

MODUL 1

1.1 Capaian Praktikum Pertemuan 1

Mahasiswa mampu mengelola struktur direktori dan file menggunakan Python. Mahasiswa mampu menggunakan modul `os` untuk membuat folder dan file secara otomatis. Mahasiswa mampu melakukan pengecekan eksistensi file dan folder dengan `os.path.exists()`. Mahasiswa juga mampu menulis data ke dalam file teks menggunakan Python. Mahasiswa mampu membuat program otomatisasi sederhana dengan perulangan dan logika pemrograman dasar.

1.2 Indikator Capaian

Mahasiswa mampu membuat folder utama menggunakan Python sebagai bukti pemahaman dasar terhadap manipulasi system file menggunakan modul `os`. Mahasiswa mampu membuat subfolder secara dinamis menggunakan perulangan (looping). Mahasiswa mampu membuat lima subfolder secara otomatis dengan menggabungkan path dengan benar menggunakan `os.path.join`. Mahasiswa mampu membuat dan menuliskan isi ke dalam file teks secara otomatis. Mahasiswa menguasai kemampuan dasar dalam otomatisasi pengelolaan file, direktori, dan penerapan logika pemrograman Python.

1.3 Landasan Teori

Menurut Gutttag (2016), otomatisasi dalam pemrograman Python, termasuk dalam pengelolaan file dan direktori, merupakan salah satu kekuatan utama Python yang menjadikannya sangat cocok untuk scripting dan pemrosesan data skala kecil hingga menengah. Dengan logika pemrograman dasar, seperti percabangan dan perulangan, pengguna dapat mengembangkan alat bantu sederhana yang sangat berguna untuk tugas-tugas berulang.

Menurut Summerfield (2010) menyatakan bahwa penggunaan struktur kontrol seperti perulangan `for` dalam Python sangat efisien untuk menciptakan objek secara

berulang, baik untuk data maupun struktur file. Ini sangat berguna saat membuat banyak folder atau file dengan pola nama berurutan, seperti folder1, folder2, dan seterusnya, karena menyederhanakan kode program dan mengurangi duplikasi

1.4 Pelaksanaan Praktikum

1.4.1 Percobaan Pertama

Mahasiswa diperintahkan untuk membuat struktur folder dan file secara otomatis menggunakan Python, dengan tujuan memahami penggunaan modul os dan dasar-dasar manipulasi file dan direktori.

a. Script / Setting Program

```
import os

main_folder = 'data_scraping'

subfolders = [f'dir{i}' for i in range(1, 6)]

if not os.path.exists(main_folder):
    os.makedirs(main_folder)
    print(f"Folder '{main_folder}' berhasil dibuat.")
else:
    print(f"Folder '{main_folder}' sudah tersedia.")

for folder in subfolders:
    folder_path = os.path.join(main_folder, folder)

    if not os.path.exists(folder_path):
        os.makedirs(folder_path)
        print(f"Subfolder '{folder}' dibuat.")
    else:
        print(f"Subfolder '{folder}' sudah tersedia.")

    for i in range(1, 6):
        file_name = f'file{i}.txt'
        file_path = os.path.join(folder_path, file_name)

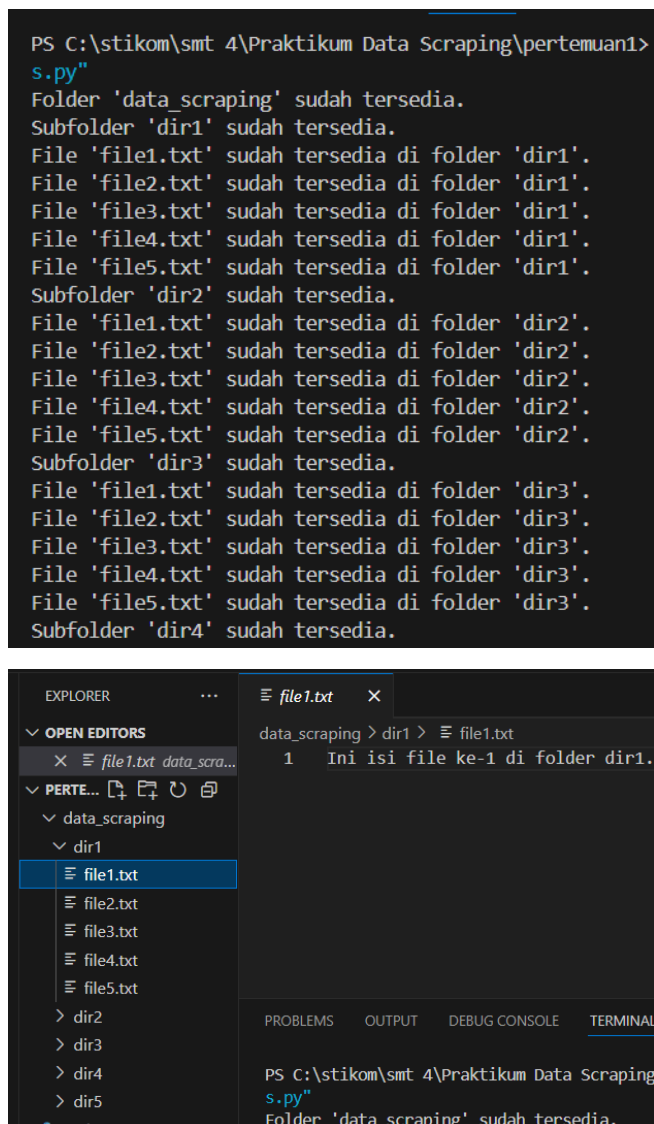
        if not os.path.exists(file_path):
            with open(file_path, 'w') as f:
                f.write(f"Ini isi file ke-{i} di folder {folder}.")
            print(f"File '{file_name}' dibuat di folder '{folder}'.")
        else:
            print(f"File '{file_name}' sudah tersedia di folder '{folder}'.")
```

