

**PENGEMBANGAN SISTEM CERDAS PREDIKSI KELULUSAN
MAHASISWA TEPAT WAKTU MENGGUNAKAN ALGORITMA
NAIVE BAYES DAN *K-NEAREST NEIGHBORS***

TUGAS AKHIR

Diajukan oleh:

YUDHI JAUHARI

210705090

Mahasiswa Fakultas Sains dan Teknologi

Program Studi Teknologi Informasi



**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGRI AR-RANIRY**

BANDA ACEH

2026 M/1447 H

LEMBAR PERSETUJUAN

PENGEMBANGAN SISTEM CERDAS PREDIKSI KELULUSAN MAHASISWA TEPAT WAKTU MENGGUNAKAN ALGORITMA NAIVE BAYES DAN K-NEAREST NEIGHBORS

TUGAS AKHIR

Diajukan kepada Fakultas Sains dan Teknologi
Universitas Islam Negeri Ar-Raniry Banda Aceh
Sebagai Salah Satu Beban Studi Memperoleh Gelar Sarjana (S1)
Dalam Prodi Teknologi Informasi

Oleh:

YUDHI JAUHARI

NIM. 210705090

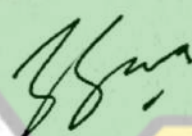
Mahasiswa Fakultas Sains dan Teknologi

Program Studi Teknologi Informasi

Disetujui Untuk Munaqasyah Oleh:

Pembimbing I,

Pembimbing II,

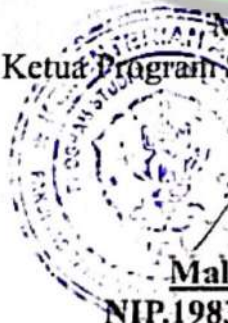
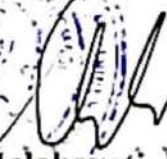

Khairan AR, M.Kom

NIP.198607042014031001.


Bustami, M.Sc

NIP. 198604082014031001

Mengetahui,
Ketua Program Studi Teknologi Informasi



Malahayati, M.T
NIP.198301272015032003

LEMBAR PENGESAHAN

PENGEMBANGAN SISTEM PREDIKSI KELULUSAN MAHASISWA TEPAT WAKTU MENGGUNAKAN ALGORITMA NAIVE BAYES DAN K-NEAREST NEIGHBORS (K-NN)

TUGAS AKHIR

Telah Diuji Oleh Panitia Ujian Munaqasyah Tugas Akhir
Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh dan Dinyatakan Lulus
Serta Diterima Sebagai Salah Satu Beban Studi Program Sarjana (S-1)
Dalam Program Studi Teknologi Informasi

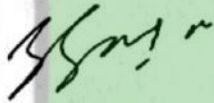
Pada Hari/Tanggal: Jum'at, 30 Januari 2026

11 Sya'ban 1447 H

Di Darussalam, Banda Aceh

Panitia Ujian Munaqasyah Tugas Akhir:

Ketua,



Khairan AR, M.Kom
NIP. 198607042014031001

Sekretaris,



Bustami, M.Sc
NIP. 198604082014031001

Penguji I,



Mulkan Fadhli, M.T.
NIP. 198811282020121006

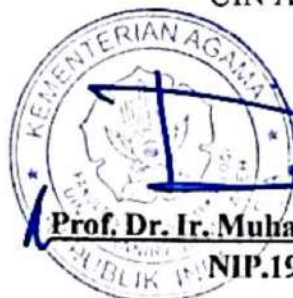
Penguji II,



Nurrizqa, M.T
NIP. 199704302025052001

Mengetahui:

Dekan Fakultas Sains dan Teknologi
UIN Ar-Raniry Banda Aceh,



Prof. Dr. Ir. Muhammad Dirhamyah, M.T., IPU
NIP.196210021988111001

LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Yudhi Jauhari
NIM : 210705090
Program Studi : Teknologi Informasi
Fakultas : Sains dan Teknologi
Judul : PENGEMBANGAN SISTEM CERDAS PREDIKSI
KELULUSAN MAHASISWA TEPAT WAKTU
MENGUNAKAN ALGORITMA NAÏVE BAYES DAN
K-NEAREST NEIGHBORS (K-NN)

Dengan ini menyatakan bahwa dalam penulisan tugas akhir ini, saya:

1. Tidak menggunakan ide orang lain tanpa mampu mengembangkan dan mempertanggungjawabkan;
2. Tidak melakukan plagiasi terhadap naskah karya orang lain;
3. Tidak menggunakan karya orang lain tanpa menyebutkan sumber asli atau tanpa izin pemilik karya;
4. Tidak memanipulasi dan memalsukan data;
5. Mengerjakan sendiri karya ini dan mampu bertanggungjawab atas karya ini.

Bila dikemudian hari ada tuntutan dari pihak lain atas karya saya, dan telah melalui pembuktian yang dapat dipertanggungjawabkan dan ternyata memang ditemukan bukti bahwa saya telah melanggar pernyataan ini, maka saya siap dikenai sanksi berdasarkan aturan yang berlaku di Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh. Demikian pernyataan ini saya buat dengan sesungguhnya dan tanpa paksaan dari pihak manapun.

Banda Aceh, 30 Januari 2026



Yang Menyatakan

Yudhi Jauhari

KATA PENGANTAR

Bismillahirrahmanirrahim.

Dengan menyebut nama Allah Yang Maha Pengasih lagi Maha Penyayang. Puji syukur penulis panjatkan kehadiran Allah SWT yang atas berkat, rahmat, dan karunia-Nya telah memberikan kekuatan dan kemudahan, sehingga penulis berhasil merampungkan Tugas Akhir yang berjudul **“PENGEMBANGAN SISTEM CERDAS PREDIKSI KELULUSAN MAHASISWA TEPAT WAKTU MENGGUNAKAN ALGORITMA NAIVE BAYES DAN K-NEAREST NEIGHBORS”**. Shalawat dan salam semoga senantiasa terlimpahkan kepada teladan umat manusia, Nabi Muhammad SAW, beserta keluarga, sahabat, dan para pengikutnya hingga akhir zaman.

Penulisan Tugas Akhir ini diajukan untuk memenuhi salah satu syarat akademis dalam memperoleh gelar Sarjana (S1) pada Program Studi Teknologi Informasi, Fakultas Sains dan Teknologi, Universitas Islam Negeri Ar-Raniry Banda Aceh.

Penulis menyadari bahwa perjalanan dan penyelesaian studi ini tidak lepas dari bimbingan, dukungan, dan doa dari banyak pihak. Untuk itu, dengan penuh rasa hormat dan terima kasih, penulis ingin menghaturkan penghargaan yang tulus kepada:

1. Ayahanda Dedi Miswar dan Ibunda Rahayu Dewi pilar utama kekuatan dan sumber inspirasi. Terima kasih atas limpahan kasih sayang yang tiada henti, doa yang tak pernah terputus, serta dukungan moril dan materiel yang menjadi fondasi bagi penulis dalam menempuh pendidikan hingga titik ini. Semoga Allah SWT senantiasa melimpahkan karunia kesehatan dan kebahagiaan untuk ayah dan ibu.
2. Ibu Malahayati, M.T., selaku Ketua Program Studi Teknologi Informasi. Terima kasih yang sebesar-besarnya atas kesabaran, waktu, serta bimbingan akademis yang sangat berharga. Arahkan, masukan, dan

diskusi yang diberikan Beliau sangat membantu dalam mengarahkan penulis dalam penelitian ini.

3. Bapak Khairan AR, M.Kom. , selaku Pembimbing I, yang telah banyak meluangkan waktu untuk memberikan bimbingan, saran, dan diskusi yang mencerahkan sehingga penelitian ini dapat terselesaikan dengan lebih baik.
4. Bapak Bustami, M.Sc. , selaku Pembimbing II, yang dengan sabar memberikan arahan, masukan, dan dukungan intelektual sejak awal proses penelitian hingga selesai.
5. Ibu Cut Ida Rahmadiana, S.Si., atas segala bantuan, kemudahan, dan kesabaran dalam proses pengurusan administrasi akademik yang sangat menunjang kelancaran studi penulis.
6. Bapak Prof . Dr. Ir. M. Dirhamsyah, M.T., IPU, selaku Dekan Fakultas Sains dan Teknologi UIN Ar-Raniry, atas segala dukungan dan fasilitas yang diberikan selama masa studi.
7. Seluruh Bapak dan Ibu Dosen di lingkungan Program Studi Teknologi Informasi yang telah membagikan ilmu pengetahuan dan wawasan yang tak ternilai.
8. Sahabat dan rekan perjuangan : M.Azuwan Rafsanjani, Hadi Albar dan Arief Fathin Abrar, yang telah menjadi teman berbagi dalam suka, duka, dan segala proses yang dijalani.
9. Semua pihak yang tidak dapat penulis sebutkan satu per satu, yang telah memberikan kontribusi dalam penyelesaian Tugas Akhir ini.

Penulis mengakui bahwa karya ini masih memiliki banyak kekurangan. Oleh karena itu, segala bentuk saran dan kritik yang bersifat membangun akan penulis terima dengan lapang dada demi penyempurnaan di masa yang akan datang.

Akhir kata, semoga Tugas Akhir ini dapat bermanfaat bagi penulis, pembaca, dan pengembangan ilmu pengetahuan. Semoga Allah SWT mencatatnya sebagai amal kebaikan.

Aamiin yaa Rabbal'aalamiin.

Banda Aceh. 28 Januari 2026

Penulis,

Yudhi Jauhari



ABSTRAK

Nama	: Yudhi Jauhari
NIM	: 210705090
Program Studi	: Teknologi Informasi
Fakultas	: Sains dan Teknologi
Judul Tugas Akhir	: Pengembangan Sistem Cerdas Prediksi Kelulusan Mahasiswa Tepat Waktu Menggunakan Algoritma <i>Naive Bayes</i> dan <i>K-Nearest Neighbors</i> (K-NN)
Tanggal Sidang	: 30 Januari 2026
Jumlah Halaman	: 72
Pembimbing I	: Khairan AR, M.Kom
Pembimbing II	: Bustami, M.Sc

Kelulusan tepat waktu mahasiswa menjadi salah satu indikator penting dalam menilai mutu penyelenggaraan pendidikan tinggi. Namun, rendahnya persentase mahasiswa yang mampu menyelesaikan studi sesuai masa studi ideal masih menjadi permasalahan yang perlu mendapat perhatian serius. Penelitian ini bertujuan untuk mengembangkan sistem cerdas guna memprediksi kelulusan mahasiswa tepat waktu dengan menerapkan algoritma *Naive Bayes* dan *K-Nearest Neighbors* (K-NN). Dataset yang digunakan bersumber dari Sistem Informasi Akademik UIN Ar-Raniry Banda Aceh, terdiri dari 1.828 data mahasiswa dari Fakultas Sains dan Teknologi dan Fakultas Ilmu Sosial dan Ilmu Pemerintahan tahun angkatan 2018 hingga 2021. Tahapan penelitian mencakup pengumpulan data, *preprocessing*, pembentukan target biner, pembagian data latih dan uji menggunakan *stratified split*, pelatihan model, serta evaluasi performa. Evaluasi dilakukan menggunakan *confusion matrix*, *classification report*, *learning curve*, *ROC curve*, dan nilai AUC. Hasil penelitian menunjukkan bahwa model *Naive Bayes* mencapai akurasi sebesar 79,23%, sementara model K-NN memperoleh akurasi sebesar 92,07%. Penggabungan kedua hasil model dalam sistem *hybrid*

berhasil meningkatkan kinerja prediksi secara keseluruhan. Temuan ini menunjukkan bahwa pendekatan *soft voting* efektif dalam mengklasifikasikan status kelulusan mahasiswa. Sistem yang dikembangkan diharapkan dapat membantu pihak akademik dalam mengidentifikasi mahasiswa yang berpotensi mengalami keterlambatan kelulusan sejak dini, sehingga pembinaan akademik dapat dilakukan secara lebih tepat dan terarah.

Kata kunci: Sistem Cerdas, Prediksi Kelulusan Mahasiswa, *Naive Bayes*, *K-Nearest Neighbors*, Hybrid Model.



ABSTRACT

Name : Yudhi Jauhari
Student ID : 210705090
Major : Information Technology
Faculty : Science and Technology
Final Project Title : Development of an Intelligent
System for Predicting Timely Student
Graduation Using *Naive Bayes* dan
K-Nearest Neighbors (K-NN)
Thesis Defence Date : 30 January 2026
Number Of Pages : 72
Supervisor I : Khairan AR, M.Kom
Supervisor II : Bustami, M.Sc

Timely graduation is one of the important indicators in assessing the quality of higher education implementation. However, the low percentage of students who are able to complete their studies within the ideal study period remains a significant issue that requires serious attention. This study aims to develop an intelligent system to predict timely student graduation by applying the Naive Bayes and K-Nearest Neighbors (K-NN) algorithms. The dataset used in this research was obtained from the Academic Information System of UIN Ar-Raniry Banda Aceh and consists of 1,828 student records from the Faculty of Science and Technology and the Faculty of Social and Political Sciences, covering the 2018 to 2021 cohorts. The research stages include data collection, preprocessing, binary target formation, splitting training and testing data using stratified split, model training, and performance evaluation. The evaluation was conducted using a confusion matrix, classification report, learning curve, ROC curve, and AUC value. The results show that the Naive Bayes model achieved an accuracy of 78.69%, while the K-NN model obtained a higher accuracy of 91.80%. The integration of both models into a hybrid system successfully improved the overall prediction performance. These findings

indicate that the soft voting approach is effective in classifying student graduation status. The developed system is expected to assist academic stakeholders in identifying students who are at risk of delayed graduation at an early stage, enabling more targeted and effective academic interventions.

Keywords: *Intelligent System, Student Graduation Prediction, Naive Bayes, K-Nearest Neighbors, Hybrid Model.*



DAFTAR ISI

DAFTAR ISI	xi
DAFTAR TABEL	xiii
DAFTAR GAMBAR	xiv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Ruang Lingkup Penelitian	4
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	5
1.5.1 Bagi peneliti :	5
1.5.2 Bagi mahasiswa :	5
1.5.3 Bagi pihak akademik (Program Studi, Fakultas, dan Universitas) : ..	5
BAB II LANDASAN TEORI	6
2.1 Penelitian Terdahulu	6
2.2 Landasan Teori	9
2.2.1 Sistem Cerdas	9
2.2.2 Data Mining	11
2.2.3 Klasifikasi	12
2.2.4 Naive Bayes	13
2.2.5 K-Nearest Neighbors (K-NN)	13
2.2.6 Ensemble Learning dan Hybrid Model	14
2.2.7 Evaluasi Performa Model Klasifikasi	16
2.2.8 Python dan Google Colaboratory	19
2.2.9 Kelulusan Mahasiswa	20
2.3 Kerangka Teoritis	20
BAB III METODE PENELITIAN	22
3.1 Tahapan Penelitian	22
3.1.1 Pengumpulan Data	23
3.1.2 Analisis Karakteristik Dataset	24
3.1.3 Preprocessing Data	25

3.1.4	Algoritma <i>Naive Bayes</i>	28
3.1.5	Algoritma K-Nearest Neighbors	30
3.1.6	Penggabungan Algoritma <i>Naive Bayes</i> dan K-NN.....	33
3.1.8	Klasifikasi Data Baru (Inference)	35
3.2	Alat Bantu Penelitian	35
3.3	Waktu dan Lokasi Penelitian.....	36
BAB IV HASIL DAN PEMBAHASAN.....		37
4.1	Implementasi Sistem dan Pengolahan Data	37
4.2	Hasil Pra-Pemrosesan Data (Preprocessing Data)	38
4.2.1	Hasil Pembersihan Data (Cleaning Data)	38
4.2.2	Pembentukan Target Biner.....	38
4.2.3	Standarisasi data dan Penanganan Missing Value	39
4.3	Hasil Implementasi Algoritma <i>Naive Bayes</i>	39
4.3.1	Hasil Evaluasi Model Algoritma <i>Naive Bayes</i>	40
4.4	Hasil Implementasi Algoritma K-NN	44
4.4.1	Hasil Evaluasi Model Algoritma K-NN.....	45
4.5	Perbandingan Hasil Penelitian dengan Penelitian Terdahulu.....	49
4.5.1	Perbandingan Model <i>Naive Bayes</i>	49
4.5.2	Perbandingan Model <i>K-Nearest Neighbors</i>	50
4.6	Hasil Implementasi Model Hybrid (<i>Naive Bayes</i> dan K-NN)	51
4.6.1	Hasil Evaluasi Model Hybrid	52
4.7	Ringkasan Performa Model Klasifikasi	55
BAB V PENUTUP		57
5.1	Kesimpulan	57
5.2	Saran.....	58
DAFTAR PUSTAKA.....		59

DAFTAR TABEL

Tabel 1. 1 Data Jumlah Mahasiswa.....	2
Tabel 1. 2 Penelitian Terdahulu.....	6
Tabel 3. 1 Data Mahasiswa	23
Tabel 4. 1 Dataset Mahasiswa SAINTEK dan FISIP.....	37
Tabel 4. 4 Perbandingan Performa Model.....	55



DAFTAR GAMBAR

Gambar 2. 1 Confusion Matrix	16
Gambar 2. 2 Kerangka Teoritis	21
Gambar 3. 1 Diagram Proses Preprocessing Data	25
Gambar 3. 2 Diagram Penerapan Algoritma Naive Bayes.....	28
Gambar 3. 3 Diagram Penerapan Algoritma K-NN	31
Gambar 3. 4 Diagram Penggabungan Algoritma	33
Gambar 4. 1 Hasil Evaluasi Model Naive Bayes	40
Gambar 4. 2 Confusion Matrix Model Naive Bayes.....	41
Gambar 4. 3 Grafik Learning Curve Model Naive Bayes	42
Gambar 4. 4 Grafik ROC Curve Model Naive Bayes.....	44
Gambar 4. 5 Hasil Evaluasi Model K-NN	45
Gambar 4. 6 Confusion Matrix Model K-NN.....	46
Gambar 4. 7 Grafik Learning Curve Model K-NN.....	47
Gambar 4. 8 Grafik ROC Curve Model K-NN.....	48
Gambar 4. 9 Grafik Validation Curve Model K-NN.....	49
Gambar 4. 10 Hasil Evaluasi Model Hybrid.....	52
Gambar 4. 11 Confusion Matrix Model Hybrid.....	52
Gambar 4. 12 Learning Curve Model Hybrid.....	53
Gambar 4. 13 Grafik ROC Curve Model Hybrid.....	54

BAB I

PENDAHULUAN

1.1 Latar Belakang

Agar lembaga pendidikan dapat meningkatkan kualitas lulusannya, upaya pengurangan waktu hingga kelulusan tepat waktu harus menjadi prioritas (Weiss & Robertson, 2020). Lulus tepat waktu berarti mahasiswa mampu menyelesaikan studinya sesuai waktu studi yang ditentukan dalam rencana studinya. Menganalisis lama waktu studi, kinerja akademik mahasiswa, serta langkah-langkah yang dapat diambil di masa depan menjadi tantangan besar bagi universitas-universitas modern saat ini.

Salah satu tanda kemajuan perguruan-perguruan tinggi saat ini adalah meningkatnya jumlah mahasiswa yang lulus tepat waktu. Ini menunjukkan bahwa proses akademik berjalan secara efisien dan efektif (Zainuddin, 2022). Lulus tepat waktu juga menjadi kriteria penting bagi pengelola akademik, seperti klasifikasi lulusan membantu universitas meminimalisasi kegagalan gelar mahasiswa melalui perencanaan yang lebih terfokus, pemantauan penelitian, pengawasan, dan bimbingan akademik yang lebih efektif.

Penelitian yang dilakukan oleh Subariah & Zein (2023) menunjukkan bahwa metode *Naïve Bayes* dan *K-Nearest Neighbors* (K-NN) efektif digunakan dalam memprediksi kelulusan mahasiswa tepat waktu. Hasil penelitian tersebut membuktikan bahwa kedua algoritma mampu menghasilkan tingkat akurasi yang baik dalam proses klasifikasi data akademik mahasiswa sehingga layak digunakan sebagai metode prediksi kelulusan.

Data-data mahasiswa Fakultas Sains dan Teknologi dan Fakultas Ilmu Sosial dan Ilmu Pemerintahan UIN Ar-Raniry Banda Aceh dikutip dari Data Sistem Informasi Akademik Kampus Terintegrasi (Siakad, 2025) sampai dengan tahun ajaran 2023/2024 adalah sebanyak 1828 Mahasiswa (tidak termasuk mahasiswa yang sudah putus kuliah/drop out). Rincian dari jumlah mahasiswa tersebut ditampilkan pada tabel 1.1.

Tabel 1. 1 Data Jumlah Mahasiswa

Tahun Masuk Fakultas	2018	2019	2020	2021	TOTAL
SAINTEK	361	292	167	308	1.128
FISIP	201	177	108	214	700
TOTAL					1.828

Berdasarkan tabel 1.1, pengelompokan dilakukan ke dalam dua kategori, yaitu mahasiswa yang lulus tepat waktu dan mahasiswa yang belum lulus atau lulus tidak tepat waktu. Pada Fakultas Sains dan Teknologi, untuk angkatan 2018 tercatat sebanyak 115 mahasiswa lulus tepat waktu, sementara 246 mahasiswa belum lulus atau lulus tidak tepat waktu. Pada angkatan 2019, jumlah mahasiswa yang lulus tepat waktu mencapai 106 orang, sedangkan 186 mahasiswa termasuk dalam kategori belum lulus atau lulus tidak tepat waktu. Selanjutnya, pada angkatan 2020 terdapat 53 mahasiswa yang lulus tepat waktu dan 114 mahasiswa yang belum lulus atau lulus tidak tepat waktu. Untuk angkatan 2021, seluruh mahasiswa yang berjumlah 308 orang masih berstatus belum lulus.

Sementara itu, pada Fakultas Ilmu Sosial dan Ilmu Pemerintahan, angkatan 2018 menunjukkan bahwa sebanyak 106 mahasiswa lulus tepat waktu dan 95 mahasiswa belum lulus atau lulus tidak tepat waktu. Pada angkatan 2019, terdapat 79 mahasiswa yang lulus tepat waktu, sedangkan 98 mahasiswa berada pada kategori belum lulus atau lulus tidak tepat waktu. Untuk angkatan 2020, jumlah mahasiswa yang lulus tepat waktu tercatat sebanyak 39 orang, sementara 69 mahasiswa masih belum lulus. Adapun pada angkatan 2021, seluruh mahasiswa yang berjumlah 214 orang masih berada pada status belum lulus.

Jumlah data kelulusan mahasiswa tepat waktu yang disebutkan diatas, tergolong kecil jika dibandingkan dengan jumlah mahasiswa disetiap angkatannya. Untuk mencapai kelulusan tepat waktu, persyaratan kelulusannya adalah menyelesaikan masa studi kuliah dalam jangka waktu 8 semester atau 4 tahun.

