TUGAS MANDIRI
FUNDAMENTALS OF DATA MINING

LAPORAN ANALISIS & HASIL PENGOLAHAN DATA
(PYTHON + DATA MINING)

Nama : Amelia Basiani Nariswari
NPM : 231510052
Dosen : Erlin Elisa, S.Kom., M.Kom.

PROGRAM STUDI SISTEM INFORMASI
FAKULTAS TEKNIK DAN KOMPUTER
UNIVERSITAS PUTERA BATAM

2026

1 Deskripsi Dataset

- **Sumber dataset** : Kaggle.com
  https://www.kaggle.com/datasets/khushikyad001/ai-automation-risk-by-job-role
- **Jumlah record** : 3000 data
- **Jumlah atribut** : 25 atribut
- **Tipe data** : campuran (Numerik dan Kategorikal)
  - Numerik: avg_salary_usd, experience_required_years, task_repetition_level, creativity_requirement, analytical_complexity, percent_tasks_automatable, job_growth_rate, skill_complexity_score, training_hours_needed, job_demand_index, automation_risk_score, dan atribut numerik lainnya.
  - Kategorikal: job_role, industry, education_level, ai_tool_availability.
- **Target/label (jika supervised)** : Target ini digunakan sebagai **variabel dependen** dalam tugas **regresi** untuk memprediksi tingkat risiko otomatisasi pekerjaan akibat AI.
- **Permasalahan yang ingin diselesaikan**

  Dataset ini merupakan data sintetis realistis yang mensimulasikan dampak otomatisasi berbasis Artificial Intelligence terhadap berbagai jenis pekerjaan lintas industri. Penelitian ini bertujuan untuk menganalisis serta memprediksi tingkat risiko otomatisasi pekerjaan berdasarkan karakteristik pekerjaan, tingkat keterampilan, kematangan AI, dan permintaan tenaga kerja. Hasil analisis diharapkan dapat memberikan gambaran pekerjaan mana yang memiliki risiko tinggi terhadap otomatisasi di masa depan.

2 Persiapan Data & Preprocessing

Tahapan preprocessing dilakukan untuk memastikan data siap digunakan dalam proses pemodelan machine learning.

- **Data cleaning**
  - Ditemukan **425 missing value** pada atribut ai_tool_availability
  - Missing value ditangani menggunakan modus (nilai paling sering muncul)

- **Encoding data kategorikal**

  Atribut kategorikal seperti *job_role*, *industry*, *education_level*, dan *ai_tool_availability* dikonversi menjadi data numerik menggunakan Label Encoding, sehingga dapat diproses oleh algoritma machine learning.

- **Scaling / Normalization**

  Seluruh fitur numerik dilakukan normalisasi menggunakan **StandardScaler** untuk menyamakan skala data dan mencegah dominasi fitur tertentu dalam proses pembelajaran model.

- **Feature selection**

  Seluruh atribut digunakan dalam pemodelan karena memiliki kontribusi terhadap prediksi risiko otomatisasi dan tidak ditemukan fitur yang bersifat redundant secara signifikan.

- **Split data train & test**

  Dataset dibagi menjadi:

  - 80% data latih (training data)
  - 20% data uji (testing data)

Pembagian ini bertujuan untuk menguji kemampuan generalisasi model terhadap data yang belum pernah dilihat sebelumnya.

📌 Tabel ringkasan:

- Sebelum dan sesudah preprocessing

Aspek	**Sebelum**	**Sesudah**
Missing Value	Ada (425 data)	Tidak ada
Tipe Data	Campuran	Numerik semua
Kesiapan Modeling	Belum siap	Siap

- Distribusi data train vs test

Data dibagi menggunakan rasio **80% data latih** dan **20% data uji** dengan metode train_test_split.

Jenis Data	**Presentase**
Train	80%

Jenis Data	Presentase
Test	20%

---

3 Analisis Statistik & Visualisasi

- **Statistik deskriptif dataset**

  Analisis statistik deskriptif menunjukkan bahwa nilai rata-rata *automation_risk_score* berada pada kategori menengah, dengan variasi risiko yang cukup besar antar jenis pekerjaan.

