

COMPUTER PROGRAMING 2

MODULE &
PACKAGE

MODUL 5

2023

DAFTAR ISI

DAFTAR ISI	2
MODULE & PACKAGE	3
A. Pemograman Secara Modular	3
1. Keuntungan Pemrograman Modular	3
B. Pengertian Modul Pada Python	3
1. Membuat Modul	4
2. Mengimpor Modul Dengan Statemen import	5
3. Menyembunyikan Item Agar Tidak Bisa Diimport	6
C. Pengertian Paket Pada Python	7
1. Membuat Paket	7
2. Mengimpor Paket	9
3. Impor Paket/Modul dengan Alias	9
4. Impor Paket/Modul secara Kustom	10
5. Menampilkan Path/Nama File dari Modul	10
D. Beberapa Paket dan Modul Bawaan Python	11
E. Poin-Poin Penting	11

MODULE & PACKAGE

Dalam pertemuan ini, kita akan membahas hal-hal berkaitan dengan modul dan paket pada python, apa saja manfaatnya dan bagaimana sih cara mengimplementasikannya.

Akan tetapi sebelum mulai kita akan membahas terlebih dahulu sedikit tentang pemrograman secara modular. Karena dari pendekatan modular lah kata “**modul**” merujuk.

A. Pemograman Secara Modular

Dalam dunia pemrograman, pendekatan modular berarti proses memecah program super besar menjadi beberapa bagian atau kecil yang terpisah, terorganisir, terkelola dengan lebih rapi sesuai dengan tugasnya masing-masing

1. Keuntungan Pemrograman Modular

Ada beberapa keuntungan jika kita menerapkan pendekatan modular, di antaranya:

- **Simplicity:** kode program kita menjadi lebih sederhana.
- **Maintainability:** kode program kita menjadi lebih mudah di-maintain atau ‘dipelihara’.
- **Reusability:** potongan-potongan kode program bisa digunakan dari berbagai tempat membuatnya menjadi lebih reusable.

Pada python, terdapat 3 komponen penting penyusun modularitas:

- Fungsi
- Modul
- Paket

B. Pengertian Modul Pada Python

Kita telah membahas tentang fungsi pada seri tutorial python dasar. Ia merupakan blok kode tertentu yang memiliki nama, dan bertugas untuk menyelesaikan satu set perintah tertentu, serta bisa dipanggil dari bagian program mana pun.

Hanya saja, kita belum pernah mencoba membuat fungsi pada suatu file, dan memanggilnya pada file yang lain.

Oke. Kita kenalan dengan modul dulu.

Apa itu **module** pada python? Sederhananya, module pada python **adalah** sebuah file berekstensi .py yang berisi skrip python. Nama dari modul adalah nama dari file itu sendiri. Misal kita memiliki file bernama 'matematika.py', maka kita telah membuat sebuah modul bernama 'matematika'.

Dan modul sendiri bisa memiliki berbagai macam isi, baik itu fungsi, class, maupun variabel.

1. Membuat Modul

Kita langsung praktikkan saja ya.

Buat dahulu file seperti ini:

```
src/  
├─ skrip-utama.py  
└─ matematika.py
```

Dimana "src" adalah direktori dan didalamnya terdapat skrip utama dan satunya sebagai modul.

Rencana:

- File skrip-utama.py adalah file yang akan kita running.
- File matematika.py adalah file yang akan kita jadikan modul.

Karena itu, buka file matematika.py kemudian tuliskan kode program berikut:

```
pi = 22 / 7  
  
def luas_persegi (sisi):  
    return sisi * sisi  
  
def luas_lingkaran (radius):  
    return pi * radius * radius
```

Penjelasan:

Pada modul matematika di atas, kita membuat 3 buah item:

- variabel pi dengan nilai hasil bagi dari 22/7
- fungsi luas_persegi
- fungsi luas_lingkaran

Kita akan mencoba memanggil 3 buah item tersebut dari file yang berbeda.

2. Mengimpor Modul Dengan Statemen import

Jika kita ingin memanggil fungsi atau apa pun yang ada pada file lain, maka kita harus mengimpornya terlebih dahulu.

Terdapat beberapa cara untuk mengimpor.

a. Impor semua

Yang pertama adalah: impor semua. Alias kita mengimpor semua item yang ada pada suatu modul tanpa pengecualian.

