

## Laporan Tugas 2

Link repository Github :

<https://github.com/Naufallm/Assignment-2-DIP-Pre-Processing-data-.git>

Link Kaggle dataset :

<https://www.kaggle.com/datasets/mahmoudelhemy/students-grading-dataset>

### i. Pendahuluan

Laporan ini mendokumentasikan langkah-langkah Pre-processing data yang dilakukan pada Dataset Penilaian Mahasiswa. Pipeline Pre-processing ini dirancang untuk mempersiapkan data untuk analisis.

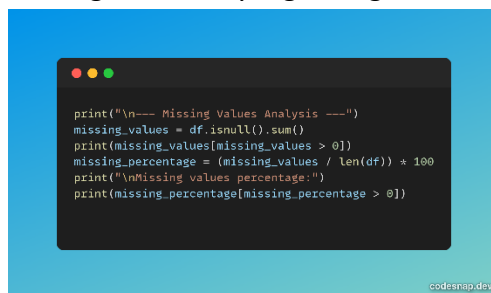
### ii. Gambaran Dataset

Dataset asli berisi 5.000 catatan mahasiswa dengan 23 kolom, termasuk:

- Identifikasi mahasiswa (Student\_ID)
- Informasi demografis (First\_Name, Last\_Name, Email, Gender, Age)
- Detail akademik (Department, Attendance, Scores, Grade)
- Kebiasaan belajar (Study\_Hours\_per\_Week)

### iii. Langkah Pre-processing yang dilakukan

#### 1. Penanganan nilai yang hilang



Nilai yang hilang ditangani menggunakan metode imputasi yang sesuai:

- Kolom numerik: Diimputasi dengan nilai median dari setiap kolom
- Kolom kategorikal: Diimputasi dengan modus (nilai yang paling sering muncul) dari setiap kolom

Pendekatan ini mempertahankan distribusi keseluruhan data sambil memastikan kelengkapan data.

## 2. Penghapusan data duplikat

```
print("\n--- Removing Duplicates ---")
duplicate_count = df.duplicated().sum()
print(f"Number of duplicate rows: {duplicate_count}")
if duplicate_count > 0:
    df.drop_duplicates(inplace=True)
    print(f"Removed {duplicate_count} duplicate rows.")
```

Data duplikat diidentifikasi dan dihapus dari dataset. Langkah ini memastikan bahwa setiap observasi dalam dataset bersifat unik, mencegah bias dalam analisis selanjutnya.

## 3. Cek konsistensi data

```
print("\n--- Checking Data Consistency ---")
# Check for consistency in categorical columns
for col in categorical_columns:
    unique_values = df[col].unique()
    print(f"Column {col}: {len(unique_values)} unique values. Consistent: {'Yes' if len(unique_values) < 10 else 'No'}
```

Pemeriksaan konsistensi dilakukan pada variabel kategorikal untuk memastikan representasi yang seragam. Temuan penting meliputi:

- Nilai Gender telah distandarisasi
- Nama Department diverifikasi untuk konsistensi
- Kategori Grade dikonfirmasi mengikuti format standar

## 4. Deteksi outlier

```
print("\n--- Outlier Detection ---")
def detect_outliers(df, column):
    Q1 = df[column].quantile(0.25)
    Q3 = df[column].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    outliers = df[(df[column] < lower_bound) | (df[column] > upper_bound)][column]
    return outliers, lower_bound, upper_bound

# Detect outliers in numeric columns
for col in numeric_columns:
    outliers, lower_bound, upper_bound = detect_outliers(df, col)
    if len(outliers) > 0:
        print(f"Column {col} has {len(outliers)} outliers")
        print(f"Outlier values: {outliers.values[:5]}..." if len(outliers) > 5 else "")
        print(f"Range: [{lower_bound}, {upper_bound}]")

# Create boxplot for columns with outliers
if len(outliers) > 0:
    plt.figure(figsize=(10, 6))
    sns.boxplot(x=col)
    plt.title(f"Boxplot of {col}")
    plt.tight_layout()
    plt.savefig(f'boxplot_{col}.png')
    plt.close()
```

Metode statistik digunakan untuk mengidentifikasi outlier dalam kolom numerik:

- Metode Interquartile Range (IQR) diterapkan
- Boxplot dibuat untuk konfirmasi visual
- Outlier diidentifikasi tetapi dipertahankan dalam dataset untuk menghindari kehilangan informasi

## 5. Normalisasi data

```
print("\n--- Normalizing and Standardizing Data ---")
# Create a copy of the dataframe before scaling
df_scaled = df.copy()

# Min-Max Scaling (0-1 range)
print("Applying Min-Max scaling to numeric columns...")
scaler = MinMaxScaler()
df_scaled[numeric_columns] = scaler.fit_transform(df[numeric_columns])

# Also create a standardized version (z-score)
df_standardized = df.copy()
print("Applying Z-score standardization to numeric columns...")
std_scaler = StandardScaler()
df_standardized[numeric_columns] = std_scaler.fit_transform(df[numeric_columns])
```

Dua transformasi terpisah diterapkan pada fitur numerik:

- **Normalisasi Min-Max:** Menskalakan semua fitur numerik ke rentang [0,1]
- **Standardisasi Z-score:** Mentransformasi fitur untuk memiliki mean=0 dan standar deviasi=1

- **Hasil**

1. **students\_cleaned.csv:** Berisi dataset setelah pembersihan dasar (penanganan nilai hilang, penghapusan duplikat, pemeriksaan konsistensi)
2. **students\_normalized.csv:** Berisi dataset yang sudah dibersihkan dengan normalisasi Min-Max diterapkan
3. **students\_standardized.csv:** Berisi dataset yang sudah dibersihkan dengan standardisasi Z-score diterapkan

- **Perubahan dalam karakteristik data**

| karakteristik  | Sebelum pre-Processing       | Setelah Pre-Processing         |
|----------------|------------------------------|--------------------------------|
| Dimensions     | 5,000 rows × 23 columns      | 5,000 rows × 23 columns        |
| Missing Values | Terdapat di beberapa kolom   | Tidak ada                      |
| Duplicates     | Tidak terdapat duplikat data | Tidak ada Tindakan penghapusan |
| Scale          | Skala campuran               | Dinormalisasi [0,1]            |

- **Kesimpulan**

Langkah-langkah pre-processing telah berhasil mengubah Dataset Penilaian Mahasiswa mentah menjadi format yang bersih, konsisten, dan terskala. Ketiga dataset yang telah diproses tersedia untuk berbagai jenis analisis:

1. **students\_cleaned.csv** untuk analisis data eksplorasi yang membutuhkan skala asli
2. **students\_normalized.csv** untuk algoritma yang sensitif terhadap rentang variable
3. **students\_standardized.csv** untuk algoritma yang mengasumsikan data terdistribusi normal