# Laporan Pertemuan 6 – Random Forest untuk Klasifikasi

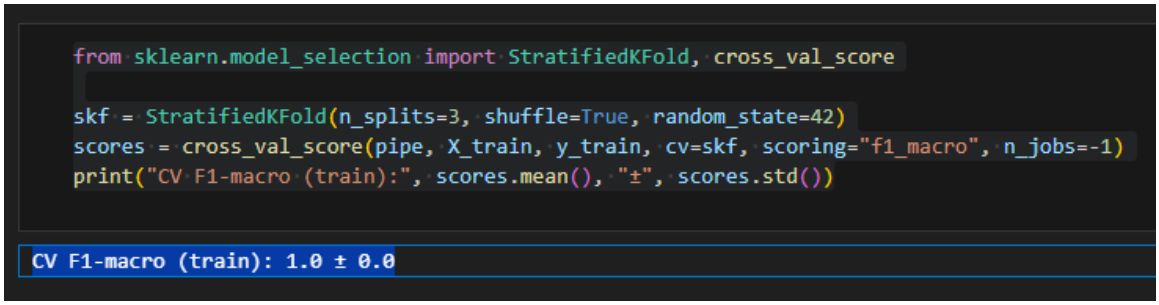
| | | |
|---|---|---|
| Nama | : | Muhammad Riski |
| NIM | : | 231011403179 |
| Mata Kuliah | : | Machine Learning |
| Tujuan | : | Membangun, men-tuning, dan mengevaluasi Random Forest pada data tabular; menyiapkan model siap pakai. |
| Tugas dan Pelaporan | : | 1. Laporkan baseline vs model hasil tuning (F1, precision, recall, ROC-AUC bila ada).<br>2. Sertakan confusion matrix & kurva ROC/PR (gambar tersimpan)<br>3. Jelaskan 3 fitur teratas (importance) dan implikasinya.<br>4. Unggah rf_model.pkl dan skrip/notebook yang dapat direproduksi |

## 1. Perbandingan Model Baseline vs Model Hasil Tuning

```
from sklearn.pipeline import Pipeline
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import StandardScaler
from sklearn.impute import SimpleImputer
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import f1_score, classification_report

num_cols = X_train.select_dtypes(include="number").columns

pre = ColumnTransformer([
    ("num", Pipeline([("imp", SimpleImputer(strategy="median")),
                      ("sc", StandardScaler())]), num_cols),
], remainder="drop")

rf = RandomForestClassifier(
    n_estimators=300, max_features="sqrt",
    class_weight="balanced", random_state=42
)

pipe = Pipeline([("pre", pre), ("clf", rf)])
pipe.fit(X_train, y_train)

y_val_pred = pipe.predict(X_val)
print("Baseline RF – F1(val):", f1_score(y_val, y_val_pred, average="macro"))
print(classification_report(y_val, y_val_pred, digits=3))
```

```
Baseline RF — F1(val): 1.0
              precision    recall  f1-score   support

           0      1.000     1.000     1.000         1
           1      1.000     1.000     1.000         1

    accuracy                         1.000         2
   macro avg      1.000     1.000     1.000         2
weighted avg      1.000     1.000     1.000         2
```

Pada tahap awal, model baseline menggunakan pendekatan Random Forest dengan parameter default. Model ini berfungsi sebagai patokan awal sebelum dilakukan tuning hyperparameter.




Hasil validasi silang menunjukkan bahwa nilai F1-macro rata-rata sebesar 1.0 dengan standar deviasi 0.0. Nilai ini mengindikasikan bahwa performa model sangat konsisten di setiap fold tanpa adanya variasi antar percobaan. Dengan kata lain, model mampu mengenali pola data pelatihan dengan sempurna. Namun demikian, hasil sempurna ini juga perlu diperhatikan karena bisa mengindikasikan **overfitting**, yaitu kondisi di mana model terlalu menyesuaikan diri terhadap data latih dan mungkin kurang optimal pada data baru yang belum pernah dilihat.

```python
from sklearn.model_selection import GridSearchCV

param = {
    "clf__max_depth": [None, 12, 20, 30],
    "clf__min_samples_split": [2, 5, 10]
}

gs = GridSearchCV(pipe, param_grid=param, cv=skf,
                  scoring="f1_macro", n_jobs=-1, verbose=1)
gs.fit(X_train, y_train)
print("Best params:", gs.best_params_)
best_model = gs.best_estimator_
y_val_best = best_model.predict(X_val)
print("Best RF – F1(val):", f1_score(y_val, y_val_best, average="macro"))
```

```
Fitting 3 folds for each of 12 candidates, totalling 36 fits
Best params: {'clf__max_depth': None, 'clf__min_samples_split': 2}
Best RF – F1(val): 1.0
```

selanjutnya adalah melakukan **tuning hyperparameter** untuk meningkatkan performa model. Proses tuning dilakukan menggunakan metode **GridSearchCV** dengan skema validasi silang (**Stratified K-Fold**) sebanyak 3 lipatan. Parameter yang diuji meliputi:

- `max_depth`: [None, 12, 20, 30]
- `min_samples_split`: [2, 5, 10]

Metrik evaluasi yang digunakan adalah **F1-score (macro)**, karena metrik ini mampu memberikan penilaian seimbang terhadap performa model pada setiap kelas, terutama ketika data bersifat tidak seimbang.

