

LAPORAN PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK (PBO)

PRAKTIKUM 5



2411102441052

Angga Maulana Saputra

FAKULTAS SAINS DAN TEKNOLOGI

PROGRAM STUDI S1 TEKNIK INFORMATIKA

UNIVERSITAS MUHAMMADIYAH KALIMANTAN TIMUR

Latihan Praktikum 5:

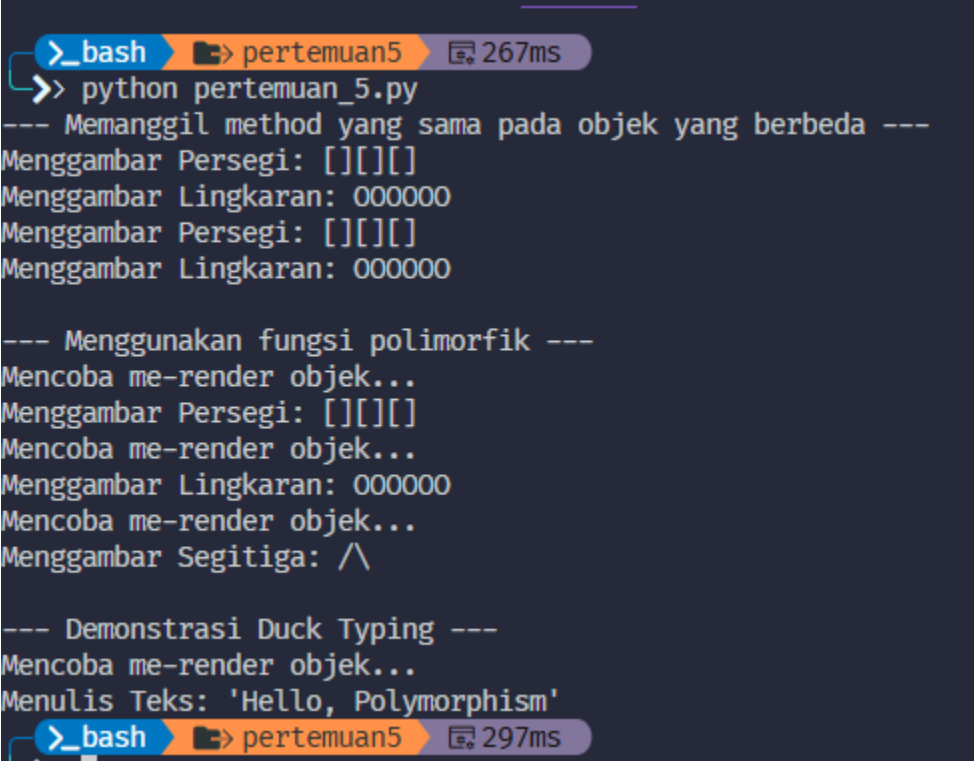
a. Screenshoot kode pertemuan_5.py

```

1  # Parent Class
2  class Bentuk:
3      def gambar(self):
4          # Method ini sengaja dibuat umum dan akan di-override
5          raise NotImplementedError("Subclass harus mengimplementasikan method ini!")
6
7  # Child Class 1
8  class Persegi(Bentuk):
9      def gambar(self):
10         print("Menggambar Persegi: [][][]")
11
12 # Child Class 2
13 class Lingkaran(Bentuk):
14     def gambar(self):
15         print("Menggambar Lingkaran: 000000")
16
17 # Child Class 3
18 class Segitiga(Bentuk):
19     def gambar(self):
20         print("Menggambar Segitiga: /\\"")
21
22 # Class Yang Tidak Berhubungan
23 class Teks:
24     def gambar(self):
25         print("Menulis Teks: 'Hello, Polymorphism'")
26
27 # Sebuah fungsi yang menunjukkan perilaku polimorfik
28 def render_objek(objek_untuk_gambar):
29     print("Mencoba me-render objek...")
30     objek_untuk_gambar.gambar()
31
32 daftar_bentuk = [Persegi(), Lingkaran(), Persegi(), Lingkaran()]
33
34 print("--- Memanggil method yang sama pada objek yang berbeda ---")
35 for bentuk in daftar_bentuk:
36     bentuk.gambar()
37
38 persegi = Persegi()
39 lingkaran = Lingkaran()
40 segitiga = Segitiga()
41 teks_biasa = Teks() # Objek ini bukan turunan dari bentuk
42
43 print("\n--- Menggunakan fungsi polimorfik ---")
44 render_objek(persegi)
45 render_objek(lingkaran)
46 render_objek(segitiga)
47
48 print("\n--- Demonstrasi Duck Typing ---")
49 render_objek(teks_biasa) # Fungsi tetap bekerja
50

```

b. Screenshoot hasil eksekusi pertemuan_5.py



```
>_bash > pertemuan5 267ms
>> python pertemuan_5.py
--- Memanggil method yang sama pada objek yang berbeda ---
Menggambar Persegi: [][][]
Menggambar Lingkaran: 000000
Menggambar Persegi: [][][]
Menggambar Lingkaran: 000000

--- Menggunakan fungsi polimorfik ---
Mencoba me-render objek...
Menggambar Persegi: [][][]
Mencoba me-render objek...
Menggambar Lingkaran: 000000
Mencoba me-render objek...
Menggambar Segitiga: /\

--- Demonstrasi Duck Typing ---
Mencoba me-render objek...
Menulis Teks: 'Hello, Polymorphism'
>_bash > pertemuan5 297ms
```

c. Screenshoot kode latihan_5.py.

```

1  class Notifikasi:
2      def kirim(self, pesan):
3          raise NotImplementedError(pesan)
4
5  class Email(Notifikasi):
6      def kirim(self, pesan):
7          print(f"[EMAIL] Mengirim: '{pesan}'")
8
9  class SMS(Notifikasi):
10     def kirim(self, pesan):
11         print(f"[SMS] Mengirim: '{pesan}'")
12
13     class PushNotif(Notifikasi):
14         def kirim(self, pesan):
15             print(f"[PUSH] Mengirim: '{pesan}'")
16
17     pesan = "Diskon Spesial! Hanya untuk Anda!"
18     daftar = [Email(), SMS(), PushNotif()]
19
20     for notif in daftar:
21         notif.kirim(pesan)

```

d. Screenshoot hasil eksekusi program latihan_5.py.

```

>_bash > pertemuan5 269ms
>> python latihan_5.py
[EMAIL] Mengirim: 'Diskon Spesial! Hanya untuk Anda!'
[SMS] Mengirim: 'Diskon Spesial! Hanya untuk Anda!'
[PUSH] Mengirim: 'Diskon Spesial! Hanya untuk Anda!'
>_bash > pertemuan5 284ms

```

e. Refleksi singkat: Dengan polimorfisme, kode jadi lebih ringkas, kita cukup panggil method yang sama, dan tiap objek mengerjakan detailnya sendiri. Kita tidak perlu deretan if-elif-else, saat menambah jenis notifikasi baru, cukup buat class baru tanpa mengubah kode yang sudah ada. Hasilnya, kode lebih rapi, mudah dirawat, dan risiko bug lebih kecil.