

Sistem Rekomendasi Penjurusan Calon Siswa SMA

Capstone Project **Laskar Ai**



MEET THE TEAM

LintasNusa



Advisor

Mochamad Rafy Ardhanie



A211XBF253

Lufi Harneni

Universitas Indraprasta PGRI



A200YBF339

Muhammad Rafif Alfarizti

Universitas Diponegoro



A006XBM143

Elyzia Janara Khansa

Universitas Brawijaya

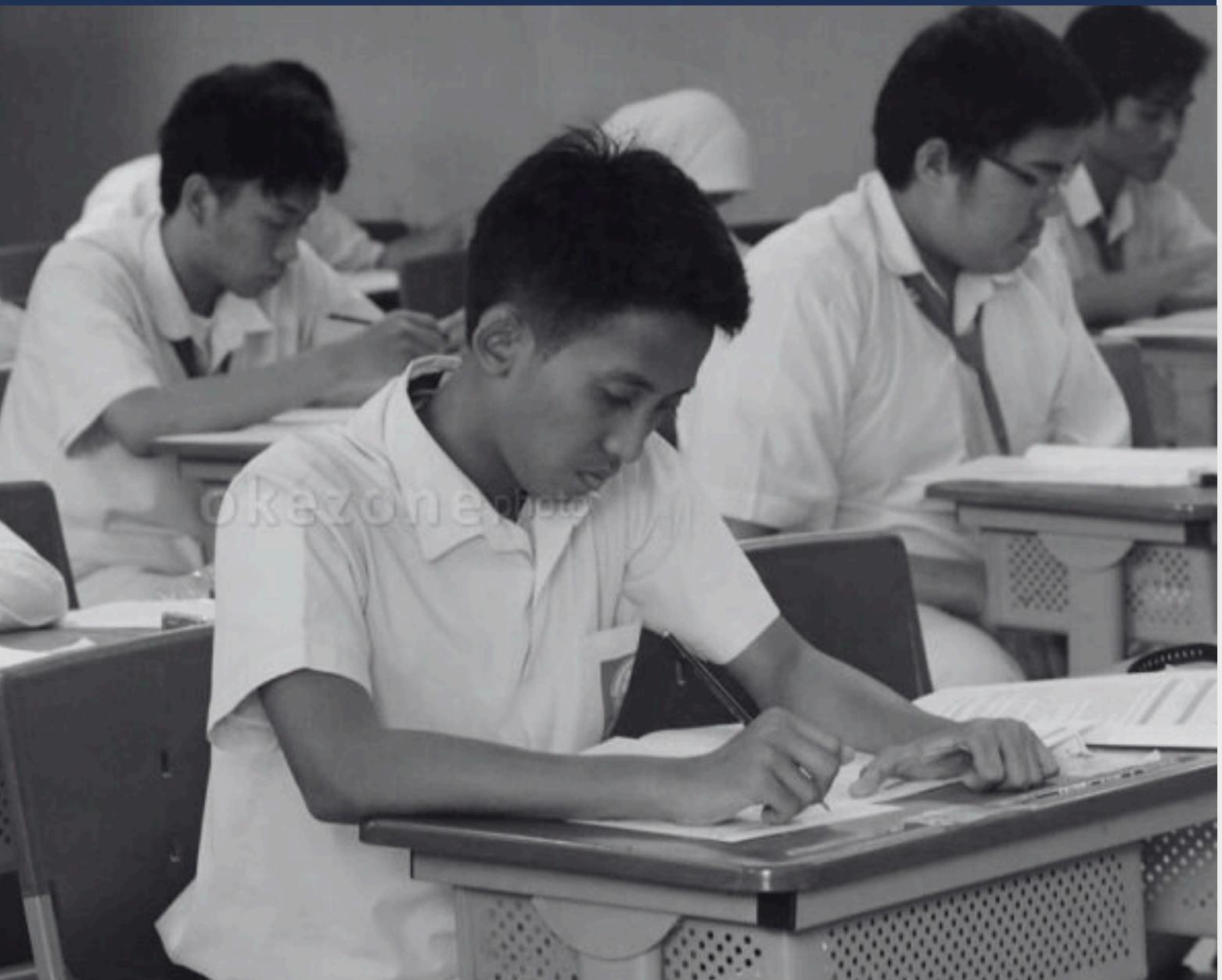


A200YBF317

Muhammad Fathi Farhat

Universitas Diponegoro

Latar Belakang



Proses penjurusan di tingkat Sekolah Menengah Atas (SMA) masih sering dilakukan tanpa mempertimbangkan data dan karakteristik individual siswa secara menyeluruh. Hal ini berisiko menimbulkan berbagai dampak negatif, seperti rendahnya motivasi belajar, ketidaksesuaian dengan jurusan di pendidikan tinggi, hingga ketidakpastian dalam merancang masa depan karier.

Seiring berkembangnya teknologi, khususnya dalam bidang kecerdasan buatan dan analisis data, pendekatan berbasis machine learning dapat menjadi solusi untuk menghasilkan sistem rekomendasi penjurusan yang lebih objektif dan akurat. Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem rekomendasi penjurusan SMA, yaitu IPA, IPS, atau Bahasa, berdasarkan data historis siswa SMP yang mencakup nilai akademik, tingkat kehadiran, serta keterlibatan dalam aktivitas non-akademik.

Outline

- 01** Alasan Pemilihan Tema
- 02** Pernyataan Masalah
- 03** Hasil yang sudah ada
- 04** Implementasi & Alasan

- 05** Hasil & Interpretasi
- 06** Dokumentasi
- 07** Rencana Pengembangan



Alasan Pemilihan Tema

Penjurusan yang Lebih Adil dan Objektif

Banyak siswa terjebak di jurusan yang tidak sesuai karena keputusan penjurusan masih sering berbasis opini, bukan data

Pemanfaatan Teknologi untuk Pemerataan Akses

sekolah-sekolah—termasuk di daerah— bisa menerapkan proses penjurusan berbasis evidence