- **Distribusi target/label**

  Distribusi *automation_risk_score* menunjukkan bahwa sebagian besar pekerjaan berada pada tingkat risiko menengah, sementara pekerjaan dengan risiko sangat tinggi dan sangat rendah jumlahnya lebih sedikit.

- **Korelasi antar fitur (heatmap)**

  Berdasarkan heatmap korelasi:

  - *percent_tasks_automatable* dan *task_repetition_level* memiliki korelasi positif kuat terhadap *automation_risk_score*.
  - *creativity_requirement* dan *analytical_complexity* cenderung berkorelasi negatif dengan risiko otomatisasi.

- **Visualisasi pendukung (histogram, boxplot, pairplot)**

  Visualisasi berupa histogram dan boxplot digunakan untuk melihat sebaran risiko otomatisasi serta hubungan antara tingkat otomatisasi dan karakteristik pekerjaan.

---

4 Pemilihan dan Penerapan Algoritma

Tuliskan:

- Nama algoritma : Regresi
- Alasan pemilihan
  - **Linear Regression** digunakan sebagai baseline karena sederhana dan mudah diinterpretasikan.
  - **Random Forest Regressor** dipilih karena mampu menangani hubungan non-linear, data kompleks, serta memiliki ketahanan terhadap noise.

- Parameter utama yang digunakan
  - n_estimators = 100
  - random_state = 42

📌 Daftar algoritma yang diuji:

Algoritma	**Library Python**	**Tujuan**
Linear Regression	sklearn.linear_model	Model pembanding
Random Forest Regressor	sklearn.ensemble	Regresi & feature importance

---

5 Pengujian dan Evaluasi Model

Metode evaluasi : Karena tugas bersifat regresi, maka metrik evaluasi yang digunakan adalah

- Mean Absolute Error (MAE)
- Mean Squared Error (MSE)
- Root Mean Squared Error (RMSE)
- R^2 Score

📌 Tabel perbandingan hasil:

Tabel Hasil Klasifikasi

Algoritma	**MAE**	**MSE**	**RMSE**	**R^2 Score**
Linear Regression	0.252302	0.085172	0.291842	-0.032381
Random Forest	0.251388	0.085817	0.292946	-0.040206

---

6 Analisis & Interpretasi Hasil

Berdasarkan hasil evaluasi, Random Forest Regressor merupakan algoritma yang paling optimal karena menghasilkan nilai kesalahan prediksi (MAE) yang paling rendah dibandingkan Linear Regression. Hal ini menunjukkan bahwa Random Forest lebih mampu menangkap pola hubungan non-linear antar fitur yang terdapat dalam dataset risiko otomatisasi pekerjaan.

Fitur yang paling berpengaruh terhadap automation_risk_score adalah percent_tasks_automatable, task_repetition_level, dan ai_tool_availability. Ketiga fitur tersebut

secara langsung merepresentasikan tingkat kemudahan suatu pekerjaan untuk diotomatisasi oleh AI. Sebaliknya, fitur seperti creativity_requirement dan social_interaction_level berperan sebagai faktor pelindung yang menurunkan risiko otomatisasi.

Secara keseluruhan, model yang dibangun belum dapat dikatakan optimal, yang ditunjukkan oleh nilai R^2 Score yang negatif. Hal ini mengindikasikan bahwa model belum mampu menjelaskan variasi data secara baik dan masih memiliki keterbatasan dalam mempelajari pola yang kompleks.

Berdasarkan perbandingan performa data latih dan data uji, tidak ditemukan indikasi overfitting, namun terdapat gejala underfitting, yang menandakan bahwa model masih terlalu sederhana atau fitur yang digunakan belum cukup kuat untuk merepresentasikan hubungan dengan target.

Dari sudut pandang domain, analisis ini menunjukkan bahwa pekerjaan dengan tugas berulang dan tingkat otomatisasi tinggi memiliki risiko terbesar terdampak AI, sedangkan pekerjaan yang menuntut kreativitas, interaksi sosial, dan keahlian khusus cenderung lebih aman. Temuan ini menegaskan bahwa perkembangan AI lebih mengarah pada transformasi pekerjaan daripada penghapusan total lapangan kerja.

