

**LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK (PBO) – [TUGAS BRP]**



Disusun Oleh

Ahmad Ali Mukti 123140155

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SUMATERA**

2025

Soal nomor 1

Penjelasan :

- **Import Library**

- Program mengimpor modul `math` untuk operasi matematika, seperti logaritma.

- **Membuat Kelas `Kalkulator`**

- Kelas `Kalkulator` dibuat untuk menangani berbagai operasi aritmatika dengan metode khusus (dunder methods).

- **Inisialisasi Objek (`__init__`)**

- Objek `Kalkulator` dibuat dengan atribut `nilai` yang menyimpan angka.

- **Implementasi Operator Aritmatika**

- `__add__`: Menangani penjumlahan (+).
- `__sub__`: Menangani pengurangan (-).
- `__mul__`: Menangani perkalian (*).
- `__truediv__`: Menangani pembagian (/), termasuk pengecekan pembagian dengan nol.
- `__pow__`: Menangani perpangkatan (**).
- `log()`: Menghitung logaritma dengan pengecekan nilai negatif atau nol.

- **Metode `__str__`**

- Mengonversi objek `Kalkulator` menjadi string agar bisa dicetak dengan `print()`.

- **Menampilkan Menu Pilihan**

- Program mencetak daftar operasi matematika yang bisa dipilih oleh pengguna.

- **Menerima Input dari Pengguna**

- Pengguna diminta memasukkan pilihan operasi, serta dua angka yang akan dihitung.

- **Melakukan Perhitungan Berdasarkan Pilihan Pengguna**

- Program menggunakan struktur `if-elif` untuk menentukan operasi berdasarkan input pengguna.

- Setiap operasi memanggil metode dunder yang sesuai (`__add__`, `__sub__`, dll.).
- Hasil perhitungan ditampilkan menggunakan `print()`.

- **Menangani Kesalahan Input**

- Jika pengguna memasukkan pilihan yang tidak valid, program menampilkan pesan error

Source Code

```
import math

class Kalkulator:
    def __init__(self, nilai):
        self.nilai = nilai

    def __add__(self, other):
        return Kalkulator(self.nilai + other.nilai)

    def __sub__(self, other):
        return Kalkulator(self.nilai - other.nilai)

    def __mul__(self, other):
        return Kalkulator(self.nilai * other.nilai)

    def __truediv__(self, other):
        if other.nilai == 0:
            raise ValueError("Tidak Bisa Membagi dengan 0")
        return Kalkulator(self.nilai / other.nilai)

    def __pow__(self, other):
        return Kalkulator(self.nilai ** other.nilai)

    def log(self, other):
        if self.nilai <= 0 or other.nilai <= 0:
            raise ValueError("Tidak Bisa Menghitung Logaritma dari 0 atau bilangan negatif")
        return Kalkulator(math.log(self.nilai, other.nilai))

    def __str__(self):
        return str(self.nilai)

print("Kalkulator sederhana : ")
print("1. + ")
print("2. - ")
print("3. x ")
print("4. : ")
print("5. ** ")
print("6. log ")

pilih = input("Masukan pilihan anda : ")

nilai = int(input("Masukan nilai 1 : "))
nilai = Kalkulator(nilai)

other = int(input("Masukan nilai 2 : "))
other = Kalkulator(other)

if pilih == "1":
    hasil_penjumlahan = nilai + other
    print(f"Hasil Penjumlahan: {hasil_penjumlahan}")

elif pilih == "2":
    hasil_pengurangan = nilai - other
    print(f"Hasil Pengurangan: {hasil_pengurangan}")

elif pilih == "3":
    hasil_perkalian = nilai * other
    print(f"Hasil Perkalian: {hasil_perkalian}")

elif pilih == "4":
    hasil_pembagian = nilai / other
    print(f"Hasil Pembagian: {hasil_pembagian}")

elif pilih == "5":
    hasil_eksponen = nilai ** other
    print(f"Hasil Eksponen: {hasil_eksponen}")

elif pilih == "6":
    hasil_logaritma = nilai.log(other)
    print(f"Hasil Logaritma: {hasil_logaritma}")

else:
    print("Pilihan tidak ada cuy")
```

Output Hasil (Screenshot)

```
PS C:\Users\ahmad\OneDrive\Dokumen\Tugas Prak PBO> & C:/Users/ahmad/OneDrive\Dokumen\Tugas Prak PBO>
● Kalkulator sederhana :
1. +
2. -
3. x
4. :
5. **
6. log
Masukan pilihan anda : 3
Masukan nilai 1 5
Masukan nilai 2: 5
Hasil Perkalian: 25
○ PS C:\Users\ahmad\OneDrive\Dokumen\Tugas Prak PBO>
```

Soal No 2

Penjelasan :

☐ **Import Library**

- Program mengimpor modul random untuk memilih golongan darah anak secara acak jika ada lebih dari satu kemungkinan.

