekerjaan, ayah_penghasilan, ayah_status, ibu_pekerjaan, ibu_penghasilan, ibu_status, wali_pekerjaan, wali_penghasilan, lulus, semesterTerakhir, kategori_IPK ,kategori_SKS dengan menggunakan kode program berikut:

```
# Definisikan fungsi untuk menentukan status potensi putus
studi
def potential_dropout_status(row):
# Define your criteria for potential dropout status
    if (row['ayah_pekerjaan'] == 2 and row['ayah_penghasilan']
in [0, 3] and row['ayah_status'] == 1) or \
        (row['ibu_pekerjaan'] == 1 and row['ibu_penghasilan']
in [0, 3] and row['ibu_status'] == 1) or \
        (row['wali_pekerjaan'] in [3, 4] and
row['wali_penghasilan'] in [0, 3]) or \
        (row['lulus'] == 0 and row['semesterTerakhir'] > 8 and
row['kategori_IPK'] <= 1) or \
        (row['kategori_SKS'] == 1):
        return 'Potensial Putus Studi'
    else:
        return 'Tidak Potensial'

# Terapkan fungsi untuk menambahkan kolom status potensi
putus studi
df['status_potensi_putus_studi'] =
df.apply(potential_dropout_status, axis=1)

# Simpan hasil ke CSV baru dengan kolom status potensi
putus studi
output_file_path =
'hasil_dengan_status_potensi_putus_studi.csv'
df.to_csv(output_file_path, index=False)

# Tampilkan beberapa baris pertama untuk verifikasi
print(df.head())
```

id	semester	asal	ayah_pekerjaan	ayah_penghasilan	ayah_status	ibu_pekerjaan	ibu_penghasilan	ibu_status	sekolah_pekerjaan	sekolah_penghasilan	Lulus	semester	terakhir	kategori_IPK	kategori_SKS	status_gubernil	pekerjaan_studi
2013	20130	1	0	0	5	0	0	0	0	0	0	0	1	1	1	Potensial	Potus Studi
2013	20130	1	0	0	5	0	0	0	0	0	0	1	12	2	0	Tidak Potensial	
2013	20130	1	0	0	5	0	0	0	0	0	0	3	2	1	1	Potensial	Potus Studi
2013	20130	1	0	0	5	0	0	0	0	0	0	3	3	1	1	Potensial	Potus Studi
2013	20130	0	0	0	0	0	0	0	0	0	1	18	2	0	0	Tidak Potensial	

Gambar 7. Hasil Codingan Atribut Target

c. Mengkategorikan Atribut IPK Terakhir dan SKS Lulus

Pada atribut ipk terakhir dan sks lulus data yang dimiliki terdiri dari banyak variasi data sehingga perlu dikelompokkan kedalam beberapa kategori. Berikut kategori data pada atribut ipk terakhir dan sks lulus:

1) IPK Terakhir

Pada atribut ipk terakhir, data akan dibagi menjadi 5 (lima) kategori diantaranya sebagai berikut:

Tabel 3. Kategori pada atribut ipk terakhir

IPK	Kategori
0.00 – 1.99	Tidak Memuaskan
2.00 – 2.75	Cukup Memuaskan
2.76 – 3.24	Memuaskan
3.25 – 3.79	Sangat Memuaskan
3.80 – 4.00	Dengan Pujian

2) SKS Lulus

Data sks lulus yang digunakan dalam penelitian ini merupakan jumlah sks pada semester genap tahun 2022. Untuk atribut sks lulus akan dibagi menjadi 3 (dua) kategori diantaranya:

Tabel 4. Kategori pada atribut sks lulus

Semester	SKS Lulus	Kategori
4 (empat)	0-40	Tidak Memenuhi
	>40	Memenuhi
8 (delapan)	0-85	Tidak Memenuhi

	>85	Memenuhi
14 (empat belas)	<144	Tidak Memenuhi
	>=144	Memenuhi

Berikut hasil data setelah dilakukan pengkategorian terhadap atribut ipk terakhir dan sks lulus:

	IPK_TERAKHIR	kategori_IPK	SKS_LULUS	semesterTerakhir	kategori_SKS
0	2.90	Memuaskan	60.0	8	Tidak Memenuhi
1	3.08	Memuaskan	156.0	12	Memenuhi
2	2.91	Memuaskan	22.0	3	Tidak Memenuhi
3	2.80	Memuaskan	30.0	3	Tidak Memenuhi
4	3.18	Memuaskan	156.0	10	Memenuhi

Gambar 8. Codingan dan Hasil Kategori

d. Seleksi Atribut

Atribut yang digunakan untuk proses klasifikasi hanya terdiri dari beberapa atribut yaitu yaitu angkatan, semesterAwal, nim, nama, jenisKelamin, ayah_pekerjaan, ayah_penghasilan, ayah_status, ibu_pekerjaan, ibu_penghasilan, ibu_status, wali_pekerjaan, wali_penghasilan, lulus, semesterTerakhir, kategori_IPK, kategori_SKS dan status_putus_studi. Berikut kode program yang digunakan dalam proses seleksi:

```
import pandas as pd

# Baca kembali data dari CSV yang telah disimpan
df = pd.read_csv('hasil_dengan_kategori.csv')

# Seleksi hanya kolom-kolom yang diinginkan
selected_columns = [ 'angkatan', 'semesterAwal',
                    'nim', 'nama', 'jenisKelamin', 'ayah_pekerjaan',
                    'ayah_penghasilan', 'ayah_status',
                    'ibu_pekerjaan',
                    'ibu_penghasilan', 'ibu_status', 'wali_pekerjaan',
                    'wali_penghasilan', 'lulus', 'semesterTerakhir',
                    'kategori_IPK', 'kategori_SKS',
                    'status_putus_studi']
df_selected = df[selected_columns]
```

```

# Menampilkan beberapa baris pertama dengan format
teratur
print("Data yang akan di seleksi:")
print(df_selected.head().to_string(index=False))

# Simpan kembali data yang telah dipilih ke dalam
file CSV jika diperlukan
df_selected.to_csv('data_mahasiswa_selected.csv',
index=False)
print('Data berhasil disimpan ke
data_mahasiswa_selected.csv')

```

```

Data setelah seleksi:
angkatan semesterAwal nim nama jenisKelamin ayah_pekerjaan ayah_penghasilan ayah_status ibu_peker
0 2013 20131 10501182313 MUHAMMAD FARYD HIKSYAL L NaN NaN NaN NaN
1 2013 20131 10501182413 MUHAMMAD SAKIR L TANI 500000 HIDUP NaN NaN
2 2013 20131 10501182513 FAISAL AKBAR L NaN NaN NaN NaN
3 2013 20131 10501182613 MUNIR MUHAIDIR L NaN NaN NaN NaN
4 2013 20131 10501182713 ST FATIMA P MIRASKASTA 500001-1000000 HIDUP MIRASI
Data berhasil disimpan ke data_mahasiswa_selected.csv

```

Gambar 9. Hasil Seleksi Atribut

e. Menghapus Atribut Nim, Nama, Jenis Kelamin

Pada penelitian ini, atribut nim, nama dan jenis kelamin tidak digunakan dalam proses klasifikasi. Atribut nim, nama, jenis kelamin hanya digunakan untuk menjelaskan identitas mahasiswa sehingga atribut tersebut akan dihapus. Atribut nim, nama dan jenis kelamin akan dihapus menggunakan kode program dan hasilnya sebagai berikut:

```

import pandas as pd

# Baca data dari CSV
file_path = 'data_dibersihkan.csv' # Ganti dengan
path file Anda
df = pd.read_csv(file_path)

# Menghapus kolom 'nim' dan 'nama'
df.drop(['nim', 'nama', 'jenisKelamin'], axis=1,
inplace=True)

# Simpan hasil ke CSV baru tanpa kolom 'nim' dan
'nama'
output_file_path = 'hasil_tanpa_nim_nama_jk.csv'
df.to_csv(output_file_path, index=False)

```

```
# Tampilkan beberapa baris pertama untuk verifikasi
print(df.head())
```

	angkatan	semesterAwal	ayah_pekerjaan	ayah_penghasilan	ayah_status	ibu_pekerjaan	ibu_penghasilan	ibu_status	wali_pekerjaan
0	2013	20131	TANI	500000	HIDUP	-	500000	HIDUP	WIRASWASTA
1	2013	20131	TANI	500000	HIDUP	-	500000	HIDUP	WIRASWASTA
2	2013	20131	TANI	500000	HIDUP	-	500000	HIDUP	WIRASWASTA
3	2013	20131	TANI	500000	HIDUP	-	500000	HIDUP	WIRASWASTA
4	2013	20131	WIRASWASTA	500001-1000000	HIDUP	WIRASWASTA	500001-1000000	HIDUP	WIRASWASTA

Gambar 10. Hasil Penghapusan Atribut

2. Data Cleaning

Data cleaning atau pembersihan data pada penelitian ini dilakukan dengan menghapus data-data missing value. Data missing value merupakan data yang bernilai kosong (null) yang tidak dapat digunakan dalam penelitian, sehingga didalam penelitian saya, saya mengisi data yang missing value dengan nilai mean atau rata-rata. Jumlah data missing value pada tiap atribut dapat dilihat pada tabel berikut:

Tabel 5. Jumlah Data Missing Value

Atribut	Jumlah Data <i>Missing Value</i>
angkatan	0
semesterAwal	0
ayah_pekerjaan	410
ayah_penghasilan	421
ayah_status	416
ibu_pekerjaan	411
ibu_penghasilan	428
ibu_status	416
wali_pekerjaan	1266
wali_penghasilan	1298
lulus	0
semesterTerakhir	0
kategori_IPK	4
kategori_SKS	0

Jumlah *missing value* diatas merupakan jumlah data kosong pada setiap kolom atribut, dalam satu baris data terdapat lebih dari satu atribut yang memiliki data *missing value*. Dari 14 atribut yang digunakan, 9 diantaranya memiliki data

missing value yaitu atribut ayah_pekerjaan memiliki jumlah data *missing value* sebesar 410 data, atribut ayah_penghasilan sebesar 421 data, atribut ayah_status sebesar 416 data, atribut ibu_pekerjaan sebesar 411 data, atribut ibu_penghasilan sebesar 428 data, atribut ibu_status sebesar 416 data, atribut wali_pekerjaan sebesar 1266 data, atribut wali_penghasilan sebesar 1298 data, atribut kategori_IPK sebesar 4 data. Berikut contoh data *missing value*:

ayah_status	ibu_pekerjaan	ibu_penghasilan	ibu_status	wali_pekerjaan	wali_penghasilan	lulus	semesterTerakhir
HIDUP	WIRASWASTA	3000001-5000000	HIDUP	NaN	500000	False	6
HIDUP	PNS	1500001-2000000	HIDUP	NaN	500000	True	9
HIDUP	PENSUIN	1000001-1500000	HIDUP	SWASTA	2000001-3000000	False	15
MENINGGAL	PNS	3000001-5000000	HIDUP	NaN	500000	True	13
MENINGGAL	-	500000	HIDUP	NaN	500000	True	14

Gambar 11. Data missing value

Tabel 6. data missing value setelah diubah

Atribut	Jumlah Data Missing Value
angkatan	0
semesterAwal	0
ayah_pekerjaan	0
ayah_penghasilan	0
ayah_status	0
ibu_pekerjaan	0
ibu_penghasilan	0
ibu_status	0
wali_pekerjaan	0
wali_penghasilan	0
lulus	0
semesterTerakhir	0
kategori_IPK	0
kategori_SKS	0

Dari gambar diatas dapat dilihat bahwa tidak ada lagi data *missing value* atau data yang kosong. Jumlah data pada proses pembersihan data dapat dilihat pada tabel berikut:

Tabel 7. Jumlah Data Hasil Penghapusan

	Jumlah Data
Data awal	1493
Data <i>missing</i>	5070
Data bersih	1493

Jumlah data bersih yang diperoleh setelah mengatasi data *missing value* berjumlah 1493 data. Setelah semua data bersih, selanjutnya melakukan transformasi data agar sesuai untuk proses data mining.

3. Data Transformation

Data dalam penelitian ini terdiri dari data kategorikal dan numeral sehingga perlu diubah kedalam bentuk data yang sama. Data akan diubah menggunakan LabelEncoder. Data yang diubah yaitu ayah_pekerjaan, ayah_penghasilan, ayah_status, ibu_pekerjaan, ibu_penghasilan, ibu_status, wali_pekerjaan, wali_penghasilan, lulus, kategori_IPK, kategori_SKS dan status_putus_studi. Berikut program untuk memberikan label data:

```
import pandas as pd

# Baca data dari CSV
file_path = 'hasil_tanpa_nim_nama_jk.csv'
df = pd.read_csv(file_path)

# Definisikan kamus pelabelan untuk kategori IPK, SKS, dan status
label_dict = {
    'ayah_status': {'HIDUP': 0, 'MENINGGAL': 1},
    'lulus': {False: 0, True: 1},
    'kategori_IPK': {
        'Tidak Memuaskan': 0, 'Cukup Memuaskan': 1,
        'Memuaskan': 2,
        'Sangat Memuaskan': 3, 'Dengan Pujian': 4
    },
    'kategori_SKS': {'Memenuhi': 0, 'Tidak Memenuhi': 1},
    'ibu_status': {'HIDUP': 0, 'MENINGGAL': 1}
}
```

```

# Definisikan fungsi untuk mengelompokkan pekerjaan wali
menjadi 5 kategori
def categorize_wali_job(job):
    if job in ['SWASTA', 'WIRASWASTA', 'MERINTIS USAHA',
'PEGAWAI BUMN', 'Berdagang']:
        return 'Wiraswasta/Usaha'
    elif job in ['PNS', 'Dosen', 'ASN PPP3K', 'POLDA
SULSEL', 'POLRI']:
        return 'PNS/BUMN'
    elif job in ['PENSUIN', 'HONORER', 'HONORER DI SEKOLAH
DASAR']:
        return 'Pensiunan/Honor'
    elif job in ['TANI', 'Nelayan', 'Buruh Bangunan',
'TUKANG BANGUNAN', 'Bengkel las']:
        return 'Petani/Buruh'
    elif job in ['MAHASISWA', 'MAHASISWA S2', 'Asisten
apoteker', 'Tenaga kontrak', 'Kontraktor', 'Grab car']:
        return 'Mahasiswa/Profesional'
    else:
        return 'Lainnya'

# Terapkan kategori baru pada kolom pekerjaan wali
df['wali_pekerjaan'] =
df['wali_pekerjaan'].map(categorize_wali_job)

# Buat kamus pelabelan untuk kategori pekerjaan wali
wali_job_label_dict = {
    'Wiraswasta/Usaha': 0,
    'PNS/BUMN': 1,
    'Pensiunan/Honor': 2,
    'Petani/Buruh': 3,
    'Mahasiswa/Profesional': 4,
    'Lainnya': 5
}

# Terapkan pelabelan pada kolom pekerjaan wali
df['wali_pekerjaan'] =
df['wali_pekerjaan'].map(wali_job_label_dict)

# Definisikan fungsi untuk mengelompokkan penghasilan
menjadi 3 kategori
def categorize_income(income):
    if income in ['500000', '500001-1000000']:
        return 'Rendah'
    elif income in ['1000001-1500000', '1500001-2000000']:
        return 'Sedang'
    elif income in ['> 5000000', '2000001-3000000',
'3000001-5000000']:
        return 'Tinggi'

```

```

else:
    return 'Tidak Diketahui'

# Terapkan kategori baru pada kolom penghasilan
df['wali_penghasilan'] =
df['wali_penghasilan'].map(categorize_income)
df['ibu_penghasilan'] =
df['ibu_penghasilan'].map(categorize_income)
df['ayah_penghasilan'] =
df['ayah_penghasilan'].map(categorize_income)

# Buat kamus pelabelan untuk kategori penghasilan
income_label_dict = {
    'Rendah': 0,
    'Sedang': 1,
    'Tinggi': 2,
    'Tidak Diketahui': 3
}

# Terapkan pelabelan pada kolom penghasilan
df['wali_penghasilan'] =
df['wali_penghasilan'].map(income_label_dict)
df['ibu_penghasilan'] =
df['ibu_penghasilan'].map(income_label_dict)
df['ayah_penghasilan'] =
df['ayah_penghasilan'].map(income_label_dict)

# Definisikan fungsi untuk mengelompokkan pekerjaan ayah
dan ibu menjadi 5 kategori
def categorize_father_job(job):
    if job in ['WIRASWASTA', 'SWASTA', 'PNS']:
        return 'Wiraswasta/PNS'
    elif job in ['KEPOLISIAN RI (POLRI)', 'PENSUIN']:
        return 'Polisi/Pensiunan'
    elif job in ['TANI', 'NELAYAN']:
        return 'Petani/Nelayan'
    elif job in ['BURUH NELAYAN/PERIKANAN', 'SECURITY']:
        return 'Buruh/Security'
    elif job in ['CLEANING SERVICE', 'PEGAWAI BUMN',
'Pekerja Perusahaan']:
        return 'Pekerjaan Lain'
    else:
        return 'Lainnya'

def categorize_mother_job(job):
    if job in ['WIRASWASTA', 'PNS', 'PENSUIN']:
        return 'Wiraswasta/PNS/Pensiunan'
    elif job in ['IBU RUMAH TANGGA', 'MENGURUS RUMAH
TANGGA']:

