Mengiris & Mengindeks: Indeks Eksplisit

Sejauh ini, kita selalu menggunakan indeks *default* dari Pandas, yaitu angka yang dimulai dari 0 (0, 1, 2, ...). Meskipun berguna, seringkali lebih mudah jika kita menggunakan salah satu kolom data sebagai **indeks eksplisit** atau **indeks kustom**. Ini membuat proses pemilihan data (*subsetting*) menjadi lebih intuitif, terutama jika Anda memiliki pengenal unik untuk setiap baris.

Mengatur dan Mereset Indeks

Ada dua metode utama untuk ini:

1. **`.set_index("nama_kolom")`**: Mengubah kolom yang ada menjadi indeks DataFrame.
2. **`.reset_index()`**: Mengembalikan indeks menjadi *default* (angka 0, 1, 2, ...) dan mengubah indeks kustom yang lama menjadi kolom biasa.
    - **`drop=True`**: Jika Anda ingin menghapus indeks kustom sepenuhnya tanpa mengubahnya menjadi kolom, gunakan argumen `.reset_index(drop=True)`.

Contoh:

```
# Mengatur 'name' sebagai indeks
dogs_ind = dogs.set_index("name")

# Mengembalikan ke indeks default, 'name' menjadi kolom lagi
dogs_ind.reset_index()
```

Keuntungan Indeks Kustom

Keuntungan utamanya adalah kemudahan dalam memilih data menggunakan `.loc[]`, karena Anda bisa menggunakan nilai yang lebih deskriptif daripada hanya angka.

```
# Cara lama dengan filtering
dogs[dogs["name"].isin(["Bella", "Stella"])]

# Cara baru yang lebih ringkas dengan indeks kustom
dogs_ind.loc[["Bella", "Stella"]]
```

Indeks Multi-Level

Anda bahkan bisa mengatur **beberapa kolom** sebagai indeks. Ini disebut *multi-level index* atau *hierarchical index*. Ini sangat berguna untuk data yang memiliki struktur bertingkat (misalnya, data penjualan per toko per tanggal).

Contoh:

```
# Mengatur 'breed' dan 'color' sebagai indeks
dogs_ind3 = dogs.set_index(["breed", "color"])

# Memilih berdasarkan level luar (breed)
dogs_ind3.loc[["Labrador", "Chihuahua"]]

# Memilih berdasarkan kombinasi level luar dan dalam (breed
dan color)
dogs_ind3.loc[[("Labrador", "Brown"), ("Chihuahua", "Tan")]]
```

Latihan Singkat

Gunakan `df_penjualan` kita.

1. Atur kolom '**Tanggal**' sebagai indeks dari `df_penjualan`. Simpan hasilnya ke variabel baru bernama `df_tgl`.
2. Gunakan `.loc[]` pada `df_tgl` untuk memilih semua transaksi yang terjadi pada tanggal '**2024-07-03**'.

Mengiris & Mengindeks: Slicing DataFrames

Slicing adalah teknik untuk memilih sekelompok data berdasarkan rentang posisinya. Jika Anda sudah terbiasa dengan slicing pada list Python (contoh: `my_list[2:5]`), konsepnya di Pandas sangat mirip.

Aturan Emas: Urutkan Indeks Terlebih Dahulu!

Aturan paling penting sebelum melakukan slicing berdasarkan label dengan `.loc` adalah: **indeks harus diurutkan (sorted)**. Jika tidak, Pandas akan memberikan error atau hasil yang tidak terduga.

```
# Contoh: Mengatur indeks dan langsung mengurutkannya
dogs_srt = dogs.set_index(["breed", "color"]).sort_index()
```

Slicing dengan `.loc` (Berdasarkan Label)

Saat Anda melakukan slicing dengan `.loc`, Anda menentukan label awal dan akhir dari rentang yang ingin Anda pilih.

Penting: Slicing dengan `.loc` bersifat **inklusif**, artinya nilai akhir yang Anda tentukan **akan ikut terpilih**.

Contoh 1: Slicing Baris

```
# Memilih semua anjing dari "Chow Chow" hingga "Poodle"
dogs_srt.loc["Chow Chow":"Poodle"]
```

[Gambar dari DataFrame Pandas yang telah diiris]

Contoh 2: Slicing Baris dan Kolom Sekaligus Anda juga bisa mengiris baris dan kolom secara bersamaan.

```
# Memilih baris dari Labrador Coklat hingga Schnauzer Abu-abu,
# dan kolom dari 'name' hingga 'height_cm'
dogs_srt.loc[("Labrador", "Brown"):("Schnauzer", "Grey"),
"name":"height_cm"]
```

Slicing dengan `.iloc` (Berdasarkan Posisi)

Anda juga bisa melakukan slicing berdasarkan posisi integer menggunakan `.iloc`. Sama seperti list Python, slicing dengan `.iloc` bersifat **eksklusif**, artinya nilai akhir **tidak akan ikut terpilih**.

```
# Memilih baris di posisi 2, 3, dan 4 (indeks 5 tidak termasuk)
# dan kolom di posisi 1, 2, dan 3 (indeks 4 tidak termasuk)
dogs.iloc[2:5, 1:4]
```

Latihan Singkat

Gunakan DataFrame `df_tgl` dari latihan kita sebelumnya, yang indeksnya sudah berupa tanggal dan terurut.

```
# df_tgl dari latihan sebelumnya
df_tgl = df_penjualan.set_index('Tanggal').sort_index()
```

1. Gunakan **slicing dengan `.loc`** untuk memilih semua transaksi dari tanggal **'2024-07-02' hingga '2024-07-04'**.