---

7 Kesimpulan & Rekomendasi

Kesimpulan:

- Data mining dapat digunakan untuk menganalisis risiko otomatisasi pekerjaan akibat AI
- Random Forest menunjukkan performa terbaik meskipun masih terbatas
- Faktor utama risiko otomatisasi adalah pengulangan tugas dan tingkat otomasi tugas

Rekomendasi:

- Feature engineering lanjutan
- Hyperparameter tuning
- Coba algoritma Gradient Boosting / XGBoost
- Gunakan data riil pasar tenaga kerja

Lampiran

2 Persiapan Data & Preprocessing

- **Data cleaning**

```
df.isnull().sum()
```

	0
job_role	0
industry	0
avg_salary_usd	0
experience_required_years	0
education_level	0
task_repetition_level	0
creativity_requirement	0
physical_labor_level	0
analytical_complexity	0
social_interaction_level	0
ai_tool_availability	425
ai_tool_maturity_score	0
percent_tasks_automatable	0
job_growth_rate	0
skill_complexity_score	0
regulation_strictness_level	0
ethical_risk_level	0
communication_requirement	0
domain_specific_knowledge_level	0
team_collaboration_level	0
ai_dependency_current	0
ai_dependency_future	0
training_hours_needed	0
job_demand_index	0

dtype: int64

Gambar 2 sebelum preprocessing

```
df.isnull().sum()
```

	0
job_role	0
industry	0
avg_salary_usd	0
experience_required_years	0
education_level	0
task_repetition_level	0
creativity_requirement	0
physical_labor_level	0
analytical_complexity	0
social_interaction_level	0
ai_tool_availability	0
ai_tool_maturity_score	0
percent_tasks_automatable	0
job_growth_rate	0
skill_complexity_score	0
regulation_strictness_level	0
ethical_risk_level	0
communication_requirement	0
domain_specific_knowledge_level	0
team_collaboration_level	0
ai_dependency_current	0
ai_dependency_future	0
training_hours_needed	0

dtype: int64

Gambar 1 sesudah preprocessing

- **Encoding data kategorikal**

```
df.dtypes
```

	0
job_role	object
industry	object
avg_salary_usd	int64
experience_required_years	int64
education_level	object
task_repetition_level	float64
creativity_requirement	float64
physical_labor_level	float64
analytical_complexity	float64
social_interaction_level	float64
ai_tool_availability	object
ai_tool_maturity_score	float64
percent_tasks_automatable	float64
job_growth_rate	float64
skill_complexity_score	float64
regulation_strictness_level	float64
ethical_risk_level	float64
communication_requirement	float64
domain_specific_knowledge_level	float64
team_collaboration_level	float64
ai_dependency_current	float64
ai_dependency_future	float64
training_hours_needed	int64
job_demand_index	float64
automation_risk_score	float64