```



```

        return 'Rumah Tangga'
    elif job in ['TANI', 'PEDAGANG']:
        return 'Petani/Pedagang'
    elif job in ['SWASTA', 'KARYAWAN BUMD']:
        return 'Swasta/Karyawan'
    elif job in ['HONORER', 'Guru']:
        return 'Honor/Guru'
    else:
        return 'Lainnya'

# Terapkan kategori baru pada kolom pekerjaan ayah dan ibu
df['ayah_pekerjaan'] =
df['ayah_pekerjaan'].map(categorize_father_job)
df['ibu_pekerjaan'] =
df['ibu_pekerjaan'].map(categorize_mother_job)

# Buat kamus pelabelan untuk kategori pekerjaan ayah dan
ibu
father_job_label_dict = {
    'Wiraswasta/PNS': 0,
    'Polisi/Pensiunan': 1,
    'Petani/Nelayan': 2,
    'Buruh/Security': 3,
    'Pekerjaan Lain': 4,
    'Lainnya': 5
}

mother_job_label_dict = {
    'Wiraswasta/PNS/Pensiunan': 0,
    'Rumah Tangga': 1,
    'Petani/Pedagang': 2,
    'Swasta/Karyawan': 3,
    'Honor/Guru': 4,
    'Lainnya': 5
}

# Terapkan pelabelan pada kolom pekerjaan ayah dan ibu
df['ayah_pekerjaan'] =
df['ayah_pekerjaan'].map(father_job_label_dict)
df['ibu_pekerjaan'] =
df['ibu_pekerjaan'].map(mother_job_label_dict)

# Terapkan pelabelan pada kolom yang sesuai
for column, mapping in label_dict.items():
    if column in df.columns:
        df[column] = df[column].map(mapping)

# Definisikan fungsi untuk menentukan status potensi putus
studi

```

```

def potential_dropout_status(row):
    # Define your criteria for potential dropout status
    if row['ayah_pekerjaan'] == 2 and
row['ayah_penghasilan'] in [0, 3] and row['ayah_status'] ==
1 or row['ibu_pekerjaan'] == 1 and row['ibu_penghasilan']
in [0, 3] and row['ibu_status'] == 1 or
    row['wali_pekerjaan'] in [3, 4] and
row['wali_penghasilan'] in [0, 3] or row['lulus'] == 0 and
row['semesterTerakhir'] > 8 and row['kategori_IPK'] <= 1 or
row['kategori_SKS'] == 1:
        return 'Potensial Putus Studi'
    else:
        return 'Tidak Potensial'

# Terapkan fungsi untuk menambahkan kolom status potensi
putus studi
df['status_potensi_putus_studi'] =
df.apply(potential_dropout_status, axis=1)

# Simpan hasil ke CSV baru dengan kolom status potensi
putus studi
output_file_path =
'hasil_dengan_status_potensi_putus_studi.csv'
df.to_csv(output_file_path, index=False)

# Tampilkan beberapa baris pertama untuk verifikasi
print(df.head())

# Definisikan kamus pelabelan untuk status potensi putus
studi
dropout_status_label_dict = {
    'Potensial Putus Studi': 0,
    'Tidak Potensial': 1
}

# Terapkan pelabelan pada kolom status potensi putus studi
df['status_potensi_putus_studi'] =
df['status_potensi_putus_studi'].map(dropout_status_label_d
ict)

# Simpan hasil ke CSV baru dengan kolom status potensi
putus studi
output_file_path =
'hasil_dengan_pelabelan_status_potensi_putus_studi.csv'
df.to_csv(output_file_path, index=False)

# Tampilkan beberapa baris pertama untuk verifikasi
print(df.head())

```

	angkatan	semester	ayah_pekerjaan	ayah_penghasilan	ayah_status	ibu_pekerjaan	ibu_penghasilan	ibu_status	wali_pekerjaan	wali_penghasilan	lulus	semesterakhir	kategori_IPK	kategori_SIS	status_potensi_petas_studi	
0	2013	20131	1	0	0	5	0	0	0	0	0	0	0	2	1	0
1	2013	20131	1	0	0	5	0	0	0	0	1	12	2	0	1	1
2	2013	20131	1	0	0	5	0	0	0	0	0	3	2	1	0	0
3	2013	20131	1	0	0	5	0	0	0	0	0	3	2	1	0	0
4	2013	20131	0	0	0	0	0	0	0	0	1	10	2	0	1	1

Gambar 12. Codingan dan Hasil Transformasi Data

Berikut rincian pelabelan data pada setiap atribut:

Tabel 8. Pelabelan Pada Atribut ayah_pekerjaan

ayah_pekerjaan	Label
Wiraswasta/PNS	0
Polisi/Pensiunan	1
Petani/Nelayan	2
Buruh/Security	3
Pekerjaan Lain	4
Lainnya	5

Tabel 9. Pelabelan Pada Atribut ayah_penghasilan

ayah_penghasilan	Label
500000, 500001-1000000	0
1000001-1500000, 1500001-2000000	1
> 5000000, 2000001-3000000, 3000001-5000000	2

Tabel 10. Pelabelan Pada Atribut ayah_status

ayah_status	Label
Hidup	0
Meninggal	1

Tabel 11. Pelabelan Pada Atribut ibu_pekerjaan

ibu_pekerjaan	Label
Wiraswasta/PNS/Pensiunan	0
Rumah Tangga	1
Petani/Pedagang	2
Swasta/Karyawan	3
Honor/Guru	4
Lainnya	5

Tabel 12. Pelabelan Pada Atribut ibu_penghasilan

ibu_penghasilan	Label
500000, 500001-1000000	0
1000001-1500000, 1500001-2000000	1
> 5000000, 2000001-3000000, 3000001-5000000	2

Tabel 13. Pelabelan Pada Atribut ibu_status

ibu_status	Label
Hidup	0
Meninggal	1

Tabel 14. Pelabelan Pada Atribut wali_pekerjaan

wali_pekerjaan	Label
Wiraswasta/Usaha	0
PNS/BUMN	1
Pensiunan/Honor	2
Petani/Buruh	3
Mahasiswa/Profesional	4
Lainnya	5

Tabel 15. Pelabelan Pada Atribut wali_penghasilan

wali_penghasilan	Label
500000, 500001-1000000	0
1000001-1500000, 1500001-2000000	1
> 5000000, 2000001-3000000, 3000001-5000000	2

Tabel 16. Pelabelan Pada Atribut lulus

lulus	Label
FALSE	0
TRUE	1

Tabel 17. Pelabelan Pada Atribut Kategori_IPK

kategori_IPK	Label
Tidak Memuaskan	0
Cukup Memuaskan	1
Memuaskan	2
Sangat Memuaskan	3
Dengan Pujian	4

Tabel 18. Pelabelan Pada Atribut kategori_SKS

kategori_SKS	Label
Memenuhi	0
Tidak Memenuhi	1

Tabel 19. Pelabelan Pada status_putus_studi

status_putus_studi	Label
Potensial Putus Studi	0
Tidak Potensial	1

D. Pembagian Data

Setelah dilakukan *Preprocessing* data, selanjutnya akan dilakukan pembagian data. Pada tahap ini, data akan dibagi menjadi dua yaitu data *training* dan data *testing*. Pembagian data *training* dan data *testing* ini berdasarkan atribut target yang telah memiliki *class* data. Data *training* merupakan data yang digunakan untuk melatih *algoritma*. Tujuannya agar *algoritma* dapat dipelajari pola dari data yang diberikan. Sedangkan data *testing* merupakan data yang digunakan untuk melihat performa dari *algoritma* yang telah dilatih. Dalam penelitian ini data akan dibagi dengan proporsi 80% data *training* dan 20% data *testing*. Berikut jumlah data setelah dilakukan pembagian:

Tabel 20. Pembagian Data

Klasifikasi	Jumlah Data	Data Training	Data Testing
		(80%)	(20%)
Potensial Putus Studi	771	612	159
Tidak Potensial	722	582	140
Total	1493	1194	299

Dari tabel diatas diketahui bahwa jumlah data *taining* yang dihasilkan adalah 329 data dan data *testing* berjumlah 83 data.

E. Klasifikasi Menggunakan Algoritma Decision Tree C4.5

Setelah data dibagi menjadi data *taining* dan data *testing*, tahap selanjutnya ialah proses klasifikasi data. Pada tahap klasifikasi data, akan dilakukan pembangunan model menggunakan *algoritma decision tree C4.5*. pada proses ini data *training* akan mempelajari pola dari algoritma sehingga didapat model yang dihasilkan. Setelah itu dapat dilihat dari performa dari algoritma yang telah dilatih berdasarkan akurasi yang dihasilkan dari data *testing* atau data uji. Berikut hasil akurasi dari *algoritma decision tree C4.5*:

Tabel 21. Hasil Akurasi

Algoritma	Akurasi(%)
<i>Decision Tree C4.5</i>	100%

F. Pengujian Menggunakan Cross Validation

Setelah didapat hasil akurasi dari proses klasifikasi, selanjutnya ialah melakukan pengujian menggunakan *cross validation*. Jenis pengujian *cross validation* yang digunakan ialah *k-fold cross validation* dimana $k=10$ yang berarti data akan dibagi menjadi 10 bagian dengan 9 bagian sebagai data *training* dan 1 bagian sebagai data *testing* (*10-fold cross validation*). *K-fold cross validation* adalah salah satu metode dengan melakukan perulangan dengan mengacak atribut masukkan sehingga sistem teruji untuk beberapa atribut input yang acak. Berikut

ini hasil akurasi dari *algoritma decision tree C4.5* menggunakan pengujian *10-fold cross validation*:

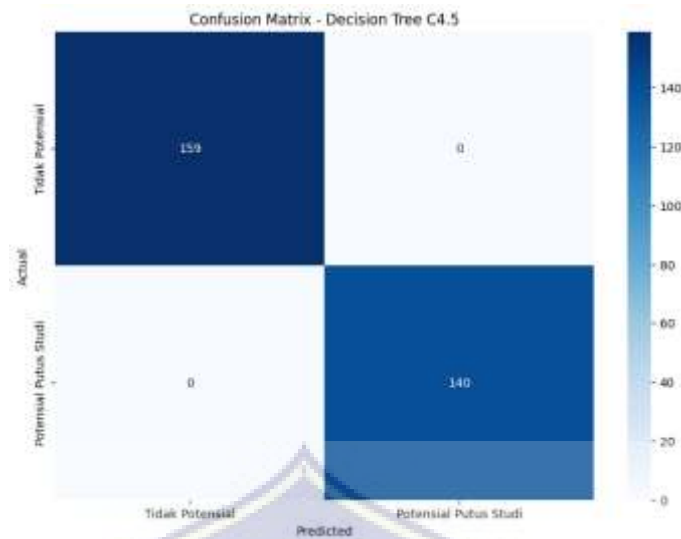
Tabel 22. Hasil Pengujian 10-fold cross validation

<i>k-Fold</i>	Akurasi(%)
1	99,33%
2	100%
3	100%
4	99,33%
5	100%
6	99,33%
7	99,33%
8	100%
9	99,33%
10	100%
Rata-Rata	99,66%

Berdasarkan hasil dari pengujian diatas dapat dilihat bahwa akurasi *algoritma decision tree C4.5* mengalami penurunan dari hasil akurasi sebelum dilakukan pengujian walau tidak terlalu signifikan. Rata-rata akurasi yang didapat ialah 99,66%

G. Evaluasi Hasil Klasifikasi

Evaluasi hasil klasifikasi pada *algoritma decision tree* juga menggunakan metode *confussion matrix*, hasil evaluasi dapat dilihat pada gambar berikut:



Gambar 13. *Confusion Matrix* Pada Algoritma *Decision Tree C4.5*

Dari gambar diatas, hasil evaluasi menggunakan *confusion matrix* menghasilkan jumlah data TP sebesar 159 data, jumlah FN sebesar 0 data, data FP berjumlah 0 data dan TN berjumlah 140 data. Lebih jelasnya dapat dilihat pada tabel berikut:

Tabel 23. *Confusion Matrix* Pada Algoritma *Decision Tree C4.5*

Data Sebenarnya	Klasifikasi	
	Tidak Potensial	Potensial Putus Studi
Tidak Potensial	159 (TP)	0 (FN)
Potensial Putus Studi	0 (FP)	140 (TN)

Dari hasil klasifikasi pada tabel *confusion matrix* di atas dapat dilihat bahwa,

- True Positive* (TP) menjelaskan dimana data terklasifikasi Tidak Potensial, memang benar Tidak Potensial. Dalam hal ini jumlah data yang didapat sebanyak 159 data.
- False Positive* (FP) menjelaskan bahwa data yang terklasifikasi Tidak Potensial ternyata Potensial Putus Studi. Jumlah data yang didapat sebesar 0 data.
- False Negative* (FN) menjelaskan bahwa data yang terklasifikasi Potensial Putus Studi, memang benar Potensial Putus Studi. Jumlah data yang didapat adalah 0 data.

- d. *True Negative* (TN) menjelaskan bahwa data yang terklasifikasi Potensial Putus Studi, memang benar Potensial Putus Studi. Jumlah data yang didapat sebanyak 140 data.

Dari hasil *confussion matrix* yang telah diperoleh, dihasilkan nilai akurasi, *precision*, *recall*, dan *f1-score* dengan hasil sebagai berikut:

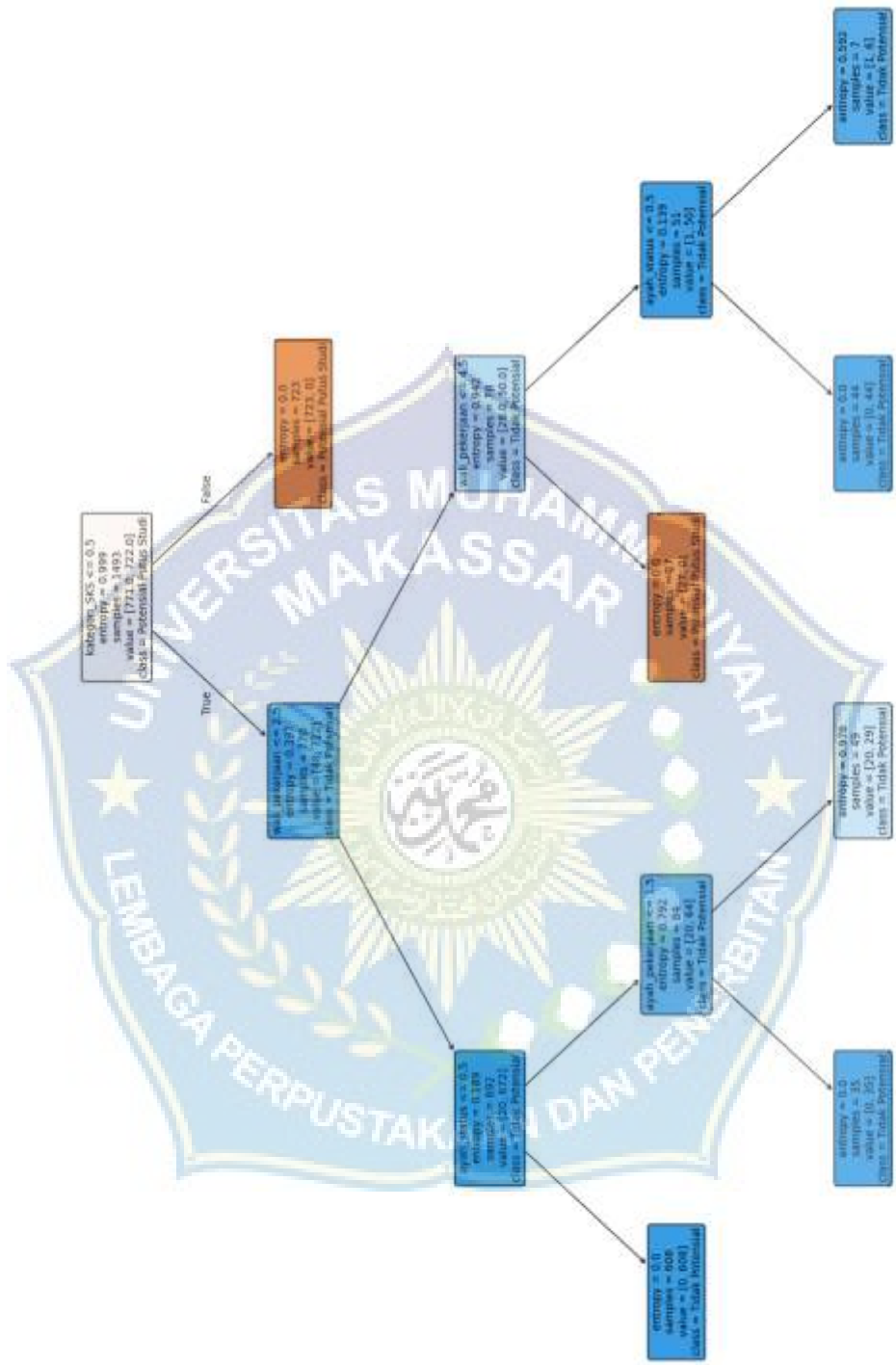
Classification Report:				
	precision	recall	f1-score	support
0	1.00	1.00	1.00	159
1	1.00	1.00	1.00	140
accuracy			1.00	299
macro avg	1.00	1.00	1.00	299
weighted avg	1.00	1.00	1.00	299

