

<p>Nama: <b>Dimas Alip Priyono</b></p> <p>NIM: <b>064102400032</b></p>	 <p><b>Praktikum Algoritma &amp; Pemrograman</b></p>	<h1>MODUL 9</h1> <p>Nama Dosen: <b>Binti solihah, S.T, M.KOM</b></p> <p>Nama Asisten Labratorium:</p> <ol style="list-style-type: none"><li><b>Yustianas Rombon - 064002300015</b></li><li><b>Vira Aditya Kurniawan - 065002300012</b></li></ol>
<p>Hari/Tanggal: <b>Jumat, 22 November 2024</b></p>		

## 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.

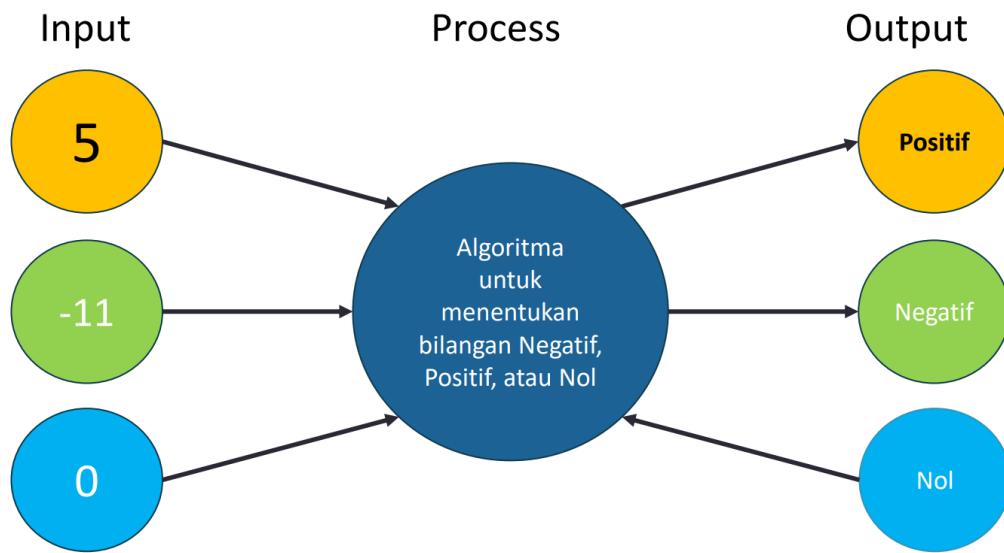
## IPO (Input Process Output)

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.



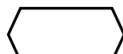
**Flow line** yang digunakan untuk menunjukkan arah aliran pada program.



**Process** menunjukkan proses yang dilakukan pada masukan.



**Input atau output** untuk menunjukkan masukan dan keluaran.



**Preparation** digunakan untuk membuat deklarasi nilai awal.



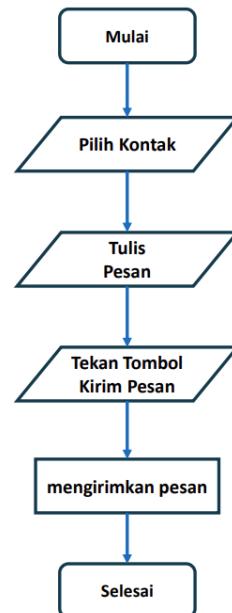
**On Page Connector** digunakan untuk menghubungkan antar *flowchart*.



**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](mailto:amanda064002000010@std.trisakti.ac.id)

Dosen Wali: Pak Abdul



## IPO

### 1. Input

- **Data yang diminta:** Nama, umur, alamat, email, dan dosen wali (dari input pengguna).
- **Fungsi tulis\_biodata:** Menerima data sebagai parameter.
- **Fungsi baca\_biodata:** Tidak memerlukan input langsung.