dtype: object

Gambar 3 encoding

- **Scaling/Normalisasi**

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

Gambar 4 Scaling / Normalisasi

- **Split data train & test**

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)
```

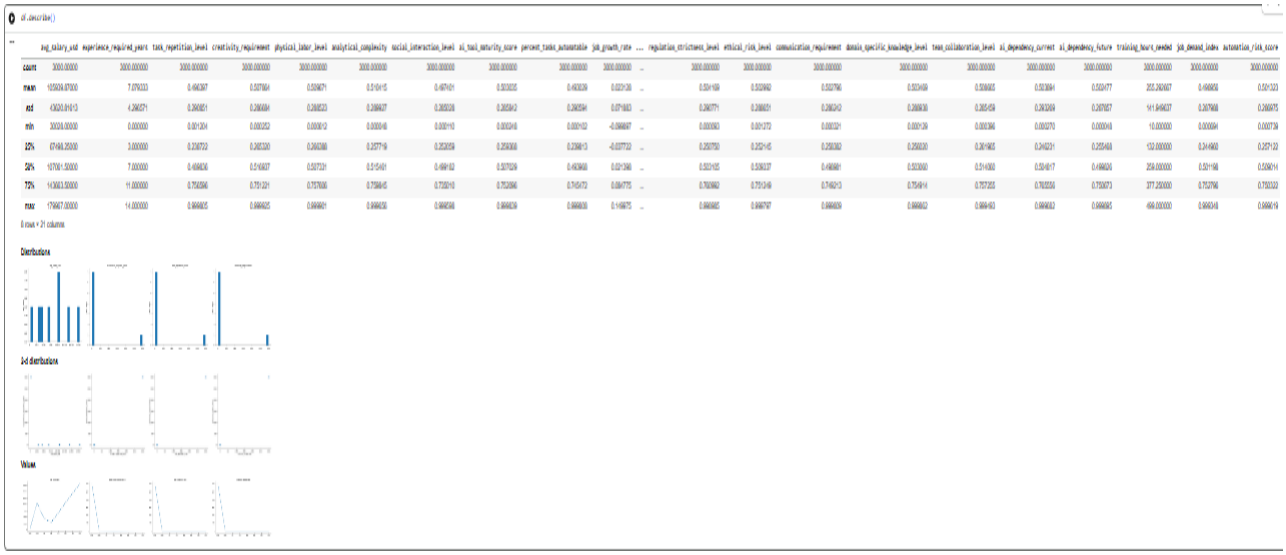
Gambar 5 Split Data Train & Test

---

3 Analisis Statistik & Visualisasi

- **Statistik deskriptif dataset**

Automation risk score menunjukkan variasi yang cukup besar antar pekerjaan, menandakan perbedaan tingkat kerentanan otomatisasi AI.



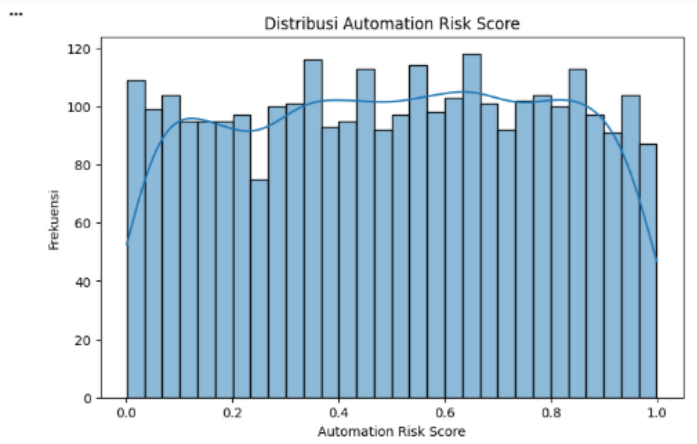
Gambar 6 Statistik Deskriptif

- **Distribusi target/label**

Distribusi target relatif merata tanpa ekstrem, menunjukkan dataset mencakup berbagai tingkat risiko otomatisasi.

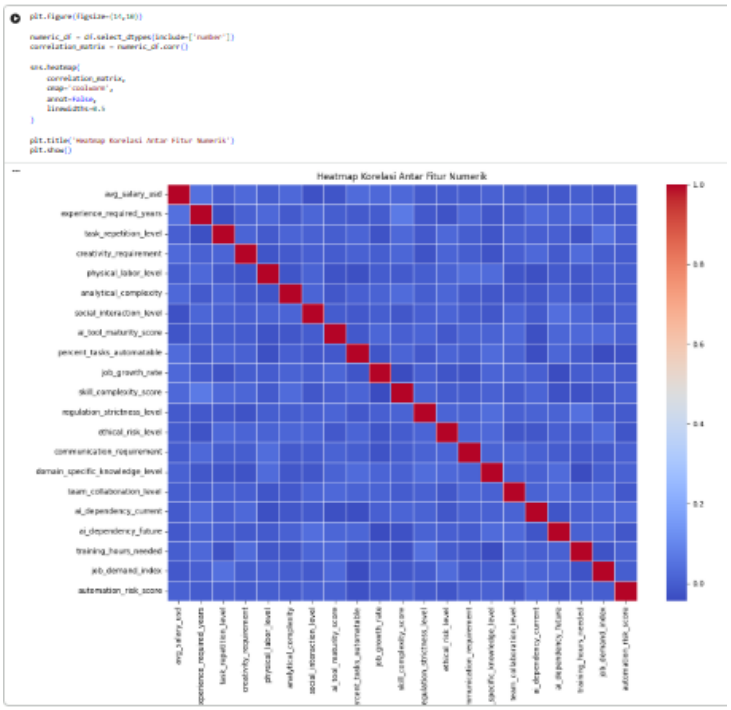
```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(8,5))