Gambar 14. Hasil Pengukuran Kinerja *Algoritma Decision Tree C4.5*

Berdasarkan gambar di atas hasil akurasi yang didapat sebesar 100% dengan nilai *precision* untuk data Potensial Putus Studi dan Tidak Potensial masing-masing sebesar 100% dan 100%. Nilai *recall* dari data Potensial Putus Studi dan Tidak Potensial yang dihasil yaitu 100% dan 100% . selanjutnya untuk nilai *f1-score* dari data Potensial Putus Studi dan Tidak Potensial yang didapat yaitu 100% dan 100%. Untuk nilai *precision*, *recall* dan *f1-score* secara keseluruhan dapat dilihat pada nilai rata-rata *macro* (*macro average*) dari *precision*, *recall* dan *f1-score*. Nilai rata-rata (*macro average*) dari *precision*, *recall* dan *f1-score* masing-masing sebesar 100%, 100% dan 100%.

Visualisasi *Decision Tree* (Pohon Keputusan)

dalam klasifikasi menggunakan algoritma *decision tree c4.5* dihasilkan pohon keputusan. Berikut pohon keputusan sederhana dari klasifikasi menggunakan algoritma *decision tree c4.5* :



gambar 15. Pohon Keputusan

Gambar diatas merupakan hasil pohon keputusan sederhana dalam penelitianku yang mana atribut Kategori_SKS adalah akar dari pohon keputusan tersebut. Itu artinya atribut kategori_SKS paling berpengaruh dalam klasifikasi mahasiswa berpotensi putus studi dilanjutkan dengan semessterTerakhir.

Berikut nilai gain informasi pada masing-masing atribut yang sudah diseleksi menghasilkan nilai gain sebagai berikut:

Tabel 24. Nilai Gain Informasi Pada Masing-Masing Atribut

Angkatan	0.0003
SemsesterAwal	0.0003
Ayah pekerjaan	0.00759
Ayah penghasilan	0.0601
Ayah status	0.0004
Ibu pekerjaan	0.0405
Ibu penghasilan	0.0205
Ibu status	0.0085
Wali pekerjaan	0.0377
Wali penghasilan	0.0097
Lulus	0.5370
SemesterTerakhir	0.6819
Kategori IPK	0.3625
Kategori SKS	0.8256

BAB V PENUTUP

A. Kesimpulan

1. Algoritma *Decision Tree C4.5* telah berhasil diterapkan untuk mengklasifikasikan mahasiswa yang berpotensi putus studi. Proses klasifikasi dilakukan dengan mengolah dataset yang terdiri dari atribut-atribut seperti pekerjaan dan penghasilan orang tua/wali, IPK, SKS, serta variabel lainnya yang relevan dengan status akademik mahasiswa. Penggunaan *Decision Tree C4.5* memungkinkan pembentukan pohon keputusan yang efektif dalam mengidentifikasi mahasiswa dengan risiko putus studi berdasarkan pola yang ada dalam data, serta menghasilkan aturan-aturan klasifikasi yang dapat dengan mudah diinterpretasikan.
2. Hasil penelitian menunjukkan bahwa *Algoritma Decision Tree C4.5* menghasilkan nilai akurasi 100%. Dengan nilai rata-rata *precision* dan *recall* yaitu 100% dan 100%. Nilai rata-rata *f1-score* sebesar 100%. Menunjukkan bahwa *algoritma* ini memiliki performa yang tinggi, sehingga menjadi salah satu metode yang efektif dalam Mengklasifikasi Mahasiswa Berpotensi Putus Studi Pada Fakultas Teknik Universitas Muhammadiyah Makassar.

B. Saran

Saran untuk penelitian selanjutnya yaitu:

1. Untuk proses klasifikasi dapat menggunakan algoritma yang lain seperti *Knearest Neighbour*, *Naive Bayes*, *Support Vector Machines(SVM)* atau menggunakan *Decision tree* yang lain contoh salah satunya adalah *CART (Classification and Regression Trees)*. Bisa juga dengan melakukan 2 perbandingan algoritma klasifikasi atau bahkan lebih.
2. Untuk pengukuran kinerja klasifikasi dapat ditambah dengan metode lain agar hasil lebih optimal.

DAFTAR PUSTAKA

- Arfyanti, I., Fahmi, M., & Adytia, P. (2022). Penerapan Algoritma Decision Tree Untuk Penentuan Pola Penerima Beasiswa KIP Kuliah. *Building of Informatics, Technology and Science (BITS)*, 4(3), 1196–1201. <https://doi.org/10.47065/bits.v4i3.2275>
- Gaol, N. Y. L. (2020). Prediksi Mahasiswa Berpotensi Non Aktif Menggunakan Data Mining dalam Decision Tree dan Algoritma C4.5. *Jurnal Informasi & Teknologi*, 2, 23–29. <https://doi.org/10.37034/jidt.v2i1.22>
- Hakam, M., Sudarno, & Hoyyi, A. (2015). ANALISIS JALUR TERHADAP FAKTOR-FAKTOR YANG MEMPENGARUHI INDEKS PRESTASI KUMULATIF (IPK) MAHASISWA STATISTIKA UNDIP. *Gaussian*, 4(3), 70. <https://doi.org/10.2307/2532808>
- Hermanto. (2020). Prediksi Kelulusan dan Putus Studi Mahasiswa dengan Pendekatan Bertingkat pada Perguruan Tinggi. *Jurnal Sistem Informasi & Manajemen Basis Data (SIMADA)*, 3(2), 140–148. <https://jurnal.darmajaya.ac.id/index.php/SIMADA/article/view/2359>
- Mudawarma, & Rokhmatin, I. (2022). Analisis perbedaan prestasi akademik PAI siswa ditinjau dari input jalur penerimaan peserta didik baru. *Etheses IAIN Kediri*, 1(1), 11–17. <https://etheses.iainkediri.ac.id/5684/>
- Nasrullah, A. H. (2021). Implementasi Algoritma Decision Tree Untuk Klasifikasi Produk Laris. *Jurnal Ilmiah Ilmu Komputer*, 7(2), 45–51. <https://doi.org/10.35329/jiik.v7i2.203>
- Paskalis, E., Orpa, K., & Ripanti, E. F. (2019). Model Prediksi Awal Masa Studi Mahasiswa. *JUSTIN (Jurnal Sistem Dan Teknologi Informasi)*, 7(4), 272–278. <https://jurnal.untan.ac.id/index.php/justin/article/view/33163>
- Quadri, M., & Kalyankar, D. (2010). Drop out feature of student data for academic performance using decision tree techniques. *Global Journal of Computer*, 10(2), 2–5. <http://computerresearch.org/stpr/index.php/gjst/article/viewArticle/128>
- Rahmadeyan, A., & Mustakim, M. (2023a). Seleksi Fitur pada Supervised Learning: Klasifikasi Prestasi Belajar Mahasiswa Saat dan Pasca Pandemi COVID-19. *Jurnal Nasional Teknologi Dan Sistem Informasi*, 9(1), 21–32. <https://doi.org/10.25077/teknosi.v9i1.2023.21-32>
- Rahmadeyan, A., & Mustakim, M. (2023b). Seleksi Fitur pada Supervised Learning: Klasifikasi Prestasi Belajar Mahasiswa Saat dan Pasca Pandemi COVID-19. *Jurnal Nasional Teknologi Dan Sistem Informasi*, 9(1), 21–32. <https://doi.org/10.25077/teknosi.v9i1.2023.21-32>
- Salmawati, Yuyun, H. (2021). Klasifikasi Mahasiswa Berpotensi Drop Out Menggunakan Algoritma Decision Tree C4 . 5 Dan Naive Bayes Di Universitas Jambi. *Jurnal Ilmiah Ilmu Komputer*, 8(2), 115032. <https://repository.unja.ac.id/25341/>

- Samasil, S., Yuyun, Y., & Hazriani, H. (2022). Klasifikasi Mahasiswa Berpotensi Drop Out Menggunakan Algoritma Naive Bayes Dan Decision Tree. *Jurnal Ilmiah Ilmu Komputer*, 8(2), 108–114. <https://doi.org/10.35329/jiik.v8i2.242>
- Sang, A. I., Sutoyo, E., & Darmawan, I. (2021). Analisis Data Mining Untuk Klasifikasi Data Kualitas Udara Dki Jakarta Menggunakan Algoritma Decision Tree Dan Support Vector Machine Data Mining Analysis for Classification of Air Quality Data Dki Jakarta Using Decision Tree Algorithm and Support Vector. *E-Proceeding of Engineering*, 8(5), 8954–8963. <https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/15900>
- Sartika, D., & Senses, D. I. (2017). Perbandingan Algoritma Klasifikasi Naive Bayes, Nearest Neighbour, dan Decision Tree pada Studi Kasus Pengambilan Keputusan Pemilihan Pola Pakaian. *Jatisi*, 1(2), 151–161. <https://jurnal.mdp.ac.id/index.php/jatisi/article/view/78>
- Sinaga, D., Solaiman, E. J., & Kaunang, F. J. (2021). Penerapan Algoritma Decision Tree C4.5 Untuk Klasifikasi Mahasiswa Berpotensi Drop out Di Universitas Advent Indonesia. *TeKa*, 11(2), 167–173. <https://doi.org/10.36342/teika.v11i2.2613>
- Umamah, K. N., Anggraini, M. P., Edyta, N., & Faradiba, A. T. (2018). PRESTASI AKADEMIK DITINJAU DARI KETERLIBATAN REMAJA DALAM KEGIATAN EKSTRAKURIKULER. *Jurnal Muara Ilmu Sosial, Humaniora, Dan Seni*, 2(1), 108. <https://doi.org/10.24912/jmishumsen.v2i1.1688>
- Untung Susilo, & Mohammad Arifin. (2020). Analisis Hubungan Indeks Prestasi Semester Dan Indeks Prestasi Kumulatif Dengan Prestasi Mahasiswa Fakultas Ekonomi Universitas Kediri. *Jurnal Riset Bisnis Dan Ekonomi*, 1(1). <http://ojs.unik-kediri.ac.id/index.php/jimek>

LAMPIRAN

Lampiran 1. Data Mentah

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	kodeProdi	angkatan	semester	nm	nama	jenisKelamin	tahunAkad	tanggalLulus	lulus	mesaStudi	ayah_ayah_per	ayah_status	ibu_penger	ibu_penger	ibu_status	wali_pekerwal_per	
2	22202	2013	20131	10581182	MUHAMMIL				False								
3	22202	2013	20131	10581182	MUHAMMIL		20182		True	5 Tahun, 1TANI	50000	HIDUP	-	50000	HIDUP		
4	22202	2013	20131	10581182	FAISAL AIL				False								
5	22202	2013	20131	10581182	MUNIR M L				False								
6	22202	2013	20131	10581182	ST FATIMP		20172		True	4 Tahun, 8WIRASW	500001-1	CHIDUP	WIRASW	500001-1	CHIDUP		
7	22202	2013	20131	10581182	SAIFUL R L		20191		True	6 Tahun, 2WIRASW	500001-1	CHIDUP	WIRASW	500000	HIDUP		
8	22202	2013	20131	10581182	JLPRIL				False								
9	22202	2013	20131	10581183	HENRA A L		20182		True	5 Tahun, 9WIRASW	200001-3	CHIDUP	IRT	50000	HIDUP		
10	22202	2013	20131	10581183	MOHAMMIL				False	TANI	50000	MENINGG	-	500001-1	CHIDUP	-	50000
11	22202	2013	20131	10581183	HILMAN AL				False	TANI	500001-1	CHIDUP	WIRASW	500001-1	CHIDUP		
12	22202	2013	20131	10581183	ILHAM L				False	TANI	500001-1	CHIDUP	URT	500001-1	CHIDUP		
13	22202	2013	20131	10581183	FIRDAUS L				False								
14	22202	2013	20131	10581183	SUPARDIL				False								
15	22202	2013	20131	10581183	NURHIDAP		20172		True	4 Tahun, 1TANI	200001-3	CHIDUP	-	50000	HIDUP		
16	22202	2013	20131	10581183	GAFFARUEL		20182		True	5 Tahun, 1TANI	50000	HIDUP	IBU RUM	500000	HIDUP	WIRASWASTA	
17	22202	2013	20131	10581183	ARTIA P		20172		True	4 Tahun, 8Nelayan	500001-1	CHIDUP	WIRASW	500000	HIDUP		
18	22202	2013	20131	10581183	ARIZAL A L				False								
19	22202	2013	20131	10581184	ASMAJUL P		20172		True	4 Tahun, 9PNS	300001-5	CHIDUP	IRT	50000	HIDUP	PNS	300000
20	22202	2013	20131	10581184	NINING H P		20172		True	4 Tahun, 1TANI	50000	HIDUP	TANI	50000	HIDUP		
21	22202	2013	20131	10581184	MLH ILHAL		20182		True	5 Tahun, 9TANI	800001-1	CHIDUP	-	500000	HIDUP		
22	22202	2013	20131	10581184	HERMAWL				False								
23	22202	2013	20131	10581184	USMANSIL		20182		True	5 Tahun, 9TANI	500001-1	CMENINGCIRT	-	500001-1	CHIDUP	TANI	150000
24	22202	2013	20131	10581184	SYAFRI Z L		20182		True	5 Tahun, 9WIRASW	1500001-2	CHIDUP	-	50000	HIDUP		
25	22202	2013	20131	10581184	USMANSIL				False								
5635	95202	2024	20241	10584112	Mih Rahe L				False	PNS	300001-5	CHIDUP	ASN PPP	2000001-3	CHIDUP		
5636	95202	2024	20241	10584112	Ayabillah e L				False								
5637	95202	2024	20241	10584112	ADRIAN EL				False								
5638	95202	2024	20241	10584112	SALWA ILP				False								
5639	95202	2024	20241	10584112	Nabila Am P				False								
5640	95201	2024	20241	10585110	MARWAHP				False								
5641	95201	2024	20241	10585110	Sabrio L				False	TANI	500001-1	CHIDUP	-	50000	HIDUP	SWASTA > 5000	
5642	95201	2024	20241	10585110	NUR ISDYP				False	TANI	500001-1	CHIDUP	-	50000	HIDUP	TANI	50000
5643	95201	2024	20241	10585110	ILHAM NLL				False	PNS	200001-3	CHIDUP	Ibu Rumah	500000	HIDUP	Ibu Rumah	100000
5644	95201	2024	20241	10585110	Ah Ubaidi L				False	PNS	3000001-5	CHIDUP	PNS	3000001-5	CHIDUP		
5645	95201	2024	20241	10585110	IRENE ALP				False	WIRASW	500001-1	CHIDUP	SWASTA	3000001-5	CHIDUP		
5646	95201	2024	20241	10585110	MUH HEFL				False	TANI	1000001-1	CHIDUP	IBU RUM	500000	HIDUP		
5647	95201	2024	20241	10585110	ARVIN L				False								
5648	95201	2024	20241	10585110	RITRI RAJ P				False	-	50000	MENINGG	WIRASW	500001-1	CHIDUP		
5649	95201	2024	20241	10585110	MUH SHAL				False	SWASTA	500000	MENINGG	Pegawai	1500001-2	CHIDUP	WIRASW	300000
5650	95201	2024	20241	10585110	YOHANIS L				False								
5651	95201	2024	20241	10585110	Ramdayar P				False	TANI	50000	HIDUP	-	50000	HIDUP		
5652	95201	2024	20241	10585110	Hadji maki L				False	SWASTA	500001-1	CHIDUP	-	50000	HIDUP		
5653	95201	2024	20241	10585110	citra cahye P				False	TANI	1500001-2	CHIDUP	Ibu rumah	500000	HIDUP	TANI	150000
5654	95201	2024	20241	10585110	INDRA KILL				False	TANI	1000001-1	CHIDUP	TANI	1000001-1	CHIDUP	SWASTA	150000
5655	95201	2024	20241	10585110	MUHAMMIL				False	PNS	3000001-5	CHIDUP	-	50000	HIDUP		
5656	95201	2024	20241	10585110	MUH. ILHAL				False								
5657	95201	2024	20241	10585110	APRIAND L				False								
5658	95201	2024	20241	10585110	Ardi Farel L				False								

