

**KLASIFIKASI MAHASISWA BERPOTENSI PUTUS STUDI
MENGUNAKAN *ALGORITMA NAIVE BAYES* PADA FAKULTAS
TEKNIK UNISMUH MAKASSAR**

SKRIPSI

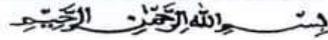
Diajukan Sebagai Salah Satu Syarat Untuk Mendapat
Gelar Sarjana Komputer (S.Kom) Program Studi Informatika



RIANITA KAMSURYA

105841102620

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH MAKASSAR
2024**



PENGESAHAN

Skripsi atas nama Rianita Kamsurya dengan nomor induk Mahasiswa 105 84 11026 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, S.T., M.T., IPU

b. Dekan Fakultas Teknik Universitas Hasanuddin

Prof. Dr. Eng. Muhammad Isran Ramli, S.T., M.T.

2. Penguji

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

b. Sekretaris : Lukman, S.Kom., M.T.

3. Anggota

1. Fahrin Irfhamna Rahman, S.Kom., M.T.

2. Titin Wahyuni, S.Pd., M.T.

3. Lukman Anas, S.Kom., M.T.

Mengetahui :

Pembimbing I

Pembimbing II

Muhyiddin A. M. Hayat, S.Kom., M.T.

Rizki Yusliana Bakti, S.T., M.T.

Dekan





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 NAIVE BAYES PADA FAKULTAS TEKNIK UNISMUH MAKASSAR**

Nama : **RIANITA KAMSURYA**
Stanbuk : **105841102620**

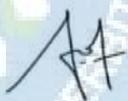
Makassar, 09 September 2024

Telah Diperiksa dan Disetujui
Oleh Dosen Pembimbing;

Pembimbing I

Pembimbing II


Muhyiddin A. M. Nayat, S.Kom., M.T.


Rizki Yuliana Bakti, ST., M.T.

Mengetahui,

Ketua Program Studi Informatika


Muhyiddin A. M. Nayat, S.Kom., M.T.
NBM 1504577

ABSTRAK

Penelitian ini bertujuan untuk mengidentifikasi dan menganalisis faktor-faktor yang mempengaruhi potensi putus studi mahasiswa di Fakultas Teknik Universitas Muhammadiyah Makassar. Penelitian ini dilakukan di Universitas Muhammadiyah Makassar, khususnya di Fakultas Teknik, pada periode 2013 sampai 2015. Metode penelitian yang digunakan meliputi pengumpulan data, praproses data, pembagian data, klasifikasi menggunakan algoritma Naive Bayes, dan pengujian sistem. Dari hasil yang didapat, algoritma naive bayes memiliki hasil peforma yang cukup tinggi dengan akurasi sebesar 97%. Sehingga algoritma ini menjadi salah satu algoritma yang baik dalam mengklasifikasi mahasiswa potensi putus studi dan tidak potensi pada jurusan Teknik Informatika Universitas Muhammadiyah Makassar. Pengujian menggunakan 10-fold cross-validation menunjukkan rata-rata akurasi 95,38%. Hasil ini mengindikasikan bahwa algoritma Naïve Bayes dapat memberikan hasil yang konsisten dan andal dalam mengklasifikasikan mahasiswa berpotensi putus studi.

Kata kunci: Naïve Bayes, Klasifikasi, Preprocessing Data, Cross-Validation, Status Putus Studi.

ABSTRACT

This study aims to identify and analyze the factors that influence the potential for students to drop out of study at the Faculty of Engineering, Muhammadiyah University of Makassar. This research was conducted at Muhammadiyah University of Makassar, specifically at the Faculty of Engineering, in the period of 2013 to 2016 . The research methods used include data collection, data preprocessing, data division, classification using the Naive Bayes algorithm, and system testing. From the results obtained, the naive bayes algorithm has a fairly high performance with an accuracy of 97%. So this algorithm is one of the good algorithms in classifying students with the potential to drop out of study and those without potential in the Informatics Engineering Department, Muhammadiyah University of Makassar. Testing using 10-fold cross-validation showed average accuracy of 95,38%. These results indicate that the Naïve Bayes algorithm can provide consistent and reliable results in classifying students with the potential to drop out of study.

Keywords: *Naïve Bayes, Classification, Data Preprocessing, Cross-Validation, Study Dropout Status.*



KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Segala puji bagi Allah Subhanallahu Wa Ta'ala atas limpahan rahmat dan karunianya sehingga penulis dapat menyelesaikan Laporan Tugas Akhir yang berjudul “ KLASIFIKASI MAHASISWA BERPOTENSI PUTUS STUDI MENGGUNAKAN ALGORITMA NAÏVE BAYES PADA FAKULTAS TEKNIK UNISMUH MAKASSAR”. Shalawat serta Salam senantiasa penulis panjatkan kepada Nabi Muhammad SAW , yang telah membawa kita dari Zaman jahiliah menuju Zaman yang serba modern seperti ini.

Penulis menyadari bahwa didalam penulisan proposal bukanlah tujuan akhir dari suatu pembelajaran, penulis juga menyadari masih jauh dari kata sempurna baik dari segi isi, bahasa maupun dari segi penulisannya, hal ini disebabkan keterbatasan penulis dari segi pengetahuan. Untuk itu penulis mohon Maaf atas segala kekurangan tersebut. Dan tidak menutup kemungkinan untuk segala saran dan kritik serta masukan yang bersifat membangun bagi diri penulis.

Proposal tugas akhir ini dapat terwujud berkat adanya bantuan, arahan dan kerendahan diri berbagai pihak. Oleh karena itu, dengan segala ketulusan dan kerendahan hati, penulis mengucapkan terimah kasih kepada :

1. Kepada Kedua Orang tua penulis dan keluarga yang telah memberikan wejangan dan motivasi baik secara moril maupun materil.
2. Ibu Dr. Nurnawaty, S.T., M.T. IPM. Sebagai Dekan Fakultas Teknik Universitas Muhammadiyah Makassar.
3. Pak Muhyiddin M Hayat, S.Kom., M.T sebagai Ketua Prodi Informatika dan juga sebagai Dosen Pembimbing 1 Proposal
4. Ibu Rizki Yusliana Bakti, S.T.,M.T, selaku Dosen Pembimbing 2 Proposal
5. Segenap Bapak-bapak dan Ibu Dosen Prodi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar yang telah memberikan bakat dan ilmu pengetahuan serta mendidik penulis selama proses belajar mengajar di Universitas Muhammadiyah Makassar.
6. Teman-teman serta semua pihak yang telah membantu.

Semoga semua pihak tersebut di atas mendapat pahala yang berlipat ganda di sisi Allah SWT dan Proposal Tugas Akhir yang sederhana ini dapat bermanfaat bagi penulis, rekan-rekan, Masyarakat serta bangsa dan Negara. Aamiin

billahi fii sabililhaq fastabiqul khairat.

Wassalamu'alaikum Wr.Wb.

Makassar, 22 Mei 2024

Rianita Kamsurya



DAFTAR ISI

ABSTRAK	ii
<i>ABSTRACT</i>	iii
KATA PENGANTAR	iv
DAFTAR ISI.....	vi
DAFTAR GAMBAR	viii
DAFTAR TABEL.....	ix
DAFTAR LAMPIRAN.....	x
DAFTAR ISTILAH	xi
BAB I PENDAHULUAN	1
A. Latar Belakang Masalah	1
B. Rumusan Masalah.....	2
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 Pikir	19
BAB III METODE PENELITIAN.....	20
A. Tempat dan Waktu penelitian.....	20
B. Alat dan Bahan	20
C. Perancangan Sistem	20
D. Teknik Pengujian Sistem	24
E. Teknik Analisis Data.....	24
BAB IV HASIL DAN PEMBAHASAN	26
A. Deskripsi Dataset.....	26
B. Analisis Data Mentah	26
C. Data Preprocessing	28
D. Pembagian Data.....	41
E. Klasifikasi Menggunakan Algoritma <i>Naïve Bayes</i>	42

F. Pengujian Menggunakan <i>Cross Validation</i>	42
G. Evaluasi Hasil Klasifikasi.....	43
BAB V PENUTUP.....	46
A. Kesimpulan.....	46
B. Saran	46
DAFTAR PUSTAKA	47
LAMPIRAN.....	49



DAFTAR GAMBAR

Gambar 1. Kerangka Pikir	19
Gambar 2. <i>Flowchart</i> Perancangan Sistem	21
Gambar 3. <i>Flowchart Naïve Bayes</i>	23
Gambar 4. Data Mentah	28
Gambar 5. Codingan IPK Terakhir	29
Gambar 6. Codingan SKS Lulus	29
Gambar 7. Codingan Semester	30
Gambar 8. Hasil Penambahan Atribut	30
Gambar 9. Codingan Atribut Target	31
Gambar 10. Codingan dan Hasil Kategorisasi	32
Gambar 11. Codingan dan Hasil Seleksi Atribut	33
Gambar 12. Codingan dan Hasil Penghapusan Atribut	33
Gambar 13. Data <i>Missing Value</i>	35
Gambar 14 Codingan Transformation Data	42
Gambar 15. Codingan dan Hasil Pembagian Data	42
Gambar 16. Confussion Matrix pada Algoritma Naïve Bayes	44
Gambar 17. Hasil Pengukuran Kinerja Algoritma Naïve Bayes	45

DAFTAR TABEL

Tabel 1. Confussion matrix	11
Tabel 2. Simbol <i>Flowchart</i>	13
Tabel 3. Penelitian Terkait	14
Tabel 4. Kategori pada atribut ipk terakhir	31
Tabel 5. Kategori pada atribut sks lulus.....	32
Tabel 6. Jumlah Data <i>Missing Value</i>	34
Tabel 7. Jumlah Data <i>Missing Value</i> Setelah menggunakan Data Valid.....	35
Tabel 8. Jumlah Data Hasil Proses Pembersihan	36
Tabel 9. Pelabelan Pada Atribut ayah_pekerjaan.....	38
Tabel 10. Pelabelan Pada Atribut ayah_penghasilan	38
Tabel 11. Pelabelan Pada Atribut ayah_status	39
Tabel 12. Pelabelan Pada Atribut ibu_pekerjaan	39
Tabel 13. Pelabelan Pada Atribut ibu_penghasilan.....	39
Tabel 14. Pelabelan Pada Atribut ibu_status.....	39
Tabel 15. Pelabelan Pada Atribut wali_pekerjaan	40
Tabel 16. Pelabelan Pada Atribut wali_penghasilan.....	40
Tabel 17. Pelabelan Pada Atribut lulus.....	40
Tabel 18. Pelabelan Pada Atribut kategori_IPK	40
Tabel 19. Pelabelan Pada Atribut kategori_SKS	41
Tabel 20. Pelabelan Pada Atribut status_putus_studi	41
Tabel 21. Pembagian Data	41
Tabel 22. Hasil Akurasi.....	42
Tabel 23. Hasil pengujian <i>10-fold cross validation</i>	43
Tabel 24. Confussion Matrix Pada Algoritma Naive Bayes	45

DAFTAR LAMPIRAN

Lampiran 1. Data Mentah	50
Lampiran 2. Data Preprocessing.....	51
Lampiran 3. Transformasi Data	52
Lampiran 4. Codingan Dan Output.....	53
Lampiran 5. Permohonan Penelitian.....	83
Lampiran 6. Pengantar Penelitian.....	84
Lampiran 7. Surat Keterangan Bebas Plagiat.....	85
Lampiran 8. Hasil Plagiat Per BAB	86



DAFTAR ISTILAH

Klasifikasi	Proses pengelompokan atau pengkategorian data berdasarkan karakteristik tertentu.
Algoritma Naive Bayes	Algoritma Naive Bayes adalah salah satu metode klasifikasi dalam <i>machine learning</i> yang berbasis pada Teorema Bayes. Algoritma ini digunakan untuk memprediksi kelas dari sebuah objek berdasarkan probabilitas dari setiap fitur yang dimiliki objek tersebut.
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.
Precision	Ukuran seberapa banyak hasil prediksi yang relevan dari total hasil yang diprediksi sebagai positif.
Recall	Ukuran seberapa baik model menemukan semua data relevan dari keseluruhan data yang seharusnya diprediksi sebagai positif.
f1-score	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.
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 Masalah

Banyaknya mahasiswa yang putus studi adalah masalah yang saat ini dihadapi oleh banyak institusi pendidikan tinggi di berbagai negara. Mahasiswa yang putus studi dapat menghambat pertumbuhan ekonomi, daya saing, dan produktivitas, yang berdampak pada siswa dan masyarakat secara keseluruhan (Realinho, Machado, Baptista, & Martins, 2022). (Wijyaningrum et al., 2019). Permasalahan mahasiswa putus studi di Universitas Muhammadiyah Makassar merupakan isu penting dan kompleks yang dipengaruhi oleh berbagai faktor, baik internal maupun eksternal. Aspek internal meliputi rendahnya motivasi belajar, kesulitan akademik dan masalah kesehatan mental, sedangkan faktor eksternal mencakup masalah keuangan, kondisi keluarga yang tidak mendukung, lingkungan sosial yang tidak kondusif, dan tekanan untuk bekerja daripada melanjutkan studi. Penentuan mahasiswa yang putus studi saat ini masih dilakukan secara manual yang tidak hanya memakan waktu dan tenaga tetapi juga rentan terhadap kesalahan dan kurang efisien dalam mengidentifikasi mahasiswa yang berisiko. Proses manual ini sering kali tidak mampu memberikan gambaran akurat tentang berbagai faktor penentu. Untuk mengatasi masalah ini diperlukan pendekatan yang lebih sistematis dan berbasis data. Salah satu solusi yang diusulkan adalah melakukan pengembangan sistem klasifikasi tingkay putus studi mahasiswa.

Klasifikasi adalah proses dalam penggalian data yang bertujuan untuk menemukan model atau fungsi yang mendeskripsikan data dan membedakan data ke dalam kelas-kelas. Proses ini melibatkan memeriksa karakteristik objek untuk memastikan bahwa objek tersebut dimasukkan ke dalam salah satu kelas yang sudah didefinisikan sebelumnya. Praproses data adalah proses persiapan data mentah sebelum dilakukan proses klasifikasi data (Salmawati, Yuyun, 2021).

Salah satu algoritma yang digunakan untuk klasifikasi data adalah Algoritma Naive bayes. Naive bayes merupakan salah satu metode pembelajaran mesin yang memanfaatkan perhitungan probabilitas dan statistik yang dikemukakan oleh ilmuwan Inggris Thomas Bayes, yaitu memprediksi probabilitas dimasa depan berdasarkan pengalaman di masa sebelumnya (Samasil et al., 2022). Terdapat penelitian terdahulu yang menggunakan perbandingan metode Deep Learning, Naive Bayes, dan Random Forest untuk menerapkan data mining pada penerimaan mahasiswa baru di Universitas Islam Negeri Raden Fatah. Hasil menunjukkan bahwa metode Naive Bayes menghasilkan akurasi sebesar 99.79%, jauh lebih tinggi daripada metode lainnya, seperti Deep Learning (52,65%) dan Random Forest (44,65%). Algoritma Naive Bayes juga dianggap lebih akurat dan sering menjadi pilihan terbaik. (Ulinuha & Fanani, 2022).

Berdasarkan latar belakang diatas, penelitian ini akan fokus pada pengembangan sistem klasifikasi tingkat putus studi mahasiswa di UNISMUH Makassar dengan menggunakan algoritma Naive bayes. Penelitian ini diharapkan dapat memberikan kontribusi signifikan dalam upaya peningkatan kualitas pendidikan dan memberikan kontribusi signifikan dalam upaya peningkatan kualitas pendidikan dan pengurangan angka putus studi.

B. Rumusan Masalah

Dengan mempertimbangkan hasil penelitian di atas, peneliti dapat membuat kesimpulan tentang masalah berikut:

1. Bagaimana penggunaan algoritma Naive bayes untuk mengklasifikasi mahasiswa putus studi ?
2. Bagaimana tingkat akurasi dari sistem klasifikasi yang menggunakan algoritma Naive bayes dalam mendeteksi mahasiswa yang berpotensi putus studi ?

C. Tujuan Penelitian

Dengan mempertimbangkan latar belakang dan rumusan masalah di atas, peneliti dapat menetapkan tujuan penelitian berikut:

1. Untuk mengetahui bagaimana penggunaan algoritma Naive bayes untuk mengklasifikasi mahasiswa putus studi.
2. Untuk mengetahui tingkat akurasi sistem klasifikasi yang menggunakan algoritma Naive bayes untuk mendeteksi mahasiswa putus studi.

