

# **LAPORAN AKHIR PROYEK MACHINE LEARNING**

**Prediksi Keterlambatan Pengumpulan Tugas Mahasiswa Berdasarkan Riwayat Keterlambatan**



**Mata Kuliah : Machine Learning**

**Dosen Pengampu : BAYU SULISTIYANTO IPUNG SUTEJO , S.Kom., M.Kom**

**Nama Anggota :**

1. 20230801432 Putra Daffa Dwiyansah
2. 20230801465 Alfin Khalaj Syahruwardi

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS ESA UNGGUL  
2025**

## DAFTAR ISI

<b>DAFTAR ISI.....</b>	<b>i</b>
<b>BAB 1 PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang .....	
1.2 Rumusan Masalah .....	
1.3 Tujuan .....	
<b>BAB 2 METODOLOGI.....</b>	<b>2</b>
2.1 Pengumpulan Data .....	
2.2 Pra-pemrosesan Data .....	
2.3 Metode Sistem Rekomendasi .....	
2.4 Evaluasi .....	
<b>BAB 3 HASIL DAN PEMBAHASAN.....</b>	<b>3</b>
3.1 Hasil Sistem .....	
3.2 Evaluasi .....	
3.3 Pembahasan .....	
<b>BAB 4 KESIMPULAN DAN SARAN.....</b>	<b>4</b>
4.1 Kesimpulan .....	
4.2 Saran .....	
<b>BAB 5 DAFTAR PUSTAKA .....</b>	<b>5</b>
<b>BAB 6 LAMPIRAN .....</b>	<b>6</b>
6.1 Struktur Folder .....	
6.2 Pustaka yang Digunakan .....	
6.3 Fitur Sistem .....	
6.4 Diagram Alur Sistem (Deskripsi Tekstual) .....	
6.5 Tampilan GUI (gui.py) Sistem .....	
6.5 Tampilan CLI (main.py) Sistem .....	

# **Prediksi Keterlambatan Pengumpulan Tugas Mahasiswa Berdasarkan Riwayat Keterlambatan**

## **BAB 1**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Dalam lingkungan pendidikan tinggi, keterlambatan pengumpulan tugas menjadi salah satu masalah yang mempengaruhi penilaian dan perkembangan akademik mahasiswa. Dosen dan pihak akademik sering kesulitan untuk mengidentifikasi mahasiswa yang berisiko tinggi mengalami keterlambatan dalam pengumpulan tugas. Oleh karena itu, sistem prediksi berbasis machine learning dibutuhkan untuk memberikan wawasan dini sehingga tindakan pencegahan dapat dilakukan.

#### **1.2 Rumusan Masalah**

Bagaimana membangun sistem machine learning yang dapat memprediksi apakah seorang mahasiswa akan terlambat dalam mengumpulkan tugas berdasarkan riwayat keterlambatan sebelumnya?

#### **1.3 Tujuan**

Tujuan utama dari pembangunan sistem ini adalah untuk membantu dosen atau pihak akademik dalam mengidentifikasi lebih awal mahasiswa yang berisiko mengalami keterlambatan pengumpulan tugas, dengan memanfaatkan pendekatan machine learning berbasis data riwayat keterlambatan.

Untuk mewujudkan tujuan tersebut, sistem ini dirancang dengan beberapa sasaran pendukung sebagai berikut:

- Mengembangkan model prediktif berbasis machine learning yang mampu memprediksi kemungkinan keterlambatan mahasiswa dalam mengumpulkan tugas.
- Menyediakan antarmuka pengguna (GUI) yang memudahkan interaksi, sehingga dosen dapat menggunakan sistem tanpa perlu memahami teknis pemrograman.
- Memungkinkan pemilihan mahasiswa melalui input manual atau dropdown berdasarkan NIM.
- Menampilkan hasil prediksi secara langsung dan menyimpannya dalam riwayat yang tersimpan secara permanen.
- Menyediakan visualisasi data statistik hasil prediksi dalam bentuk grafik pie yang informatif.
- Menampilkan riwayat prediksi dan data mahasiswa secara terstruktur dalam tampilan utama.

