

LAPORAN PROYEK AKHIR MATA KULIAH

INFORMASI PROYEK

- **Judul Proyek:** Klasifikasi Pendonor Darah Menggunakan Dataset Blood Transfusion Service Center
- **Nama Mahasiswa:** Muhammad Daffa Samudra
- **NIM:** 233307054
- **Program Studi:** Teknologi Informasi
- **Mata Kuliah:** Data Science
- **Dosen Pengampu:** Gus Nanang Syaifuddiin
- **Tahun Akademik:** 2025/Semester 5
- **Link GitHub Repository:** <https://github.com/daffasamudra/Blood-Transfusion-Model.git>
- **Link Video Pembahasan:** https://drive.google.com/drive/folders/1PEU-K95x_u5mSGGtHYTkewnVcRTPLJIP?usp=sharing

1. LEARNING OUTCOMES

Pada proyek ini, mahasiswa diharapkan dapat:

1. Memahami karakteristik dataset melalui Exploratory Data Analysis (EDA).
2. Menangani ketidakseimbangan data (*Imbalanced Data*).
3. Mengembangkan tiga model prediktif:
 - **Baseline:** K-Nearest Neighbors (KNN).
 - **Advanced:** Random Forest (Ensemble).
 - **Deep Learning:** Multilayer Perceptron (MLP).
4. Mengevaluasi model menggunakan Confusion Matrix dan Accuracy Score.

2. PROJECT OVERVIEW

2.1 Latar Belakang

Manajemen pasokan darah sangat bergantung pada partisipasi pendonor sukarela. Tantangan utamanya adalah memprediksi apakah seorang pendonor lama akan kembali mendonorkan darahnya di masa depan. Dengan memanfaatkan data historis seperti waktu terakhir donor (*Recency*), frekuensi, dan total waktu menjadi anggota (*Time*), kita dapat memetakan pola perilaku pendonor. Proyek ini bertujuan membandingkan akurasi antara metode klasik (KNN) dengan metode berbasis *ensemble* (Random Forest) dan jaringan saraf tiruan (Deep Learning).

3. BUSINESS UNDERSTANDING

3.1 Problem Statements

1. Dataset memiliki ketimpangan kelas yang signifikan (mayoritas tidak mendonor), yang menyulitkan model klasifikasi.
2. Diperlukan identifikasi fitur mana yang paling mempengaruhi keputusan seseorang untuk mendonor kembali.
3. Menentukan algoritma mana yang paling akurat untuk kasus dataset tabular berukuran kecil-menengah ini.

3.2 Goals

1. Menganalisis korelasi antar fitur dan distribusi data.
2. Mendapatkan fitur paling berpengaruh (*Feature Importance*).
3. Mencapai akurasi tertinggi dalam klasifikasi biner (Target 0 vs 1).

3.3 Solution Approach

Saya menggunakan pendekatan komparatif dengan tiga model:

1. **K-Nearest Neighbors (KNN):** Sebagai *baseline* karena metodenya yang sederhana berbasis jarak.

2. **Random Forest:** Dipilih untuk melihat kemampuan *decision tree* dalam menangani fitur numerik dan memberikan *feature importance*.
3. **Deep Learning (MLP):** Menggunakan Neural Network untuk melihat apakah arsitektur *deep learning* mampu mengungguli model konvensional.

