

MODUL FILE HANDLING

MEMBUAT APLIKASI SEDERHANA UNTUK DATA MAHASISWA, MENERAPKAN PENGGUNAAN FILE HANDLING YANG BISA MELAKUKAN BEBERAPA FITUR SEPERTI BERIKUT:

- Bisa Create Data
- Bisa Read Data
- Bisa Update Data
- Bisa Delete Data
- Bisa Search Data (Opsional)

Contoh Output

```
=====APLIKASI KELOLA DATA MAHASISWA=====
1. Tambah Data
2. Tampilkan Data
3. Update Data
4. Delete Data
5. Keluar
Pilih menu: 1
Masukkan Nama: Megumi
Masukkan NIM: 2022
Masukkan Mata Kuliah: Japanese
Masukkan Semester: 3
Data mahasiswa berhasil ditambahkan.
=====APLIKASI KELOLA DATA MAHASISWA=====
1. Tambah Data
2. Tampilkan Data
3. Update Data
4. Delete Data
5. Keluar
Pilih menu: 2
Menampilkan Data:
Data ke-1:
Nama: naufal
NIM: 2022
Mata Kuliah: proglan
Semester: 3

Data ke-2:
Nama: Megumi
NIM: 2022
Mata Kuliah: Japanese
Semester: 3

=====APLIKASI KELOLA DATA MAHASISWA=====
1. Tambah Data
2. Tampilkan Data
3. Update Data
4. Delete Data
5. Keluar
Pilih menu: 5
Keluar dari aplikasi.
PS D:\NOPAL\College\SEMESTER 3\FILE HANDLING>
```

Catatan Penting

- **a (append)**
Fungsi `append()` digunakan untuk menambahkan data baru ke akhir file yang sudah ada. Fungsi ini tidak akan menimpa data yang sudah ada dalam file.
- **w (write)**
Fungsi `write()` digunakan untuk menulis data baru ke file. Fungsi ini akan menimpa data yang sudah ada dalam file.
- **r (read)**
Fungsi `read()` digunakan untuk membaca data dari file. Fungsi ini akan mengembalikan data yang dibaca dari file sebagai string.

Penjelasan

```
def tambah_data():
    nama = input("Masukkan Nama: ")
    nim = input("Masukkan NIM: ")
    matkul = input("Masukkan Mata Kuliah: ")
    semester = input("Masukkan Semester: ")

    data_mahasiswa = f"{nama},{nim},{matkul},{semester}\n"

    with open("data_mahasiswa.txt", "a") as file:
        file.write(data_mahasiswa)

    print("Data mahasiswa berhasil ditambahkan.")
```

Disini terdapat fungsi dengan nama **tambah_data()**. Line-line ini mendefinisikan fungsi **tambah_data()**. Fungsi ini digunakan untuk menambahkan data mahasiswa baru. Fungsi ini meminta pengguna untuk memasukkan data mahasiswa, yaitu nama, NIM, mata kuliah, dan semester. Data tersebut kemudian disimpan dalam file teks **data_mahasiswa.txt**.

Line 1 mendeklarasikan fungsi **tambah_data()**. Line 2-5 meminta pengguna untuk memasukkan data mahasiswa. Line 6 menggabungkan data mahasiswa menjadi string. Line 7 membuka file teks **data_mahasiswa.txt** dalam **mode a (append)**. Mode a memungkinkan data baru untuk ditambahkan ke akhir file. Line 8 menulis data mahasiswa baru ke dalam file teks. Line 9 menampilkan pesan bahwa data mahasiswa telah berhasil ditambahkan.

```
def tampilkan_data():
    with open("data_mahasiswa.txt", "r") as file:
        lines = file.readlines()

    if not lines:
        print("Tidak ada data mahasiswa.")
        return

    print("Menampilkan Data:")
    for i, line in enumerate(lines, start=1):
        data = line.strip().split(',')
        print(f>Data ke-{i}:")
        print(f>Nama: {data[0]}")
        print(f"NIM: {data[1]}")
        print(f"Mata Kuliah: {data[2]}")
        print(f>Semester: {data[3]}")
        print()
```

Disini terdapat fungsi dengan nama **tampilkan_data()**. Line-line ini mendefinisikan fungsi **tampilkan_data()**. Fungsi ini digunakan untuk menampilkan semua data mahasiswa. Fungsi ini membaca data dari file teks **data_mahasiswa.txt** dan menampilkannya dalam format tabel.

Line 1 mendeklarasikan fungsi **tampilkan_data()**. Line 2 membuka file teks **data_mahasiswa.txt** dalam **mode r (read)**. Mode r memungkinkan data dari file teks dibaca. Line 3 menyimpan data dari file teks ke dalam variabel **lines**. Line 4 memeriksa apakah file teks kosong. Jika file teks kosong, maka fungsi ini menampilkan pesan bahwa tidak ada data mahasiswa. Line 5-15 menampilkan data mahasiswa dalam format tabel.

```
def update_data():
    nim_target = input("Masukkan NIM mahasiswa yang akan diupdate: ")

    with open("data_mahasiswa.txt", "r") as file:
        lines = file.readlines()

    found = False
    with open("data_mahasiswa.txt", "w") as file:
        for line in lines:
            data = line.strip().split(',')
            if data[1] == nim_target:
                nama = input("Masukkan Nama baru: ")
                matkul = input("Masukkan Mata Kuliah baru: ")
                semester = input("Masukkan Semester baru: ")
                updated_data = f"{nama},{nim_target},{matkul},{semester}\n"
                file.write(updated_data)
                found = True
            else:
                file.write(line)

    if found:
        print("Data mahasiswa berhasil diupdate.")
    else:
        print("Data mahasiswa tidak ditemukan.")
```