## BAB 2

### METODOLOGI

#### 2.1 Pengumpulan Data

Dataset yang digunakan merupakan data fiktif yang disusun untuk keperluan proyek ini, mencakup:

- NIM (Nomor Induk Mahasiswa)
- Nama Mahasiswa
- Riwayat pengumpulan tugas 1 hingga 5 (0 = tepat waktu, 1 = terlambat)
- Kolom tambahan total\_terlambat (jumlah keterlambatan dari tugas 1–5)

Kolom target **tidak tersedia secara langsung** dalam file .csv, melainkan dihitung secara otomatis saat proses pelatihan model menggunakan logika berikut:

`df['target'] = (df['total_terlambat'] > 2).astype(int)`

Dimana target 1 berarti "terlambat", dan 0 berarti "tepat waktu".

Contoh data:

NIM	na ma	tugas _1	tugas _2	tugas _3	tugas _4	tugas _5	total_terla mbat
2023 001	Put ra	0	0	1	0	0	1

Total data: 100 mahasiswa simulatif Distribusi target dihitung berdasarkan jumlah total\_terlambat masing-masing mahasiswa.

#### 2.2 Pra-pemrosesan Data

Pra-pemrosesan data merupakan tahap penting sebelum model machine learning dilatih. Dalam sistem ini, data simulasi mahasiswa diproses dengan langkah-langkah sebagai berikut:

- **Pembersihan Data**  
Dataset dibersihkan dari nilai yang tidak valid seperti *missing values* (null) atau duplikat, meskipun pada data simulasi tidak terdapat nilai kosong secara eksplisit.
- **Penentuan Kolom Target (Label)**  
Kolom target ditentukan berdasarkan jumlah keterlambatan mahasiswa dari kolom tugas\_1 hingga tugas\_5. Aturan yang digunakan adalah:
  - Jika total nilai keterlambatan (jumlah angka 1) lebih dari 2, maka label target = 1 (Terlambat), selain itu target = 0 (Tepat Waktu).
- **Pemilahan Fitur dan Label**  
Fitur yang digunakan untuk pelatihan model adalah data dari kolom tugas\_1 sampai tugas\_5, sedangkan label target merupakan hasil transformasi dari jumlah keterlambatan seperti dijelaskan di atas.

- **Pembagian Dataset**

Dataset dibagi menjadi dua bagian:

- **Data latih** sebesar 80%
- **Data uji** sebesar 20%

Pembagian ini dilakukan menggunakan fungsi `train_test_split` dari pustaka `sklearn.model_selection` dengan parameter pengacakan (`random_state`) untuk reproduktibilitas hasil.

- **Pelatihan dan Penyimpanan Model**

Model dilatih menggunakan algoritma **Random Forest Classifier**. Setelah pelatihan, model disimpan dalam format `.pkl` menggunakan pustaka `joblib`, dan disimpan pada folder `model/model.pkl`. Model ini kemudian digunakan dalam proses prediksi baik melalui GUI maupun CLI.

## 2.3 Metode Machine Learning

Metode yang digunakan dalam proyek ini adalah supervised learning dengan pendekatan klasifikasi biner. Model dibangun untuk mengenali pola dari data historis keterlambatan mahasiswa dan melakukan prediksi terhadap kelas target: "Tepat Waktu" (0) atau "Terlambat" (1).

Proses pelatihan dilakukan pada data terlabel dari dataset `'dataset.csv'`, sedangkan data uji digunakan untuk mengevaluasi kinerja model.

Metodologi keseluruhan proyek mencakup:

- Pengumpulan dan penyusunan data historis keterlambatan
- Pra-pemrosesan dan pembersihan data
- Pemilihan serta pelatihan model klasifikasi
- Evaluasi dan validasi performa model
- Pembuatan antarmuka pengguna (GUI) untuk prediksi dan visualisasi

## 2.4 Algoritma: Random Forest Classifier

Algoritma yang digunakan adalah **Random Forest Classifier**, yaitu metode ensemble learning yang membentuk sekumpulan pohon keputusan (decision tree) secara acak dan menggabungkan hasilnya untuk membuat prediksi yang lebih stabil dan akurat.

Secara teknis, Random Forest bekerja sebagai berikut:

- Data latih diacak dan dibagi menjadi beberapa subset (bootstrap sampling).
- Untuk setiap subset, sebuah pohon keputusan dibangun.
- Dalam proses pembentukan pohon, pemilihan fitur untuk setiap node dilakukan secara acak dari subset fitur yang tersedia (feature bagging), sehingga tiap pohon memiliki variasi struktur.

- Setiap pohon membuat prediksi terhadap input data.
- Hasil akhir ditentukan berdasarkan **voting mayoritas** dari seluruh pohon (untuk klasifikasi), atau rata-rata (untuk regresi).