### 2. Process

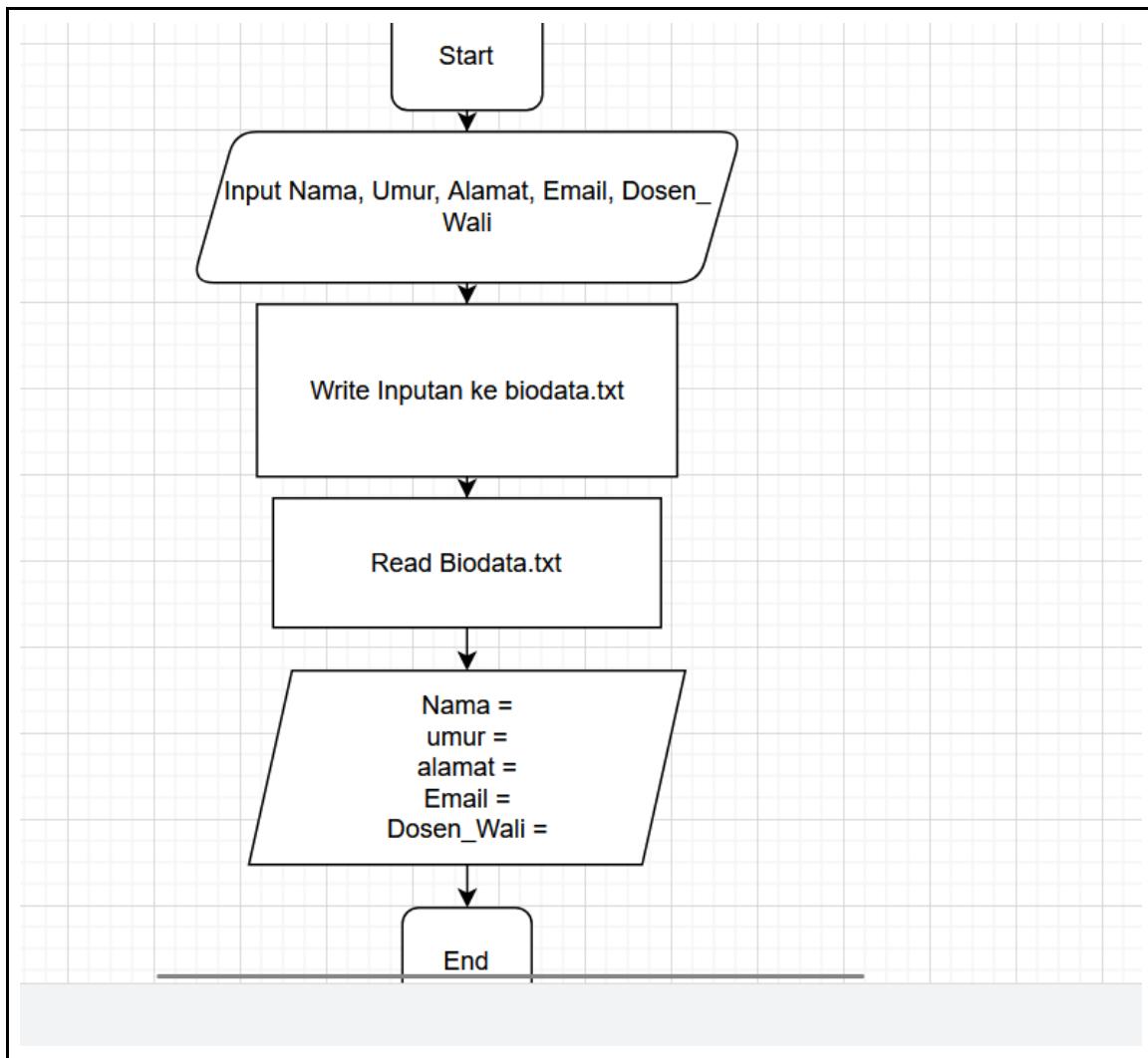
- **Fungsi tulis\_biodata:**
  - Menulis data ke file Biodata.txt dalam format teks.
- **Fungsi baca\_biodata:**
  - Membaca dan mencetak isi file, atau menampilkan pesan jika file tidak ditemukan.

### 3. Output

- **Fungsi tulis\_biodata:** Menampilkan pesan sukses menulis file.
- **Fungsi baca\_biodata:** Menampilkan isi file atau pesan error jika file tidak ada.