Lampiran 2. Data Preprocessing

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	angkatan	semester	Prim	nama	jenisKelamin	ayah_peker	ayah_pens	ayah_stat	ibu_peker	ibu_pengh	ibu_status	wali_peker	wali_pengh	kelas	semesterT	kategori_II	kategori_SKS
2	2013	20131	10581182	MUHAMMIL										False	8	Memuaskan	Tidak Memenuhi
3	2013	20131	10581182	MUHAMMIL	TANI	500000	HIDUP	-	500000	HIDUP				True	12	Memuaskan	Memenuhi
4	2013	20131	10581182	FAISAL AIL										False	3	Memuaskan	Tidak Memenuhi
5	2013	20131	10581182	MUNIR M L										False	3	Memuaskan	Tidak Memenuhi
6	2013	20131	10581182	ST FATIMP	WRASW	500001	1CHIDUP		WRASW	500001	1CHIDUP			True	10	Memuaskan	Memenuhi
7	2013	20131	10581182	SAFUL R L	WRASW	500001	1CHIDUP		WRASW	500000	HIDUP			True	13	Sangat Mr	Memenuhi
8	2013	20131	10581182	JUPRI L										False	4	Tidak Men	Tidak Memenuhi
9	2013	20131	10581183	HENRA A L	WRASW	2000001	2HIDUP	IRT	500000	HIDUP				True	12	Memuaskan	Memenuhi
10	2013	20131	10581183	MOHAMMIL	TANI	300000	MENINGG	-	500001	1CHIDUP	-	500000		False	12	Cukup Mei	Tidak Memenuhi
11	2013	20131	10581183	HILMAN AL	TANI	500001	1CHIDUP		WRASW	500001	1CHIDUP			False	12	Cukup Mei	Tidak Memenuhi
12	2013	20131	10581183	LHAM L	TANI	500001	1CHIDUP		URT	500001	1CHIDUP			False	10	Cukup Mei	Tidak Memenuhi
13	2013	20131	10581183	FIRDAUS L										False	7	Cukup Mei	Tidak Memenuhi
14	2013	20131	10581183	SUPARDIL										False	5	Memuaskan	Tidak Memenuhi
15	2013	20131	10581183	NURHIDAP	TANI	2000001	2HIDUP		500000	HIDUP				True	10	Sangat Mr	Memenuhi
16	2013	20131	10581183	GAFARULL	TANI	500000	HIDUP	IBU RUM	500000	HIDUP	WRASWASTA			True	12	Memuaskan	Memenuhi
17	2013	20131	10581183	ARTIA P	Nelayan	500001	1CHIDUP		WRASW	500000	HIDUP			True	10	Sangat Mr	Memenuhi
18	2013	20131	10581183	ARIZAL A L										False	3	Memuaskan	Tidak Memenuhi
19	2013	20131	10581184	ASMAUL P	PNS	3000001	5HIDUP	IRT	500000	HIDUP	PNS	3000001	5	True	10	Sangat Mr	Memenuhi
20	2013	20131	10581184	NINING H P	TANI	500000	HIDUP	TANI	500000	HIDUP				True	10	Sangat Mr	Memenuhi
21	2013	20131	10581184	MUH ILHAL	TANI	500001	1CHIDUP		500000	HIDUP				True	12	Memuaskan	Memenuhi
22	2013	20131	10581184	HERMAWIL										False	3	Tidak Men	Tidak Memenuhi
23	2013	20131	10581184	USMANSIL	TANI	500001	1CMENINGG	IRT	500001	1CHIDUP	TANI	1500001	2	True	12	Memuaskan	Memenuhi
24	2013	20131	10581184	SYAFRI Z L	WRASW	1500001	2HIDUP		500000	HIDUP				True	12	Sangat Mr	Memenuhi
1471	2015	20151	10583001	TAKDIR RL	TANI	500001	1CMENINGG	IRT	500000	HIDUP				False	17	Memuaskan	Tidak Memenuhi
1472	2015	20151	10583001	RISNA P	TANI	500001	1CHIDUP	TANI	500001	1CHIDUP				True	12	Sangat Mr	Memenuhi
1473	2015	20151	10583001	A ARHAML										False	17	Tidak Men	Tidak Memenuhi
1474	2015	20151	10583001	HAMZAH L	TANI	1000001	1HIDUP	IBU RUM	500001	1CHIDUP				True	13	Sangat Mr	Memenuhi
1475	2015	20151	10583001	PRAWAN L	TANI	1000001	1HIDUP	URT	500000	HIDUP				False	17	Cukup Mei	Tidak Memenuhi
1476	2015	20151	10583001	A ARWIAL	TANI	500000	HIDUP	TANI	500000	HIDUP				True	14	Memuaskan	Memenuhi
1477	2015	20151	10583001	APRIANTIL	WRASW	500000	MENINGG	IRT	500000	MENINGG	WASTA	3000001	5	True	14	Memuaskan	Memenuhi
1478	2015	20151	10583001	SUPARDIL										False	17	Tidak Men	Tidak Memenuhi
1479	2015	20151	10583001	SUCI RAHP										False	17	Sangat Mr	Tidak Memenuhi
1480	2015	20151	10583001	CALMA WILP	PNS	3000001	5MENINGG	PNS	3000001	5HIDUP				True	14	Sangat Mr	Memenuhi
1481	2015	20151	10583001	MUFAKKIL										False	17	Tidak Men	Tidak Memenuhi
1482	2015	20151	10583001	YUSRIAD L	TANI	500001	1CHIDUP		500000	HIDUP				True	14	Memuaskan	Memenuhi
1483	2015	20151	10583001	RAJA FATIL	WRASW	1000001	1MENINGG	ibu rumah	1000001	1HIDUP				True	14	Memuaskan	Memenuhi
1484	2015	20151	K1058215	AKHMAD L	PNS	2000001	3HIDUP	-	500000	HIDUP				True	7	Sangat Mr	Memenuhi
1485	2015	20151	K1058216	DHINA FEP	PENSUN	2000001	3HIDUP	PNS	3000001	5HIDUP				True	5	Sangat Mr	Memenuhi
1486	2015	20151	K1058216	KHAIRUL L	PNS		HIDUP	PNS		HIDUP				False	17	Sangat Mr	Tidak Memenuhi
1487	2015	20151	K1058216	RAEHAN/P	WRASW	1500001	2HIDUP	PENSUN	2000001	3HIDUP				True	5	Sangat Mr	Memenuhi
1488	2015	20151	K1058216	SUKMAWP		500000	HIDUP	-	500000	HIDUP				True	5	Sangat Mr	Memenuhi
1489	2015	20151	K1058216	IN SULAFP										False	17	Cukup Mei	Tidak Memenuhi
1490	2015	20151	K1058216	IRWAN L	TANI	500001	1CHIDUP	-	500000	MENINGG	WRASW	> 500000		False	17	Sangat Mr	Tidak Memenuhi
1491	2015	20151	K1058216	FITRIANT P	WRASW	3000001	5MENINGG	-	500000	HIDUP	WRASWASTA			True	5	Sangat Mr	Memenuhi
1492	2015	20151	K1058216	HADRAY/P	PNS	3000001	5MENINGG	PNS	3000001	5MENINGG	WASTA	2000001	5	True	5	Memuaskan	Memenuhi
1493	2015	20151	K1058216	FITRI RAHP	WRASW	2000001	3HIDUP	PNS	3000001	5HIDUP				True	5	Sangat Mr	Memenuhi
1494	2015	20151	K1058217	ABDUL M L	PENSUN	1500001	2MENINGG	-	500000	MENINGG	PNS	3000001	5	True	5	Sangat Mr	Memenuhi

Lampiran 3. Transformasi Data

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	angkatan	semester	Fayah_peki	ayah_peki	ayah_stat	ibu_peker	ibu_pengh	ibu_status	wali_peker	wali_pengk	ulus	semester1	kategori_1	kategori_	Estatus_potensi	putus_studi	
2	2013	20131	2	0	0	5	0	0	0	0	0	8	2	1	0		
3	2013	20131	2	0	0	5	0	0	0	0	1	12	2	0	1		
4	2013	20131	2	0	0	5	0	0	0	0	0	5	2	1	0		
5	2013	20131	2	0	0	5	0	0	0	0	0	5	2	1	0		
6	2013	20131	0	0	0	0	0	0	0	0	1	10	2	0	1		
7	2013	20131	0	0	0	0	0	0	0	0	1	13	3	0	1		
8	2013	20131	2	0	0	5	0	0	0	0	0	4	0	1	0		
9	2013	20131	0	2	0	5	0	0	0	0	1	12	2	0	1		
10	2013	20131	2	0	1	5	0	0	5	0	0	12	1	1	0		
11	2013	20131	2	0	0	0	0	0	0	0	0	12	1	1	0		
12	2013	20131	2	0	0	5	0	0	0	0	0	10	1	1	0		
13	2013	20131	2	0	0	5	0	0	0	0	0	7	1	1	0		
14	2013	20131	0	0	0	5	0	0	0	0	0	5	2	1	0		
15	2013	20131	2	2	0	5	0	0	0	0	1	10	5	0	1		
16	2013	20131	2	0	0	1	0	0	0	0	1	12	2	0	1		
17	2013	20131	0	0	0	0	0	0	0	0	1	10	3	0	1		
18	2013	20131	2	0	0	5	0	0	0	0	0	9	2	1	0		
19	2013	20131	0	2	0	5	0	0	1	2	1	10	5	0	1		
20	2013	20131	2	0	0	2	0	0	0	0	1	10	3	0	1		
21	2013	20131	2	0	0	5	0	0	0	0	1	12	2	0	1		
22	2013	20131	2	0	0	5	0	0	0	0	0	3	0	1	0		
23	2013	20131	2	0	1	5	0	0	1	1	1	12	2	0	0		
24	2013	20131	0	1	0	5	0	0	0	0	1	12	3	0	1		
25	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
26	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
27	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
28	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
29	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
30	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
31	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
32	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
33	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
34	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
35	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
36	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
37	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
38	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
39	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
40	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
41	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
42	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
43	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
44	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
45	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
46	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
47	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
48	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
49	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
50	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
51	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
52	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
53	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
54	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
55	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
56	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
57	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
58	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
59	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
60	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
61	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
62	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
63	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
64	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
65	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
66	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
67	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
68	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
69	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
70	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
71	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
72	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
73	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
74	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
75	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
76	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
77	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
78	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
79	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
80	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
81	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
82	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
83	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
84	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
85	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
86	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
87	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
88	2013	20131	0	0	0	5	0	0	0	0	0	6	0	0	0		
89	2013	20131	0	0	0	5	0	0	0	0	0</						

Lampiran 4. Codingan Dan Output

%pip install aiohttp pandas

```
Requirement already satisfied: aiohttp in c:\users\asus\appdata\local\cache\pythonsoftwarefoundation.python.3.10\sh2\2fr8b@localcache\local-req
Requirement already satisfied: pandas in c:\users\asus\appdata\local\cache\pythonsoftwarefoundation.python.3.10\sh2\2fr8b@localcache\local-cache
Requirement already satisfied: aiomysql in c:\users\asus\appdata\local\cache\pythonsoftwarefoundation.python.3.10\sh2\2fr8b@localcache\local-cache
Requirement already satisfied: aiohttp=1.3.2 in c:\users\asus\appdata\local\cache\pythonsoftwarefoundation.python.3.10\sh2\2fr8b@localcache\local-cache
Requirement already satisfied: attrs>=17.3.0 in c:\users\asus\appdata\local\cache\pythonsoftwarefoundation.python.3.10\sh2\2fr8b@localcache\local-cache
Requirement already satisfied: frozenlist=1.1.1 in c:\users\asus\appdata\local\cache\pythonsoftwarefoundation.python.3.10\sh2\2fr8b@localcache\local-cache
Requirement already satisfied: multidict<7.0, >=4.5 in c:\users\asus\appdata\local\cache\pythonsoftwarefoundation.python.3.10\sh2\2fr8b@localcache\local-cache
Requirement already satisfied: yarl<2.0, >=1.0 in c:\users\asus\appdata\local\cache\pythonsoftwarefoundation.python.3.10\sh2\2fr8b@localcache\local-cache
Requirement already satisfied: async-timeout<5.0, >=4.0 in c:\users\asus\appdata\local\cache\pythonsoftwarefoundation.python.3.10\sh2\2fr8b@localcache\local-cache
Requirement already satisfied: numpy>=1.22.4 in c:\users\asus\appdata\local\cache\pythonsoftwarefoundation.python.3.10\sh2\2fr8b@localcache\local-cache
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\asus\appdata\local\cache\pythonsoftwarefoundation.python.3.10\sh2\2fr8b@localcache\local-cache
Requirement already satisfied: pytz>=2020.1 in c:\users\asus\appdata\local\cache\pythonsoftwarefoundation.python.3.10\sh2\2fr8b@localcache\local-cache
Requirement already satisfied: tzdata>=2022.7 in c:\users\asus\appdata\local\cache\pythonsoftwarefoundation.python.3.10\sh2\2fr8b@localcache\local-cache
Requirement already satisfied: six>=1.5 in c:\users\asus\appdata\local\cache\pythonsoftwarefoundation.python.3.10\sh2\2fr8b@localcache\local-cache
Requirement already satisfied: idna>=2.0 in c:\users\asus\appdata\local\cache\pythonsoftwarefoundation.python.3.10\sh2\2fr8b@localcache\local-cache
Note: you may need to restart the kernel to use updated packages.

[notice] A new release of pip is available: 24.1.2 -> 24.2
[notice] To update, run: c:\users\asus\appdata\local\cache\pythonsoftwarefoundation.python.3.10\sh2\2fr8b@python.exe -m pip install
- - -
```

```
import aiohttp
import asyncio
import pandas as pd

async def query_graphql(session, nim):
    url = 'https://sicekcock.if.unismuh.ac.id/graphql' # Ganti
    dengan URL endpoint GraphQL yang sesuai
    query = """
    query($nim: String!) {
      mahasiswa(nim: $nim) {
        nim
        kodeProdi
        angkatan
        nama
        jenisKelamin
        semesterAwal
        tahunAkademikLulus
        lulus
        masaStudi
        ayah{
          pekerjaan
          penghasilan
          status
        }
        ibu {
          pekerjaan
          penghasilan
          status
        }
        wali{
          pekerjaan
        }
      }
    }
    """
```

```

        penghasilan
    }
    khs {
        tahunAkademik
        ips
        sksSmt
        ipk
        sksTotal
        statusMahasiswa
    }
}
}
}
"""
variables = {'nim': str(nim)} # Pastikan NIM dikonversi ke
string
async with session.post(url, json={'query': query, 'variables':
variables}) as response:
    return await response.json()
nim_list = pd.read_csv('nim.csv')['nim'].tolist()
data_list = []
total = len(nim_list)
max_khs = 0
print(f"Total NIM: {total}")

```

0.0s Python
Total NIM: 5657

```

async def get_mahasiswa():
    async with aiohttp.ClientSession() as session:
        tasks = []
        for nim in nim_list:
            tasks.append(query_graphql(session, nim))
        responses = await asyncio.gather(*tasks)
        for i, result in enumerate(responses):
            nim = nim_list[i]
            # print(f"Result for NIM {nim}: {result.get('data',
            {}).get('mahasiswa', {}).get('nama')}")
            mahasiswa = result.get('data', {}).get('mahasiswa', {})
            if mahasiswa:
                row = {
                    'kodeProdi': mahasiswa.get('kodeProdi'),
                    'angkatan': mahasiswa.get('angkatan'),
                    'semesterAwal': mahasiswa.get('semesterAwal'),
                    'nim': mahasiswa.get('nim'),
                }

