

Hasil Praktikum PBO2 Pertemuan 2

Nama: Alima Aprilia

Nim: 230511168

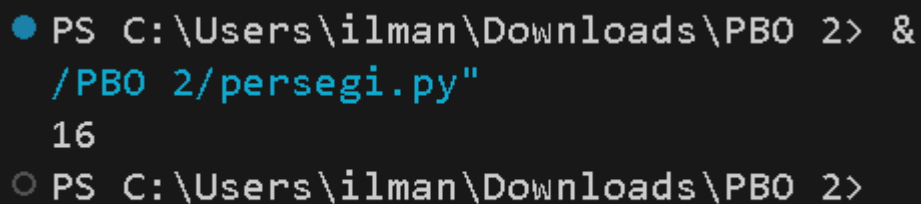
TI23 E

A. Buatlah 10 contoh class dengan tema perhitungan matematika

1. Class Perhitungan Luas Persegi

```
class LuasPersegi:  
    def hitung_luas(self, sisi):  
        return sisi ** 2
```

```
persegi = LuasPersegi()  
print(persegi.hitung_luas(4)) # Output: 16
```



```
PS C:\Users\ilman\Downloads\PBO 2> &  
/PBO 2/persegi.py"  
16  
PS C:\Users\ilman\Downloads\PBO 2>
```

2. Class Perhitungan Luas Segitiga

```
class LuasSegitiga:  
    def hitung_luas(self, alas, tinggi):  
        return 0.5 * alas * tinggi
```

#Membuat

```
segitiga = LuasSegitiga()
```

#Memasukkan

```
alas = float(input("Masukkan panjang alas segitiga: "))
```

```
tinggi = float(input("Masukkan tinggi segitiga: "))
```

#Menghitung

```
luas = segitiga.hitung_luas(alas, tinggi)
```

```
print(f"Luas segitiga dengan alas {alas} dan tinggi {tinggi} adalah {luas}")
```

```
● PS C:\Users\ilman\Downloads\PBO 2> & C:/Users/ilman/AppData/Local/Programs/Python/Python38-32/Python.exe C:/Users/ilman/AppData/Local/Programs/Python/Python38-32/Python.exe /PB0 2/segitiga.py"
Masukkan panjang alas segitiga: 45
Masukkan tinggi segitiga: 67
Luas segitiga dengan alas 45.0 dan tinggi 67.0 adalah 1507.5
○ PS C:\Users\ilman\Downloads\PBO 2> 
```

3. Class Perhitungan Keliling Lingkaran

```
import math
```

```
class KelilingLingkaran:
```

```
def hitung_keliling(self, jari_jari):
```

```
return 2 * math.pi * jari_jari
```

#Membuat

```
lingkaran = KelilingLingkaran()
```

#Memasukkan

```
jari_jari = float(input("Masukkan panjang jari-jari lingkaran: "))
```

#Menghitung

```
keliling = lingkaran.hitung_keliling(jari_jari)
```

```
print(f"Keliling lingkaran dengan jari-jari {jari_jari} adalah {keliling:.2f}")
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
● PS C:\Users\ilman\Downloads\PBO 2> & C:/Users/ilman/AppD
/PBO 2/kelilinglingkaran.py"
Masukkan panjang jari-jari lingkaran: 56
Keliling lingkaran dengan jari-jari 56.0 adalah 351.86
○ PS C:\Users\ilman\Downloads\PBO 2> █
```

4. Class Perhitungan Volume Kubus

```
class VolumeKubus:
```

```
def hitung_volume(self, sisi):
```

```
    return sisi ** 3
```

```
#Membuat objek dari class VolumeKubus
```

```
kubus = VolumeKubus()
```

```
#Memasukkan nilai sisi kubus
```

```
sisi = float(input("Masukkan panjang sisi kubus: "))
```

```
#Menghitung dan menampilkan hasil volume kubus
```

```
volume = kubus.hitung_volume(sisi)
```

```
print(f"Volume kubus dengan sisi {sisi} adalah {volume}")
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
● PS C:\Users\ilman\Downloads\PBO 2> & C:/Users/i
/PBO 2/volume kubus.py"
Masukkan panjang sisi kubus: 43
Volume kubus dengan sisi 43.0 adalah 79507.0
○ PS C:\Users\ilman\Downloads\PBO 2> █
```

