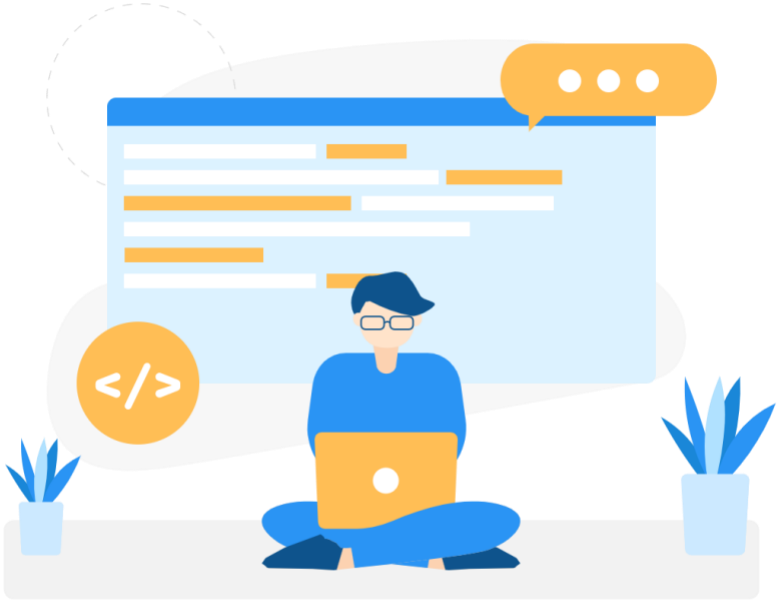
BAB 11

Python Library



11.1 Tujuan

1. Dapat menjelaskan konsep penggunaan library external pada python
2. Dapat mengimplementasikan library external seperti pathlib dan pytest pada pemrograman bahasa python

11.2 Pengantar

Python adalah salah satu bahasa pemrograman yang sangat populer di dunia. Popularitasnya tidak hanya karena sintaks yang sederhana dan mudah dipelajari, tetapi juga karena dukungan ekosistemnya yang luas. Python menyediakan banyak library bawaan (built-in library) seperti math, os, random, datetime, dan lain sebagainya. Library-library ini sudah mencakup berbagai kebutuhan dasar pemrograman. Misalnya, kita bisa menggunakan math untuk operasi matematika, os untuk berinteraksi dengan sistem operasi, dan random untuk menghasilkan angka acak.

Namun, dalam praktiknya, kebutuhan pengembangan perangkat lunak tidak selalu dapat dipenuhi oleh library bawaan. Seiring berkembangnya aplikasi, kita sering memerlukan fitur tambahan seperti pemrosesan data berskala besar, komunikasi jaringan, pengolahan gambar, bahkan machine learning. Pada titik inilah library eksternal berperan penting.

11.2.1 Konsep Library External

Library eksternal adalah paket kode Python yang dibuat oleh komunitas atau pengembang lain di luar distribusi standar Python. Artinya, library ini tidak langsung tersedia saat kita pertama kali menginstal Python, melainkan perlu diunduh dan dipasang secara terpisah.

Dengan adanya library eksternal, seorang pengembang tidak perlu membangun semua fungsionalitas dari nol. Katakanlah kita ingin mengambil data dari internet. Jika menggunakan cara manual, kita harus membuka koneksi socket, mengirim request HTTP, membaca response, dan menguraikannya. Itu pekerjaan rumit. Tetapi dengan library eksternal seperti requests, kita cukup menulis satu baris:

```
1. import requests
2. response = requests.get("https://example.com")
3. print(response.text)
```

Untuk menggunakan library eksternal, Python menyediakan package manager bernama pip (Python Package Installer). pip memungkinkan kita menginstal, memperbarui, dan menghapus library eksternal dengan perintah sederhana di terminal atau command prompt. Hampir semua library eksternal Python didistribusikan melalui Python Package Index (PyPI), yaitu repositori resmi yang menampung lebih dari 400 ribu paket Python. Situs resminya dapat diakses di https://pypi.org

Contoh instalasi library eksternal:

```
1. pip install requests
2. pip install numpy
3. pip install pytest
```

Beberapa library eksternal yang sering digunakan antara lain:

- requests : untuk komunikasi HTTP, misalnya mengambil data dari API.
- numpy : untuk perhitungan numerik dan operasi pada array multidimensi.
- pandas : untuk analisis data, manipulasi tabel, dan pengolahan dataset.
- matplotlib : untuk membuat grafik dan visualisasi data.
- scikit-learn : untuk machine learning.
- pytest : untuk pengujian otomatis.