Mendorong Siswa Berkembang Sesuai Minat dan Bakat

Inovasi ini membuka peluang agar setiap siswa bisa berkembang secara optimal

Pernyataan Masalah

- 1. Bagaimana cara pembuatan sistem rekomendasi penjurusan calon siswa SMA?**
- 2. Bagaimana alur pemberian rekomendasi penjurusan calon siswa SMA?**
- 3. Bagaimana hasil evaluasi seluruh model yang diuji?**

Hasil yang Sudah Ada

Ruang Guru

- Persentase Kesamaan: $\pm 30\%$
- Fitur: Tes minat bakat dan saran pembelajaran berbasis peminatan.
- Kekurangan:
- Fokusnya lebih ke persiapan UTBK dan akademik umum.
- Tidak secara khusus memberikan rekomendasi jurusan (IPA, IPS, Bahasa) untuk siswa kelas 9 berdasarkan data historis akademik.

Psikotes Manual Sekolah

- Persentase Kesamaan: $\pm 50\%$
- Fitur: Penjurusan berbasis hasil psikotes manual.
- Kekurangan:
- Hanya dilakukan sekali.
- Bergantung pada subjektivitas guru atau konselor.
- Tidak terintegrasi dengan data akademik siswa secara menyeluruh.

Komponen yang Belum Terpenuhi

- Belum ada sistem yang menggabungkan data nilai akademik historis + tren belajar + machine learning secara langsung untuk penjurusan.
- Belum tersedia dashboard rekomendasi berbasis data yang dapat digunakan sekolah untuk memantau dan menyesuaikan hasil penjurusan.
- Tidak semua siswa memiliki akses ke bimbingan penjurusan yang objektif dan berbasis potensi diri.

Implementasi Model

Kandidat Model & Pertimbangan

XGBoost

Plus: Performa sangat tinggi
Minus: Sulit dijelaskan (kurang interpretabel).

Logistic Regression

Plus: Sangat mudah dijelaskan (interpretabel).
Minus: Performa bisa rendah, kurang cocok untuk data kompleks.

Random Forest Classifier

Plus: Keseimbangan terbaik antara performa dan interpretasi.
Minus: Bukan yang terbaik jika hanya fokus pada performa atau interpretasi.