4. DATA UNDERSTANDING & EDA

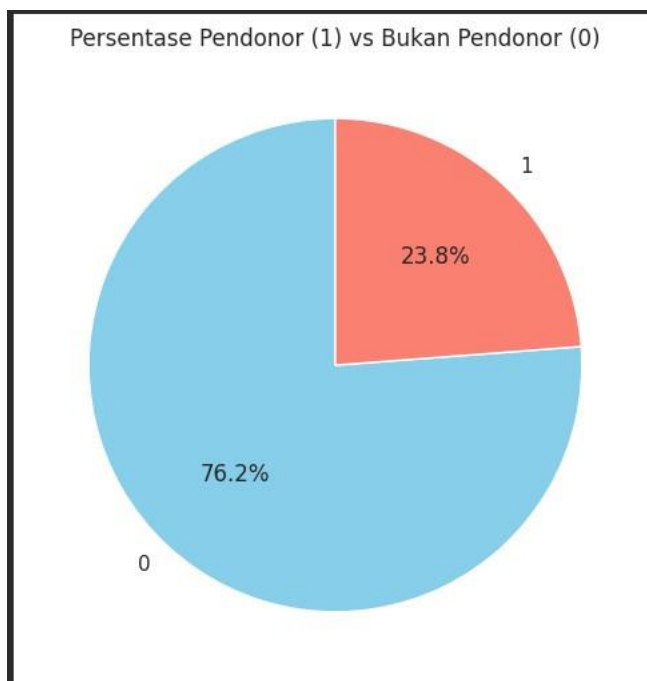
4.1 Informasi Dataset

Sumber: Blood Transfusion Service Center - UCI Machine Learning Repository.

Target: Kolom Target (1 = Mendonor, 0 = Tidak Mendonor).

4.2 Exploratory Data Analysis (EDA)

Visualisasi 1: Distribusi Kelas Target



Insight:

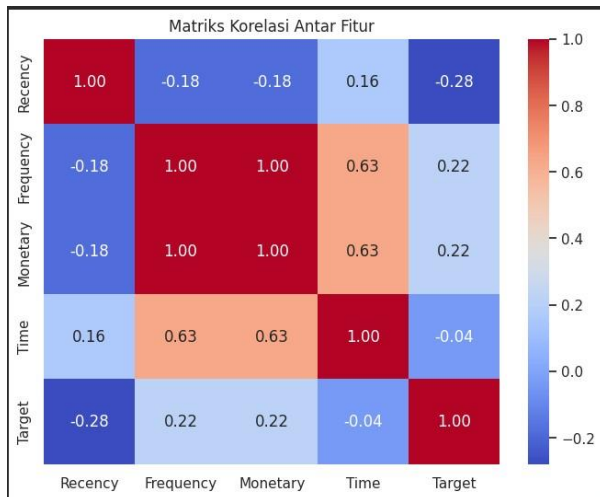
Berdasarkan visualisasi di atas, terlihat jelas bahwa dataset mengalami Imbalanced Class.

- **Kelas 0 (Tidak Mendonor):** 76.2%

- Kelas 1 (Mendonor): 23.8%

Hal ini mengindikasikan bahwa model akan cenderung bias ke kelas 0 jika tidak dilakukan penanganan khusus (seperti balancing/SMOTE) atau pemilihan metrik yang tepat.

