

LAPORAN PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK (PBO)

PRAKTIKUM 7



Nim : 2411102441242

Nama : Muhammad Rifqi Ihsan

FAKULTAS SAINS DAN TEKNOLOGI

PROGRAM STUDI S1 TEKNIK INFORMATIKA

UNIVERSITAS MUHAMMADIYAH KALIMANTAN TIMUR

EMAIL : 2411102441242@umkt.ac.id

Langkah 1: Eksplorasi Pustaka math

```
pertemuan_7.py > ...
1  import math
2
3  class KalkulatorLingkaran:
4      def __init__(self, radius):
5          self.__radius = 0
6          self.set_radius(radius)
7          print(f"Objek lingkaran dengan radius {self.__radius} dibuat.")
8
9      def set_radius(self, radius):
10         if radius > 0:
11             self.__radius = radius
12         else:
13             print("Error: Radius harus lebih besar dari 0.")
14             self.__radius = 1
15
16     def hitung_luas(self):
17         luas = math.pi * (self.__radius ** 2)
18         return luas
19
20     def hitung_keliling(self):
21         keliling = 2 * math.pi * self.__radius
22         return keliling
23
24     lingkaran_1 = KalkulatorLingkaran(7)
25     luas_lingkaran = lingkaran_1.hitung_luas()
26     keliling_lingkaran = lingkaran_1.hitung_keliling()
27
28     print(f"\nRadius: 7")
29     print(f"Luas Lingkaran: {luas_lingkaran:.2f}")
30     print(f"Keliling Lingkaran: {keliling_lingkaran:.2f}")
31
```

Output:

```
[Running] python -u "c:\Users\Asus\Downloads\P1 (PBO)\pertemuan_7.py"
Objek lingkaran dengan radius 7 dibuat.

Radius: 7
Luas Lingkaran: 153.94
Keliling Lingkaran: 43.98

[Done] exited with code=0 in 0.201 seconds
```

Langkah 2: Integrasi Pustaka datetime

```

pertemuan_7.py > ...
1  from datetime import datetime
2
3  class LogPesan:
4      def __init__(self, pengirim, isi_pesan):
5          self.__pengirim = pengirim
6          self.__isi_pesan = isi_pesan
7          self.__timestamp = datetime.now()
8
9      def tampilkan_log(self):
10         waktu_terformat = self.__timestamp.strftime("%d %B %Y, Pukul %H:%M:%S")
11         print("--- Log Pesan Masuk ---")
12         print(f"Pengirim : {self.__pengirim}")
13         print(f"Waktu      : {waktu_terformat}")
14         print(f"Pesan       : {self.__isi_pesan}")
15
16     pesan_1 = LogPesan("Admin", "Server akan segera di-restart untuk maintenance.")
17     pesan_1.tampilkan_log()
18
19     pesan_2 = LogPesan("User01", "Pekerjaan saya sudah disimpan, silakan restart.")
20     pesan_2.tampilkan_log()
21

```

Output:

```

[Running] python -u "c:\Users\Asus\Downloads\P1 (PBO)\pertemuan_7.py"
--- Log Pesan Masuk ---
Pengirim : Admin
Waktu      : 14 October 2025, Pukul 18:08:05
Pesan      : Server akan segera di-restart untuk maintenance.
--- Log Pesan Masuk ---
Pengirim : User01
Waktu      : 14 October 2025, Pukul 18:08:05
Pesan      : Pekerjaan saya sudah disimpan, silakan restart.

```

Studi Kasus: Analisis File Sederhana

```

pertemuan_7.py > ...
1  import os
2  from datetime import datetime
3
4  class FileAnalyzer:
5      def __init__(self, file_path):
6          self.__file_path = file_path
7
8      def __file_exists(file_path):
9          self.__file_ada = True
10         self.__file_size = os.path.getsize(file_path)
11     else:
12         self.__file_ada = False
13         print(f"Error: File '{file_path}' tidak ditemukan.")
14
15     def get_file_size(self, unit="bytes"):
16         if not self.__file_ada:
17             return None
18         if unit.lower() == "kb":
19             return self.__file_size / 1024
20         return self.__file_size
21
22     def get_modification_time(self):
23         if not self.__file_ada:
24             return None
25         mod_time = os.path.getmtime(self.__file_path)
26         waktu_terformat = datetime.fromtimestamp(mod_time)
27         return waktu_terformat.strftime("%d %B %Y, %H:%M:%S")
28
29     def analyze(self):
30         print("\n=== Hasil Analisis File ===")
31         print(f>Nama File : {self.__file_path}")
32
33         if not self.__file_ada:
34             print("Status      : File tidak ditemukan. Analisis dibatalkan.")
35             return
36
37         ukuran_kb = self.get_file_size("KB")
38         waktu_modifikasi = self.get_modification_time()
39
40         print("Status      : File ditemukan")
41         print(f"Ukuran      : {ukuran_kb:.2f} KB")
42         print(f"Terakhir diubah : {waktu_modifikasi}")
43         print("=====\n")
44
45     analyzer1 = FileAnalyzer("dokumen.txt")
46     analyzer1.analyze()
47
48     analyzer2 = FileAnalyzer("file_khayalan.txt")
49     analyzer2.analyze()
50

```

```

≡ dokumen.txt
1  # --- Bagian Utama Program ---
2  # Pastikan file dokumen.txt ada di folder yang sama
3  analyzer1 = FileAnalyzer("dokumen.txt")
4  analyzer1.analyze()
5
6  # (Opsional) Uji file yang tidak ada
7  analyzer2 = FileAnalyzer("file_khayalan.txt")
8  analyzer2.analyze()

```

Output:

```
=== Hasil Analisis File ===  
Nama File : dokumen.txt  
Status    : File ditemukan  
Ukuran    : 0.06 KB  
Terakhir diubah : 14 October 2025, 18:28:48  
=====
```

Error: File 'file_khayalan.txt' tidak ditemukan.

```
=== Hasil Analisis File ===  
Nama File : file_khayalan.txt  
Status    : File tidak ditemukan. Analisis dibatalkan.
```

Refleksi:

Dengan memasukkan pustaka standar ke dalam class, kode menjadi lebih modular dan fleksibel. Pendekatan ini juga mempermudah penggunaan kembali (reuse) serta mengurangi kemungkinan konflik antar fungsi yang berdiri sendiri.