



Sesi 1

Pandas Fundamental

Hari 1 - 120 menit

Bootcamp Data Science

Oleh: Kurnia Ramadhan, ST., M.Eng

Tahun: 2025



Objektif Sesi

Setelah sesi ini, Anda dapat:

- Memuat & mengeksplorasi data dengan Pandas
- Menyeleksi kolom & baris
- Memfilter data dengan boolean indexing
- Melakukan agregasi dengan `groupby`
- Menangani missing values
- Membuat statistik ringkasan



Apa itu Pandas?

Pandas = Python Data Analysis Library

- Struktur data: **DataFrame & Series**
- Operasi: filter, merge, agregasi, reshape
- Integrasi: NumPy, Matplotlib, SQL
- Performa: Optimal untuk data besar

Kasus penggunaan: Pembersihan, analisis, dan transformasi data



Konsep Inti

DataFrame

```
# Struktur tabel dengan baris & kolom
df = pd.DataFrame({
    'nama': ['Alice', 'Bob', 'Charlie'],
    'umur': [25, 30, 35],
    'kota': ['Jakarta', 'Bandung', 'Surabaya']
})
```

Series

```
# Satu kolom dari DataFrame
ages = df['umur'] # Series
```

1 Memuat Data

```
import pandas as pd

# Load dari Parquet
df = pd.read_parquet('data.parquet')

# Load dari CSV
df = pd.read_csv('data.csv')

# Load dari Excel
df = pd.read_excel('data.xlsx')
```

Format Parquet:

- Penyimpanan kolom (columnar)
- Terkompresi & cepat
- Standar industri

2 Eksplorasi Data

```
# 5 baris pertama  
df.head()  
  
# 5 baris terakhir  
df.tail()  
  
# Dimensi (rows, columns)  
df.shape  
  
# Tipe data & info null  
df.info()  
  
# Statistik ringkas  
df.describe()
```



Struktur Dataset RUP

```
df.shape # (16430, 48)

# Kolom kunci:
- nama_paket          # Nama paket pengadaan
- pagu                 # Nilai (Rupiah)
- metode_pengadaan    # Tender, E-Purchasing, dll
- jenis_pengadaan     # Barang, Jasa, Konstruksi
- nama_satker          # Satuan Kerja
- status_pdn            # PDN / Non-PDN
- status_ukm             # UKM / Non-UKM
```

3 Memilih Kolom

```
# Satu kolom (Series)
nama_paket = df['nama_paket']

# Banyak kolom (DataFrame)
subset = df[['nama_paket', 'pagu', 'metode_pengadaan']]

# 5 kolom pertama
df.iloc[:, :5]
```

Best Practice: Pilih hanya kolom yang diperlukan untuk efisiensi

4 Memilih Baris

```
# Berdasarkan posisi index (iloc)
first_10 = df.iloc[0:10]
last_5 = df.iloc[-5:]

# Berdasarkan label (loc)
specific_rows = df.loc[0:4, ['nama_paket', 'pagu']]

# Sampel acak
sample = df.sample(100)
```

5 Memfilter Data

Boolean Indexing

```
# Kondisi tunggal
high_value = df[df['pagu'] > 1_000_000_000]

# Banyak kondisi (AND)
filtered = df[
    (df['pagu'] > 1_000_000_000) &
    (df['metode_pengadaan'] == 'Tender')
]

# Banyak kondisi (OR)
filtered = df[
    (df['metode_pengadaan'] == 'Tender') |
    (df['metode_pengadaan'] == 'Seleksi')
]
```



Metode Query

Sintaks alternatif yang lebih mudah dibaca:

```
# Menggunakan query()
result = df.query('pagu > 1_000_000_000')

# Dengan variabel
threshold = 1_000_000_000
result = df.query('pagu > @threshold')

# Kondisi kompleks
result = df.query(
    'pagu > 1e9 and metode_pengadaan == "Tender"'
)
```

6 GroupBy & Agregasi

```
# Groupby sederhana
pagu_per_metode = df.groupby('metode_pengadaan')['pagu'].sum()

# Banyak agregasi
stats = df.groupby('metode_pengadaan')['pagu'].agg([
    'count',
    'sum',
    'mean',
    'median'
])

# Agregasi kustom
stats = df.groupby('metode_pengadaan')['pagu'].agg(
    total_miliar=lambda x: x.sum() / 1e9,
    rata_juta=lambda x: x.mean() / 1e6
)
```