D. Manfaat Penelitian

Penelitian tentang klasifikasi mahasiswa yang berpotensi putus studi di fakultas teknik Unismuh Makassar akan memberikan manfaat berikut:

1. Manfaat bagi peneliti
 - a. Penelitian ini dapat membantu peneliti untuk meningkatkan pemahaman pada pengetahuan tentang algoritma Naive bayes
 - b. Penelitian ini dapat membantu memecahkan masalah yang berkaitan dengan putus studi mahasiswa.
2. Manfaat bagi dosen
 - a. Sistem prediksi yang dihasilkan akan memudahkan dosen Pembimbing Akademik dalam melakukan identifikasi dini terhadap mahasiswa yang berpotensi mengalami masalah akademik.
 - b. Dengan informasi yang akurat mengenai potensi putus studi, dosen pembimbing akademik dapat memberikan saran yang lebih spesifik dan efektif dalam sesi pembimbingan akademik, membantu mahasiswa untuk mengelola beban studi mereka dengan lebih baik.
3. Manfaat bagi mahasiswa
 - a. Dengan intervensi yang tepat waktu dan dukungan yang dipersonalisasi, mahasiswa memiliki kesempatan yang lebih besar untuk memperbaiki IPK dan IPS, yang pada gilirannya akan meningkatkan kinerja akademik secara

- b. Mahasiswa yang menerima dukungan berdasarkan hasil prediksi memiliki kemungkinan yang lebih tinggi untuk menyelesaikan studi mereka tepat waktu, mengurangi resiko putus studi dan mencapai keberhasilan akademik.

E. Ruang Lingkup Penelitian

Ruang lingkup penelitian adalah istilah yang digunakan untuk mendefinisikan "fokus penelitian". Ruang lingkup penelitian juga digunakan untuk menentukan wilayah yang akan dipelajari serta batas-batas objek atau fenomena yang diamati.

1. Penelitian ini membutuhkan data mengenai mahasiswa, seperti Jenis kelamin, IPK, SKS, Usia dan Status Orang tua serta status mahasiswa sebagai Target kelas. 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 Naive Bayes* untuk mengolah data yang telah dikumpulkan.

F. Sistematika Penulisan

Sistematika tulisan yang digunakan selama proses penyusunan laporan proposal penelitian ini adalah sebagai berikut:

BAB I PENDAHULUAN

Bab ini membahas latar belakang, rumusan masalah, tujuan, manfaat, dan ruang lingkup penelitian.

BAB II TINJAUAN PUSTAKA

Bab ini membahas studi pustaka tentang teori penelitian dan elemen pendukung penelitian..

BAB III METODOLOGI PENELITIAN

Bab ini membahas semua aspek pengidentifikasian masalah, pengumpulan data, pengoptimalan algoritma, dan penggunaan 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 merupakan salah satu metode data mining yang paling populer. Klasifikasi sendiri merupakan proses data mining yang bertujuan menemukan model untuk memprediksi tingkah laku konsumen dimasa yang akan datang berdasarkan pengklasifikasian rekaman dari database yang kemudian akan dijelaskan atau dibedakan kelas-kelasnya pada setiap data. Klasifikasi sendiri adalah sebuah proses mengumpulkan data yang memiliki kesamaan menurut peneliti yang telah ditentukan sebelum dianalisis. Contohnya, untuk sebuah perusahaan menilai tingkat kepuasan pelanggannya dilakukan sebuah analisis berdasarkan faktor-faktor yang dinilai berkaitan serta dapat dimodelkan untuk memprediksi kepuasan pelanggannya agar dimasa yang akan datang perusahaan dapat memprediksi sistem-sistem yang dinilai masih kurang memuaskan bagi para *customer* (Arum et al., 2022).

Dalam proses pembelajaran mesin, klasifikasi adalah tugas memprediksi label kelas dari sampel yang diberikan berdasarkan sekumpulan fitur atau karakteristik. Misalnya, diberikan data tentang usia, jenis kelamin dan pendapatan seseorang, tujuan tugas klasifikasi mungkin untuk memprediksi apakah mereka cenderung memiliki pekerjaan bergaji tinggi atau tidak (Syahril Dwi Prasetyo et al., 2022)

Dalam penelitian ini, pengklasifikasian mengacu pada proses menggunakan algoritma Naive Bayes untuk memprediksi apakah seorang mahasiswa memiliki potensi untuk putus studi berdasarkan data yang dianalisis. Proses ini melibatkan beberapa langkah. Pertama, data mahasiswa akan menjalani serangkaian praproses untuk mempersiapkannya sebelum dilakukan analisis lebih lanjut, termasuk pembersihan data, penghilangan data yang tidak relevan atau hilang, pengkodean variabel, dan normalisasi data. Selanjutnya atribut yang dianggap penting dalam

memprediksi potensi mahasiswa putus studi dipilih dalam model klasifikasi dan dijadikan atribut signifikan. Kemudian data historis mahasiswa yang telah diproses akan digunakan untuk melatih model klasifikasi Naive bayes, dimana proses pelatihan ini akan memungkinkan model untuk belajar pola dari data dan membangun fungsi probabilitas yang membedakan mahasiswa putus studi dan yang tidak. Setelah model dilatih, langkah berikutnya adalah mengevaluasi kinerjanya menggunakan data yang terpisah (misalnya data uji) untuk mengukur seberapa baik model prediksi label kelas yang benar. Kemudian model akan diuji pada data yang belum pernah dilihat sebelumnya untuk memeriksa kinerjanya secara keseluruhan dan akan memberikan pemahaman yang lebih baik tentang seberapa baik model dapat digunakan untuk memprediksi potensi mahasiswa putus studi.

2. Mahasiswa berpotensi Putus studi

Pendidikan adalah pilar demokrasi di banyak negara dan sangat penting untuk kemajuan suatu negara. Setiap warga negara harus setidaknya mendapatkan pendidikan dasar lebih lanjut jika mereka ingin hidup makmur di masa mendatang. Selain itu, setiap warga negara harus mendapatkan pendidikan tinggi untuk memperoleh kemampuan khusus dan keterampilan umum lainnya. Jumlah mahasiswa yang putus sekolah adalah masalah yang sedang dihadapi oleh banyak institusi pendidikan tinggi di berbagai negara. Mahasiswa yang putus sekolah dapat menghambat kemajuan ekonomi, daya saing, dan produktivitas, yang berdampak pada siswa dan masyarakat secara keseluruhan. (Wijyaningrum et al., 2019).

Dalam kamus Bahasa Indonesia, potensi adalah kemampuan yang mempunyai kemungkinan untuk dikembangkan. Berkaitan dengan keberhasilan studi di perguruan tinggi, kemampuan yang mungkin dikembangkan adalah kemampuan di perguruan tinggi, diperlukan suatu batas minimal kemampuan dasar akademik sehingga seorang mahasiswa diyakini akan mampu menyelesaikan studi di perguruan tinggi. Secara umum ukuran keberhasilan studi adalah ketika mahasiswa mampu

menyelesaikan program pendidikan sesuai kecepatan belajar masing-masing dengan tidak melebihi ketentuan batas waktu yang ditetapkan perguruan tinggi. Dalam rangka memaksimalkan tugas perguruan tinggi dalam mengembangkan potensi untuk keberhasilan studi, identifikasi dini atas potensi mahasiswa perlu dilakukan guna mengetahui kemungkinan keberhasilan mahasiswa dalam penyelesaian studi dan mengantisipasi kemungkinan ketidakberhasilan. Penting untuk memprediksi, apakah seorang mahasiswa yang masuk perguruan tinggi akan dapat keluar atau menyelesaikan studi (Yuniarti et al., 2020).

Dalam penelitian ini, mahasiswa putus studi merujuk kepada mahasiswa yang menghentikan atau tidak menyelesaikan studi mereka di Fakultas Teknik Universitas Muhammadiyah Makassar dalam rentang waktu antara tahun 2013-2015. Data mahasiswa yang digunakan dalam penelitian ini meliputi angkatan yang terdaftar di fakultas tersebut selama periode tersebut. Mahasiswa yang diidentifikasi sebagai putus studi adalah mereka yang tidak berhasil menyelesaikan program studi mereka dalam waktu yang ditentukan atau dianggap sebagai risiko putus studi.

Penelitian ini akan menganalisis data historis dari mahasiswa-mahasiswa tersebut untuk mengidentifikasi pola-pola atau karakteristik yang mungkin mempengaruhi kecenderungan putus studi. Dengan demikian, penelitian ini akan membantu dalam pengembangan sistem klasifikasi yang dapat memprediksi potensi mahasiswa putus studi.

3. Algoritma Naive Bayes

Berdasarkan teorema Bayes, metode klasifikasi Naive Bayes menggunakan metode probabilitas dan statistik dan berusaha memprediksi peluang masa depan berdasarkan pengalaman masa lalu. Metode ini dikenal sebagai Teorema Bayes. Ciri utama dari Naive Bayes ini adalah keyakinan yang kuat, atau naif, bahwa setiap situasi atau peristiwa berdiri sendiri. Naive Bayes menghitung probabilitas dengan asumsi bahwa kelas keputusan adalah benar, menurut Oslon delen (2008). mengingat

vektor data objek. Algoritma ini menganggap bahwa atribut objek berbeda satu sama lain. Jumlah frekuensi dari tabel keputusan "master" dapat dihitung untuk menentukan kemungkinan yang lebih tepat untuk menghasilkan perkiraan akhir. Naive Bayes menunjukkan kinerja yang luar biasa dibandingkan dengan model classifier lainnya. Menurut Xhemali, Hinde Stone, "Naive Bayes vs. Decision Tree vs. Neural Networks in the classification of training Web Pages", "Naive Bayes memiliki tingkat akurasi yang lebih baik dibandingkan dengan model Classifier lainnya." Keuntungan penggunaan metode ini adalah bahwa estimasi parameter yang diperlukan dalam proses pengklasifikasian dapat dibuat dengan menggunakan jumlah data pelatihan yang kecil. Karena dianggap sebagai variabel independen, hanya varians suatu variable dalam kelas yang diperlukan untuk menentukan klasifikasi, bukan matriks kovarians secara keseluruhan. (Rayuwati et al., 2022).

Naive bayes adalah salah satu metode yang digunakan untuk mengklasifikasikan data. *Bayesian classification* merupakan pengklasifikasian statistik yang dapat digunakan untuk memprediksi probabilitas keanggotaan suatu class. *Naive bayes* dianggap sebagai penyederhanaan nilai atribut secara kondisional saling bebas jika diberi nilai output. Keuntungan menggunakan *Naive bayes* adalah metode ini hanya membutuhkan dua data yaitu data latih dan data uji untuk menguji data yang diinginkan. *Naive bayes* sering bekerja jauh lebih baik dari yang diharapkan dalam situasi dunia nyata yang paling kompleks (Zarti et al., 2022).

Berikut persamaan *Teorema Bayes* :

$$P(H|X) = \frac{P(H) \cdot P(X|H)}{p(H)} \dots\dots\dots (1)$$

Keterangan :

X = Data dengan class yang belum diketahui

H = Hipotesis data X merupakan suatu class spesifik

P(H|X) = Probabilitas hipotesis H berdasarkan kondisi X (*posteriori prob*)

$P(H)$ = Probabilitas hipotesis H (*prior prob*)

$P(X|H)$ = Probabilitas X berdasarkan kondisi tersebut

$P(X)$ = Probabilitas dari X

4. Cross Validation

Metode atau algoritma cross-validation adalah tindakan pembuktian. Ini dilakukan dengan membagi data menjadi data pelatihan dan data tes dengan komposisi yang berbeda. Pembagian data acak menjadi sepuluh bagian, dengan satu bagian untuk pemeriksaan data dan sembilan bagian lainnya untuk pelatihan data, adalah pembagian yang paling sering digunakan dalam penelitian klasifikasi data mining. Salah satu nama validasi ini adalah k-fold Cross Validation, dengan $k=10$. Menurut penelitian Mulajati (2017), validasi ini digunakan untuk menentukan rata-rata keberhasilan sistem dengan melakukan perulangan dengan atribut masukan sehingga sistem diuji untuk beberapa atribut masukan yang acak. Setiap kelas dalam kelompok data harus diwakili secara proporsional antara data pelatihan dan data pengujian. Data didistribusikan secara acak ke dalam masing-masing kelas dengan menggunakan metrik yang sama untuk perbandingan. Untuk mengurangi bias yang disebabkan oleh sampel tertentu, proses pelatihan dan pengujian diulangi berulang kali. Rata-rata dari masing-masing sampel akan dihitung untuk menghasilkan tingkat kesalahan total. Model terbaik memiliki kesalahan terkecil. (Salmawati, Yuyun, 2021).

5. Confusion matrix

Confusion matrix adalah hasil evaluasi klasifikasi data mining yang ditampilkan dalam sebuah tabel dan berisi perhitungan jumlah objek pengujian data yang diproyeksikan masuk ke dalam sebuah kelas dengan klasifikasi yang sebenarnya. Confusion matrix menghasilkan pengukuran kinerja klasifikasi dalam bentuk nilai ketepatan dan nilai precision dan recall. Nilai ketepatan dan recall ini berkisar antara 0-1. Semakin besar nilai yang diterima, semakin baik (Salmawati, Yuyun, 2021).

- a. *Accuracy* adalah gambaran ketepatan daftar data yang diidentifikasi benar.
- b. *Recall* adalah jumlah kasus positif yang benar-benar ditemukan.
- c. *Precision* adalah jumlah kasus yang ditemukan positif dengan benar.

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

Tabel 1. Confussion 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 *confussion 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):

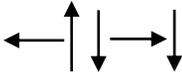
$$F1 - Score = 2 \times \frac{Precision \times recall}{Precision + Recall} \dots\dots\dots(4)$$

6. Flowchart

Bagan atau Flowchart adalah diagram yang menunjukkan langkah-langkah dan keputusan dalam menyelesaikan suatu proses pemrograman. Setiap langkah dipresentasikan dalam bentuk diagram dan hubungkan dengan garis atau panah. Diagram alir memainkan peran penting dalam menentukan fase atau fungsionalitas proyek perangkat lunak . selain itu, penggunaan diagram alir proses program lebih jelas, ringkas dan mengurangi kemungkinan salah tafsir. Namun flowchart juga mempunyai beberapa kelemahan seperti tidak menggambarkan detail yang sangat kompleks dan tidak fleksibel dalam mempresentasikan variasi dalam sebuah proses. Oleh karenanya penggunaan sebuah flowchart harus dilakukan dengan bijak dan dipertimbangkan dengan cermat.

Flowchart atau sering disebut dengan diagram alir merupakan suatu jenis diagram yang mempresentasikan algoritma atau langkah-langkah instruksi yang berurutan dalam sistem. Seorang analis sistem menggunakan flowchart sebagai bukti dokumentasi untuk menjelaskan gambaran logis sebuah sistem yang akan dibangun kepada programmer. Dengan begitu, Flowchart dapat membantu untuk memberikan solusi terhadap masalah yang bisa saja terjadi dalam membangun sistem. Pada dasarnya, flowchart digambarkan dengan menggunakan simbol-simbol. Setiap simbol mewakili suatu proses tertentu. Sedangkan untuk menghubungkan satu proses ke proses selanjutnya digambarkan dengan menggunakan garis penghubung (Rosaly & Prasetyo, 2020).

Tabel 2. Simbol *Flowchart*

Simbol	Arti simbol
	flow direction symbol atau disebut juga <i>connecting line</i> adalah simbol yang berfungsi untuk menghubungkan antara satu simbol dengan simbol lainnya. Garis panah ini juga mewakili arah aliran proses
	Terminator symbol digunakan sebagai simbol diawal (<i>start</i>) atau akhir (<i>stop</i>) dari sebuah program. Bentuk oval ini menggambarkan awal dan akhir dari sebuah algoritma.
	Connector symbol adalah simbol yang berfungsi untuk keluar dan masuk menyambungkan proses didalam lembar/halaman yang sama.
	Connector symbol adalah simbol yang berfungsi untuk keluar dan masuk menyambungkan proses didalam lembar/halaman yang berbeda.
	Processing symbol adalah simbol yang menunjukkan pengolahan yang dilakukan oleh komputer. Di dalam akan disini dengan tindakan utama dari algoritma atau logika utama program.
	Manual operation symbol adalah simbol yang menunjukkan pengolahan yang tidak dilakukan oleh komputer.
	Decision symbol adalah simbol untuk memilih proses berdasarkan kondisi tertentu yang menghasilkan kemungkinan dua jawaban ya atau tidak.
	Input-output symbol adalah simbol yang berfungsi untuk menyatakan proses <i>input</i> dan <i>output</i> tanpa bergantung dengan jenis peralatannya.
	Manual input symbol adalah simbol yang berfungsi untuk memasukkan data secara manual <i>online keyboard</i> .
	Preparation symbol adalah simbol yang digunakan untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam <i>storage/penyimpanan</i> .