11.2.2 pathlib

pathlib adalah library di Python yang digunakan untuk mempermudah manipulasi path (lokasi file dan direktori). Sebelum pathlib diperkenalkan, Python sudah memiliki os.path untuk tujuan serupa, namun pathlib menawarkan pendekatan berbasis objek yang lebih intuitif dan modern.

Fungsi utama pathlib:

- Membuat path ke file atau folder.
- Mengecek apakah file/folder ada.

- Membaca isi file.
- Menulis teks ke dalam file.
- Membuat folder baru.

11.2.3 pytest

pytest adalah library eksternal untuk melakukan pengujian otomatis di Python. Library ini sangat populer di kalangan pengembang karena mudah digunakan, ringkas, dan fleksibel. Dalam pemrograman, kita sering membuat fungsi-fungsi untuk menyelesaikan masalah tertentu. Namun, bagaimana memastikan fungsi itu selalu bekerja sesuai harapan? Dengan pengujian otomatis, kita bisa mendeteksi bug sejak dini.

11.3 Kegiatan Praktikum

11.3.1 Kegiatan Praktikum 1 : Implementasi pathlib

1. Buat sebuah file program baru, kemudian tuliskan kode berikut ini

```
1. from pathlib import Path
2.
3. # 1. Membuat folder baru
4. base_dir = Path("praktikum_data")
5. base_dir.mkdir(exist_ok=True)  # hanya dibuat jika belum ada
6. print(f"Folder dibuat di: {base_dir.resolve()}")
7.
8. # 2. Membuat file teks di dalam folder
9. file1 = base_dir / "catatan.txt"
10.file1.write_text("Ini adalah catatan pertama.\n")
11.print(f"File {file1.name} berhasil dibuat.")
12.
13.# 3. Menambahkan teks ke file (mode append)
14.with file1.open("a", encoding="utf-8") as f:
15.    f.write("Baris kedua ditambahkan.\n")
16.    f.write("Baris ketiga juga ditambahkan.\n")
17.print(f"Teks berhasil ditambahkan ke {file1.name}")
18.
19.# 4. Membaca isi file
20.print("\nIsi file catatan.txt:")
21.print(file1.read_text(encoding="utf-8"))
22.
```

```python
# 5. Membuat beberapa file tambahan
for i in range(3):
    sub_file = base_dir / f"data_{i+1}.txt"
    sub_file.write_text(f"Ini isi file data_{i+1}.txt\n")
print("Beberapa file tambahan berhasil dibuat.")

# 6. Iterasi semua file dalam folder
print("\nDaftar file dalam folder praktikum_data:")
for f in base_dir.iterdir():
    if f.is_file():
        print(f"- {f.name}")

# 7. Filter hanya file dengan ekstensi .txt
print("\nFile dengan ekstensi .txt:")
for f in base_dir.glob("*.txt"):
    print(f"- {f.name}")

# 8. Membuat subfolder dan memindahkan file
subfolder = base_dir / "arsip"
subfolder.mkdir(exist_ok=True)
file1.rename(subfolder / file1.name)
print(f"\nFile {file1.name} dipindahkan ke folder 'arsip'.")

# 9. Menampilkan path absolute semua file dalam folder utama
print("\nPath absolut semua file di praktikum_data:")
for f in base_dir.rglob("*"):
    print(f.resolve())
```

2. Amati hasilnya kemudian tulis analisis singkat mengenai kegiatan 1 ini

11.3.2 Kegiatan Praktikum 2 : Implementasi pytest

1. Buka terminal pada VSCODE kemudian ketikkan kode berikut untuk membuat virtual environment

```
# membuat virtual environment
python -m venv venv

# mengaktifkan virtual environment di Windows
venv\Scripts\activate