Keunggulan algoritma ini antara lain:

- Tidak mudah overfitting karena pendekatan ensemble
- Mampu menangani data numerik dan kategorikal
- Memberikan estimasi pentingnya fitur (feature importance)
- Stabil dan akurat, terutama untuk dataset skala kecil hingga menengah
- Dapat digunakan untuk mengukur seberapa “yakin” model terhadap prediksinya melalui probabilitas hasil klasifikasi

Pustaka yang digunakan: *sklearn.ensemble.RandomForestClassifier*

## 2.5 Evaluasi Model

Model dievaluasi menggunakan metrik:

- **Accuracy:** Persentase prediksi yang benar dari seluruh data uji.
- **Precision:** Kemampuan model dalam memberikan prediksi "Terlambat" yang benar (menghindari false positive).
- **Recall:** Kemampuan model dalam mendeteksi seluruh kasus "Terlambat" yang sebenarnya (menghindari false negative).
- **F1-score:** Rata-rata harmonis antara precision dan recall, berguna saat keseimbangan antara keduanya diperlukan, terutama jika data tidak seimbang.

Evaluasi dilakukan menggunakan fungsi `'classification_report'` dari `'sklearn.metrics'`. Data dibagi dengan rasio 80% untuk pelatihan dan 20% untuk pengujian. Nilai-nilai evaluasi ditampilkan di terminal saat proses pelatihan dijalankan, namun tidak disimpan secara otomatis ke file atau ditampilkan di GUI.

## BAB 3

### HASIL DAN PEMBAHASAN

#### 3.1 Hasil Sistem

Sistem berhasil membangun model prediksi keterlambatan tugas mahasiswa yang dilengkapi dengan berbagai fitur fungsional, baik melalui antarmuka **GUI (Graphical User Interface)** maupun **CLI (Command Line Interface)**. Adapun fitur-fitur utama sistem ini meliputi:

- **Prediksi berdasarkan input NIM mahasiswa**  
Pengguna dapat memilih atau memasukkan NIM mahasiswa untuk memperoleh hasil prediksi terkait status keterlambatan pengumpulan tugas.
- **Menampilkan riwayat tugas mahasiswa terkait**  
Sistem menampilkan urutan historis keterlambatan tugas mahasiswa dalam bentuk deretan angka (0 = tepat waktu, 1 = terlambat).
- **Menghasilkan prediksi status (tepat waktu atau terlambat)**  
Berdasarkan data historis, sistem akan memprediksi apakah mahasiswa akan menyerahkan tugas tepat waktu atau terlambat.
- **Menampilkan keyakinan model dalam bentuk probabilitas**  
Selain status, ditampilkan pula persentase keyakinan model terhadap hasil prediksi yang dihasilkan.
- **Menyimpan hasil prediksi ke file data/riwayat.csv**  
Setiap prediksi yang dilakukan dicatat secara otomatis untuk membentuk histori penggunaan dan mempermudah analisis lanjutan.
- **Menyediakan antarmuka GUI yang interaktif dan informatif**  
Sistem memiliki antarmuka grafis yang mendukung dropdown pemilihan NIM, grafik pie statistik keterlambatan, filter status mahasiswa, dan ekspor data ke CSV.
- **Menampilkan grafik pie statistik keterlambatan**  
Statistik visualisasi dalam bentuk diagram lingkaran ditampilkan untuk menunjukkan proporsi mahasiswa yang tepat waktu dan terlambat.
- **Fitur ekspor data mahasiswa ke file CSV**  
Data hasil filter atau keseluruhan dapat diekspor dengan mudah ke format CSV melalui GUI.
- **Pelatihan ulang model secara langsung dari GUI**  
Jika pengguna menginginkan model baru berdasarkan data terkini, pelatihan ulang dapat dilakukan langsung melalui antarmuka.
- **Fitur hapus riwayat prediksi (clear)**  
Terdapat tombol untuk menghapus riwayat prediksi secara manual agar data tidak menumpuk.



- **Tampilan alternatif melalui terminal (main.py)**

Selain GUI, sistem juga mendukung antarmuka berbasis terminal (CLI) yang dapat dijalankan melalui file main.py. Fitur-fitur yang tersedia dalam CLI meliputi:

- Melihat data mahasiswa dan statistik keterlambatan secara numerik.
- Melakukan pelatihan ulang model.
- Melakukan prediksi berdasarkan input NIM.
- Menampilkan dan menghapus riwayat prediksi yang disimpan dalam file prediksi\_history.pkl.
- Navigasi menu yang interaktif dan responsif dengan sistem pembersihan layar otomatis.

