Pertemuan 01: Dasar Python untuk Machine Learning

Nama	NIM
Muhammad Zaky Farhan	105841110523

Tujuan Praktikum:

Praktikum ini bertujuan untuk menyiapkan *environment* Python yang dikhususkan untuk kebutuhan Machine Learning. Fokus utamanya adalah membiasakan penggunaan *library* dasar seperti NumPy, Pandas, dan Matplotlib, serta mencoba memuat dataset bawaan dari scikit-learn langsung di dalam Jupyter Notebook.

Penjelasan Kode Ringkas

Langkah awal yang perlu dilakukan adalah memanggil *library* utama agar fungsinya bisa dipakai di dalam program. Perintah `import` digunakan untuk memasukkan *library*, dan tambahan `as` dipakai untuk memberi nama singkatan biar penulisan kodenya nanti tidak kepanjangan, misalnya `numpy` disingkat jadi `np`. Khusus untuk dataset, perintah `from` ditambahkan agar fungsi `load_iris` bisa ditarik langsung dari dalam `sklearn.datasets`.

In [1]:
```
# Import library
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
```

Di bagian NumPy, perintah `np.array()` berfungsi mengubah rentetan angka biasa menjadi format array khusus. Format ini membuat proses hitung-hitungan di komputer berjalan jauh lebih cepat. Setelah array terbentuk, nilai rata-ratanya bisa langsung dicari lewat fungsi `.mean()`.

In [2]:
```
# --- NumPy ---
# Membuat array angka lalu menghitung rata-rata.
x = np.array([1, 2, 3, 4, 5])
print("mean:", x.mean())
```

```
mean: 3.0
```

Masuk ke bagian Pandas, ada perintah `pd.DataFrame()` yang tugasnya mengubah data mentah berbentuk *dictionary* (data di dalam kurung kurawal) menjadi tabel yang rapi. Teks "nama" dan "nilai" otomatis menjadi judul kolom, dan daftar isinya berjejer ke bawah sebagai baris data, sehingga tampilannya jauh lebih mudah dibaca.

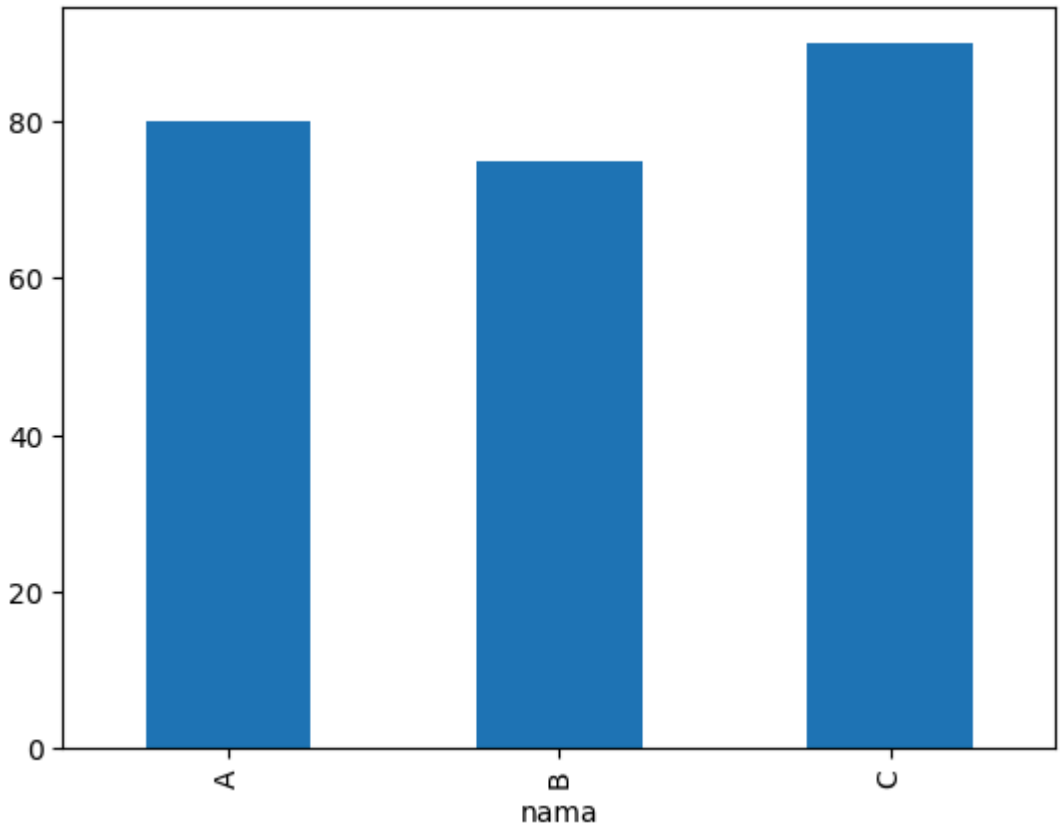
In [3]:
```
# Pandas
# Membuat tabel kecil berisi nama dan nilai, lalu menampilkannya.
```

```
df = pd.DataFrame({"nama": ["A", "B", "C"], "nilai": [80, 75, 90]})
print(df)
```

```
  nama  nilai
0    A     80
1    B     75
2    C     90
```