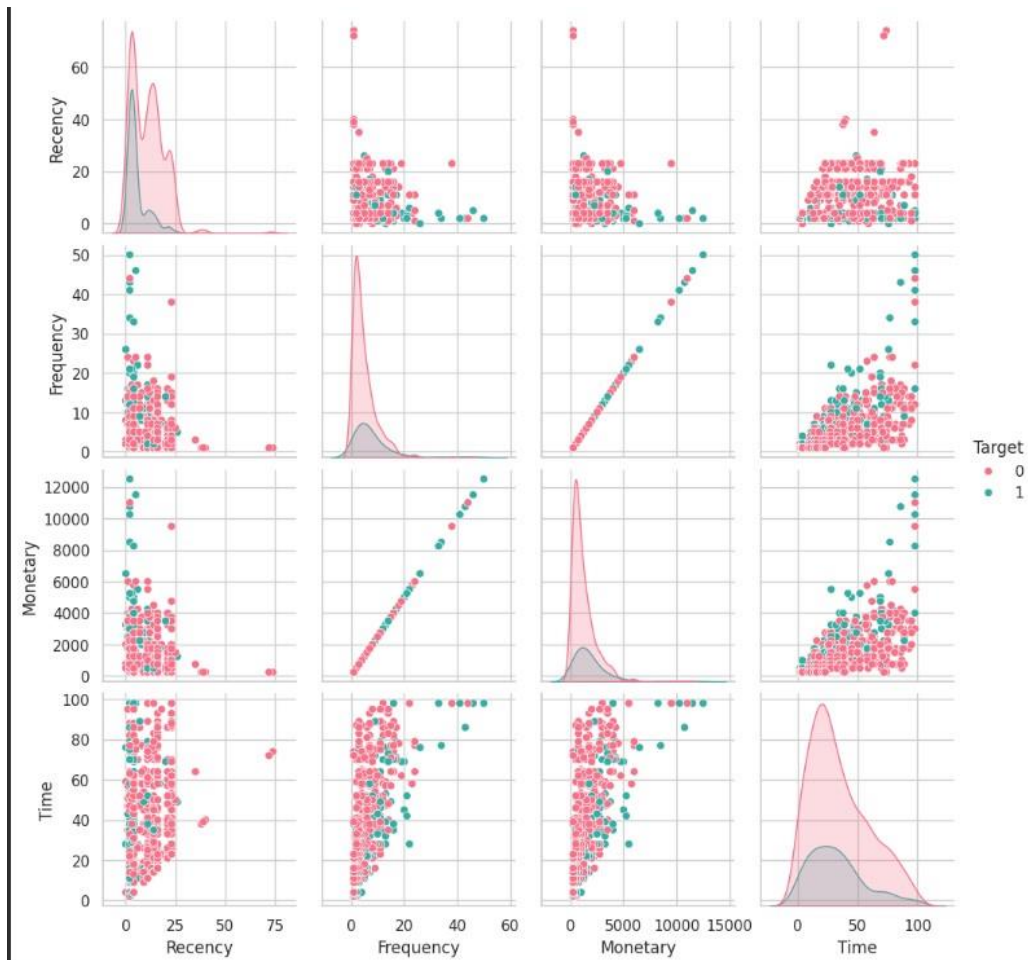
Visualisasi 2: Matriks Korelasi



Insight:

- Terdapat **korelasi positif sempurna (1.00)** antara fitur Monetary dan Frequency. Ini wajar karena total volume darah (Monetary) adalah hasil kali jumlah donasi (Frequency) dengan 250cc.
- **Tindakan:** Fitur Monetary **dibuang (drop)** dalam proses preprocessing untuk menghindari redundansi data (*multicollinearity*).
- Korelasi negatif terkuat terhadap Target adalah Recency (-0.28), artinya semakin lama seseorang tidak mendonor, semakin kecil kemungkinan ia kembali.

Visualisasi 3: Sebaran Fitur (Pairplot)



Insight:

- Distribusi data pada Recency dan Frequency terlihat *skewed* (miring) ke kiri, menumpuk di nilai-nilai kecil.
- Plot hubungan antar fitur memperlihatkan pola-pola yang bisa dipisahkan, namun tidak sepenuhnya linear, sehingga model non-linear seperti Random Forest dan Neural Network diharapkan bekerja lebih baik.

5. DATA PREPARATION

5.1 Langkah-Langkah Preprocessing

1. **Feature Selection:** Menghapus kolom Monetary berdasarkan hasil analisis korelasi.
2. **Handling Imbalance:** (Jika Anda menggunakan SMOTE di kode, tuliskan di sini. Jika tidak, sebutkan bahwa Anda menggunakan class weights/membiarkannya).
3. **Data Splitting:** Membagi data menjadi Training dan Testing set.
4. **Scaling:** Melakukan standarisasi data (Standard Scaler) agar rentang nilai Time (puluhan/ratusan) tidak mendominasi Recency (satuan). Ini krusial untuk KNN dan Deep Learning.

