LAPORAN PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK (PBO)

PRAKTIKUM 6

2411102441053

Radhitya Andromeda Barito

FAKULTAS SAINS DAN TEKNOLOGI

PROGRAM STUDI S1 TEKNIK INFORMATIKA

UNIVERSITAS MUHAMMADIYAH KALIMANTAN TIMUR

EMAIL : 2411102441053@umkt.ac.id

1. Proyek Kalkulator

1.1 main.py

pertemuan_6/proyek_kalkulator/main.py

```python
from bangun_datar.lingkaran import Lingkaran
from bangun_datar.persegi import Persegi

def jalankan_program():
    print("--- Menghitung Luas Bangun Datar (Versi Modular) ---")

    lingkaran_A = Lingkaran(7)
    luas_lingkaran = lingkaran_A.hitung_luas()
    print(f"Luas Lingkaran dengan radius 7 adalah {luas_lingkaran:.2f}")

    persegi_B = Persegi(5)
    luas_persegi = persegi_B.hitung_luas()
    print(f"Luas Persegi dengan sisi 5 adalah {luas_persegi}")

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

1.2 lingkaran.py

pertemuan_6/proyek_kalkulator/bangun_datar/lingkaran.py

```python
import math

class Lingkaran:
    def __init__(self,radius):
        self.radius = radius

    def hitung_luas(self):
        return math.pi * (self.radius ** 2)

```

1.3 persegi.py

pertemuan_6/proyek_kalkulator/bangun_datar/persegi.py

```python
class Persegi:
    def __init__(self, sisi):
        self.sisi = sisi

    def hitung_luas(self):
        return self.sisi ** 2

```

EMAIL : 2411102441053@umkt.ac.id

Output :

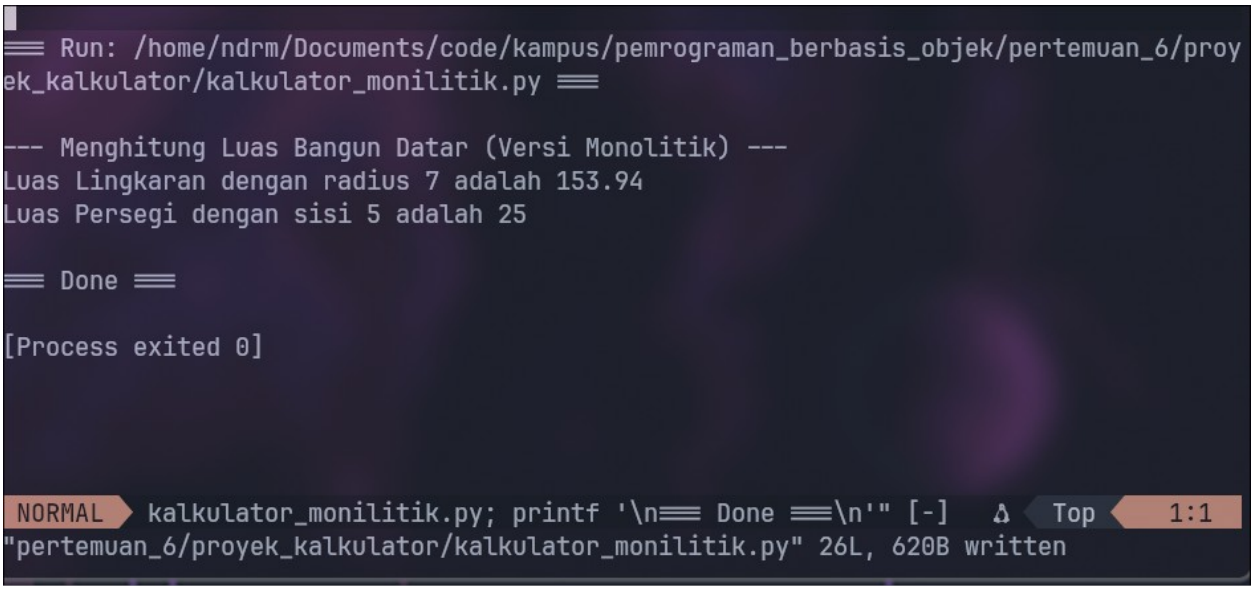
```
=== Run: /home/ndrm/Documents/code/kampus/pemrograman_berbasis_objek/pertemuan_6/proy
ek_kalkulator/main.py ===

--- Menghitung Luas Bangun Datar (Versi Modular) ---
Luas Lingkaran dengan radius 7 adalah 153.94
Luas Persegi dengan sisi 5 adalah 25

=== Done ===

[Process exited 0]

NORMAL  main.py; printf '\n=== Done ===\n'" [-]                    Top  1:1
"pertemuan_6/proyek_kalkulator/main.py" 16L, 514B written
```

1.4 kalkulator_monilitik.py

pertemuan_6/proyek_kalkulator/kalkulator_monilitik.py