Untuk mengatasi permasalahan kecilnya angka kelulusan mahasiswa secara tepat waktu dilingkungan program studi Teknologi Informasi, teknik *machine learning* yang biasa digunakan adalah penggunaan *Data Mining* dengan metode yang biasa dipakai untuk memprediksi status kelulusan mahasiswa yaitu menggunakan klasifikasi. Permodelan ini dilakukan dengan metode prediksi probablistik seperti *Naive Bayes* dan K-NN (*K-Nearest Neighbors*) dapat berguna untuk mengukur tingkat kelulusan mahasiswa tepat waktu (Subariah & Zein, 2023).

Meskipun teknik klasifikasi adalah sebuah teknik pendekatan untuk mengklasifikasikan fitur data, data mining adalah bidang ilmiah yang tujuan utamanya adalah menemukan, mengeksplorasi, dan mengekstrak pengetahuan yang bernilai yang terdiri dari Kumpulan data dan informasi yang tersedia. Data mining biasa disebut dengan *Knowledge Discovery in Database* (KDD). KDD mencakup serangkaian proses, mulai dari pengumpulan hingga pemanfaatan data historis dengan tujuan untuk mengidentifikasi pola, keteraturan, atau hubungan dalam kumpulan data yang berukuran besar (Komang Aditya Pratama et al., 2020).

Pada penelitian ini, peneliti menggunakan dua algoritma klasifikasi, yaitu algoritma *Naive Bayes* dan K-NN (*K-Nearest Neighbors*). Hal ini dikarenakan kedua algoritma tersebut merupakan algoritma pengklasifikasian yang sangat terkenal dengan tingkat keakuratan yang sangat baik. Selain itu, berbeda dengan algoritma klasifikasi dengan regresi *logistic ordinal* maupun nominal yang memprediksi satu variabel dependen berdasarkan beberapa variabel independen, kedua algoritma ini tidak memerlukan adanya permodelan ataupun pengujian statistik seperti uji signifikansi.

Penelitian ini nantinya akan mengembangkan sebuah sistem cerdas untuk memprediksi kelulusan mahasiswa tepat waktu menggunakan komparasi dari hasil prediksi dua algoritma, yaitu *Naive Bayes* dan K-NN dengan ruang lingkup pada Fakultas Sains dan Teknologi dan Fakultas Ilmu Sosial dan Ilmu Pemerintahan UIN Ar-Raniry Banda Aceh. Sistem cerdas ini diharapkan mampu memberikan kontribusi dan dampak yang positif dalam upaya meningkatkan kelulusan tepat

waktu mahasiswa, serta dapat dijadikan acuan referensi bagi institusi pendidikan lain dalam menghadapi masalah yang serupa.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang masalah yang telah disebutkan sebelumnya, dapat diuraikan rumusan masalah sebagai berikut :

1. Bagaimana cara mengembangkan sistem cerdas yang dapat memprediksi kelulusan tepat waktu mahasiswa Fakultas Sains dan Teknologi dan Fakultas Ilmu Sosial dan Ilmu Pemerintahan UIN Ar-Raniry Banda Aceh menggunakan algoritma *Naive Bayes* dan *K-Nearest Neighbors* (K-NN)?
2. Bagaimana mengukur akurasi dan kinerja dari sistem cerdas prediksi kelulusan mahasiswa tepat waktu yang telah dibangun?

1.3 Ruang Lingkup Penelitian

1. Data yang akan digunakan dalam penelitian ini hanya mencakup data mahasiswa-mahasiswa Fakultas Sains dan Teknologi dan Fakultas Ilmu Sosial dan Ilmu Pemerintahan UIN Ar-Raniry Angkatan 2018, 2019, 2020 dan 2021, dan data tersedia sampai dengan tahun akademik 2022/2023.
2. Fitur atau variabel yang digunakan dalam model prediksi terbatas pada data akademik yang tersedia dalam sistem informasi akademik UIN Ar-Raniry, seperti Nilai Indeks Prestasi Semester, dan Tahun Masuk mahasiswa
3. Algoritma yang akan digunakan dalam penelitian ini adalah *Naive Bayes* dan K-NN (*K-Nearest Neighbors*), sehingga tidak mencakup perbandingan dengan algoritma klasifikasi atau prediksi lainnya.
4. Penelitian ini hanya berfokus pada pengembangan model prediksi dan tidak mencakup implementasi lengkap sistem cerdas dalam sistem informasi akademik yang ada.

1.4 Tujuan Penelitian

Berdasarkan latar belakang masalah serta rumusan masalah yang telah dijelaskan sebelumnya, tujuan penelitian ini adalah sebagai berikut:

1. Untuk mengembangkan sebuah sistem cerdas yang dapat memprediksi status kelulusan mahasiswa tepat waktu pada Fakultas Sains dan Teknologi dan Fakultas Ilmu Sosial dan Ilmu Pemerintahan UIN Ar-Raniry Banda Aceh menggunakan algoritma *Naive Bayes* dan K-NN.
2. Untuk mengukur akurasi dan kinerja dari sistem cerdas prediksi kelulusan tepat waktu mahasiswa yang dibangun.

1.5 Manfaat Penelitian

1.5.1 Bagi peneliti :

- a. Dapat mengimplementasikan dan menerapkan ilmu *data mining* dan *machine learning* yang telah dipelajari selama masa studi, khususnya algoritma *Naive Bayes* dan K-NN, dengan kasus nyata penerapan untuk memprediksi kelulusan tepat waktu mahasiswa.
- b. Meningkatkan kemampuan dan skill dalam mengembangkan sistem cerdas yang dapat memberikan solusi bagi permasalahan di dunia nyata.

1.5.2 Bagi mahasiswa :

- a. Membantu mengidentifikasi mahasiswa yang berisiko mengalami keterlambatan kelulusan sejak dini, sehingga dapat dicari solusi berupa upaya pencegahan dan pembimbingan.
- b. Membantu mendorong mahasiswa untuk meningkatkan kinerja akademik agar dapat lulus tepat waktu.

1.5.3 Bagi pihak akademik (Program Studi, Fakultas, dan Universitas) :

- a. Meningkatkan nilai kualitas lulusan dan daya saing institusi pendidikan.
- b. Memberikan informasi yang bermanfaat terkait pengambilan keputusan dan perencanaan strategis dalam upaya meningkatkan angka kelulusan tepat waktu mahasiswa.
- c. Membantu mengidentifikasi faktor-faktor yang memengaruhi kelulusan tepat waktu, sehingga dapat dirumuskan kebijakan dan program yang tepat sasaran.

BAB II LANDASAN TEORI

2.1 Penelitian Terdahulu

Dalam pelaksanaan penelitian ini, diperlukan adanya referensi atau kajian terkait sebagai upaya untuk menghindari terjadinya duplikasi serta plagiarisme. Penelitian-penelitian sebelumnya digunakan sebagai acuan, pedoman, dan bahan perbandingan. Selain itu, penulis juga menyertakan hasil-hasil penelitian terdahulu pada bagian landasan teori untuk mengidentifikasi perbedaan dengan penelitian ini, yang dijabarkan didalam tabel 1.2.

Tabel 1. 2 Penelitian Terdahulu

NOMOR	PENELITI (TAHUN)	JUDUL PENELITIAN	MASALAH	HASIL PENELITIAN
1	Apridiansyah ,Yovi et al., (2021)	Prediksi Kelulusan Mahasiswa Fakultas Teknik Informatika Universitas Muhammadiyah Bengkulu Menggunakan Metode <i>Naive Bayes</i>	Setiap tahun, terdapat banyak lulusan, khususnya dari Fakultas Teknik Informatika, namun data mengenai mereka belum sepenuhnya tersedia. Informasi mengenai lulusan dari fakultas teknik hanya dapat dilihat berdasarkan periode wisuda tertentu. Oleh karena itu, penerapan algoritma <i>Naive Bayes</i> dalam klasifikasi data	Algoritma <i>Naive Bayes</i> dapat diterapkan untuk memprediksi apakah seorang mahasiswa lulus tepat waktu atau tidak, yang berguna sebagai sumber informasi dan masukan bagi perguruan tinggi dalam merumuskan kebijakan di masa mendatang. Prediksi dengan algoritma <i>Naive</i>

			diharapkan dapat membantu memprediksi jumlah lulusan mahasiswa dari setiap angkatan.	<i>Bayes</i> yang menggunakan atribut nama, NPM, dan nilai IPK menghasilkan tingkat presisi sebesar 90%, <i>recall</i> sebesar 100%, serta akurasi sebesar 90%.
2	Nazla Chairina (2023)	Penggunaan Metode <i>Naive Bayes</i> untuk Memprediksi Kelulusan Mahasiswa Pendidikan Teknologi Informasi Menggunakan Rapidminer	Jumlah mahasiswa terus bertambah setiap tahun, namun yang lulus tepat waktu sangat sedikit. Hal ini menyebabkan ketidakseimbangan data dan perlunya sistem prediksi kelulusan untuk membantu pengambilan keputusan secara dini di perguruan tinggi.	Penelitian ini menggunakan metode <i>Naive Bayes</i> dengan alat bantu <i>Rapidminer Studio 9.10</i>. <i>Training data</i>ing terdiri dari 536 data mahasiswa angkatan 2014–2018, dan <i>testing data</i>ing berasal dari mahasiswa angkatan 2021 sebanyak 62 data. Hasil pengujian akurasi metode <i>Naive Bayes</i> menunjukkan

				tingkat akurasi sebesar 68,22% .
3	Anwarudin, Andriyani Widyastuti, et al., (2022)	The Prediction on the Students' Graduation Timeliness Using <i>Naive Bayes Classification</i> and K-Nearest Neighbor	Pada kasus Program Studi Teknologi Laboratorium Medik STIKES Guna Bangsa belum memiliki alat untuk memprediksi ketepatan waktu kelulusan mahasiswa. Masalah umum yang muncul ialah mahasiswa tidak lulus tepat waktu atau lulus tepat waktu namun dengan IPK rendah (≤ 2.50).	Penelitian ini menunjukkan bahwa perbandingan antara penggunaan algoritma <i>Naive Bayes</i> dan K-NN pada sistem prediksi, dan hasilnya untuk algoritma <i>Naive Bayes</i> mendapatkan akurasi model sebesar 96,11% dan algoritma K-NN menghasilkan akurasi model sebesar 97,68%.
4	Fajar Embun Wati, Elvi, et al., (2023)	Prediction of Student Graduation Using the <i>K-Nearest Neighbors Method</i>	BanyakNYA mahasiswa tidak lulus tepat waktu, mempengaruhi akreditasi perguruan tinggi. Diperlukan adanya sistem prediksi kelulusan untuk membantu prodi mengawasi	Dari 150 dataset mahasiswa Teknik Informatika, dengan memanfaatkan variable gender, umur, status pernikahan, nilai, dan status pekerjaan,

			mahasiswa dan memberikan pembinaan sejak awal.	penelitian ini diuji dengan K-NN menggunakan RapidMiner 9.9 dengan tingkat akurasi rata-rata 100%.
5	Hakim et al. (2020)	Aplikasi Prediksi Kelulusan Mahasiswa Berbasis K-Nearest Neighbor (K-NN)	Rendahnya tingkat kelulusan mahasiswa tepat waktu (sekitar 25%) serta belum tersedianya sistem prediksi untuk mengidentifikasi mahasiswa yang berpotensi terlambat lulus.	Metode K-NN menghasilkan akurasi tertinggi sebesar 80% pada pengujian K-Fold Cross Validation dan mencapai 98% pada pengujian Confusion Matrix, sehingga efektif digunakan untuk prediksi kelulusan mahasiswa.

2.2 Landasan Teori

Terdapat beberapa teori yang menjadi dasar pendukung dalam penelitian ini, yang diuraikan sebagai berikut :

2.2.1 Sistem Cerdas

Terdiri dari 2 kata, yaitu “Sistem” dan “Cerdas” yang merujuk kepada komponen dari sebuah kata frasa “Sistem Cerdas”. Kata “Sistem” merujuk pada sekumpulan unsur yang saling berkaitan, terpadu, dan terhubung untuk membentuk suatu kesatuan. Disini lain, kata "cerdas" berarti memiliki perkembangan akal yang optimal dalam hal berpikir, memahami, dan sejenisnya. (KBBI Daring, s.v.”kamus”, diakses 31 Mei 2024, <https://kbbi.kemdikbud.go.id/>)

Sistem cerdas sendiri adalah sistem yang dibangun menggunakan teknik dari kecerdasan buatan yang berfungsi sebagai pengambil keputusan melalui pemanfaatan logika dan pengetahuan yang sudah ada sebelumnya.

Dikutip dari buku “*Sistem Cerdas dan Pendukung Keputusan*” oleh Dr. Sri Kusumadewi, S.Si., M.T., Sistem cerdas atau yang biasa dikenal dengan *Intelligent System* atau Sistem Cerdas adalah sistem-sistem yang dibangun menggunakan metode dan teknik Kecerdasan Buatan atau *Artificial Intelligent (AI)*. Didalam sistem cerdas ini, terdapat 3 hal dasar yaitu basis pengetahuan (*knowledge base*), mesin inferensi (*inference engine*), dan antarmuka pengguna (*user interface*). (Kusumadewi et al., 2021)

Basis pengetahuan atau *knowledge base* adalah suatu komponen untuk menyimpan informasi dan pengetahuan yang diperlukan oleh sistem cerdas untuk penalaran dan pengambilan keputusan, bersifat akurat, konsisten dan up-to-date. Selain itu, mesin inferensi atau *inference engine* diketahui sebagai “otak” dari sebuah sistem cerdas yang pastinya sudah menerapkan aturan dan algoritma tertentu untuk menghasilkan kesimpulan atau rekomendasi baru dari pengetahuan yang ada. Terakhir merupakan antarmuka pengguna atau *user interface* yang merupakan komponen wajib bagi setiap sistem berbasis komputer. Komponen ini ditujukan sebagai wadah untuk komunikasi antara pengguna dengan sebuah sistem cerdas.

Sistem cerdas dan kecerdasan buatan merupakan 2 hal yang sangat erat kaitannya. Keduanya saling berhubungan satu sama lain. Didalam sebuah sistem cerdas pastinya terdapat pengimplementasian dari kecerdasan buatan atau *artificial intelligence*.

Istilah “kecerdasan buatan” sering disebut dengan “artificial intelligence (AI)” mengacu pada kecerdasan yang dimiliki oleh entitas/pakar ilmiah. Agar mesin dapat melakukan tugas dengan cara yang sama seperti manusia, maka pada saat ini banyak sistem yang sudah mengimplemntasikan dan mengembangkan AI ini didalam sistem operasinya. Penerapan AI dalam sebuah sistem inilah yang disebut dengan “Sistem Cerdas”.

Machine learning memiliki tiga paradigma utama yang digunakan untuk mengekstraksi pola serta pengetahuan dari data yaitu *Supervised Learning*, *Unsupervised Learning*, dan *Reinforcement Learning* (Nurhalizah et al., 2024). *Supervised Learning* merupakan pendekatan pembelajaran mesin yang memanfaatkan data berlabel sebagai acuan untuk proses prediksi atau klasifikasi (Mestika et al., 2022). Sedangkan *Unsupervised Learning*, model yang ingin dibangun harus menemukan pola dari data sendiri tanpa bantuan label eksternal (Montavon et al., 2022).

Terakhir *Reinforcement Learning* yaitu sebuah algoritma yang mengumpulkan informasi untuk memilih langkah-langkah menuju hasil yang diinginkan dengan tingkat yang optimal (Alfarizi et al., 2023). Meskipun tujuan dari ketiga model ini adalah untuk menghasilkan model yang mengidentifikasi pola pada data, perbedaan mendasar dalam metode pendekatan yang digunakan memberikan pengaruh yang signifikan terhadap penerapan di dunia nyata serta hasil kinerja dan performa yang diperoleh.

2.2.2 Data Mining

Data mining adalah suatu proses dengan tujuan untuk menemukan pola serta informasi yang penting dan bernilai dari kumpulan data dengan jumlah yang besar. Dengan adanya teknologi *big data* dan kemajuan teknologi penyimpanan data, penggunaan teknik *data mining* sendiri telah meningkat pesat dalam beberapa masa terakhir, sehingga dapat mengubah data mentah menjadi pengetahuan bernilai dan dapat dimanfaatkan oleh organisasi atau perusahaan. (Subariah & Zein, 2023)

Data mining juga diartikan sebagai proses analisis data dari berbagai sudut pandang yang berbeda dan merumuskannya menjadi informasi yang berguna, dimana informasi yang diperoleh dipresentasikan untuk digunakan dalam membuat keputusan bisnis yang penting (Kolang Aditya Pratama et al., 2020).

Terdapat istilah *Knowledge Discovery Database (KDD)* yaitu salah satu bagian metode dari data mining untuk mencari informasi-informasi, pola-pola yang terdapat didalam data, dan melibatkan algoritma-algoritma tertentu untuk

mengidentifikasi pola-pola data tersebut. Ada beberapa tahapan didalam metode KDD ini, yaitu sebagai berikut :

1. *Data Selection* (Seleksi Data), adalah tahap yang dilakukan sebelum proses penggalian informasi dalam KDD dimulai. Hasil dari seleksi data ini disimpan dalam sebuah folder yang terpisah dari database operasional.
2. Pemilihan Data (Preprocessing / Cleaning), proses preprocessing mencakup kegiatan seperti menghapus data duplikat, memeriksa inkonsistensi dalam data, dan memperbaiki kesalahan pada data, seperti kesalahan ketik (*typography*). Selain itu terdapat tahapan enrichment, yaitu proses "memperkaya" data yang tersedia dengan tambahan informasi atau data lain yang relevan dan dibutuhkan untuk KDD, termasuk data atau informasi dari sumber eksternal.
3. Transformasi (*Transformation*), yaitu tahap yang dilakukan dengan mengubah bentuk data yang belum memiliki entitas yang jelas dan belum terbaca menjadi data yang valid atau siap untuk diproses dalam *Data Mining*.
4. *Data Mining*, pada tahapan ini, dilakukan penerapan algoritma atau metode untuk mencari pengetahuan. Proses ini merupakan langkah krusial di mana teknik kecerdasan digunakan untuk mengekstrak pola informasi yang berpotensi berguna dari data yang telah dipilih.
5. Interpretasi/Evaluasi (*Interpretation/Evaluation*), pada tahapan terakhir ini, dilakukan proses untuk menghasilkan keluaran yang berasal dari pola informasi yang diperoleh melalui proses *Data Mining*.

2.2.3 Klasifikasi

Klasifikasi merupakan proses evaluasi terhadap objek data dan mengelompokkannya ke dalam beberapa kategori tertentu dari sejumlah kelas yang ada. Dalam klasifikasi, terdapat dua tugas utama yang dilakukan, yaitu :

1. Membangun model sebagai prototipe yang disimpan sebagai memori,
2. Menerapkan model tersebut untuk mengenali, mengklasifikasikan, atau memprediksi objek data lainnya.

Sehingga dapat diketahui ke dalam kategori mana objek data tersebut tergolong dalam model yang mudah disimpan..(Bendesa Subawa, 2019)

Banyak pengaplikasian yang menerapkan sistem klasifikasi, diantaranya yaitu sistem klasifikasi hewan. Dengan berbagai atribut yang tersedia, jika ada hewan baru, hewan itu dapat dengan mudah diidentifikasi. Matriks klasifikasi sendiri memberikan informasi perbandingan antara klasifikasi yang sebenarnya dengan klasifikasi yang dihasilkan oleh sistem pengklasifikasian.

2.2.4 Naive Bayes

Naive Bayes adalah model pengklasifikasian *machine learning* yang sangat sederhana, namun terbukti sangat efektif dan sering digunakan. Algoritma ini bekerja sebagai pengklasifikasi probabilistik yang melakukan klasifikasi dengan menerapkan aturan *Maximum A Posteriori* dalam konteks *Bayesian*. Selain itu, model ini dapat direpresentasikan menggunakan jaringan Bayesian yang sederhana. *Naive Bayes* sangat populer dan terkenal dalam klasifikasi teks dan sering digunakan sebagai solusi klasik untuk masalah seperti deteksi spam.

Berikut merupakan kelebihan dan kekurangan dari algoritma *Naive Bayes*.(Subariah & Zein, 2023)

1. Kelebihan Model *Naive Bayes*:
 - Proses klasifikasi pada 40 data uji dapat dilakukan dengan waktu yang relatif singkat.
 - Tidak terlalu dipengaruhi oleh adanya fitur yang kurang relevan
 - Konsep dan mekanismenya mudah dipahami
 - Proses pengkodean data yang dibutuhkan relatif sederhana
2. Kekurangan *Naive Bayes* :
 - Mengandalkan asumsi bahwa data yang diberikan bersifat independent
 - Kerhasilan model tergantung pada seberapa bagusnya pengetahuan awal yang dimiliki.

2.2.5 K-Nearest Neighbors (K-NN)

K-NN atau *K-Nearest Neighbors* adalah algoritma yang melakukan klasifikasi dengan menilai kedekatan antara kasus baru dengan kasus lama/ yang

sudah ada. Prinsip dasar algoritma ini adalah menemukan jarak terpendek antara data yang diperlukan untuk evaluasi dengan terdekatnya .(Safitri et al., 2023). Secara sederhana, K-NN memprediksi berdasarkan data baru yang didekati dengan tetangga terdekatnya.

Pada dasarnya, algoritma K-NN ini bekerja tanpa proses yang sulit atau kompleks (non-parametric), sehingga proses komputasi lebih banyak terjadi pada saat proses prediksi nantinya

Cara kerja dari algoritma K-NN ini adalah dengan menghitung jarak antara data baru dengan semua data pelatihan, lalu mengidentifikasi k tetangga terdekat. Setelah itu, untuk klasifikasi, model akan memilih label yang paling umum diantara tetangga tersebut.

Ada beberapa kelebihan dan kekurangan dari algoritma K-NN, diantaranya sebagai berikut :

1. Kelebihan :

- Efektif pada data besar
- Tidak membutuhkan asumsi distribusi data
- Tangguh terhadap noise/data yang tidak relevan
- Sangat mudah diimplementasikan pada beberapa kasus

2. Kekurangan :

- Membutuhkan lebih banyak penyimpanan
- Sangat sensitive terhadap nilai k
- Diperlukan penskalaan fitur

2.2.6 Ensemble Learning dan Hybrid Model

Ensemble learning adalah metode dalam machine learning yang mengombinasikan beberapa model atau algoritma pembelajaran untuk menghasilkan satu model prediksi yang lebih kuat dan stabil dibandingkan penggunaan satu model tunggal. *Ensemble learning* dapat didefinisikan sebagai model dasar pembelajaran di mana beberapa pengklasifikasi dasar dilatih dan

digabungkan untuk menghasilkan keputusan klasifikasi akhir (Rachmatullah, M. I. C., 2025)

Hybrid model merupakan salah satu contoh dari penerapan *ensemble learning*, di mana dua atau lebih algoritma yang berbeda dikombinasikan untuk saling melengkapi kelebihan masing-masing algoritma. Dengan menggabungkan karakteristik antar algoritma yang berbeda, model ini mampu memanfaatkan kelebihan dari masing-masing model sehingga dapat menghasilkan performa yang lebih baik dan optimal dalam proses klasifikasi maupun prediksi.

