

An aerial photograph of a city skyline. The central focus is a tall, dark skyscraper with a grid-like window pattern. To its left, another skyscraper with a lighter, more regular window pattern is visible. The top of the central building shows a flat roof with several circular vents and a small structure. The overall scene is captured in a high-angle, slightly dimly lit perspective, suggesting an urban environment.

Hari 3

Analisis Data dengan Pandas

Import di python

Untuk menggunakan library di Python, kita perlu mengimpornya terlebih dahulu.

```
import pandas as pd
import matplotlib.pyplot as plt
import random
```

Contoh di atas mengimpor library `pandas` dengan alias `pd`, `matplotlib.pyplot` dengan alias `plt`, dan `random`.

Apa itu Pandas?

- Pandas adalah library Python untuk analisis data.
- Memiliki struktur data utama:
 - Series → seperti kolom tunggal.
 - DataFrame → seperti tabel (baris & kolom).

Contoh Series

```
import pandas as pd

data = [10, 20, 30, 40, 50]
series = pd.Series(data)
print(series)
```

Contoh DataFrame

```
import pandas as pd

data = {
    'Nama': ['Andi', 'Siti', 'Budi'],
    'Umur': [25, 30, 22],
    'Kota': ['Jakarta', 'Bandung', 'Surabaya']
}

df = pd.DataFrame(data)
print(df)
```

Membaca Data dengan Pandas

Kode berikut dapat digunakan untuk membuat contoh data dan menyimpan ke file CSV.

```
import pandas as pd

data = {
    "nama": ["Andi", "Siti", "Budi", "Rina", "Tono", "Dewi", "Agus", "Maya"],
    "umur": [32, 28, 41, 36, 29, 27, 45, 33],
    "saldo": [5200000, 2300000, 8900000, 4500000, 3100000, 2600000, 12000000, 5600000],
    "status": ["aktif", "tidak aktif", "aktif", "aktif", "tidak aktif", "tidak aktif", "aktif", "aktif"]
}


df = pd.DataFrame(data)
df.to_csv("nasabah.csv", index=False) # Simpan file untuk latihan
```

Membaca File CSV

```
df = pd.read_csv("nasabah.csv")
print("Dataset berhasil dibaca!")
display(df.head()) # Menampilkan 5 baris pertama
```

Mengeksplorasi Struktur Data

```
df.info()           # Struktur kolom & tipe data
df.describe()      # Statistik deskriptif (rata-rata, min, max, dst)
df.shape           # Jumlah baris & kolom
df.columns         # Daftar kolom
df.dtypes          # Jenis data per kolom
```

 `df` diatas diasumsikan sebagai variable yang berisi DataFrame baik dari hasil membaca file atau membuat manual

Data Cleaning

Tambahkan baris data yang tidak lengkap sebagai contoh:

```
# Tambahkan 1 baris dengan data hilang
df.loc[len(df)] = ["Rudi", None, None, "tidak aktif"]

df.tail()
```

💡 Fungsi `df.loc` digunakan untuk memanggil/akses data. Akses dapat melalui nama kolom, index kolom atau bahkan rentang kolom

💡 Fungsi `len(df)` digunakan untuk mendapatkan jumlah baris pada DataFrame `df`, sehingga `df.loc[len(df)]` menambahkan baris baru di akhir DataFrame.

Data Cleaning

memeriksa data kosong

```
# Cek data yang hilang  
df.isnull().sum()
```

Mengisi nilai kosong

```
# Isi nilai kosong pada kolom 'umur' dengan rata-rata umur  
df['umur'].fillna(df['umur'].mean(), inplace=True)  
# Isi nilai kosong pada kolom 'saldo' dengan 0  
df['saldo'].fillna(0, inplace=True)
```

Menghapus baris dengan data kosong

```
df.dropna(inplace=True)
```

Data Cleaning

Mengganti Nilai

```
df['status'].replace({'tidak aktif': 'non-aktif'}, inplace=True)
```

Memfilter data

```
# Filter nasabah dengan saldo > 5 juta  
high_balance = df[df['saldo'] > 5000000]  
display(high_balance)
```

Mengurutkan data

```
# Urutkan berdasarkan saldo secara menurun  
sorted_df = df.sort_values(by='saldo', ascending=False)  
display(sorted_df)
```


Analisis Data Sederhana

```
rata_saldo = df["saldo"].mean()
print(f"💰 Rata-rata saldo nasabah: Rp {rata_saldo:,.0f}")

rata_saldo_aktif = df[df["status"] == "aktif"]["saldo"].mean()
print(f"💎 Rata-rata saldo nasabah aktif: Rp {rata_saldo_aktif:,.0f}")

umur_rata = df["umur"].mean()
print(f"👤 Umur rata-rata nasabah: {umur_rata:.1f} tahun")
```

Grouping Data

```
# Hitung rata-rata saldo berdasarkan status
grouped = df.groupby("status")["saldo"].mean().reset_index()
grouped
```

Visualisasi Hasil Grouping

```
import matplotlib.pyplot as plt

plt.bar(group["status"], group["saldo"], color=["lightblue", "lightcoral"])
plt.title("Rata-rata Saldo Berdasarkan Status Nasabah")
plt.xlabel("Status")
plt.ylabel("Rata-rata Saldo")
plt.show()
```