

<p>Nama: Steven Raja P M</p> <p>NIM: 064102400018</p>	 <p>Praktikum Algoritma & Pemrograman</p>	<h1>MODUL 9</h1> <p>Nama Dosen: Binti solihah, S.T, M.KOM</p> <p>Nama Asisten Labratorium:</p> <ol style="list-style-type: none">Yustianas Rombon - 064002300015Vira Aditya Kurniawan - 065002300012
-----------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

File Handling

1. Teori Singkat

File Handling

File Handling merupakan sebuah istilah penanganan file dalam Bahasa Pemrograman. Dalam konsep file handling sendiri kita dapat membuka, menutup, membaca, menulis, menambahkan, serta mengcopy file. Python sendiri dalam memperlakukan beberapa file berbeda, entah itu dalam bentuk biner ataupun sebagai teks. Dalam mengimplementasikannya pada Bahasa pemrograman python sendiri, pemanggilanya, sintaksnya adalah: `file = open('nama file', 'mode')`. Ada tiga jenis mode, yang disediakan Python dan bagaimana file dapat dibuka:

- "`r`", untuk membaca.
- "`w`", untuk menulis.
- "`a`", untuk menambahkan.
- "`r+`", untuk membaca dan menulis

Dan jangan lupa jika kita telah selesai melakukan sesuatu pada file tersebut kita harus memanggil method untuk menutup file tersebut. Pemanggilan methodnya sendiri adalah dengan `file.close()`



Berikut Contoh programnya



```
# Write File
file = open("Test.txt", "w")
file.write("Nadiya Amanda Rizkania")
file.close()

# Read File
file = open("Test.txt", "r")
text = file.read()
print(text)
file.close()

# Add Text
file = open("Test.txt", "a")
file.write(" - 20 Tahun - Jakarta")
file.close()

# Read & Write File
file = open("Test.txt", "r+")
text = file.read()
print(text)
file.close()
```

2. Alat dan Bahan

Hardware : Laptop/PC



Software : Spyder (Anaconda Python)

3. Elemen Kompetensi

a. Latihan pertama

Buatlah sebuah Text File dengan nama Biodata.txt menggunakan implementasi File Handling dengan output seperti dibawah ini yang diinputkan oleh user:

Nama: Nama Kalian

Umur: Umur Kalian

Alamat: Alamat Kalian

Email: Email Kalian

Dosen Wali: Dosen Wali Kalian

Masukkan kedua metode tulis dan metode baca kedalam fungsi agar program lebih terstruktur.

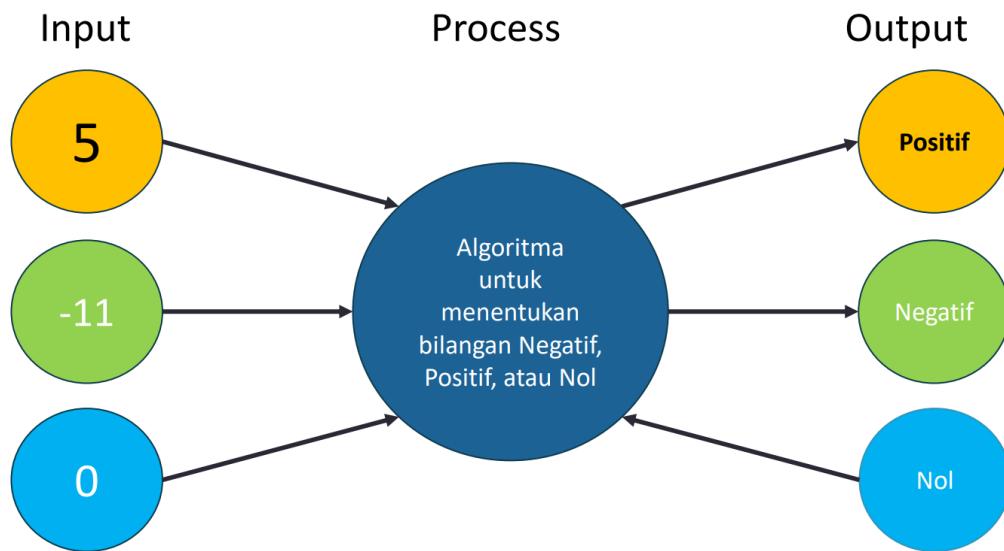
IPO (Input Process Output)