```

```

        'nama': mahasiswa.get('nama'),
        'jenisKelamin': mahasiswa.get('jenisKelamin'),
        'tahunAkademikLulus':
mahasiswa.get('tahunAkademikLulus'),
        'tanggalLulus': mahasiswa.get('tanggalLulus'),
        'lulus': mahasiswa.get('lulus'),
        'masaStudi': mahasiswa.get('masaStudi'),
        'ayah_pekerjaan': mahasiswa.get('ayah',
{ }).get('pekerjaan'),
        'ayah_penghasilan': mahasiswa.get('ayah',
{ }).get('penghasilan'),
        'ayah_status': mahasiswa.get('ayah',
{ }).get('status'),
        'ibu_pekerjaan': mahasiswa.get('ibu',
{ }).get('pekerjaan'),
        'ibu_penghasilan': mahasiswa.get('ibu',
{ }).get('penghasilan'),
        'ibu_status': mahasiswa.get('ibu',
{ }).get('status'),
        'wali_pekerjaan': mahasiswa.get('wali',
{ }).get('pekerjaan'),
        'wali_penghasilan': mahasiswa.get('wali',
{ }).get('penghasilan'),
    }
    khs = mahasiswa.get('khs', [])
    global max_khs
    max_khs = max(max_khs, len(khs))
    for idx, k in enumerate(khs, start=1):
        row.update({
            f'khs{idx}_tahunAkademik':
k.get('tahunAkademik'),
            f'khs{idx}_ips': k.get('ips'),
            f'khs{idx}_sksSmt': k.get('sksSmt'),
            f'khs{idx}_ipk': k.get('ipk'),
            f'khs{idx}_sksTotal': k.get('sksTotal'),
            f'khs{idx}_statusMahasiswa':
k.get('statusMahasiswa'),
        })
        data_list.append(row)
        # print(f'Downloading data {i + 1}/{total} berhasil
di download')
    await get_mahasiswa()
    print(f'{total} Data berhasil di download')

```

5657 Data berhasil di download

```
import os
import pandas as pd

# Membuat DataFrame dari data_list
df = pd.DataFrame(data_list)

# Mendefinisikan kolom untuk DataFrame
columns = [
    'kodeProdi', 'angkatan', 'semesterAwal', 'nim', 'nama',
    'jenisKelamin', 'tahunAkademikLulus', 'tanggalLulus', 'lulus',
    'masaStudi', 'ayah_pekerjaan', 'ayah_penghasilan',
    'ayah_status',
    'ibu_pekerjaan', 'ibu_penghasilan', 'ibu_status',
    'wali_pekerjaan', 'wali_penghasilan'
] + [
    f'khs{idx}_{field}' for idx in range(1, max_khs + 1) for field
in
    ['tahunAkademik', 'ips', 'sksSmt', 'ipk', 'sksTotal',
    'statusMahasiswa']
]

# Mengatur ulang kolom DataFrame sesuai urutan yang diinginkan
df = df.reindex(columns=columns)

# Memfilter DataFrame hanya untuk mahasiswa angkatan 2013 hingga
2015
df_angkatan_2013_2015 = df[df['angkatan'].between(2013, 2015)]

# Menyimpan hasil filter ke file CSV
df_angkatan_2013_2015.to_csv('data_mahasiswa_angkatan_2013_2015.csv'
, index=False)

# Menampilkan pesan sukses
print('Data berhasil disimpan ke
data_mahasiswa_angkatan_2013_2015.csv')
```


✓ 0.2s

Python

Data berhasil disimpan ke data_mahasiswa_angkatan_2013_2015.csv

```
pd.set_option('display.max_columns', None) # Menampilkan semua kolom
pd.set_option('display.expand_frame_repr', False) # Menghindari pembungkusan frame
df.head(10)
```

	kodeProdi	angkatan	semesterAwal	nim	nama	jenisKelamin	tahunAkademikLulus	tanggalLulus	lulus	masaStudi	ayah_pelejaan	ayah_penghasilan
0	22202	2013	20131	10581182313	MUHAMMAD FASID HIKSYAL	L	None	None	False	None	None	None
1	22202	2013	20131	10581182413	MUHAMMAD SAKIR	L	20182	None	True	5 Tahun, 11 Bulan	TANI	500000
2	22202	2013	20131	10581182513	FAISAL AKBAR	None	None	None	False	None	None	None
3	22202	2013	20131	10581182613	MUNIR MUHAIDIN	L	None	None	False	None	None	None
4	22202	2013	20131	10581182713	ST FATMA	F	20172	None	True	4 Tahun, 0 Bulan	WIRASWASTA	500001-1000000
5	22202	2013	20131	10581182813	SAPUL ROZAQ	L	20191	None	True	6 Tahun, 2 Bulan	WIRASWASTA	500001-1000000
6	22202	2013	20131	10581182913	JHFRU	L	None	None	False	None	None	None
7	22202	2013	20131	10581183013	HENRAADI SAPUTRA	L	20182	None	True	5 Tahun, 9 Bulan	WIRASWASTA	200001-3000000
8	22202	2013	20131	10581183113	MUHAMMAD HAER	L	None	None	False	None	TANI	500000
9	22202	2013	20131	10581183213	HILMAN ATANCI TASLIM	L	None	None	False	None	TANI	500001-1000000

```
# Validasi data yang sudah di download dengan NIM yang dicari
nim_yang_dicari = '10581183213' # Ganti dengan NIM yang ingin Anda cari
```

```
data_nim = df.loc[df['nim'] == nim_yang_dicari]
```

```
# Menampilkan hasil pencarian dan cocokkan dengan KHS yang ada
if not data_nim.empty:
```

```
    for index, row in data_nim.iterrows():
        print(f>Data Mahasiswa dengan NIM: {row['nim']}")
        print("=====")
        for column in data_nim.columns:
            print(f"{column}: {row[column]}")
```

```
else:
```

```
    print("Data dengan NIM tersebut tidak ditemukan.")
```

```
Data Mahasiswa dengan NIM: 10581183213
```

```
-----
kodeProdi: 22202
angkatan: 2013
semesterAwal: 20131
nim: 10581183213
nama: HILMAN AFANDI TASLIM
jenisKelamin: L
tahunAkademikLulus: None
tanggalLulus: None
lulus: False
masaStudi: None
ayah_pekerjaan: TANI
ayah_penghasilan: 500001-1000000
ayah_status: HIRUP
ibu_pekerjaan: MIRASMASTA
ibu_penghasilan: 500001-1000000
ibu_status: HIRUP
wali_pekerjaan:
wali_penghasilan:
khs1_tahunAkademik: 20131
khs1_ips: 1.17
khs1_sksSem: 24.0
khs1_ipk: 2.33
khs1_sksTotal: 12.0
...
khs19_sksSem: nan
khs19_ipk: nan
khs19_sksTotal: nan
khs19_statusMahasiswa: nan
Output is truncated. View as a HTML element or copy in a text editor. Adjust cell output settings.
```

```
df = pd.DataFrame(data_list)
df.to_csv('data_mahasiswa.csv', index=False)
print('Data berhasil disimpan ke data_mahasiswa.csv')
```

Python

```
Data berhasil disimpan ke data_mahasiswa.csv
```

```
import pandas as pd

# Baca data dari CSV
file_path = 'data_mahasiswa_angkatan_2016.csv' # Ganti dengan path
file Anda
df = pd.read_csv(file_path)

# Identifikasi kolom SKS Total per semester
sks_total_columns = [col for col in df.columns if 'sksTotal' in col]

# Ambil nilai SKS terakhir yang tidak null
df['SKS_LULUS'] = df[sks_total_columns].ffill(axis=1).iloc[:, -1]

# Simpan hasil ke CSV baru
output_file_path = 'hasil_sks_lulus.csv'
df.to_csv(output_file_path, index=False)

# Tampilkan beberapa baris pertama untuk verifikasi
print(df[['SKS_LULUS']].head())
```

	SKS_LULUS
0	60.0
1	156.0
2	22.0
3	30.0
4	156.0

```
import pandas as pd

# Baca data dari CSV
file_path = 'hasil_sks_lulus.csv' # Ganti dengan path file Anda
df = pd.read_csv(file_path)

# Identifikasi kolom IPK per semester
ipk_columns = [col for col in df.columns if 'ipk' in col]

# Ambil nilai IPK terakhir yang tidak null
df['IPK_TERAKHIR'] = df[ipk_columns].ffill(axis=1).iloc[:, -1]

# Simpan hasil ke CSV baru
output_file_path = 'hasil_ipk_terakhir.csv'
df.to_csv(output_file_path, index=False)

# Tampilkan beberapa baris pertama untuk verifikasi
print(df[['IPK_TERAKHIR']].head())
```

	IPK_TERAKHIR
0	2.90
1	3.08
2	2.91
3	2.80
4	3.18

```
import pandas as pd
import re

# Baca data dari CSV
df = pd.read_csv('hasil_ipk_terakhir.csv')

# Fungsi untuk mengkonversi masa studi menjadi semester
def convert_masa_studi_to_semester(masa_studi):
    if isinstance(masa_studi, str): # Pastikan hanya string yang diproses
        match = re.match(r'(\d+) Tahun, (\d+) Bulan', masa_studi)
        if match:
```

```

        years = int(match.group(1))
        months = int(match.group(2))
        total_months = (years * 12) + months
        semesters = total_months // 6
        return semesters
    return None

# Terapkan fungsi ke kolom 'masaStudi' dan simpan hasilnya ke kolom
baru 'semester'
df['semester'] =
df['masaStudi'].apply(convert_masa_studi_to_semester)

# Simpan kembali data ke CSV
df.to_csv('data_mahasiswa_dengan_semester.csv', index=False)
print('Data berhasil disimpan ke
data_mahasiswa_dengan_semester.csv')

```

Python

Data berhasil disimpan ke data_mahasiswa_dengan_semester.csv

```

import pandas as pd

# Baca data dari CSV
file_path = 'hasil_ipk_terakhir.csv' # Ganti dengan path file Anda
df = pd.read_csv(file_path)

# Identifikasi kolom tahun akademik per semester
tahun_akademik_columns = [col for col in df.columns if
'tahunAkademik' in col]

# Ambil nilai tahun akademik terakhir yang tidak null
df['tahunAkademikTerakhir'] =
df[tahun_akademik_columns].ffill(axis=1).iloc[:, -1]

# Tangani nilai NaN dalam tahunAkademikTerakhir
df['tahunAkademikTerakhir'] =
df['tahunAkademikTerakhir'].fillna(0).astype(int)

# Hitung selisih semester terakhir dari semester awal
# Misalkan semesterAwal adalah tahun + semester (contoh: 20161 untuk
tahun 2016 semester 1)
df['tahunAwal'] = df['semesterAwal'] // 10 # Mengambil tahun dari
semesterAwal
df['semesterAwalNumber'] = df['semesterAwal'] % 10 # Mengambil

```

```
semester dari semesterAwal
```

```
# Hitung jumlah semester antara tahun awal dan tahun terakhir
df['tahunTerakhir'] = df['tahunAkademikTerakhir'].apply(lambda x:
int(str(x)[:4]) if str(x) else 0) # Mengambil tahun dari
tahunAkademikTerakhir
df['semesterTerakhirNumber'] =
df['tahunAkademikTerakhir'].apply(lambda x: int(str(x)[4:]) if
len(str(x)) > 4 else 0) # Mengambil semester dari
tahunAkademikTerakhir
```

```
# Hitung jumlah semester antara tahun awal dan tahun terakhir
df['semesterTerakhir'] = (df['tahunTerakhir'] - df['tahunAwal']) * 2
+ (df['semesterTerakhirNumber'] - df['semesterAwalNumber'] + 1)
```

```
# Simpan hasil ke CSV baru
output_file_path = 'hasil_semester_terakhir.csv'
df.to_csv(output_file_path, index=False)
```

```
# Tampilkan beberapa baris pertama untuk verifikasi
print(df[['semesterAwal', 'tahunAkademikTerakhir',
'semesterTerakhir']].head())
```

```
✓ 0.1s Python
```

	semesterAwal	tahunAkademikTerakhir	semesterTerakhir
0	20131	20162	8
1	20131	20182	12
2	20131	20141	3
3	20131	20141	3
4	20131	20172	10

```
import pandas as pd
```

```
# Baca data dari CSV
```

```
file_path = 'hasil_semester_terakhir.csv' # Ganti dengan path file
Anda
```

```
df = pd.read_csv(file_path)
```

```
# Kategorisasi IPK_TERAKHIR
```

```
bins_ipk = [0.00, 1.99, 2.75, 3.24, 3.79, 4.00]
```

```
labels_ipk = ['Tidak Memuaskan', 'Cukup Memuaskan', 'Memuaskan',
'Sangat Memuaskan', 'Dengan Pujian']
```

```
df['kategori_IPK'] = pd.cut(df['IPK_TERAKHIR'], bins=bins_ipk,
labels=labels_ipk, right=False)
```

```

# Kategorisasi SKS_LULUS berdasarkan semester
def kategorisasi_sks(semester, sks_lulus):
    if semester in range(1, 5):
        if sks_lulus > 40:
            return 'Memenuhi'
        else:
            return 'Tidak Memenuhi'
    elif semester in range(5, 9):
        if sks_lulus > 85:
            return 'Memenuhi'
        else:
            return 'Tidak Memenuhi'
    elif semester in range(9, 15):
        if sks_lulus >= 144:
            return 'Memenuhi'
        else:
            return 'Tidak Memenuhi'
    else:
        return 'Tidak Memenuhi' # Jika semester diluar rentang yang
diberikan

df['kategori_SKS'] = df.apply(lambda row:
kategorisasi_sks(row['semesterTerakhir'], row['SKS_LULUS']), axis=1)

# Simpan hasil ke CSV baru
output_file_path = 'hasil_dengan_kategori.csv'
df.to_csv(output_file_path, index=False)

# Tampilkan beberapa baris pertama untuk verifikasi
print(df[['IPK_TERAKHIR', 'kategori_IPK', 'SKS_LULUS',
'semesterTerakhir', 'kategori_SKS']].head())

```

✓ 0.1s Python

	IPK_TERAKHIR	kategori_IPK	SKS_LULUS	semesterTerakhir	kategori_SKS
0	2.90	Memuaskan	60.0	8	Tidak Memenuhi
1	3.08	Memuaskan	156.0	12	Memenuhi
2	2.91	Memuaskan	22.0	3	Tidak Memenuhi
3	2.80	Memuaskan	30.0	3	Tidak Memenuhi
4	3.18	Memuaskan	156.0	10	Memenuhi

```
import pandas as pd
```

```

# Baca kembali data dari CSV yang telah disimpan
df = pd.read_csv('hasil_dengan_kategori.csv')

```

```
# Seleksi hanya kolom-kolom yang diinginkan
selected_columns = ['angkatan', 'semesterAwal', 'nim', 'nama',
'jenisKelamin', 'ayah_pekerjaan', 'ayah_penghasilan', 'ayah_status',
'ibu_pekerjaan', 'ibu_penghasilan',
'ibu_status', 'wali_pekerjaan', 'wali_penghasilan', 'lulus',
'semesterTerakhir', 'kategori_IPK',
'kategori_SKS' ]
```

```
df_selected = df[selected_columns]
```

```
# Menampilkan beberapa baris pertama untuk memastikan seleksi data
berhasil
```

```
print("Data setelah seleksi:")
print(df_selected.head())
```

```
# Simpan kembali data yang telah dipilih ke dalam file CSV jika
diperlukan
```

```
df_selected.to_csv('data_mahasiswa_selected.csv', index=False)
print('Data berhasil disimpan ke data_mahasiswa_selected.csv')
```

```
Data setelah seleksi:
  angkatan semesterAwal  nim  nama jenisKelamin ayah_pekerjaan ayah_penghasilan ayah_status ibu_pekerjaan ibu_penghasilan ibu_status wali_pekerjaan wali_penghasilan wali_status
0  2003  20031  0806100113  HANAFUZZ FAWZI HUSNUL  Laki  Laki  Laki  Ibu  Ibu  Ibu  Ibu  Ibu  Ibu  Ibu
1  2003  20031  0806100210  YUNAYNUH SAKEL  Perempuan  500000  Tidak  Ibu  Ibu  Ibu  Ibu  Ibu  Ibu  Ibu
2  2003  20031  0806100211  FAISAL ABBAS  Laki  Laki  Laki  Ibu  Ibu  Ibu  Ibu  Ibu  Ibu  Ibu
3  2003  20031  0806100212  NALIN NIKMATUL  Perempuan  1000000  Tidak  Ibu  Ibu  Ibu  Ibu  Ibu  Ibu  Ibu
4  2003  20031  0806100213  SY FATMA  Perempuan  1000000  Tidak  Ibu  Ibu  Ibu  Ibu  Ibu  Ibu  Ibu
data berhasil disimpan ke data_mahasiswa_selected.csv
```

```
# Baca data dari CSV yang telah dipilih kolom-kolomnya
```

```
df = pd.read_csv('data_mahasiswa_selected.csv')
```

```
# Menampilkan informasi mengenai nilai yang hilang sebelum
penghapusan
```

```
print("Jumlah missing values sebelum penghapusan:")
print(df.isnull().sum())
```



```

Jumlah missing values sebelum penghapusan:
angkatan          0
semesterAwal      0
nim               0
nama              0
jenisKelamin      0
ayah_pekerjaan    410
ayah_penghasilan  421
ayah_status       416
ibu_pekerjaan     411
ibu_penghasilan   428
ibu_status        416
wali_pekerjaan    1266
wali_penghasilan  1298
lulus             0
semesterTerakhir  0
kategori_IPK      4
kategori_SKS      0
dtype: int64