Flowchart





### Source Code

```
def tulis_biodata(nama, umur, alamat, email, 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')
    print("Biodata berhasil ditulis ke file.")

def baca_biodata():
    try:
```



```
with open("Biodata.txt", "r") as file:  
    print("\nIsi file Biodata.txt:")  
    print(file.read())  
except FileNotFoundError:  
    print("File Biodata.txt tidak ditemukan.silahkan ditulis dahulu.")  
  
nama = input("Masukkan Nama mu: ")  
umur = input("Masukkan Umur mu: ")  
alamat = input("Masukkan Alamatmu: ")  
email = input("Masukkan Emailmu: ")  
dosen_wali = input("Masukkan Dosen Walimu: ")  
  
tulis_biodata(nama, umur, alamat, email, dosen_wali)  
  
baca_biodata()
```

### Output

```
Masukkan Nama mu: ibnuae  
Masukkan Umur mu: 15  
Masukkan Alamatmu: kalideres  
Masukkan Emailmu: ahmadibnubatutah@gmail.com  
Masukkan Dosen Walimu: bu binti  
Biodata berhasil ditulis ke file.  
  
Isi file Biodata.txt:  
nama: ibnuae  
umur: 15  
alamat: kalideres  
email: ahmadibnubatutah@gmail.com  
dosen wali: bu binti
```

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

- **Input**

- Nama file yang dimasukkan oleh pengguna.
- Pilihan menu (1: Buat file, 2: Baca file, 3: Tambah teks, 4: Keluar).
- Teks tambahan (jika pengguna memilih menu "Tambah teks").

- **Proses**

- **Membuat file:**

- Membuka atau membuat file dengan mode tulis ('w'), sehingga file kosong dibuat.

- **Membaca file:**

- Membuka file dengan mode baca ('r').
- Menangani kesalahan jika file tidak ditemukan (`FileNotFoundException`).
- Membaca isi file dan menampilkan ke layar, atau menampilkan pesan "File kosong".

- **Menambahkan teks ke file:**

- Membuka file dengan mode tambah ('a').
- Menulis teks input pengguna ke dalam file di baris baru.

- **Keluar program:**

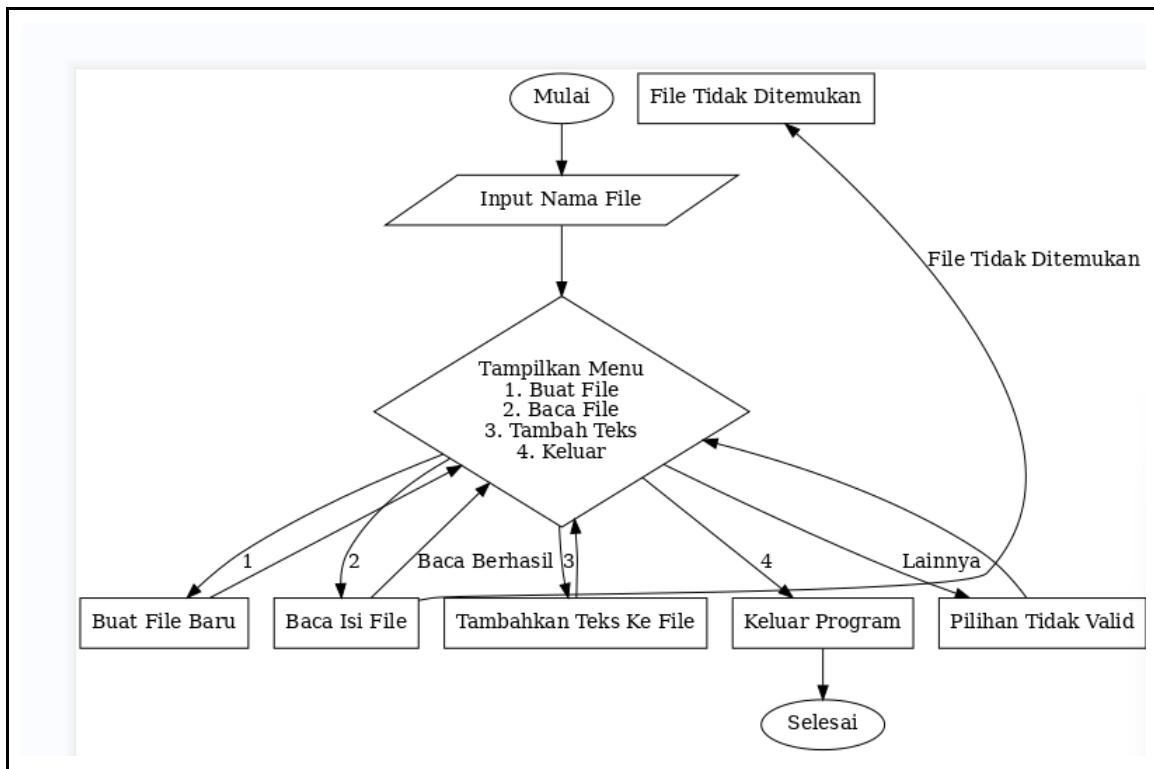
- Mengakhiri perulangan dengan perintah `break`.

