

Cheatsheet 2

Oke, ini **cheat-sheet Python (pandas)** buat **subset, grouping, dan filtering** — dari **paling dasar sampai yang sering dipakai di data real**. Contohnya aku sesuaikan dengan data kamu (provinsi, kab, gender, dll).

1. SUBSET (ambil sebagian data)

a) Subset kolom

```
df[["provinsi", "kab", "gender_ketua_kelompok"]]
```

b) Subset baris (index-based)

```
df.iloc[:10] # 10 baris pertama  
df.iloc[5:20] # slice
```

```
df.loc[100:200] # berdasarkan index label
```

c) Subset baris + kondisi

```
df[df["provinsi"] == "JAWA BARAT"]
```

2 FILTER (kondisi logika)

a) Filter satu kondisi

```
df[df["gender_ketua_kelompok"] == "L"]
```

b) Multiple kondisi (&, |)

```
df[(df["provinsi"] == "JAWA BARAT") &  
(df["gender_ketua_kelompok"] == "P")]
```

⚠ WAJIB pakai tanda kurung!

c) Filter dengan isin

```
df[df["provinsi"].isin(["JAWA BARAT", "JAWA TENGAH"])]
```

d) Filter string (contains, startswith)

```
df[df["nama_kelompok"].str.contains("NELAYAN", case=False, na=False)]
```

e) Filter angka

```
df[df["skor_kelompok"] >= 80]
```

3 □ GROUPING (ringkas data)

a) Group by satu kolom

```
df.groupby("provinsi").size()
```

Sama dengan:

```
df["provinsi"].value_counts()
```

b) Group + agregasi

```
df.groupby("provinsi")["nama_kelompok"].nunique()
```

c) Group by multiple kolom

```
df.groupby(["provinsi", "gender_ketua_kelompok"]).size()
```

d) Banyak agregasi sekaligus

```
df.groupby("provinsi").agg(  
    total_kelompok=("nama_kelompok", "count"),  
    unik_kelompok=("nama_kelompok", "nunique"),  
    rata_skor=("skor_kelompok", "mean"),  
)
```

4 SUBSET + GROUP + FILTER (REAL CASE)

□ Jumlah kelompok per provinsi, hanya yang > 1000

```
(  
df.groupby("provinsi")  
  .size()  
  .reset_index(name="jumlah")  
  .query("jumlah > 1000")  
  .sort_values("jumlah", ascending=False)  
)
```

□ Distribusi gender per provinsi

```
(  
df.groupby(["provinsi", "gender_ketua_kelompok"])  
  .size()  
  .unstack(fill_value=0)  
)
```

□ Kelompok perempuan di Jawa Barat

```
df[  
  (df["provinsi"] == "JAWA BARAT") &  
  (df["gender_ketua_kelompok"] == "P")  
][["nama_kelompok", "kab", "desa_kelompok"]]
```

5 FILTER DENGAN `query()` (lebih readable)

```
df.query("provinsi == 'JAWA BARAT' and gender_ketua_kelompok == 'P'")
```

Dengan `in`:

```
df.query("provinsi in ['JAWA BARAT','JAWA TENGAH'])"
```

6 GROUP + FILTER (HAVING-style)

Contoh: provinsi dengan ≥ 100 kelompok perempuan

```
g = df[df["gender_ketua_kelompok"] == "P"]
(
  g.groupby("provinsi")
  .size()
  .loc[lambda s: s >= 100]
)
```

7 Best practice (penting banget)

✓ Jangan lupa normalisasi dulu

```
df["provinsi"] = df["provinsi"].str.strip().str.upper()
```

```
df["gender_ketua_kelompok"] = df["gender_ketua_kelompok"].str.strip().str.upper()
```

✓ Hindari chained indexing

□

```
df[df["provinsi"] == "JAWA BARAT"]["kab"]
```

✓

```
df.loc[df["provinsi"] == "JAWA BARAT", "kab"]
```

TL;DR (pola paling sering)

```
df.loc[ df["kolom"].isin([...]), ["kolom1", "kolom2"] ]
```

```
df.groupby([...]).agg(...)
```

```
df.query("kondisi")
```

Revision #1

Created 23 January 2026 04:15:55 by Sandi Wibowo

Updated 23 January 2026 04:17:00 by Sandi Wibowo