



Laporan Program Python Implementasi Penggunaan Class dan Library Turtle

Mata Kuliah:

Pemrograman Berorientasi Objek

Oleh:

Gerry Moeis Mahardika Dwi Putra

23091397164

2023E

<https://github.com/gerrymoeis/pbo-1>



PROGRAM STUDI MANAJEMEN INFORMATIKA FAKULTAS VOKASI

UNIVERSITAS NEGERI SURABAYA

2024



Laporan Bagian 1: Pembuatan Class Mahasiswa untuk Merepresentasikan dan Menampilkan Data Mahasiswa

Langkah	Praktikum
1.	<pre>class Mahasiswa:</pre> <p>Pertama kita membuat class Mahasiswa sebagai representasi objek Mahasiswa di dunia nyata.</p> <pre>class Mahasiswa: def update(self, nama, kelas, nim, dpa, jurusan="D4 Manajemen Informatika", fakultas="Vokasi", kampus="Universitas Negeri Surabaya"): self.nama = nama self.kelas = kelas self.nim = nim self.dpa = dpa self.jurusan = jurusan self.fakultas = fakultas self.kampus = kampus</pre> <p>Selanjutnya, disini sebelum membuat method <code>__init__()</code>, saya ingin membuat method kustom yaitu <code>update()</code>. Tujuannya adalah agar kita dapat memanggil fungsi ini lagi untuk keperluan inisiasi maupun update data nantinya.</p> <p>Beberapa di antara parameter di atas saya buat dalam bentuk default parameter sehingga bila user tidak menentukan isiannya maka akan ditentukan secara default pada value parameter di atas.</p>

	<pre>def __init__(self, *args): self.update(*args)</pre> <p>Dengan dibuatnya method update() sebelumnya, sekarang untuk special method utama yang diperlukan yaitu __init__() kita tinggal memanggil method update() tersebut menggunakan *args parameter, yang mana artinya memanggil method update dengan list argumen-argumen yang perlu disesuaikan dalam method tanpa perlu menspesifikasikannya satu per satu.</p>
2.	<pre>def printData(self): for key, value in vars(self).items(): if key == "fakultas": print(f"{key.title()}\t:\t{value}") else: print(f"{key.title()}\t:\t:\t{value}") print()</pre> <p>Nah untuk tugas utama yang diberikan yaitu tampilkan data disini saya membuat method printData() yang mana melakukan looping dari seluruh data atau atribut yang dimiliki class Mahasiswa sehingga kita dapat menampilkannya tanpa perlu dispesifikasi satu per satu.</p> <p>Untuk if else statement di atas hanya untuk pemformatan teks agar hasil output dapat terlihat lebih rapi.</p> <pre>def editData(self, **kwargs): data = vars(self) for key, value in kwargs.items(): data[key] = value self.update(**data)</pre> <p>Method terakhir yang saya tambahkan adalah editData(), tujuan dari pembuatan method ini adalah untuk menunjukkan penerapan *args dan **kwargs dalam fitur-fitur yang ditawarkan konsep class di Python.</p> <p>Penjelasan sederhana untuk cara kerja method ini dapat dipahami seperti mengambil keyword argument yang di input oleh user lalu melakukan edit menggunakan konsep dictionary melalui vars function, dan terakhir memanggil method update untuk menginisiasi ulang value atribut yang di update.</p>



3.

```
mahasiswa_1 = Mahasiswa("Gerry", "2023E", "164", "Pak Agung")
mahasiswa_1.printData()
mahasiswa_1.editData(nama="Gerry Moeis", kelas="2023F")
mahasiswa_1.printData()
```

Setelah kita membuat class Mahasiswa atau blueprint mahasiswa tersebut. Sekarang kita praktekkan dalam membuat objek dari class itu. Untuk input datanya saya memakai data diri saya. Setelah itu ditampilkan menggunakan method printData()

Lalu setelah itu kita coba untuk editData() dengan menginput keyword arguments berupa nama dan kelas yang akan di update. Setelah itu di print lagi.

```
(env) D:\Gerry\Programmer\Best Terbaik(2024)\Python Programming\OOP\Tugas 1>mahasiswa_gerry.py
Nama      :      Gerry
Kelas    :      2023E
Nim       :      164
Dpa       :      Pak Agung
Jurusan   :      D4 Manajemen Informatika
Fakultas  :      Vokasi
Kampus    :      Universitas Negeri Surabaya

Nama      :      Gerry Moeis
Kelas    :      2023F
Nim       :      164
Dpa       :      Pak Agung
Jurusan   :      D4 Manajemen Informatika
Fakultas  :      Vokasi
Kampus    :      Universitas Negeri Surabaya
```

Dapat dilihat di atas hasil output dari program Python yang telah saya buat ini. Semua fungsi yang diinginkan berjalan dengan baik juga ditampilkan dalam format yang rapi. Jadi itulah bagaimana saya mengerjakan tugas dalam membuat dan merepresentasikan objek Mahasiswa dalam konsep OOP.