```

2. Instalasi pytest menggunakan pip

```
pip install pytest

```

3. Buat file bernama math_utils.py, isikan dengan kode berikut

```
# math_utils.py

def add(a, b):
    """Menjumlahkan dua bilangan"""
    return a + b

def subtract(a, b):
    """Mengurangkan dua bilangan"""
    return a - b

def multiply(a, b):
    """Mengalikan dua bilangan"""
    return a * b

def divide(a, b):
    """Membagi dua bilangan, dengan pengecekan pembagi nol"""
    if b == 0:
        raise ValueError("Tidak bisa membagi dengan nol")
    return a / b
```

4. Buat file bernama test_math_utils.py, pastikan file ini berada dalam 1 folder yang sama. isikan dengan kode berikut.

```
# test_math_utils.py

import pytest
from math_utils import add, subtract, multiply, divide

def test_add():
    assert add(2, 3) == 5
    assert add(-1, 1) == 0
    assert add(0, 0) == 0

def test_subtract():
```

```
    assert subtract(5, 3) == 2
    assert subtract(0, 5) == -5

def test_multiply():
    assert multiply(4, 3) == 12
    assert multiply(-2, 3) == -6

def test_divide():
    assert divide(10, 2) == 5
    assert divide(9, 3) == 3

def test_divide_by_zero():
    with pytest.raises(ValueError, match="Tidak bisa membag
   i dengan nol"):
        divide(5, 0)
```

5. Buka terminal kemudian jalankan perintah berikut

```
pytest

```

6. Amati hasilnya kemudian tulis analisis singkat mengenai kegiatan 2 ini

11.3.3 Kegiatan Praktikum 3 : Implementasi pytest bagian 2

1. Buat folder baru bernama string, buat virtual environment seperti pada Kegiatan 2.
2. Buat file bernama string_utils.py. Isikan dengan kode berikut

```
# string_utils.py

def reverse_string(s: str) -> str:
    """Membalik string, contoh: 'python' -> 'nohtyp'"""
    return s[::-1]

def is_palindrome(s: str) -> bool:
    """Mengecek apakah string palindrom"""
    cleaned = s.replace(" ", "").lower()
    return cleaned == cleaned[::-1]

def capitalize_words(s: str) -> str:
    """Mengubah huruf awal tiap kata menjadi kapital"""
```

```
    return " ".join(word.capitalize() for word in s.split()
)

def is_valid_email(email: str) -> bool:
    """Validasi sederhana alamat email"""
    return "@" in email and "." in email.split("@")[-1]

```

3. Buat file bernama test_string_utils.py, isikan dengan kode berikut

```
# test_string_utils.py

import pytest
from string_utils import reverse_string, is_palindrome, cap
italize_words, is_valid_email

def test_reverse_string():
    assert reverse_string("python") == "nohtyp"
    assert reverse_string("level") == "level"

def test_is_palindrome():
    assert is_palindrome("level") is True
    assert is_palindrome("kasur rusak") is True
    assert is_palindrome("python") is False

def test_capitalize_words():
    assert capitalize_words("belajar python pytest") == "Be
lajar Python Pytest"
    assert capitalize_words("halo dunia") == "Halo Dunia"

def test_is_valid_email():
    assert is_valid_email("user@example.com") is True
    assert is_valid_email("hello.world@gmail.com") is True
    assert is_valid_email("invalid-email") is False
    assert is_valid_email("user@domain") is False

```

4. Buka terminal kemudian jalankan perintah berikut

```
pytest

```

5. Amati hasilnya kemudian tulis analisis singkat mengenai kegiatan 3 ini

11.4 Tugas

1. Apa perbedaan utama antara library bawaan Python dan library eksternal?
2. Mengapa kita tidak sebaiknya menulis semua kode dari nol?
3. Apa manfaat utama menulis pengujian dengan pytest?
4. Menurut Anda, apakah semua fungsi perlu diuji dengan pytest? Mengapa?