```

```

import pandas as pd

# Asumsikan df adalah DataFrame Anda

# 1. Mengisi missing value hanya untuk kolom numerik dengan mean
numeric_cols = df.select_dtypes(include=['number']).columns
df[numeric_cols] = df[numeric_cols].fillna(df[numeric_cols].mean())

# 2. Mengisi missing value untuk kolom kategori (string) dengan mode
categorical_cols = df.select_dtypes(include=['object']).columns
df[categorical_cols] = df[categorical_cols].fillna(df[categorical_cols].mode().iloc[0])

# Tampilkan hasil untuk verifikasi
print(df.head())

```

angkatan	semesterAwal	nim	nama	jenisKelamin	ayah_pekerjaan	ayah_penghasilan	ayah_status	ibu_pekerjaan	ibu_penghasilan	ibu_status	wali_pekerjaan	wali_penghasilan
0	2013	18581382723	MUHAMMAD FARIZ HIKMADI	L	TANI	50000	HIDUP	-	50000	HIDUP	HIRASASTA	50000
1	2013	18581382823	MURAHYAG SAHIR	L	TANI	50000	HIDUP	-	50000	HIDUP	HIRASASTA	50000
2	2013	18581382923	FACHAL AKBAR	L	TANI	50000	HIDUP	-	50000	HIDUP	HIRASASTA	50000
3	2013	18581383023	MARIZ HANAZOR	L	TANI	50000	HIDUP	-	50000	HIDUP	HIRASASTA	50000
4	2013	18581383123	SY FATMA	P	HIRASASTA	50000-100000	HIDUP	HIRASASTA	50000-100000	HIDUP	HIRASASTA	50000-100000

```

import pandas as pd

# Asumsikan df adalah DataFrame Anda setelah pembersihan data

# Menampilkan informasi mengenai nilai yang hilang setelah pembersihan
missing_values_info = df.isnull().sum()
print("Informasi nilai yang hilang setelah pembersihan:")
print(missing_values_info)

```

```

# Jika Anda ingin menampilkan hanya kolom yang masih memiliki
missing values
missing_values_remaining = missing_values_info[missing_values_info >
0]
if not missing_values_remaining.empty:
    print("\nKolom yang masih memiliki nilai yang hilang:")
    print(missing_values_remaining)
else:
    print("\nTidak ada nilai yang hilang yang tersisa setelah
pembersihan.")

# Simpan kembali data yang telah dibersihkan ke dalam file CSV baru
output_file_path = 'data_dibersihkan.csv' # Ganti dengan path file
yang diinginkan
df.to_csv(output_file_path, index=False)

print(f"\nData telah disimpan ke dalam file: {output_file_path}")

```

Informasi nilai yang hilang setelah pembersihan:

```

angkatan      0
semesterAwal  0
nim           0
nama          0
jeniskelamin  0
ayah_pekerjaan  0
ayah_penghasilan  0
ayah_status   0
ibu_pekerjaan  0
ibu_penghasilan  0
ibu_status    0
wali_pekerjaan  0
wali_penghasilan  0
lulus         0
semesterTerakhir  0
kategori_IPK  0
kategori_SKS  0
dtype: int64

```

Tidak ada nilai yang hilang yang tersisa setelah pembersihan.

Data telah disimpan ke dalam file: data_dibersihkan.csv

```

import pandas as pd

# Baca data dari CSV yang telah dibersihkan
df_cleaned = pd.read_csv('data_dibersihkan.csv')

```

```
# Menampilkan jumlah data setelah proses pembersihan
num_rows, num_cols = df_cleaned.shape
print(f"Jumlah data setelah proses pembersihan: {num_rows} baris,
{num_cols} kolom")
```

```
✓ 0.0s Python
Jumlah data setelah proses pembersihan: 1493 baris, 17 kolom
```

```
import pandas as pd

# Baca data dari CSV
file_path = 'data_dibersihkan.csv' # Ganti dengan path file Anda
df = pd.read_csv(file_path)

# Menghapus kolom 'nim' dan 'nama'
df.drop(['nim', 'nama', 'jenisKelamin'], axis=1, inplace=True)

# Simpan hasil ke CSV baru tanpa kolom 'nim' dan 'nama'
output_file_path = 'hasil_tanpa_nim_nama_jk.csv'
df.to_csv(output_file_path, index=False)

# Tampilkan beberapa baris pertama untuk verifikasi
print(df.head())
```

angkatan	semester	id_peserta	ayah_status	ibu_status	lulus	kategori_IPK	kategori_SKS	ibu_status	kategori_IPK
2013	2013	1481	50000	HIDUP	50000	KIDAP	MEMENUHI	50000	0
2013	2013	1481	50000	HIDUP	50000	KIDAP	MEMENUHI	50000	1
2013	2013	1481	50000	HIDUP	50000	KIDAP	MEMENUHI	50000	1
2013	2013	1481	50000	HIDUP	50000	KIDAP	MEMENUHI	50000	1
2013	2013	1481	50000	HIDUP	50000	KIDAP	MEMENUHI	50000	1

```
import pandas as pd

# Baca data dari CSV
file_path = 'hasil_tanpa_nim_nama_jk.csv'
df = pd.read_csv(file_path)

# Definisikan kamus pelabelan untuk kategori IPK, SKS, dan status
label_dict = {
    'ayah_status': {'HIDUP': 0, 'MENINGGAL': 1},
    'lulus': {False: 0, True: 1},
    'kategori_IPK': {
        'Tidak Memuaskan': 0, 'Cukup Memuaskan': 1, 'Memuaskan': 2,
        'Sangat Memuaskan': 3, 'Dengan Pujian': 4
    },
    'kategori_SKS': {'Memenuhi': 0, 'Tidak Memenuhi': 1},
    'ibu_status': {'HIDUP': 0, 'MENINGGAL': 1}
}
```

```

# Definisikan fungsi untuk mengelompokkan pekerjaan wali menjadi 5
kategori
def categorize_wali_job(job):
    if job in ['SWASTA', 'WIRASWASTA', 'MERINTIS USAHA', 'PEGAWAI
BUMN', 'Berdagang']:
        return 'Wiraswasta/Usaha'
    elif job in ['PNS', 'Dosen', 'ASN PPP3K', 'POLDA SULSEL',
'POLRI']:
        return 'PNS/BUMN'
    elif job in ['PENSUIN', 'HONORER', 'HONORER DI SEKOLAH DASAR']:
        return 'Pensiunan/Honor'
    elif job in ['TANI', 'Nelayan', 'Buruh Bangunan', 'TUKANG
BANGUNAN', 'Bengkel las']:
        return 'Petani/Buruh'
    elif job in ['MAHASISWA', 'MAHASISWA S2', 'Asisten apoteker',
'Tenaga kontrak', 'Kontraktor', 'Grab car']:
        return 'Mahasiswa/Profesional'
    else:
        return 'Lainnya'

# Terapkan kategori baru pada kolom pekerjaan wali
df['wali_pekerjaan'] = df['wali_pekerjaan'].map(categorize_wali_job)

# Buat kamus pelabelan untuk kategori pekerjaan wali
wali_job_label_dict = {
    'Wiraswasta/Usaha': 0,
    'PNS/BUMN': 1,
    'Pensiunan/Honor': 2,
    'Petani/Buruh': 3,
    'Mahasiswa/Profesional': 4,
    'Lainnya': 5
}

# Terapkan pelabelan pada kolom pekerjaan wali
df['wali_pekerjaan'] = df['wali_pekerjaan'].map(wali_job_label_dict)

# Definisikan fungsi untuk mengelompokkan penghasilan menjadi 3
kategori
def categorize_income(income):
    if income in ['500000', '500001-1000000']:
        return 'Rendah'
    elif income in ['1000001-1500000', '1500001-2000000']:
        return 'Sedang'
    elif income in ['> 5000000', '2000001-3000000', '3000001-

```

```

5000000']:
    return 'Tinggi'
else:
    return 'Tidak Diketahui'

# Terapkan kategori baru pada kolom penghasilan
df['wali_penghasilan'] =
df['wali_penghasilan'].map(categorize_income)
df['ibu_penghasilan'] = df['ibu_penghasilan'].map(categorize_income)
df['ayah_penghasilan'] =
df['ayah_penghasilan'].map(categorize_income)

# Buat kamus pelabelan untuk kategori penghasilan
income_label_dict = {
    'Rendah': 0,
    'Sedang': 1,
    'Tinggi': 2,
    'Tidak Diketahui': 3
}

# Terapkan pelabelan pada kolom penghasilan
df['wali_penghasilan'] =
df['wali_penghasilan'].map(income_label_dict)
df['ibu_penghasilan'] = df['ibu_penghasilan'].map(income_label_dict)
df['ayah_penghasilan'] =
df['ayah_penghasilan'].map(income_label_dict)

# Definisikan kamus pelabelan untuk pekerjaan ayah dan ibu menjadi 5
kategori
def categorize_father_job(job):
    if job in ['WIRASWASTA', 'SWASTA', 'PNS']:
        return 'Wiraswasta/PNS'
    elif job in ['KEPOLISIAN RI (POLRI)', 'PENSUIN']:
        return 'Polisi/Pensiunan'
    elif job in ['TANI', 'NELAYAN']:
        return 'Petani/Nelayan'
    elif job in ['BURUH NELAYAN/PERIKANAN', 'SECURITY']:
        return 'Buruh/Security'
    elif job in ['CLEANING SERVICE', 'PEGAWAI BUMN', 'Pekerja
Perusahaan']:
        return 'Pekerjaan Lain'
    else:
        return 'Lainnya'

def categorize_mother_job(job):

```

```

if job in ['WIRASWASTA', 'PNS', 'PENSUIN']:
    return 'Wiraswasta/PNS/Pensiunan'
elif job in ['IBU RUMAH TANGGA', 'MENGURUS RUMAH TANGGA']:
    return 'Rumah Tangga'
elif job in ['TANI', 'PEDAGANG']:
    return 'Petani/Pedagang'
elif job in ['SWASTA', 'KARYAWAN BUMD']:
    return 'Swasta/Karyawan'
elif job in ['HONORER', 'Guru']:
    return 'Honor/Guru'
else:
    return 'Lainnya'

# Terapkan kategori baru pada kolom pekerjaan ayah dan ibu
df['ayah_pekerjaan'] =
df['ayah_pekerjaan'].map(categorize_father_job)
df['ibu_pekerjaan'] = df['ibu_pekerjaan'].map(categorize_mother_job)

# Buat kamus pelabelan untuk kategori pekerjaan ayah dan ibu
father_job_label_dict = {
    'Wiraswasta/PNS': 0,
    'Polisi/Pensiunan': 1,
    'Petani/Nelayan': 2,
    'Buruh/Security': 3,
    'Pekerjaan Lain': 4,
    'Lainnya': 5
}

mother_job_label_dict = {
    'Wiraswasta/PNS/Pensiunan': 0,
    'Rumah Tangga': 1,
    'Petani/Pedagang': 2,
    'Swasta/Karyawan': 3,
    'Honor/Guru': 4,
    'Lainnya': 5
}

# Terapkan pelabelan pada kolom pekerjaan ayah dan ibu
df['ayah_pekerjaan'] =
df['ayah_pekerjaan'].map(father_job_label_dict)
df['ibu_pekerjaan'] = df['ibu_pekerjaan'].map(mother_job_label_dict)

# Terapkan pelabelan pada kolom yang sesuai
for column, mapping in label_dict.items():
    if column in df.columns:

```

```

df[column] = df[column].map(mapping)

# Simpan hasil ke CSV baru dengan kolom yang sudah dilabeli
output_file_path = 'hasil_dengan_pelabelan_semua.csv'
df.to_csv(output_file_path, index=False)

# Tampilkan beberapa baris pertama untuk verifikasi
print(df.head())

```

angkatan	semesterawal	ayah_pekerjaan	ayah_penghasilan	ayah_status	ibu_pekerjaan	ibu_penghasilan	ibu_status	wali_pekerjaan	wali_penghasilan	lulus	semesterterakhir	kategori
0	2013	20231	2	0	0	0	0	0	0	0	0	0
1	2013	20231	2	0	0	0	0	0	0	0	1	13
2	2013	20231	2	0	0	0	0	0	0	0	0	3
3	2013	20231	2	0	0	0	0	0	0	0	0	3
4	2013	20231	0	0	0	0	0	0	0	0	1	10

```

import pandas as pd

# Baca data dari CSV
file_path = 'hasil_tanpa_nim_nama_jk.csv'
df = pd.read_csv(file_path)

# Definisikan kamus pelabelan untuk kategori IPK, SKS, dan status
label_dict = {
    'ayah_status': {'HIDUP': 0, 'MENINGGAL': 1},
    'lulus': {False: 0, True: 1},
    'kategori_IPK': {
        'Tidak Memuaskan': 0, 'Cukup Memuaskan': 1, 'Memuaskan': 2,
        'Sangat Memuaskan': 3, 'Dengan Pujian': 4
    },
    'kategori_SKS': {'Memenuhi': 0, 'Tidak Memenuhi': 1},
    'ibu_status': {'HIDUP': 0, 'MENINGGAL': 1}
}

# Definisikan fungsi untuk mengelompokkan pekerjaan wali menjadi 5
kategori
def categorize_wali_job(job):
    if job in ['SWASTA', 'WIRASWASTA', 'MERINTIS USAHA', 'PEGAWAI
BUMN', 'Berdagang']:
        return 'Wiraswasta/Usaha'
    elif job in ['PNS', 'Dosen', 'ASN PPP3K', 'POLDA SULSEL',
'POLRI']:
        return 'PNS/BUMN'
    elif job in ['PENSUIN', 'HONORER', 'HONORER DI SEKOLAH DASAR']:
        return 'Pensiunan/Honor'
    elif job in ['TANI', 'Nelayan', 'Buruh Bangunan', 'TUKANG
BANGUNAN', 'Bengkel las']:
        return 'Petani/Buruh'

```



```

        elif job in ['MAHASISWA', 'MAHASISWA S2', 'Asisten apoteker',
'Tenaga kontrak', 'Kontraktor', 'Grab car']:
            return 'Mahasiswa/Profesional'
        else:
            return 'Lainnya'

# Terapkan kategori baru pada kolom pekerjaan wali
df['wali_pekerjaan'] = df['wali_pekerjaan'].map(categorize_wali_job)

# Buat kamus pelabelan untuk kategori pekerjaan wali
wali_job_label_dict = {
    'Wiraswasta/Usaha': 0,
    'PNS/BUMN': 1,
    'Pensiunan/Honor': 2,
    'Petani/Buruh': 3,
    'Mahasiswa/Profesional': 4,
    'Lainnya': 5
}

# Terapkan pelabelan pada kolom pekerjaan wali
df['wali_pekerjaan'] = df['wali_pekerjaan'].map(wali_job_label_dict)

# Definisikan fungsi untuk mengelompokkan penghasilan menjadi 3
kategori
def categorize_income(income):
    if income in ['500000', '500001-1000000']:
        return 'Rendah'
    elif income in ['1000001-1500000', '1500001-2000000']:
        return 'Sedang'
    elif income in ['> 5000000', '2000001-3000000', '3000001-
5000000']:
        return 'Tinggi'
    else:
        return 'Tidak Diketahui'

# Terapkan kategori baru pada kolom penghasilan
df['wali_penghasilan'] =
df['wali_penghasilan'].map(categorize_income)
df['ibu_penghasilan'] = df['ibu_penghasilan'].map(categorize_income)
df['ayah_penghasilan'] =
df['ayah_penghasilan'].map(categorize_income)

# Buat kamus pelabelan untuk kategori penghasilan
income_label_dict = {
    'Rendah': 0,