Untuk urusan visualisasi, fungsi `.plot()` bisa ditempelkan langsung pada tabel Pandas untuk menggambar grafik. Parameter `kind="bar"` disematkan untuk memastikan grafiknya berbentuk batang, sementara parameter `x` dan `y` bertugas menentukan kolom mana yang menjadi sumbu mendatar dan sumbu tegaknya. Atribut `legend=False` ditambahkan untuk menyembunyikan kotak keterangan warna yang memang tidak diperlukan pada grafik tunggal ini. Fungsi `plt.show()` kemudian diletakkan di baris paling akhir agar wujud grafiknya benar-benar dirender ke layar.

In [4]:
```
# Visualisasi
# Membuat bar chart dari tabel di atas.
df.plot(kind="bar", x="nama", y="nilai", legend=False)
plt.show()
```



Terakhir, dataset bawaan dipanggil menggunakan `load_iris()`. Ada atribut `as_frame=True` yang disisipkan di dalam kurung supaya kumpulan angka dari dataset tersebut langsung dikonversi menjadi format tabel Pandas. Begitu datanya tersimpan di variabel `iris`, tabel utuhnya bisa diakses lewat properti `.frame`. Fungsi `.head()` kemudian dipakai untuk menampilkan lima baris teratas saja sebagai cuplikan awal.

In [5]:
```
# Dataset ML
# Memuat dataset Iris (150 bunga, 4 fitur, 3 kelas).
```

```
iris = load_iris(as_frame=True)
print(iris.frame.head())
```

```
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \
0                5.1               3.5                1.4               0.2
1                4.9               3.0                1.4               0.2
2                4.7               3.2                1.3               0.2
3                4.6               3.1                1.5               0.2
4                5.0               3.6                1.4               0.2

   target
0       0
1       0
2       0
3       0
4       0
```

Pada bagian output operasi NumPy, tercetak angka 3.0 yang merupakan hasil perhitungan rata-rata array. Tepat di bawahnya, eksekusi kode Pandas menghasilkan sebuah tabel kecil dengan dua kolom bernama "nama" dan "nilai" yang memuat tiga baris data. Selanjutnya, fungsi visualisasi dari Matplotlib menampilkan sebuah grafik batang (bar chart) yang merepresentasikan perbandingan nilai antara A, B, dan C. Pada bagian paling akhir, pemanggilan dataset dari scikit-learn memperlihatkan lima baris pertama dari tabel dataset Iris yang memuat rincian fitur ukuran sepal dan petal, lengkap dengan kolom target kelasnya.

Tugas Praktikum

1. Buat array random 20 angka, hitung `mean`, `median`, `std`.
2. Buat DataFrame nilai 10 mahasiswa dan cari 3 nilai tertinggi.
3. Buat 2 grafik dari data nilai.
4. Load dataset `load_wine()` atau `load_breast_cancer()` lalu tulis 3 insight singkat.

Pengerjaan Tugas

Tugas 1

Soal: Buat array random 20 angka, hitung `mean`, `median`, `std`.

Soal ini minta dibuatkan 20 angka secara acak, lalu menghitung nilai rata-rata, nilai tengah, dan standar deviasinya. Penyelesaiannya bertumpu pada fungsi acak bawaan NumPy untuk menghasilkan angka, lalu dilanjutkan dengan memakai fungsi statistik dasar langsung pada kumpulan angka tersebut.