☐ **Membuat Kelas Father dan Mother**

- Kelas Father dan Mother dibuat untuk merepresentasikan golongan darah ayah dan ibu.
- Masing-masing memiliki atribut blood_type yang menyimpan golongan darah yang diberikan saat inisialisasi.

☐ **Membuat Kelas Child**

- Kelas Child dibuat untuk menentukan golongan darah anak berdasarkan golongan darah ayah dan ibu.
- Pada inisialisasi, kelas ini menerima objek Father dan Mother, lalu menyimpan golongan darah mereka ke atribut father_blood_type dan mother_blood_type.
- Golongan darah anak ditentukan melalui metode determine_blood_type().

☐ **Metode determine_blood_type()**

- Metode ini menentukan golongan darah anak berdasarkan kombinasi golongan darah ayah dan ibu.

- Jika ada lebih dari satu kemungkinan, golongan darah dipilih secara acak menggunakan `random.choice()`.
- Jika kombinasi golongan darah tidak dikenal, metode mengembalikan "Darah alien".

☐ **Menerima Input dari Pengguna**

- Program meminta pengguna memasukkan golongan darah ayah dan ibu melalui `input()`.

☐ **Membuat Objek Father dan Mother**

- Objek Father dan Mother dibuat berdasarkan input pengguna.

☐ **Menentukan Golongan Darah Anak**

- Objek Child dibuat dengan parameter father dan mother.
- Golongan darah anak dihitung secara otomatis menggunakan metode `determine_blood_type()`.

☐ **Menampilkan Hasil**

- Program mencetak golongan darah anak berdasarkan hasil perhitungan yang telah dilakukan.

Source Code :

```
import random

class Father:
    def __init__(self, blood_type):
        self.blood_type = blood_type

class Mother:
    def __init__(self, blood_type):
        self.blood_type = blood_type

class Child:
    def __init__(self, father, mother):
        self.father_blood_type = father.blood_type
        self.mother_blood_type = mother.blood_type
        self.blood_type = self.determine_blood_type()

    def determine_blood_type(self):
        if self.father_blood_type == "A" and self.mother_blood_type == "A":
            return "A"
        elif self.father_blood_type == "A" and self.mother_blood_type == "O":
            random_blood_type = ["A", "O"]
            return random.choice(random_blood_type)
        elif self.father_blood_type == "O" and self.mother_blood_type == "A":
            random_blood_type = ["A", "O"]
            return random.choice(random_blood_type)
        elif self.father_blood_type == "A" and self.mother_blood_type == "B":
            random_blood_type = ["A", "B", "AB"]
            return random.choice(random_blood_type)
        elif self.father_blood_type == "B" and self.mother_blood_type == "A":
            random_blood_type = ["A", "B", "AB"]
            return random.choice(random_blood_type)
        elif self.father_blood_type == "B" and self.mother_blood_type == "B":
            return "B"
        elif self.father_blood_type == "B" and self.mother_blood_type == "O":
            random_blood_type = ["B", "O"]
            return random.choice(random_blood_type)
        elif self.father_blood_type == "O" and self.mother_blood_type == "B":
            random_blood_type = ["B", "O"]
            return random.choice(random_blood_type)
        elif self.father_blood_type == "O" and self.mother_blood_type == "O":
            return "O"
        else:
            return "Darah alien"

father = input("Masukkan golongan darah ayah : ")
mother = input("Masukkan golongan darah ibu : ")
father = Father(father)
mother = Mother(mother)

determining_blood_type = Child(father, mother)
print("anak memiliki golongan darah : ", determining_blood_type.blood_type)
```

Output kode :

```
● PS C:\Users\ahmad\OneDrive\Dokumen\Tugas Prak PBO> & C:/Users/ahmad/OneDrive/Dokumen/Tugas Prak PBO/Program PBO.exe
Masukkan golongan darah ayah : A
Masukkan golongan darah ibu : O
anak memiliki golongan darah : A
● PS C:\Users\ahmad\OneDrive\Dokumen\Tugas Prak PBO> & C:/Users/ahmad/OneDrive/Dokumen/Tugas Prak PBO/Program PBO.exe
Masukkan golongan darah ayah : A
Masukkan golongan darah ibu : B
anak memiliki golongan darah : AB
● PS C:\Users\ahmad\OneDrive\Dokumen\Tugas Prak PBO> & C:/Users/ahmad/OneDrive/Dokumen/Tugas Prak PBO/Program PBO.exe
Masukkan golongan darah ayah : A
Masukkan golongan darah ibu : B
anak memiliki golongan darah : B
● PS C:\Users\ahmad\OneDrive\Dokumen\Tugas Prak PBO> & C:/Users/ahmad/OneDrive/Dokumen/Tugas Prak PBO/Program PBO.exe
Masukkan golongan darah ayah : B
Masukkan golongan darah ibu : A
anak memiliki golongan darah : B
○ PS C:\Users\ahmad\OneDrive\Dokumen\Tugas Prak PBO>
```

Lampiran :

<https://chatgpt.com/c/67d95f3e-2e34-8004-87e1-5f4c59b93790>

+ Github copilot