Gimana caranya?

Silakan buka file **skrip-utama.py** kemudian tuliskan kode program berikut:

```
import matematika

print("PI:", matematika.pi)
print(matematika.luas_lingkaran(radius=21))
print(matematika.luas_persegi(sisi=12))
```

Pada kode program di atas, kita telah memanggil modul matematika dengan perintah:

```
import matematika
```

Hal ini akan membuat kita memiliki akses terhadap modul tersebut dan apa yang ada di dalamnya. Oleh karena itu ketika kita jalankan program, kita akan mendapatkan output seperti berikut:

```
PI: 3.142857142857143
1386.0
144
```

b. Impor secara eksplisit

Selain mengimpor semua item, kita juga bisa mengimpor hanya item-item tertentu saja secara eksplisit. Misalkan pada modul matematika, kita hanya ingin mengimpor fungsi **luas_persegi()** dan **luas_lingkaran()** saja, kita bisa melakukannya seperti ini:

```
from matematika import luas_persegi, luas_lingkaran

print(luas_persegi(20))
print(luas_lingkaran(10))
```

3. Menyembunyikan Item Agar Tidak Bisa Diimport

Selain dengan mendefinisikan satu-satu, kita juga bisa melakukan **import *** seperti ini:

```
from matematika import *

print('nilai pi:', pi)
```

Perintah di atas akan mengimport semua item yang ada pada modul matematika, kecuali item yang namanya diawali dengan `_`.

Untuk mencobanya, kita bisa membuat sebuah fungsi di dalam modul matematika dengan nama **`_luas_segitiga()`**.

```
...

def _luas_segitiga (alas, tinggi):
    return 0.5 * alas * tinggi
```

Panggil dengan cara `import *` pada file skrip-utama.py.

```
from matematika import *

print(luas_persegi(20))
print(luas_lingkaran(10))
print(_luas_segitiga(20, 10))
```

Semuanya akan terimport, kecuali satu fungsi saja yaitu **`_luas_segitiga`**, karena ia diawali dengan **underscore**:

```
Exception has occurred: NameError
name '_luas_segitiga' is not defined
```

C. Pengertian Paket Pada Python

Paket pada python **adalah** sebuah cara untuk mengelola dan mengorganisir modul-modul python dalam bentuk direktori, memungkinkan sebuah modul untuk diakses menggunakan “**namespace**” dan dot lokasi.

Secara sederhana, paket (atau *packages*) pada python adalah direktori biasa, yang di dalamnya terdapat modul-modul python.

Bedanya, di dalam direktori tersebut, harus ada satu file bernama **__init__.py** yang memberitahukan interpreter bahwa direktori tersebut adalah paket yang berisi modul-modul.

Kita langsung praktekkan saja ya.

1. Membuat Paket

Anggap saja kita akan membuat aplikasi untuk menghitung luas bangun datar dan juga volume bangun ruang.

Bangun datar di antaranya:

- persegi
- lingkaran
- segitiga

Bangun ruang di antaranya:

- kubik
- trapesium
- bola

Dan kita akan membuat setiap bangun datar atau setiap bangun ruang menjadi satu modul tersendiri. Sehingga, kita akan memiliki total 6 modul. Dan dari 6 modul tersebut akan kita bagi menjadi 2 paket. Sehingga penampilan struktur programnya adalah berikut:

```
src/
├── luas
│   ├── __init__.py
│   ├── lingkaran.py
│   ├── persegi.py
│   └── segitiga.py
├── volume
│   ├── __init__.py
│   ├── bola.py
│   ├── kubik.py
│   └── trapesium.py
└── skrip-utama.py
```

Ingat, setiap direktori yang akan kita jadikan sebagai paket, harus memiliki file `__init__.py`. Apa isi file tersebut? Kita bisa kosongkan saja, tidak perlu diisi apa pun.

