

# Laporan Singkat: Pemodelan & Evaluasi (Pertemuan 5)

Tanggal: 25 Oktober 2025 Disusun oleh: [Nama Anda] Mata Kuliah: [Mata Kuliah Anda]

## 1. Pendahuluan

Laporan ini mendokumentasikan langkah-langkah yang diambil dalam Pertemuan 5, yang berfokus pada pembangunan, *tuning*, dan evaluasi model *machine learning*. Tujuan dari tahap ini adalah untuk menggunakan data yang telah diproses dari Pertemuan 4 (`processed_kelulusan.csv`) untuk melatih beberapa model klasifikasi, membandingkannya, dan memilih model terbaik untuk evaluasi akhir pada *test set*.

## 2. Langkah 1: Pemuatan Data dan Pemisahan (Splitting)

- Pemuatan Data:** Dataset `processed_kelulusan.csv` yang merupakan hasil dari Pertemuan 4 berhasil dimuat.
- Pemisahan Data:** Dataset dibagi menjadi tiga bagian menggunakan `train_test_split` dengan strategi `stratify=y` untuk memastikan proporsi kelas (Lulus/Gagal) seimbang di setiap set:
  - Train Set (70%):** Digunakan untuk melatih model.
  - Validation Set (15%):** Digunakan untuk membandingkan performa model *baseline* vs. *tuned* (setelah *tuning*).
  - Test Set (15%):** "Data tersembunyi" yang hanya digunakan satu kali di akhir untuk evaluasi final model terpilih.

## 3. Langkah 2 & 3: Pipeline dan Pemodelan Baseline

Untuk memastikan perbandingan yang adil dan mencegah kebocoran data (*data leakage*), sebuah *Pipeline preprocessing* universal dibuat. *Pipeline* ini secara otomatis menerapkan langkah-langkah berikut:

- Imputasi:** Mengisi nilai yang hilang (jika ada) menggunakan median (`SimpleImputer`).
- Scaling:** Menstandarisasi fitur numerik (`StandardScaler`), yang penting untuk model seperti Logistic Regression.

Dua model dievaluasi pada *validation set*:

### 3.1. Model 1: Logistic Regression (Baseline)

- **Tujuan:** Sebagai model dasar pembandingan.
- **Hasil (Validation Set):** Model ini dilatih dan dievaluasi. Laporan `classification_report` memberikan metrik F1-Macro *baseline*. (Misalnya, *F1 Macro (val): 0.909...*)

### 3.2. Model 2: Random Forest (Alternatif)

- **Tujuan:** Sebagai model *ensemble* yang lebih kompleks.
- **Hasil (Validation Set):** Model ini juga dilatih dan dievaluasi. (Misalnya, *F1 Macro (val): 1.0...*)

Dari perbandingan awal pada *validation set*, Random Forest menunjukkan performa yang lebih unggul daripada Logistic Regression.

## 4. Langkah 4: Validasi Silang dan Tuning (GridSearch)

Model Random Forest dipilih untuk dioptimasi lebih lanjut menggunakan `GridSearchCV` dengan *5-fold Stratified Cross-Validation* pada **train set**.

- **Tujuan:** Untuk menemukan kombinasi parameter terbaik (`max_depth` dan `min_samples_split`) secara sistematis.
- **Hasil Tuning:**
  - **Best CV F1:** Skor F1-Macro rata-rata terbaik dari *cross-validation*. (Misalnya, *Best CV F1: 0.969...*)
  - **Best Params:** Kombinasi parameter terbaik yang ditemukan. (Misalnya, `{'clf__max_depth': None, 'clf__min_samples_split': 2}`)
- **Hasil (Validation Set):** Model terbaik hasil *tuning* kemudian dievaluasi pada *validation set*, mengkonfirmasi performa kuatnya. (Misalnya, *Best RF F1(val): 1.0...*)

## 5. Langkah 5 & 6: Evaluasi Akhir dan Penyimpanan Model

### 5.1. Evaluasi pada Test Set

Model terbaik dari hasil `GridSearchCV` (`best_rf`) dipilih sebagai model final. Model ini dievaluasi **satu kali** pada **test set** yang belum pernah terlihat sebelumnya.

- **Metrik Kinerja (Test Set):**
  - **F1 Macro:** [Isi skor F1-Macro dari output Anda]
  - **ROC-AUC:** [Isi skor ROC-AUC dari output Anda]

#### Confusion Matrix:

```
[[R_Gagal_Pred_Gagal, R_Gagal_Pred_Lulus],
 [R_Lulus_Pred_Gagal, R_Lulus_Pred_Lulus]]
```

- (Salin hasil Confusion Matrix dari output Anda di sini)

- **Classification Report:** (Salin hasil *Classification Report (test)* dari output Anda di sini)

## 5.2. Visualisasi ROC

Kurva ROC (Receiver Operating Characteristic) dibuat dan disimpan sebagai `p5_roc_test.png`. Plot ini secara visual menunjukkan kemampuan model final dalam membedakan antara kelas Lulus dan Gagal pada *test set*. Nilai AUC (Area Under the Curve) yang tinggi (mendekati 1.0) mengindikasikan performa yang sangat baik.

## 5.3. Penyimpanan Model

Model final yang telah dilatih dan divalidasi disimpan ke dalam file `model_p5.pkl` menggunakan `joblib`. Model ini sekarang siap untuk digunakan untuk *inference* atau di-deploy ke dalam aplikasi (seperti API Flask).