Algoritma *Naive Bayes* memiliki karakteristik komputasi yang cepat dan dapat bekerja sangat baik pada data yang memiliki probabilitas distribusi yang jelas, sedangkan algoritma K-NN memiliki karakteristik yang lebih unggul dalam menentukan kedekatan pola berdasarkan sampel terdekat (nilai k) tanpa asumsi distribusi data.

Selain itu, alasan penggabungan antar hasil prediksi kedua algoritma ini adalah supaya meningkatkan performa dan kinerja prediksi dengan menggabungkan hasil prediksi kedua model (*Naive Bayes* dan K-NN) yang telah dibangun sebelumnya. Penggabungan ini juga dapat mengurangi kelemahan dari tiap model yang telah dibangun, meningkatkan stabilitas proses prediksi, dan tentunya meningkatkan akurasi.

Jenis-jenis dari *ensemble learning* yang umum digunakan antara lain :

1. *Voting Ensemble*, menentukan prediksi akhir berdasarkan mekanisme *majority voting* (suara terbanyak) atau *soft voting* (perataan probabilitas) dari beberapa *classifier* (Ardianti et al., 2025). Metode ini cenderung sederhana namun efektif dalam meningkatkan performa klasifikasi disaat model dasar memiliki tingkat keragaman yang baik.
2. *Bagging (Bootstrap Aggregating)*, adalah metode *ensemble learning* dengan melatih beberapa model pada subset data hasil *bootstrap sampling* untuk menurunkan variansi dan meningkatkan stabilitas prediksi (Churniansyah & Utomo, 2024).

3. *Boosting*, yaitu metode *ensemble learning* yang melatih model secara berurutan, dengan model selanjutnya lebih terfokus pada data yang salah diklasifikasikan oleh model sebelumnya untuk memperbaiki kesalahan prediksi secara bertahap. Selain untuk meningkatkan akurasi model, metode ini juga efektif digunakan dalam penanganan ketidakseimbangan kelas ketika dikombinasikan dengan metode ensemble lainnya (Chen et al., 2024).

2.2.7 Evaluasi Performa Model Klasifikasi

Evaluasi performa model klasifikasi merupakan tahapan yang penting dalam penelitian pengembangan sistem cerdas prediksi kelulusan mahasiswa tepat waktu berbasis *machine learning*. Tahapan ini bertujuan untuk mengukur seberapa baik dan bagus performa dari model yang dibangun dalam memprediksi secara akurat terhadap data baru yang belum pernah dilihat sebelumnya.

Dalam penelitian ini, evaluasi performa model dilakukan dengan beberapa metode, yaitu *confusion matrix*, *accuracy*, *precision*, *recall*, *f1-score*, *learning curve*, *ROC Curve* & Nilai AUC dan *validation curve*.

1. *Confusion matrix* merupakan tabel evaluasi model yang digunakan untuk menggambarkan perbandingan antara hasil prediksi model yang dibangun dengan keadaan yang sebenarnya (*ground truth*). Metrik tabel ini sangat berguna untuk mengetahui jumlah prediksi yang benar dan salah pada setiap kelas dalam proses klasifikasi. (Eskiyaturrofikoh et al., 2024)

Pada kasus klasifikasi biner, *confusion matrix* terdiri dari 4 komponen utama, dapat dilihat pada gambar 2.1.

		Predicted Values	
		Positive	Negative
Actual Values	Positive	TP	FN
	Negative	FP	TN

Gambar 2. 1 Confusion Matrix

- True Positive (TP)
Menunjukkan jumlah data yang diprediksi sebagai kelas positif dan hasil prediksi tersebut sesuai dengan kondisi sebenarnya.
- True Negative (TN)
Menunjukkan jumlah data yang diprediksi sebagai kelas negatif dan hasil prediksi tersebut sesuai dengan kondisi sebenarnya.
- False Positive (FP)
Menunjukkan jumlah data yang diprediksi sebagai kelas positif, tetapi pada kenyataannya termasuk ke dalam kelas negatif.
- False Negative (FN)
Menunjukkan jumlah data yang diprediksi sebagai kelas negatif, namun pada kenyataannya termasuk ke dalam kelas positif.

2. *Accuracy* atau akurasi metrik evaluasi yang digunakan untuk menilai seberapa tepat model dalam memprediksi kelas target secara menyeluruh. Metrik ini mengukur proporsi prediksi yang benar terhadap seluruh data uji, sehingga memberikan gambaran umum mengenai kinerja model dalam melakukan klasifikasi data (Pramudyanto et al., 2024). Rumus dari metrik akurasi dapat dilihat pada persamaan 1.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Nilai *accuracy* yang tinggi menunjukkan bahwa model memiliki kemampuan prediksi terhadap kelas target dengan baik. Namun pada kasus dataset yang tidak seimbang (*imbalanced data*), penggunaan tingkat akurasi saja tingkat cukup untuk menggambarkan performa model secara menyeluruh, sehingga diperlukan juga nilai dari *precision*, *recall*, dan *f1-score*.

3. *Precision* atau presisi merupakan sebuah nilai dari metrik evaluasi untuk mengukur tingkat ketepatan dari model yang dibangun dalam memprediksi kelas positif. Rumus metrik *Precision* dapat dilihat pada persamaan 2.

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

Hasil *precision* ini menunjukkan seberapa besar proporsi data yang diprediksi sebagai kelas positif benar-benar termasuk kedalam kelas positif. Oleh sebab itu, Nilai *precision* yang tinggi menunjukkan bahwa model memiliki tingkat kesalahan prediksi positif yang rendah

4. *Recall* atau nilai sensitivitas adalah nilai yang mengukur kemampuan model dalam mendeteksi seluruh data yang termasuk kedalam kelas positif. *Recall* ini menunjukkan seberapa banyak data positif yang berhasil dikenali oleh model yang dibangun. Rumus dari nilai *Recall* dapat dilihat pada persamaan 3.

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

Nilai *recall* yang tinggi mengindikasikan bahwa model mampu mengidentifikasi sebagian besar data yang benar-benar termasuk kedalam kelas positif.

5. *F1-score* merupakan metrik evaluasi yang terdiri dari gabungan nilai *precision* dengan nilai *recall* dalam satu ukuran. *F1-score* digunakan untuk mengukur keseimbangan antara ketepatan prediksi kelas positif dengan kemampuan model untuk mengenali dan mendeteksi kelas positif. Rumus nilai *F1-Score* dapat dilihat pada persamaan 4.

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

6. *Learning curve* atau kurva pembelajaran yaitu sebuah grafik kurva yang menggambarkan hubungan antara jumlah dari data latih (*training data*) dengan performa model, baik pada data latih maupun data validasi. Apabila kurva *training* memiliki performa yang tinggi dan kurva *validation* jauh lebih rendah, maka model cenderung mengalami *overfitting* (model terlalu menyesuaikan dengan data latih). Sebaliknya apabila kedua kurva memiliki performa yang rendah, maka model cenderung mengalami *underfitting* (model terlalu sederhana).

7. *ROC Curve (Receiver Operating Characteristic)* dan nilai *AUC (Area Under Curve)* merupakan grafik yang digunakan untuk mengukur seberapa bagus dan baik model mampu membedakan antara kelas negatif dengan kelas positif. Pada *ROC Curve*, semakin kurva mendekati sudut kiri atas grafik, maka semakin bagus performa modelnya. Sedangkan pada nilai *AUC*, semakin mendekati nilai 1 maka semakin bagus performa model dalam melakukan klasifikasi.
8. *Validation Curve* atau kurva validasi adalah grafik yang digunakan untuk mengevaluasi pengaruh nilai *hyperparameter* terhadap performa model. didalam penelitian ini, grafik ini menunjukkan hubungan antara parameter tertentu seperti nilai k pada algoritma K-NN dengan nilai performa model pada data latih dan data validasi

2.2.8 Python dan Google Colaboratory

Python merupakan =bahasa pemrograman yang populer dan sangat terkenal dan banyak digunakan dalam pengembangan aplikasi web, perangkat lunak, ilmu data, serta *machine learning* (ML). Bahasa pemrograman ini dipilih oleh para pengembang/developer karena efisiensinya dan kemudahan dalam pembelajaran, serta kemampuannya untuk dijalankan pada berbagai platform. Selain itu Python bersifat *open source* yang artinya dapat diunduh secara gratis, terintegrasi dengan semua jenis sistem, dan mampu mempercepat proses pengembangan perangkat lunak.. (Amazon Web Services, 2024)

Terdapat beberapa library yang dapat digunakan pada bahasa pemrograman python, diantaranya yaitu :

- **Matplotlib.** Merupakan Pustaka yang digunakan oleh pengembang untuk memvisualisasikan data dalam grafik berkualitas tinggi, baik dalam dua dimensi (2D) maupun tiga dimensi (3D). Library ini banyak dimanfaatkan dalam aplikasi ilmiah, dan memungkinkan untuk penyajian data dalam berbagai bentuk diagram, seperti diagram batang dan diagram garis.
- **Pandas.** Pandas menyediakan struktur data yang fleksibel dan efisien, dapat digunakan untuk memanipulasi data terstruktur maupun data deret waktu,

seperti tabel dan deret. Implementasi Pandas sendiri dapat digunakan untuk membaca, menulis, menggabungkan, memfilter, dan mengelompokkan data.

- **NumPy.** Adalah pustaka yang terkenal dan populer, dan biasa digunakan oleh para pengembang untuk membuat serta mengelola deret dengan mudah, memanipulasi bentuk logis. Library ini juga dapat melakukan operasi aljabar linier. NumPy juga mendukung integrasi dengan berbagai bahasa pemrograman seperti C dan C++.

Sementara itu, Google Colaboratory adalah sebuah *executable document* atau dokumen yang dapat dieksekusi dan dapat digunakan untuk menyimpan, menulis, serta membagikan program yang telah dibuat melalui Google Drive.

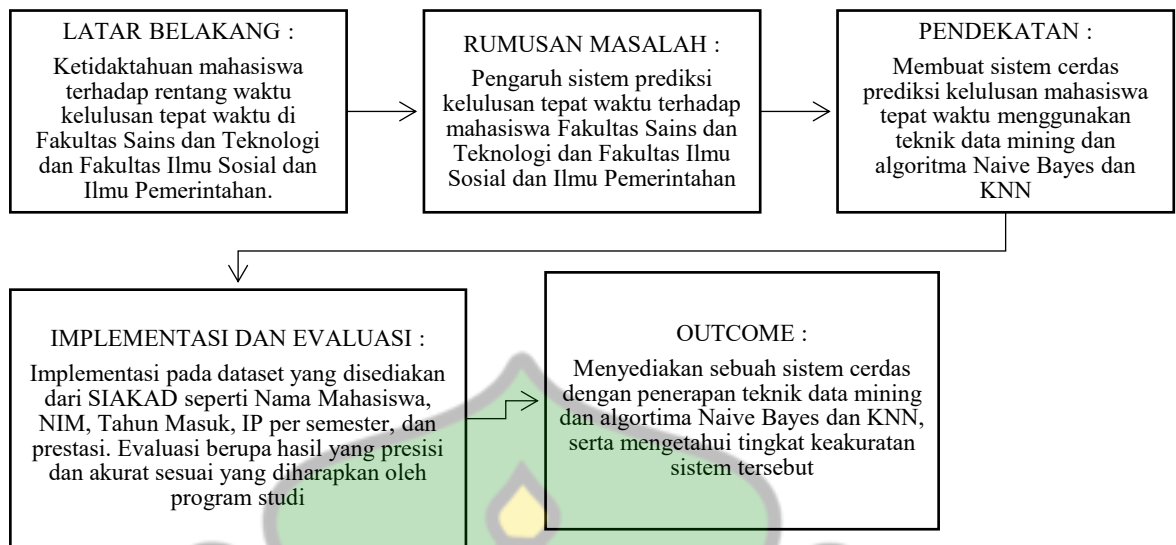
Perangkat lunak ini memungkinkan penggunanya untuk menjalankan kode dengan bahasa pemrograman Python tanpa perlu menginstal dan melakukan konfigurasi setup lainnya. Oleh karena itu, aplikasi ini merupakan pilihan terbaik bagi para *programmer* yang ingin mengasah pengetahuan mereka mengenai Python.

2.2.9 Kelulusan Mahasiswa

Kelulusan mahasiswa menandai selesainya masa studi seorang mahasiswa di perguruan tinggi. Kelulusan juga merupakan hasil akhir yang membanggakan dari pencapaian dalam menempuh suatu pendidikan pada tingkat tertentu. Mahasiswa pada program pendidikan Strata Satu (S-1) diwajibkan menempuh minimal 145 satuan kredit semester (SKS) dalam waktu 8 semester, dengan batas maksimum penyelesaian selama 14 semester.

2.3 Kerangka Teoritis

Kerangka Pemikiran atau teoritis menggambarkan struktur paradigma berpikir yang digunakan sebagai landasan oleh penulis untuk memahami, menganalisis dan memecahkan masalah. Kerangka Pemikiran membantu mengorganisir informasi, konsep menjadi satu rangkaian logis. Alur kerangka teoritis dapat dilihat pada gambar 2.6.



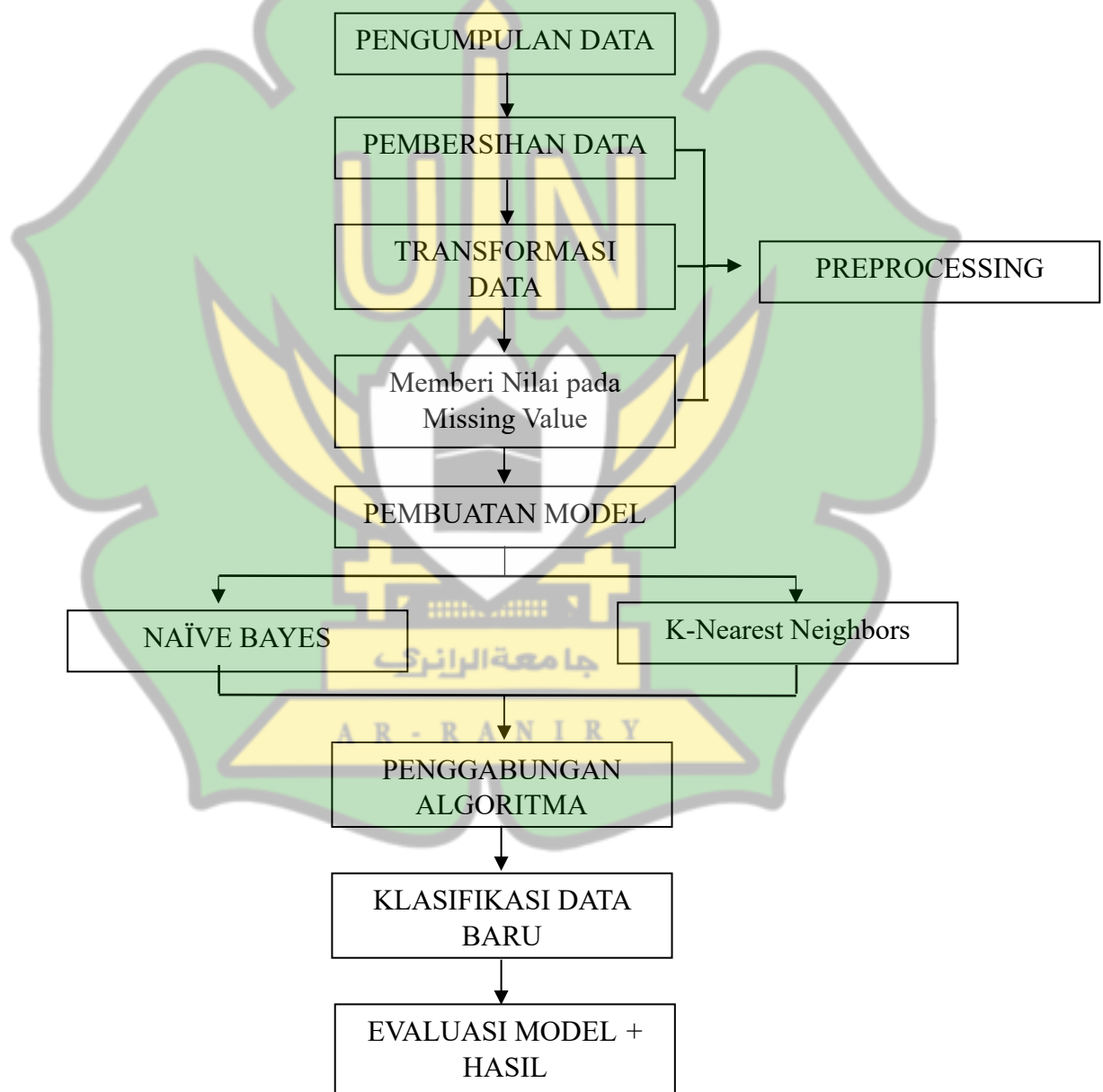
Gambar 2. 2 Kerangka Teoritis



BAB III METODE PENELITIAN

3.1 Tahapan Penelitian

Tahapan penelitian merujuk pada serangkaian langkah dan proses yang perlu diikuti dari awal sampai akhir oleh peneliti untuk menyelesaikan sebuah penelitian. Didalam Penelitian ini, peneliti menggunakan beberapa tahapan penelitian yang digambarkan pada gambar 3.1 :



Gambar 3. 13 Tahapan Penelitian

3.1.1 Pengumpulan Data

Pada tahap awal, peneliti menggunakan dataset yang telah disediakan didalam Sistem Informasi Akademik UIN Ar-Raniry Banda Aceh. Peneliti mengumpulkan data-data mahasiswa Fakultas Sains dan Teknologi dan Fakultas Ilmu Sosial dan Ilmu Pemerintahan dalam rentang tahun ajaran ganjil 2018/2019 hingga tahun ajaran ganjil 2023/2024.

Adapun data-data atribut yang peneliti kumpulkan berupa Nama, NIM, Jurusan, Fakultas, Tahun Masuk, dan Indeks Prestasi Semester (IPS) mahasiswa. Dalam hal ini, penelitian ini hanya memasukkan 4 angkatan mahasiswa Fakultas Sains dan Teknologi dan Fakultas Ilmu Sosial dan Ilmu Pemerintahan, yaitu dari tahun masuk 2018 sampai dengan 2021.

Pada penelitian ini, peneliti menggunakan data populasi (census data) sebagai teknik pengambilan data. Oleh karena itu, penelitian ini tidak menerapkan teknik sampling, baik *random sampling* maupun *stratified sampling*, karena seluruh data yang memenuhi kriteria penelitian digunakan sebagai objek penelitian.

Namun, pada tahap pembagian data uji dan data latih dalam proses pemodelan *machine learning* nantinya, digunakan teknik *stratified random sampling split* untuk menjaga proporsi kelas target dengan tujuan agar hasil evaluasi model lebih objektif.

Jumlah dataset yang peneliti gunakan pada penelitian ini sebanyak 1.828 mahasiswa dari Fakultas Sains dan Teknologi dan Fakultas Ilmu Sosial dan Ilmu Pemerintahan (SIKAD, 2025), dengan rincian pada tabel 3.1.

Tabel 3. 1 Data Mahasiswa

Tahun Masuk Fakultas	2018	2019	2020	2021	TOTAL
SAINTEK	361	292	167	308	1.128
FISIP	201	177	108	214	700
TOTAL					1.828

3.1.2 Analisis Karakteristik Dataset

Proses analisis karakteristik dari dataset mahasiswa dilakukan sebagai langkah awal untuk memahami struktur, distribusi, dan kondisi data yang digunakan dalam penelitian sebelum dilakukan proses pemodelan. Tahapan ini bertujuan untuk memastikan bahwa dataset memiliki kualitas yang baik serta sesuai dengan kebutuhan algoritma klasifikasi yang diterapkan, yaitu *Naive Bayes* dan *K-Nearest Neighbors* (K-NN).

1. Struktur Dataset

Struktur dataset terdiri dari beberapa atribut utama yang digunakan dalam proses permodelan, yaitu :

- Tahun Masuk
Atribut ini menunjukkan tahun awal mahasiswa yang terdaftar sebuah diperguruan tinggi. Data ini dipakai untuk mendukung penentuan jangka waktu belajar mahasiswa serta hubungan dengan kondisi status kelulusan mereka.
- Indeks Prestasi Semester (IPS)
Atribut IPS mahasiswa terdiri dari nilai IPS semester 1 sampai dengan semester 8 yang bersifat numerik dengan rentang nilai 0,00 sampai dengan 4,00. Atribut ini nantinya menjadi atribut fitur utama dalam proses klasifikasi karena dapat mewakili performa akademik mahasiswa selama masa studi.
- Status Kelulusan Mahasiswa (Target Biner)
Atribut ini merupakan label target yang digunakan dalam proses klasifikasi, yang dibentuk menjadi dua kelas, yaitu: Lulus tepat waktu (kelas 1) dan Lulus tidak tepat waktu atau belum lulus (kelas 0).

2. Distribusi Kelas Target

Distribusi kelas pada dataset menunjukkan adanya ketimpangan jumlah data antara mahasiswa yang berhasil menyelesaikan studi tepat waktu dengan mahasiswa yang lulus tidak tepat waktu atau belum lulus. Kondisi ini menggambarkan adanya fenomena ketidakseimbangan kelas (*class imbalance*),

yang artinya jumlah data pada kelompok mahasiswa yang lulus tidak tepat waktu atau belum lulus lebih besar dibandingkan dengan kelompok mahasiswa yang lulus tepat waktu.

Hal ini berpotensi mempengaruhi kinerja model dalam pengklasifikasian, khususnya dalam mengidentifikasi kelas dengan jumlah data yang lebih sedikit. Untuk meminimalisir hal tersebut, penelitian ini menerapkan metode pembagian data menggunakan teknik *stratified split* pada proses pemisahan antara data latih (*training data*) dan data uji (*testing data*). Teknik ini berguna untuk menjaga proporsi/keseimbangan masing-masing kelas tetap konsisten, sehingga evaluasi model yang dihasilkan lebih akurat.

3.1.3 Preprocessing Data

Tahapan selanjutnya adalah tahapan preprocessing data. Tahapan ini sangat penting disaat ingin menerapkan *machine learning* dengan tujuan untuk memastikan dataset yang digunakan dalam pelatihan berada dalam kondisi yang bersih, konsisten, dan siap untuk diproses oleh algoritma yang diterapkan dalam penelitian ini. Proses *Preprocessing Data* digambarkan pada gambar 3.2.