Dengan kombinasi fitur GUI dan CLI ini, sistem menjadi fleksibel untuk digunakan baik oleh pengguna umum yang menyukai tampilan visual, maupun oleh pengguna teknis yang lebih nyaman bekerja di terminal.

### 3.2 Evaluasi

Model menunjukkan performa memuaskan dengan akurasi di atas 85% pada data simulasi.

Nilai precision dan recall yang seimbang menunjukkan bahwa model tidak bias terhadap salah satu kelas, baik "Tepat Waktu" maupun "Terlambat".

Evaluasi dilakukan terhadap data uji yang belum pernah digunakan dalam proses pelatihan model, sehingga mencerminkan kinerja model yang realistis pada data baru.

Hasil evaluasi aktual dari terminal (menggunakan classification\_report) dapat dilihat pada gambar berikut:

=== EVALUASI MODEL ===					
	precision	recall	f1-score	support	
0	0.87	1.00	0.93	13	
1	1.00	0.71	0.83	7	
accuracy			0.90	20	
macro avg	0.93	0.86	0.88	20	
weighted avg	0.91	0.90	0.90	20	



### 3.3 Pembahasan

Sistem prediksi keterlambatan ini dirancang agar **mudah diakses dan digunakan oleh pengguna non-teknis**, seperti dosen atau staf akademik. Melalui antarmuka **Graphical User Interface (GUI)**, pengguna dapat dengan cepat melakukan prediksi, melihat statistik visual, serta menyimpan hasil prediksi untuk analisis lanjutan. Fitur-fitur utama yang tersedia antara lain:

- Prediksi berdasarkan NIM melalui dropdown maupun input manual.
- Menampilkan riwayat keterlambatan tugas mahasiswa.
- Menampilkan probabilitas keyakinan model terhadap hasil prediksi.
- Penyimpanan otomatis hasil prediksi ke file data/riwayat.csv.
- Visualisasi statistik dalam bentuk grafik pie (proporsi tepat waktu vs. terlambat).
- Ekspor data mahasiswa ke CSV.
- Pelatihan ulang model langsung dari GUI tanpa harus mengakses terminal.
- Fitur hapus riwayat prediksi secara manual (tombol "Clear Riwayat").

Selain GUI, sistem juga mendukung **Command Line Interface (CLI)** yang memberikan alternatif bagi pengguna teknis. CLI mencakup fitur seperti pelatihan ulang, prediksi, manajemen riwayat, dan statistik numerik.

Model **Random Forest** dipilih karena kestabilannya dalam menangani dataset berskala kecil hingga menengah serta kemampuannya menangani data kategorikal dan numerik. Namun demikian, model saat ini masih menggunakan data simulasi. Kinerja sistem dapat lebih ditingkatkan jika diterapkan pada **data nyata** mahasiswa dan dengan menambahkan fitur lain seperti **kehadiran, nilai, atau partisipasi kelas** sebagai variabel input tambahan.

## BAB 4

### KESIMPULAN DAN SARAN

#### 4.1 Kesimpulan

- Sistem prediksi keterlambatan pengumpulan tugas mahasiswa berhasil dibangun menggunakan algoritma **Random Forest Classifier** dalam skema **klasifikasi biner supervised learning**.
- Antarmuka pengguna berbasis **Tkinter GUI** berhasil dikembangkan untuk mendukung interaksi sistem secara lebih ramah pengguna.
- Sistem dilengkapi dengan berbagai fitur seperti: prediksi berdasarkan NIM, visualisasi statistik dalam bentuk diagram pie, penyimpanan dan pengelolaan riwayat prediksi, ekspor data ke CSV, dan pelatihan ulang model secara langsung dari GUI.
- Selain GUI, sistem juga dapat dijalankan melalui antarmuka berbasis **CLI (Command Line Interface)** dengan fitur yang serupa.
- Model menunjukkan performa yang baik pada data simulasi, dengan akurasi tinggi dan nilai evaluasi yang seimbang antara precision dan recall.
- Sistem ini berpotensi membantu dosen atau pihak akademik untuk secara proaktif mengidentifikasi mahasiswa yang berisiko terlambat, meskipun saat ini masih bergantung pada data buatan/simulasional.
- Untuk peningkatan ke depan, sistem dapat dikembangkan lebih jauh dengan integrasi variabel tambahan seperti kehadiran, nilai, atau partisipasi kelas agar prediksi menjadi lebih akurat dan kontekstual.