- **Output**

- Pesan berhasil membuat file (contoh: File 'nama\_file.txt' berhasil dibuat.).
- Isi file (jika berhasil dibaca) atau pesan "File kosong" atau "File tidak ditemukan".
- Pesan berhasil menambahkan teks (contoh: Teks berhasil ditambahkan.).
- Pesan keluar program (contoh: Program selesai.).
- Pesan untuk pilihan tidak valid (contoh: Pilihan tidak valid. Coba lagi.).

## Flowchart





### Source Code

```
def manage_file():
    print("Program Manajemen File")
    file_name = input("Masukkan nama file (dengan ekstensi): ")

    while True:
        print("\nMenu:\n1. Buat file\n2. Baca file\n3. Tambah
teks\n4. Keluar")
        choice = input("Pilih menu (1/2/3/4): ")

        if choice == '1':
            open(file_name, 'w').close()
            print(f"File '{file_name}' berhasil dibuat.")
        elif choice == '2':
            try:
                with open(file_name, 'r') as file:
                    print("Isi file:\n" + (file.read() or "File
kosong."))
            except FileNotFoundError:
                print(f"File '{file_name}' tidak ditemukan.")
        elif choice == '3':
            text = input("Masukkan teks: ")
            with open(file_name, 'a') as file:
                file.write(text + '\n')
            print("Teks berhasil ditambahkan.")
        elif choice == '4':
            print("Program selesai.")
            break
        else:
            print("Pilihan tidak valid. Coba lagi.")

if __name__ == "__main__":
    manage_file()
```

### Output



```
① Program Manajemen File
Masukkan nama file (dengan ekstensi): fatur
...
Menu:
1. Buat file
2. Baca file
3. Tambah teks
4. Keluar
Pilih menu (1/2/3/4): 1
File 'fatur' berhasil dibuat.

Menu:
1. Buat file
2. Baca file
3. Tambah teks
4. Keluar
Pilih menu (1/2/3/4): 3
Masukkan teks: akuu adalah ai
Teks berhasil ditambahkan.

Menu:
1. Buat file
2. Baca file
3. Tambah teks
4. Keluar
Pilih menu (1/2/3/4):
```



#### 4. File Praktikum

Github Repository:

<https://github.com/Alip1023/prak9algo.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. File handling sangat penting dalam bahasa pemrograman karena memungkinkan aplikasi untuk menyimpan, membaca, mengedit, dan menghapus data yang disimpan dalam file.
2. Program **Manajemen File** yang telah dibuat berfungsi untuk memungkinkan pengguna melakukan beberapa operasi dasar pada file, yaitu membuat file, membaca isi file, menambah teks ke dalam file, dan keluar dari program. Program ini menggunakan konsep **file handling** dasar yang memungkinkan pengguna untuk berinteraksi dengan file secara langsung.

#### 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 Secara keseluruhan, program ini memberikan gambaran praktis tentang bagaimana mengelola file di dalam aplikasi, serta cara memanfaatkan input pengguna untuk berinteraksi dengan file tersebut dalam berbagai cara yang berguna.

#### 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	60 Menit	Menarik
2.	Latihan Kedua	60 Menit	Menarik

Keterangan:

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