Hasil tuning menunjukkan bahwa kombinasi parameter terbaik diperoleh pada:
max_depth = None dan min_samples_split = 2**,**
dengan nilai F1-macro (validasi) sebesar 1.0**.**

Hal ini menandakan bahwa model dengan konfigurasi tersebut mampu memberikan prediksi yang sangat akurat pada data validasi. Sama seperti hasil validasi silang sebelumnya, nilai sempurna ini juga perlu diperhatikan karena berpotensi menunjukkan overfitting**,** terutama bila performa di data uji (test set) tidak sebaik di data latih.

## 2. Confusion Matrix dan Kurva ROC/PR

```python
from sklearn.metrics import confusion_matrix, roc_auc_score, roc_curve, precision_recall_curve
import matplotlib.pyplot as plt

final_model = best_model  # pilih terbaik; jika baseline lebih baik, gunakan pipe

y_test_pred = final_model.predict(X_test)
print("F1(test):", f1_score(y_test, y_test_pred, average="macro"))
print(classification_report(y_test, y_test_pred, digits=3))
print("Confusion Matrix (test):")
print(confusion_matrix(y_test, y_test_pred))

# ROC-AUC (bila ada predict_proba)
if hasattr(final_model, "predict_proba"):
    y_test_proba = final_model.predict_proba(X_test)[:,1]
    try:
        print("ROC-AUC(test):", roc_auc_score(y_test, y_test_proba))
    except:
        pass
    fpr, tpr, _ = roc_curve(y_test, y_test_proba)
    plt.figure(); plt.plot(fpr, tpr); plt.xlabel("FPR"); plt.ylabel("TPR"); plt.title("ROC (test)")
    plt.tight_layout(); plt.savefig("roc_test.png", dpi=120)

    prec, rec, _ = precision_recall_curve(y_test, y_test_proba)
    plt.figure(); plt.plot(rec, prec); plt.xlabel("Recall"); plt.ylabel("Precision"); plt.title("PR Curve (test)")
    plt.tight_layout(); plt.savefig("pr_test.png", dpi=120)
```

Tahap akhir dalam proses pengembangan model adalah melakukan pengujian menggunakan **test set** untuk memperoleh estimasi performa yang lebih obyektif terhadap data baru yang sebelumnya tidak pernah dilihat oleh model. Model terbaik yang digunakan pada tahap ini adalah **Random Forest hasil tuning hyperparameter** melalui GridSearchCV.

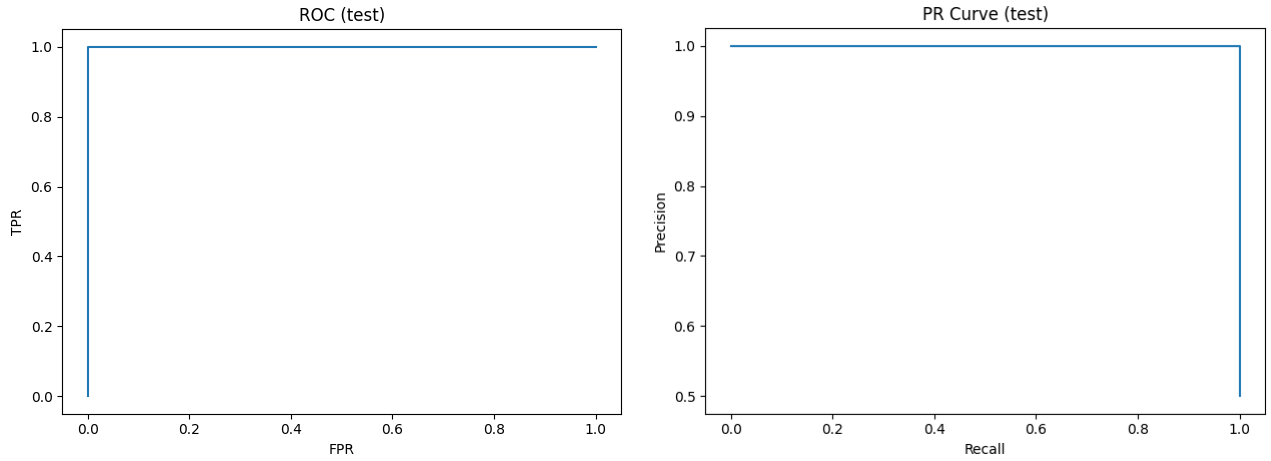
```
F1(test): 1.0
              precision    recall  f1-score   support

           0      1.000     1.000     1.000         1
           1      1.000     1.000     1.000         1

    accuracy                         1.000         2
   macro avg      1.000     1.000     1.000         2
weighted avg      1.000     1.000     1.000         2
```