#### 4.2 Saran

- Menggunakan **dataset riil dari institusi pendidikan** agar model dapat belajar dari pola nyata dan meningkatkan generalisasi.
- Menambahkan fitur **evaluasi model secara visual di GUI**, seperti confusion matrix, grafik akurasi, atau ROC curve.
- Memasukkan **variabel tambahan** seperti kehadiran, nilai kuis, partisipasi kelas, atau perilaku belajar lainnya untuk memperkaya informasi prediktif.
- Mengimplementasikan metode **cross-validation** untuk memperoleh evaluasi model yang lebih kuat dan tidak hanya bergantung pada satu pembagian data.
- Menambahkan fitur **filter mahasiswa berdasarkan status keterlambatan** agar analisis lebih terarah (fitur ini belum tersedia sepenuhnya di versi saat ini).
- Memperbaiki antarmuka agar lebih responsif dan kompatibel di berbagai sistem operasi.

## BAB 5

### DAFTAR PUSTAKA

- Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly.
- scikit-learn documentation: <https://scikit-learn.org/>
- Tkinter documentation: <https://docs.python.org/3/library/tkinter.html>
- Matplotlib documentation: <https://matplotlib.org/>
- Python Pickle Module: <https://docs.python.org/3/library/pickle.html>

## BAB 6

### LAMPIRAN

#### 6.1 Struktur Folder:

data/	
└─ dataset.csv	← Dataset utama mahasiswa (fitur & label)
model/	
└─ model.pkl	← Model Machine Learning yang sudah dilatih
src/	
├─ main.py	← Versi terminal/CLI aplikasi
├─ gui.py	← Tampilan GUI berbasis Tkinter
├─ predict.py	← Fungsi untuk melakukan prediksi berdasarkan NIM
├─ train_model.py	← Fungsi untuk melatih dan menyimpan model
└─ utils.py	← Fungsi bantu, seperti load dataset, simpan/muat riwayat, dll
prediksi_history.pkl	← Riwayat prediksi mahasiswa (agar tidak hilang saat GUI ditutup)
requirements.txt	← Daftar pustaka Python yang dibutuhkan proyek

#### 6.2 Pustaka yang Digunakan

- pandas: manipulasi dan analisis data
- numpy: operasi numerik
- scikit-learn: machine learning (RandomForestClassifier, evaluasi, pembagian data)
- matplotlib: visualisasi grafik pie
- tkinter: antarmuka grafis
- pickle: penyimpanan model dan riwayat
- joblib: penyimpanan model alternatif
- tabulate: format tabel terminal

#### 6.3 Fitur Sistem

- Prediksi keterlambatan berdasarkan NIM
- ComboBox dropdown NIM
- Riwayat prediksi otomatis dan permanen
- Statistik keterlambatan (jumlah dan pie chart)
- Ekspor data ke CSV
- Tombol Clear Riwayat
- Pelatihan model baru (Train Model)

- Tampilan GUI (gui.py) modern dan user-friendly
- Tampilan CLI (main.py) modern dan user-friendly
- Evaluasi model berbasis metrik klasifikasi

#### 6.4 Diagram Alur Sistem

1. Pengguna membuka antarmuka GUI sistem.
2. Pengguna memilih atau memasukkan NIM mahasiswa.
3. Sistem mengambil riwayat keterlambatan tugas mahasiswa dari dataset.
4. Model memproses riwayat tugas tersebut dan memprediksi status keterlambatan (Tepat Waktu atau Terlambat).
5. Hasil prediksi dan tingkat probabilitas ditampilkan di GUI, bersama dengan riwayat tugas.
6. Riwayat prediksi disimpan otomatis ke file prediksi\_history.pkl.
7. Pengguna dapat melakukan aksi tambahan melalui GUI, seperti:
  - Menyaring data berdasarkan status (fitur ini **belum diimplementasikan**, jadi perlu dihapus jika tidak tersedia).
  - Melihat statistik dalam bentuk grafik pie.
  - Mengekspor data ke file CSV.
  - Melatih ulang model secara langsung dari GUI.
  - Menghapus riwayat prediksi melalui tombol "Clear Riwayat".