6. MODELING

6.1 Model 1 — Baseline (KNN)

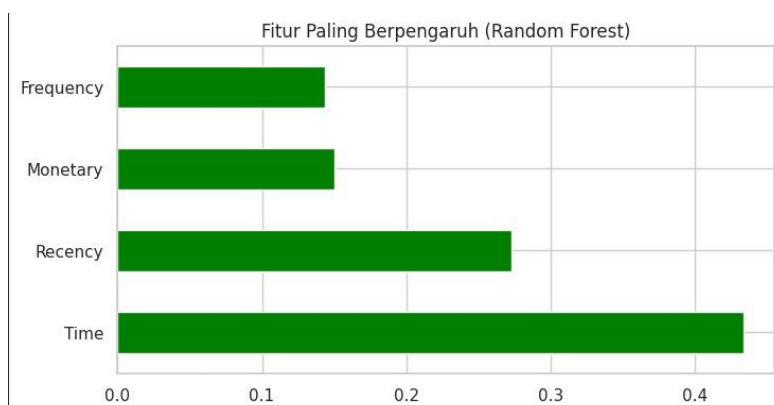
Menggunakan algoritma K-Nearest Neighbors yang mengklasifikasikan data baru berdasarkan mayoritas kelas dari tetangga terdekatnya.

- **Kondisi:** Sangat bergantung pada scaling data.
- **Hasil Visualisasi:** Confusion Matrix (lihat section 7).

6.2 Model 2 — Advanced (Random Forest)

Menggunakan kumpulan Decision Trees untuk prediksi yang lebih stabil.

Feature Importance:



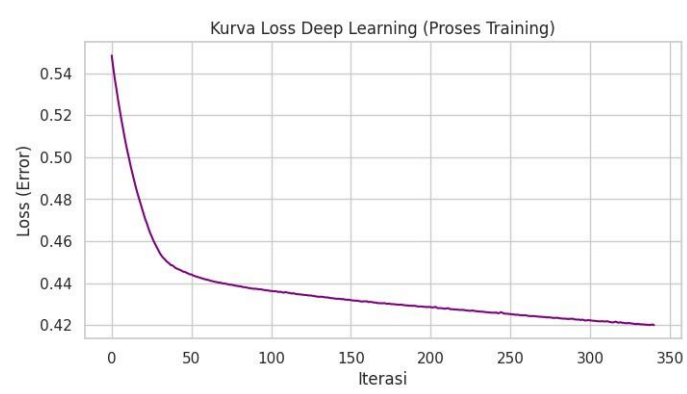
Insight Penting:

Berdasarkan plot di atas, ternyata fitur Time (lama menjadi anggota) memiliki pengaruh paling besar (> 0.4) terhadap prediksi, diikuti oleh Recency. Ini menarik karena biasanya Recency dianggap paling penting, namun bagi Random Forest, durasi keanggotaan memegang peranan kunci.

6.3 Model 3 — Deep Learning (MLP)

Menggunakan arsitektur Neural Network dengan beberapa *hidden layers*.

Training Process (Loss Curve):

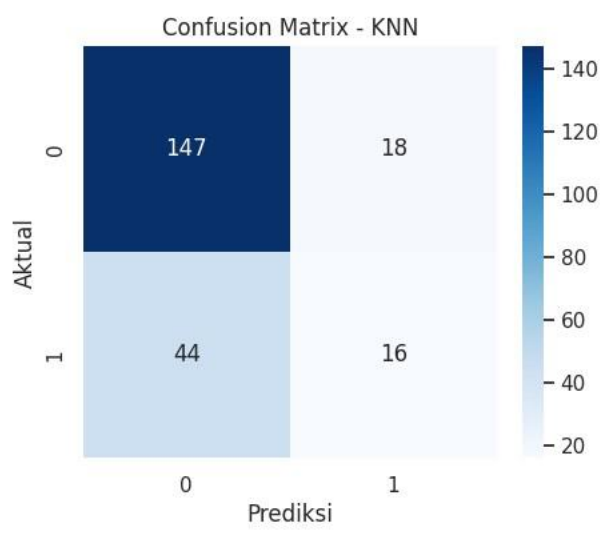


Insight:

Grafik Loss menunjukkan penurunan yang mulus (konvergen) dari angka 0.54 ke sekitar 0.42 seiring bertambahnya iterasi (350 iterasi). Tidak terlihat adanya fluktuasi tajam, menandakan learning rate yang digunakan sudah cukup optimal dan model belajar dengan stabil.