B. Penelitian Terkait

Peneliti memberikan segudang inspirasi dan informasi untuk penyempurnaan proposal ini dari penelitian sebelumnya. Penelitian yang terkait sebelumnya meliputi :

Tabel 3. Penelitian Terkait

Peneliti	Tujuan/Kasus	Metode/Algoritma	Hasil Penelitian
(Nuralia et al., 2022)	Tujuan dari penelitian ini adalah untuk mengetahui seberapa akurat Naive Bayes Classifier melakukan prediksi terhadap studi mahasiswa yang berlangsung lama. Penelitian ini akan membagi lima ratus dataset menjadi tiga skenario pengujian yang berbeda untuk mengetahui tingkat akurasi tersebut. Skenario pertama akan membandingkan data pelatihan dengan 80 hingga 20 persen, skenario kedua akan membandingkan dengan 50 hingga 50 persen, dan skenario ketiga akan membandingkan dengan 20 hingga 80 persen.	<i>Naive bayes Classifier</i>	Senario i memiliki kemampuan untuk menghasilkan nilai akurasi tertinggi dengan skor f1- di atas 90%, menurut hasil pengujian skenario tiga.

Selanjutnya, analisis hasil pengujian akan dilakukan pada matriks kekacauan.

(Andreans yach et al., 2022)	Salah satu topik masalah yang menjadi bahan diskusi terkait kegagalan mahasiswa sarjana ialah tentang tidak tepat waktu. Mahasiswa yang lulus pada waktu yang tidak tepat waktu tentu akan menimbulkan masalah dan kemungkinan putus sekolah, yang juga akan mempengaruhi kualitas pendidikan dan akreditasi.	<i>Algoritma naive Bayes</i>	Mengetahui seberapa banyak mahasiswa yang lulus tidak sesuai dengan masa studi yang sudah ditentukan dengan metode naive bayes merupakan suatu tujuan dari penelitian ini. Hasil dari penelitian ini berupa sistem yang dapat memprediksi ketepatan waktu penyelesaian.
(Yuniarti et al., 2020)	Tujuan penelitian ini adalah untuk memprediksi keberhasilan metode klasifikasi naive bayes.	<i>Naive Bayes Classifier</i>	Dalam klasifikasi potensi keberhasilan studi ini, metode klasifikasi naive bayes memberikan akurasi 95,8% hingga 99,41% untuk empat tujuan.

			Perguruan tinggi harus memperbaiki proses perekrutan siswa dan memahami bahwa uang orang tua memengaruhi kuliah.
(Mulyadi & Juniadi, 2019)	Membuat model prediksi keaktifan mahasiswa pada semester berikutnya dengan teknik data mining menggunakan <i>Algoritma Naive bayes</i>	<i>Algoritma Naive bayes</i>	Hasilnya model tersebut mempunyai tingkat akurasi sebesar 77,2%, nilai presisi sebesar 82,1% dan nilai recall sebesar 92,2%.
(Mahangga & Laksito, 2019)	Menghitung sekumpulan probabilitas dari jumlah frekuensi dan menggabungkan nilai dari dataset untuk membuat prediksi pengunduran diri siswa.	<i>Algoritma Naive bayes</i>	Hasil uji coba dari 120 dataset yang dibagi menjadi 70% data pelatihan dan 30% data pengujian menunjukkan nilai kesalahan 22,22% dan tingkat akurasi 77,78% dengan hasil prediksi 22 mahasiswa

			diprediksi bertahan dan 14 mahasiswa mengundurkan diri.
(Kamila & Subastian, 2019)	Penelitian ini membahas bagaimana perbandingan KNN dan Naive Bayes dalam memprediksi potensi putus kuliah mahasiswa.	<i>Algoritma KNN dan Naive Bayes</i>	Nilai akurasi yang dihasilkan dari penelitian ini menunjukkan Naive Bayes (92%) lebih unggul dalam memprediksi status putus kuliah mahasiswa dibandingkan dengan algoritma KNN (85%).

Penelitian pertama oleh (Nuralia et al., 2022) “ Implementasi Naive Bayes Classifier dalam memprediksi kelulusan mahasiswa” tujuan dari penelitian ini adalah Untuk mengetahui tingkat akurasi tersebut, penelitian ini membagi lima ratus dataset menjadi tiga skenario pengujian. Skenario I menggunakan perbandingan data pelatihan dan pengujian 80:20, skenario II menggunakan perbandingan 50:50, dan skenario III menggunakan perbandingan 20:80. Dengan menggunakan confusion matrix, analisis hasil pengujian dilakukan. Hasilnya menunjukkan bahwa skenario I memiliki kemampuan untuk menghasilkan nilai akurasi tertinggi dengan nilai f1, dan skor yang dihasilkannya juga lebih dari 90%.

Penelitian kedua oleh (Andreansyach et al., 2022) “ Prediksi persentase kelulusan Mahasiswa STMIK Widuri menggunakan Algoritma Naive bayes” salah satu topik yang masih menjadi bahan diskusi terkait kegagalan mahasiswa sarjana

adalah kelulusan yang tidak tepat waktu, yang dapat menimbulkan masalah dan kemungkinan putus sekolah, serta mempengaruhi kualitas pendidikan dan akreditasi. Mengetahui seberapa banyak mahasiswa yang lulus tidak sesuai dengan masa studi yang sudah ditentukan menggunakan metode Naive bayes merupakan tujuan dari penelitian ini. Hasil dari penelitian ini adalah sebuah sistem yang dapat memprediksi ketepatan waktu penyelesaian studi.

Penelitian ketiga oleh (Yuniarti et al., 2020) “identifikasi potensi keberhasilan Studi menggunakan Naive bayes Classifier” penelitian ini bertujuan melakukan prediksi keberhasilan studi dengan menggunakan metode Naive bayes. Dalam potensi keberhasilan studi ini, metode Naive Bayes Classifier memberikan akurasi antara 95,8% dan 99,41% untuk empat target.

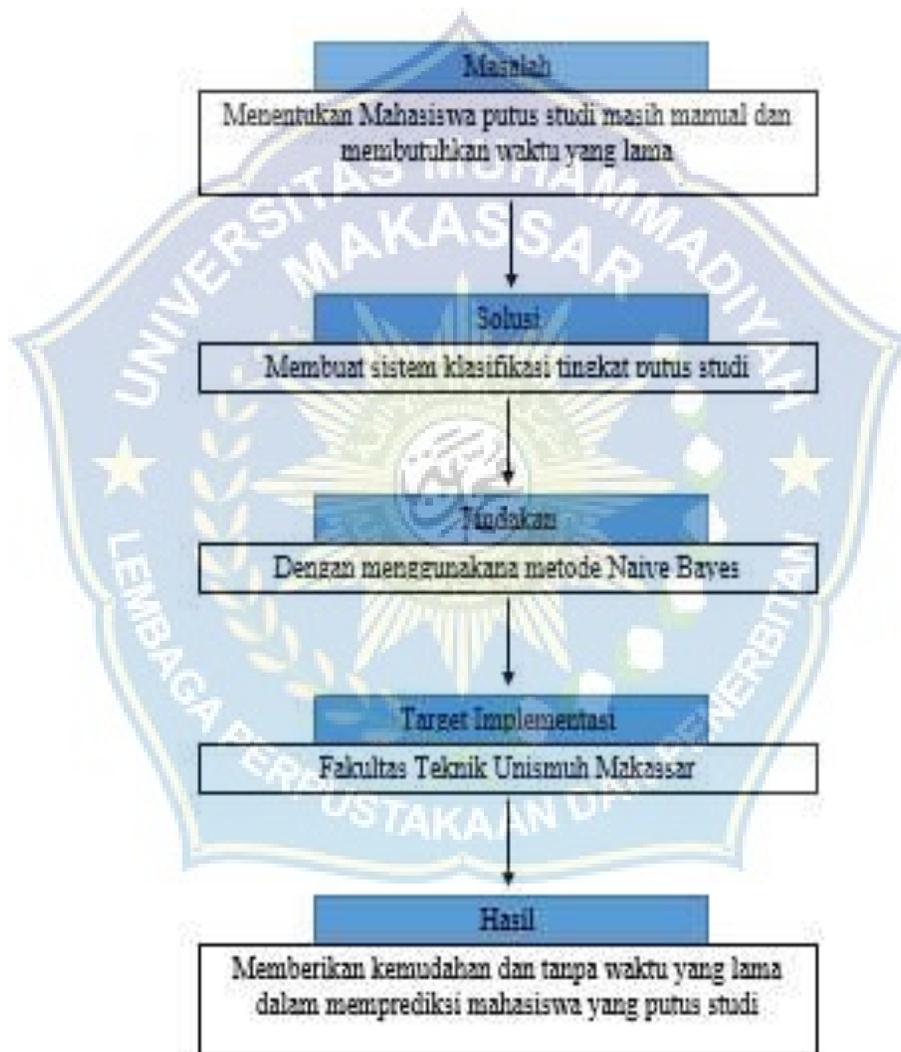
Penelitian keempat oleh (Mulyadi & Juniadi, 2019) “Prediksi keaktifan studi mahasiswa baru dengan Algoritma Naive Bayes” Membuat model prediksi keaktifan mahasiswa pada semester berikutnya dengan teknik data mining menggunakan algoritma Naive bayes menghasilkan model dengan tingkat akurasi sebesar 77.2%, nilai presisi sebesar 82.1% dan nilai reccal sebesar 92.2%.

Penelitian kelima oleh (Mahanggara & Laksito, 2019) “Prediksi pengunduran diri mahasiswa Universitas Amikom Yogyakarta menggunakan Metode Naive bayes” Melakukan menggunakan Algoritma Naive Bayes untuk menghitung sekumpulan probabilitas dari jumlah frekuensi dan menggabungkan nilai dari 120 dataset. Hasil uji coba menunjukkan nilai kesalahan 22,22% dan tingkat akurasi 77,78%, dengan perkiraan bahwa 22 siswa akan bertahan dan 14 siswa akan mengundurkan diri.

Penelitian keenam oleh (Kamila & Subastian, 2019) “KNN vs Naive Bayes untuk deteksi dini putus kuliah pada profil akademik mahasiswa” penelitian ini membahas perbandingan KNN dan Naive Bayes dalam memprediksi potensi putus kuliah pada mahasiswa. Algoritma KNN dan Naive Bayes diterapkan menggunakan *Lybrary Scikit-learn* pada *Python*, dan hasil penelitian ini diharapkan menjadi

pedoman dalam menerapkan algoritma ke dalam sistem deteksi dini putus kuliah. Nilai akurasi yang dihasilkan dari penelitian ini menunjukkan Naive Bayes (92%) lebih unggul dalam memprediksi status putus kuliah mahasiswa dibandingkan dengan algoritma KNN (85%).

C. Kerangka Pikir



Gambar 1 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

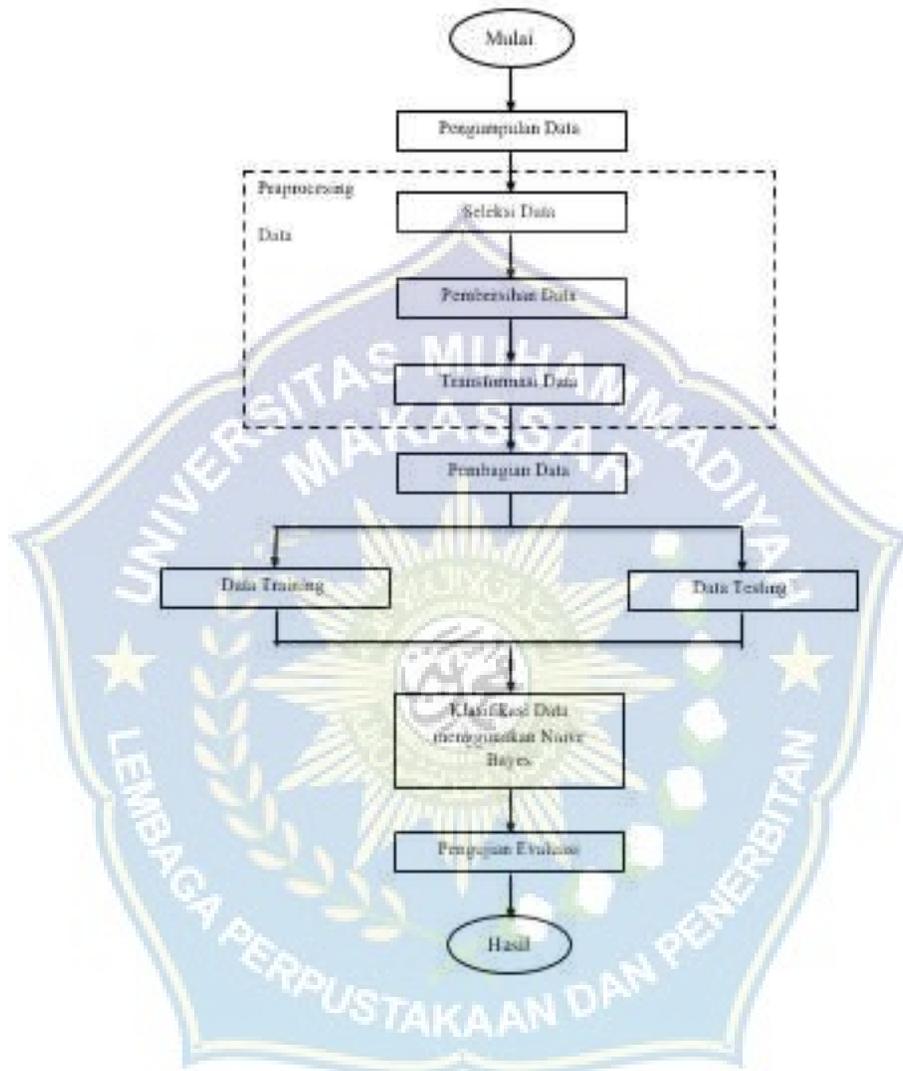
B. Alat dan Bahan

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

1. Kebutuhan *Hardware* (Perangkat Keras)
 - a. Laptop Lenovo idea pad 130
 - b. Ram 4.00 GB
 - c. OS Windows 10
 - d. Processor AMD A9-9425 RADEON R5, 5 COMPUTE CORES 2C=3G 3,10 HZ
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 2 Flowchart Perancangan sistem

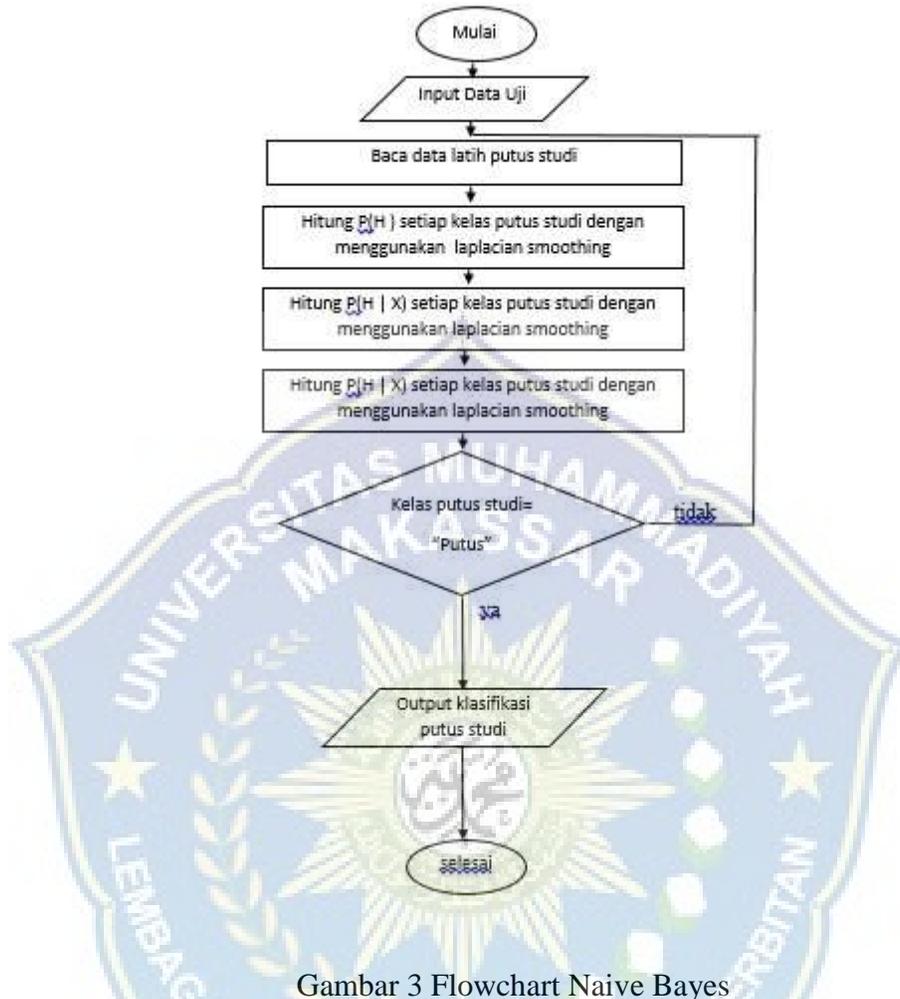
Tahap pengumpulan data adalah langkah awal dalam penelitian ini. Data yang dikumpulkan meliputi informasi tentang Jenis kelamin, IPK, SKS, Usia dan Status Orang tua serta status mahasiswa sebagai Target kelas dari mahasiswa Fakultas Teknik Universitas mahasiswa Muhammadiyah Makassar. data ini diambil dari catatan akademik mahasiswa untuk angkatan 2013-2015.