Gambar 3. 1 Diagram Proses Preprocessing Data

Pada penelitian ini, ada beberapa tahapan preprocessing data yang peneliti terapkan, yaitu :

A. Pembersihan Data (Cleaning Data)

Pada tahap selanjutnya, peneliti melakukan pembersihan data / cleaning data. Tahapan ini dilakukan sebelum masuk ke proses mining pada data. Cleaning data sendiri merupakan tahapan dimana membersihkan atau menghapus atribut-atribut yang tidak berpengaruh dalam perhitungan.

Tujuan dari tahapan ini adalah melakukan pengenalan dan perbaikan pada dataset yang akan diteliti. Tahapan cleaning data pada algoritma *Naive Bayes* sangat diperlukan mengingat sangat diperlukannya pengecekan kualitas data sebelum digunakan dalam proses klasifikasi. Adapun tahapan-tahapan cleaning data dalam konteks penerapan algoritma *Naive Bayes* dan *K-Nearest Neighbors* adalah:

1. Identifikasi data yang tidak konsisten dan error

Langkah pertama adalah inspeksi untuk menemukan data yang tidak konsisten, error, atau tidak valid, misalnya Data duplikat yaitu jika ada data yang berulang, data tersebut harus dihapus, data yang salah format yaitu nilai numerik yang tercampur dengan data teks yang tidak sesuai format, dan data outlier yaitu data aneh yang mungkin merupakan kesalahan atau tidak sesuai dengan distribusi umum data.

2. Penghapusan atau penanganan data yang tidak relevan

Data yang tidak relevan untuk diklasifikasi harus dihapus. Beberapa atribut mungkin tidak memiliki pengaruh signifikan atau bermakna terhadap prediksi dan klasifikasi, seperti NIM, nama, Jurusan, dan Fakultas. Atribut-atribut seperti ini tidak mempengaruhi proses klasifikasi didalam *Naive Bayes* maupun K-NN, karena tidak membawa informasi yang relevan untuk perhitungan probabilitas. Menghapus atribut-atribut tersebut juga membantu menyederhanakan model dan menghindari *noise* dalam data.

3. Konversi format angka IP

Beberapa nilai IP semester masih menggunakan format koma (,) sebagai pemisah decimal. Dikarenakan bahasa pemrograman python hanya mengenali titik

(.) sebagai pemisah decimal, maka peneliti perlu melakukan konversi terhadap angka IP yaitu :

- Mengganti koma(,) menjadi titik (.) sebagai pemisah desimal
- Mengubah seluruh nilai angka IP menjadi tipe data float

B. Penanganan *missing value*

Dalam sebuah penelitian, data yang diteliti biasanya sering ditemukan *missing value* (nilai kosong). Penanganan terhadap data yang kosong ini penting karena algoritma *Naive Bayes* dan K-NN tidak dapat memproses data yang tidak lengkap. Beberapa teknik dapat dilakukan untuk menangani *missing value*, antara lain pengisian dengan nilai rata-rata dan nilai tengah (mean/median), modus (nilai yang paling sering muncul), serta estimasi menggunakan algoritma tertentu.

C. Membuat target biner (Label Encoding)

Pada model *machine learning*, tidak dapat memproses nilai dalam bentuk teks (string), oleh sebab itu diperlukan konversi kedalam bentuk angka. Tahapan ini biasa juga disebut dengan Konversi Data Kategorikal ke Numerik. Dalam hal ini, peneliti mengklasifikasikan status kelulusan mahasiswa kedalam 2 kategori, yaitu mahasiswa yang lulus tepat waktu dan mahasiswa yang lulus tidak tepat waktu/belum lulus. Pemetaannya dilakukan dengan cara sebagai berikut :

- Lulus Tepat Waktu $\rightarrow 1$
- Lulus Tidak Tepat Waktu $\rightarrow 0$
- Belum Lulus $\rightarrow 0$

Pemetaan ini menghasilkan kolom baru berupa target biner yang menjadi dasar klasifikasi. Dengan adanya target biner, model dapat melakukan pelatihan untuk membedakan antara mahasiswa yang mampu menyelesaikan studi kuliah secara tepat waktu dan yang tidak.

D. Standarisasi Data

Tahap preprocessing data berikutnya adalah melakukan standarisasi data. Tahapan ini merupakan proses transformasi fitur-fitur numerik agar memiliki skala yang beragam. Tujuan utama proses ini adalah untuk mengubah distribusi setiap variable memiliki :

- Rata-rata (mean) = 0

- Nilai tengah (median) $\neq 0$
- Standar deviasi = 1

Pada penelitian ini, peneliti menerapkan `StandardScaler()` yang digunakan pada model *Naive Bayes*, K-NN, dan *VotingClassifier* (NB dan K-NN). Kegunaan `StandardScaler()` ini sendiri adalah meningkatkan performa model secara signifikan.

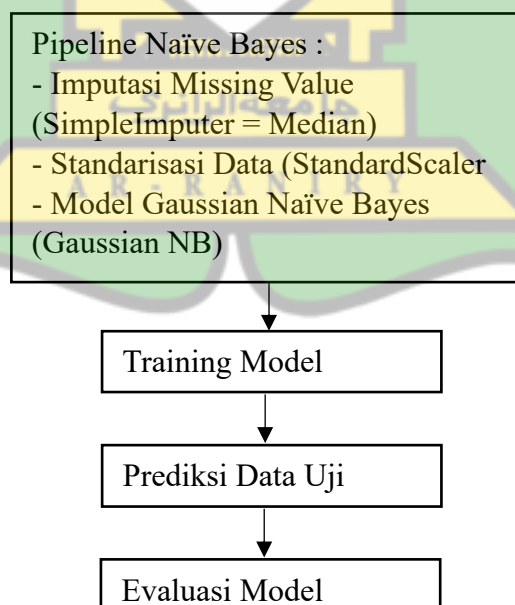
3.1.4 Algoritma *Naive Bayes*

Pada tahap selanjutnya sebelum mendapatkan output yang diinginkan, peneliti perlu pengimplementasian dari algoritma *Naive Bayes* yang sudah ditetapkan oleh peneliti sebelumnya.

Naive Bayes adalah sebuah konsep dari probabilitas yang dipakai untuk menentukan kelompok kelas dokumen teks dan juga dapat mengolah data dalam jumlah yang besar serta menghasilkan akurasi yang cukup tinggi. *Naive Bayes* memiliki tingkat akurasi yang lebih baik dibanding model classifier lainnya.

Pada proses training model menggunakan algoritma *Naive Bayes* ini, peneliti menggunakan pipeline (menggabungkan beberapa proses machine learning dalam satu tahapan secara beraturan) yang terdiri dari tahapan Preprocessing data dan tahapan model menjadi satu objek.

Proses penerapan algoritma *Naive Bayes* digambarkan pada gambar 3.3.



Gambar 3. 2 Diagram Penerapan Algoritma *Naive Bayes*

Diagram tersebut menggambarkan alur kerja dari penerapan algoritma *Naive Bayes* yang diimplementasikan melalui sebuah pipeline dalam proses *machine learning*. Pipeline tersebut digunakan dengan tujuan untuk menggabungkan tahapan preprocessing data dan pelatihan model kedalam satu alur yang terstruktur dan konsisten sampai dengan proses pelatihan model selesai/

1. Pipeline *Naive Bayes*

Pada tahap ini dilakukan proses preprocessing dan pemodelan secara berurutan, yaitu:

- Imputasi Missing Value (SimpleImputer – Median)

Digunakan untuk menangani data yang memiliki nilai kosong dengan menggantinya menggunakan nilai median. Pemilihan median bertujuan agar hasil imputasi tidak terpengaruh oleh nilai ekstrem (outlier).

- Standarisasi Data (StandardScaler)

Tahap ini bertujuan untuk menyamakan skala seluruh fitur numerik agar memiliki nilai mean = 0 dan standar deviasi = 1, sehingga distribusi data menjadi lebih stabil dan model dapat bekerja secara optimal.

- Model Gaussian *Naive Bayes*

Setelah data diproses, algoritma Gaussian *Naive Bayes* digunakan untuk membangun model klasifikasi berdasarkan asumsi distribusi normal pada fitur numerik.

2. Training Model

Pada tahap ini, pipeline dilatih menggunakan data latih (training data). Seluruh proses imputasi, standarisasi, dan pembelajaran model dilakukan secara otomatis dalam satu alur, sehingga mencegah terjadinya kebocoran data (data leakage).

3. Prediksi Data Uji

Model yang telah dilatih kemudian digunakan untuk memprediksi label pada data uji (testing data). Data uji diproses menggunakan preprocessing

yang sama seperti data latih, sehingga hasil prediksi tetap konsisten dan valid.

4. Evaluasi Model

Tahap terakhir adalah mengevaluasi performa model menggunakan beberapa metrik, yaitu:

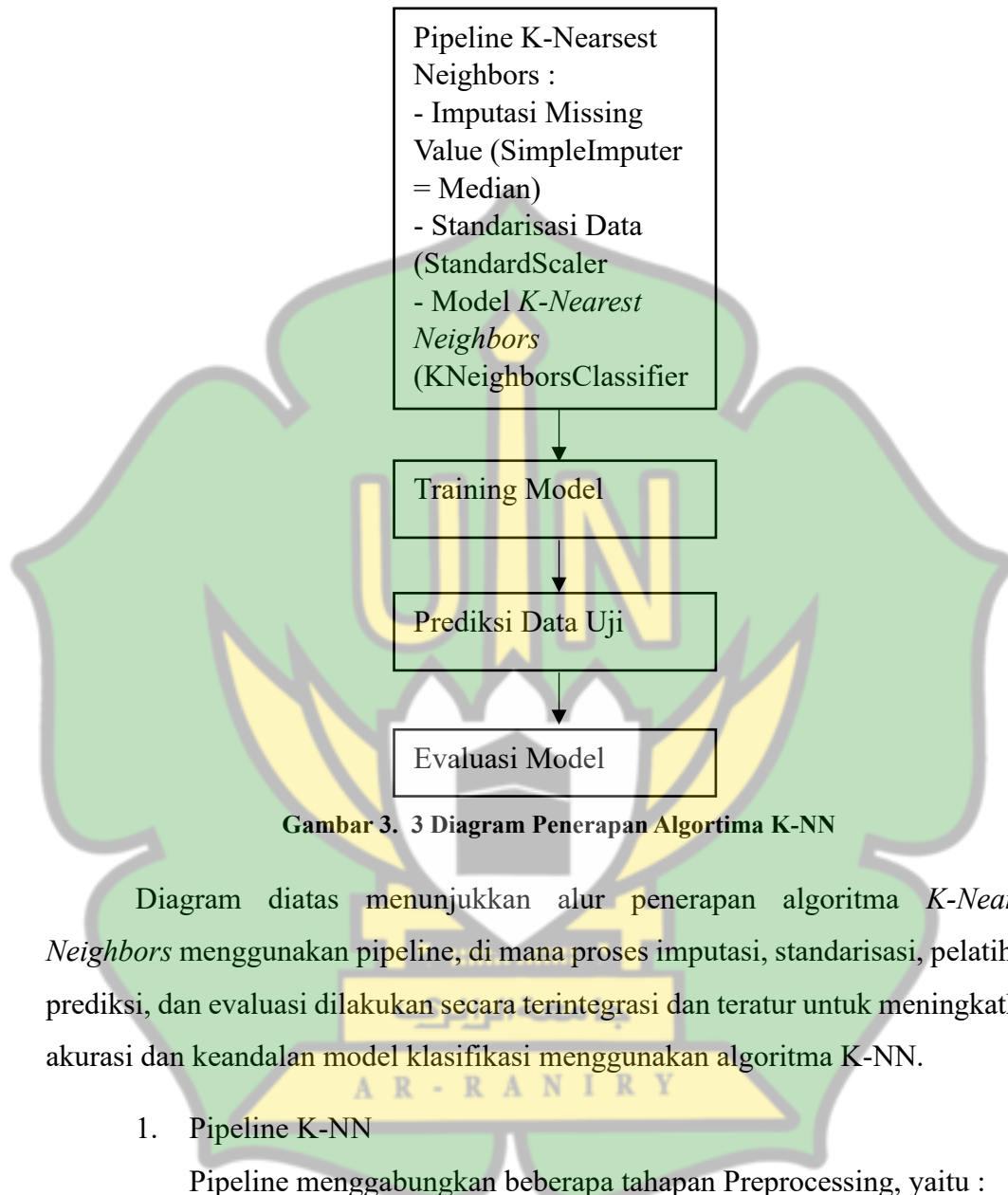
- Akurasi model, untuk mengukur tingkat ketepatan prediksi secara keseluruhan.
- Hasil klasifikasi (classification report), yang mencakup *precision*, *recall*, dan *f1-score*.
- Confusion matrix, untuk mengetahui perbandingan antara hasil prediksi dan data sebenarnya.
- Learning Curve, untuk mengetahui hubungan antara banyaknya data latih yang disediakan dengan performa model.
- ROC Curve, digunakan untuk mengevaluasi kemampuan model *Naive Bayes* dalam membedakan kelas positif dan negatif dengan memplot nilai True Positive Rate terhadap False Positive Rate.
- Nilai AUZ, untuk menunjukkan seberapa baik model dalam membedakan kelas positif dan negatif berdasarkan hasil grafik ROC Curve.

3.1.5 Algoritma K-Nearest Neighbors

Tahapan berikutnya, setelah menerapkan algoritma *Naive Bayes*, peneliti juga menerapkan algoritma *K-Nearest Neighbors* (K-NN) agar nantinya peneliti bisa menggabungkan kedua hasil prediksi algoritma ini dan menghasilkan model yang lebih akurat.

Pada penelitian ini, peneliti menggunakan *Distance Matric* untuk menghitung nilai k data terdekat dengan menggabungkan dua algoritma, yaitu *Euclidean Distance* (jarak Euclid) dan *Manhattan Distance* (jarak Manhattan / Cityblock). Pada percobaan menghitung jarak terdekat data menggunakan *Distance Matric*, peneliti mencoba melalui proses *GridsearchCV*.

Proses penerapan algoritma *K-Nearest Neighbors* digambarkan dalam pada gambar 3.4.



Gambar 3. 3 Diagram Penerapan Algoritma K-NN

Diagram diatas menunjukkan alur penerapan algoritma *K-Nearest Neighbors* menggunakan pipeline, di mana proses imputasi, standarisasi, pelatihan, prediksi, dan evaluasi dilakukan secara terintegrasi dan teratur untuk meningkatkan akurasi dan keandalan model klasifikasi menggunakan algoritma K-NN.

1. Pipeline K-NN

Pipeline menggabungkan beberapa tahapan Preprocessing, yaitu :

- Imputasi Missing Value (SimpleImputer – Median)

Digunakan untuk menggantikan nilai kosong pada fitur numerik dengan nilai median agar tidak dipengaruhi oleh outlier/data yang nilainya sangat berbeda atau aneh

- Standarisasi Data (StandardScaler)

Tahapan ini sangat penting pada algoritma K-NN karena perhitungan jarak (*distance matrix*) sangat berpengaruh terhadap perbedaan skala data

- Model *K-Nearest Neighbors*

Algoritma K-NN menjalankan klasifikasi berdasarkan kedekatan jarak antara data baru dengan data latih menggunakan nilai k tetangga terdekat

2. Training Model

Pada tahap ini, pipeline dilatih menggunakan data latih. Seluruh proses Preprocessing dan pelatihan model dilakukan secara otomatis dalam satu alur, sehingga mencegah terjadinya kebocoran data (*data leakage*).

3. Prediksi Data Uji

Model yang telah dilatih kemudian digunakan untuk memprediksi kelas pada data uji. Jarak antara data uji (*testing data*) dan data latih (*training data*) dihitung, kemudian kelas ditentukan berdasarkan kelas terbanyak dari nilai k tetangga terdekat.

4. Evaluasi Model

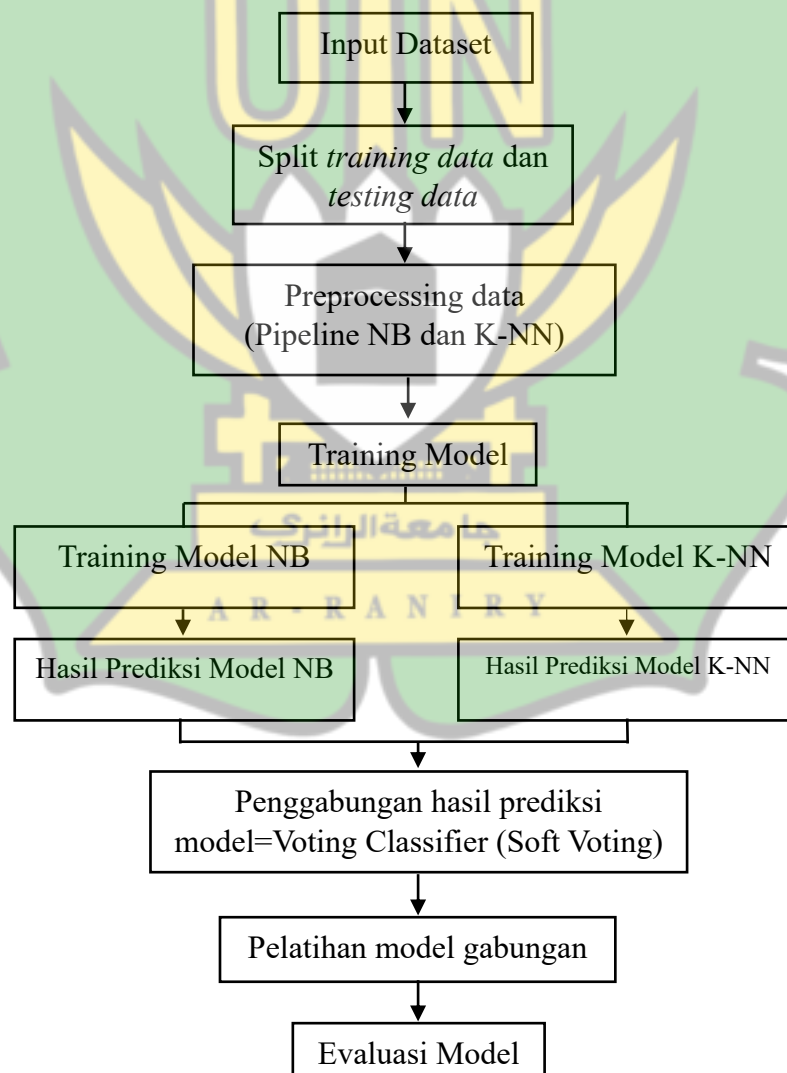
Performa model K-NN dievaluasi menggunakan beberapa output, yaitu:

- Akurasi model, untuk mengukur tingkat ketepatan prediksi model.
- Hasil klasifikasi (*classification report*), untuk melihat tingkat *precision*, *recall*, dan *f1-score* dari model yang telah dibuat.
- Confusion matrix, untuk mengetahui jumlah prediksi benar dan salah pada setiap kelas.
- Learning Curve, yaitu grafik yang digunakan untuk mengetahui hubungan antara jumlah data latih dengan performa model.
- ROC Curve, digunakan untuk mengevaluasi kemampuan model K-NN dalam membedakan kelas positif dan negatif berdasarkan nilai True Positive Rate dan False Positive Rate.

- Nilai AUZ, yaitu nilai yang menunjukkan seberapa baik model K-NN membedakan antara kelas positif dan negatif berdasarkan hasil grafik *ROC Curve*.
- Validation Curve K-NN, untuk mengetahui pengaruh perubahan *hyperparameter k* terhadap kinerja model K-NN.

3.1.6 Penggabungan Algoritma Naive Bayes dan K-NN

Setelah peneliti membangun kedua model secara terpisah menggunakan algoritma *Naive Bayes* dan K-NN, selanjutnya peneliti menggabungkan kedua hasil prediksi model algoritma tersebut untuk mendapatkan performa klasifikasi yang lebih optimal. Alasannya karena masing-masing algoritma tersebut memiliki karakteristik yang saling melengkapi. Berikut gambaran flowchart dari proses penggabungan kedua algoritma:



Gambar 3. 4 Diagram Penggabungan Algoritma

Pada proses penggabungan hasil prediksi kedua algoritma ini, peneliti menggunakan pendekatan *Voting Classifier*, yaitu sebuah metode ensemble learning untuk menggabungkan dua atau lebih model yang dapat memberikan kontribusi prediksi sebelum nantinya keputusan akhir ditentukan.

Pada pendekatan tersebut, peneliti menerapkan teknik *Soft voting*, yaitu proses pemilihan kelas berdasarkan probabilitas, bukan data mayoritas dari kedua algoritma yang digunakan. Teknik ini peneliti gunakan karena dapat mengambil nilai probabilitas yang dihasilkan *Naive Bayes* serta estimasi/perkiraan kelas yang dihasilkan K-NN.

Pada teknik *Soft Voting*, model cenderung mempertimbangkan seberapa yakin masing-masing model terhadap kelas target yang dipilih, sebagai contoh jika model dari *Naive Bayes* memilih “LULUS” dengan tingkat probabilitas 60% dan model dari K-NN memilih “LULUS” dengan tingkat probabilitas 90%, maka model gabungan akan menentukan kelas “LULUS” dengan tingkat probabilitas 75% (rata-rata dari probabilitas masing-masing model).

Penggabungan hasil prediksi algoritma *Naive Bayes* dan K-NN ini pada dasarnya menggunakan pipeline dari masing-masing algoritma yang telah dibuat sebelumnya. Pipeline yang telah dibuat mencakup imputasi missing value, standarisasi data, dan penerapan masing-masing algoritma.

3.1.7 Hasil Analisis Data

Hasil analisis data pada penelitian ini diperoleh melalui tahapan pemodelan dan evaluasi menggunakan penggabungan hasil prediksi algoritma *Naive Bayes* dan *K-Nearest Neighbors* (K-NN). Tahapan ini bertujuan untuk mengetahui kinerja masing-masing algoritma dan juga mengevaluasi efektivitas dari penggabungan hasil prediksi kedua model tersebut (*Naive Bayes* dan *K-Nearest Neighbors*) dalam meningkatkan akurasi klasifikasi status kelulusan mahasiswa.

3.1.8 Klasifikasi Data Baru (Inference)

Setelah model gabungan telah dibangun dan dilatih, model tersebut dapat digunakan untuk mengklasifikasi data baru. Model ini akan menghitung probabilitas untuk setiap kelas berdasarkan fitur-fitur data baru tersebut dan memilih kelas dengan probabilitas tertinggi sebagai hasil prediksi.

Tahapan inferensi ini juga dilakukan dengan menerapkan proses *preprocessing* yang sama pada data pelatihan, yaitu dengan menggunakan pipeline, yang meliputi tahapan imputasi nilai kosong dan standarisasi data sehingga konsistensi data pelatihan tetap terjaga dan hasil prediksi yang dihasilkan lebih akurat.

