VISUALISASI DATA

Manipulasi Data dengan Pandas

Magister Teknik Informatika

Dr. Chairani, S.Kom., M.Eng
IIB DARMAJAYA, 2023/2024

Manipulasi Data dengan Pandas

01 **Mengenal Pandas**

Lorem ipsum dolor sit amet, consectetur adipiscing. Donec risus dolor, porta venenatis neque sit amet, pharetra luctus felis.

02 **Mengenal Dataframe**

Lorem ipsum dolor sit amet, consectetur adipiscing. Donec risus dolor, porta venenatis neque sit amet, pharetra luctus felis.

03 **Mengakses Dataframe**

Lorem ipsum dolor sit amet, consectetur adipiscing. Donec risus dolor, porta venenatis neque sit amet, pharetra luctus felis.

04 **Pengelompokan Data**

Lorem ipsum dolor sit amet, consectetur adipiscing. Donec risus dolor, porta venenatis neque sit amet, pharetra luctus felis.

05 **Modifikasi Dataframe**

Lorem ipsum dolor sit amet, consectetur adipiscing. Donec risus dolor, porta venenatis neque sit amet, pharetra luctus felis.

06 **Operasi Lambda**

Lorem ipsum dolor sit amet, consectetur adipiscing. Donec risus dolor, porta venenatis neque sit amet, pharetra luctus felis.

Apa itu Pandas?

Pandas adalah library open-source yang menyediakan struktur data dan fungsi untuk memanipulasi dan menganalisis data dalam Python

Mengapa Pandas?

- Kemudahan Penggunaan
- DataFrame
- Fungsionalitas yang Kuat
- Kompatibilitas dengan Data yang Beragam

Implementasi Pandas dalam Data Analytic

- Data Manipulation
- Data Cleaning
- Data Analysis
- Data Visualization

Apa itu Dataframe?

Struktur data dua dimensi yang disediakan oleh library Pandas dalam bahasa pemrograman Python

Komponen Dataframe

- Baris
- Kolom
- Indeks Baris
- Indeks Kolom
- Tipe Data
- Attribut

	Nama	Usia	Kota
0	John	28	New York
1	Anna	35	Paris
2	Peter	40	Berlin
3	Linda	32	London

Tampilan Dataframe

Hands-on:

- Memuat data ke dalam Dataframe
- Menampilkan Dataframe
- Mengidentifikasi atribut Dataframe

Mengakses Kolom Dataframe

- Menggunakan nama kolom
- Menggunakan indeks kolom

```
# Mengakses satu kolom

df['Nama_Kolom']

# Mengakses beberapa kolom

df[['Nama_Kolom_1', 'Nama_Kolom_2']]


# Mengakses satu kolom berdasarkan indeks

df.iloc[:, indeks_kolom]

# Mengakses beberapa kolom berdasarkan indeks

df.iloc[:, [indeks_kolom_1, indeks_kolom_2]]
```

Cara Mengakses Baris Dataframe

- Menggunakan indeks baris
- Menggunakan label indeks
- Menggunakan kriteria

```
# Mengakses baris berdasarkan indeks

df.iloc[indeks_baris]

# Mengakses baris berdasarkan rentang indeks

df.iloc[indeks_awal:indeks_akhir]

# Mengakses baris terakhir

df.iloc[-1]

# Mengakses baris berdasarkan kriteria

df[df['Kolom_Kriteria'] == nilai_kriteria]

# Menggunakan operasi logika untuk masking

df[(df['Kolom_1'] > nilai_1) & (df['Kolom_2'] < nilai_2)]


# Menggunakan fungsi loc untuk mengakses baris berdasarkan label indeks

df.loc[label_indeks]

# Menggunakan fungsi iloc untuk mengakses baris berdasarkan label indeks

df.iloc[label_index]
```

Konsep Pengelompokan Data

Data yang tersedia pada umumnya memiliki kolom yang dapat dikelompokkan berdasarkan kategorinya.

Misal:

- Menghitung rata-rata nilai berdasarkan kelas
- Menghitung rata-rata tinggi bada berdasarkan usia

Kode syntaks: dataframe.groupby()

Jenis perintah agregasi data

- sum
- mean
- count
- unique

Mengubah nilai kolom

```
# Mengubah nilai dalam satu kolom

df['Kolom'] = nilai_baru


# Mengubah nilai dalam beberapa kolom

df[['Kolom_1', 'Kolom_2']] = nilai_baru
```

Menambah kolom

```
# Menambahkan kolom baru dengan nilai statis

df['Kolom_Baru'] = nilai_baru


# Menambahkan kolom baru dengan hasil operasi kolom lain

df['Kolom_Baru'] = df['Kolom_1'] + df['Kolom_2']
```

Menghapus kolom

```
# Menghapus satu kolom

df.drop('Kolom', axis=1, inplace=True)


# Menghapus beberapa kolom

df.drop(['Kolom_1', 'Kolom_2'], axis=1, inplace=True)
```

Menambah baris baru

```
# Menambahkan baris baru dengan nilai

df = df.append({'Kolom_1': nilai_1, 'Kolom_2': nilai_2},
ignore_index=True)
```

Menghapus baris

```
# Menghapus baris berdasarkan indeks

df.drop(indeks_baris, inplace=True)


# Menghapus baris berdasarkan kriteria

df = df[df['Kolom'] != nilai_kriteria]
```

Mengatur ulang index

```
# Mereset indeks DataFrame

df.reset_index(drop=True, inplace=True)
```

Apa itu Operasi Lambda?

- Fungsi anonim dalam Python yang didefinisikan tanpa nama menggunakan kata kunci lambda.
- Berguna ketika Anda memerlukan fungsi sederhana untuk digunakan sekali atau sebagai argumen dalam fungsi lain.
- Digunakan bersama dengan Pandas DataFrame, terutama dalam operasi seperti transformasi data, filtering, atau dalam operasi yang memerlukan fungsi anonim sederhana

Jenis Perintah Operasi Lambda

- apply
- map
- filter

```
# Contoh: Menggunakan lambda untuk menghitung kuadrat dari setiap elemen dalam kolom 'Usia'
df['Usia_Kuadrat'] = df['Usia'].apply(lambda x: x**2)


# Contoh: Menggunakan lambda untuk mengonversi setiap elemen dalam kolom 'Nilai' menjadi huruf A jika lebih dari 80, dan B jika tidak
df['Nilai_Grade'] = df['Nilai'].map(lambda x: 'A' if x > 80 else 'B')


# Contoh: Menggunakan lambda untuk memfilter baris berdasarkan kondisi tertentu dalam kolom 'Usia'
df_filtered = df[df['Usia'].apply(lambda x: x >= 30)]
```

Terimakasih