Tahap praproses data bertujuan untuk mempersiapkan data mentah sebelum dilakukan analisis lebih lanjut, dengan langkah-langkah meliputi seleksi data, yaitu memilih atribut yang paling relevan dan signifikan untuk meningkatkan akurasi model klasifikasi; pembersihan data, yaitu menghilangkan data yang tidak lengkap atau salah; transformasi data, yaitu mengubah data ke dalam format yang sesuai, seperti mengubah nilai-nilai kategori menjadi numerik jika diperlukan

Tahap pembagian data melibatkan pembagian dataset yang telah diproses menjadi data training dan data testing. Data training digunakan untuk melatih model algoritma naive bayes, sehingga model dapat belajar dan memahami pola dari data yang tersedia. kemudian model menggunakan data training untuk menghitung probabilitas dan membangun aturan klasifikasi yang akan digunakan untuk prediksi. Data testing digunakan untuk menguji dan mengevaluasi kinerja model yang telah dilatih. Model yang telah dilatih menggunakan data training yang diaplikasikan pada data testing untuk memprediksi apakah seorang mahasiswa akan lulus tepat waktu atau berpotensi putus studi. Hasil prediksi ini kemudian dibandingkan dengan model sebenarnya pada data testing untuk mengukur akurasi dan kinerja model.

Tahap klasifikasi menggunakan algoritma naive bayes meliputi langkah-langkah seperti training model, dimana model Naive bayes dilatih menggunakan data training dari masing-masing skenario, dan algoritma ini akan menghitung probabilitas berdasarkan frekuensi kemunculan atribut dan class target, serta diprediksi dimana model yang telah dilatih digunakan untuk memprediksi class target dari data testing.

Tahap pengujian menggunakan validation cross dan kemudian data akan dievaluasi menggunakan confusion matrix



Gambar 3 Flowchart Naive Bayes

Proses awal yaitu memasukan data uji kemudian membaca data latih mengenai status putus studi mahasiswa, lalu menghitung probabilitas prior untuk setiap hipotesis “putus” dan “tidak putus” dengan Laplacian smoothing setelah itu menghitung probabilitas likelihood dari atribut untuk setiap hipotesis, kemudian menghitung probabilitas posterior menggunakan teorema Bayes untuk menentukan kelas putus studi, dan menghasilkan output klasifikasi status mahasiswa apakah berpotensi putus studi atau tidak.

D. Teknik Pengujian Sistem

Teknik pengujian yang digunakan adalah cross-validation dan confusion matrix. Cross-validation adalah teknik yang umum digunakan untuk menilai 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 10-fold cross-validation membagi dataset 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 Positives (TP), True Negatives (TN), False Positives (FP), False Negatives (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 subjek penelitian dikumpulkan, salah satu proses penelitian adalah melakukan analisis data. Selama proses penelitian, analisis data tidak dapat diabaikan karena sangat penting untuk menggunakan instrumen analisis dengan benar untuk menentukan hasil yang akurat. Kesalahan dalam menentukan alat analisis dapat berdampak negatif pada kesimpulan yang dicapai dan akan berdampak lebih besar pada penggunaan dan pelaksanaan hasil penelitian.

Secara umum, metode analisis data termasuk dalam dua kategori: analisis kuantitatif dan kualitatif. Perbedaan utama antara kedua metode ini adalah jenis data yang dipelajari: analisis kualitatif (tidak dapat diangkakan) digunakan untuk analisis

data kualitatif, 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. Deskripsi Dataset

Dalam bab ini, akan dibahas hasil penelitian mengenai Klasifikasi Mahasiswa Berpotensi Putus Studi Menggunakan Algoritma *Naive Bayes* Pada Fakultas Teknik Unismuh Makassar. Hasil eksperimen mencakup pengelompokan data mahasiswa berdasarkan atribut, seperti nama, NIM, Angkatan, semester awal, jenis kelamin, pekerjaan ayah, penghasilan ayah, status ayah, pekerjaan ibu, penghasilan ibu, status ibu, pekerjaan wali, penghasilan wali, lulus, semester terakhir, IPK, SKS dan status putus studi dengan menggunakan Algoritma *Naive Bayes*.

B. Analisis Data Mentah

Data yang diolah dalam penelitian ini berasal dari mahasiswa Program Studi Teknik Informatika di 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, 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
Setiap `row`, yang berisi data lengkap dari satu mahasiswa, ditambahkan ke dalam `data_list`. `data_list` ini akan berisi data mentah dari semua mahasiswa setelah semua permintaan diproses.
 6. Mengatur variabel global `max_khs`
`max_khs = max(max_khs, len(khs))` digunakan untuk melacak jumlah maksimum catatan KHS yang dimiliki oleh mahasiswa dalam dataset. Ini berguna jika Anda perlu menormalkan jumlah kolom KHS dalam data akhir.
 7. Menjalankan fungsi asinkron dan mencetak status

- await get_mahasiswa() memanggil fungsi get_mahasiswa secara asinkron dan menunggu sampai semua data mahasiswa berhasil diambil.
- print(f'{total} Data berhasil di download') mencetak jumlah total data mahasiswa yang berhasil diambil.

Berikut hasilnya :

kodeProdi	angkatan	semesterKawal	nim	nama	jenisKelamin	tahunAkademikLulus	tanggalLulus	lulus	masaStudi	ayah_pekerjaan	ayah_penghasilan	ayah_status	ibu_pekerjaan	ibu_penghasilan
8	2020	2013	2013110581162313	MUHAMMAD RABIQ HICDIN	L	None	None	False	None	None	None	None	None	None
9	2020	2013	2013110581162413	MUHAMMAD SAVER	L	20182	None	True	8 Tahun, 11 Bulan	Tidak	500000	HOLUP	-	500000
1	2020	2013	2013110581162513	SALAL AKBAR	L	None	None	False	None	None	None	None	None	None
3	2020	2013	2013110581162613	MUHAMMAD MUNIR MUHADIS	L	None	None	False	None	None	None	None	None	None
4	2020	2013	2013110581162713	ETERNIA RIZKA	P	20172	None	True	4 Tahun, 8 Bulan	Wirahasta	800001-1000000	HOLUP	Wirahasta	800001-1000000
5	2020	2013	2013110581162813	SATUL RIZKA	P	20182	None	True	6 Tahun, 2 Bulan	Wirahasta	100001-1000000	HOLUP	Wirahasta	1000000
6	2020	2013	2013110581162913	ZARA SAPUTRA	L	None	None	False	None	None	None	None	None	None
7	2020	2013	2013110581163013	MUSLADA SAPUTRA	L	20182	None	True	5 Tahun, 9 Bulan	Wirahasta	200001-2000000	HOLUP	IT	800000
8	2020	2013	2013110581163113	MUHAMMAD HAFID	L	None	None	False	None	None	None	None	None	200001-1000000
9	2020	2013	2013110581163213	MELISSA ANAND TALUH	L	None	None	False	None	Tidak	800001-1000000	HOLUP	Wirahasta	800001-1000000

Gambar 4. Data Mentah

Data mentah mahasiswa program studi teknik informatika dengan memanfaatkan data set dari angkatan 2013-2015 yang terdiri dari 1493 data sebagai data set pelatihan, sistem ini akan dievaluasi untuk akurasi menggunakan metode *Naïve Bayes*. Pengaturan seperti nilai klasifikasi atau prediksi akan diterapkan dalam proses penentuan status putus studi sebagai data uji.

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 IPK, SKS dan Semester:

1) 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 tersedia di setiap baris. *Output* dari proses ini yaitu nilai IPK terakhir yang terisi untuk setiap mahasiswa. Berikut program yang dibuat pada *Visual Studio Code* :

```
import pandas as pd

# Baca data dari CSV
file_path = '/content/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 = '/content/hasil_ipk_terakhir.csv'
df.to_csv(output_file_path, index=False)

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

Gambar 5. Codingan IPK Terakhir

2) 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 di setiap baris. *Output* dari proses ini yaitu nilai total SKS lulus yang terisi untuk setiap mahasiswa. Berikut program yang dibuat pada *Visual Studio Code*:

```
import pandas as pd

# Baca data dari CSV
file_path = '/content/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 = '/content/hasil_sks_lulus.csv'
df.to_csv(output_file_path, index=False)

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

Gambar 6. Codingan SKS Lulus

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 program yang dibuat pada Visual Studio Code :

```
# 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 berupa 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')
```

Gambar 7. Codingan Semester

Berikut hasil penambahan atribut SKS Lulus, IPK Terakhir dan Semester

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

Gambar 8. 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:

```

# Definisi fungsi untuk menentukan status prestasi siswa studi
def tentukan_status_prestasi_siswa(studi):
    # Definisi variabel untuk parameter status
    if studi['sks_terakhir'] <= 1 and studi['sks_lulus'] <= 1 and studi['ipk_terakhir'] <= 1 and studi['ipk_lulus'] <= 1 and studi['sks_terakhir'] <= 1 and studi['sks_lulus'] <= 1 and studi['ipk_terakhir'] <= 1 and studi['ipk_lulus'] <= 1:
        return "Tidak Memuaskan"
    elif studi['sks_terakhir'] <= 2 and studi['sks_lulus'] <= 2 and studi['ipk_terakhir'] <= 2 and studi['ipk_lulus'] <= 2 and studi['sks_terakhir'] <= 2 and studi['sks_lulus'] <= 2 and studi['ipk_terakhir'] <= 2 and studi['ipk_lulus'] <= 2:
        return "Cukup Memuaskan"
    elif studi['sks_terakhir'] <= 3 and studi['sks_lulus'] <= 3 and studi['ipk_terakhir'] <= 3 and studi['ipk_lulus'] <= 3 and studi['sks_terakhir'] <= 3 and studi['sks_lulus'] <= 3 and studi['ipk_terakhir'] <= 3 and studi['ipk_lulus'] <= 3:
        return "Memuaskan"
    elif studi['sks_terakhir'] <= 4 and studi['sks_lulus'] <= 4 and studi['ipk_terakhir'] <= 4 and studi['ipk_lulus'] <= 4 and studi['sks_terakhir'] <= 4 and studi['sks_lulus'] <= 4 and studi['ipk_terakhir'] <= 4 and studi['ipk_lulus'] <= 4:
        return "Sangat Memuaskan"
    else:
        return "Dengan Pujian"

# Fungsi untuk menentukan status prestasi siswa studi
def tentukan_status_prestasi_siswa(studi):
    return tentukan_status_prestasi_siswa(studi)

# Contoh data
data = {
    'sks_terakhir': 1, 'sks_lulus': 1, 'ipk_terakhir': 1, 'ipk_lulus': 1, 'sks_terakhir': 2, 'sks_lulus': 2, 'ipk_terakhir': 2, 'ipk_lulus': 2, 'sks_terakhir': 3, 'sks_lulus': 3, 'ipk_terakhir': 3, 'ipk_lulus': 3, 'sks_terakhir': 4, 'sks_lulus': 4, 'ipk_terakhir': 4, 'ipk_lulus': 4, 'sks_terakhir': 5, 'sks_lulus': 5, 'ipk_terakhir': 5, 'ipk_lulus': 5
}

# Menampilkan informasi data prestasi siswa
print(data)

```

Gambar 9. 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 4. 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

Sumber: Pengolahan data, 2024

2) SKS Lulus

Data sks lulus yang digunakan dalam penelitian ini merupakan jumlah sks pada semester genap tahun 2013-2015. Untuk atribut sks lulus akan dibagi menjadi 2 (dua) kategori diantaranya:

Tabel 5. 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

Sumber: Pengolahan data, 2024

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

```
import pandas as pd
# Baca data dari CSV
file_path = 'hasil_siswa_persekolahan' + 'data dengan path file.xlsx'
df = pd.read_csv(file_path)

# Kategorisasi IPK TERAKHIR
def kategori_ipk(ipk):
    if ipk >= 80:
        return 'Memenuhi'
    else:
        return 'Tidak Memenuhi'

# Kategorisasi SKS_LULUS berdasarkan semester
def kategorisasi_sks_semester(semester, sks_lulus):
    if semester in ['Genap', 'Ganjil']:
        if sks_lulus >= 40:
            return 'Memenuhi'
        else:
            return 'Tidak Memenuhi'
    elif semester in ['Genap', 'G1']:
        if sks_lulus >= 85:
            return 'Memenuhi'
        else:
            return 'Tidak Memenuhi'
    elif semester in ['Genap', 'G2']:
        if sks_lulus >= 144:
            return 'Memenuhi'
        else:
            return 'Tidak Memenuhi'
    else:
        return 'Tidak Memenuhi' # Jika semester tidak sesuai yang diberikan

df['kategori_sks'] = df.apply(lambda row: kategorisasi_sks(row['semesterakhir'], row['sks_lulus']), axis=1)

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

# Tampilkan beberapa baris pertama untuk verifikasi
print(df[['IPK_TERAKHIR', 'kategori_ipk', 'sks_lulus', 'semesterakhir', 'kategori_sks']].head())
```

IPK_TERAKHIR	kategori_ipk	sks_lulus	semesterakhir	kategori_sks
81	Memenuhi	40.0	8	Tidak Memenuhi
81	Memenuhi	100.0	12	Memenuhi
82	Memenuhi	22.0	8	Tidak Memenuhi
83	Memenuhi	80.0	8	Tidak Memenuhi
84	Memenuhi	100.0	10	Memenuhi

Gambar 10. Codingan dan Hasil Kategorisasi

d. Seleksi Atribut

Atribut yang digunakan untuk proses klasifikasi hanya terdiri dari beberapa atribut 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 diunduh
df = pd.read_csv('hasil_bingung_kategori1.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")
```

angkatan	semesterAwal	NIM	nama	jenisKelamin	ayah_pekerjaan	ayah_penghasilan	ayah_status	ibu_pekerjaan	ibu_penghasilan
0	2013	20131	18581182313	MUHAMMAD	TAJID	L	500000	HIDUP	-
1	2013	20131	18581182413	MUHAMMAD	TAJID	L	500000	HIDUP	500000
2	2013	20131	18581182513	MUHAMMAD	TAJID	L	500000	HIDUP	500000
3	2013	20131	18581182613	MUHAMMAD	TAJID	L	500000	HIDUP	500000
4	2013	20131	18581182713	MUHAMMAD	TAJID	L	500000	HIDUP	500000

Gambar 11. Codingan dan Hasil Seleksi Atribut

e. Menghapus Atribut nim, nama dan jenis kelamin

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

```
import pandas as pd

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

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

# Simpan hasil ke CSV baru tanpa kolom 'nim' dan 'nama'
output_file_path = 'hasil_hapus_nim_nama.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_IPK	kategori_SKS	status_putus_studi
0	2013	20131	TAJID	500000	HIDUP	-	500000	HIDUP	500000	True	2013	1	1	False
1	2013	20131	TAJID	500000	HIDUP	500000	500000	HIDUP	500000	True	2013	1	1	False
2	2013	20131	TAJID	500000	HIDUP	500000	500000	HIDUP	500000	True	2013	1	1	False
3	2013	20131	TAJID	500000	HIDUP	500000	500000	HIDUP	500000	True	2013	1	1	False
4	2013	20131	TAJID	500000	HIDUP	500000	500000	HIDUP	500000	True	2013	1	1	False

Gambar 12. Codingan dan Hasil Penghapusan Atribut

2. Data Cleaning

Data cleaning atau pembersihan data pada penelitian ini dilakukan dengan menggunakan nilai mean atau nilai yang valid. Data *missing value* merupakan data yang bernilai kosong (*null*) yang tidak dapat digunakan dalam penelitian. Jumlah data *missing value* pada tiap atribut dapat dilihat pada tabel berikut.