Penjelasan Kode: Pengaturan awal menggunakan perintah `np.random.seed(10)`. Fungsinya adalah mengunci generator angka acak agar deretan angka yang keluar tetap sama persis meskipun kodenya dijalankan berulang kali. Perintah `np.random.uniform(0, 100, 20)` kemudian bertugas memproduksi 20 angka

desimal yang tersebar secara acak dari rentang 0 sampai 100. Hasilnya ditampung ke dalam variabel bernama `data_acak`.

Untuk merapikan tampilan saat dicetak, fungsi `.round(2)` dipanggil supaya deretan angkanya dibulatkan menjadi maksimal dua digit di belakang koma. Bagian perhitungan statistiknya cukup singkat. Fungsi `np.mean()` digunakan untuk mencari nilai rata-rata keseluruhan angka. Fungsi `np.median()` mengekstrak nilai yang letaknya ada persis di tengah setelah datanya diurutkan. Sementara itu, `np.std()` menghitung standar deviasi untuk melihat seberapa lebar rentang sebaran datanya. Semuanya dibungkus dengan format `:.2f` di dalam teks agar hasil cetaknya seragam memiliki dua angka desimal.

In [6]:
```python
import numpy as np

# Mengatur seed agar hasil angka acak tetap konsisten
np.random.seed(10)

# Menghasilkan 20 angka acak dengan rentang 0 sampai 100
data_acak = np.random.uniform(0, 100, 20)

print("Daftar Angka Acak:")
print(data_acak.round(2))
print(f"\nRata-rata       : {np.mean(data_acak):.2f}")
print(f"Nilai Tengah    : {np.median(data_acak):.2f}")
print(f"Standar Deviasi : {np.std(data_acak):.2f}")
```

```
Daftar Angka Acak:
[77.13  2.08 63.36 74.88 49.85 22.48 19.81 76.05 16.91  8.83 68.54 95.34
  0.39 51.22 81.26 61.25 72.18 29.19 91.78 71.46]

Rata-rata       : 51.70
Nilai Tengah    : 62.31
Standar Deviasi : 30.01
```

Tampil deretan 20 angka acak dengan format desimal yang rapi, seperti 77.13, 2.08, 63.36, dan seterusnya. Tepat di bawah deretan angka tersebut, ada rincian hasil perhitungan berupa Rata-rata sebesar 51.70, Nilai Tengah di angka 62.31, dan Standar Deviasi senilai 30.01.

Tugas 2

Soal: Buat DataFrame nilai 10 mahasiswa dan cari 3 nilai tertinggi.

Tugas ini mengharuskan pembuatan tabel yang berisi daftar nama berserta nilai ujiannya, lalu mencari tiga data dengan nilai paling tinggi. Datanya akan disusun dulu menggunakan format *dictionary*, diubah menjadi tabel Pandas, dan disaring memakai fungsi urutan otomatis dari Pandas.

Penjelasan Kode: Pembuatan data dimulai dengan menyusun sebuah *dictionary* yang diberi nama `data_akademik`. Di dalamnya terdapat daftar sepuluh nama mahasiswa untuk mengisi kolom `Nama_Mahasiswa`, dan sepuluh angka nilai ujian untuk kolom `Skor_Ujian`. Perintah `pd.DataFrame()` kemudian membungkus *dictionary* ini dan

menyulapnya menjadi bentuk tabel baris dan kolom yang utuh, lalu menyimpannya di variabel `df_akademik`.

Pencarian nilai tertingginya mengandalkan metode `.nlargest(3, 'Skor_Ujian')`. Angka 3 mewakili jumlah baris data yang mau diambil, dan teks 'Skor_Ujian' merupakan nama kolom yang dijadikan acuan penilaian. Fungsi ini sangat praktis karena langsung mencari dan mengurutkan nilai dari yang paling besar tanpa memerlukan proses pengurutan data secara manual. Hasil saringannya ditampung di variabel `peringkat_atas` untuk kemudian dicetak ke layar bersanding dengan tabel utuhnya.

