



UNS
UNIVERSITAS
SEBELAS MARET



MODUL PRAKTIKUM PROGRAMA KOMPUTER

**PROGRAM STUDI TEKNIK INDUSTRI
UNIVERSITAS SEBELAS MARET**

**TIM ASISTEN LABORATORIUM
PERANCANGAN DAN OPTIMASI
SISTEM INDUSTRI 2020**

MODUL VIII

FILE, DIREKTORI, MODUL, DAN PAKET

A. TUJUAN

Berikut merupakan tujuan Praktikum Programa Komputer Modul VIII.

1. Memahami teknik mengakses dan mengubah file dan direktori
2. Mampu memahami konsep Modul dan Paket.
3. Mampu menerapkan penggunaan Modul dan Paket.

B. PENDAHULUAN

Membaca, menulis, serta mengolah **File** dan **Direktori** adalah teknik dasar yang harus dipahami dalam pemrograman Python, karena banyak digunakan untuk pengolahan dan pemrosesan data. Memahami cara membaca dan menulis file dengan Python akan membuat python dapat membuat aplikasi yang bisa mengambil dan menyimpan data ke file. Selain itu juga, bertujuan agar dapat lebih mudah memahami beberapa materi Python lanjutan, seperti baca dan *parsing* file JSON, XML, CSV, XLS, dan lainnya.

Modul adalah bagian kode yang ditempatkan ke dalam suatu file terpisah, sedangkan **paket** adalah kumpulan modul yang disatukan ke dalam suatu direktori atau subdirektori tertentu. Python sendiri mengelompokkan kode-kode untuk keperluan spesifik kedalam modul-modul “siap pakai” yang berlainan. Modul-modul ini disebut sebagai **Python Standard Library**. Sebagai contoh, modul math hanya berisi fungsi-fungsi yang berkaitan dengan perhitungan matematis dan modul os hanya berisi fungsi-fungsi yang diperlukan untuk menangani pekerjaan yang berkaitan dengan sistem operasi.

C. FILE PADA PYTHON

Python sudah menyediakan fungsi **open()** untuk membaca dan menulis file, Fungsi tersebut memiliki 2 parameter, yaitu nama file dan mode.



```
file = open("file.txt", "r")
```

Gambar 1 Format Penulisan Objek File

Objek file adalah variabel objek yang menampung isi file. Bertujuan untuk memproses suatu file.

Nama file bisa kita isi langsung apabila file-nya terletak dalam satu direktori dengan skrip python. Namun, apabila terletak di direktori yang berbeda, maka kita harus memberikan alamat path file-nya.

```
obj_file = open("\path\Modul_08\Biodata.txt","r")
```

Gambar 2 Format Penulisan Objek File Dengan Mencari Lokasi

Kemudian untuk parameter mode yang fungsinya untuk menentukan hak akses terhadap file. Ada beberapa mode yang tersedia:

Tabel 1 Parameter Mode dari File

Mode	Ketereangan
"r"	hanya baca saja
"w"	akses untuk menulis file, jika file sudah ada, maka file akan di replace dan diganti dengan yang baru ditulis
"a"	digunakan untuk <i>append</i> atau menambah data ke file, artinya jika sudah ada data dalam file, maka akan ditambahkan dan tidak di-replace
"r+"	digunakan untuk membaca sekaligus menulis data ke file

1. Membaca File per Baris

Berikut ini merupakan program untuk membaca file

```
# buka file
file_biodata = open("Biodata.txt","r")
# baca isi file
print(file_biodata.readlines())
# tutup file
file_biodata.close()
```

Gambar 3 Membaca File dalam Bentuk List

Program ini berfungsi untuk membuka file dengan fungsi **open()**, selanjutnya mencoba membaca isinya per baris dengan method **readlines()**, dan terakhir menutup file dengan **close()** agar dihapus dalam memori.

Saat dieksekusi, kode di atas akan menghasilkan output berupa list, karena kode tersebut menggunakan method **readlines()**

2. Membaca Semua teks dalam file

Selanjutnya mencoba membaca semua teks menggunakan method **read()**.

```
# buka file
file_biodata = open("Biodata.txt", "r")
# baca isi file
print(file_biodata.read())
# tutup file
file_biodata.close()
```

Gambar 4 Membaca File dalam Bentuk Text

3. Perbedaan method **read()** dengan **readlines()**

- Method **read()** membaca seluruh teks dan akan mengembalikan nilai string.
- **readlines()** membaca isi file per baris dan akan mengembalikan nilai berupa list.

Perlu diketahui, method **read()** dan **readlines()** hanya sekali pakai. Artinya hanya dieksekusi sekali saja. Eksekusi pertama, dia akan mengembalikan nilai berdasarkan isi filenya. Eksekusi kedua, dia akan mengembalikan nilai kosong

4. Cara Menulis File di Python

Seperti yang sudah kita ketahui, ada tiga mode yang digunakan bila ingin menulis file, yaitu: **"w"**, **"a"**, dan **"r+"**.