Konsep Dasar Input, Process, dan Output (IPO)

- Konsep input, process, dan output adalah prinsip dasar dalam pemrograman dan pengembangan algoritma.
- Setiap algoritma melibatkan tiga tahap utama: mengambil data masukan (input), melakukan operasi atau pengolahan data (process), dan menghasilkan hasil akhir (output).
- Konsep ini menggambarkan bagaimana algoritma beroperasi untuk memproses informasi.



Gambaran IPO (Menentukan Bilangan)



Notasi Algoritma Flowchart

1. Flowchart adalah representasi visual atau diagram alir yang digunakan untuk menggambarkan langkah-langkah dan urutan proses suatu algoritma atau program.
2. Flowchart menyajikan langkah-langkah dalam bentuk simbol-simbol grafis yang saling terhubung, membantu dalam memvisualisasikan bagaimana informasi mengalir dan bagaimana proses dilakukan.
3. Dalam kaitannya dengan notasi deskriptif, notasi algoritma yang menggunakan flowchart dapat lebih cepat dibaca dan dilihat alur dan hubungannya.

Simbol-simbol pada Flowchart

1. Setiap elemen flowchart dihubungkan oleh garis aliran bertanda panah
2. Garis aliran dimulai dari atas symbol dan keluar dari bagian bawah, kecuali symbol keputusan yang alirannya keluar dari bawah atau samping
3. Aliran bergerak dari atas ke bawah
4. Proses awal dan akhir menggunakan symbol terminal.



... Simbol-simbol pada Flowchart



Terminator yang menandakan *Start* (awal) atau *End* (akhir) program.



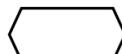
Flow line yang digunakan untuk menunjukkan arah aliran pada program.



Process menunjukkan proses yang dilakukan pada masukan.



Input atau output untuk menunjukkan masukan dan keluaran.



Preparation digunakan untuk membuat deklarasi nilai awal.



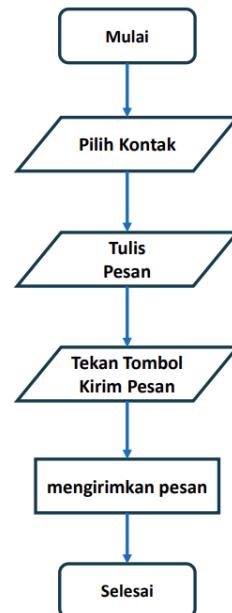
On Page Connector digunakan untuk menghubungkan antar *flowchart*.



Decision menunjukkan keputusan atau kondisi untuk memilih keputusan.

**Contoh sederhana
Penggunaan *flowchart*
untuk menunjukkan algoritma**

**Kasus/Aliran:
Mengirim pesan WhatsApp**



1. Latihan pertama

Buatlah sebuah Text File dengan nama Biodata.txt menggunakan implemetasi File Handling dengan output seperti dibawah ini yang diinputkan oleh user:

Nama: Nama Kalian

Umur: Umur Kalian

Alamat: Alamat Kalian

Email: Email Kalian

Dosen Wali: Dosen Wali Kalian

Masukkan kedua metode tulis dan metode baca kedalam fungsi agar program lebih terstruktur.

Contoh Output

```
Masukkan Nama mu: Amanda
Masukkan Umur mu: 20
Masukkan Alamatmu: Jakarta
Masukkan Emailmu: amanda064002000010@std.trisakti.ac.id
Masukkan Dosen Walimu: Pak Abdul
```

