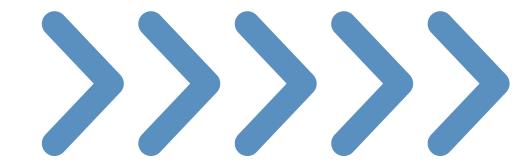


# PROYEK EXPLORATORY DATA ANALYSIS



Portfolio Data Analyst  
Data Cleaning

11 Mei 2025



# MATERI PRESENTASI >>>>

01 Latar Belakang

Pengecekan Deskripsi Data

02 Tools Dan Teknologi

Handling & Verifikasi  
Missing Value

03 Pengecekan Missing Value

Pengecekan Duplikat

07 Kesimpulan

04

05

06

# LATAR BELAKANG



- **Tujuan Project:** Membersihkan data dari missing values (nilai kosong) dan duplikat untuk memastikan analisis akurat.
- **SumberDataset:**  
<https://www.kaggle.com/datasets/jayaantanaath/student-habits-vs-academic-performance?resource=download>
- **Dataset:**  
student\_habits\_performance.csv(1.000 data siswa, 16 variabel)

## Mengapa Penting?

Data yang kotor (missing/duplikat) bisa menyebabkan kesalahan dalam pengambilan keputusan suatu organisasi



# TOOLS & TEKNOLOGI

Bahasa : Phyton

1

Libraries : Pandas (untuk memanipulasi data)

2

Metode:

- Pengecekan dan Pengisian missing values (modus/mean).
- Pengecekan duplikat.

3



# PENGECEKAN MISSING VALUE

```
#Mengimpor dataset csv ke Phyton  
import pandas as pd  
data = pd.read_csv('/content/student_habits_performance.csv')  
  
#Mengecek Data yang Hilang (Missing Value)  
#Cara 1  
data.info()  
  
#Mengecek Data yang Hilang (Missing Value)  
#Cara 2  
data.isna().sum()
```

Penjelasan: Setelah dilakukan pengecekan dengan cara satu dan dua, hanya 1 kolom yaitu `parental_education_level` (tingkat pendidikan orang tua) yang memiliki 91 missing values / nilai kosong sedangkan Kolom lain sudah lengkap

## Output Cara 1

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 1000 entries, 0 to 999  
Data columns (total 16 columns):  
 #   Column           Non-Null Count  Dtype     
---  --  
 0   student_id      1000 non-null    object    
 1   age              1000 non-null    int64     
 2   gender           1000 non-null    object    
 3   study_hours_per_day  1000 non-null  float64  
 4   social_media_hours 1000 non-null  float64  
 5   netflix_hours    1000 non-null  float64  
 6   part_time_job    1000 non-null  object    
 7   attendance_percentage 1000 non-null  float64  
 8   sleep_hours      1000 non-null  float64  
 9   diet_quality     1000 non-null  object    
 10  exercise_frequency 1000 non-null  int64     
 11  parental_education_level 909 non-null  object    
 12  internet_quality 1000 non-null  object    
 13  mental_health_rating 1000 non-null  int64     
 14  extracurricular_participation 1000 non-null  object    
 15  exam_score       1000 non-null  float64  
dtypes: float64(6), int64(3), object(7)  
memory usage: 125.1+ KB  
0s completed at 9:13PM
```

## Output Cara 2

student_id	0
age	0
gender	0
study_hours_per_day	0
social_media_hours	0
netflix_hours	0
part_time_job	0
attendance_percentage	0
sleep_hours	0
diet_quality	0
exercise_frequency	0
parental_education_level	91
internet_quality	0
mental_health_rating	0
extracurricular_participation	0
exam_score	0

# PENGECEKAN DESKRIPSI DATA

```
✓ 0s  #Cari Statistik Dari Data Data Set  
data.describe()  
  