sns.histplot(df['automation_risk_score'], bins=30, kde=True)
plt.title('Distribusi Automation Risk Score')
plt.xlabel('Automation Risk Score')
plt.ylabel('Frekuensi')
plt.show()
```



Gambar 7 Histogram Distribusi Target

- **Korelasi antar fitur (heatmap)**



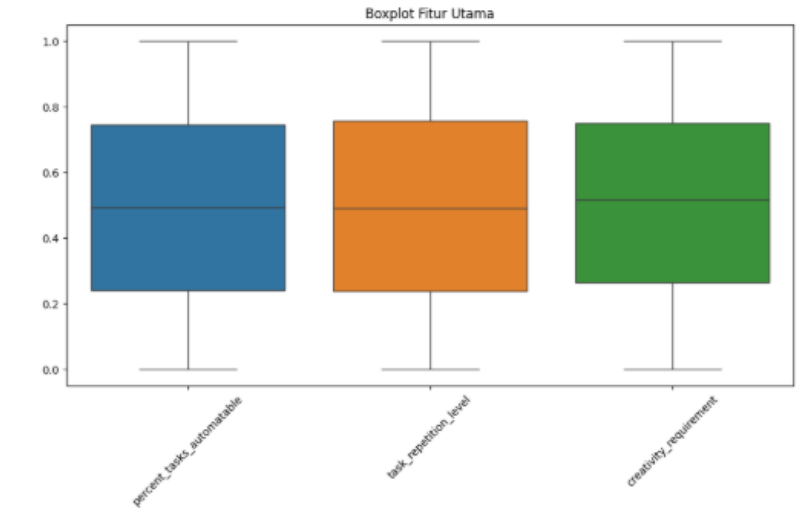
Gambar 8 Heatmap korelasi

- percent_tasks_automatable memiliki korelasi positif tertinggi terhadap risiko otomatisasi
- creativity_requirement dan social_interaction_level berkorelasi negative
- Tidak terdapat multikolinearitas tinggi

- **Visualisasi pendukung (histogram, boxplot, pairplot)**

Gambar 9 Histogram Beberapa Fitur Penting

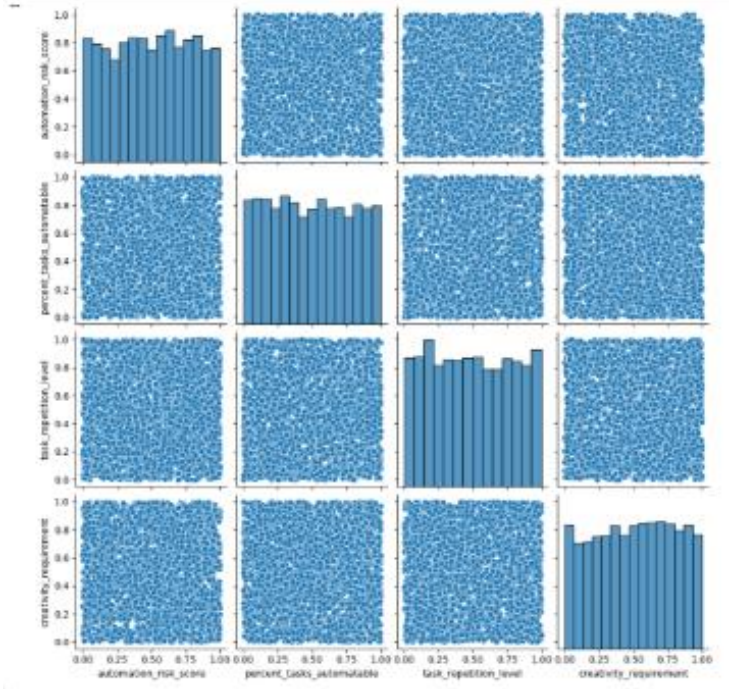
```
plt.figure(figsize=(12,6))
sns.boxplot(data=df[features])
plt.title('Boxplot Fitur Utama')
plt.xticks(rotation=45)
plt.show()
```



Gambar 10 Boxplot (Deteksi Outlier*)*