Hasil klasifikasi dari data baru ini memberikan informasi berupa prediksi status kelulusan mahasiswa, yaitu lulus tepat waktu (1) atau tidak lulus tepat waktu/belum lulus (0). Informasi ini nantinya dapat dimanfaatkan oleh pihak akademik sebagai dasar dalam pengambilan Keputusan (*decision making*), seperti melakukan pembinaan akademik secara dini terhadap mahasiswa yang berpotensi mengalami keterlambatan kelulusan. Dengan demikian, sistem cerdas yang peneliti kembangkan tidak hanya bertujuan sebagai alat prediksi, tetapi juga sebagai pendukung dalam perencanaan strategi peningkatan kualitas kelulusan mahasiswa.

3.2 Alat Bantu Penelitian

Alat bantu yang peneliti gunakan dalam penelitian ini berupa perangkat keras/*hardware* dan perangkat lunak/*software*. Perangkat keras yang peneliti gunakan adalah satu unit laptop HP Laptop 15s-du3577TU, dengan spesifikasi sebagai berikut :

- *Processor* : 11th Gen Intel(R) Core(TM) i3-1115G4 @ 3.00GHz 3.00 GHz
- *Display* : HD 15.6" diagonal (1366 x 768), *refresh rate* 60Hz
- *RAM* : 8 GB DDR4-3200 MHz RAM (2 x 4 GB)
- *Storage* : 512 GB PCIe® NVMe™ M.2 SSD

Perangkat lunak yang peneliti gunakan adalah sistem operasi *Windows 11 Home Single Language+Office Home and Student 2019*. Untuk bahasa pemrograman menggunakan bahasa Python yang dijalankan di Google Colaboratory.

3.3 Waktu dan Lokasi Penelitian

Penelitian ini dilakukan dalam jangka waktu 15 bulan, yang dimulai dari bulan Oktober 2024 sampai dengan bulan Januari 2026. Untuk lokasi penelitian, peneliti melakukan tahapan-tahapan penelitian di tempat tinggal peneliti, dan laboratorium program studi Teknologi Informasi



BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi Sistem dan Pengolahan Data

Pada tahap ini, seluruh proses yang telah peneliti rancang pada bab III diimplementasikan menggunakan bahasa pemrograman Python dengan memanfaatkan pustaka *machine learning Scikit-Learn* yang dijalankan melalui Google Colaboratory. Dataset mahasiswa yang diperoleh dan telah dikumpulkan dari Sistem Informasi Akademik UIN Ar-Raniry Banda Aceh digunakan sebagai sumber data utama dalam penelitian ini.

Jumlah data yang digunakan dalam penelitian ini sebanyak 1.828 data mahasiswa, yang mencakup mahasiswa dari Fakultas Sains dan Teknologi dan Fakultas Ilmu Sosial dan Ilmu Pemerintahan dengan rentang tahun masuk 2018 sampai dengan tahun 2021. Dataset tersebut kemudian dibagi menjadi data latih (*training data*) dan data uji (*testing data*) menggunakan teknik *stratified split*.

Rincian dataset nilai IP semester 1 sampai dengan 8, tahun masuk, dan status kelulusannya dapat dilihat pada tabel 4.1.

Tabel 4. 1 Dataset Mahasiswa SAINTEK dan FISIP

tahun_masuk	ip_1	ip_2	ip_3	ip_4	ip_5	ip_6	ip_7	ip_8	status_kelulusan
2018	3,36	3,71	3,88	3,50	3,71	3,92	3,79	0,00	lulus_tepat_waktu
2018	3,82	3,54	3,50	3,75	3,77	3,78	3,78	0,00	lulus_tepat_waktu
2018	3,18	3,38	3,23	3,41	3,52	3,84	4,00	0,00	lulus_tepat_waktu
2018	3,18	3,44	3,64	3,61	3,90	3,73	4,00	0,00	lulus_tepat_waktu
2018	3,32	3,43	3,29	3,27	3,16	3,50	3,86	0,00	lulus_tepat_waktu
Dst.....									
2019	3,43	3,86	3,37	3,56	3,77	3,76	3,74	3,87	lulus_tepat_waktu
2019	3,18	3,48	2,98	3,47	3,43	3,44	3,20	3,78	lulus_tepat_waktu
2019	3,64	4,00	3,81	3,92	3,68	3,90	3,47	4,00	lulus_tepat_waktu
2019	3,55	3,50	3,56	3,72	3,69	3,96	3,64	4,00	lulus_tepat_waktu
2019	3,64	3,67	3,74	3,74	3,26	3,79	3,59	4,00	lulus_tepat_waktu
Dst.....									
2020	3,83	3,65	3,88	3,83	3,96	3,82	4,00	0,00	lulus_tepat_waktu

2020	3,68	3,81	3,89	3,90	3,70	3,35	3,88	0,00	lulus_tepat_waktu
2020	3,82	3,67	3,74	3,57	3,52	3,88	3,92	0,00	lulus_tepat_waktu
2020	3,38	3,26	3,39	3,47	3,50	3,45	3,88	0,00	lulus_tepat_waktu
2020	3,82	3,87	3,78	3,89	3,74	3,94	3,67	0,00	lulus_tepat_waktu
Dst.....									
2021	3,29	3,41	3,36	2,56	3,06	3,41	0,00	0,00	belum_lulus
2021	1,51	0,67	1,94	0,14	0,27	0,50	0,00	0,00	belum_lulus
2021	3,70	3,27	3,33	1,12	2,39	2,32	0,00	0,00	belum_lulus
2021	3,39	2,71	3,03	2,82	2,60	3,14	0,00	0,00	belum_lulus
2021	3,83	3,47	3,52	3,60	3,52	3,68	0,00	0,00	belum_lulus
Dst.....									

4.2 Hasil Pra-Pemrosesan Data (Preprocessing Data)

Tahapan ini peneliti lakukan untuk memastikan kualitas data sebelum digunakan dalam proses pelatihan model. Tahapan Preprocessing Data yang peneliti terapkan meliputi antara lain :

4.2.1 Hasil Pembersihan Data (Cleaning Data)

Pada tahap ini, dilakukan proses penghapusan atribut-atribut yang tidak relevan terhadap pemodelan algoritma *Naive Bayes* dan K-NN nantinya. Atribut seperti NIM, Fakultas, dan jurusan perlu dihapus karena atribut tersebut tidak berpengaruh secara langsung terhadap proses klasifikasi dan pemodelan.

Selain itu, pada tahap ini peneliti juga melakukan konversi format angka yang awalnya masih menggunakan format koma (,) menjadi format titik (.) sebagai pemisah desimal pada atribut IP mahasiswa. Selain itu, seluruh nilai IP mahasiswa dikonversi ke dalam tipe data numerik bertipe float agar dapat diproses secara optimal oleh algoritma klasifikasi.

4.2.2 Pembentukan Target Biner

Pada tahap ini, peneliti melakukan proses pembentukan label target dalam bentuk numerik (*binary encoding*) dengan tujuan untuk menyesuaikan format data dengan kebutuhan algoritma yang digunakan (*Naive Bayes* dan K-NN). Status kelulusan mahasiswa yang awalnya berbentuk format data kategorikal, seperti *lulus*

tepat waktu, lulus tidak tepat waktu, dan belum lulus , kemudian dipetakan kedalam dua kelas, yaitu :

- Lulus tepat waktu $\Rightarrow 1$
- Lulus tidak tepat waktu/belum lulus $\Rightarrow 0$

Penggabungan kategori *lulus tidak tepat waktu* dengan *belum lulus* kedalam satu kelas dikarenakan kedua kategori tersebut memiliki makna yang sama dalam konteks penelitian, yaitu mahasiswa yang **belum berhasil** menyelesaikan studi sesuai dengan waktu ideal yang telah ditetapkan dan ditentukan oleh universitas.

Setelah proses pemetaan dilakukan, terdapat kolom target baru (*status_kelulusan*) yang digunakan sebagai label utama dalam proses pelatihan dan pengujian model. Label ini nantinya menjadi dasar bagi algoritma *Naive Bayes* dan K-NN untuk mempelajari pola hubungan antara nilai IP mahasiswa dengan status kelulusan yang dihasilkan.

4.2.3 Standarisasi data dan Penanganan Missing Value

Penerapan proses standarisasi data pada penelitian ini peneliti menggunakan *StandardScaler*, seluruh fitur numerik pada dataset berhasil ditransformasikan kedalam skala yang seragam dengan nilai rata-rata mendekati 0 dan standar deviasi mendekati 1. Tahapan ini menghasilkan distribusi data yang seimbang antar fitur.

Hasil standarisasi data ini memberikan dampak positif terhadap kinerja kedua algoritma (*Naive Bayes* dan K-NN) terutama pada algoritma K-NN yang menggunakan perhitungan jarak antar data menjadi lebih akurat karena setiap fitur memiliki kontribusi yang proporsional dalam menentukan tetangga terdekat.

4.3 Hasil Implementasi Algoritma Naive Bayes

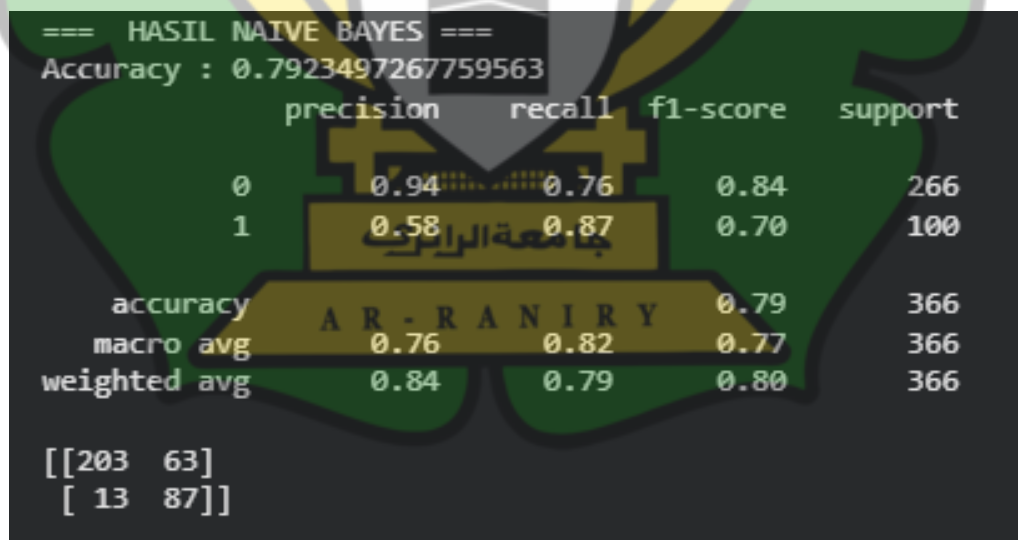
Model algoritma *Naive Bayes* dibangun menggunakan pipeline yang terdiri dari proses imputasi missing value menggunakan media, standarisasi data menggunakan *StandardScaler*, dan penerapan algoritma *Gaussian Naive Bayes*, dengan tujuan agar semua tahapan-tahapan mulai dari tahapan preprocessing data sampai dengan pelatihan model dapat dijalankan secara terstruktur dan konsisten.

Model kemudian dilatih menggunakan data latih (*training data*) dan diuji menggunakan data uji (*testing data*) untuk memperoleh hasil performa klasifikasi. Berdasarkan hasil pengujian yang peneliti lakukan, model *Naive Bayes* memperoleh nilai akurasi sebesar **79,23%**. Nilai tersebut menunjukkan bahwa model mampu mengklasifikasikan status kelulusan mahasiswa Fakultas Saintek dan Fisip dengan tingkat ketepatan yang **cukup baik**, dimana sebagian besar data uji berhasil diprediksi dengan benar

Model algoritma *Naive Bayes* ini menunjukkan kemampuan yang baik dalam mempelajari pola distribusi data numerik, khususnya terhadap nilai IP mahasiswa yang menjadi atribut utama dalam penelitian ini.

4.3.1 Hasil Evaluasi Model Algoritma *Naive Bayes*

Evaluasi terhadap performa dari model algoritma *Naive Bayes* dilakukan menggunakan beberapa metrik, yaitu *accuracy*, *classification report*, dan *confusion matrix*. Penggunaan beberapa metrik ini bertujuan untuk memberikan gambaran dari performa model *Naive Bayes* secara lebih komprehensif. Hasil evaluasi dapat dilihat pada gambar 4.1.



```

=== HASIL NAIVE BAYES ===
Accuracy : 0.7923497267759563

      precision    recall  f1-score   support

     0       0.94       0.76       0.84       266
     1       0.58       0.87       0.70       100

 accuracy      0.79       366
 macro avg     0.76       0.82       0.77       366
 weighted avg  0.84       0.79       0.80       366

[[203  63]
 [ 13  87]]
  
```

Gambar 4. 1 Hasil Evaluasi Model Naive Bayes

Berdasarkan gambar 4.1, hasil *classification report* model *Naive Bayes*, untuk kelas target 0 (lulus tidak tepat waktu/belum lulus), diperoleh nilai *precision* sebesar **0,94** yang artinya dari seluruh data mahasiswa Fakultas Saintek dan Fisip

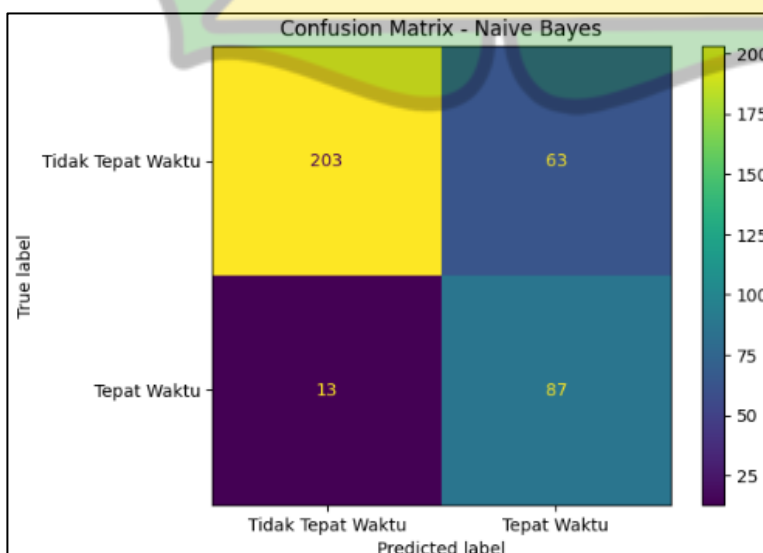
yang diprediksi sebagai kelas 0, sebanyak **94% benar**. Nilai *precision* yang sangat tinggi menunjukkan bahwa model memiliki tingkat ketepatan yang sangat baik dalam memprediksi mahasiswa yang termasuk ke dalam kategori tidak lulus tepat waktu/belum lulus. Ini mengindikasikan bahwa jumlah kesalahan dalam prediksi positif untuk kelas ini cenderung sangat sedikit.

Nilai *recall* sebesar **0,76** menunjukkan bahwa model mampu mengidentifikasi sekitar 76% dari seluruh data mahasiswa yang memang tidak lulus tepat waktu atau belum lulus. Hal ini menggambarkan bahwa meskipun model cukup akurat dalam memastikan kebenaran prediksi pada kelas 0, model masih memiliki keterbatasan dalam menjangkau seluruh data yang seharusnya termasuk dalam kategori tersebut.

Untuk kelas target 1 (lulus tepat waktu), diperoleh nilai *precision* sebesar **0,58** dan nilai *recall* sebesar **0,87**. Artinya nilai *recall* yang sangat tinggi ini mengindikasikan bahwa model *Naive Bayes* memiliki kemampuan yang sangat bagus dalam proses prediksi mahasiswa yang benar-benar lulus tepat waktu.

Namun demikian, nilai *precision* yang relative rendah (0,58) menunjukkan bahwa masih terdapat sejumlah prediksi terhadap mahasiswa lulus tepat waktu yang tidak sesuai dengan kondisi actual (keadaan yang sebenarnya). Dengan nilai sebesar 0,57, dapat diartikan bahwa dari seluruh mahasiswa yang diprediksi lulus tepat waktu oleh model *Naive Bayes*, hanya sekitar yang benar-benar lulus tepat waktu, sementara sisanya merupakan estimasi prediksi yang tidak akurat/keliru.

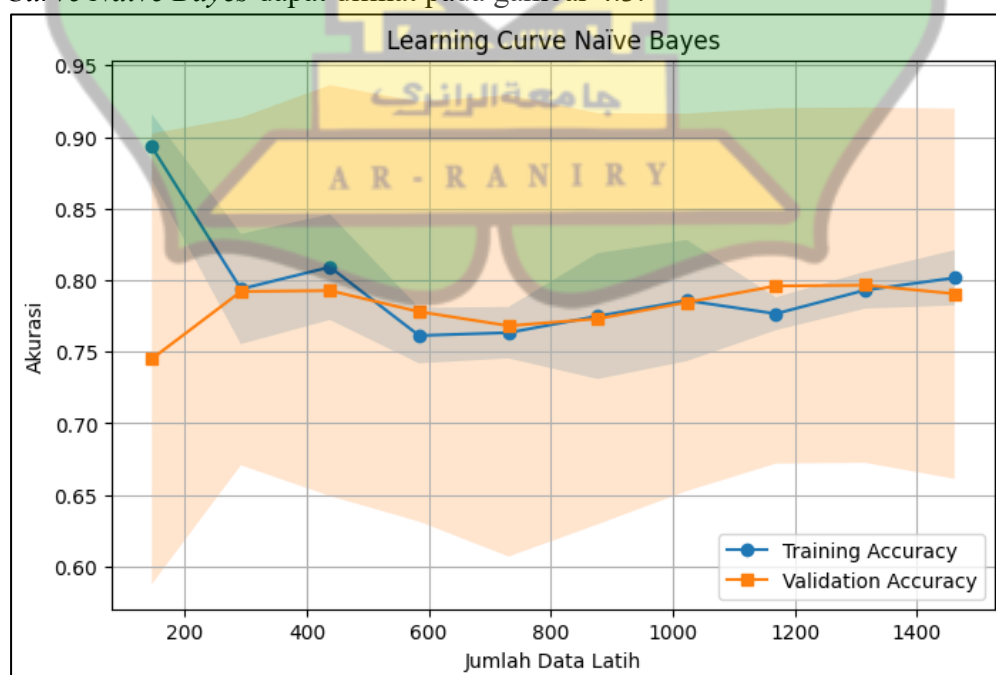
Untuk hasil *confusion matrix* dapat dilihat pada gambar 4.2 berikut :



Gambar 4. 2 Confusion Matrix Model Naive Bayes

- True Negative (203 data uji) : menunjukkan jumlah data mahasiswa yang diprediksi lulus tidak tepat waktu (kelas 0) oleh model, dan hasil p
- rediksi tersebut sesuai dengan keadaan yang sebenarnya. Artinya terdapat 203 data mahasiswa dari data uji yang berhasil diklasifikasikan dengan benar kedalam kelas 0 (lulus tidak tepat waktu).
- False Positive (63) : menunjukkan jumlah data mahasiswa yang diprediksi lulus tepat waktu (kelas 1) oleh model, tapi sebenarnya mahasiswa tersebut lulus tidak tepat waktu.
- False Negative (17) : menunjukkan jumlah data mahasiswa yang diprediksi lulus tidak tepat waktu oleh model, tetapi kenyataannya mahasiswa tersebut lulus tepat waktu.
- True Positive (87) : menunjukkan jumlah data mahasiswa yang diprediksi lulus tepat waktu oleh model dan sesuai dengan keadaan yang sebenarnya.

Hasil evaluasi model *Naive Bayes* yang telah dilatih dan diuji juga dapat dilihat performanya melalui grafik *Learning Curve* untuk mengetahui hubungan antara jumlah data latih dengan performa model *Naive Bayes*. Grafik *Learning Curve Naive Bayes* dapat dilihat pada gambar 4.3.



Gambar 4. 3 Grafik Learning Curve Model Naive Bayes

1. Pada Akurasi Training :

- Pada jumlah data kecil (~150): Akurasi training sangat tinggi (± 0.89). Ini menunjukkan model sangat menyesuaikan diri dengan data latih
- Namun saat data bertambah: Akurasi training menurun dan stabil di kisaran 0.77 – 0.80. Artinya model mulai belajar pola umum, bukan hanya menghafal data

2. Pada Akurasi Validasi :

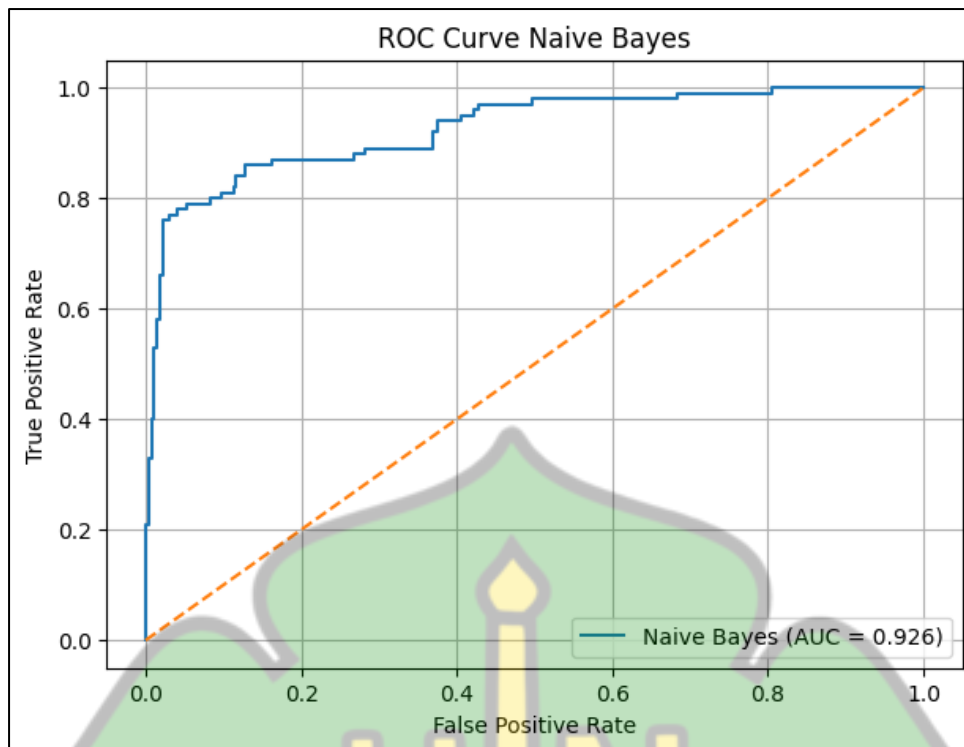
- Pada data sedikit : Akurasi validasi lebih rendah (± 0.74). Ini wajar karena model belum mempunyai cukup data untuk belajar pola
- Saat data bertambah : Akurasi validasi naik dan stabil di sekitar 0.78 – 0.80. Hal ini menunjukkan performa model pada data baru semakin baik

Setelah jumlah data latih melebihi 800 data, nilai akurasi validasi mengalami peningkatan dan semakin mendekati nilai akurasi training. Hal ini menunjukkan bahwa model *Naive Bayes* mulai mencapai kondisi yang lebih stabil dalam melakukan generalisasi terhadap data baru.