age  study_hours_per_day  social_media_hours  net
```

	age	study_hours_per_day	social_media_hours	netflix_hours	attendance_percentage	sleep_hours	exercise_frequency	mental_health_rating	exam_score
count	1000.0000	1000.00000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000
mean	20.4980	3.55010	2.505500	1.819700	84.131700	6.470100	3.042000	5.438000	69.601500
std	2.3081	1.46889	1.172422	1.075118	9.399246	1.226377	2.025423	2.847501	16.888564
min	17.0000	0.00000	0.000000	0.000000	56.000000	3.200000	0.000000	1.000000	18.400000
25%	18.7500	2.60000	1.700000	1.000000	78.000000	5.600000	1.000000	3.000000	58.475000
50%	20.0000	3.50000	2.500000	1.800000	84.400000	6.500000	3.000000	5.000000	70.500000
75%	23.0000	4.50000	3.300000	2.525000	91.025000	7.300000	5.000000	8.000000	81.325000
max	24.0000	8.30000	7.200000	5.400000	100.000000	10.000000	6.000000	10.000000	100.000000

Penjelasan: Setelah dilakukan pengecekan deskripsi data, secara keseluruhan, nilai minimum dan maksium masuk akal untuk setiap kolom dan mean mendekati median taitu 50% disetiap kolo yang menunjukkan distribusi normal

# PENGISIAN & VERIFIKASI MISSING VALUE

## Pengisian Missing Value

```
for column in data.columns:  
    #jika kolom tipe objek isi dengan modus  
    if data[column].dtype == 'object' :  
        data[column].fillna(data[column].mode()[0], inplace=True)  
    else :  
        #jika kolom tipe numerik isi dengan mean  
        data[column].fillna(data[column].mean(), inplace=True)  
  
<ipython-input-43-b0d141ac9ebe>:4: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through  
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we  
For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[  
  
    data[column].fillna(data[column].mode()[0], inplace=True)  
<ipython-input-43-b0d141ac9ebe>:7: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through  
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we  
For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[
```

untuk kolom dengan nilai kosong tipe object maka diisi dengan nilai modus, sedangkan kolom dengan nilai kosong tipe numerik diisi dengan nilai mean dan perubahan langsung disimpan kedata. Setelah dilakukan pengecekan missing value kembali, menunjukkan tidak ada lagi kolom dengan nilai kosong

## Verifikasi Missing Value

```
data [column].fillna(data[column].mean(), inplace=True)  
  
4] # Mengecek apakah masih ada missing values di dataset  
print("Total missing values setelah imputasi:")  
print(data.isnull().sum())  
  
Total missing values setelah imputasi:  
student_id          0  
age                 0  
gender              0  
study_hours_per_day 0  
social_media_hours  0  
netflix_hours       0  
part_time_job        0  
attendance_percentage 0  
sleep_hours          0  
diet_quality         0  
exercise_frequency   0  
parental_education_level 0  
internet_quality    0  
mental_health_rating 0  
extracurricular_participation 0  
exam_score           0  
dtype: int64  
  
5] #Mengecek apakah ada data duplikat
```

# PENGECEKAN DUPLIKASI DATA

```
dtype: int64

#Mengecek apakah ada data duplikat
check_duplicate = data.duplicated().sum()
print(f"Jumlah data duplikat = {check_duplicate}")

→ Jumlah data duplikat = 0
```

Penjelasan: Setelah dilakukan pengecekan apakah terdapat data yang duplikat disetiap kolom, hasilnya menunjukkan tidak terdapat data yang duplikat. Ini menunjukkan bahwa data set telah bersih dan bisa dilanjut untuk visualisasi data



# KESIMPULAN

Setelah melalui proses pembersihan data, dataset kini telah bersih dari nilai yang hilang dan duplikat, sehingga siap untuk analisis lebih lanjut. Langkah ini sangat penting karena data yang bersih akan menghasilkan analisis yang lebih akurat dan kesimpulan yang lebih dapat diandalkan. Dengan memastikan setiap nilai terisi (tidak missing value) dengan benar dan tidak ada data yang berulang (duplikat), saya telah meningkatkan kualitas dataset, korelasi, dan bahkan prediksi yang lebih tepat. Ini adalah dasar penting dalam setiap analisis data proyek.

# PROFIL ANALYST

Rozalinda Titalia Putri

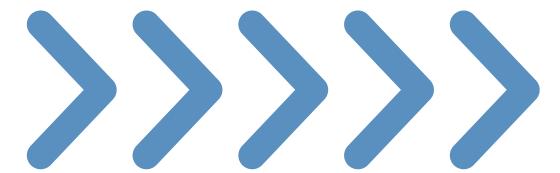
 surabaya, Jawa Timur

 lindaroza509@gmail.com

 RozalindaPutri

Saya adalah mahasiswa Sistem Informasi semester 2 yang memiliki semangat tinggi dibidang IT dan data Analyst. Sebagai calon data Analyst, saya sangat antusias untuk menerapkan kemampuan analisis dan pengetahuan saya dalam menyelesaikan masalah terutama dibidang data.





# TERIMA KASIH

Terbuka untuk kolaborasi proyek data lebih lanjut