In [7]:
```
import pandas as pd

# Penyusunan data nama mahasiswa dan skor ujian
data_akademik = {
    'Nama_Mahasiswa': [
        'Prabowo Subianto', 'Bahlil Lahadalia', 'Gibran Rakabuming',
        'Sri Mulyani', 'Luhut Binsar', 'Erick Thohir',
        'Retno Marsudi', 'Ganjar Pranowo', 'Anies Baswedan', 'Mahfud MD'
    ],
    'Skor_Ujian': [85, 98, 82, 95, 90, 88, 92, 75, 100, 99]
}

df_akademik = pd.DataFrame(data_akademik)

# Mengambil tiga baris dengan skor tertinggi
peringkat_atas = df_akademik.nlargest(3, 'Skor_Ujian')

print("Tabel Data Nilai Lengkap:")
print(df_akademik)
print("\nTiga Mahasiswa dengan Skor Tertinggi:")
print(peringkat_atas)
```

```
Tabel Data Nilai Lengkap:
      Nama_Mahasiswa  Skor_Ujian
0   Prabowo Subianto          85
1   Bahlil Lahadalia          98
2  Gibran Rakabuming          82
3        Sri Mulyani          95
4       Luhut Binsar          90
5       Erick Thohir          88
6      Retno Marsudi          92
7     Ganjar Pranowo          75
8     Anies Baswedan         100
9          Mahfud MD          99

Tiga Mahasiswa dengan Skor Tertinggi:
     Nama_Mahasiswa  Skor_Ujian
8    Anies Baswedan         100
9         Mahfud MD          99
1  Bahlil Lahadalia          98
```

Muncul satu tabel penuh yang berisi nomor urut di sebelah kiri, diikuti oleh sepuluh nama mahasiswa, dan skor ujian mereka masing-masing. Di bagian bawahnya, tampil tabel kedua yang berukuran lebih ringkas, di mana isinya hanya memuat tiga baris nama mahasiswa dengan skor tertinggi, yaitu Anies Baswedan (100), Mahfud MD (99), dan Bahlil Lahadalia (98).

Tugas 3

Soal: Buat 2 grafik dari data nilai.

Data nilai dari tabel sebelumnya akan diubah menjadi dua jenis grafik agar pergerakan nilainya lebih mudah dipahami secara visual. Kanvas gambar akan dibagi menjadi dua sisi; sisi kiri untuk grafik garis yang menunjukkan naik-turunnya nilai, dan sisi kanan untuk grafik titik yang memperlihatkan titik persebaran nilainya.

Penjelasan Kode: Pengaturan tata letak gambar dilakukan lewat perintah `plt.subplots(1, 2, figsize=(14, 6))`. Perintah ini menyiapkan sebuah area kanvas dengan panjang 14 dan tinggi 6, lalu membaginya rata menjadi satu baris dan dua kolom. Area kiri diwakili oleh variabel `ax1` dan area kanan oleh `ax2`.

Di area pertama (`ax1`), fungsi `.plot()` dipakai untuk menggambar grafik garis. Kolom nama dipasang pada sumbu mendatar dan kolom skor pada sumbu tegak. Atribut `marker='o'` disisipkan agar muncul titik tebal di setiap perhentian garis, dengan pewarnaan `teal` (biru kehijauan) dan ketebalan garis diatur di angka 2. Di area kedua (`ax2`), fungsi `.scatter()` dijalankan untuk menghasilkan grafik berupa titik-titik yang berdiri sendiri. Ukuran titiknya diperbesar lewat parameter `s=150`, diberi warna `crimson` (merah tua), dan garis pinggiran titiknya diberi warna hitam biar terlihat lebih tegas.

Fungsi `.set_title()` dan `.set_ylabel()` dipakai untuk memberikan judul dan keterangan pada masing-masing sumbu grafik. Supaya teks nama-nama mahasiswa di bagian bawah grafik tidak saling menumpuk, perintah `.tick_params(axis='x', rotation=45)` diaplikasikan agar teksnya diputar miring 45 derajat. Sebelum ditampilkan, `plt.tight_layout()` dipanggil untuk memastikan letak seluruh elemennya rapi dan tidak ada teks yang terpotong, lalu diakhiri dengan `plt.show()`.