Seiring dengan bertambahnya jumlah data latih, akurasi validasi cenderung meningkat dan konvergen dengan akurasi training, yang menandakan bahwa model tidak mengalami overfitting dan memiliki kemampuan generalisasi yang cukup baik.

Namun demikian, peningkatan performa yang terjadi tidak terlalu signifikan setelah titik tertentu, sehingga dapat disimpulkan bahwa model *Naive Bayes* memiliki keterbatasan dalam menangkap pola data yang lebih kompleks, terutama pada kasus prediksi kelulusan yang melibatkan banyak variabel akademik

Selain itu, hasil evaluasi terhadap model algoritma *Naive Bayes* juga bisa ditampilkan melalui grafik ROC Curve untuk melihat seberapa bagus model membedakan antara kelas negatif dengan kelas positif, Grafiknya dapat dilihat pada gambar 4.4.



Gambar 4. 4 Grafik ROC Curve Model Naive Bayes

Berdasarkan grafik, kurva berada jauh di atas garis diagonal putus-putus, yang menandakan bahwa model memiliki kemampuan klasifikasi yang baik. Meskipun demikian, bentuk kurva ROC yang tidak halus dan terlihat bertingkat disebabkan oleh perubahan nilai threshold yang bersifat diskrit serta keterbatasan jumlah data uji yang digunakan.

Nilai AUC sebesar **0.926** mengindikasikan bahwa *Naive Bayes* mampu membedakan kelas positif dan negatif dengan tingkat akurasi sekitar 92%, sehingga performa model termasuk dalam kategori sangat baik (*excellent*).

4.4 Hasil Implementasi Algoritma K-NN

Pengujian model algoritma *K-Nearest Neighbors* (K-NN) juga dilakukan dengan menggunakan pipeline preprocessing data yang sama dengan model *Naive Bayes*, yang meliputi tahapan imputasi missing value menggunakan median, standarisasi data dengan menggunakan *StandardScaler*, dan penerapan algoritma *KNeighborsClassifier*.

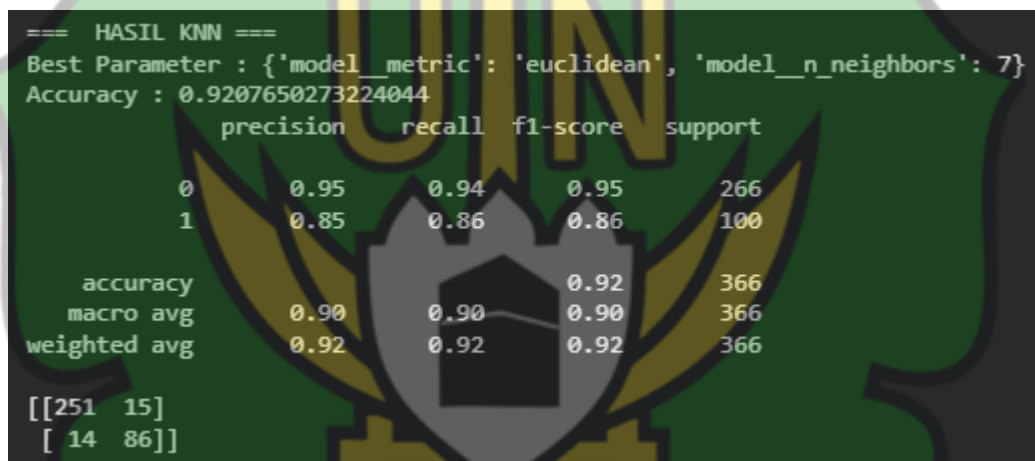
Model K-NN ini dilatih dengan *training data* dan diuji dengan *testing data* yang sudah dibagi pada tahap *splitting data* sebelumnya (80% *training data* =

1.462, dan 20% *testing data* = 366) untuk mengukur performa klasifikasi terhadap status kelulusan mahasiswa.

Berdasarkan hasil pengujian, diperoleh nilai akurasi sebesar **92,07%**. Nilai ini menunjukkan bahwa model K-NN dapat mengidentifikasi pola kedekatan di antara data mahasiswa dengan akurasi yang sangat baik berdasarkan nilai Indeks Prestasi semester mahasiswa.

4.4.1 Hasil Evaluasi Model Algoritma K-NN

Evaluasi model algoritma K-NN dilakukan setelah proses pelatihan model. berdasarkan hasil pengujian, diperoleh parameter terbaik (*hyperparameter tuning*) yaitu penggunaan metrik jarak Euclidean dengan jumlah tetangga terdekat (*n_neighbors*) sebanyak 9.



```

=== HASIL KNN ===
Best Parameter : {'model_metric': 'euclidean', 'model_n_neighbors': 7}
Accuracy : 0.9207650273224044

```

	precision	recall	f1-score	support
0	0.95	0.94	0.95	266
1	0.85	0.86	0.86	100
accuracy			0.92	366
macro avg	0.90	0.90	0.90	366
weighted avg	0.92	0.92	0.92	366

```

[[251 15]
 [ 14 86]]

```

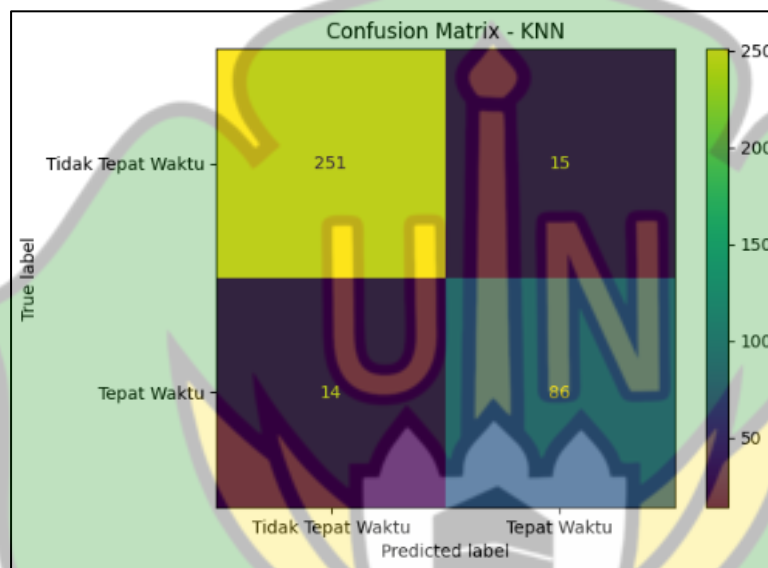
Gambar 4. 5 Hasil Evaluasi Model K-NN

Berdasarkan hasil *classification report*, performa model K-NN pada masing-masing kelas target (kelas 0 dan 1) diperoleh hasil berikut :

- Pada kelas 0 (lulus tidak tepat waktu/belum lulus), model K-NN memperoleh nilai *precision* dan *recall* sebesar 0,95 dan 0,94 yang artinya sebanyak sekitar 95% data mahasiswa yang diprediksi kedalam kelas 0, diprediksi dengan benar, dan sebanyak 94% juga model mampu mengenali data mahasiswa yang lulus tidak tepat waktu/belum lulus.

- Pada kelas 1 (lulus tepat waktu), model K-NN memperoleh nilai *precision* dan *recall* sebesar 0,85 dan 0,86 dalam memprediksi status kelulusan mahasiswa, yang artinya model ini mampu mengenali sekitar 85% dan berhasil memprediksi sebanyak 86% data mahasiswa kedalam kelas target lulus tepat waktu (kelas 1).

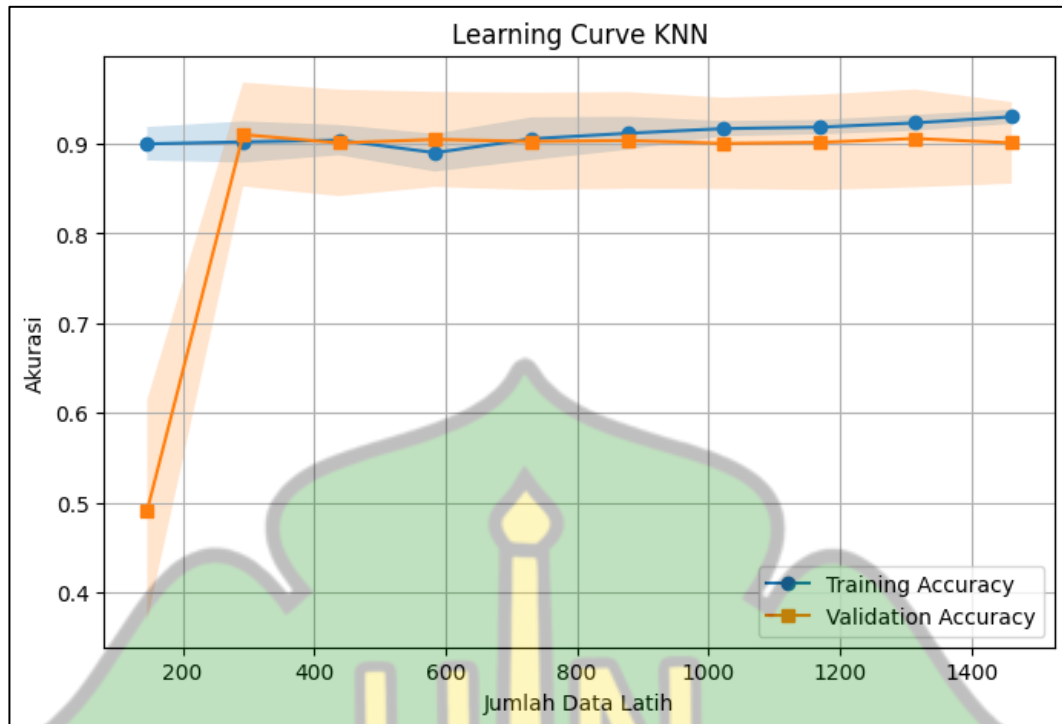
Untuk hasil *confusion matrix* dari model K-NN, diperoleh hasil yang dapat dilihat pada gambar 4.6, dari total 366 data uji mahasiswa :



Gambar 4. 6 Confusion Matrix Model K-NN

- 251 data berhasil diklasifikasikan dengan benar sebagai kelas 0 (true negative),
- 15 data salah diklasifikasikan sebagai kelas 1 (false positive),
- 14 data salah diklasifikasikan sebagai kelas 0 (false negative).
- 86 data berhasil diklasifikasikan dengan benar sebagai kelas 1 (true positive).

Selain itu, hasil evaluasi model K-NN dapat dilihat melalui grafik *learning curve* untuk mengetahui hubungan antara jumlah data latih dengan performa dari model K-NN yang dilatih. Grafik *Learning Curve* K-NN dapat dilihat pada gambar 4.7.



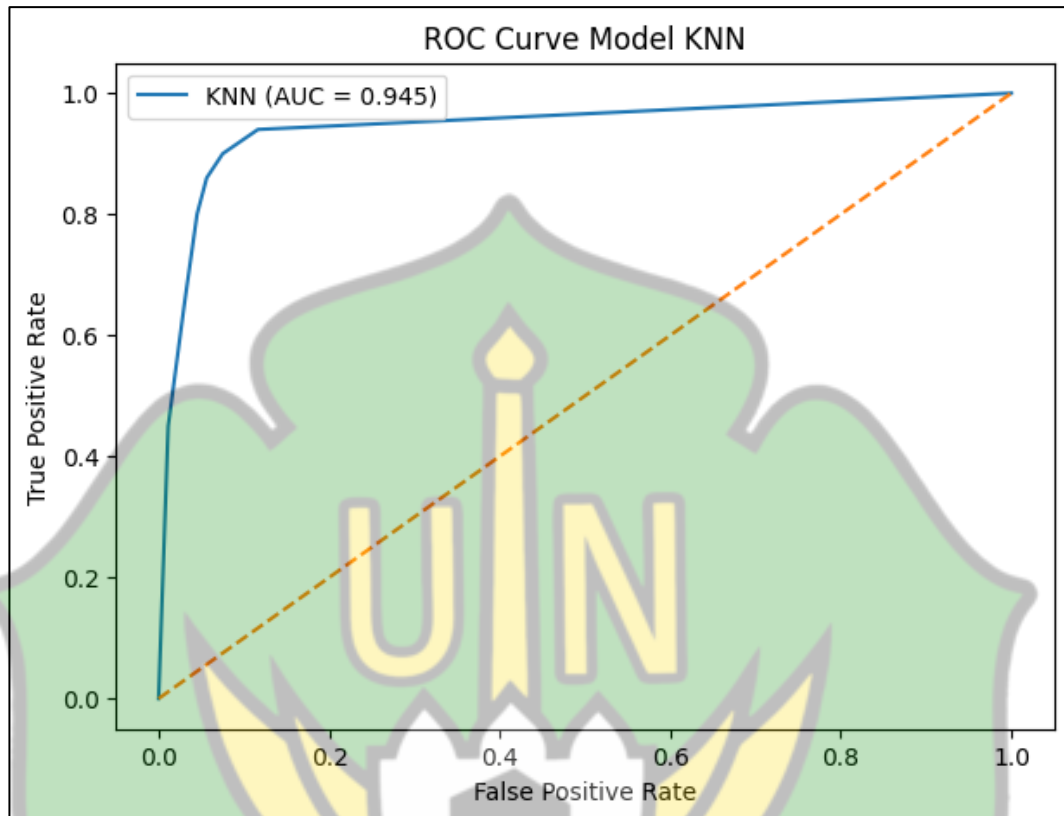
Gambar 4. 7 Grafik Learning Curve Model K-NN

Dari grafik terlihat bahwa pada jumlah data latih yang kecil, akurasi training sudah berada pada nilai yang cukup tinggi, namun akurasi validasi masih relatif rendah. Hal ini menunjukkan bahwa pada tahap awal pelatihan, model KNN cenderung menyesuaikan diri dengan data latih namun belum mampu melakukan generalisasi dengan baik akibat keterbatasan jumlah data. Seiring dengan bertambahnya jumlah data latih, akurasi validasi mengalami peningkatan yang signifikan dan mulai mendekati akurasi training.

Setelah jumlah data latih melebihi sekitar 300–400 data, kedua kurva menunjukkan pola yang semakin stabil dan konvergen, yang menandakan bahwa model KNN memiliki kemampuan generalisasi yang baik serta tidak mengalami overfitting secara signifikan. Namun, meskipun performa model sudah tinggi dan stabil, peningkatan akurasi yang terjadi setelah titik tersebut relatif kecil, sehingga dapat disimpulkan bahwa KNN telah mencapai batas optimal performanya pada dataset yang digunakan dan memiliki keterbatasan dalam meningkatkan akurasi lebih lanjut tanpa penambahan fitur atau optimasi parameter yang lebih kompleks.

Hasil evaluasi dari model K-NN juga dapat dilihat dari ROC Curve dan Nilai AUC untuk melihat kemampuan model K-NN dapat membedakan antara

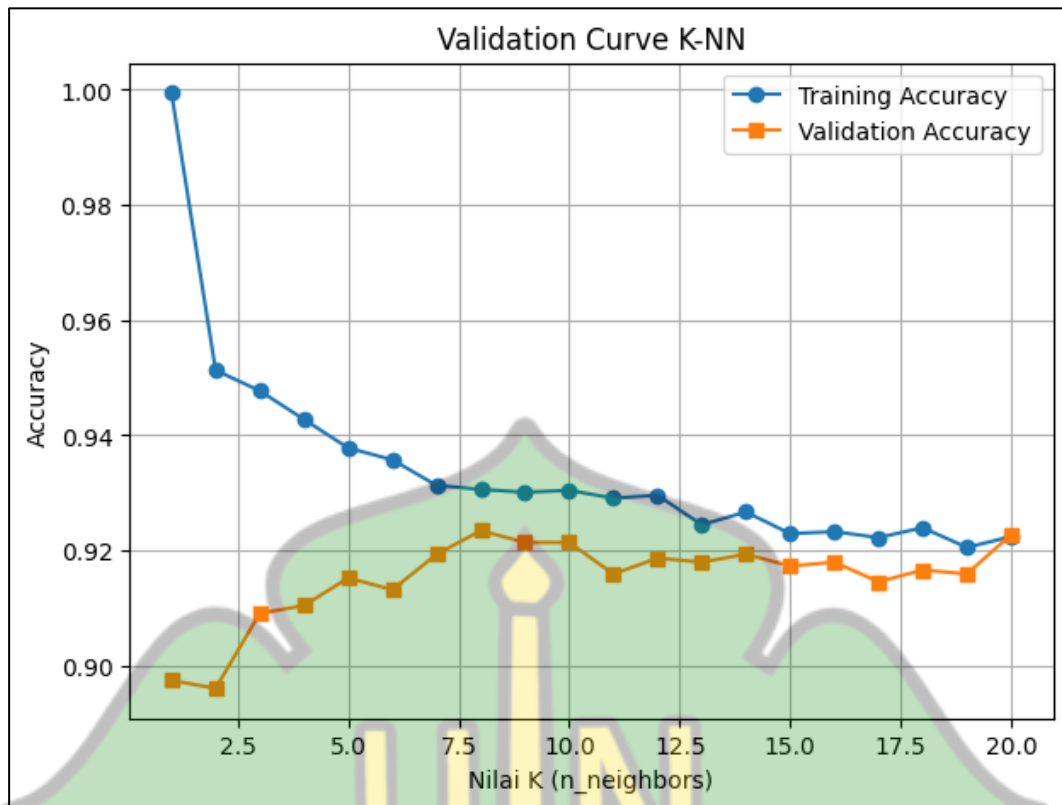
kelas negatif dan kelas positif. Grafik *ROC Curve* K-NN dapat dilihat pada gambar 4.8.



Gambar 4. 8 Grafik ROC Curve Model K-NN

Grafik diatas menunjukkan garis dari kurva ROC mendekati sudut kiri atas dan menjauhi garis putus-putus, yang menandakan bahwa model memiliki kemampuan yang sangat baik dalam membedakan kelas positif dan negatif. Nilai AUC sebesar 0.945 menunjukkan bahwa model KNN memiliki tingkat performa yang sangat tinggi, di mana model mampu melakukan pemisahan kelas dengan tingkat keakuratan sekitar **94,5%**.

Terakhir hasil evaluasi model K-NN juga dapat dilihat melalui grafik *validation curve* guna mengetahui pengaruh nilai k terhadap akurasi training dan akurasi validasi. Grafik *Validation Curve* K-NN dapat dilihat pada gambar 4.9.



Gambar 4. 9 Grafik Validation Curve Model K-NN

Pada grafik, nilai k yang kecil (1-3), akurasi training sangat tinggi sedangkan akurasi validasi sangat rendah, yang mengindikasikan bahwa terjadinya *overfitting*. Seiring meningkatnya nilai k , akurasi training cenderung menurun dan akurasi validasi meningkat dan mencapai kondisi yang lebih stabil. Hal ini terjadi di kisaran nilai k sedang (7-11).

Tetapi jika nilai k membuat nilai akurasi training yang menurun dratis dapat menyebabkan terjadinya *underfitting* pada model. oleh sebab itu pemilihan nilai k pada rentang tengah dapat menghasilkan performa terbaik dari model K-NN.

4.5 Perbandingan Hasil Penelitian dengan Penelitian Terdahulu.

Dari hasil penelitian model *Naive Bayes* dan K-NN terhadap prediksi status kelulusan mahasiswa yang diteliti, berikut akan dibahas perbandingan hasil yang diperoleh dengan penelitian terdahulu guna melihat keunggulan dan perbedaan performa model yang digunakan.

4.5.1 Perbandingan Model *Naive Bayes*

Penelitian yang dilakukan oleh Nazla Chairina (2023) dengan judul “Penggunaan Metode *Naive Bayes* untuk Memprediksi Kelulusan Mahasiswa Pendidikan Teknologi Informasi Menggunakan *RapidMiner*” memperoleh tingkat

akurasi sebesar **68.22%** dari total 598 data mahasiswa jurusan Pendidikan Teknologi Informasi. Sementara itu, pada penelitian ini model *Naive Bayes* menghasilkan nilai akurasi yang lebih tinggi yaitu sebesar **79,23%**, yang mengindikasikan bahwa penerapan teknik preprocessing data, normalisasi, serta pembagian data latih dan data uji yang digunakan memberikan pengaruh positif terhadap kinerja model.

Selain penerapan tahapan-tahapan yang disebutkan yang membuat perbedaan pada tingkat akurasi kedua penelitian, jumlah dataset mahasiswa yang digunakan untuk proses klasifikasi juga berbeda. Penelitian ini menggunakan data mahasiswa yang lebih banyak daripada penelitian terdahulu.

Hal ini tentunya memberikan keuntungan dalam proses pelatihan model *Naive Bayes*, karena model dapat mempelajari pola data yang lebih beragam dan representative, dan juga berkontribusi terhadap peningkatan performa prediksi status kelulusan mahasiswa apabila dibandingkan dengan penelitian terdahulu yang menggunakan jumlah data yang lebih sedikit.

4.5.2 Perbandingan Model *K-Nearest Neighbors*

Perbandingan performa model K-NN dalam penelitian ini mengacu pada penelitian terdahulu yang dilakukan oleh Lalu Abd Rahman Hakim (2020) yang juga menerapkan algoritma K-NN untuk memprediksi status kelulusan mahasiswa.

Pada penelitian tersebut, didapatkan tingkat akurasi tertinggi sebesar **80%** dengan pendekatan K-Fold Cros Validation, dan sebesar **98%** menggunakan metrik *confusion matrix* pada nilai k tertentu. Sedangkan pada penelitian yang dilakukan oleh peneliti, model K-NN menghasilkan tingkat akurasi sebesar **92,07%** yang menunjukkan bahwa model yang dikembangkan memiliki performa yang lebih stabil dan konsisten dalam melakukan klasifikasi data kelulusan mahasiswa.

Perbedaan hasil akurasi ini dipengaruhi oleh beberapa faktor, di antaranya perbedaan jumlah data, karakteristik dataset, teknik preprocessing data, serta strategi pembagian data latih dan data uji yang digunakan.

4.6 Hasil Implementasi Model Hybrid (*Naive Bayes* dan K-NN)

Proses implementasi model hybrid antara model *Naive Bayes* dengan model K-NN dilakukan dengan pendekatan *soft voting classifier*, yaitu dengan menggabungkan hasil dari nilai probabilitas output dari masing-masing model (*Naive Bayes* dan K-NN) untuk menentukan kelas akhir.