b. Langkah Uji Coba

- **import os** Mengimpor modul os, yang digunakan untuk berinteraksi dengan sistem operasi, seperti membuat folder dan file, mengecek apakah file/folder sudah ada, menggabungkan path, dll.
- **main_folder = 'data_scraping'** Mendefinisikan nama folder utama (data_scraping).
- **subfolders = [f'dir{i}' for i in range(1, 6)]** Membuat daftar 5 subfolder (dir1 sampai dir5) secara otomatis menggunakan list comprehension.
- **if not os.path.exists(main_folder):** Mengecek apakah folder data_scraping sudah ada.
- **os.makedirs(main_folder)** Jika belum ada, maka folder tersebut akan dibuat menggunakan os.makedirs().
- **print(f"Folder '{main_folder}' berhasil dibuat.")** mencetak bahwa folder berhasil dibuat.
- **else: print(f"Folder '{main_folder}' sudah tersedia.")** Jika sudah ada, maka mencetak bahwa folder sudah tersedia.
- **for folder in subfolders:** Melakukan iterasi untuk setiap nama subfolder.
- **folder_path = os.path.join(main_folder, folder)** digunakan untuk menyusun path lengkap subfolder (data_scraping/dir1, dst.).
- **if not os.path.exists(folder_path):** Mengecek apakah subfolder (dir1 sampai dir5) sudah ada di dalam folder utama.
- **os.makedirs(folder_path)** Mencetak informasi keberhasilan atau jika subfolder sudah ada.
- **print(f"Subfolder '{folder}' dibuat.")** **else:** **print(f"Subfolder '{folder}' sudah tersedia.")** Jika sudah, maka akan mencetak bahwa subfolder sudah tersedia.
- **for i in range(1, 6):** Melakukan perulangan sebanyak 5 kali untuk membuat file file1.txt sampai file5.txt di setiap subfolder.
- **file_name = f'file{i}.txt'** file_name menyimpan nama filenya.

- **file_path = os.path.join(folder_path, file_name)** file_path menyimpan path lengkap ke file tersebut.
- **if not os.path.exists(file_path):** Mengecek apakah file tersebut sudah ada.
- **with open(file_path, 'w') as f:** Jika belum ada, file dibuat menggunakan with open(..., 'w'), artinya file dibuka untuk ditulis.
- **f.write(f"Ini isi file ke-{i} di folder {folder}.")** Jika sudah ada, program mencetak bahwa file tersebut sudah tersedia.
- **print(f"File '{file_name}' dibuat di folder '{folder}'.")** Jika file sudah ada, hanya mencetak bahwa file sudah tersedia.

c. Hasil Uji Coba



Gambar 1.1 Hasil Uji Coba Membuat 5 Directory

d. Analisa Hasil

Kode ini berhasil membuat folder utama dengan nama data_scraping, kemudian di dalamnya menghasilkan lima subfolder bernama dir1 hingga dir5. Setiap subfolder tersebut berisi lima file teks (file1.txt hingga file5.txt) yang masing-masing telah diisi dengan kalimat yang menunjukkan urutan file dan nama foldernya. Selain itu, program ini juga dilengkapi dengan pengecekan eksistensi sebelum membuat folder atau file, sehingga saat dijalankan ulang tidak terjadi duplikasi atau error.

1.5 Kesimpulan

1.5.1 Kesimpulan Percobaan 1

Pada percobaan ini, mahasiswa berhasil mengimplementasikan fungsionalitas untuk membuat struktur folder dan file secara otomatis menggunakan modul os pada Python. Program ini menunjukkan bahwa mahasiswa memahami cara dalam membuat folder, subfolder, serta file, lengkap dengan penulisan isi dan pengecekan eksistensi sebelum proses pembuatannya dilakukan. Hal ini membuktikan bahwa mahasiswa dapat memahami konsep dasar pengelolaan file dan direktori secara terstruktur, efisien, dan aman. Program juga dapat dijalankan berulang kali tanpa menghasilkan error atau duplikasi, yang mengindikasikan penggunaan logika kontrol yang tepat dan efektif.

Mengetahui:

Dosen Pengampu Mata Kuliah

Arif Hadi Sumitro , M.Kom

NIKP. xxx

DAFTAR PUSTAKA

1. Guttag, J. V. (2016). *Introduction to Computation and Programming Using Python* (2nd ed.). MIT Press.
2. Zelle, J. M. (2017). *Python Programming: An Introduction to Computer Science* (3rd ed.). Franklin, Beedle & Associates Inc.
3. Python Software Foundation. (2024). Reading and Writing Files. Diakses dari <https://docs.python.org/3/tutorial/inputoutput.html#reading-and-writing-files>