Disini terdapat fungsi dengan nama **update_data()**. Fungsi **update_data()** digunakan untuk memperbarui data mahasiswa dalam file teks **data_mahasiswa.txt**. Fungsi ini memiliki dua parameter, yaitu **nim_target** dan **data_baru**. Parameter **nim_target** adalah NIM mahasiswa yang akan diperbarui. Parameter **data_baru** adalah data mahasiswa baru yang akan menggantikan data mahasiswa lama.

Fungsi **update_data()** bekerja dengan cara berikut:

1. Pertama, fungsi ini membuka file teks **data_mahasiswa.txt** dalam mode **r** (read).
2. Kemudian, fungsi ini membaca semua baris dari file teks.
3. Untuk setiap baris, fungsi ini memecah baris tersebut menjadi array.
4. Fungsi ini memeriksa array tersebut untuk melihat apakah NIM mahasiswa yang dicari ditemukan.

5. Jika NIM mahasiswa yang dicari ditemukan, maka fungsi ini mengganti data mahasiswa tersebut dengan data baru.
6. Fungsi ini kemudian menulis semua baris yang telah diubah ke file teks.

Secara singkat, fungsi **update_data()** akan melakukan hal-hal berikut:

1. Membaca data mahasiswa dari file teks
2. Mencari data mahasiswa dengan NIM yang ditentukan
3. Mengganti data mahasiswa yang ditemukan dengan data baru
4. Menulis data mahasiswa yang telah diubah ke file teks

```
def delete_data():
    nim_target = input("Masukkan NIM mahasiswa yang akan dihapus: ")

    with open("data_mahasiswa.txt", "r") as file:
        lines = file.readlines()

    found = False
    with open("data_mahasiswa.txt", "w") as file:
        for line in lines:
            data = line.strip().split(',')
            if data[1] == nim_target:
                found = True
            else:
                file.write(line)

    if found:
        print("Data mahasiswa berhasil dihapus.")
    else:
        print("Data mahasiswa tidak ditemukan.")
```

Disini terdapat fungsi dengan nama **delete_data()**. Fungsi **delete_data()** digunakan untuk menghapus data mahasiswa. Fungsi ini meminta pengguna untuk memasukkan NIM mahasiswa yang akan dihapus. Kemudian, data mahasiswa tersebut dihapus dari file teks **data_mahasiswa.txt**.

- Line 2-3 meminta pengguna untuk memasukkan NIM mahasiswa yang akan dihapus.
- Line 4 membuka file teks data_mahasiswa.txt dalam mode r (read).
- Line 5 menyimpan data dari file teks ke dalam variabel lines.
- Line 6-17 memeriksa apakah data mahasiswa yang dicari ditemukan. Jika ditemukan, maka data mahasiswa tersebut dihapus dari file teks.
- Line 18-21 menampilkan pesan bahwa data mahasiswa telah berhasil dihapus.
- Line 22-23 menampilkan pesan bahwa data mahasiswa tidak ditemukan.

```

def main():
    while True:
        print("====APLIKASI KELOLA DATA MAHASISWA====")
        print("1. Tambah Data")
        print("2. Tampilkan Data")
        print("3. Update Data")
        print("4. Delete Data")
        print("5. Keluar")

        pilihan = input("Pilih menu: ")

        if pilihan == "1":
            tambah_data()
        elif pilihan == "2":
            tampilkan_data()
        elif pilihan == "3":
            update_data()
        elif pilihan == "4":
            delete_data()
        elif pilihan == "5":
            print("Keluar dari aplikasi.")
            break
        else:
            print("Pilihan tidak valid. Silakan pilih lagi.")

```

Disini terdapat fungsi dengan nama **main ()**. Disinilah semua fungsi yang diatas tadi akan dipanggil Ketika user memilih menu antara nomer 1 -5, selain itu tidak valid. Ketika user memilih antara nomer tersebut maka akan langsung terhubung ke salah satu fungsi yang berjalan sesuai dengan nomer tersebut.

```

if __name__ == "__main__":
    main()

```

Bagian kode **if __name__ == "__main__":** digunakan untuk menentukan apakah file Python sedang dijalankan sebagai modul atau sebagai skrip. Jika file Python dijalankan sebagai modul, maka bagian kode tersebut tidak akan dijalankan.

Dalam kasus ini, file Python tersebut dijalankan sebagai skrip, sehingga bagian kode tersebut akan dijalankan. Bagian kode tersebut akan memanggil fungsi **main()**, yang merupakan fungsi utama dari aplikasi.

Secara umum, bagian kode **if __name__ == "__main__":** digunakan untuk menjalankan kode tertentu hanya jika file Python tersebut dijalankan sebagai skrip.