

# PRAKTIKUM MACHINE LEARNING

## UNIT 5 : DATA CLEANING



`adult_klasifikasi.ipynb`

Eka Praja Wiyata Mandala, S.Kom, M,Kom, CADS

# Unit 5 : Data Cleaning

2

## ✓ Unit 5: Membersihkan Data

Tujuan: Melakukan pembersihan data untuk meningkatkan kualitas dataset sebelum analisis dan pemodelan lebih lanjut.

---

# Unit 5 : Data Cleaning

## ✓ Menangani Missing Values

✓  
0s

```
# Cek missing values
print("Missing values sebelum pembersihan:")
print(df.isnull().sum())

# Menangani missing values
for column in df.columns:
    if df[column].dtype == 'object':
        # Untuk kolom kategorikal, isi dengan modus
        df[column].fillna(df[column].mode()[0], inplace=True)
    else:
        # Untuk kolom numerik, isi dengan median
        df[column].fillna(df[column].median(), inplace=True)

print("\nMissing values setelah pembersihan:")
print(df.isnull().sum())
```

# Unit 5 : Data Cleaning

4

## ▼ Menangani Outlier

✓  
3s

```
def plot_boxplot(df, column):  
    plt.figure(figsize=(10, 6))  
    sns.boxplot(x=df[column])  
    plt.title(f'Boxplot of {column}')  
    plt.show()  
  
# Contoh untuk kolom numerik  
numeric_columns = df.select_dtypes(include=[np.number]).columns
```

# Unit 5 : Data Cleaning

```
for column in numeric_columns:
    plot_boxplot(df, column)

    # Menangani outlier dengan IQR method
    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

    df[column] = np.where(df[column] > upper_bound, upper_bound,
                          np.where(df[column] < lower_bound, lower_bound, df[column]))

    print(f"Outliers pada {column} sudah dihandel.")
    plot_boxplot(df, column)
```

# Unit 5 : Data Cleaning

6

Contoh salah satu Penjelasan untuk Feature **age**

Terlihat bahwa **age** sebelum dibersihkan terdapat outlier pada **diatas angka 78**.

Setelah dibersihkan, outlier sudah hilang sehingga Box Plotnya menampilkan hasil yang berbeda antara sebelum dan sesudah dibersihkan.

**Lanjutkan menjelaskan untuk semua Box Plot nya**

✓  
0s



```
df.info()
```



# Unit 5 : Data Cleaning

7

## ✓ Menangani Duplikat

```
✓ [29] # Cek duplikat  
0s      duplicate_count = df.duplicated().sum()  
      print(f"Jumlah baris duplikat: {duplicate_count}")  
  
      # Hapus duplikat  
      df.drop_duplicates(inplace=True)  
  
      print(f"Jumlah baris setelah menghapus duplikat: {len(df)}")
```

Terdapat **duplikasi** data sebanyak **53 record**, sehingga perlu dihapus.


Setelah dilakukan penghapusan record, maka tersisa sebanyak **32508 record** yang **tidak ada duplikasi**

```
✓ df.info()  
0s
```

# Unit 5 : Data Cleaning

8

## ✓ Validasi Hasil Pembersihan

✓  
0s  df.head()

## ✓ Menyimpan dataset yang sudah dibersihkan

✓  
0s [32] # Simpan data yang telah dibersihkan  
sys.path.append(f'{folder\_name}')

```
df.to_csv(f'{folder_name}/adult_income_cleaned.csv', index=False)  
print("Data yang telah dibersihkan telah disimpan sebagai 'adult_income_cleaned.csv'")
```