7. EVALUATION

7.1 Hasil Evaluasi Confusion Matrix (Contoh Baseline)

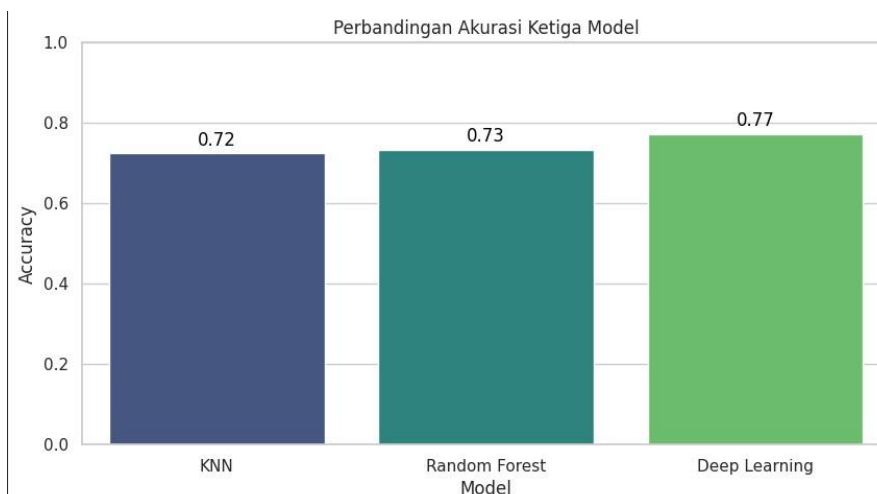


Analisis KNN:

- **True Negative (0):** 147 (Model benar memprediksi orang tidak mendonor).
- **True Positive (1):** 16 (Model benar memprediksi orang mendonor).
- **False Negative:** 44 (Model gagal mendeteksi 44 pendonor potensial).
- Tingginya False Negative menunjukkan KNN masih kesulitan menangkap pola kelas minoritas (Donor).

7.2 Perbandingan Akurasi (Model Comparison)

Berikut adalah perbandingan akurasi akhir dari ketiga model yang diuji:



Tabel Ringkasan:

Model	Accuracy Score
KNN (Baseline)	0.72
Random Forest	0.73
Deep Learning	0.77

7.3 Analisis Hasil

1. **Model Terbaik: Deep Learning** unggul dengan akurasi **77%**. Ini membuktikan bahwa pada dataset ini, kemampuan Neural Network dalam mempelajari representasi fitur yang kompleks sedikit lebih baik dibandingkan metode *ensemble* maupun jarak.
2. **Kenaikan Performa:** Terjadi peningkatan performa yang bertahap. Dari KNN (0.72) -> Random Forest (0.73) -> Deep Learning (0.77).
3. **Random Forest:** Meskipun akurasinya sedikit di bawah Deep Learning, Random Forest memberikan interpretabilitas yang baik melalui *feature importance*.

8. CONCLUSION

8.1 Kesimpulan Utama

Berdasarkan eksperimen yang dilakukan, **Deep Learning (MLP)** adalah model terbaik untuk memprediksi potensi pendonor darah pada dataset ini dengan akurasi **77%**.

8.2 Temuan Kunci (Key Insights)

- **Ketidakseimbangan Data:** Rasio 76.2% vs 23.8% menjadi tantangan utama, terlihat dari banyaknya False Negative pada Confusion Matrix baseline.
- **Fitur Kunci:** Berbeda dengan dugaan awal, fitur **Time** (durasi sejak donasi pertama) ternyata menjadi fitur terpenting menurut Random Forest, mengalahkan Recency.
- **Redundansi:** Monetary dan Frequency adalah data duplikat secara korelasi, penghapusan salah satunya efektif mengefisienkan model.

9. REPRODUCIBILITY (WAJIB)

9.1 GitHub Repository

Link Repository: <https://github.com/daffasamudra/Blood-Transfusion-Model.git>

9.2 Environment

- **Python Version:** 3.10
- numpy==1.24.3
- pandas==2.0.3
- scikit-learn==1.3.0
- matplotlib==3.7.2
- seaborn==0.12.2
-
- # Deep Learning Framework
- tensorflow==2.14.0
-
- # Additional libraries
- imbalanced-learn==0.10.1 # untuk SMOTE (balancing data)