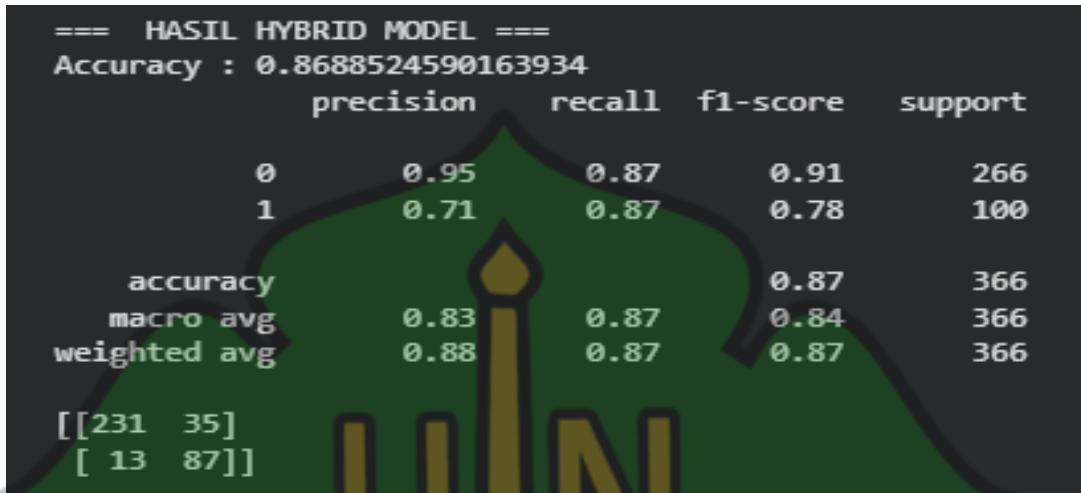
Sebelum masuk ke proses pelatihan model, penerapan model hybrid juga terlebih dahulu melewati proses *preprocessing* data yang sama seperti pada model *Naive Bayes* dan K-NN, yaitu penanganan *missing value* menggunakan metode imputasi median, normalisasi data menggunakan *standardisasi scaler*, serta penerapan untuk kedua algoritma model *Naive Bayes* dan K-NN. Seluruh proses tersebut diintegrasikan dalam pipeline untuk menjaga konsistensi dari alur pemrosesan data dan menghindari terjadi data leakage.

Pada proses penggabungan hasil prediksi kedua model (*Naive Bayes* dan K-NN), hasil prediksi dari model *Naive Bayes* dan K-NN dengan output tingkat probabilitasnya terhadap kedua kelas target (0 dan 1) akan digabungkan, dan diambil rata-rata dari hasil gabungan probabilitas keduanya. Contoh :

1. Output Prediksi Model *Naive Bayes*
 Kelas 0 : 0.30
 Kelas 1 : 0.70
2. Output Prediksi Model K-NN
 Kelas 0 : 0.20
 Kelas 1 : 0.80
3. *VotingClassifier* (Gabungan hasil prediksi)
 Kelas 0 : $(0.30 + 0.20) / 2 = 0.25$
 Kelas 1 : $(0.70 + 0.80) / 2 = 0.75$
4. Keputusan Akhir
 Kelas 1 : 0.75 (Lebih besar), maka prediksi akhir = kelas 1

4.6.1 Hasil Evaluasi Model Hybrid

Evaluasi model hybrid gabungan antara model *Naive Bayes* dan K-NN juga dilakukan dengan beberapa metrik seperti *accuracy*, *classification report*, dan *confusion matrix*.

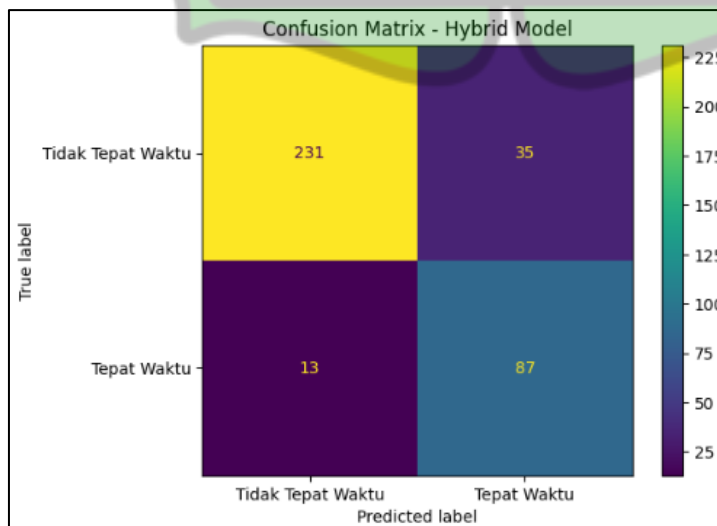


Gambar 4. 10 Hasil Evaluasi Model Hybrid

Dari hasil *classification report*, diperoleh nilai *precision* dan *recall* untuk masing-masing kelas (0 dan 1) sebagai berikut :

- Pada kelas 0 (lulus tidak tepat waktu dan belum lulus), model hybrid memperoleh nilai *precision* sebesar 95%, *recall* sebesar 87%, dan *f1-score* sebesar 91%.
- Pada kelas 1 (lulus tepat waktu), model memperoleh nilai *precision* sebesar 71%, *recall* sebesar 87%, dan *f1-score* sebesar 78%.

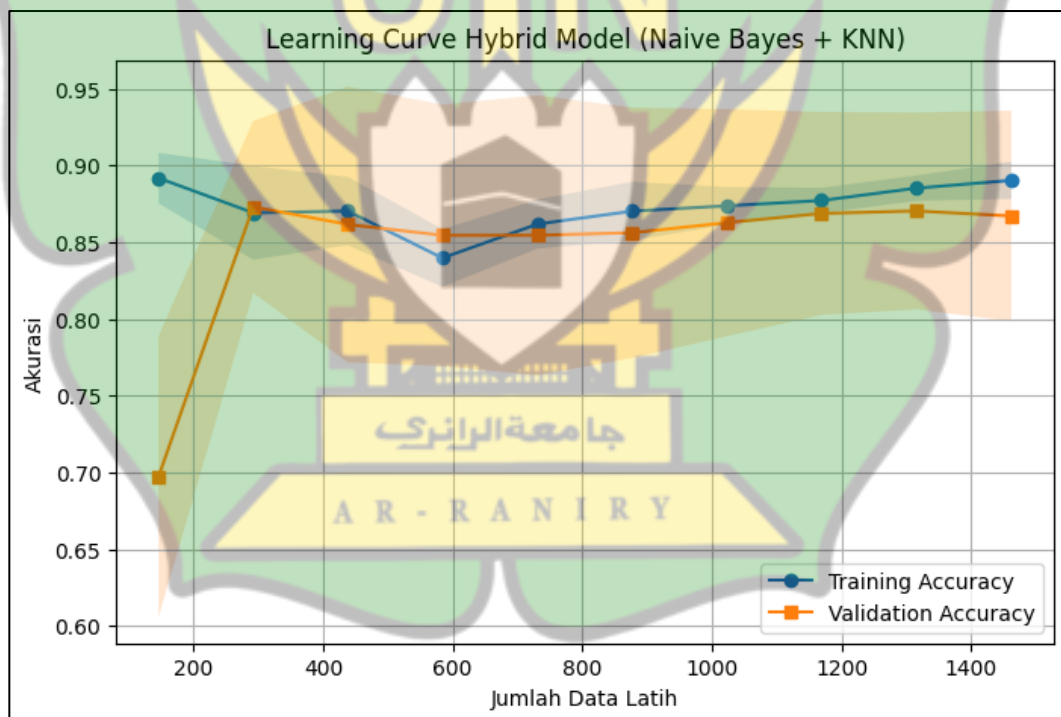
Sedangkan untuk *confusion matrix* dari model hybrid, diperoleh hasil pada gambar 4.11.



Gambar 4. 11 Confusion Matrix Model Hybrid

- True Negative (TN) : 231 data mahasiswa berhasil diklasifikasikan dengan benar sebagai kelas 0 (lulus tidak tepat waktu atau belum lulus),
- False Positive (FP) : 13 data salah diklasifikasikan sebagai kelas 1 (lulus tepat waktu),
- False Negative : 35 data salah diklasifikasikan sebagai kelas 0 (lulus tidak tepat waktu atau belum lulus).
- True Positive : 87 data berhasil diklasifikasikan dengan benar sebagai kelas 1 (lulus tepat waktu).

Sama seperti model *Naive Bayes* dan K-NN, hasil evaluasi dari model hybrid ini juga dapat dilihat dari grafik *learning curve*, ROC Curve, dan nilai AUC. Grafik-grafik ini digunakan dengan tujuan untuk mengetahui seberapa bagus performa model gabungan dari hasil prediksi model *Naive Bayes* dan K-NN dapat memprediksi status kelulusan mahasiswa.



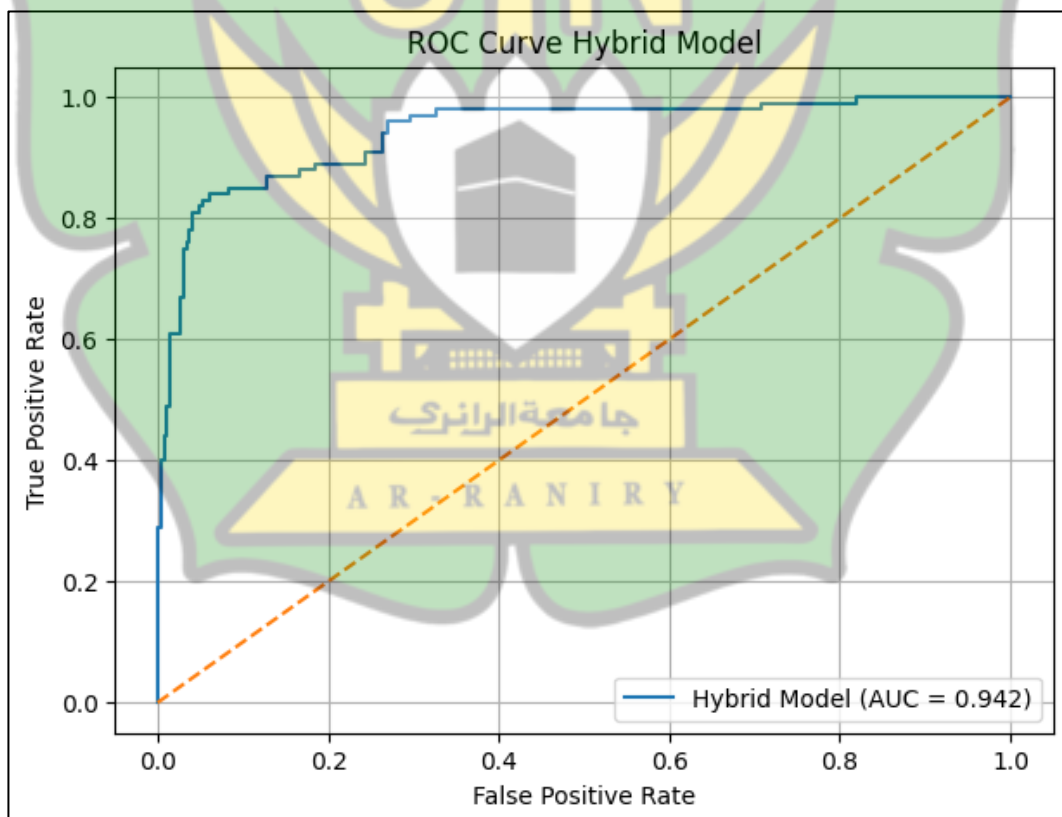
Gambar 4. 12 Learning Curve Model Hybrid

Dari gambar 4.12, diperoleh grafik *learning curve*, dapat dilihat bahwa terjadi peningkatan terhadap performa model dari akurasi training dan akurasi validation sejalan dengan bertambahnya jumlah dari data latih. Hal ini

menunjukkan bahwa jumlah data latih pada model hybrid sangat berpengaruh terhadap performa pelatihan model.

Walaupun pada jumlah data yang kecil, terdapat selisih yang cukup jelas antara kurva akurasi training dan validation, yang menunjukkan bahwa model hybrid ini masih belum optimal dalam melakukan generalisasi atau prediksi yang akurat pada data yang baru. Hal tersebut dikarenakan pada rentang jumlah data yang sedikit, model masih kekurangan informasi untuk proses pembelajaran, sehingga proses tersebut masih terbatas.

Performa dari model hybrid ini juga dapat dilihat melalui grafik ROC Curve dan nilai AUC, dengan tujuan dapat diketahui tingkat performa model dalam membedakan antara kelas negatif dan kelas positif. Dalam penelitian ini, kelas negatif diberi label kelas 0 (lulus tidak tepat waktu/belum lulus) dan kelas positif diberi label kelas 1 (lulus tepat waktu).



Gambar 4. 13 Grafik ROC Curve Model Hybrid

Dari gambar 4.13, dapat dilihat grafik ROC Curve model hybrid, dapat dilihat bahwa kurva ROC berada jauh dari garis putus-putus diagonal, yang berarti

model hybrid dapat mengklasifikasikan status kelulusan mahasiswa dengan sangat baik. Terlebih bentuk kurva yang mendekati sudut kiri menandakan bahwa model mampu membedakan antara kelas negatif dan kelas positif dengan baik, meskipun terdapat sedikit kekurangan yang diakibatkan distribusi data pada kelas negatif dan kelas positif tidak seimbang.

Nilai AUC dari grafik ROC Curve terhadap model hybrid yang mendekati ke angka 1 (**0,949**) yang mana model ini memiliki skor penilaian yang sangat tinggi (*excellent*), berarti model memiliki 94,93% peluang untuk memberikan skor prediksi yang lebih tinggi pada mahasiswa yang benar-benar lulus tepat waktu. Dengan kata lain, model sangat andal dalam membedakan dua kelas target, yaitu kelas negatif dan kelas positif.

4.7 Ringkasan Performa Model Klasifikasi

Model	Accuracy (%)	Precision	Recall	F1-Score	AUC
<i>Naive Bayes</i>	79,23%	0,76	0,82	0,77	0,920
K-NN	92,07%	0,90	0,90	0,90	0,949
Hybrid (NB+K-NN)	89,34%	0,83	0,87	0,84	0,9493

Tabel 4. 2 Perbandingan Performa Model

Berdasarkan tabel 4.4, diperoleh hasil pengujian yang ditampilkan pada Tabel perbandingan performa model, dapat diketahui bahwa model K-Nearest Neighbor (K-NN) menghasilkan performa terbaik secara keseluruhan dibandingkan dengan model *Naive Bayes* dan Hybrid. K-NN memperoleh nilai akurasi sebesar **92,07%**, dengan nilai precision, recall, dan F1-score yang seimbang yaitu masing-masing sebesar 0,90, serta nilai AUC sebesar 0,949, yang menunjukkan kemampuan klasifikasi dan diskriminasi kelas yang sangat baik. Hal ini mengindikasikan bahwa K-NN mampu menangkap pola hubungan antar data

mahasiswa dengan lebih efektif, terutama pada dataset yang memiliki karakteristik numerik dan pola non-linear.

Model *Naive Bayes* menunjukkan performa yang lebih rendah dibandingkan dua model lainnya, dengan nilai akurasi sebesar **79,23%** dan F1-score sebesar 0,77, meskipun memiliki nilai AUC yang cukup tinggi yaitu 0,920. Hal ini menunjukkan bahwa meskipun *Naive Bayes* mampu membedakan kelas dengan cukup baik, asumsi independensi antar fitur yang dimiliki oleh metode ini membatasi kemampuannya dalam menangkap hubungan kompleks antar variabel akademik, sehingga berdampak pada penurunan akurasi dan keseimbangan metrik klasifikasi.

Sementara itu, model Hybrid (*Naive Bayes* + K-NN) menghasilkan performa yang lebih baik dibandingkan *Naive Bayes* tunggal, dengan akurasi sebesar **89,34%**, precision 0,83, recall 0,87, dan F1-score 0,84. Nilai AUC sebesar 0,9493 yang hampir setara dengan K-NN menunjukkan bahwa hybrid model memiliki kemampuan diskriminasi kelas yang sangat baik. Namun, akurasi hybrid masih sedikit lebih rendah dibandingkan K-NN, yang mengindikasikan bahwa meskipun pendekatan ensemble mampu meningkatkan stabilitas dan keseimbangan prediksi, kontribusi *Naive Bayes* dalam hybrid model juga membawa keterbatasan yang mempengaruhi performa akhir.



BAB V PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan, maka dapat disimpulkan beberapa poin berikut:

1. Pengembangan Sistem Cerdas Prediksi Kelulusan

Pengembangan sistem cerdas prediksi kelulusan mahasiswa tepat waktu telah berhasil dilakukan dengan menerapkan algoritma *Naive Bayes* dan *K-Nearest Neighbors* (K-NN). Sistem yang dibangun mampu melakukan proses klasifikasi status kelulusan mahasiswa ke dalam dua kelas, yaitu lulus tepat waktu dan tidak lulus tepat waktu atau belum lulus, berdasarkan data akademik mahasiswa yang digunakan sebagai input. Selain itu, tahapan preprocessing data, pembentukan target biner, serta penerapan teknik stratified split terbukti berperan penting dalam meningkatkan kualitas data, menjaga keseimbangan distribusi kelas, serta meningkatkan stabilitas dan performa model yang dihasilkan.

2. Kinerja dan Akurasi Model Prediksi

Hasil pengujian menunjukkan bahwa algoritma *Naive Bayes* memperoleh tingkat akurasi sebesar 79,23%, sedangkan algoritma *K-Nearest Neighbors* (K-NN) menghasilkan akurasi yang lebih tinggi, yaitu sebesar 92,07%. Selain itu, penerapan model hybrid melalui penggabungan hasil prediksi model *Naive Bayes* dan K-NN dengan metode *soft voting* terbukti mampu meningkatkan stabilitas serta performa prediksi secara keseluruhan dibandingkan dengan penggunaan model tunggal. Evaluasi kinerja model menggunakan *confusion matrix*, *classification report*, *learning curve*, *ROC curve*, dan nilai AUC juga menunjukkan bahwa sistem yang dikembangkan memiliki kemampuan klasifikasi yang baik dalam membedakan mahasiswa yang berpotensi lulus tepat waktu dan mahasiswa yang tidak lulus tepat waktu.

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, terdapat beberapa saran yang dapat dijadikan bahan pertimbangan untuk pengembangan penelitian selanjutnya, yaitu sebagai berikut:

1. Penelitian selanjutnya disarankan untuk menambahkan jumlah dan variasi fitur yang digunakan, seperti data aktivitas akademik, kehadiran, serta riwayat mata kuliah, agar model dapat menangkap pola yang lebih kompleks dan menghasilkan prediksi yang lebih akurat.
2. Penggunaan algoritma lain seperti *Random Forest*, *Support Vector Machine* (SVM), atau metode ensemble yang lebih kompleks dapat dipertimbangkan untuk dibandingkan dengan model yang telah digunakan dalam penelitian ini guna memperoleh performa yang lebih optimal.
3. Proses validasi model dapat diperluas dengan menggunakan teknik *K-Fold Cross Validation* yang lebih intensif agar hasil evaluasi menjadi lebih stabil dan mampu merepresentasikan performa model secara menyeluruh.
4. Sistem yang dikembangkan pada penelitian ini masih bersifat eksperimental, sehingga pada penelitian selanjutnya disarankan untuk mengimplementasikan model ke dalam bentuk aplikasi berbasis web atau dashboard visual agar dapat digunakan secara langsung oleh pihak akademik.
5. Penelitian lanjutan juga disarankan untuk memperhatikan permasalahan ketidakseimbangan kelas pada dataset, dengan menerapkan teknik penyeimbangan data seperti *oversampling* atau *undersampling* agar hasil prediksi menjadi lebih adil dan akurat.