Laporan Bagian 2: Penggunaan Library Turtle dalam Visualisasi Penerapan Konsep OOP dengan Membuat Bentuk Segitiga

Langkah	Praktikum
1.	<pre>import turtle import pandas as pd from mahasiswa_gerry import Mahasiswa</pre> <p>Untuk bagian 2, disini pertama kita import library yang diperlukan, pastinya turtle, lalu disini saya juga menggunakan pandas dan terakhir mengimport class Mahasiswa saya (hanya untuk eksperimen).</p> <pre>class Pena: def __init__(self, mahasiswa, color): self.mahasiswa = Mahasiswa(*mahasiswa) self.t = turtle.Turtle() self.t.shape("classic") self.t.color(color) self.t.speed(7)</pre> <p>Disini saya membuat class Pena sebagai representasi pena di dunia nyata. Class Pena mengambil 2 parameter utama yaitu data mahasiswa dalam bentuk list dan color sebagai warna pena yang diinginkan.</p> <p>Atribut yang dibuat di antaranya self.mahasiswa dan self.t yang dibuat dari class Turtle(). Saya juga memberikan beberapa kustomisasi dalam hal bentuk, warna, dan kecepatannya.</p> <pre>def gambar_segitiga(self, ukuran): self.t.penup() self.t.forward(-400) self.t.pendown() self.t.begin_fill() for _ in range(3): self.t.forward(ukuran) self.t.left(120) self.t.end_fill()</pre> <p>Untuk method yang saya buat sebagai poin utama tugas ini ialah gambar_segitiga(), method ini mengambil 1 parameter yaitu ukuran. Cara kerja method ini yaitu dengan memindah pena terlebih dahulu ke tempat yang pas dengan penup, forward, dan pendown. Lalu diblok warna dengan begin_fill(). Dan fungsi utama berupa looping sebanyak 3 untuk membuat segitiga dari garis sesuai ukuran input.</p>



2.	<pre>df = pd.read_excel("data_mi23.xlsx", "Table 1") data = df.values.tolist()</pre> <p>Setelah class Pena dibuat. Sekarang kita masuk ke penerapan. Disini saya melakukan eksperimen untuk mengambil data-data mahasiswa yang telah saya sederhanakan dan bersihkan pada excel menggunakan library Panda.</p> <pre>mahasiswa_mi = [] for mahasiswa in data: color = "red" if mahasiswa[1] == "2023A": color = "purple" if mahasiswa[1] == "2023B": color = "blue" if mahasiswa[1] == "2023C": color = "green" if mahasiswa[1] == "2023D": color = "orange" if mahasiswa[1] == "2023E": color = "gold" if mahasiswa[1] == "2023F": color = "red" mahasiswa_mi.append(Pena(mahasiswa, color))</pre> <p>Untuk menampung objek Pena yang dibuat, saya membuat list mahasiswa_mi. Lalu melakukan looping pada data excel tersebut dan mengecek untuk menentukan warna pada masing-masing kelas yang ditemukan. Sehingga pena yang dibuat dapat unik dan memiliki warna sesuai kelas masing-masing.</p>

3.

```
screen = turtle.Screen()
screen.bgcolor("white")
```

Terakhir kita konfigurasi layar dengan library turtle. Kita beri background color pada layar tersebut dengan warna putih.

```
for i, pena in enumerate(mahasiswa_mi):
    pena.gambar_segitiga(50)
    if i < len(mahasiswa_mi) - 1:
        mahasiswa_mi[i+1].t.penup()
        mahasiswa_mi[i+1].t.goto(75*(i+1), 0)
        mahasiswa_mi[i+1].t.pendown()
```

Kemudian kita looping terhadap list mahasiswa_mi sebelumnya dan mengambil 2 variable berupa index dan pena. Untuk masing-masing pena kita panggil method gambar_segitiga() tersebut dengan ukuran yang disesuaikan.

Lalu saya beri kondisi untuk merubah posisi tiap pena agar tidak saling menutupi satu sama lain melainkan menggambar segitiga pada posisi masing-masing sesuai index.

```
screen.exitonclick()
```

Untuk memastikan screen tidak langsung menutup, saya memanggil method exitonclick() pada screen turtle yang dideklarasikan sebelumnya.

Python Turtle Graphics

— □ ×



Dan itulah hasil akhir untuk visualisasi objek mahasiswa (dalam hal ini pewarnaan masing-masing kelas agar terlihat unik) dengan bentuk segitiga.



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI
UNIVERSITAS NEGERI SURABAYA
Kampus Unesa 1, jalan Ketintang Surabaya 60231
Laman: <https://vokasi.unesa.ac.id/> E-mail: vokasi@unesa.ac.id

--#BeraksiBerprestasiBersinergi--