Tabel 6. Jumlah Data *Missing Value*

Atribut	Jumlah Data <i>Missing Value</i>
angkatan	0
semesterAwal	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
status_putus_studi	0

Sumber: Pengolahan data, 2024

Jumlah *missing value* diatas merupakan jumlah data kosong pada tiap kolom atribut. Dalam satu baris data terdapat lebih dari satu atribut yang memiliki data *missing value*. Dari 16 atribut yang digunakan, 8 diantaranya memiliki data *missing value* yaitu atribut ayah_pekerjaan 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. Berikut contoh data *missing value*:

ibu_pekerjaan	ibu_penghasilan	ibu_status	wali_pekerjaan	wali_penghasilan	lulus	semesterTerakhir	k
NaN	NaN	NaN	NaN	NaN	False		8
	500000	HIDUP	NaN	NaN	True		12
NaN	NaN	NaN	NaN	NaN	False		3
NaN	NaN	NaN	NaN	NaN	False		3
WIRASWASTA	500001-1000000	HIDUP	NaN	NaN	True		10

Gambar 13. Data *Missing Value*

Tabel 7. Jumlah Data *Missing Value* Setelah menggunakan Mean/Data Valid

Atribut	Jumlah Data <i>Missing Value</i>
angkatan	0
semesterAwal	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
status_putus_studi	0

Sumber: Pengolahan data, 2024

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 8. Jumlah Data Hasil Proses Pembersihan

	Jumlah Data
Data awal	1493
Data <i>missing</i>	5.070
Data bersih	1493

Sumber: Pengolahan data, 2024

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 *categorical* dan *numerical* sehingga perlu diubah ke dalam bentuk data yang sama. Data akan diubah menggunakan *LabelEncoder*. Data yang akan diubah yaitu jenisKelamin, 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 dan contoh data yang telah di ubah:

```
[ ] 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 kategorize_wali_job(job):
    if job in ['SWASTA', 'WIRASWASTA', 'HERINTIS USAHA', 'PEGAWAI BUMN', 'Berdagang']:
        return 'Wiraswasta/Usaha'
    elif job in ['PNS', 'Dosen', 'ASN PPSK', 'POLDA SULSEL', 'POLRI']:
        return 'PNS/BUMN'
    elif job in ['PENSUN', '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', 'arab car']:
        return 'Mahasiswa/Profesional'
    else:
        return 'Lainnya'

# Terapkan kategori baru pada kolom pekerjaan wali
df['wali_pekerjaan'] = df['wali_pekerjaan'].map(kategorize_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 ['> 2000000', '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,
  'Honorer/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)

# Contoh fungsi untuk menambahkan kolom status pernikahan suami
df['status_pernikahan_suami'] = df[['ayah_pekerjaan', 'ibu_pekerjaan']].apply(
  lambda row: 'Lajang' if row['ayah_pekerjaan'] != 'Rumah Tangga' and row['ibu_pekerjaan'] != 'Rumah Tangga' else 'Menikah', axis=1)

# Contoh fungsi untuk menambahkan kolom status pernikahan istri
df['status_pernikahan_istri'] = df[['ayah_pekerjaan', 'ibu_pekerjaan']].apply(
  lambda row: 'Menikah' if row['ayah_pekerjaan'] != 'Rumah Tangga' and row['ibu_pekerjaan'] != 'Rumah Tangga' else 'Lajang', axis=1)

# Tampilkan informasi basis data untuk verifikasi
df.info()

```

Gambar 14. Codingan Transformasi Data
Berikut rincian pelabelan data pada setiap atribut:

Tabel 9. 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

Sumber: Pengolahan data, 2024

Tabel 10. Pelabelan Pada Atribut ayah_penghasilan

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

Sumber: Pengolahan data, 2024

Tabel 11. Pelabelan Pada Atribut ayah_status

ayah_status	Label
Hidup	0
Meninggal	1

Sumber: Pengolahan data, 2024

Tabel 12. 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

Sumber: Pengolahan data, 2024

Tabel 13. Pelabelan Pada Atribut ibu_penghasilan

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

Sumber: Pengolahan data, 2024

Tabel 14. Pelabelan Pada Atribut ibu_status

ayah_status	Label
Hidup	0
Meninggal	1

Sumber: Pengolahan data, 2024

Tabel 15. 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

Sumber: Pengolahan data, 2024

Tabel 16. Pelabelan Pada Atribut wali_penghasilan

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

Sumber: Pengolahan data, 2024

Tabel 17. Pelabelan Pada Atribut lulus

lulus	Label
FALSE	0
TRUE	1

Sumber: Pengolahan data, 2024

Tabel 18. Pelabelan Pada Atribut kategori_IPK

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

Sumber: Pengolahan data, 2024

Tabel 19. Pelabelan Pada Atribut kategori_SKS

kategori_SKS	Label
Memenuhi	0
Tidak Memenuhi	1

Sumber: Pengolahan data, 2024

Tabel 20. Pelabelan Pada Atribut status_putus_studi

status_putus_studi	Label
Potensial Putus	
Studi	0
Tidak Potensial	1

Sumber: Pengolahan data, 2024

D. Pembagian Data

Setelah dilakukan *preprocessing* data, selanjutnya akan dilakukan pembagian data. Pada tahap ini, data akan dibagi menjadi dua bagian 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 mempelajari 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 di bagi dengan proporsi 80% data training dan 20% data testing. Berikut jumlah data setelah dilakukan pembagian:

Tabel 21. Pembagian Data

Klasifikasi	Jumlah Data	Data Training (80%)	Data Testing (20%)
Potensial Putus Studi	722	582	140
Tidak Potensial	771	612	159
Total	1.493	1.194	299

Sumber: Pengolahan data, 2024

Dari tabel diatas diketahui bahwa jumlah data training yang dihasilkan adalah 1.493 data dan data testing berjumlah 299 data. Berikut kode program dan hasilnya:

```

# Baca data dari CSV
file_path = 'hasil_dengan_pelemban_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}')

```

✓ 0.1s
 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

Gambar 14. Codingan dan Hasil Pembagian Data

E. Klasifikasi Menggunakan Algoritma *Naïve Bayes*

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

Tabel 22. Hasil Akurasi

Algoritma	Akurasi (%)
<i>Naïve Bayes</i>	97%

Sumber: Pengolahan data, 2024

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 yang digunakan untuk mengetahui rata-rata keberhasilan dari suatu sistem dengan melakukan perulangan dengan mengacak atribut masukan sehingga sistem teruji untuk beberapa atribut input yang acak. Berikut ini hasil akurasi dari algoritma *naïve bayes* menggunakan pengujian *10-fold cross validation*:

Tabel 23. Hasil pengujian *10-fold cross validation*

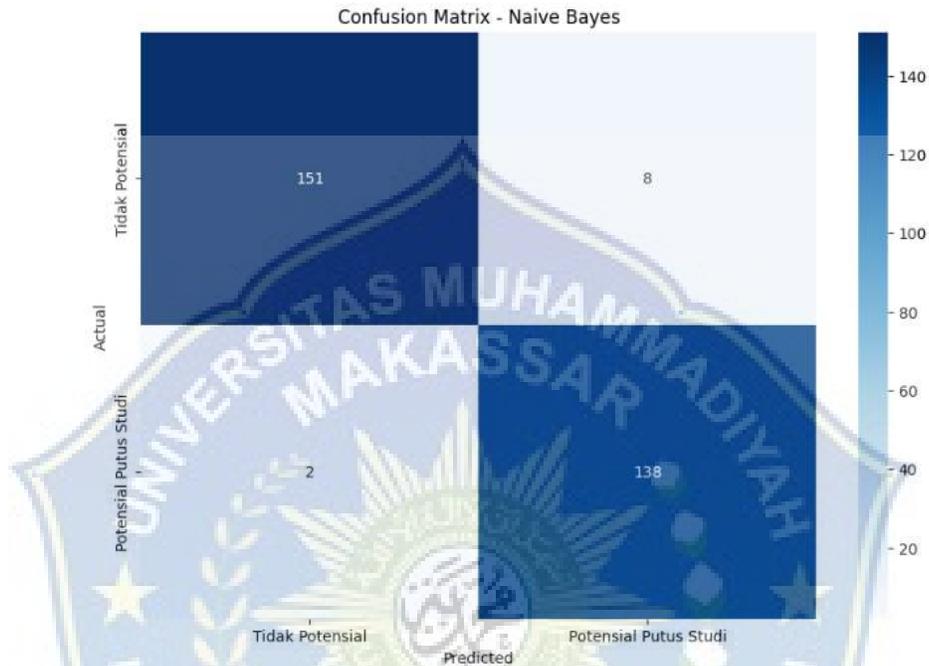
<i>K-Fold</i>	Akurasi (%)
1	95,33%
2	95,33%
3	95,33%
4	97,99%
5	96,64%
6	95,30%
7	93,29%
8	96,64%
9	95,30%
10	92,62%
Rata-Rata	95,38%

Sumber: Pengolahan data, 2024

Berdasarkan hasil dari pengujian diatas dapat dilihat bahwa akurasi algoritma *naïve bayes* sama dengan hasil akurasi sebelum dilakukan pengujian. Rata-rata akurasi dari setiap *fold* yang didapat ialah sebesar **95,38%**

G. Evaluasi Hasil Klasifikasi

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



Gambar 16. Confussion Matrix Pada Algoritma Naive Bayes

Dari gambar diatas, hasil evaluasi menggunakan *confussion matrix* menghasilkan jumlah data TP sebesar 151 data, jumlah FN sebesar 8 data, data FP berjumlah 2 data dan TN berjumlah 138 data. Lebih jelasnya dapat dilihat pada tabel berikut:

Tabel 24. Confussion Matrix Pada Algoritma Naive Bayes

Data Sebenarnya	Klasifikasi	
	Potensial Putus Studi	Tidak Potensial
Potensial Putus Studi	151 (TP)	8 (FN)
Tidak Potensial	2 (FP)	138 (TN)

Sumber: Pengolahan data, 2024

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

- a. *True Positive* (TP) menjelaskan dimana data terklasifikasi Potensial Putus Studi, memang benar Potensial Putus Studi. Dalam hal ini jumlah data yang didapat sebanyak 31 data.
- b. *False Positive* (FP) menjelaskan bahwa data yang terklasifikasi Potensial Putus Studi ternyata Tidak Potensial. Jumlah data yang didapat sebesar 0 data.
- c. *False Negative* (FN) menjelaskan bahwa data yang terklasifikasi Tidak Potensial sebenarnya adalah Potensial Putus Studi. Data yang dihasilkan berjumlah 1 data.
- d. *True Negative* (TN) menjelaskan dimana data yang terklasifikasi Tidak Potensial, memang benar Tidak Potensial. Jumlah data yang didapat sebanyak 51 data

Pengukuran kinerja algoritma *naive bayes* dapat dilihat pada nilai *akurasi*, *precision*, *recall*, dan *f1-score* yang dihasilkan berdasarkan data dari *confusion matrix*. Berikut nilai, *precision*, *recall* dan *f1-score* yang dihasilkan

```

Confusion Matrix:
[[151  8]
 [ 2 138]]
Classification Report:

```

	precision	recall	f1-score	support
0	0.99	0.95	0.97	159
1	0.95	0.99	0.97	140
accuracy			0.97	299
macro avg	0.97	0.97	0.97	299
weighted avg	0.97	0.97	0.97	299

```

Rata-rata Precision: 0.9660667920136091
Rata-rata Recall: 0.9676999101527404
Rata-rata F1-Score: 0.9664918414918415

```

Gambar 17. Hasil Pengukuran Kinerja Algoritma *Naive Bayes*

Pada gambar di atas dapat dilihat bahwa hasil *precision* untuk data Potensial Putus Studi dan Tidak Potensial masing-masing sebesar 99% dan 95%. Nilai *recall* dari data Potensial Putus Studi dan Tidak Potensial yang dihasilkan yaitu 95% dan 99%. Selanjutnya untuk nilai *f1-score* dari data Potensial Putus Studi dan Tidak Potensial yang didapat yaitu 97% dan 97%. Nilai rata-rata (*macro average*) dari *precision*, *recall* dan *f1-score* masing-masing sebesar 97%, 97% dan 97%.

BAB V

PENUTUP

A. Kesimpulan

Dari penelitian yang telah dilakukan didapat Kesimpulan:

Algoritma Naive Bayes 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 Naive Bayes memungkinkan pemodelan probabilistik yang sederhana namun efektif dalam mengidentifikasi mahasiswa dengan risiko putus studi berdasarkan pola yang ada dalam data.

Dari hasil yang didapat, algoritma *naive bayes* memiliki hasil performa yang cukup tinggi dengan akurasi sebesar 97%. Sehingga algoritma ini menjadi salah satu algoritma yang baik dalam mengklasifikasi mahasiswa potensi putus studi dan tidak potensi pada jurusan Teknik Informatika Universitas Muhammadiyah Makassar.

B. Saran

Saran untuk penelitian selanjutnya yaitu:

1. Untuk proses klasifikasi dapat menggunakan algoritma yang lain seperti *Decision Tree C4.5*, *KNearest Neighbour* dan *Support Vector Machines* (SVM). 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

- Andreansyach, C. R., Fardiansyah, T. S., Apriani, D., Sani, A., & Kunci, K. (2022). *Prediksi Persentase Kelulusan Mahasiswa STMIK Widuri Menggunakan Algoritma Naïve Bayes*. 75-84.
- Arum, L., Putri, B., Matematika, P. S., Sains, F., Teknologi, D. A. N., Islam, U., Maulana, N., & Ibrahim, M. (2022). *KLASIFIKASI FAKTOR PENYEBAB SISWA PUTUS SEKOLAH MENGGUNAKAN METODE DECISION TREE ID3*. *KLASIFIKASI FAKTOR PENYEBAB SISWA PUTUS SEKOLAH MENGGUNAKAN METODE DECISION TREE ID3*.
- Kamila, V. Z., & Subastian, E. (2019). KNN vs Naive Bayes Untuk Deteksi Dini Putus Kuliah Pada Profil Akademik Mahasiswa. *Jurnal Rekayasa Teknologi Informasi (JURTI)*, 3(2), 116. <https://doi.org/10.30872/jurti.v3i2.3097>
- Mahanggara, A., & Laksito, A. D. (2019). Prediksi Pengunduran Diri Mahasiswa Universitas Amikom Yogyakarta Menggunakan Metode Naive Bayes. *Simetris: Jurnal Teknik Mesin, Elektro dan Ilmu Komputer*, 10(1), 273-280. <https://doi.org/10.24176/simet.v10i1.2967>
- Mulyadi, C., & Juniadi, M. N. (2019). *Prediksi Keaktifan Studi Mahasiswa Baru dengan Algoritma Naive Bayes*. 299-303.
- Nuralia, S., Harliana, H., & Prabowo, T. (2022). Implementasi Naïve Bayes Classifier Dalam Memprediksi Kelulusan Mahasiswa. *Journal Automation Computer Information System*, 3(1), 63-72. <https://doi.org/10.47134/jacis.v3i1.57>
- Rayuwati, Husna Gemasih, & Irma Nizar. (2022). IMPLEMENTASI ALGORITMA NAIVE BAYES UNTUK MEMPREDIKSI TINGKAT PENYEBARAN COVID. *Jurnal Riset Rumpun Ilmu Teknik*, 1(1), 38-46. <https://doi.org/10.55606/jurritek.v1i1.127>
- Rosaly, R., & Prasetyo, A. (2020). Flowchart Beserta Fungsi dan Simbol-Simbol. *Journal of Chemical Information and Modeling*, 2(3), 5-7.
- 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>