```python
import math

class Lingkaran:
    def __init__(self, radius):
        self.radius = radius

    def hitung_luas(self):
        return math.pi * (self.radius ** 2)

class Persegi:
    def __init__(self, sisi):
        self.sisi = sisi

    def hitung_luas(self):
        return self.sisi ** 2


print("--- Menghitung Luas Bangun Datar (Versi Monolitik) ---")

lingkaran_A = Lingkaran(7)
luas_lingkaran = lingkaran_A.hitung_luas()
print(f"Luas Lingkaran dengan radius 7 adalah {luas_lingkaran:.2f}")

persegi_B = Persegi(5)
luas_persegi = persegi_B.hitung_luas()
print(f"Luas Persegi dengan sisi 5 adalah {luas_persegi}")
```

Output :



Penjelasan :

Kode di atas merupakan contoh penerapan modularisasi dalam Python, di mana program dibagi menjadi beberapa file agar lebih terstruktur dan mudah dikelola. File `lingkaran.py` berisi class Lingkaran dengan method untuk menghitung luas menggunakan rumus $\pi \times r^2$, sedangkan `persegi.py` berisi class Persegi dengan method untuk menghitung luas sisi². Kedua modul tersebut diimpor ke dalam `main.py`, yang berfungsi sebagai program utama untuk membuat objek, menghitung luas, dan menampilkan hasil. Struktur modular ini memudahkan pengembangan program karena setiap bagian memiliki fungsi yang terpisah dan dapat digunakan kembali tanpa mengubah kode utama.

Sedangkan file kalkulator_monolitik.py merupakan versi awal dari program yang ditulis secara monolitik, di mana seluruh logika berada dalam satu file tanpa pemisahan modul. Program tersebut juga menghitung luas lingkaran dan persegi dengan cara yang sama, namun semua kelas dan eksekusi program ditempatkan di satu tempat. Meskipun cara ini lebih sederhana untuk program kecil, pendekatan monolitik menjadi kurang efisien ketika program berkembang, karena sulit dikelola dan dimodifikasi. Oleh karena itu, versi modular lebih disarankan untuk membuat kode yang terstruktur, mudah diperbarui, dan dapat digunakan kembali.

2. Buku Kontak sederhana

main.py

pertemuan_6/buku_kontak/main.py

```python
from models.kontak import Kontak

if __name__ == "__main__":
    daftar_kontak = []

    kontak1 = Kontak("Syahrul", "081234567890")
    kontak2 = Kontak("Babloi", "089876543210")
    kontak3 = Kontak("Jarpis", "087712345678")

    daftar_kontak.extend([kontak1, kontak2, kontak3])

    print("--- Daftar Kontak ---")
    for kontak in daftar_kontak:
        kontak.tampilkan_info()
```

kontak.py

pertemuan_6/buku_kontak/models/kontak.py

```python
class Kontak:
    def __init__(self, nama, nomor_telepon):
        self.__nama = nama
        self.__nomor_telepon = nomor_telepon

    def tampilkan_info(self):
        print(f"Nama: {self.__nama}, Nomor Telepon: {self.__nomor_telepon}")

    # Getter (opsional)
    def get_nama(self):
        return self.__nama

    def get_nomor_telepon(self):
        return self.__nomor_telepon

    # Setter (opsional)
    def set_nama(self, nama):
        self.__nama = nama

    def set_nomor_telepon(self, nomor_telepon):
        self.__nomor_telepon = nomor_telepon
```

EMAIL : 2411102441053@umkt.ac.id

Penjelasan :

File `kontak.py` berisi definisi class Kontak yang merepresentasikan data kontak seseorang dengan dua atribut privat, yaitu `__nama` dan `__nomor_telepon`. Penggunaan atribut privat ini bertujuan untuk melindungi data agar tidak diakses atau diubah secara langsung dari luar kelas (konsep enkapsulasi). Class ini juga memiliki method `tampilkan_info()` untuk menampilkan informasi kontak, serta getter dan setter sebagai cara aman untuk mengakses dan mengubah nilai atribut tersebut.

File `main.py` berfungsi sebagai program utama yang mengimpor class Kontak dari modul `models.kontak`. Di dalam blok `if __name__ == "__main__":`, program membuat tiga objek kontak dengan nama dan nomor telepon berbeda, lalu menyimpannya ke dalam list `daftar_kontak`. Melalui perulangan `for`, setiap objek dalam daftar ditampilkan menggunakan method `tampilkan_info()`. Pendekatan ini membuat program lebih terstruktur, mudah dikembangkan, serta memisahkan logika data (model) dengan eksekusi program utama.