Untuk masing-masing file, silakan buat fungsi **luas_<nama-bangun-datar>** dan **volume_<bangun-ruang>**. Kita bisa kosongkan saja isi fungsinya dengan menggunakan statemen `pass`.

a. File `luas/lingkaran.py`

```
def luas_lingkaran():  
    pass
```

b. File `luas/persegi.py`:

```
def luas_persegi():  
    pass
```

c. File `luas/segitiga.py`:

```
def luas_segitiga():  
    pass
```

d. File `volume/bola.py`:

```
def volume_bola():  
    pass
```

e. File `volume/kubik.py`:

```
def volume_kubik():  
    pass
```

f. File `volume/trapesium.py`:

```
def volume_trapesium():  
    pass
```

2. Mengimpor Paket

Kita bisa mengimpor paket dengan cara yang sama seperti mengimpor modul biasa. Kita hanya perlu menambahkan <nama-direktori> plus dot (.) untuk pemisahannya.

Perhatikan contoh berikut:

```
from luas.segitiga import luas_segitiga
from luas import lingkaran, persegi
from volume.kubik import volume_kubik
import volume.bola
from volume.trapesium import *

# panggil fungsi masing-masing
```

NB: paket juga bisa bersifat nested alias kita bisa membuat paket yang ada di dalam paket lain dan seterusnya.

3. Impor Paket/Modul dengan Alias

Dengan semakin banyaknya modul dan paket, tidak menutup kemungkinan akan adanya item-item dengan nama yang sama.

Misalkan pada modul lingkaran terdapat fungsi dengan nama **hitung_luas()**, dan di modul persegi juga ada fungsi dengan nama yang sama.

Bagaimana cara memanggilnya?

Kita bisa memanfaatkan fitur alias dengan statemen **as**.

Perhatikan contoh berikut:

File **lingkaran.py**:

```
def luas (radius):
    return 22 / 7 * radius * radius
```

File **persegi.py**:

```
def luas (sisi):
    return sisi * sisi
```

File skrip-utama.py:

```
from persegi import luas as luas_persegi
from lingkaran import luas as luas_lingkaran

print(luas_persegi(10))
print(luas_lingkaran(21))
```

Output:

```
100
1386.0
```

4. Impor Paket/Modul secara Kustom

Selain menggunakan alias, kita juga bisa menggunakan percabangan if else.

Perhatikan contoh berikut:

```
bidang = 'lingkaran'

if bidang == 'lingkaran':
    from lingkaran import luas
elif bidang == 'persegi':
    from persegi import luas

# hitung luas
print(luas(20))
```

5. Menampilkan Path>Nama File dari Modul

Selain itu, kita juga bisa menampilkan lokasi file dari modul yang kita import.

Perhatikan contoh berikut, kita akan coba mengetahui di mana lokasi dari modul lingkaran yang telah kita buat.

```
import lingkaran

print(lingkaran.__file__)
```

Output (tentunya berbeda-beda tergantung lokasi file)

```
/home/ibnuzakariyya/workspaces/python/python-menengah/03-
modul/07-lokasi-modul/lingkaran.py
```

Ya mungkin meskipun informasi seperti ini tidak terlalu penting untuk aplikasi kita, tapi bisa cukup penting untuk debugging.

D. Beberapa Paket dan Modul Bawaan Python

Selain kita bisa membuat modul dan paket sendiri, sebenarnya python pun secara bawaan juga memiliki banyak sekali modul default yang akan memudahkan tugas-tugas kita sebagai developer.

Contohnya seperti modul **datetime**, **math**, **time**, **json**, dan lain sebagainya.

Cara memanggilnya pun sama saja:

```
import math
import datetime

print('Tanggal dan waktu sekarang: ',
      datetime.datetime.now())
print('Nilai konstanta pi: ', math.pi)
```

Kode program di atas akan menghasilkan output sebagai berikut:

```
Tanggal dan waktu sekarang:  2021-02-19 13:32:55.622208
Nilai konstanta pi:  3.141592653589793
```

E. Poin-Poin Penting

Beberapa poin penting yang bisa kita simpulkan dari pertemuan kali ini:

- Pendekatan pemrograman modular adalah satu konsep di mana kita memecah bagian besar program menjadi bagian-bagian kecil yang lebih bermakna.
- Penerapan modular programming membuat kode program lebih terstruktur dan mudah diorganisir.
- Modul pada python adalah sebuah file berekstensi .py yang berisi kode program python.
- Modul bisa kita panggil dari file yang lain.
- Paket adalah sebuah direktori yang memiliki satu file `__init__.py` dan di dalamnya berisi modul-modul python.
- Python memiliki modul dan paket-paket secara default.

=== To be Continue ===