- Syahril Dwi Prasetyo, Shofa Shofiah Hilabi, & Fitri Nurapriani. (2022). Analisis Sentimen Relokasi Ibukota Nusantara Menggunakan Algoritma Naïve Bayes dan KNN. *Jurnal KomtekInfo*, 10, 1-7. <https://doi.org/10.35134/komtekinfo.v10i1.330>
- Ulinuha, N., & Fanani, A. (2022). Klasifikasi Status Drop Out Mahasiswa Menggunakan Naïve Bayes dengan Seleksi Fitur Information Gain. *Techno.Com*, 22(4), 1014-1025. <https://doi.org/10.33633/tc.v22i4.9004>
- Wijyaningrum, V. N., Putri, I. K., Kirana, A. P., Mubarak, M. R., Harahap, D. M., Hamesha, B. R., Informasi, J. T., Malang, P. N., Correlation, P., & Syaraf, J. (2019). *Analisis performa seleksi atribut untuk menentukan potensi mahasiswa putus studi*. 237-244.
- Yuniarti, W. D., Faiz, A. N., & Setiawan, B. (2020). Identifikasi Potensi Keberhasilan Studi Menggunakan Naïve Bayes Classifier. *Walisono Journal of Information Technology*, 2(1), 1. <https://doi.org/10.21580/wjit.2020.2.1.5204>
- Zarti, M. N., Sahputra, E., Sonita, A., & Apridiansyah, Y. (2022). Application Of Data Mining Using The Naïve Bayes Classification Method To Predict Public Interest Participation In The 2024 Elections. *Jurnal Komputer, Informasi dan Teknologi*, 3(1), 105-114. <https://doi.org/10.53697/jkomitek.v3i1.1192>





Lampiran 1. Data Mentah

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	
1	kodePred	angkatan	semester	nim	nama	jenisKelamin	tahunAka	tanggalLulus	masaStudi	ayah_pegawai	pen_ayah_stat	ibu_pegawai	ibu_stat	wali_pegawai	wali_penghis				
2	20201	2016	20161	10596666666	Muhamm P				FALSE	WIRASWA	> 500000	HIDUP	WIRASWA	3000001-5	HIDUP		500000	2	
3	20201	2016	20161	105821100316	MUH. RAFL		20201		TRUE	4 Tahun, 1	KEPOLISIA	> 500000	HIDUP	PNS	1500001-2	HIDUP		500000	2
4	20201	2016	20161	105821113816	MUHAMM L				FALSE	PENSIUN	1500001-2	HIDUP	PENSIUN	1000001-1	HIDUP	SWASTA	1000001-2	2	
5	20201	2016	20161	105821100216	AFRIANSY L		20221		TRUE	6 Tahun, 5	PENSIUN	1000001-1	MEMENGG	PNS	3000001-5	HIDUP		500000	2
6	20201	2016	20161	105821100116	MURGHAIL		20222		TRUE	6 Tahun, 1-	500000	MEMENGG	-	500000	HIDUP			500000	2
7	20201	2016	20161	105821100716	ILMAN NUL				FALSE	TANI	1500001-2	HIDUP	-	> 500000	HIDUP			500000	2
8	20201	2016	20161	1058211008216	SYAMSUL L		20201		TRUE	4 Tahun, 5	WIRASWA	1000001-3	HIDUP	-	500000	HIDUP		500000	2
9	20201	2016	20161	105821100416	SULTAN H L		20192		TRUE	3 Tahun, 1	WIRASWA	1000001-1	HIDUP	IBU RUMA	500000	HIDUP		500000	2
10	20201	2016	20161	1058211007416	LUKMAN F L		20192		TRUE	3 Tahun, 1	SWASTA	2000001-2	HIDUP	-	500000	HIDUP	SWASTA	2000001-2	2
11	20201	2016	20161	105821100516	TSABT L		20192		TRUE	3 Tahun, 1	TANI	1500001-2	HIDUP	IBU RUMA	500000	HIDUP		500000	2
12	20201	2016	20161	105821113316	MUHAMM L		20221		TRUE	6 Tahun, 4	SWASTA	3000001-5	HIDUP	WIRASWA	500000	HIDUP		500000	2
13	20201	2016	20161	105821100816	FADEL MCL		20222		TRUE	6 Tahun, 1	PNS	> 500000	HIDUP	-	3000001-5	HIDUP			2
14	20201	2016	20161	105821100916	BAYU FATIL		20222		TRUE	6 Tahun, 1	PENSIUN	500001-1	HIDUP	-	500000	HIDUP		500000	2
15	20201	2016	20161	105821100916	ANDI KHVL				FALSE	SWASTA	500000	HIDUP	PNS	5000001-1	HIDUP			500000	2
16	20201	2016	20161	105821100616	MUHAMM L		20201		TRUE	4 Tahun, 5	NELAYAN	500000	HIDUP	-	500000	HIDUP		500000	2
17	20201	2016	20161	105821100716	SASMITAS P		20192		TRUE	3 Tahun, 1	TANI	5000001-1	HIDUP	-	500000	HIDUP		500000	2
18	20201	2016	20161	105821100616	RYAN BASIL				FALSE	PNS	2000001-2	HIDUP	IBU RUMA	500000	HIDUP	PNS	2000001-2	2	
19	20201	2016	20161	105821100816	S MARLIN L		20222		TRUE	6 Tahun, 1	PENSIUN	500001-1	HIDUP	IBU RUMA	500000	HIDUP		500000	2
20	20201	2016	20161	105821100616	ABDUL SYIL		20201		TRUE	4 Tahun, 5	PNS	1000001-1	HIDUP	IBU RUMA	500000	HIDUP	PENSIUN	500001-1	2
21	20201	2016	20161	105821100416	AMF BUCI L		20221		TRUE	6 Tahun, 2	TANI	5000001-1	HIDUP	TANI	500000	HIDUP		500000	2
22	20201	2016	20161	105821101116	TASHM L				FALSE	TANI	500001-1	HIDUP	-	500000	HIDUP			500000	2
23	20201	2016	20161	105821101216	SUWARDI L				FALSE	TANI	500000	HIDUP	TANI	500000	HIDUP	TANI		500000	2
24	20201	2016	20161	105371105116	ANHRUDD L				FALSE	TANI	5000001-1	HIDUP	TANI	5000001-1	HIDUP			500000	2
25	20201	2016	20161	105371100316	MOHAMM L				FALSE	WIRASWA	5000001-1	HIDUP	-	500000	HIDUP			500000	2
26	20201	2016	20161	105821101316	ILHAM TO L		20222		TRUE	6 Tahun, 1	PNS	1000001-3	HIDUP	URT	500000	HIDUP	PNS	1000001-2	2
27	20201	2016	20161	105821101616	MUHAMM L		20202		TRUE	4 Tahun, 1	TANI	500000	HIDUP	TANI	500000	HIDUP	TANI	500000	2
28	20201	2016	20161	105821101716	PRIDANIA P		20201		TRUE	4 Tahun, 1	TANI	5000001-1	HIDUP	WIRASWA	500000	HIDUP		500000	2

4135	20241	10584112	Alya shita P	False	WIRASWA	2000001-3	HIDUP	PNS	3000001-4	HIDUP	-	500000							
4136	20241	10584112	ALWAN F L	False	WIRASWA	1500001-2	HIDUP	-	500000	HIDUP									
4137	20241	10584112	EGI ADRI L	False															
4138	20241	10584112	M. ahsan. r L	False															
4139	20241	10584112	BAKRI L	False	NELAYAN	500000	HIDUP	-	500000	HIDUP									
4140	20241	10584112	Muh. Abra L	False	SWASTA	3000001-5	HIDUP	-	500000	HIDUP									
4141	20241	10584112	ARDI FEB L	False															
4142	20241	10584112	NAUFAL F L	False	WIRASWA	5000001-1	HIDUP	Pengurus	500000	HIDUP	-								
4143	20241	10584112	JABAL L	False	TANI	1000001-1	HIDUP	-	500000	HIDUP	SWASTA	1500001-2							
4144	20241	10584112	RAHMAD L	False	WIRASWA	> 500000	HIDUP	-	500000	MEMENGGAL									
4145	20241	10584112	Aulia Putri P	False	PNS	3000001-5	HIDUP	TANI	500000	HIDUP									
4146	20241	10584112	DAFA RIZ L	False															
4147	20241	10584112	Sofia Rahy P	False	WIRASWA	1500001-2	HIDUP	Ibu Ruma	500000	HIDUP									
4148	20241	10584112	MUH. REZKY TRY ANANDA	False	SWASTA	2000001-3	HIDUP	-	1500001-2	HIDUP									
4149	20241	10584112	Nurul Inay P	False	Marbot me	500000	HIDUP	-	500000	HIDUP	-								500001-10
4150	20241	10584112	CINDY PAP	False															
4151	20241	10584112	IBNU SYAL	False															
4152	20241	10584112	Riska Amz P	False															
4153	20241	10584112	THALITA J P	False															
4154	20241	10584112	FERDIAN L	False															
4155	20241	10584112	Muh. Rahi L	False	PNS	3000001-5	HIDUP	ASN PPP1	2000001-3	HIDUP									
4156	20241	10584112	Avashillah L	False															

Lampiran 2. Data Preprocessing

1	angkatan	semester	nama	jenisKelas	ayah	pekerjaan_ayah	ibu	pekerjaan_ibu	penghasilan	statal	wali	pekerjaan_wali	pendidikan	sementar	kategori	(kategori_SKS
2	2013	20131	10581182313 MUHAMM L											FALSE	8	Memuaskan Tidak Memenuhi
3	2013	20131	10581182413 MUHAMM L	TANI	500000	HIDUP		500000	HIDUP					TRUE	12	Memuaskan Memenuhi
4	2013	20131	10581182513 FAISAL AK L											FALSE	3	Memuaskan Tidak Memenuhi
5	2013	20131	10581182613 MUNIR M L											FALSE	5	Memuaskan Tidak Memenuhi
6	2013	20131	10581182713 SY FATIMAH P	WIRASWA	500001-10	HIDUP	WIRASWA	500001-10	HIDUP					TRUE	10	Memuaskan Memenuhi
7	2013	20131	10581182813 SAIFUL RCL	WIRASWA	500001-10	HIDUP	WIRASWA	500000	HIDUP					TRUE	11	Sangat Me Memenuhi
8	2013	20131	10581182913 JUPRI L											FALSE	4	Tidak Mer Tidak Memenuhi
9	2013	20131	10581183013 HENRA ACL	WIRASWA	2000001-3	HIDUP	IRT	500000	HIDUP					TRUE	12	Memuaskan Memenuhi
10	2013	20131	10581183113 MOHAMM L	TANI	500000	MENINGG		500001-10	HIDUP		500000			FALSE	12	Cukup Me Tidak Memenuhi
11	2013	20131	10581183213 HILMAN AL	TANI	500001-10	HIDUP	WIRASWA	500001-10	HIDUP					FALSE	12	Cukup Me Tidak Memenuhi
12	2013	20131	10581183313 RUMAM L	TANI	500001-10	HIDUP	URT	500001-10	HIDUP					FALSE	10	Cukup Me Tidak Memenuhi
13	2013	20131	10581183413 FIRDAUS FL											FALSE	7	Cukup Me Tidak Memenuhi
14	2013	20131	10581183513 SUPARDI FL											FALSE	5	Memuaskan Tidak Memenuhi
15	2013	20131	10581183613 NURHIDAY P	TANI	2000001-3	HIDUP		500000	HIDUP					TRUE	10	Sangat Me Memenuhi
16	2013	20131	10581183713 GIYAFUDD L	TANI	500000	HIDUP	IBU RUMA	500000	HIDUP	WIRASWASTA				TRUE	12	Memuaskan Memenuhi
17	2013	20131	10581183813 AKTIA P	Nelayan	500001-10	HIDUP	WIRASWA	500000	HIDUP					TRUE	10	Sangat Me Memenuhi
18	2013	20131	10581183913 ARIZAL AS L											FALSE	3	Memuaskan Tidak Memenuhi
19	2013	20131	10581184013 ASMAUL F P	PNS	3000001-5	HIDUP	IRT	500000	HIDUP	PNS	3000001-5			TRUE	10	Sangat Me Memenuhi
20	2013	20131	10581184113 NINING H P	TANI	500000	HIDUP	TANI	500000	HIDUP					TRUE	10	Sangat Me Memenuhi
21	2013	20131	10581184213 MUH LHA L	TANI	500001-10	HIDUP		500000	HIDUP					TRUE	12	Memuaskan Memenuhi
22	2013	20131	10581184313 HERMAW L											FALSE	3	Tidak Mer Tidak Memenuhi
23	2013	20131	10581184413 SYAMSAL I	TANI	500001-10	MENINGG	IRT	500001-10	HIDUP	TANI	500001-10			TRUE	12	Memuaskan Memenuhi
278	2013	20131	K10582127513 SETIAWAN L	TUKANG	5000001-1	HIDUP								TRUE	6	Memuaskan Memenuhi
279	2014	20141	10581202914 AMRAN B	TANI	500000	HIDUP	TANI	500000	HIDUP					TRUE	11	Sangat Me Memenuhi
280	2014	20141	10581203014 MUH BWA L	TANI	500000	HIDUP	IBU RUMA	500000	HIDUP					TRUE	11	Sangat Me Memenuhi
281	2014	20141	10581203114 MUHAMM L	PNS	2000001-3	HIDUP	PNS	1500001-2	HIDUP					FALSE	10	Cukup Me Tidak Memenuhi
282	2014	20141	10581203214 AHMAD A L	TANI	500000	HIDUP	TANI	500000	HIDUP					TRUE	13	Memuaskan Memenuhi
283	2014	20141	10581203314 EDWIN HEL	TANI	500000	HIDUP	IRT	500000	HIDUP					FALSE	10	Cukup Me Tidak Memenuhi
284	2014	20141	10581203414 NOR IMAH L	TANI	500001-10	HIDUP	TANI	500001-10	HIDUP					TRUE	9	Sangat Me Memenuhi
285	2014	20141	10581203514 ABDUL SYL	TANI	500000	HIDUP	Tani	500000	HIDUP	Proses S2	500000			TRUE	11	Sangat Me Memenuhi
286	2014	20141	10581203614 RETNO M L											FALSE	10	Tidak Mer Tidak Memenuhi
287	2014	20141	10581203714 IRFAN L											TRUE	12	Sangat Me Memenuhi
288	2014	20141	10581203814 ZUL HIDAY L	TANI	500000	HIDUP	TANI	500000	HIDUP					TRUE	11	Sangat Me Memenuhi
289	2014	20141	10581203914 AKBAR L	TANI	500000	HIDUP	TANI	500000	HIDUP		500000			TRUE	11	Sangat Me Memenuhi
290	2014	20141	10581204014 MIFTA UL P	WIRASWA	500000	HIDUP		500000	HIDUP					TRUE	8	Sangat Me Memenuhi
291	2014	20141	10581204114 WAHYU S L	PNS	300001-10	HIDUP		500001-10	HIDUP					TRUE	10	Memuaskan Memenuhi
292	2014	20141	10581204214 MUHAMM L											FALSE	10	Memuaskan Tidak Memenuhi
293	2014	20141	10581204314 EDI EDW L											FALSE	10	Tidak Mer Tidak Memenuhi
294	2014	20141	10581204414 RISAL L	TANI	500000	HIDUP	TANI	500000	HIDUP					TRUE	11	Sangat Me Memenuhi
295	2014	20141	10581204514 MUH IRI L	WIRASWA	500001-10	HIDUP	PNS	2000001-3	HIDUP					TRUE	11	Memuaskan Memenuhi
296	2014	20141	10581204614 ANJUNORAH P											FALSE	10	Sangat Me Tidak Memenuhi
297	2014	20141	10581204714 MUH IDRIS L											FALSE	10	Tidak Mer Tidak Memenuhi
298	2014	20141	10581204814 FIRMAN M L	TANI	500000	HIDUP	TANI	500000	HIDUP					TRUE	11	Memuaskan Memenuhi
299	2014	20141	10581204914 MOH NOL L											FALSE	10	Tidak Mer Tidak Memenuhi
300	2014	20141	10581205014 ARTI RAH L	TANI	500000	HIDUP	TANI	500000	HIDUP					TRUE	15	Sangat Me Tidak Memenuhi
371	2015	20151	1.0583E+11 A ABHAM L											FALSE	17	Tidak Mer Tidak Memenuhi
474	2015	20151	1.0583E+11 HANZAH L	TANI	1000001-1	HIDUP	IBU RUMA	500001-10	HIDUP					TRUE	13	Sangat Me Memenuhi
475	2015	20151	1.0583E+11 PRATIWI FL	TANI	1000001-1	HIDUP	URT	500000	HIDUP					FALSE	17	Cukup Me Tidak Memenuhi
476	2015	20151	1.0583E+11 A. ARWIA L	TANI	500000	HIDUP	TANI	500000	HIDUP					TRUE	14	Memuaskan Memenuhi
477	2015	20151	1.0583E+11 APRIANTO L	WIRASWA	500000	MENINGG	IRT	500000	MENINGG	SWASTA	3000001-5			TRUE	14	Memuaskan Memenuhi
478	2015	20151	1.0583E+11 SUPARDI FL											FALSE	17	Tidak Mer Tidak Memenuhi
479	2015	20151	1.0583E+11 SUICI RAHI P											FALSE	17	Sangat Me Tidak Memenuhi
480	2015	20151	1.0583E+11 AUMA WIC P	PNS	3000001-5	MENINGG	PNS	3000001-5	HIDUP					TRUE	14	Sangat Me Memenuhi
481	2015	20151	1.0583E+11 MUFAKKIL L											FALSE	17	Tidak Mer Tidak Memenuhi
482	2015	20151	1.0583E+11 YUSRIADI L	TANI	500001-10	HIDUP		500000	HIDUP					TRUE	14	Memuaskan Memenuhi
483	2015	20151	1.0583E+11 RAJA FATI L	WIRASWA	1000001-1	MENINGG	IBU RUMAH	1000001-1	HIDUP					TRUE	14	Memuaskan Memenuhi
484	2015	20151	K10582164515 ARHMAD L	PNS	2000001-3	HIDUP		500000	HIDUP					TRUE	7	Sangat Me Memenuhi
485	2015	20151	K10582164615 DHIVA FIT P	PENSUIN	2000001-3	HIDUP	PNS	3000001-5	HIDUP					TRUE	5	Sangat Me Memenuhi
486	2015	20151	K10582164915 KHARUL F L	PNS		HIDUP	PNS		HIDUP					FALSE	17	Sangat Me Tidak Memenuhi
487	2015	20151	K10582165315 RASHANA P	WIRASWA	1500001-2	HIDUP	PENSUIN	2000001-3	HIDUP					TRUE	5	Sangat Me Memenuhi
488	2015	20151	K10582165415 SUKMAWIN P					500000	HIDUP					TRUE	5	Sangat Me Memenuhi
489	2015	20151	K10582165515 HN SULAS P											FALSE	17	Cukup Me Tidak Memenuhi
490	2015	20151	K10582165915 IRWAN L	TANI	500001-10	HIDUP		500000	MENINGG	WIRASWA >	5000000			FALSE	17	Sangat Me Tidak Memenuhi
491	2015	20151	K10582166215 FITRIANTI P	WIRASWA	3000001-5	MENINGG		500000	HIDUP	WIRASWASTA				TRUE	5	Sangat Me Memenuhi
492	2015	20151	K10582167215 HADRIYA P	PNS	3000001-5	MENINGG	PNS	3000001-5	MENINGG	SWASTA	2000001-3			TRUE	5	Memuaskan Memenuhi
493	2015	20151	K10582167515 FITRI RAMP	WIRASWA	2000001-3	HIDUP	PNS	3000001-5	HIDUP					TRUE	5	Sangat Me Memenuhi
494	2015	20151	K10582170915 ABDUL M L	PENSUIN	1500001-2	MENINGG		500000	MENINGG	PNS	3000001-5			TRUE	5	Sangat Me Memenuhi