Berikut Ini Data Kamu

Nama: Amanda

Umur: 20

Alamat: Jakarta

Email: amanda064002000010@std.trisakti.ac.id

Dosen Wali: Pak Abdul



IPO

```
def biodata():
    nama = input("Nama: ")
    umur = input("Umur: ")
    alamat = input("Alamat: ")
    email = input("Email: ")
    dosen_wali = input("Dosen Wali: ")

    with open("Biodata.txt", "w") as file:
        file.write(f"Nama: {nama}\n")
        file.write(f"Umur: {umur}\n")
        file.write(f"Alamat: {alamat}\n")
        file.write(f"Email: {email}\n")
        file.write(f"Dosen Wali: {dosen_wali}\n")

    with open("Biodata.txt", "r") as file:
        print("\nIsi file Biodata.txt:")
        print(file.read())

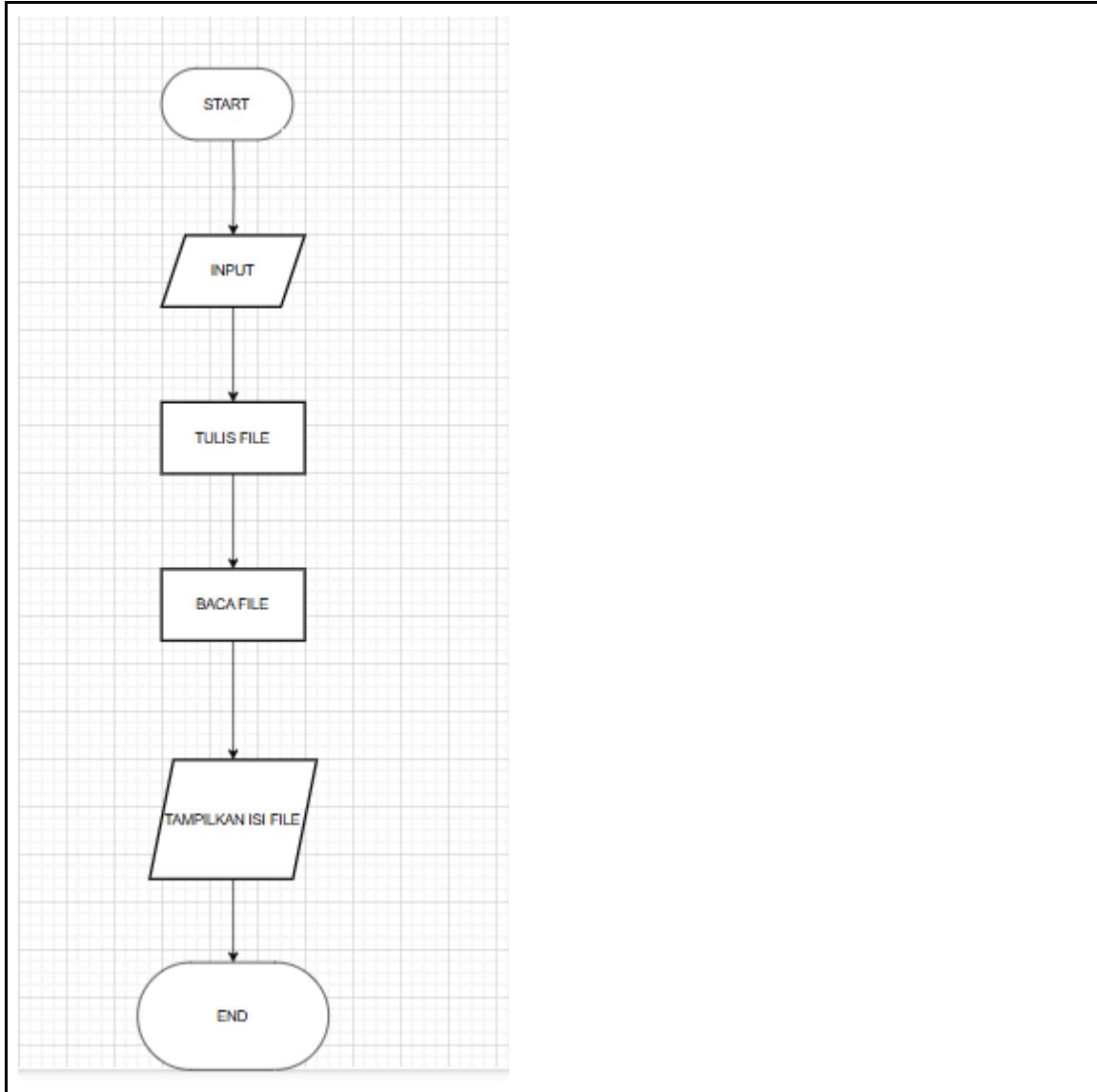
biodata()
```