Agregasi Umum

Fungsi	Tujuan
count()	Jumlah baris
sum()	Total nilai
mean()	Rata-rata
median()	Nilai tengah
min()	Nilai minimum
max()	Nilai maksimum
std()	Standard deviation

7 Mengurutkan Data

```
# Sortir dengan satu kolom
df_sorted = df.sort_values('pagu', ascending=False)

# Sortir dengan beberapa kolom
df_sorted = df.sort_values(
    ['metode_pengadaan', 'pagu'],
    ascending=[True, False]
)

# Ambil top N
top_10 = df.nlargest(10, 'pagu')
bottom_10 = df.nsmallest(10, 'pagu')
```

8 Missing Values

```
# Cek missing values  
df.isnull().sum()  
  
# Drop baris yang ada null  
df_clean = df.dropna()  
  
# Drop baris dengan null di kolom tertentu  
df_clean = df.dropna(subset=['pagu'])  
  
# Isi nilai yang hilang  
df_filled = df.fillna(0)  
df_filled = df.fillna({'pagu': 0, 'metode': 'Unknown'})
```



Value Counts

```
# Hitung nilai unik  
metode_counts = df['metode_pengadaan'].value_counts()  
  
# Dengan persentase  
metode_pct = df['metode_pengadaan'].value_counts(normalize=True)  
  
# Nilai teratas  
top_5_metode = df['metode_pengadaan'].value_counts().head(5)
```



Statistik Ringkasan

```
# Untuk satu kolom  
df['pagu'].describe()  
  
# Persentil kustom  
df['pagu'].describe(percentiles=[.25, .5, .75, .9, .95])  
  
# Perhitungan manual  
print(f"Mean: {df['pagu'].mean()}")  
print(f"Median: {df['pagu'].median()}")  
print(f"Std Dev: {df['pagu'].std()}")  
print(f"Min: {df['pagu'].min()}")  
print(f"Max: {df['pagu'].max()}")
```



Contoh Praktis

Tugas: Analisis paket pengadaan dengan pagu > 1M

```
# Filter
high_value = df[df['pagu'] > 1_000_000_000]

# Group & agregasi
summary = high_value.groupby('metode_pengadaan').agg({
    'pagu': ['count', 'sum', 'mean']
})

# Sortir
summary = summary.sort_values(('pagu', 'sum'), ascending=False)

# Format
summary.columns = ['Jumlah', 'Total', 'Rata-rata']
summary['Total_M'] = summary['Total'] / 1e9
```



Praktik Terbaik

- ✓ Selalu cek **shape & info** setelah load data
- ✓ Gunakan **nama variabel yang jelas**
- ✓ **Chain operations** untuk kode yang bersih
- ✓ Copy **DataFrame** saat memodifikasi: `df.copy()`
- ✓ Gunakan **query()** untuk filter kompleks
- ✓ Tambahkan **komentar seperlunya** untuk kejelasan
- ✗ Jangan memodifikasi **DataFrame original**
- ✗ Hindari loop jika bisa vectorized

⚡ Tips Performa

```
# Buruk (lambat)
for i in range(len(df)):
    df.loc[i, 'new_col'] = df.loc[i, 'pagu'] / 1e9

# Baik (cepat - vectorized)
df['new_col'] = df['pagu'] / 1e9

# Gunakan inplace untuk efisiensi memori
df.drop(columns=['col'], inplace=True)

# Gunakan categorical untuk string berulang
df['metode'] = df['metode'].astype('category')
```



Latihan Praktik

Di Jupyter Notebook:

1. Load dataset RUP
2. Eksplorasi: shape, kolom, info
3. Filter: paket dengan pagu > 500 juta
4. GroupBy: total pagu per metode pengadaan
5. Top 10: satker dengan total pagu terbesar
6. Export: hasil analisis ke CSV

Waktu: 30 menit



Ringkasan Inti

- ✓ Pandas = pisau Swiss Army untuk analisis data
- ✓ DataFrame = struktur mirip tabel
- ✓ Filtering dengan boolean indexing atau query()
- ✓ GroupBy untuk agregasi & ringkasan
- ✓ Selalu cek & tangani missing values
- ✓ Vectorization > Loops

Referensi

- **Pandas Docs:** <https://pandas.pydata.org/docs/>
- **Cheat Sheet:** https://pandas.pydata.org/Pandas_Cheat_Sheet.pdf
- **10 Minutes to Pandas:** https://pandas.pydata.org/docs/user_guide/10min.html



Waktunya Istirahat!

Selanjutnya: Sesi 2 - DuckDB & Visualisasi