Alasan Pemilihan Model

Model Terpilih: Random Forest Classifier

Model ini memberikan keseimbangan optimal antara:

- 1). Performa yang baik dalam memprediksi jurusan.
- 2). Kemampuan interpretasi yang memadai untuk menjelaskan keputusan.

Hasil

Classification Report

```
[ ] # akurasi prediksi model
train_accuracy = accuracy_score(y_train, y_pred_train)
test_accuracy = accuracy_score(y_test, y_pred_test)

print(f"Training Accuracy: {train_accuracy:.4f}")
print(f"Testing Accuracy: {test_accuracy:.4f}")
```

```
→ Training Accuracy: 1.0000
Testing Accuracy: 0.8500
```

```
[ ] # Print classification report
print("\n--- Classification Report (Test Set) ---")
print(classification_report(y_test, y_pred_test))
```

```
--- Classification Report (Test Set) ---
precision    recall    f1-score   support
BAHASA        0.90      0.72      0.80       65
IPA           0.86      0.89      0.87       61
IPS           0.81      0.93      0.87       74
accuracy                           0.85      200
macro avg     0.86      0.85      0.85      200
weighted avg  0.86      0.85      0.85      200
```

1. Testing Accuracy: 85%

Model memiliki performa yang cukup baik, mampu mengklasifikasikan jurusan dengan benar sebanyak 85 dari 100 siswa secara rata-rata.

2. Training Accuracy: 100%

Ini menandakan model kemungkinan mengalami overfitting, tetapi tetap berhasil menjaga generalisasi cukup baik di testing set (gap 15%).

Temuan Masalah

Model tidak sensitif terhadap siswa BAHASA (recall = 0.72). Ketika dilakukan prediksi, sistem bisa salah menentukan penjurusan siswa ke jurusan lain.

Hasil

Confusion Matrix



Analisis Per Baris (True Label)

- Actual BAHASA:** Cukup rendah, berarti banyak siswa BAHASA tercampur ke IPA dan IPS.
- Actual IPA:** Sangat baik. Model sangat tepat dan sensitif mengenali siswa IPA.
- Actual IPS:** Paling tinggi. Model sangat kuat dalam mengenali siswa IPS.

Analisis Per Kolom (Predicted Label)

- Predicted BAHASA:** model jarang salah menyebut orang lain sebagai BAHASA, tapi kurang bagus dalam menemukan semua siswa BAHASA.
- Predicted IPA:** model cukup baik mengenali siswa IPA.
- Predicted IPS:** Ini kelas dengan precision terendah, karena banyak siswa lain salah dikira IPS.

Peningkatan

Hyperparameter Tuning

```
[ ] param_grid = {
    'n_estimators': [50, 100, 200],
    'max_depth': [None, 10, 20, 30],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}

[ ] print("Performing Grid Search...")
grid_search = GridSearchCV(
    RandomForestClassifier(random_state=42),
    param_grid,
    cv=3,
    scoring='accuracy',
    n_jobs=-1,
    verbose=1
)

grid_search.fit(X_train, y_train)

print(f"Best parameters: {grid_search.best_params_}")
print(f"Best cross-validation score: {grid_search.best_score_.:.4f}")

→ Performing Grid Search...
Fitting 3 folds for each of 108 candidates, totalling 324 fits
Best parameters: {'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 100}
Best cross-validation score: 0.8198
```

```
== FINAL MODEL EVALUATION ==
Final Model Performance:
Accuracy: 0.8500
Precision: 0.8584
Recall: 0.8500
F1-Score: 0.8488

--- Final Classification Report ---
precision    recall   f1-score   support
BAHASA       0.92     0.74     0.82      65
IPA          0.87     0.87     0.87      61
IPS          0.79     0.93     0.86      74
accuracy           0.85      200
macro avg       0.86     0.85     0.85      200
weighted avg    0.86     0.85     0.85      200
```

Interpretasi Parameter Terbaik

Best Parameter

1. max_depth = None

Definisi: Tidak ada batasan maksimum terhadap kedalaman setiap decision tree dalam ensemble.