→ Nama: Steven
Umur: 18
Alamat: stev@gmail.com
Email: stev
Dosen Wali: stev

Isi file Biodata.txt:
Nama: Steven
Umur: 18
Alamat: stev@gmail.com
Email: stev
Dosen Wali: stev



Flowchart



[Source Code](#)



```
def biodata():
    nama = input("Nama: ")
    umur = input("Umur: ")
    alamat = input("Alamat: ")
    email = input("Email: ")
    dosen_wali = input("Dosen Wali: ")

    with open("Biodata.txt", "w") as file:
        file.write(f"Nama: {nama}\n")
        file.write(f"Umur: {umur}\n")
        file.write(f"Alamat: {alamat}\n")
        file.write(f"Email: {email}\n")
        file.write(f"Dosen Wali: {dosen_wali}\n")

    with open("Biodata.txt", "r") as file:
        print("\nIsi file Biodata.txt:")
        print(file.read())

biodata()
```

Output

→ Nama: Steven
Umur: 18
Alamat: stev@gmail.com
Email: stev
Dosen Wali: stev

Isi file Biodata.txt:
Nama: Steven
Umur: 18
Alamat: stev@gmail.com
Email: stev
Dosen Wali: stev



b. Latihan Kedua

Buatlah sebuah program dimana program tersebut dapat membuat file, membaca file dan menambahkan text ke dalam file yang dimana, nama file didapat dari hasil inputan user dan juga data yang ditambahkan kedalam file didapat dari inputan user, implementasikan program kedalam fungsi dan juga implementasikan percabangan serta perulangan pada program seperti yang diajarkan pada materi sebelumnya dimana jika pilihan menu close tidak dipilih (diinputkan oleh user) maka program akan terus berjalan.