Menulis dengan mode "w"

```
print("Selamat Datang di Program Biodata")
print("-----")

# Mengambil input dari user
nama = input("Nama: ")
umur = input("Umur: ")
alamat = input("Alamat: ")

# Format teks
teks = "Nama: {}\nUmur: {}\nAlamat: {}".format(nama, umur, alamat)

# Buka file untuk ditulis
file_biodata = open("Biodata.txt", "w")

# Tulis teks ke file
file_biodata.write(teks)

# tutup file
file_biodata.close()

print("Data anda berhasil ditulis")
```

Gambar 5 Menulis Ulang File

Menulis dengan mode "a"

Apabila tidak ingin menulis ulang atau menindih file yang sudah ada, bisa menggunakan mode "a" (append) untuk menulisnya.

```
print("Selamat Datang di Program Biodata")
print("-----")

# Mengambil input dari user
nama = input("Nama: ")
umur = input("Umur: ")
alamat = input("Alamat: ")

# Format teks
teks = "\n\nNama: {}\nUmur: {}\nAlamat: {}".format(nama, umur, alamat)

# Buka file untuk ditambahkan
file_biodata = open("Biodata.txt", "a")

# Tulis teks ke file
file_biodata.write(teks)

# tutup file
file_biodata.close()

print("Data anda berhasil ditambahkan")
```

Gambar 6 Menambahkan Text pada File

D. DIREKTORI PADA PYTHON

1. Membuat Direktori

Dalam python, pembuatan direktori baru dilakukan dengan fungsi **mkdir()**. Bentuk umum penggunaan:

os.mkdir>NamaDirektori)

Berikut ialah contoh yang menunjukkan penggunaan fungsi **mkdir()**

```
import os

def main():
    os.mkdir("modul_08")

ifname:"main"
main()
```

Gambar 7 Membuat Direktori Baru

2. Mengubah Nama Direktori

Untuk mengubah nama direktori, kita perlu menggunakan fungsi **rename()** sama seperti pada saat kita mengubah nama file.

Bentuk umum penggunaan:

os.rename>NamaDirektori,>NamaDirektoriBaru)

Berikut ialah contoh yang menunjukkan penggunaan fungsi **rename()**

```
import os

def main():
    os.rename("modul_08","submodul_08")

ifname:"main"
main()
```

Gambar 8 Mengganti Nama Direktori

3. Menghapus Direktori

Untuk mengubah nama direktori, kita perlu menggunakan fungsi **rmdir()**

Bentuk umum penggunaan:

os.rmdir>NamaDirektori)

Contohnya ialah sebagai berikut:

```
import os

def main():
    os.rmdir("submodul_08")

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

Gambar 9 Menghapus Direktori

E. MODUL PADA PYTHON

Modul bisa berupa kode apa saja, bisa berupa fungsi, kelas, maupun kode-kode lain yang tidak dibungkus ke dalam fungsi maupun kelas. Sebagai contoh, Anda bisa mengumpulkan daftar variabel yang akan diperankan sebagai konstanta ke dalam satu modul tersendiri.

```
luas.py > ...
1  import math
2
3  # Fungsi menghitung luas persegi panjang
4  def luaspersegi panjang(p,l):
5      return p*l
6
7  # Fungsi menghitung luas persegi
8  def luaspersegi(s):
9      return s*s
10
11 # Fungsi menghitung luas lingkaran
12 def luaslingkaran (r):
13     return 2*math.pi*r
14
15 #Fungsi menghitung luas segitiga
16 def luassegitiga (a,t):
17     return (a*t)/2
18
```

Gambar 10 Membuat Modul Dalam Python

Simpan kode di atas ke dalam file dengan nama **luas.py**. Nama file akan menjadi nama modul. Dengan demikian telah dimiliki modul dengan nama **luas.py**, yang siap digunakan di dalam kode program manapun.

F. PERINTAH IMPORT (MODUL)

Untuk dapat menggunakan kode-kode dalam modul, kita perlu mengimport atau menyertakan nama modul tersebut dengan perintah *import*. Perlu diingat bahwa modul harus disimpan pada folder yang sama agar dapat di-*import*. Bentuk umum penggunaan perintah *import* adalah:

```
import NamaModul
```

Gambar 11 Sintaks Import Modul

Berikut contoh kode program yang akan menunjukkan cara kerja penggunaan modul.

```
import luas

# Diketahui panjang dan lebar dari sebuah persegi panjang sebagai berikut
p = 2
l = 4
# Hitunglah luas dari persegi panjang

hasil = luas.luaspersegi panjang(p,l)
print(hasil)
```

[4] ✓ 1.8s

... 8

Gambar 12 Program Menggunakan Modul

Berdasarkan tampilan program di atas, langkah pertama yaitu melakukan import modul. Dikarenakan kita mempunyai modul bernama **luas** sehingga kita harus mengetikkan sintaks **import luas** kemudian mendefinisikan variable yang diketahui. Setelah itu panggil fungsi di dalam modul dengan sintaks **luas.luaspersegi panjang(p,l)**.

G. PERINTAH FROM...IMPORT (MODUL)

Ketika menggunakan perintah *import*, semua kode yang ada di dalam modul akan diimport. Pada saat tertentu, kita hanya ingin mengimpor kode tertentu saja. Untuk melakukan hal ini, kita dapat menggunakan perintah **from...import**. Contohnya sebagai berikut:


```
# mengimpor fungsi luaslingkaran
# dari modul luas

from luas import luaslingkaran

r = 10

hasil = luaslingkaran(r)
print(f"Luas lingkaran yaitu : {hasil}")

✓ 0.1s
```

Gambar 13 Mengakses Sebuah Fungsi Dalam Sebuah Modul

Berdasarkan tampilan program di atas, kita hanya ingin mengakses salah satu fungsi yaitu fungsi **luaslingkaran** yang berada di dalam modul **luas** dengan cara mengetikkan sintaks **from luas import luaslingkaran**.

H. MEMBUAT NAMA ALIAS/NAMA LAIN (MODUL)

Jika ingin mengimpor modul yang memiliki judul yang panjang atau sulit dibaca, anda dapat membuat alias atau nama lain dari modul tersebut. Bentuk umum:

```
import NamaModul as NamaAlias💡
```

Gambar 14 *Sintaks* Mengubah Nama Modul Menjadi Nama Alias

```
import luas as L

# Diketahui panjang dan lebar dari sebuah persegi panjang sebagai berikut
p = 2
l = 4
# Hitunglah luas dari persegi panjang

hasil = L.luaspersegipanjang(p,l)
print(hasil)

✓ 0.0s
```

Gambar 15 Mengubah Nama Modul Luas Menjadi L

```
# mengimpor fungsi luaslingkaran
# dari modul luas
# nama alias LK

from luas import luaslingkaran as LK

r = 10

hasil = LK(r)
print(f"Luas lingkaran yaitu : {hasil}")

✓ 0.1s
```



Gambar 16 Mengubah Nama Fungsi Luas Lingkaran Menjadi LK

Berdasarkan tampilan-tampilan program di atas, kita dapat mengubah nama modul dan fungsi yang kita *import* menjadi lebih sederhana dengan cara **import luas as L** atau bisa dengan cara **from luas import luaslingkaran as LK**.

I. PAKET PADA PYTHON

Paket (Package) adalah suatu direktori yang digunakan untuk menyimpan kumpulan modul (file .py). Di dalam direktori tersebut harus terdapat satu file khusus dengan nama `__init__.py`. File ini berisi nama-nama file di dalam direktori yang akan dibungkus ke dalam paket. Berikut langkah-langkah yang diperlukan untuk membuat suatu paket di dalam python.

1. Buat subdirektori di dalam direktori kerja anda dengan nama **matematika**.
2. Pindahkan (*move*) file **luas.py** ke dalam subdirektori baru tersebut.
3. Buatlah file baru dengan nama **__init__.py** dan simpan ditempat yang sama dengan file **luas.py**
4. Isikan kode berikut ke dalam file **__init__.py**

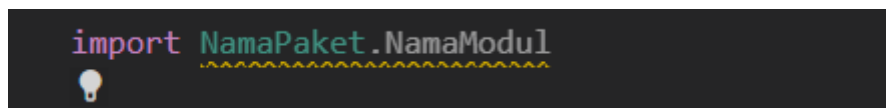
```
matematika >  __init__.py
1 from import luas 
```

Gambar 17 Mengakses Modul Dalam Sebuah Paket

5. Simpan file.


J. MENGGUNAKAN PAKET

Paket yang sudah dibuat dapat digunakan menggunakan perintah *import*, dengan cara berikut.



```
import NamaPaket.NamaModul
```

Gambar 18 *Import* Sebuah Modul Dalam Sebuah Paket



```
import matematika.luas

def main():
    # segitiga
    a = 6
    t = 10

    hasil = matematika.luas.luassegitiga(a,t)

    print("Menghitung luas segitiga")
    print(f"Luas segitiga adalah : {hasil}")

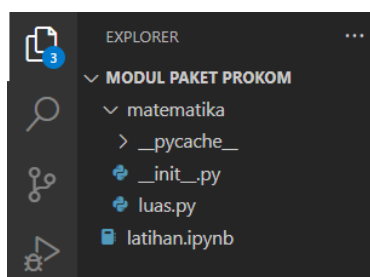
if __name__ == "__main__":
    main()
```

[10] ✓ 0.1s

... Menghitung luas segitiga
Luas segitiga adalah : 30.0

Gambar 19 Program Mengakses Modul Luas Dalam Paket Matematika

Berdasarkan tampilan program di atas, kita dapat mengakses sebuah fungsi bernama **luassegitiga** dalam sebuah modul yang bernama **luas** dimana modul tersebut berada di dalam sebuah paket bernama **matematika** dengan cara mengetikkan sintaks **import matematika.luas** kemudian memanggilnya dengan cara **matematika.luas.luassegitiga(a,t)**.



Gambar 20 Tampilan *Eksplorer*

Gambar di atas merupakan tampilan *eksplorer* yang seharusnya.