Implikasi: Memberikan fleksibilitas maksimal dalam membentuk pembelahan (splits), memungkinkan pohon mempelajari struktur relasi fitur-target secara mendalam. Namun, juga meningkatkan risiko overfitting, terutama jika tidak dikendalikan oleh parameter lain seperti min_samples_split atau min_samples_leaf

2. min_samples_leaf = 1

Definisi: Jumlah minimum sampel yang diperlukan untuk berada pada satu daun (terminal node) adalah 1.

Implikasi: Memberikan model granularitas maksimum dalam membentuk keputusan klasifikasi, memungkinkan identifikasi outlier dan minor classes. Dalam model single-tree seperti Decision Tree biasa, hal ini akan sangat rentan terhadap high variance. Namun dalam Random Forest, variansi dikurangi secara substansial oleh mekanisme ensemble, sehingga lebih aman.

3. min_samples_split = 5

Definisi: Minimum jumlah sampel yang dibutuhkan dalam satu node untuk memicu pembelahan selanjutnya.

Implikasi: Membatasi pertumbuhan pohon terhadap pembelahan yang tidak signifikan secara statistik. Secara efektif bertindak sebagai mekanisme regularisasi, menghindari pemecahan node yang berdasarkan pada noise atau variasi tidak substansial.

4. n_estimators = 100

Definisi: Jumlah pohon keputusan (decision trees) yang dibentuk dalam ensemble.

Implikasi: Jumlah pohon yang lebih besar secara umum meningkatkan stabilitas model dan memperkecil variansi prediksi. Namun peningkatan jumlah pohon juga akan menaikkan kompleksitas komputasi dan waktu pelatihan.

Best Cross-Validation Score: 0.8198

Nilai ini menunjukkan estimasi kinerja model terhadap data tak terlihat, dihitung melalui k-fold cross-validation. Dengan rata-rata akurasi 81.98%, model menunjukkan kemampuan generalisasi yang baik, menunjukkan bahwa parameter hasil tuning memfasilitasi keseimbangan optimal antara overfitting dan underfitting.

Dokumentasi

Menjalankan Aplikasi Secara Lokal

1. Pastikan terdapat struktur folder berikut

- ├── pred.py
- ├── random_forest_model.pkl
- ├── scaler.pkl
- ├── README.md
- ├── requirements.txt
- └── (venv/)

2. Aktifkan virtual environment

- source venv/bin/activate # Untuk Mac/Linux
- venv\Scripts\activate # Untuk Windows

3. Jalankan Streamlit melalui terminal

- streamlit run pred.py

4. Jika berhasil, Streamlit akan membuka browser secara otomatis atau menampilkan URL lokal seperti

- Local URL: http://localhost:8501
- Network URL: http://192.168.68.111:8501

Menjalankan Aplikasi melalui tautan Streamlit Cloud

Buka tautan berikut dengan browser Anda:

<https://predsiswa.streamlit.app/>

[Link Git README.md](#)

Rencana Pengembangan Proyek

Total Estimasi: Rp10.500.000

Bulan 1 – Validasi Data & Model

- Kumpulkan data tambahan & tuning Random Forest
- Resource: 1 Data Scientist, Guru BK
- Biaya: Rp1.000.000

Bulan 4 – Uji Coba Terbatas

- Uji sistem di SMPN 2 Wungu + ambil feedback
- Resource: Data Analyst, Guru
- Biaya: Rp1.000.000

Bulan 2 – Pembuatan Dashboard Web

- Buat UI web sederhana untuk hasil rekomendasi
- Resource: Frontend & Backend Dev
- Biaya: Rp2.500.000 (hosting, domain)

Bulan 5 – Perbaikan Sistem

- Perbaiki UI & tingkatkan akurasi model
- Resource: UI Designer, ML Engineer
- Biaya: Rp1.500.000

Bulan 3 – Integrasi Data Sekolah

- Buat UI web sederhana untuk hasil rekomendasi
- Resource: Frontend & Backend Dev
- Biaya: Rp2.500.000 (hosting, domain)

Bulan 6 – Peluncuran Lokal

- Rilis beta ke sekolah-sekolah mitra + workshop
- Resource: PM, Tim Teknis
- Biaya: Rp3.000.000

Terima Kasih