Lampiran 3. Transformasi Data

id	angkatan	semester	ayah_peki	ayah_pen	ayah_stat	ibu_peki	ibu_pengl	ibu_statu	wali_peki	wali_peng	lulus	semester	kategori_1	kategori_1_status	potensi_pulus	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	3	2	1	0	
5	2013	20131	2	0	0	5	0	0	0	0	0	3	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	2	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	3	0	1	
16	2013	20131	2	0	0	1	0	0	0	0	1	12	2	0	1	
17	2013	20131	5	0	0	0	0	0	0	0	1	10	3	0	1	
18	2013	20131	2	0	0	5	0	0	0	0	0	3	2	1	0	
19	2013	20131	0	2	0	5	0	0	1	2	1	10	3	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	0	5	0	0	1	1	1	10	2	0	0	
178	2013	20131	5	1	0	5	0	0	0	0	1	6	2	0	1	
179	2014	20141	2	0	0	2	0	0	0	0	0	1	11	3	0	1
180	2014	20141	2	0	0	1	0	0	0	0	1	11	3	0	1	
181	2014	20141	0	2	0	0	1	0	0	0	0	19	1	1	0	
182	2014	20141	2	0	0	2	0	0	0	0	1	13	2	0	1	
183	2014	20141	2	0	0	5	0	0	0	0	0	19	1	1	0	
184	2014	20141	2	0	0	2	0	0	0	0	1	9	3	0	1	
185	2014	20141	2	0	0	2	0	0	5	0	1	11	3	0	1	
186	2014	20141	2	0	0	5	0	0	0	0	0	19	0	1	0	
187	2014	20141	5	0	1	5	0	0	0	0	1	13	3	0	1	
188	2014	20141	2	0	0	2	0	0	0	0	1	11	3	0	1	
189	2014	20141	2	0	0	2	0	0	5	0	1	11	3	0	1	
190	2014	20141	0	0	0	5	0	0	0	0	1	8	3	0	1	
191	2014	20141	0	0	0	5	0	0	0	0	1	10	2	0	1	
192	2014	20141	2	0	0	5	0	0	0	0	0	19	2	1	0	
193	2014	20141	2	0	0	5	0	0	0	0	0	19	0	1	0	
194	2014	20141	2	0	0	2	0	0	0	0	1	11	3	0	1	
195	2014	20141	0	0	0	0	2	0	0	0	1	11	2	0	1	
196	2014	20141	2	0	0	5	0	0	0	0	0	19	3	1	0	
197	2014	20141	2	0	0	5	0	0	0	0	0	19	0	1	0	
198	2014	20141	2	0	0	2	0	0	0	0	1	11	2	0	1	
199	2014	20141	2	0	0	5	0	0	0	0	0	19	0	1	0	
100	2014	20141	2	0	0	2	0	0	0	0	1	15	3	1	0	
1473	2015	20151	2	0	0	5	0	0	0	0	0	17	0	1	0	
1474	2015	20151	2	1	0	1	0	0	0	0	1	13	3	0	1	
1475	2015	20151	2	1	0	5	0	0	0	0	0	17	1	1	0	
1476	2015	20151	2	0	0	2	0	0	0	0	1	14	2	0	1	
1477	2015	20151	0	0	1	5	0	1	0	2	1	14	2	0	1	
1478	2015	20151	2	0	0	5	0	0	0	0	0	17	0	1	0	
1479	2015	20151	2	0	0	5	0	0	0	0	0	17	3	1	0	
1480	2015	20151	0	2	1	0	2	0	0	0	1	14	3	0	1	
1481	2015	20151	2	0	0	5	0	0	0	0	0	17	0	1	0	
1482	2015	20151	2	0	0	5	0	0	0	0	1	14	2	0	1	
1483	2015	20151	0	1	1	5	1	0	0	0	1	14	2	0	1	
1484	2015	20151	0	2	0	5	0	0	0	0	1	7	3	0	1	
1485	2015	20151	1	2	0	0	2	0	0	0	1	5	3	0	1	
1486	2015	20151	0	0	0	0	0	0	0	0	0	17	3	1	0	
1487	2015	20151	0	1	0	0	2	0	0	0	1	5	3	0	1	
1488	2015	20151	5	0	0	5	0	0	0	0	1	5	3	0	1	
1489	2015	20151	2	0	0	5	0	0	0	0	0	17	1	1	0	
1490	2015	20151	2	0	0	5	0	1	0	2	0	17	3	1	0	
1491	2015	20151	0	2	1	5	0	0	0	0	1	5	3	0	1	
1492	2015	20151	0	2	1	0	2	1	0	2	1	5	2	0	1	
1493	2015	20151	0	2	0	0	2	0	0	0	1	5	3	0	1	

Lampiran 4. Codingan Dan Output

%pip install aiohttp pandas

```
Requirement already satisfied: aiohttp in c:\users\agus\appdata\local\packages\pythonsoftwarefoundation.python.3.10_x-ww8r63qf6e@localcache\local-cache
Requirement already satisfied: pandas in c:\users\agus\appdata\local\packages\pythonsoftwarefoundation.python.3.10_x-ww8r63qf6e@localcache\local-cache
Requirement already satisfied: aiohttp>=1.1.1 in c:\users\agus\appdata\local\packages\pythonsoftwarefoundation.python.3.10_x-ww8r63qf6e@localcache\local-cache
Requirement already satisfied: attrs>=17.3.0 in c:\users\agus\appdata\local\packages\pythonsoftwarefoundation.python.3.10_x-ww8r63qf6e@localcache\local-cache
Requirement already satisfied: frozenlist>=1.1.1 in c:\users\agus\appdata\local\packages\pythonsoftwarefoundation.python.3.10_x-ww8r63qf6e@localcache\local-cache
Requirement already satisfied: multidict<7.0, >=4.5 in c:\users\agus\appdata\local\packages\pythonsoftwarefoundation.python.3.10_x-ww8r63qf6e@localcache\local-cache
Requirement already satisfied: yarl<2.0, >=1.0 in c:\users\agus\appdata\local\packages\pythonsoftwarefoundation.python.3.10_x-ww8r63qf6e@localcache\local-cache
Requirement already satisfied: async-timeout<5.0, >=4.0 in c:\users\agus\appdata\local\packages\pythonsoftwarefoundation.python.3.10_x-ww8r63qf6e@localcache\local-cache
Requirement already satisfied: numpy>=1.22.4 in c:\users\agus\appdata\local\packages\pythonsoftwarefoundation.python.3.10_x-ww8r63qf6e@localcache\local-cache
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\agus\appdata\local\packages\pythonsoftwarefoundation.python.3.10_x-ww8r63qf6e@localcache\local-cache
Requirement already satisfied: pytz>=2020.1 in c:\users\agus\appdata\local\packages\pythonsoftwarefoundation.python.3.10_x-ww8r63qf6e@localcache\local-cache
Requirement already satisfied: tzdata>=2022.7 in c:\users\agus\appdata\local\packages\pythonsoftwarefoundation.python.3.10_x-ww8r63qf6e@localcache\local-cache
Requirement already satisfied: aio>=1.8 in c:\users\agus\appdata\local\packages\pythonsoftwarefoundation.python.3.10_x-ww8r63qf6e@localcache\local-cache
Requirement already satisfied: idna>=3.0 in c:\users\agus\appdata\local\packages\pythonsoftwarefoundation.python.3.10_x-ww8r63qf6e@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\agus\appdata\local\packages\pythonsoftwarefoundation.python.3.10_x-ww8r63qf6e@python.exe -m pip install
```

```
import aiohttp
import asyncio
import pandas as pd
```

```
async def query_graphql(session, nim):
    url = 'https://sicekcok.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('ft-2016.csv')['nim'].tolist()
data_list = []
total = len(nim_list)
max_khs = 0
print(f"Total NIM: {total}")

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')
```

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	akhir	nim	nama	jenisKelamin	tahunAkademikLulus	tanggalLulus	lulus	masaStudi	ayah_pekerjaan	ayah_penghasilan	ayah
0	22202	2013	2013	10581182313	MUHAMMAD FARYD HIKSYAL	L	None	None	False	None	None	None	None
1	22202	2013	2013	10581182413	MUHAMMAD SAKIR	L	20182	None	True	3 Tahun, 11 Bulan	TANI	500000	None
2	22201	2013	2013	10581182513	FASAL AKSAR	L	None	None	False	None	None	None	None
3	22202	2013	2013	10581182613	MUHAMMAD MULHADIF	L	None	None	False	None	None	None	None
4	22202	2013	2013	10581182713	ST FATIMA	L	20172	None	True	4 Tahun, 8 Bulan	WIRASWASTA	500001-1000000	None
5	22202	2013	2013	10581182813	SAIFUL ROZAQ	L	20191	None	True	6 Tahun, 2 Bulan	WIRASWASTA	500001-1000000	None
6	22202	2013	2013	10581182913	SIPRI	L	None	None	False	None	None	None	None
7	22202	2013	2013	10581183013	HEHYA ADI SAPUTRA	L	20182	None	True	3 Tahun, 9 Bulan	WIRASWASTA	2000001-3000000	None
8	22202	2013	2013	10581183113	MUHAMMAD HABIR	L	None	None	False	None	TANI	500000	MEN
9	22202	2013	2013	10581183213	HILMAH AFANDI TASLIM	L	None	None	False	None	TANI	500001-1000000	None

```
# Validasi data yang sudah di download dengan NIM yang dicari  
nim_yang_dicari = '105821100817' # 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: 105821100817

=====

kodeProdi: 20201
angkatan: 2017
semesterAwal: 20171
nim: 105821100817
nama: ASWIN DWI PUTRA
jenisKelamin: L
tahunAkademikLulus: 20222
tanggalLulus: None
lulus: True
masaStudi: 5 Tahun, 11 Bulan
ayah_pekerjaan: TANI
ayah_penghasilan: 500000
ayah_status: HIDUP
ibu_pekerjaan: -
ibu_penghasilan: 500000
ibu_status: HIDUP
wali_pekerjaan:
wali_penghasilan:
khs1_tahunAkademik: 20171
khs1_ips: 3.36
khs1_sksSmt: 22.0
khs1_ipk: 3.36
khs1_sksTotal: 22.0
...
khs19_sksSmt: nan
khs19_ipk: nan
khs19_sksTotal: nan
khs19_statusMahasiswa: nan

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

+ Code

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

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())
```

	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())
```

	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  
0 2013 20131 10581102313 MUHAMMAD FARID HIKSYAL L laki laki 500000 HIDUP - 500000 laki laki  
1 2013 20131 10581102413 MUHAMMAD SAQIB L laki laki 500000 HIDUP - 500000 laki laki  
2 2013 20131 10581102513 FAISAL ANBAR L laki laki 500000 HIDUP - 500000 laki laki  
3 2013 20131 10581102613 MUNIR MUHAMMAD L laki laki 500000 HIDUP - 500000 laki laki  
4 2013 20131 10581102713 ST FATIMA P MIRASWASTA 500001-1000000 HIDUP MIRASWASTA 500001-1000000 laki laki  
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	nama	nama_serikat	jenis_kelamin	ayah_pekerjaan	ayah_penghasilan	ayah_status	ibu_pekerjaan	ibu_penghasilan	ibu_status
0	2013	2013110501102213	MUHAMMAD FARYO HEKSYAL	L	TAJIR	500000	HIDUP	-	500000	HIDUP
1	2013	20131110501102413	MUHAMMAD SAKIR	L	TAJIR	500000	HIDUP	-	500000	HIDUP
2	2013	20131110501102513	FAISAL AKBAR	L	TAJIR	100000	HIDUP	-	500000	HIDUP
3	2013	20131110501102613	MUHAMMAD MUHAMMAD	L	TAJIR	500000	HIDUP	-	500000	HIDUP
4	2013	20131110501102713	SI FATMA	P	WIRASWASTA	500001-1000000	HIDUP	WIRASWASTA	500001-1000000	HIDUP

```
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")
```

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	asal_pekerjaan	ayah_penghasilan	ayah_status	ibu_pekerjaan	ibu_penghasilan	ibu_status	wali_pekerjaan	wali_penghasilan	lulus	semesterT
0	2013	TAHI	500000	HIDUP	-	500000	HIDUP	WIRASWASTA	500000	False	
1	2013	TAHI	500000	HIDUP	-	500000	HIDUP	WIRASWASTA	500000	True	
2	2013	TAHI	900000	HIDUP	-	900000	HIDUP	WIRASWASTA	900000	False	
3	2013	TAHI	500000	HIDUP	-	500000	HIDUP	WIRASWASTA	500000	False	
4	2013	WIRASWASTA	500000-1000000	HIDUP	WIRASWASTA	500000-1000000	HIDUP	WIRASWASTA	500000	True	

```
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	seme
0	2013	20131	2	0	0	5	0	0	0	0	0	0
1	2013	20131	2	0	0	5	0	0	0	0	0	1
2	2013	20131	2	0	0	5	0	0	0	0	0	0
3	2013	20131	2	0	0	5	0	0	0	0	0	0
4	2013	20131	0	0	0	0	0	0	0	0	0	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 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())
```