DAFTAR PUSTAKA

- Agwil, W., Fransiska, H., & Hidayati, N. (2020). Analisis Ketepatan Waktu Lulus Mahasiswa Dengan Menggunakan Bagging Cart. *FIBONACCI: Jurnal Pendidikan Matematika Dan Matematika*, 6(2), 155. <https://doi.org/10.24853/fbc.6.2.155-166>
- Aldian Sakbani Nasution, J. (2021). *JUTSI (Jurnal Teknologi dan Sistem Informasi)*. 1(3), 219–226.
- Alfarizi, M. R. S., Al-farish, M. Z., Taufiqurrahman, M., Ardiansah, G., and Elgar, M., “Penggunaan Python sebagai bahasa pemrograman untuk machine learning dan deep learning,” 2023.
- Apridiansyah, Y., Veronika, N. D. M., & Putra, E. D. (2021). Prediksi Kelulusan Mahasiswa Fakultas Teknik Informatika Universitas Muhammadiyah Bengkulu Menggunakan Metode *Naive Bayes*. *JSai (Journal Scientific and Applied Informatics)*, 4(2), 236–247. <https://doi.org/10.36085/jsai.v4i2.1701>
- Ar-raniry, U. I. N., Tarbiyah, F., Keguruan, D. A. N., Studi, P., & Teknologi, P. (2023). *PENGUNAAN METODE NAIVES BAYES UNTUK SKRIPSI Diajukan Oleh*.
- Ardianti, M., Nurhayati, O. D., & Warsito, B. (2025). *Model prediksi kinerja siswa berdasarkan data log LMS menggunakan ensemble machine learning. Jurnal Sains dan Teknologi*, 12(3).
- Bendesa Subawa, I. G. (2019). Prediksi Kelulusan Mahasiswa Menggunakan Teorema Teorema Bayes. *Janapati*, 8(3), 227–236.
- Churniansyah, F., & Utomo, D. W. (2024). *Teknik Bagging pada ensemble learning untuk kategorisasi produk e-commerce. TEKNOSI*, 10(1), 92–99
- Dengen, C. N., Kusrini, K., & Luthfi, E. T. (2020). Implementasi Decision Tree
- Hakim, L. A. R., Rizal, A. A., & Ratnasari, D. (2019). Aplikasi Prediksi Kelulusan Mahasiswa Berbasis K-Nearest Neighbor (K-NN). *JTIM : Jurnal Teknologi Informasi Dan Multimedia*, 1(1), 30–36. <https://doi.org/10.35746/jtim.v1i1.11>
- Khasanah, N., Salim, A., Afni, N., Komarudin, R., & Maulana, Y. I. (2022). Prediksi Kelulusan Mahasiswa Dengan Metode *Naive Bayes*. *Technologia : Jurnal Ilmiah*, 13(3), 207. <https://doi.org/10.31602/tji.v13i3.7312>
- Komang Aditya Pratama, Gede Aditra Pradnyana, & I Ketut Resika Arthana. (2020). Pengembangan Sistem Cerdas Untuk Prediksi Daftar Kembali Mahasiswa Baru Dengan Metode *Naive Bayes* (Studi Kasus: Universitas Pendidikan Ganesha). *SINTECH (Science and Information Technology) Journal*, 3(1), 22–34. <https://doi.org/10.31598/sintechjournal.v3i1.523>

- Kuncheva, L. I. (2014). *Combining Pattern Classifiers: Methods and Algorithms*. Wiley.
- Kurniadi, D., Nuraeni, F., & Lestari, S. M. (2022). Implementasi Algoritma *Naive Bayes* Menggunakan Feature Forward Selection dan SMOTE Untuk Memprediksi Ketepatan Masa Studi Mahasiswa Sarjana. *Jurnal Sistem Cerdas*, 5(2), 63–82. <https://doi.org/10.37396/jsc.v5i2.215>
- Kusumadewi, S., Wahyuni, E. G., & Mulyani, S. (2021). *Sistem Cerdas dan Pendukung Keputusan* (I). UII Press Yogyakarta.
- M. I. C. Rachmatullah, “Ensemble Learning untuk Klasifikasi: Tinjauan Komprehensif Metode, Aplikasi, dan Perkembangan Terkini,” *IMPROVE*, vol. 17, no. 1, pp. 29–31, Apr. 2025.
- Mestika, J. C., Selan, M. O., and Qadafi, M. I., “Menjelajahi teknik-teknik supervised learning untuk pemodelan prediktif menggunakan Python,” *Buletin Ilmiah Ilmu Komputer dan Multimedia*, pp. 216–219, 2022.
- Montavon, G., Kauffmann, J., Samek, W., and Müller, K. R., “Explaining the predictions of unsupervised learning models,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Science and Business Media Deutschland GmbH, 2022, pp. 117–138, doi: 10.1007/978-3-031-04083-2_7.
- Murtopo, A. A. (2016). Prediksi Kelulusan Tepat Waktu Mahasiswa STMIK YMI Tegal Menggunakan Algoritma *Naive Bayes*. *CSRID (Computer Science Research and Its Development Journal)*, 7(3), 145. <https://doi.org/10.22303/csrid.7.3.2015.145-154>
- Neighbors, P. K., & Lightgbm, D. A. N. (2024). *Penggabungan K-Nearest Neighbors dan lightgbm untuk prediksi diabetes pada dataset pima indians: menggunakan pendekatan exploratory data analysis 1*. 9(3), 1133–1144.
- Nisa, Z. K., Kurniadi, D., & Julianto, I. T. (2025). Penerapan metode ensemble voting pada algoritma supervised learning dalam memprediksi penyakit gagal ginjal kronis. *Jurnal Aplikasi Informatika dan Multimedia*, 1(1), 1–8.
- Nurhalizah, R. S., Ardianto, R., and Purwono, “Analisis supervised dan unsupervised learning pada machine learning: Systematic literature review,” *Jurnal Ilmu Komputer dan Informatika (JIKI)*, vol. 4, no. 1, pp. 61–72, Jun. 2024, doi: 10.54082/jiki.168.
- Permana, A. A., Taufiq, R., Destriana, R., & Nur'aini, A. (2024). Implementasi Algoritma *Naive Bayes* Untuk Prediksi Kelulusan Mahasiswa. *TS: Jurnal Teknik*, 13(1), 65–70. <http://jurnal.umat.ac.id/index.php/jt/index>
- Purwati, N., & Dwi Januanti, A. (2021). Aplikasi Data Mining Dengan Algoritma *Naive Bayes* Untuk Memprediksi Tingkat Kelulusan Mahasiswa. *Jurnal Pepadun*, 2(1), 123–137. <https://doi.org/10.23960/pepadun.v2i1.38>
- Riadi, I., Umar, R., Anggara, R., & Ahmad Dahlan, U. (2024). *Prediksi Kelulusan*

Tepat Waktu Berdasarkan Riwayat Akademik Menggunakan Metode K-Nearest Neighbor. 11(2), 249–256.

- Rohmawan, E. P. (2018). Prediksi Kelulusan Mahasiswa Tepat Waktu Menggunakan Metode Decision Tree dan Artificial Neural Network. *Jurnal Ilmiah MATRIK Vol.20 No.1, April 2018:21-30, 20(1), 21–30.*
- Safitri, D., Hilabi, S. S., & Nurapriani, F. (2023). *Analisis penggunaan algoritma klasifikasi dalam prediksi kelulusan menggunakan orange data mining 1) 1,2,3). 8(1), 75–81.*
- Setiyani, L., Wahidin, M., Awaludin, D., & Purwani, S. (2020). Analisis Prediksi Kelulusan Mahasiswa Tepat Waktu Menggunakan Metode Data Mining Naive Bayes : Systematic Review. *Faktor Exacta, 13(1), 35.* <https://doi.org/10.30998/faktorexacta.v13i1.5548>
- Subariah, R., & Zein, A. (2023). Penerapan Metode Naive Bayes Untuk Prediksi Kelulusan Mahasiswa Tepat Waktu Di Prodi Sistem Informasi Universitas Pamulang. *Sainstech: Jurnal Penelitian Dan Pengkajian Sains Dan Teknologi, 33(2), 47–51.* <https://doi.org/10.37277/stch.v33i2.1585>
- Suganda, F., Nugroho, H. W., & Asmuni, I. (2021). Prediction of Graduation of Students of the Lampung School of Technology Nusantara using the K-Nearest Neighbor and Naive Bayes Algorithm. *Proceeding International Conference on Information Technology and Business, 0(0), 69–72.* <https://jurnal.darmajaya.ac.id/index.php/icitb/article/view/3045>
- Sulistiyawan, M. R., Studi, P., Informatika, T., & Teknik, F. (2021). *Pemodelan prediksi tingkat kelulusan mahasiswa dengan pendekatan algoritma Naive Bayes. 405–414.*
- Thaniket, R., Kusriani, & Luthf, E. T. (2019). Prediksi Kelulusan Mahasiswa Tepat Waktu Menggunakan Algoritma Support Vector Machine. *JURNAL FATEKSA: Jurnal Teknologi Dan Rekayasa, 13(2), 69–83.*
- Untuk Prediksi Kelulusan Mahasiswa Tepat Waktu. *Sisfotenika, 10(1), 1.* <https://doi.org/10.30700/jst.v10i1.484>
- Yusuf, B., Qalbi, M., Basrul, B., Dwitawati, I., Malahayati, M., & Ellyadi, M. (2020). Implementasi Algoritma Naive Bayes Dan Random Forest Dalam Memprediksi Prestasi Akademik Mahasiswa Universitas Islam Negeri Ar-Raniry Banda Aceh. *Cyberspace: Jurnal Pendidikan Teknologi Informasi, 4(1), 50.* <https://doi.org/10.22373/cj.v4i1.7247>
- KBBI Daring.* October 18, 2024. <https://kbbi.kemdikbud.go.id/>

LAMPIRAN

```
# =====  
# IMPORT LIBRARY  
# =====  
import pandas as pd  
import numpy as np  
  
from sklearn.model_selection import train_test_split,  
GridSearchCV  
from sklearn.preprocessing import StandardScaler  
from sklearn.impute import SimpleImputer  
  
from sklearn.naive_bayes import GaussianNB  
from sklearn.neighbors import KNeighborsClassifier  
  
from sklearn.pipeline import Pipeline  
from sklearn.ensemble import VotingClassifier  
  
from sklearn.metrics import accuracy_score,  
classification_report, confusion_matrix  
import joblib  
  
# =====  
# LOAD DATASET  
# =====  
df = pd.read_csv('/content/data_mahasiswa.csv', sep=';',  
decimal=',')  
print("Dataset berhasil dimuat\n")  
  
# =====  
# CLEANING DATA  
# =====  
# Ambil kolom IP  
ip_cols = [col for col in df.columns if 'ip_' in col.lower()]  
  
# Ganti koma jadi titik jika ada  
df[ip_cols] = df[ip_cols].apply(lambda x:  
x.astype(str).str.replace(',', '.'))  
df[ip_cols] = df[ip_cols].astype(float)
```

```

# =====
# LABEL ENCODING
# =====
df['target_binary'] = df['status_kelulusan'].map({
    'lulus_tepat_waktu': 1,
    'lulus_tidak_tepat_waktu': 0,
    'belum_lulus': 0
})

print("Distribusi Label:")
print(df['target_binary'].value_counts())
print("\n")

```

```

# =====
# FEATURE ENGINEERING
# =====
# Rata-rata IP
df['ip_mean'] = df[ip_cols].mean(axis=1)

# IP semester terakhir yang tersedia
df['ip_last'] = df[ip_cols].ffill(axis=1).iloc[:, -1]

ip_awal = df[ip_cols].iloc[:, 0]

# Tren IP (akhir - awal)
df['ip_trend'] = ((df['ip_last'] - ip_awal) / ip_awal) * 100

print(" Feature engineering selesai\n")

```

```

# =====
# FEATURE SELECTION
# =====
X = df.drop(columns=['status_kelulusan', 'target_binary',
    'tahun_masuk'], errors='ignore')
y = df['target_binary']

```

```

# =====
# SPLITTING DATA
# =====
X_train, X_test, y_train, y_test = train_test_split(
    X, y,

```

```

        test_size=0.2,
        random_state=42,
        stratify=y
    )
    print("Jumlah Data Training :", X_train.shape[0])
    print("Jumlah Data Testing  :", X_test.shape[0])
    print("\n")

```

```

# =====
# PREPROCESSING PIPELINE
# =====
preprocessor = Pipeline([
    ('imputer', SimpleImputer(strategy='median')),
    ('scaler', StandardScaler())
])

```

```

# =====
# MODEL NAIVE BAYES
# =====
nb_pipeline = Pipeline([
    ('preprocess', preprocessor),
    ('model', GaussianNB())
])
nb_pipeline.fit(X_train, y_train)

y_pred_nb = nb_pipeline.predict(X_test)

#print("===  HASIL NAIVE BAYES ===")
#print("Accuracy :", accuracy_score(y_test, y_pred_nb))
#print(classification_report(y_test, y_pred_nb))
#print(confusion_matrix(y_test, y_pred_nb))
#print("\n")

# =====
# MODEL KNN + GRID SEARCH
# =====
knn_pipeline = Pipeline([
    ('preprocess', preprocessor),
    ('model', KNeighborsClassifier())
])

```

```

param_grid = {
    'model__n_neighbors': [3, 5, 7, 9],
    'model__metric': ['euclidean', 'manhattan']
}

grid_knn = GridSearchCV(
    knn_pipeline,
    param_grid,
    cv=5,
    scoring='accuracy',
    n_jobs=-1
)

grid_knn.fit(X_train, y_train)
best_knn = grid_knn.best_estimator_
y_pred_knn = best_knn.predict(X_test)

#print("===  HASIL KNN ===")
#print("Best Parameter :", grid_knn.best_params_)
#print("Accuracy :", accuracy_score(y_test, y_pred_knn))
#print(classification_report(y_test, y_pred_knn))
#print(confusion_matrix(y_test, y_pred_knn))
#print("\n")

```

```

# =====
# ENSEMBLE (HYBRID MODEL)
# =====
hybrid_model = VotingClassifier(
    estimators=[
        ('nb', nb_pipeline),
        ('knn', best_knn)
    ],
    voting='soft'
)

hybrid_model.fit(X_train, y_train)
y_pred_hybrid = hybrid_model.predict(X_test)

#print("===  HASIL HYBRID MODEL ===")
#print("Accuracy :", accuracy_score(y_test, y_pred_hybrid))
#print(classification_report(y_test, y_pred_hybrid))
#print(confusion_matrix(y_test, y_pred_hybrid))
#print("\n")

```

```
# =====
# SIMPAN MODEL TERBAIK
# =====
joblib.dump(hybrid_model, 'model_kelulusan_hybrid.pkl')

print(" Model hybrid berhasil disimpan\n")
```

```
# =====
# INFERENCE (PREDIKSI DATA BARU)
# =====
data_baru = pd.DataFrame({
    'ip_1': [3.00],
    'ip_2': [3.00],
    'ip_3': [3.00],
    'ip_4': [3.00],
    'ip_5': [3.00],
    'ip_6': [3.00],
    'ip_7': [3.00],
    'ip_8': [3.00]
})

# Feature engineering saat inference
data_baru['ip_mean'] = data_baru[ip_cols].mean(axis=1)
data_baru['ip_last'] = data_baru['ip_8']
data_baru['ip_trend'] = ((data_baru['ip_last'] -
data_baru['ip_1']) / data_baru['ip_1']) * 100

# Prediksi
prediksi = hybrid_model.predict(data_baru)
probabilitas = hybrid_model.predict_proba(data_baru)

print("=== 🔍 HASIL INFERENCE ===")
print(data_baru)
print("\nPrediksi Kelulusan :", prediksi[0])
print("0 = Tidak Tepat Waktu / Belum Lulus")
print("1 = Lulus Tepat Waktu")
print("\nProbabilitas :", probabilitas)
```

tahun_masuk	ip_1	ip_2	ip_3	ip_4	ip_5	ip_6	ip_7	ip_8	status_kelulusan
2018	2,55	3,11	3,76	3,67	3,67	3,56	3,83	0,00	lulus_tepat_waktu
2018	2,73	3,11	3,52	3,38	3,25	3,57	3,89	0,00	lulus_tepat_waktu
2018	3,36	3,67	3,58	3,71	3,52	3,86	3,77	3,33	lulus_tepat_waktu
2018	3,27	2,48	3,17	3,45	3,49	3,81	3,35	4,00	lulus_tepat_waktu
2018	3,27	3,19	3,73	3,75	3,86	3,96	3,95	4,00	lulus_tepat_waktu
2018	3,36	3,57	3,75	3,83	3,70	3,92	3,93	4,00	lulus_tepat_waktu
2018	3,18	3,14	3,32	3,50	3,64	3,42	3,74	4,00	lulus_tepat_waktu
2018	3,00	3,05	3,59	3,54	3,77	3,96	3,64	3,33	lulus_tepat_waktu
2018	3,36	3,43	3,86	3,83	3,63	3,92	3,71	4,00	lulus_tepat_waktu
2018	3,09	3,14	3,55	3,29	3,74	3,68	3,77	3,33	lulus_tepat_waktu
2018	2,82	3,00	1,82	3,00	3,33	3,55	3,49	3,52	lulus_tidak_tepat_waktu
2018	2,91	1,93	1,00	1,08	1,00	1,33	2,25	1,67	lulus_tidak_tepat_waktu
2018	2,09	2,73	2,10	2,13	1,09	3,39	3,16	1,89	lulus_tidak_tepat_waktu
2018	3,00	2,57	2,88	2,90	3,55	2,44	3,00	2,80	lulus_tidak_tepat_waktu
2018	3,36	2,23	0,69	2,25	3,29	2,70	2,98	2,44	lulus_tidak_tepat_waktu
2018	2,55	0,83	3,58	2,46	2,65	3,53	3,30	3,32	lulus_tidak_tepat_waktu
2018	3,45	2,29	3,38	3,27	3,72	3,56	3,47	4,00	lulus_tidak_tepat_waktu
2018	2,64	2,00	3,06	2,77	3,28	3,35	3,52	1,62	lulus_tidak_tepat_waktu
2018	3,33	3,00	2,81	2,35	1,22	3,47	3,17	2,64	lulus_tidak_tepat_waktu
2018	2,73	1,85	3,14	3,23	3,46	3,32	2,63	2,22	lulus_tidak_tepat_waktu
2018	2,82	3,00	1,82	3,00	3,33	3,55	3,49	3,52	belum_lulus
2018	2,91	1,93	1,00	1,08	1,00	1,33	2,25	1,67	belum_lulus
2018	2,09	2,73	2,10	2,13	1,09	3,39	3,16	1,89	belum_lulus
2018	3,00	2,57	2,88	2,90	3,55	2,44	3,00	0,00	belum_lulus
2018	3,36	2,23	0,69	2,25	3,29	2,70	2,98	2,44	belum_lulus
2018	2,55	0,83	3,58	2,46	2,65	3,53	3,30	3,32	belum_lulus
2018	3,45	2,29	3,38	3,27	3,72	3,56	3,47	4,00	belum_lulus
2018	2,64	2,00	3,06	2,77	3,28	3,35	3,52	1,62	belum_lulus
2018	3,33	3,00	2,81	2,35	1,22	3,47	3,17	2,64	belum_lulus
2018	2,73	1,85	3,14	3,23	3,46	3,32	2,63	0,00	belum_lulus
2019	3,43	3,86	3,37	3,56	3,77	3,76	3,74	3,87	lulus_tepat_waktu
2019	3,18	3,48	2,98	3,47	3,43	3,44	3,20	3,78	lulus_tepat_waktu
2019	3,64	4,00	3,81	3,92	3,68	3,90	3,47	4,00	lulus_tepat_waktu
2019	3,55	3,50	3,56	3,72	3,69	3,96	3,64	4,00	lulus_tepat_waktu
2019	3,64	3,67	3,74	3,74	3,26	3,79	3,59	4,00	lulus_tepat_waktu
2019	3,73	3,79	3,71	3,25	3,65	3,71	3,50	4,00	lulus_tepat_waktu
2019	3,64	3,63	3,38	3,20	3,59	3,51	3,62	4,00	lulus_tepat_waktu
2019	3,73	3,79	3,60	3,78	3,96	3,83	3,61	4,00	lulus_tepat_waktu
2019	3,50	3,33	3,36	2,94	3,20	3,48	3,42	3,84	lulus_tepat_waktu
2019	3,45	3,55	3,58	3,57	3,57	3,54	2,89	3,93	lulus_tepat_waktu
2019	3,27	3,33	3,44	1,73	3,61	3,39	3,44	3,89	lulus_tidak_tepat_waktu
2019	3,45	3,33	3,41	3,64	3,01	2,21	2,96	3,33	lulus_tidak_tepat_waktu
2019	3,09	3,43	3,29	3,42	2,86	2,72	2,62	3,73	lulus_tidak_tepat_waktu
2019	2,80	3,20	2,85	2,95	3,46	3,42	3,26	2,89	lulus_tidak_tepat_waktu
2019	2,85	3,40	3,24	2,29	3,00	3,22	2,60	2,95	lulus_tidak_tepat_waktu
2019	2,65	3,20	3,51	3,32	3,55	3,63	3,46	2,20	lulus_tidak_tepat_waktu

2019	3,05	3,59	3,42	2,91	3,52	3,54	3,13	3,50	lulus_tidak_tepat_waktu
2019	2,85	3,40	3,42	3,30	3,45	3,30	2,90	3,67	lulus_tidak_tepat_waktu
2019	3,36	3,57	3,32	3,00	3,57	3,61	3,56	4,00	lulus_tidak_tepat_waktu
2019	2,95	2,90	3,16	3,25	3,05	3,06	3,37	3,43	lulus_tidak_tepat_waktu
2019	2,45	0,33	0,17	0,00	3,06	2,68	0,00	2,44	belum_lulus
2019	3,00	3,33	2,58	3,52	3,32	3,14	2,86	3,29	belum_lulus
2019	3,18	3,71	3,04	3,14	3,46	2,93	3,00	3,00	belum_lulus
2019	3,00	3,71	3,43	3,64	3,06	2,52	3,03	3,67	belum_lulus
2019	3,09	3,05	3,20	2,30	3,49	2,67	3,11	2,95	belum_lulus
2019	2,91	2,32	2,91	3,35	2,91	1,75	2,90	1,86	belum_lulus
2019	2,64	2,80	3,08	2,44	3,29	0,73	0,00	0,92	belum_lulus
2019	2,91	3,11	3,12	3,26	3,32	2,97	2,50	2,61	belum_lulus
2019	2,91	3,53	3,51	3,81	3,15	2,91	2,65	3,67	belum_lulus
2019	3,18	3,29	3,17	3,16	3,44	2,44	2,67	3,20	belum_lulus
2020	3,83	3,65	3,88	3,83	3,96	3,82	4,00	0,00	lulus_tepat_waktu
2020	3,68	3,81	3,89	3,90	3,70	3,35	3,88	0,00	lulus_tepat_waktu
2020	3,82	3,67	3,74	3,57	3,52	3,88	3,92	0,00	lulus_tepat_waktu
2020	3,38	3,26	3,39	3,47	3,50	3,45	3,88	0,00	lulus_tepat_waktu
2020	3,82	3,87	3,78	3,89	3,74	3,94	3,67	0,00	lulus_tepat_waktu
2020	2,93	3,49	3,06	3,11	3,45	3,61	3,67	0,00	lulus_tepat_waktu
2020	3,58	3,54	3,82	3,70	3,47	3,75	3,92	0,00	lulus_tepat_waktu
2020	3,36	3,24	1,76	3,78	2,75	2,39	3,33	0,00	lulus_tepat_waktu
2020	3,58	3,20	2,89	2,91	3,47	3,56	3,94	0,00	lulus_tepat_waktu
2020	3,58	3,71	3,32	3,32	3,35	3,20	3,47	0,00	lulus_tepat_waktu
2020	3,70	3,58	3,64	3,48	2,81	3,23	3,42	0,00	belum_lulus
2020	3,52	3,08	2,52	2,19	2,44	3,37	3,05	3,06	belum_lulus
2020	3,03	1,51	2,45	2,28	2,88	2,70	2,88	3,28	belum_lulus
2020	3,36	2,91	3,03	2,64	2,74	2,93	1,54	2,28	belum_lulus
2020	3,33	2,91	3,41	3,56	2,50	3,17	2,98	3,33	belum_lulus
2020	3,61	2,70	2,96	2,43	3,04	3,44	3,21	3,42	belum_lulus
2020	3,30	3,53	3,50	2,44	2,48	1,30	2,89	3,56	belum_lulus
2020	2,94	2,67	3,49	3,12	3,33	3,50	3,57	3,74	belum_lulus
2020	3,64	2,99	2,65	2,91	1,57	3,40	3,40	3,27	belum_lulus
2020	3,55	3,25	3,55	3,44	3,30	3,65	3,48	1,60	belum_lulus
2021	3,29	3,41	3,36	2,56	3,06	3,41	0,00	0,00	belum_lulus
2021	1,51	0,67	1,94	0,14	0,27	0,50	0,00	0,00	belum_lulus
2021	3,70	3,27	3,33	1,12	2,39	2,32	0,00	0,00	belum_lulus
2021	3,39	2,71	3,03	2,82	2,60	3,14	0,00	0,00	belum_lulus
2021	3,83	3,47	3,52	3,60	3,52	3,68	0,00	0,00	belum_lulus
2021	3,67	3,33	2,87	1,88	3,72	3,50	0,00	0,00	belum_lulus
2021	3,73	3,28	3,47	3,48	3,48	3,63	0,00	0,00	belum_lulus
2021	3,55	3,54	3,00	3,58	3,06	3,52	0,00	0,00	belum_lulus
2021	3,71	3,24	3,29	3,37	3,15	3,90	0,00	0,00	belum_lulus