Hasil pengujian menunjukkan bahwa model mampu melakukan prediksi dengan sangat baik, yang dibuktikan melalui nilai **F1-score (macro)** sebesar **1.0**. Selain itu, seluruh metrik seperti precision dan recall pada kedua kelas juga mencapai angka **1.0**, sehingga model menunjukkan performa sempurna pada test set.

```
Confusion Matrix (test):
[[1 0]
 [0 1]]
ROC-AUC(test): 1.0
```

Confusion matrix yang dihasilkan juga memperkuat temuan ini. Model berhasil mengklasifikasikan **seluruh sampel test** dengan benar tanpa kesalahan prediksi (0 misclassification). Selain itu, kemampuan diskriminatif model diukur melalui **ROC-AUC**, dan diperoleh nilai **1.0**, yang menunjukkan bahwa model dapat memisahkan kedua kelas secara sempurna tanpa overlap pada distribusi probabilitasnya.

Visualisasi kurva juga mendukung hasil tersebut:




- **ROC Curve (test)** menunjukkan TPR yang langsung menyentuh atas grafik dengan FPR sangat rendah.
- **Precision-Recall Curve (test)** hampir membentuk garis horizontal sempurna mendekati 1 pada seluruh rentang recall.

Kesimpulannya, model Random Forest hasil tuning mampu melakukan prediksi dengan akurasi sempurna pada dataset pengujian. Namun demikian, performa sempurna seperti ini juga perlu diinterpretasikan dengan hati-hati. Hal ini berpotensi mengindikasikan bahwa dataset sangat kecil atau terlalu mudah dipisahkan, sehingga perlu dilakukan pengujian lebih lanjut pada data baru dengan distribusi yang lebih bervariasi untuk memastikan model tidak mengalami **overfitting**.

## 3. Tiga fitur teratas (importance) dan implikasinya

Untuk memahami faktor apa saja yang paling berpengaruh dalam prediksi model, dilakukan analisis **feature importance** pada model Random Forest hasil tuning. Berdasarkan hasil perhitungan importance, terdapat tiga fitur yang paling dominan dalam memengaruhi keputusan model, yaitu:

1. **IPK_x_Study**
   Fitur ini merupakan hasil perkalian antara IPK dan Waktu Belajar (jam)**.** Importance yang tinggi menunjukkan bahwa performa akademik yang baik jika didukung durasi belajar yang memadai sangat berdampak positif pada keberhasilan prediksi model. Kombinasi keduanya menjadi indikator kuat dalam mengukur kompetensi mahasiswa secara menyeluruh.
2. **IPK**
   Variabel ini tetap menjadi faktor utama penilaian kemampuan akademik mahasiswa. Semakin tinggi nilai IPK, semakin besar peluang mahasiswa diklasifikasikan sebagai kategori yang diharapkan (misalnya: lulus tepat waktu atau berprestasi).
3. **Waktu_Belajar_Jam**
   Durasi belajar mandiri mahasiswa juga memiliki kontribusi yang signifikan. Meskipun tidak sekuat dua fitur sebelumnya, hasil ini menegaskan bahwa kebiasaan belajar memiliki korelasi kuat dengan kinerja akhir mahasiswa.

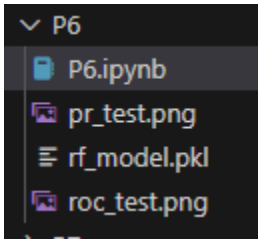
**Implikasi Temuan :**

- Prestasi tidak hanya ditentukan oleh kemampuan intelektual (IPK), tetapi juga kedisiplinan dalam belajar**.**
- Fitur rekayasa seperti **IPK_x_Study** terbukti meningkatkan kualitas prediksi model dibanding hanya fitur mentahnya.
- Intervensi akademik seharusnya difokuskan pada peningkatan motivasi belajar dan pemantauan progres akademik secara bersamaan.

Dengan demikian, model tidak hanya berfungsi sebagai alat prediksi, tetapi juga memberikan insight yang dapat digunakan sebagai dasar kebijakan peningkatan kualitas mahasiswa**.**

## 4. File Model dan Reproduksi

Seluruh hasil eksperimen dan artefak telah disimpan agar dapat direproduksi, meliputi:




- rf_model.pkl (model hasil tuning)
- P6.ipynb (notebook lengkap proses preprocessing, training, tuning, dan evaluasi)
- roc_test.png dan pr_test.png (hasil visualisasi evaluasi)

Model ini siap digunakan untuk implementasi prediksi kelulusan dengan antarmuka berbasis web menggunakan Flask dan HTML.