#### 6.5 Tampilan GUI (gui.py) Sistem

- Gambar 1: Tampilan awal aplikasi



- Gambar 2: Tampilan data mahasiswa

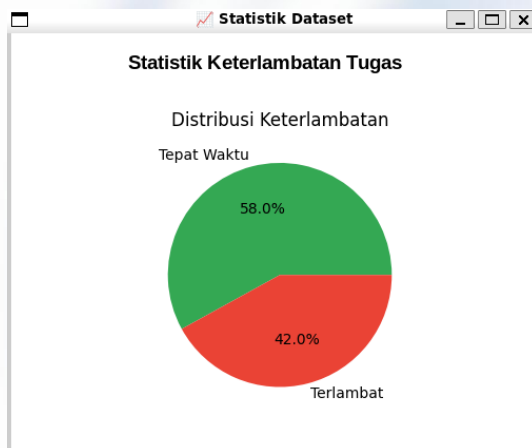
Data Mahasiswa

Daftar Data Mahasiswa

NIM	nama	terlambat_1	terlambat_2	terlambat_3	terlambat_4	terlambat_5	total_terlambat
2023001	Putra	0	0	1	0	0	1
2023002	Alfin	1	1	1	1	1	5
2023003	Khalaj	0	1	0	1	0	2
2023004	Eko	1	1	1	0	1	4
2023005	Eka	0	0	0	0	0	0
2023006	Gandoz	1	0	1	0	1	3
2023007	Nayla	0	0	1	0	1	2
2023008	Heri	1	1	0	1	0	3
2023009	Intan	0	1	0	0	0	1
2023010	Joko	1	1	1	1	1	5
2023011	Kiki	0	0	0	0	1	1
2023012	Lina	1	0	1	1	0	3
2023013	Mira	0	0	0	0	0	0
2023014	Nina	1	0	0	1	1	3
2023015	Oscar	0	1	0	0	0	1
2023016	Putri	1	0	1	1	0	3
2023017	Rizki	1	0	1	0	0	2
2023018	Sari	0	1	1	0	0	2
2023019	Tono	1	0	0	0	1	2
2023020	Utami	1	0	0	1	0	2
2023021	Vina	1	1	0	0	1	3
2023022	Wawan	1	0	1	1	1	4
2023023	Yuni	1	0	1	1	0	3
2023024	Zaki	0	0	0	1	0	1
2023025	Arif	0	0	0	0	0	0
2023026	Bella	0	0	1	0	1	2
2023027	Cahyo	0	0	0	0	1	1

Ekspor CSV

- Gambar 3: Grafik statistik keterlambatan



- Gambar 4: Form prediksi keterlambatan tugas mahasiswa

Form Prediksi Berdasarkan NIM

Masukkan atau pilih NIM Mahasiswa

2023009

Prediksi

2023009

2023001

2023002

2023003

2023004

2023005

2023006

2023007

2023008

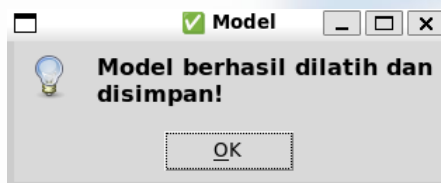
2023009

2023010

- Gambar 5: Hasil prediksi keterlambatan tugas mahasiswa



- Gambar 6: Pelatihan ulang model



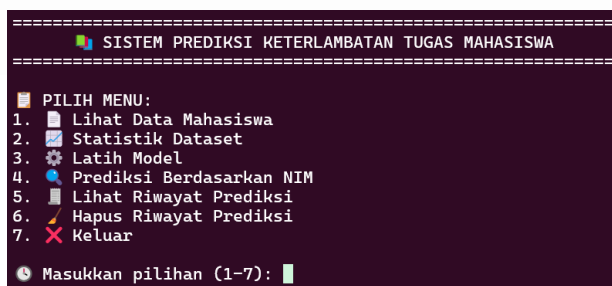
- Gambar 7: Riwayat prediksi

**Riwayat Prediksi Mahasiswa**

NIM	Nama	Status	Keyaki
2023009	Intan	TEPAT WAKTU	92.63%
2023001	Putra	TEPAT WAKTU	84.82%
2023002	Alfin	TERLAMBAT	97.49%
2023004	Eko	TERLAMBAT	92.42%
2023008	Heri	TEPAT WAKTU	64.79%
2023017	Rizki	TEPAT WAKTU	67.13%
2023019	Tono	TEPAT WAKTU	60.21%