```

```

    'Sedang': 1,
    'Tinggi': 2,
    'Tidak Diketahui': 3
}

# Terapkan pelabelan pada kolom penghasilan
df['wali_penghasilan'] =
df['wali_penghasilan'].map(income_label_dict)
df['ibu_penghasilan'] = df['ibu_penghasilan'].map(income_label_dict)
df['ayah_penghasilan'] =
df['ayah_penghasilan'].map(income_label_dict)

# Definisikan fungsi untuk mengelompokkan pekerjaan ayah dan ibu
menjadi 5 kategori
def categorize_father_job(job):
    if job in ['WIRASWASTA', 'SWASTA', 'PNS']:
        return 'Wiraswasta/PNS'
    elif job in ['KEPOLISIAN RI (POLRI)', 'PENSUIN']:
        return 'Polisi/Pensiunan'
    elif job in ['TANI', 'NELAYAN']:
        return 'Petani/Nelayan'
    elif job in ['BURUH NELAYAN/PERIKANAN', 'SECURITY']:
        return 'Buruh/Security'
    elif job in ['CLEANING SERVICE', 'PEGAWAI BUMN', 'Pekerja
Perusahaan']:
        return 'Pekerjaan Lain'
    else:
        return 'Lainnya'

def categorize_mother_job(job):
    if job in ['WIRASWASTA', 'PNS', 'PENSUIN']:
        return 'Wiraswasta/PNS/Pensiunan'
    elif job in ['IBU RUMAH TANGGA', 'MENGURUS RUMAH TANGGA']:
        return 'Rumah Tangga'
    elif job in ['TANI', 'PEDAGANG']:
        return 'Petani/Pedagang'
    elif job in ['SWASTA', 'KARYAWAN BUMD']:
        return 'Swasta/Karyawan'
    elif job in ['HONORER', 'Guru']:
        return 'Honor/Guru'
    else:
        return 'Lainnya'

# Terapkan kategori baru pada kolom pekerjaan ayah dan ibu
df['ayah_pekerjaan'] =

```

```

df['ayah_pekerjaan'].map(categorize_father_job)
df['ibu_pekerjaan'] = df['ibu_pekerjaan'].map(categorize_mother_job)

# Buat kamus pelabelan untuk kategori pekerjaan ayah dan ibu
father_job_label_dict = {
    'Wiraswasta/PNS': 0,
    'Polisi/Pensiunan': 1,
    'Petani/Nelayan': 2,
    'Buruh/Security': 3,
    'Pekerjaan Lain': 4,
    'Lainnya': 5
}

mother_job_label_dict = {
    'Wiraswasta/PNS/Pensiunan': 0,
    'Rumah Tangga': 1,
    'Petani/Pedagang': 2,
    'Swasta/Karyawan': 3,
    'Honor/Guru': 4,
    'Lainnya': 5
}

# Terapkan pelabelan pada kolom pekerjaan ayah dan ibu
df['ayah_pekerjaan'] =
df['ayah_pekerjaan'].map(father_job_label_dict)
df['ibu_pekerjaan'] = df['ibu_pekerjaan'].map(mother_job_label_dict)

# Terapkan pelabelan pada kolom yang sesuai
for column, mapping in label_dict.items():
    if column in df.columns:
        df[column] = df[column].map(mapping)

# Definisikan fungsi untuk menentukan status potensi putus studi
def potential_dropout_status(row):
    # Define your criteria for potential dropout status
    if row['ayah_pekerjaan'] == 2 and row['ayah_penghasilan'] in [0,
3] and row['ayah_status'] == 1 or row['ibu_pekerjaan'] == 1 and
row['ibu_penghasilan'] in [0, 3] and row['ibu_status'] == 1 or
    row['wali_pekerjaan'] in [3, 4] and row['wali_penghasilan'] in
[0, 3] or row['lulus'] == 0 and row['semesterTerakhir'] > 8 and
row['kategori_IPK'] <= 1 or row['kategori_SKS'] == 1:
        return 'Potensial Putus Studi'
    else:
        return 'Tidak Potensial'

```

```

# Terapkan fungsi untuk menambahkan kolom status potensi putus studi
df['status_potensi_putus_studi'] =
df.apply(potential_dropout_status, axis=1)

# Simpan hasil ke CSV baru dengan kolom status potensi putus studi
output_file_path = 'hasil_dengan_status_potensi_putus_studi.csv'
df.to_csv(output_file_path, index=False)

# Tampilkan beberapa baris pertama untuk verifikasi
print(df.head())

```

ibp_penghasilan	ayah_status	ibu_pekerjaan	ibu_penghasilan	ibu_status	wali_pekerjaan	wali_penghasilan	lulus	semesterTerakhir	kategori_DPK	kategori_DCS	status_potensi_putus_studi
2	0	0	2	0	0	0	0	8	0	1	Potensial Putus Studi
0	1	5	0	0	0	0	1	14	2	0	Tidak Potensial
1	1	0	2	0	0	2	1	13	2	0	Tidak Potensial
2	0	0	1	0	0	0	1	9	2	0	Tidak Potensial
1	0	1	0	0	0	0	1	8	1	0	Tidak Potensial

```

# Definisikan kamus pelabelan untuk status potensi putus studi
dropout_status_label_dict = {
    'Potensial Putus Studi': 0,
    'Tidak Potensial': 1
}

# Terapkan pelabelan pada kolom status potensi putus studi
df['status_potensi_putus_studi'] =
df['status_potensi_putus_studi'].map(dropout_status_label_dict)

# Simpan hasil ke CSV baru dengan kolom status potensi putus studi
output_file_path =
'hasil_dengan_pelabelan_status_potensi_putus_studi.csv'
df.to_csv(output_file_path, index=False)

# Tampilkan beberapa baris pertama untuk verifikasi
print(df.head())

```

ibp_penghasilan	ayah_status	ibu_pekerjaan	ibu_penghasilan	ibu_status	wali_pekerjaan	wali_penghasilan	lulus	semesterTerakhir	kategori_DPK	kategori_DCS	status_potensi_putus_studi
0	0	5	0	0	0	0	0	8	1	1	Potensial Putus Studi
0	0	5	0	0	0	0	1	14	2	0	Tidak Potensial
0	0	5	0	0	0	0	0	1	2	1	Potensial Putus Studi
0	0	5	0	0	0	0	0	1	2	1	Potensial Putus Studi
0	0	0	0	0	0	0	1	10	2	0	Tidak Potensial

```

import pandas as pd
from sklearn.model_selection import train_test_split

# Baca data dari CSV
file_path = 'hasil_dengan_pelabelan_status_potensi_putus_studi.csv'
df = pd.read_csv(file_path)

# Pisahkan fitur dan label target

```

```

X = df.drop(columns=['status_potensi_putus_studi'])
y = df['status_potensi_putus_studi']

# Bagi data menjadi data latih dan data uji
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42) # 20% data uji

# Hitung jumlah data latih dan data uji
num_train_samples = len(X_train)
num_test_samples = len(X_test)

# Hitung jumlah data berdasarkan kategori di data latih dan data uji
train_counts = y_train.value_counts()
test_counts = y_test.value_counts()

# Hitung jumlah total berdasarkan kategori di seluruh dataset
total_counts = df['status_potensi_putus_studi'].value_counts()

# Simpan data latih dan data uji ke file CSV
X_train.to_csv('data_latih.csv', index=False)
X_test.to_csv('data_uji.csv', index=False)
y_train.to_csv('label_latih.csv', index=False)
y_test.to_csv('label_uji.csv', index=False)

# Tampilkan jumlah data
print(f'Jumlah data latih: {num_train_samples}')
print(f'Jumlah data uji: {num_test_samples}')
print(f'Jumlah data latih berdasarkan kategori:\n{train_counts}')
print(f'Jumlah data uji berdasarkan kategori:\n{test_counts}')
print(f'Jumlah total data berdasarkan kategori:\n{total_counts}')

```

```

Jumlah data latih: 1194
Jumlah data uji: 299
Jumlah data latih berdasarkan kategori:
status_potensi_putus_studi
0    612
1    582
Name: count, dtype: int64
Jumlah data uji berdasarkan kategori:
status_potensi_putus_studi
0    159
1    140
Name: count, dtype: int64
Jumlah total data berdasarkan kategori:
status_potensi_putus_studi
0    771
1    722
Name: count, dtype: int64

```

```

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier, export_text
from sklearn.metrics import classification_report, accuracy_score
import numpy as np

# Fungsi untuk menghitung entropi
def entropy(target_col):
    elements, counts = np.unique(target_col, return_counts=True)
    entropy_val = np.sum([(-counts[i]/np.sum(counts)) *
np.log2(counts[i]/np.sum(counts)) for i in range(len(elements))])
    return entropy_val

# Fungsi untuk menghitung gain informasi
def info_gain(data, split_attribute_name,
target_name="status_potensi_putus_studi"):
    total_entropy = entropy(data[target_name])

    # Hitung entropi rata-rata untuk setiap nilai pada atribut yang
dipisah
    vals, counts = np.unique(data[split_attribute_name],
return_counts=True)
    weighted_entropy = np.sum([(counts[i]/np.sum(counts)) *
entropy(data.where(data[split_attribute_name]==vals[i]).dropna()[tar
get_name]) for i in range(len(vals))])

    # Hitung informasi gain
    information_gain = total_entropy - weighted_entropy
    return information_gain

```

```

# Baca data dari CSV
file_path = 'hasil_dengan_pelabelan_status_potensi_putus_studi.csv'
df = pd.read_csv(file_path)

# Pisahkan fitur dan label target
X = df.drop(columns=['status_potensi_putus_studi'])
y = df['status_potensi_putus_studi']

# Bagi data menjadi data latih dan data uji (random_state=42 untuk
konsistensi)
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Inisialisasi dan latih model Decision Tree dengan kriteria entropi
model = DecisionTreeClassifier(criterion='entropy', random_state=1)
model.fit(X_train, y_train)

# Lakukan prediksi
y_pred = model.predict(X_test)

# Evaluasi model
accuracy = accuracy_score(y_test, y_pred)
report = classification_report(y_test, y_pred)

# Tampilkan hasil evaluasi
print(f'Akurasi: {accuracy:.4f}')
print('Laporan Klasifikasi:')
print(report)

# Tampilkan aturan Decision Tree
tree_rules = export_text(model, feature_names=list(X.columns))
print("Aturan Decision Tree:")
print(tree_rules)

# Menghitung gain informasi untuk setiap fitur
for col in X.columns:
    gain = info_gain(df, col,
target_name='status_potensi_putus_studi')
    print(f'Gain informasi untuk {col}: {gain:.4f}')

```



```

# Loop melalui nilai k dari 2 hingga 10
for k in range(2, 11):
    # Menggunakan StratifiedKFold untuk memastikan distribusi label
    # yang seimbang di setiap fold
    kfold = StratifiedKFold(n_splits=k, shuffle=True,
random_state=1)
    scores = cross_val_score(model, X, y, cv=kfold,
scoring='accuracy')

    # Menampilkan hasil akurasi untuk setiap fold
    print(f"Akurasi untuk k={k}:")
    for i, score in enumerate(scores, start=1):
        print(f" Fold {i}: {score:.4f}")
    print(f" Rata-rata akurasi: {np.mean(scores):.4f}")
    print(f" Standar deviasi akurasi: {np.std(scores):.4f}")
    print("\n")

```

```

Akurasi untuk k=2:
Fold 1: 0.9987
Fold 2: 0.9973
Rata-rata akurasi: 0.9980
Standar deviasi akurasi: 0.0007

```

```

Akurasi untuk k=3:
Fold 1: 0.9980
Fold 2: 0.9900
Fold 3: 0.9980
Rata-rata akurasi: 0.9953
Standar deviasi akurasi: 0.0038

```

```

Akurasi untuk k=4:
Fold 1: 0.9973
Fold 2: 0.9973
Fold 3: 0.9946
Fold 4: 0.9973
Rata-rata akurasi: 0.9967
Standar deviasi akurasi: 0.0012

```

```

Akurasi untuk k=5:
...
Rata-rata akurasi: 0.9966
Standar deviasi akurasi: 0.0034

```

Output is truncated. View as a [scrollable element](#) or open in a [text editor](#). Adjust cell output [settings](#)...

```

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report, confusion_matrix,
accuracy_score
import matplotlib.pyplot as plt
import seaborn as sns

# Baca kembali data dari CSV yang telah di-transformasi
file_path = 'hasil_dengan_pelabelan_status_potensi_putus_studi.csv'
df = pd.read_csv(file_path)

# Mengganti nilai NaN dengan nilai rata-rata dari kolom yang
bersangkutan
df = df.fillna(df.mean())

# Misalkan target kolom adalah 'potensi_putus_studi'
X = df.drop(columns=['status_potensi_putus_studi'])
y = df['status_potensi_putus_studi']

# Membagi data menjadi data training dan data testing (80% training,
20% testing)
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Inisialisasi model Decision Tree C4.5 (menggunakan entropy)
model = DecisionTreeClassifier(criterion='entropy', random_state=1)

# Melatih model dengan data training
model.fit(X_train, y_train)

# Melakukan prediksi pada data testing
y_pred = model.predict(X_test)

# Menghitung dan menampilkan metrik evaluasi
accuracy = accuracy_score(y_test, y_pred)
cm = confusion_matrix(y_test, y_pred)
report = classification_report(y_test, y_pred, output_dict=True)

print(f"Accuracy: {accuracy:.4f}")
print("Confusion Matrix:")
print(cm)
print("Classification Report:")
print(classification_report(y_test, y_pred))

```

```

# Menghitung rata-rata precision, recall, dan F1-Score
avg_precision = (report['0']['precision'] +
report['1']['precision']) / 2
avg_recall = (report['0']['recall'] + report['1']['recall']) / 2
avg_f1_score = (report['0']['f1-score'] + report['1']['f1-score']) /
2

print(f"Rata-rata Precision: {avg_precision:.4f}")
print(f"Rata-rata Recall: {avg_recall:.4f}")
print(f"Rata-rata F1-Score: {avg_f1_score:.4f}")

# Plot confusion matrix
plt.figure(figsize=(10, 7))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
xticklabels=['Tidak Potensial', 'Potensial Putus Studi'],
yticklabels=['Tidak Potensial', 'Potensial Putus Studi'])
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix - Decision Tree C4.5')
plt.show()

· Accuracy: 1.00
Confusion Matrix:
[[159  0]
 [ 0 140]]
Classification Report:

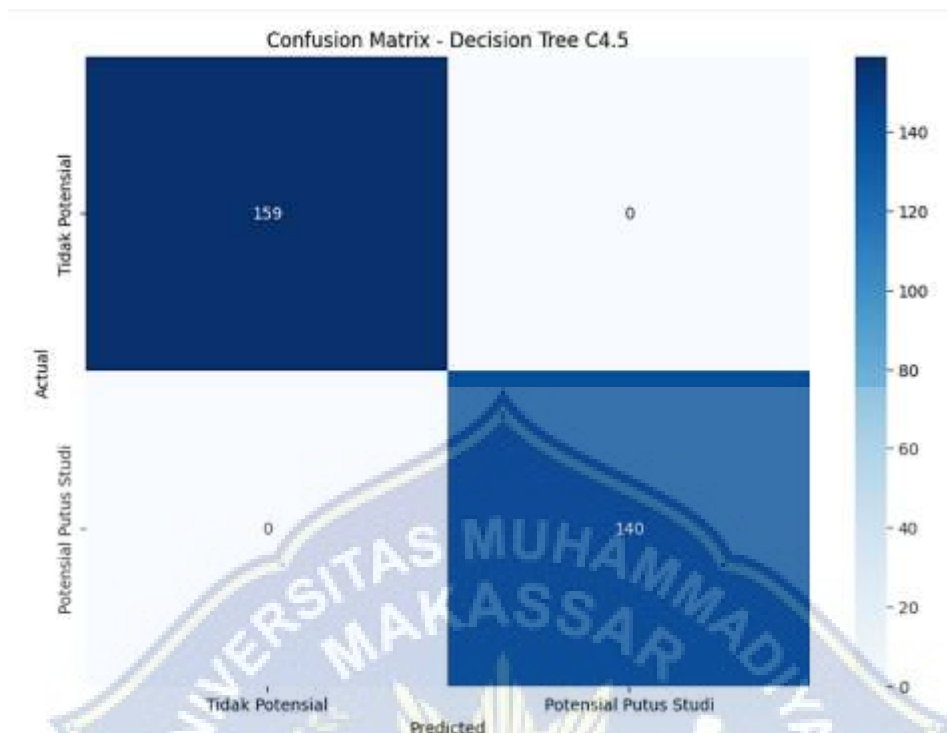
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	159
1	1.00	1.00	1.00	140
accuracy			1.00	299
macro avg	1.00	1.00	1.00	299
weighted avg	1.00	1.00	1.00	299