IPO

```
def main():
    while True:
        print("\nMenu:")
        print("1. Buat file")
        print("2. Baca file")
        print("3. Tambah data ke file")
        print("4. Tutup program")

        pilihan = input("Pilih menu (1/2/3/4): ")

        if pilihan == '1':
            nama_file = input("Masukkan nama file yang ingin dibuat: ")
            with open(nama_file, 'w') as file:
                print(f"File '{nama_file}' berhasil dibuat.")

        elif pilihan == '2':
            nama_file = input("Masukkan nama file yang ingin dibaca: ")
            try:
                with open(nama_file, 'r') as file:
                    print(f"Isi file '{nama_file}':\n{file.read()}")
            except FileNotFoundError:
                print(f"File '{nama_file}' tidak ditemukan.")

        elif pilihan == '3':
            nama_file = input("Masukkan nama file yang ingin ditambahkan data: ")
            data = input("Masukkan data yang ingin ditambahkan: ")
            with open(nama_file, 'a') as file:
                file.write(data + '\n')
            print(f"Data berhasil ditambahkan ke file '{nama_file}'.")

        elif pilihan == '4':
            print("Program ditutup.")
            break

    else:
```



```
print("Pilihan tidak valid. Silakan pilih menu yang benar.")

if __name__ == "__main__":
    main()

Menu:
1. Buat file
2. Baca file
3. Tambah data ke file
4. Tutup program
Pilih menu (1/2/3/4): 1
Masukkan nama file yang ingin dibuat: StevAlgo9
File 'StevAlgo9' berhasil dibuat.

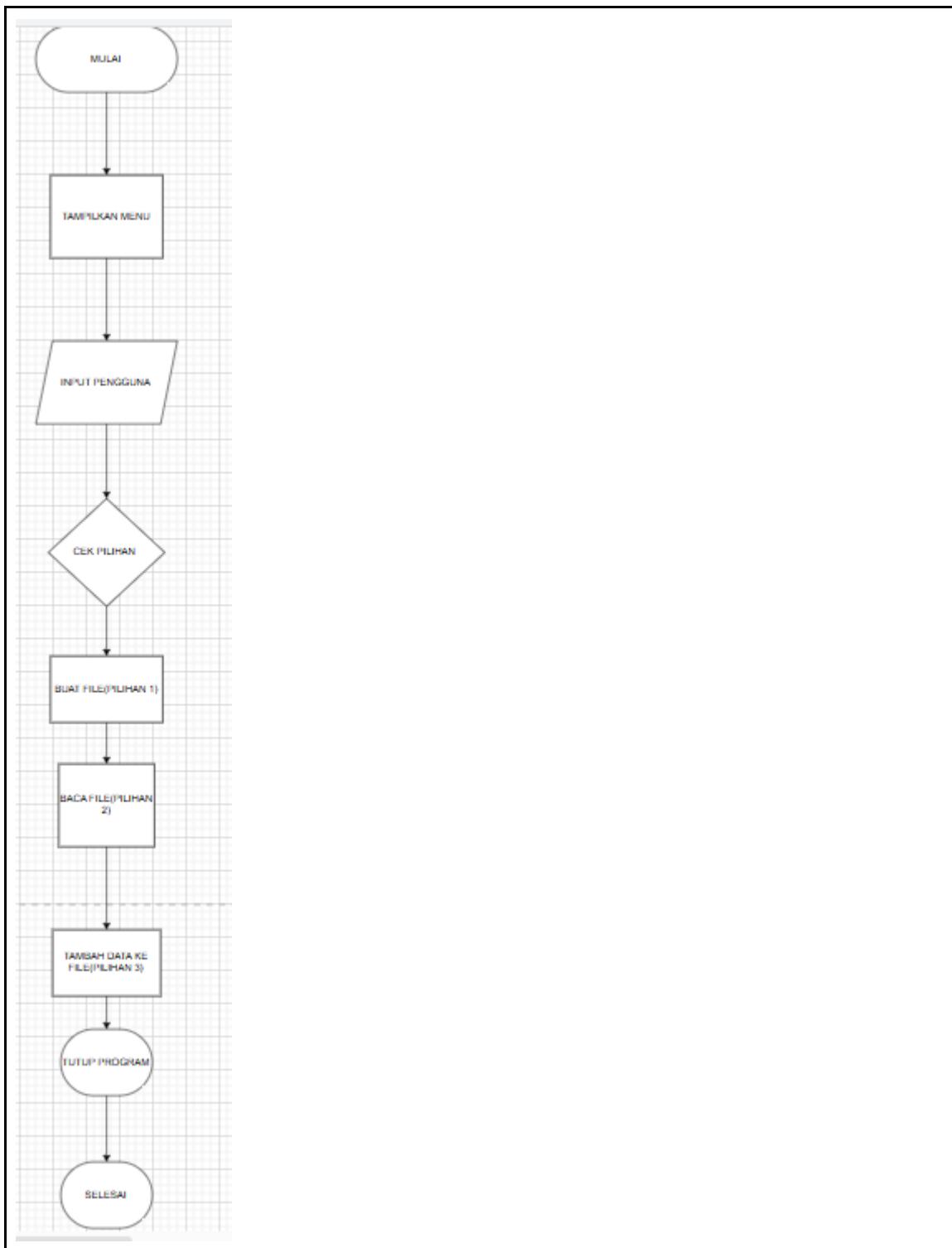
Menu:
1. Buat file
2. Baca file
3. Tambah data ke file
4. Tutup program
Pilih menu (1/2/3/4): 3
Masukkan nama file yang ingin ditambahkan data: StevAlgo9
Masukkan data yang ingin ditambahkan: PrakAlgo ke 9
Data berhasil ditambahkan ke file 'StevAlgo9'.

Menu:
1. Buat file
2. Baca file
3. Tambah data ke file
4. Tutup program
Pilih menu (1/2/3/4): 2
Masukkan nama file yang ingin dibaca: StevAlgo9
Isi file 'StevAlgo9':
PrakAlgo ke 9

Menu:
1. Buat file
2. Baca file
3. Tambah data ke file
4. Tutup program
Pilih menu (1/2/3/4): 4
Program ditutup.
```