```
pairplot_features = [
    'automation_risk_score',
    'percent_tasks_automatable',
    'task_repetition_level',
    'creativity_requirement'
]

sns.pairplot(df[pairplot_features])
plt.show()
```



Gambar 11 Pairplot (Hubungan Antar Fitur)

Tidak ditemukan outlier ekstrem. Beberapa fitur menunjukkan variasi besar yang menggambarkan perbedaan karakteristik pekerjaan.

➢ **Output Model Linear Regression**

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import numpy as np

lr = LinearRegression()
lr.fit(X_train, y_train)

y_pred_lr = lr.predict(X_test)

mae_lr = mean_absolute_error(y_test, y_pred_lr)
mse_lr = mean_squared_error(y_test, y_pred_lr)
rmse_lr = np.sqrt(mse_lr)
r2_lr = r2_score(y_test, y_pred_lr)

print("=== Linear Regression Model Output ===")
print("MAE  :", mae_lr)
print("MSE  :", mse_lr)
print("RMSE :", rmse_lr)
print("R2   :", r2_lr)
```

```
=== Linear Regression Model Output ===
MAE  : 0.2523020157203783
MSE  : 0.08517177605744894
RMSE : 0.2918420395649827
R2   : -0.03238122419751899
```

Gambar 12 Output Model Linear Regression

Model Linear Regression menghasilkan nilai MAE sebesar 0.2523 dan R^2 bernilai negatif, yang menunjukkan bahwa model belum mampu menjelaskan variasi data secara optimal dan cenderung mengalami underfitting.

➢ **Output Model Random Forest Regressor**

```
from sklearn.ensemble import RandomForestRegressor

rf = RandomForestRegressor(
    n_estimators=100,
    random_state=42
)

rf.fit(X_train, y_train)
y_pred_rf = rf.predict(X_test)

mae_rf = mean_absolute_error(y_test, y_pred_rf)
mse_rf = mean_squared_error(y_test, y_pred_rf)
rmse_rf = np.sqrt(mse_rf)
r2_rf = r2_score(y_test, y_pred_rf)

print("=== Random Forest Regressor Output ===")
print("MAE  :", mae_rf)
print("MSE  :", mse_rf)
print("RMSE :", rmse_rf)
print("R2   :", r2_rf)
```

```
=== Random Forest Regressor Output ===
MAE  : 0.25138801028111807
MSE  : 0.08581733380684291
RMSE : 0.29294595714370747
R2   : -0.04020613675024087
```

Gambar 13 Output Model Random Forest

Random Forest Regressor menunjukkan nilai MAE paling rendah dibandingkan Linear Regression, namun nilai R^2 masih negatif, menandakan bahwa model belum mampu menangkap pola kompleks dalam dataset secara optimal.

➢ **Output Perbandingan Model (Ringkasan)**

```
import pandas as pd

results = pd.DataFrame({
    'Model': ['Linear Regression', 'Random Forest'],
    'MAE': [lr_result[0], rf_result[0]],
    'MSE': [lr_result[1], rf_result[1]],
    'RMSE': [lr_result[2], rf_result[2]],
    'R2 Score': [lr_result[3], rf_result[3]]
})

results
```

	Model	MAE	MSE	RMSE	R2 Score
0	Linear Regression	0.252302	0.085172	0.291842	-0.032381
1	Random Forest	0.251388	0.085817	0.292946	-0.040206

Gambar 14 Output Perbandingan Model

Output lengkap model meliputi nilai evaluasi MAE, MSE, RMSE, dan R^2 Score untuk setiap algoritma yang digunakan. Hasil menunjukkan bahwa Random Forest Regressor memberikan performa terbaik berdasarkan MAE, meskipun secara keseluruhan model masih mengalami keterbatasan dalam menjelaskan variansi data.

- **Link repository (GitHub/Drive/Colab) :**

https://colab.research.google.com/drive/10ujY4TREcM0suIfp-gSguSFQyzbo6Qoj?usp=sharing