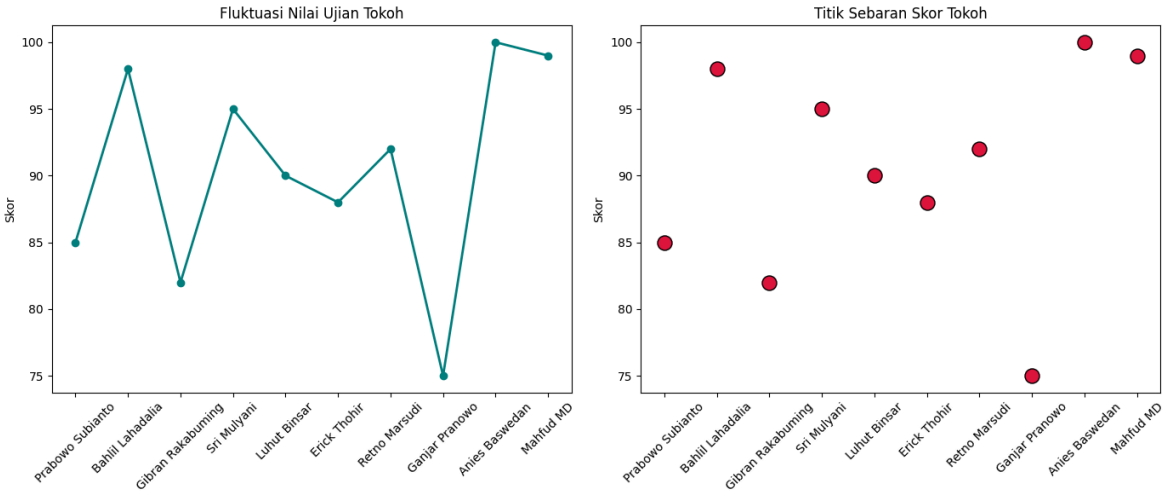
In [8]:
```
import matplotlib.pyplot as plt

# Membuat pengaturan tata letak grafik berdampingan
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(14, 6))

# Grafik 1: Line Chart untuk melihat fluktuasi nilai
ax1.plot(df_akademik['Nama_Mahasiswa'], df_akademik['Skor_Ujian'], marker='o', c
ax1.set_title('Fluktuasi Nilai Ujian Tokoh')
ax1.set_ylabel('Skor')
ax1.tick_params(axis='x', rotation=45)

# Grafik 2: Scatter Plot untuk melihat titik sebaran skor
ax2.scatter(df_akademik['Nama_Mahasiswa'], df_akademik['Skor_Ujian'], color='cri
ax2.set_title('Titik Sebaran Skor Tokoh')
ax2.set_ylabel('Skor')
ax2.tick_params(axis='x', rotation=45)

plt.tight_layout()
plt.show()
```



Tampil dua area grafik berdampingan yang menyajikan representasi visual dari data nilai. Grafik di sebelah kiri merupakan grafik garis (line chart) yang berfungsi untuk menunjukkan tren atau fluktuasi nilai ujian antar mahasiswa. Sementara itu, grafik di sebelah kanan adalah grafik titik (scatter plot) yang merepresentasikan letak persebaran skor setiap individu secara mandiri berdasarkan tinggi rendahnya nilai. Pada bagian sumbu horizontal di kedua grafik tersebut, label nama mahasiswa sengaja ditampilkan miring 45 derajat agar teks tidak saling tumpang tindih dan tetap mudah dibaca.

Tugas 4

Soal: Load dataset `load_wine()` atau `load_breast_cancer()` lalu tulis 3 insight singkat.

Penyelesaian tugas ini difokuskan pada pemuatan dataset bawaan dari scikit-learn untuk dieksplorasi isinya. Dataset yang dipilih adalah data mengenai kanker payudara (*breast cancer*). Dataset ini ditarik dan langsung dikonversi menjadi format tabel Pandas. Properti bawaan Pandas kemudian dimanfaatkan untuk mengekstrak dimensi tabel, mendaftar nama kolom, menghitung proporsi klasifikasi target, dan mencari nilai rata-rata dari salah satu fitur fisiknya. Angka-angka hasil ekstraksi ini kemudian dirangkai menjadi tiga poin wawasan (*insight*).

Penjelasan Kode: Pemrosesan diawali dengan pemanggilan perintah `from sklearn.datasets import load_breast_cancer` untuk menarik fungsi pemuat dataset dari modul scikit-learn. Fungsi `load_breast_cancer(as_frame=True)` kemudian dieksekusi. Keberadaan parameter `as_frame=True` di dalam kurung berfungsi memaksa data matriks mentah tersebut agar langsung terformat menjadi tabel DataFrame Pandas. Wujud utuh data ini ditampung dalam variabel `dataset_kanker`, lalu wujud tabelnya secara spesifik diambil melalui pemanggilan properti `.frame` dan disimpan ke dalam variabel `df_kanker`.