```

Rata-rata Precision: 1.0000
Rata-rata Recall: 1.0000
Rata-rata F1-Score: 1.0000

```



```
import pandas as pd
import numpy as np
from sklearn.tree import DecisionTreeClassifier, plot_tree,
export_text
import matplotlib.pyplot as plt

# Baca data dari CSV yang telah di-transformasi
file_path = 'hasil_dengan_pelabelan_status_potensi_putus_studi.csv'
df = pd.read_csv(file_path)

# Pisahkan fitur dan target
X = df.drop(columns=['status_potensi_putus_studi'])
y = df['status_potensi_putus_studi']

# Encode categorical features (melakukan encoding untuk fitur
kategorikal)
X_encoded = pd.get_dummies(X)

# Buat model Decision Tree dengan parameter untuk detail lebih
banyak
model = DecisionTreeClassifier(
    criterion='entropy', # Menggunakan entropy sebagai kriteria
(C4.5)
    max_depth=4, # Batasi kedalaman pohon lebih besar
```

```

untuk lebih banyak detail
    min_samples_split=5, # Jumlah sampel minimum untuk membagi
node
    min_samples_leaf=3 # Jumlah sampel minimum yang diperlukan
di setiap node daun
)
model.fit(X_encoded, y)

# Simpan gambar pohon keputusan ke file
plt.figure(figsize=(25, 15)) # Ukuran figure yang lebih besar untuk
menampilkan detail lebih baik
plot_tree(
    model,
    feature_names=X_encoded.columns,
    class_names=['Potensial Putus Studi', 'Tidak Potensial'],
    filled=True,
    rounded=True,
    fontsize=10
)
plt.savefig('decision_tree_plot.png', format='png',
bbox_inches='tight') # Simpan ke file PNG

# Tampilkan gambar pohon keputusan
plt.show()

# Ekstrak dan tampilkan semua aturan pohon keputusan
tree_rules = export_text(model,
feature_names=list(X_encoded.columns))
print(tree_rules)

# Fungsi untuk memprediksi satu-satu dan menjelaskan proses
keputusannya
def prediksi_individual_dan_jalur(data, model, feature_names):
    node_indicator = model.decision_path(data)
    leaf_id = model.apply(data)

    # Dapatkan prediksi
    prediction = model.predict(data)
    class_name = 'Tidak Potensial' if prediction[0] == 2 else
'Potensial Putus Studi'

    # Identifikasi jalur keputusan yang diambil
    node_index =
node_indicator.indices[node_indicator.indptr[0]:node_indicator.indptr[1]]

```

```

# Simpan penjelasan jalur keputusan
explanation = f"Mahasiswa ini diklasifikasikan sebagai
'{class_name}' melalui jalur berikut:\n"
for node_id in node_index:
    if leaf_id[0] == node_id:
        continue

    threshold_sign = "<=" if data.iloc[0,
model.tree_.feature[node_id]] <= model.tree_.threshold[node_id] else
">"

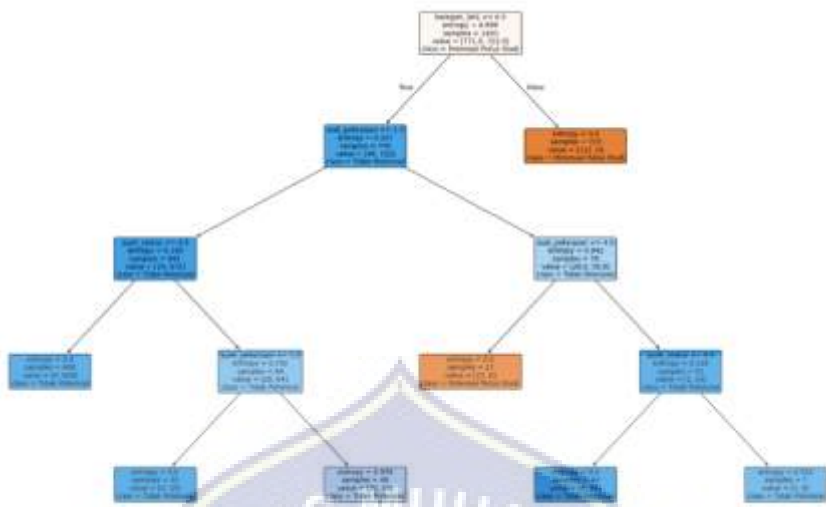
    feature = feature_names[model.tree_.feature[node_id]]
    threshold = model.tree_.threshold[node_id]

    explanation += f"- Fitur '{feature}' dengan nilai
{data.iloc[0, model.tree_.feature[node_id]]} {threshold_sign}
{threshold:.4f}\n"

return class_name, explanation

# Prediksi satu per satu untuk setiap mahasiswa dan berikan
penjelasan
for i in range(len(X_encoded)):
    data_mahasiswa = X_encoded.iloc[[i]] # Pilih satu baris data
    hasil_prediksi, penjelasan_jalur =
prediksi_individual_dan_jalur(data_mahasiswa, model,
X_encoded.columns)
    print(f"Mahasiswa {i+1}: Prediksi = {hasil_prediksi}")
    print(penjelasan_jalur)

```

```

--- kategori_SKS <= 0.50
|--- wali_pekerjaan <= 2.50
|--- ayah_status <= 0.50
|   |--- class: 1
|--- ayah_status > 0.50
|   |--- ayah_pekerjaan <= 1.50
|       |--- class: 1
|   |--- ayah_pekerjaan > 1.50
|       |--- class: 1
|--- wali_pekerjaan > 2.50
|--- wali_pekerjaan <= 4.50
|   |--- class: 0
|--- wali_pekerjaan > 4.50
|   |--- ayah_status <= 0.50
|       |--- class: 1
|   |--- ayah_status > 0.50
|       |--- class: 1
--- kategori_SKS > 0.50
|--- class: 0
  
```

Mahasiswa 1: Prediksi = Potensial Putus Studi
 Mahasiswa ini diklasifikasikan sebagai 'Potensial Putus Studi' melalui jalur berikut:
 - Fitur 'kategori_SKS' dengan nilai 1 > 0.5000

Mahasiswa 2: Prediksi = Potensial Putus Studi
 ...
 - Fitur 'wali_pekerjaan' dengan nilai 1 <= 2.5000
 - Fitur 'ayah_status' dengan nilai 1 > 0.5000
 - Fitur 'ayah_pekerjaan' dengan nilai 1 <= 1.5000

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

Lampiran 6. Permohonan Penelitian

SURAT PERMOHONAN PENELITIAN

301

Hal : Permohonan Surat Pengantar Penelitian

Kepada Yth,

Ketua Program Studi Informatika

Di

Tempat

Assalamu Alaikum Warahmatullahi Wabarakatuh

Sehubungan dengan akan dilaksanakannya Penelitian untuk penyelesaian tugas akhir dengan judul penelitian "*Klasifikasi Mahasiswa Berpotensi Putus Studi Menggunakan Algoritma Decision Tree Pada Fakultas Teknik Muhammadiyah Makassar*", yang akan dilaksanakan di Instansi oleh setiap mahasiswa Fakultas Teknik Program Studi informatika. Adapun Mahasiswa yang bersangkutan adalah sebagai berikut :

No	Nama	NIM
1	YUMI	105841102120

Maka dengan ini kami memohon dibuatkan surat pengantar atau pengajuan Penelitian pada Instansi dibawah ini.

Nama Instansi : SIMAK Universitas Muhammadiyah Makassar

Alamat : Jl. Sultan Allauddin No 259

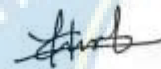
Demikian surat permohonan kami ajukan, atas dukungan dan kerjasamanya kami haturkan terima kasih.

Billahi Fii Sabiilihq, Fastabiqul Khairat

Walaikumusalam Warahmatullahi Wabarakatuh

Makassar, $\frac{7 \text{ Dzulhijjah}}{14 \text{ Juni}}$ $\frac{1445 \text{ H}}{2024 \text{ M}}$

Pemohonan



YUMI
105841102120

Lampiran 7. Pengantar Penelitian

**UNIVERSITAS MUHAMMADIYAH MAKASSAR**
FAKULTAS TEKNIK
PROGRAM STUDI INFORMATIKA

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Nomor : 384/05/C.4-VI/VII/45/2024
Lamp. : -
Hal : **Pengantar Penelitian**

Makassar, 24 Dzulhijjah 1445 H
01 Juli 2024 M

Kepada yang Terhormat,
Ketua LP3M Unismuh Makassar
Di -
Tempat

Assalamu 'Alaikum Warahmatullahi Wabarakatuh

Dengan Rahmat Allah SWT, Semoga aktivitas kita bernilai ibadah di Sisi - Nya. Dalam rangka penyelesaian Tugas Sarjana / Tugas Akhir Mahasiswa pada Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar dengan judul: "*Klasifikasi Mahasiswa Berpotensi Patus Studi Menggunakan Algoritma Decision Tree pada Fakultas Teknik Muhammadiyah Makassar*", Schubungan hal tersebut, maka kami meminta kesedian Bapak/Ibu agar kiranya berkenan membantu perihal surat tersebut. Bersama ini kami sampaikan mahasiswa(i):

No.	Stambuk	Nama
1.	105 84 11021 20	Yumi

Demikian surat kami atas perhatian dan kerja samanya kami haturkan banyak terima kasih.
Jazakumullah Khaeran Katsiran
Wassalamu 'Alaikum warahmatullah Wabarakatuh

Ketua Program Studi
Informatika

Mulyadi A. M. Hayat, S.Kom., MT.
NIDN 11010104577

Tembusan: Kepada Yang Terhormat,
1 Dekan Fakultas Teknik
2 Arsip



Gedung Menara Iqra Lantai 3
Jl. Sultan Alauddin No. 259 Telp. (0411) 866 972 Fax (0411) 865 588 Makassar 90221
Web: <https://teknik.unismuh.ac.id/>, e-mail: teknik@unismuh.ac.id



Lampiran 8. Bebas Plagiat



MAJELIS PENDIDIKAN TINGGI PIMPINAN PUSAT MUHAMMADIYAH
UNIVERSITAS MUHAMMADIYAH MAKASSAR
UPT PERPUSTAKAAN DAN PENERBITAN

Alamat Kantor: Jl. Sultan Alauddin No.259 Makassar 90221 Tlp.(0411) 866972,881593, Fax.(0411) 865588

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

SURAT KETERANGAN BEBAS PLAGIAT

UPT Perpustakaan dan Penerbitan Universitas Muhammadiyah Makassar,
Menerangkan bahwa mahasiswa yang tersebut namanya di bawah ini:

Nama : Yumi

Nim : 105841102120

Program Studi : Teknik Informatika

Dengan nilai:

No	Bab	Nilai	Ambang Batas
1	Bab 1	10 %	10 %
2	Bab 2	25 %	25 %
3	Bab 3	8 %	10 %
4	Bab 4	0 %	10 %
5	Bab 5	4 %	5 %

Dinyatakan telah lulus cek plagiat yang diadakan oleh UPT- Perpustakaan dan Penerbitan
Universitas Muhammadiyah Makassar Menggunakan Aplikasi Turnitin.

Demikian surat keterangan ini diberikan kepada yang bersangkutan untuk dipergunakan
seperlunya.

Makassar, 28 Agustus 2024

Mengetahui,

Kepala UPT- Perpustakaan dan Penerbitan,



Nurrahmi, S.Hum.,M.I.P
NIM.1964591

Jl. Sultan Alauddin no 259 makassar 90222
Telepon (0411)866972,881 593,fax (0411)865 588
Website: www.library.unismuh.ac.id
E-mail : perpustakaan@unismuh.ac.id

Lampiran 9. Hasil Plagiat Per BAB



3 I Yumi 105841102120

ORIGINALITY REPORT

10%

SIMILARITY INDEX

6%

INTERNET SOURCES

8%

PUBLICATIONS

3%

STUDENT PAPERS

PRIMARY SOURCES

1

core.ac.uk

Internet Source

2%

2

Fahrim Irfhamna Rachman, Syaikhadilah, Titin Wahyuni, Lukman Anas. "Prediksi Tingkat Kelulusan Menggunakan K-Means Pada Program Studi Informatika Unismuh Makassar", JURNAL FASILKOM, 2023

Publication

2%

3

Yudhi Pratama Tanjung, Steven R. Sentinuwo, Agustinus Jacobus. "Penentuan Daya Listrik Rumah Tangga Menggunakan Metode Decision Tree", Jurnal Teknik Informatika, 2016

Publication

2%

4

journal.unismuh.ac.id

Internet Source

2%

5

repositori.uin-alaudidin.ac.id

Internet Source

2%



BAB II Yumi 105841102120

by Tutup Tahap



Submission date: 28-Aug-2024 06:34AM (UTC+0700)

Submission ID: 2439360189

File name: SKRIPSI_BAB_II_YUMI.docx (62.58K)

Word count: 2557

Character count: 16502

II Yumi 105841102120

ORIGINALITY REPORT

25%

SIMILARITY INDEX

26%

INTERNET SOURCES


15%

PUBLICATIONS

12%

STUDENT PAPERS

PRIMARY SOURCES



1	ejurnal.seminar-id.com Internet Source	4%
2	pdfs.semanticscholar.org Internet Source	3%
3	jurnal.fmipa.unimul.ac.id Internet Source	3%
4	repository.potensi-utama.ac.id Internet Source	2%
5	18611147.medium.com Internet Source	2%
6	skripsi.tunasbangsa.ac.id Internet Source	2%
7	ojs.itb-ad.ac.id Internet Source	2%
8	jurnal.darmajaya.ac.id Internet Source	2%
9	teknik.unismuh.ac.id Internet Source	2%

jurnalteknik.unkris.ac.id
Internet Source

2%



jurnal.unej.ac.id
Internet Source

2%

Exclude quotes Off
Exclude bibliography Off

Exclude matches + 2%



BAB III Yumi 105841102120

by Tutup Tahap



Submission date: 28-Aug-2023 06:35AM (UTC+0700)

Submission ID: 2439360553

File name: SKRIPSI_BAB_III_YUMI.docx (67.9K)

Word count: 736

Character count: 4811

III Yumi 105841102120

ORIGINALITY REPORT

8%

SIMILARITY INDEX

10%

INTERNET SOURCES

4%

PUBLICATIONS

8%

STUDENT PAPERS

PRIMARY SOURCES

1

repository.uin-suska.ac.id

Internet Source

3%

2

Submitted to Universitas Muhammadiyah
Makassar

Student Paper

3%

3

repository.upi.edu

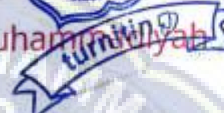
Internet Source

2%

Exclude quotes Off

Exclude bibliography Off

Exclude matches < 2%



BAB IV Yumi 105841102120

by Tutup Tahap



Submission date: 28-Aug-2024 06:36AM (UTC+0700)

Submission ID: 2439361110

File name: SKRIPSI_BAB_IV_YUMI.docx (2.21M)

Word count: 4692

Character count: 27910

AB IV Yumi 105841102120

ORIGINALITY REPORT

0%

SIMILARITY INDEX

0%

INTERNET SOURCES

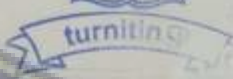
0%

PUBLICATIONS

0%

STUDENT PAPERS

PRIMARY SOURCES



Exclude quotes

OFF

Exclude matches

Exclude bibliography



BAB V Yumi 105841102120

by Tutup Tahap



Submission date: 28-Aug-2024 06:29:41AM (UTC+0700)

Submission ID: 2493461508

File name: SKRIPSI BAB V YUMI.docx (18.27K)

Word count: 196

Character count: 1347

BAB V Yumi 105841102120

ORIGINALITY REPORT

4%

SIMILARITY INDEX

4%

INTERNET SOURCES

0%

PUBLICATIONS

0%

STUDENT PAPERS

PRIMARY SOURCES



ojs.uho.ac.id
Internet Source

4%



Exclude quotes Off
Exclude bibliography Off

Exclude matches