Pandas-part-02

a. Seleksi & Ubah Data dengan .loc & .iloc @



.loc - Label-based 🦪

- **Sintaks dasar:** mie.loc[row_labels, col_labels]
- Penjelasan: Mengacu langsung ke label baris dan kolom. Ideal untuk DataFrame dengan index atau nama kolom yang bermakna (misal tanggal, nama kategori).
- Catatan penting: Range label inklusif (ujung termasuk).
- Contoh Penggunaan:

```
# Ambil baris dengan label 0, 1, dan 5
subset1 = mie.loc[[0, 1, 5]]
# Ambil baris label 0 sampai 2, termasuk 2
subset2 = mie.loc[0:2]
# Ambil baris label 0 sampai 2, dan kolom 'hari' & 'tipe_mie'
subset3 = mie.loc[0:2, ['hari', 'tipe_mie']]
```

Mengubah Nilai:

```
# Ubah sel terjual di baris label 1 menjadi 10
mie.loc[1, 'terjual'] = 10
# Ubah banyak sel dengan kondisi:
mie.loc[mie['tipe_mie'] == 'mie bangladesh', 'terjual'] = 0
```

Kapan Gunakan .loc ?

- Saat index bukan sekadar angka (misal tanggal, ID unik).
- Untuk seleksi baris berdasarkan kondisi (boolean indexing).
- Untuk data time series (index datetime).

.iloc - Position-based ?

- Sintaks dasar: mie.iloc[row_positions, col_positions]
- Penjelasan: Mengacu ke posisi baris dan kolom (0-based).
- Catatan penting: Range .iloc[a:b] bersifat setengah-terbuka (a ≤ idx < b).

Contoh Penggunaan:

```
# Ambil baris pos 0 dan 1; kolom pos 0 & 2
sub = mie.iloc[0:2, [0, 2]]
```

Mengubah Nilai:

```
# Ubah baris ke-2, kolom ke-1 menjadi 15
mie.iloc[2, 1] = 15
```

- Kapan Gunakan .iloc ?
 - Saat ingin mengambil baris/kolom berdasarkan urutan tampilan.
 - Untuk looping berdasarkan indeks integer.
 - Saat index diperlakukan sebagai integer biasa, bukan label.

b. Akses Sel Tunggal: .at , .iat & Atribut Kolom 🔦



Method	Basis Seleksi	Keunggulan	Mirip dengan
.at	Label baris & kolom	☆ Super cepat (single value)	mie.loc[label, label]
.iat	Posisi integer	☆ Super cepat (single value)	<pre>mie.iloc[pos, pos]</pre>
Atribut	Akses kolom Pythonic	Ringkas (dot/kurung)	mie['col'] atau mie.col

Contoh .at (by label):

```
nilai_at = mie.at[0, 'tipe_mie']
                                  # → ambil nilai
mie.at[0, 'terjual'] = 55
                                    # ubah sel menjadi 55
```

Contoh .iat (by posisi):

```
nilai_iat = mie.iat[1, 0]
                                          \# \rightarrow ambil baris pos 1, kol pos 0
mie.iat[2, 2] = 65
                                          # ubah sel di (2,2) jadi 65
```

Contoh Akses Atribut Kolom:

```
kolom = mie['tipe_mie']
```

```
kolom2 = mie.terjual
```

- Tips & Trik:
 - .at / .iat hanya untuk single value sangat cepat tapi tidak mendukung slicing.
 - Akses atribut (mie.nama_kolom) tidak bekerja jika nama kolom mengandung spasi atau karakter khusus.

c. Menetapkan Kolom sebagai Index 🥠



- Mengapa Index Penting?
 - Memudahkan seleksi baris berdasarkan label (misal time series).
 - Mendukung operasi join, grup, serta multi-index (hierarki).
- Buat DataFrame baru dengan index khusus:

```
mie_indexed = mie.set_index('hari') # DataFrame baru, index = kolom 'hari'
```

Ubah DataFrame asli tanpa salinan:

```
mie.set_index('hari', inplace=True) # modifikasi langsung mie
```

Reset index ke default (numerik):

Contoh Kasus Time Series ::

```
mie['tanggal'] = pd.to_datetime(mie['tanggal'])
mie.set_index('tanggal', inplace=True)
mei2025 = mie.loc['2025-05']  # ambil data penjualan Mei 2025
```