```
   angkatan  semesterAwal  ayah_pekerjaan  ayah_penghasilan  ayah_status  ibu_pekerjaan  ibu_penghasilan  ibu_status  wali_pekerjaan  wali_penghasilan  lulus  semester  
0  2013      20131      2      0      0      5      0      0      0      0      0      0  
1  2013      20131      2      0      0      5      0      0      0      0      0      1  
2  2013      20131      1      0      0      5      0      0      0      0      0      0  
3  2013      20131      2      0      0      1      0      0      0      0      0      0  
4  2013      20131      0      0      0      0      0      0      0      0      0      1
```

```
# 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())
```

```
   angkatan  semesterAwal  ayah_pekerjaan  ayah_penghasilan  ayah_status  ibu_pekerjaan  ibu_penghasilan  ibu_status  wali_pekerjaan  wali_penghasilan  lulus  semester  
0  2013      20131      2      0      0      5      0      0      0      0      0      0  
1  2013      20131      2      0      0      5      0      0      0      0      0      1  
2  2013      20131      1      0      0      5      0      0      0      0      0      0  
3  2013      20131      2      0      0      1      0      0      0      0      0      0  
4  2013      20131      0      0      0      0      0      0      0      0      0      1
```

```

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=['potensi_putus_studi'])
y = df['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['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.naive_bayes import GaussianNB
from sklearn.metrics import classification_report, accuracy_score

# 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=['potensi_putus_studi'])
y = df['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

# Inisialisasi dan latih model Naive Bayes
model = GaussianNB()
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:.2f}')
print('Laporan Klasifikasi:')
print(report)

Akurasi: 0.97
Laporan Klasifikasi:
              precision    recall  f1-score   support

0               0.99         0.95         0.97         159
1               0.95         0.99         0.97         140

 accuracy          0.97
 macro avg         0.97
 weighted avg     0.97

import pandas as pd
from sklearn.model_selection import cross_val_score, StratifiedKFold
from sklearn.naive_bayes import GaussianNB
import numpy as np

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

# Inisialisasi model Gaussian Naive Bayes
model = GaussianNB()

# 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: 0.9639
Fold 2: 0.9424
Rata-rata akurasi: 0.9531
Standar deviasi akurasi: 0.0107

```

```

Akurasi untuk k=3:
Fold 1: 0.9558
Fold 2: 0.9618
Fold 3: 0.9416
Rata-rata akurasi: 0.9531
Standar deviasi akurasi: 0.0085

```

```

Akurasi untuk k=4:
Fold 1: 0.9545
Fold 2: 0.9678
Fold 3: 0.9491
Fold 4: 0.9410
Rata-rata akurasi: 0.9531
Standar deviasi akurasi: 0.0098

```

```

Akurasi untuk k=5:
Fold 1: 0.9532
Fold 2: 0.9666
Fold 3: 0.9599
Fold 4: 0.9497
Fold 5: 0.9362

```

```

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
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 'status_putus_studi'
X = df.drop(columns=['potensi_putus_studi'])
y = df['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 Gaussian Naive Bayes
model = GaussianNB()

# 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("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}")
print(f"Rata-rata Recall: {avg_recall}")
print(f"Rata-rata F1-Score: {avg_f1_score}")

# 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 - Naive Bayes')
plt.show()

```

```

Confusion Matrix:
[[151  8]
 [ 2 138]]
Classification Report:

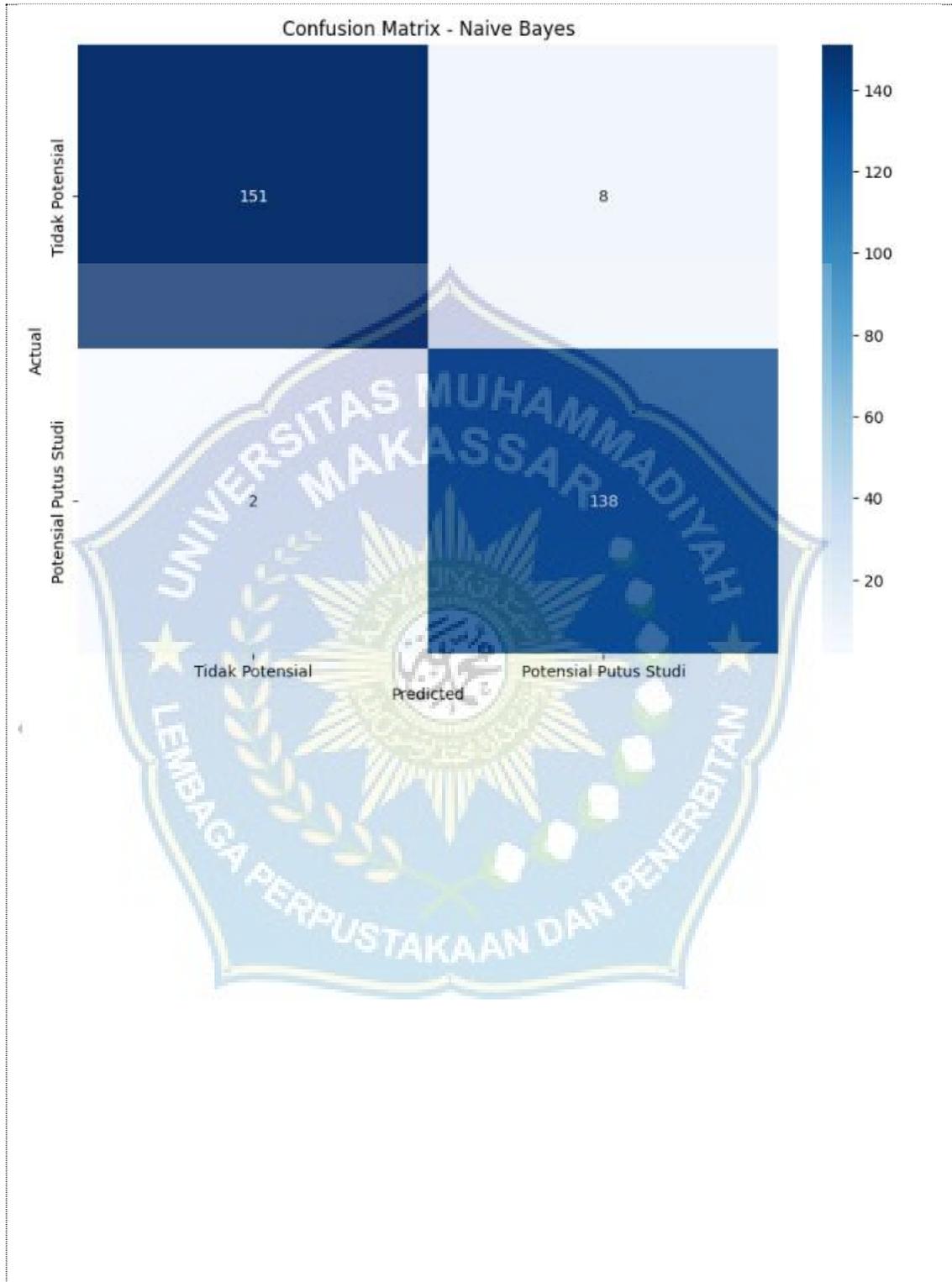
```

	precision	recall	f1-score	support
0	0.99	0.95	0.97	159
1	0.95	0.99	0.97	140
accuracy			0.97	299
macro avg	0.97	0.97	0.97	299
weighted avg	0.97	0.97	0.97	299

```

Rata-rata Precision: 0.9660667920136091
Rata-rata Recall: 0.9676999101527404
Rata-rata F1-Score: 0.9664918414918415

```



Lampiran 5. Permohonan Penelitian

SURAT PERMOHONAN PENELITIAN

Hal : Permohonan Surat Pengantar Penelitian

Kepada Yth,

Ketua Program Studi Informatika

Di

Tempat

Assalamu Alaikum Warahmatullahi Wabarakatuh

Schubungan dengan akan dilaksanakanya Penelitian untuk penyelesaian tugas akhir dengan judul penelitian "*Klasifikasi Mahasiswa Berpotensi Pulus Studi Menggunakan Algoritma Naive Bayes 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	RIANITA KAMSURYA	105841102620

Maka dengan ini kami memohon dibuatkas 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 Sabillillah, Fastabiqul Khairat

Walaikumussalam Warahmatullahi Wabarakatuh

Makassar, 7 Dzulhijjah 1445 H
14 Juni 2024 M

Permohonan


RIANITA KAMSURYA
105841102620

Lampiran 7. Surat Keterangan Bebas Plagiat



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

Alamat Kantor: Jl. Sultan Alauddin No. 259 Makassar 90221 Telp. (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 : Rianita kamsurya

Nim : 105841102620

Program Studi : Teknik Informatika

Dengan nilai:

No	Bab	Nilai	Ambang Batas
1	Bab 1	10 %	10 %
2	Bab 2	24 %	25 %
3	Bab 3	9 %	10 %
4	Bab 4	9 %	10 %
5	Bab 5	5 %	5 %

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

Demikian surat keterangan ini diberikan kepada yang bersangkutan untuk dipergunakan sebagaimana mestinya.

Makassar, 22 Agustus 2024

Mengetahui,

Kepala UPT- Perpustakaan dan Penerbitan,



Ningsih, S.Pd, M.P
NPM 964 591

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

Lampiran 8. Hasil Plagiat Per BAB



BAB I Rianita kamsurya -
105841102620

by Tahap Tutup

ssion date: 22-Aug-2024 01:40PM (UTC+0700)

ssion ID: 2435961392

me: SKRIPSI_BAB_1_ANITA.docx (349.79K)

ount: 1544

ter count: 9602

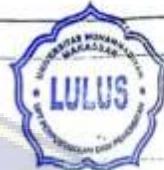
AB I Rianita kamsurya - 105841102620

ORIGINALITY REPORT

10% SIMILARITY INDEX
10% INTERNET SOURCES
6% PUBLICATIONS
0% STUDENT PAPERS

PRIMARY SOURCES

Rank	Source	Percentage
1	publikasi.dinus.ac.id Internet Source	3%
2	ejournal.fikom-unasman.ac.id Internet Source	3%
3	vdocuments.pub Internet Source	2%
4	eprints.ubhara.ac.id Internet Source	2%



Exclude quotes
 Exclude bibliography

Exclude matches



BAB II Rianita kamsurya - 105841102620

by Tahap Tutuo



upload date: 22 Aug 2014 01:42:54 UTC+07:00

upload ID: 2435763399

title: SKRIPSI_BAB_II_RIANITA (file 106.729)

count: 2409

file count: 17348

AB II Rianita kamsurya - 105841102620

ORIGINALITY REPORT

24% SIMILARITY INDEX
24% INTERNET SOURCES
5% PUBLICATIONS
0% STUDENT PAPERS

INTERNET SOURCES



 digilib.ikipgriptk.ac.id Internet Source	4%
 repository.ulb.ac.id Internet Source	3%
 penerbitadm.com Internet Source	3%
 e-journals.unmul.ac.id Internet Source	3%
 idwebhost.com Internet Source	2%
 citisee.amikompurwokerto.ac.id Internet Source	2%
 e-journal.hamzanwadi.ac.id Internet Source	2%
 jacis.pubmedia.id Internet Source	2%
 jurnal.umk.ac.id Internet Source	2%



www.researchgate.net
Internet Source

2%

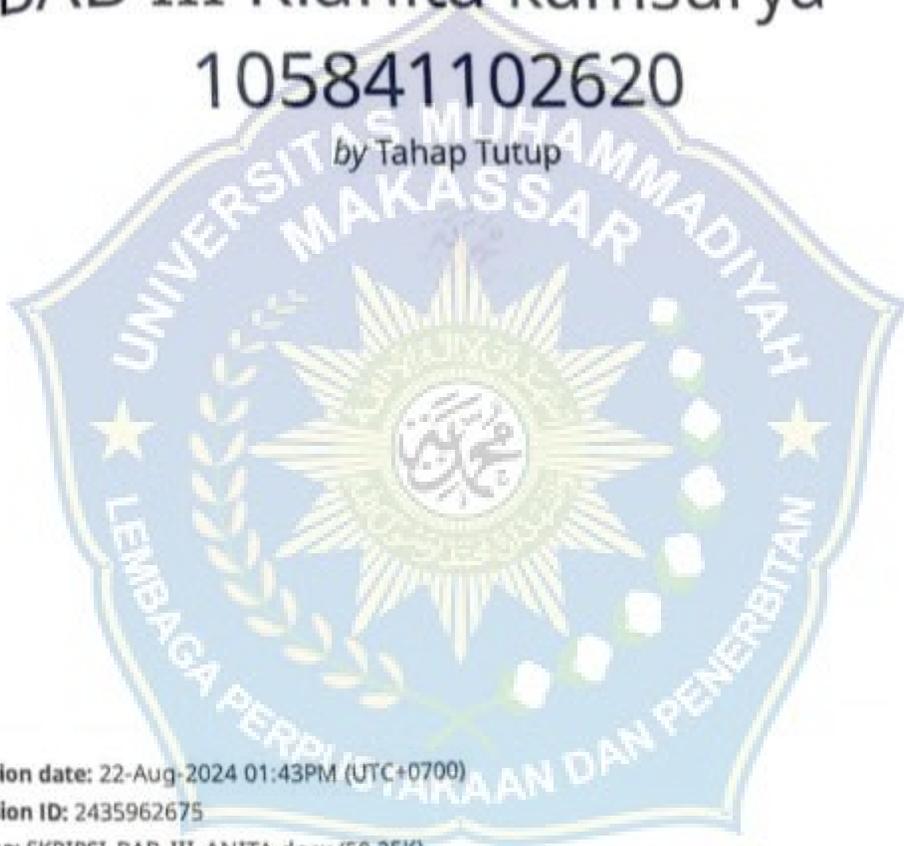
include quotes On
include bibliography On

Exclude matches < 2%



BAB III Rianita kamsurya - 105841102620

by Tahap Tutup



Submission date: 22-Aug-2024 01:43PM (UTC+0700)

Submission ID: 2435962675

File name: SKRIPSI_BAB_III_ANITA.docx (50.25K)

Word count: 906

Character count: 5837

9% SIMILARITY INDEX 9% INTERNET SOURCES 4% PUBLICATIONS % STUDENT PAPERS

PRIMARY SOURCES

1	repository.unpas.ac.id Internet Source	2%
2	repository.pelitabangsa.ac.id Internet Source	2%
3	openlibrarypublications.telkomuniversity.ac.id Internet Source	2%
4	www.djournals.com Internet Source	2%
5	elib.pnc.ac.id Internet Source	2%

Exclude matches

Exclude quotes

Exclude bibliography

BAB IV Rianita kamsurya - 105841102620

by Tahap Tutup



Creation date: 22-Aug-2024 01:44PM (UTC+0700)

Creation ID: 2435962942

File name: SKRIPSI_BAB_IV_ANITA.docx (967.23K)

Page count: 2205

Character count: 13998

AB IV Rianita kamsurya - 105841102620

ORIGINALITY REPORT

9% SIMILARITY INDEX
8% INTERNET SOURCES
6% PUBLICATIONS
% STUDENT PAPERS

PRIMARY SOURCES

- 
- 1 digilibadmin.unismuh.ac.id
Internet Source 4%
- 2 Nuru Aini, Muchamad Arif, Inka Tri Agustini,
Zulfah Binti Toyibah. "Implementasi Algoritma
Random Forest untuk Klasifikasi Bidang MSIB
di Prodi Pendidikan Informatika", Jurnal
Informatika, 2024
Publication 2%
- 3 jurnal.untan.ac.id
Internet Source 2%
- 4 repository.ub.ac.id
Internet Source 2%

include quotes On
include bibliography On

Exclude matches < 2%

BAB V Rianita kamsurya - 105841102620

by Tahap Tutup



mission date: 22-Aug-2024 01:46PM (UTC+0700)

mission ID: 2435963448

name: SKRIPSI_BAB_V_ANITA.docx (8.36K)

word count: 166

character count: 1121

BAB V Rianita kamsurya - 105841102620

ORIGINALITY REPORT

5% SIMILARITY INDEX	5% INTERNET SOURCES	0% PUBLICATIONS	% STUDENT PAPERS
-------------------------------	-------------------------------	---------------------------	----------------------------

PRIMARY SOURCES

 eprints.perbanas.ac.id Internet Source	5%
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------



Exclude quotes On
Exclude bibliography On

Exclude matches On