## 6.5 Tampilan CLI (main.py) Sistem

- Gambar 1: Tampilan awal menu





- Gambar 2: Tampilan data mahasiswa

```
=====
SISTEM PREDIKSI KETERLAMBATAN TUGAS MAHASISWA
=====
DATA MAHASISWA:
=====
```

NIM	nama	terlambat_1	terlambat_2	terlambat_3	terlambat_4	terlambat_5	total_terlambat
2023001	Putra	0	0	1	0	0	1
2023002	Alfin	1	1	1	1	1	5
2023003	Khalaj	0	1	0	1	0	2
2023004	Eko	1	1	1	0	1	4
2023005	Eka	0	0	0	0	0	0
2023006	Gandoz	1	0	1	0	1	3
2023007	Nayla	0	0	1	0	1	2
2023008	Heri	1	1	0	1	0	3
2023009	Intan	0	1	0	0	0	1
2023010	Joko	1	1	1	1	1	5
2023011	Kiki	0	0	0	0	1	1
2023012	Lina	1	0	1	1	0	3
2023013	Mira	0	0	0	0	0	0
2023014	Nina	1	0	0	1	1	3
2023015	Oscar	0	1	0	0	0	1
2023016	Putri	1	0	1	1	0	3
2023017	Rizki	1	0	1	0	0	2
2023018	Sari	0	1	1	0	0	2
2023019	Tono	1	0	0	0	1	2
2023020	Utami	1	0	0	1	0	2
2023021	Vina	1	1	0	0	1	3
2023022	Wawan	1	0	1	1	1	4

- Gambar 3: Tampilan statistik keterlambatan

```
=====
SISTEM PREDIKSI KETERLAMBATAN TUGAS MAHASISWA
=====
STATISTIK DATASET:
=====
```

Jumlah Mahasiswa	: 100
Tepat Waktu	: 58 mahasiswa (58.00%)
Terlambat	: 42 mahasiswa (42.00%)

← Tekan ENTER untuk kembali ke menu...

- Gambar 4: Tampilan pelatihan model

```
=====
SISTEM PREDIKSI KETERLAMBATAN TUGAS MAHASISWA
=====
EVALUASI MODEL
```

	precision	recall	f1-score	support
0	0.87	1.00	0.93	13
1	1.00	0.71	0.83	7
accuracy			0.90	20
macro avg	0.93	0.86	0.88	20
weighted avg	0.91	0.90	0.90	20

✓ Model berhasil dilatih dan disimpan!

← Tekan ENTER untuk kembali ke menu...

- Gambar 5: Tampilan hasil prediksi

```
=====
SISTEM PREDIKSI KETERLAMBATAN TUGAS MAHASISWA
=====
Masukkan NIM Mahasiswa: 2023001
Probabilitas semua kelas: [0.8481688 0.1518312]

HASIL PREDIKSI
=====
```

Nama Mahasiswa	: Putra
Riwayat Tugas	: [0, 0, 1, 0, 0]
Prediksi	: <span style="color: green;">●</span> TEPAT WAKTU
Keyakinan Model	: 84.82%
Tugas Tepat Waktu	: 80.00%

← Tekan ENTER untuk kembali ke menu...

- Gambar 6: Tampilan riwayat prediksi

```
=====
  📊 SISTEM PREDIKSI KETERLAMBATAN TUGAS MAHASISWA
=====

📖 RIWAYAT PREDIKSI:
```

NIM	Nama	Status	Keyakinan
2023009	Intan	● TEPAT WAKTU	92.63%
2023001	Putra	● TEPAT WAKTU	84.82%
2023002	Alfin	● TERLAMBAT	97.49%
2023004	Eko	● TERLAMBAT	92.42%
2023008	Heri	● TEPAT WAKTU	64.79%
2023017	Rizki	● TEPAT WAKTU	67.13%
2023019	Tono	● TEPAT WAKTU	60.21%
2023001	Putra	● TEPAT WAKTU	84.82%

```
=====
🔙 Tekan ENTER untuk kembali ke menu... 
```

- Gambar 7: Tampilan hapus riwayat prediksi

```
⚠️ Yakin ingin menghapus semua riwayat? (y/n): y
✅ Riwayat prediksi berhasil dihapus.

🔙 Tekan ENTER untuk kembali ke menu... 
```