Flowchart





Source Code

```
def main():
    while True:
        print("\nMenu:")
        print("1. Buat file")
        print("2. Baca file")
        print("3. Tambah data ke file")
        print("4. Tutup program")

    pilihan = input("Pilih menu (1/2/3/4): ")

    if pilihan == '1':
        nama_file = input("Masukkan nama file yang ingin dibuat: ")
        with open(nama_file, 'w') as file:
            print(f"File '{nama_file}' berhasil dibuat.")

    elif pilihan == '2':
        nama_file = input("Masukkan nama file yang ingin dibaca: ")
        try:
            with open(nama_file, 'r') as file:
                print(f"Isi file '{nama_file}':\n{file.read()}")
        except FileNotFoundError:
            print(f"File '{nama_file}' tidak ditemukan.")

    elif pilihan == '3':
        nama_file = input("Masukkan nama file yang ingin ditambahkan data: ")
        data = input("Masukkan data yang ingin ditambahkan: ")
        with open(nama_file, 'a') as file:
            file.write(data + '\n')
            print(f"Data berhasil ditambahkan ke file '{nama_file}'.")

    elif pilihan == '4':
        print("Program ditutup.")
        break

    else:
```



```
    print("Pilihan tidak valid. Silakan pilih menu yang  
benar.")  
  
if __name__ == "__main__":  
    main()
```

Output

```
Menu:  
  1. Buat file  
  2. Baca file  
  3. Tambah data ke file  
  4. Tutup program  
Pilih menu (1/2/3/4): 1  
Masukkan nama file yang ingin dibuat: StevAlgo9  
File 'StevAlgo9' berhasil dibuat.  
  
Menu:  
  1. Buat file  
  2. Baca file  
  3. Tambah data ke file  
  4. Tutup program  
Pilih menu (1/2/3/4): 3  
Masukkan nama file yang ingin ditambahkan data: StevAlgo9  
Masukkan data yang ingin ditambahkan: PrakAlgo ke 9  
Data berhasil ditambahkan ke file 'StevAlgo9'.  
  
Menu:  
  1. Buat file  
  2. Baca file  
  3. Tambah data ke file  
  4. Tutup program  
Pilih menu (1/2/3/4): 2  
Masukkan nama file yang ingin dibaca: StevAlgo9  
Isi file 'StevAlgo9':  
PrakAlgo ke 9  
  
Menu:  
  1. Buat file  
  2. Baca file  
  3. Tambah data ke file  
  4. Tutup program  
Pilih menu (1/2/3/4): 4  
Program ditutup.
```



4. File Praktikum

Github Repository:

<https://github.com/Stevenraja/PrakAlgo9.git>

5. Soal Latihan

Soal:

1. Apa saja kegunaan file handling dalam sebuah bahasa pemrograman dan kenapa file handling diperlukan dalam sebuah Bahasa pemrograman?
2. Deskripsikan serta narasikan jalannya alur source code program yang sebelumnya telah kalian buat pada Elemen Kompetensi Latihan Kedua!

Jawaban:

1. Kegunaan: penyimpanan data permanen, pengolahan data besar,pemrosesan file log
Diperlukan karna penyimpanan data yang persisten, berbagi data antar program
2. Diawal membuat tampilan menu dan pilihan, lalu proses program berdasarkan menu yang dipilih, program akan selalu menampilkan tampilan menu sampai kita menutup program(pilihan 4).

6. Kesimpulan

- a. Dalam pengerjaan program dengan bahasa pemrograman Python, kita harus benar-benar teliti dalam menginputkan suatu fungsi untuk menampilkan suatu keluaran pada layar dengan sesuai.
- b. Kita dapat mengetahui tentang bagaimana penggunaan file handling untuk mengelola data serta penggunaan fungsi secara efisien dalam melakukan operasi yang berbeda dalam membuat file

7. Cek List (✓)

No	Elemen Kompetensi	Penyelesaian	
		Selesai	Tidak Selesai
1.	Latihan Pertama	✓	
2.	Latihan Kedua	✓	

8. Formulir Umpam Balik

No	Elemen Kompetensi	Waktu Pengerjaan	Kriteria
1.	Latihan Pertama	40 Menit	Menarik
2.	Latihan Kedua	40 Menit	Menarik

Keterangan:

1. Menarik
2. Baik
3. Cukup
4. Kurang