Untuk mencetak informasi umum, properti `.shape` dipanggil guna mengekstrak tupel berisi total baris dan kolom. Diikuti dengan pemanggilan atribut `.columns` yang bertugas menyalin seluruh label kepala kolom. Atribut ini dibungkus menggunakan fungsi `list()` agar wujud tampilannya berubah menjadi struktur daftar biasa sehingga rapi saat dicetak. Untuk menyusun poin *insight*, indeks `[0]` ditempelkan pada

`df_kanker.shape` guna mengambil secara spesifik angka jumlah baris, sedangkan indeks `[1] - 1` digunakan untuk mengambil angka jumlah kolom yang dikurangi satu, mengingat satu kolom terakhir berfungsi sebagai label kelas dan bukan fitur data. Eksplorasi pada kolom target dilakukan melalui metode `.nunique()` untuk menghitung total kategori unik yang ada. Metode `.unique()` kemudian mengambil nilai spesifik dari label tersebut, yang langsung dibalut dengan fungsi `sorted()` agar urutan angkanya tertata dari kecil ke besar. Sebagai penutup, nilai rata-rata fitur dikalkulasi menggunakan metode `.mean()` khusus pada kolom `mean radius`. Hasil rata-rata ini diapit oleh fungsi `round(..., 2)` untuk membulatkan nilai desimalnya menjadi dua digit saja, lalu keseluruhan nilainya dicetak berurutan menggunakan perintah `print()`.

In [9]:
```
from sklearn.datasets import load_breast_cancer

dataset_kanker = load_breast_cancer(as_frame=True)
df_kanker = dataset_kanker.frame

print("Ukuran data:", df_kanker.shape)
print("Kolom:", list(df_kanker.columns))

print("\nInsight 1: Dataset Breast Cancer memiliki", df_kanker.shape[0], "sampel
print("Insight 2: Target memiliki", df_kanker["target"].nunique(), "kelas:", sor
print("Insight 3: Rata-rata mean radius =", round(df_kanker["mean radius"].mean(
```

```
Ukuran data: (569, 31)
Kolom: ['mean radius', 'mean texture', 'mean perimeter', 'mean area', 'mean smoot
hness', 'mean compactness', 'mean concavity', 'mean concave points', 'mean symmet
ry', 'mean fractal dimension', 'radius error', 'texture error', 'perimeter erro
r', 'area error', 'smoothness error', 'compactness error', 'concavity error', 'co
ncave points error', 'symmetry error', 'fractal dimension error', 'worst radius',
'worst texture', 'worst perimeter', 'worst area', 'worst smoothness', 'worst comp
actness', 'worst concavity', 'worst concave points', 'worst symmetry', 'worst fra
ctal dimension', 'target']

Insight 1: Dataset Breast Cancer memiliki 569 sampel dan 30 fitur.
Insight 2: Target memiliki 2 kelas: [0, 1]
Insight 3: Rata-rata mean radius = 14.13
```

Dalam output terdapat teks "Ukuran data:" yang diikuti dengan angka tupel (569, 31) sebagai representasi jumlah baris dan kolom. Tepat di bawahnya, muncul teks "Kolom:" yang merincikan daftar panjang seluruh nama kolom di dalam dataset, diawali dengan 'mean radius', 'mean texture', dan terus berlanjut hingga diakhiri oleh kolom 'target'. Setelah jeda satu baris kosong, baris teks "Insight 1" muncul dan menjelaskan bahwa Dataset Breast Cancer tersebut memiliki 569 sampel dan 30 fitur. Teks "Insight 2" kemudian menampilkan bahwa kolom Target memiliki 2 kelas, yang direpresentasikan dengan angka [0, 1]. Pada baris terakhir, teks "Insight 3" menampilkan hasil perhitungan matematika yang menunjukkan bahwa Rata-rata mean radius berada di angka 14.13.

Kesimpulan

Praktikum ini memberikan pemahaman dasar tentang cara kerja alat-alat pengolahan data di Python. NumPy sangat bisa diandalkan untuk mempercepat proses perhitungan

angka. Pandas sangat membantu dalam menata data mentah menjadi bentuk tabel yang enak dibaca dan mudah disaring. Matplotlib menyempurnakan proses tersebut dengan kemampuannya mengubah angka menjadi grafik visual yang jelas. Semua alat dasar ini, ditambah dengan tersedianya dataset bawaan dari scikit-learn, menjadi pondasi awal yang kuat sebelum beranjak ke tahapan proses pemodelan Machine Learning yang lebih rumit.