5. Class Perhitungan Volume Tabung

```
import math
```

```
class VolumeTabung:
```

```
def hitung_volume(self, jari_jari, tinggi):
```

```
return math.pi (jari_jari ** 2) tinggi
```

```
#Membuat objek dari class VolumeTabung
```

```
tabung = VolumeTabung()
```

```
#Memasukkan nilai jari-jari dan tinggi tabung
```

```
jari_jari = float(input("Masukkan jari-jari tabung: "))
```

```
tinggi = float(input("Masukkan tinggi tabung: "))
```

```
#Menghitung dan menampilkan hasil volume tabung
```

```
volume = tabung.hitung_volume(jari_jari, tinggi)
```

```
print(f"Volume tabung dengan jari-jari {jari_jari} dan tinggi {tinggi} adalah {volume:.2f}")
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
● PS C:\Users\ilman\Downloads\PBO 2> & C:/Users/ilman/AppData/Local/Programs/Python/PBO 2/volume tabung.py
Masukkan jari-jari tabung: 67
Masukkan tinggi tabung: 78
Volume tabung dengan jari-jari 67.0 dan tinggi 78.0 adalah 1100003.53
○ PS C:\Users\ilman\Downloads\PBO 2> 
```

6. Class Perhitungan Akar Kruadat

```
import math
```

```
class AkarKuadrat:
```

```
def hitung_akar(self, angka):
```

```
if angka < 0:
```

```
return "Tidak dapat menghitung akar kuadrat dari bilangan negatif"
```

```
return math.sqrt(angka)
```

```
#Membuat objek dari class AkarKuadrat
```

```
akar = AkarKuadrat()
```

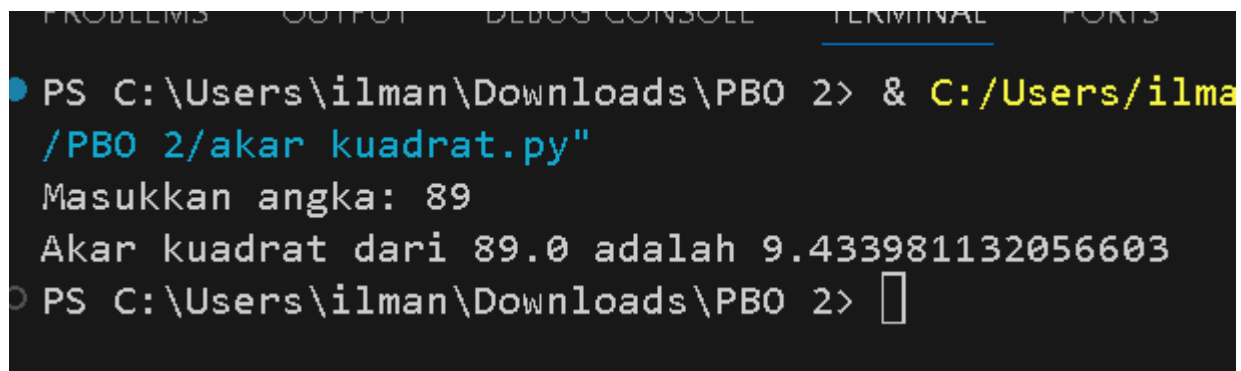
```
#Memasukkan nilai angka yang akan dihitung akar kuadratnya
```

```
angka = float(input("Masukkan angka: "))
```

```
#Menghitung dan menampilkan hasil akar kuadrat
```

```
hasil = akar.hitung_akar(angka)
```

```
print(f'Akar kuadrat dari {angka} adalah {hasil}')
```



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
● PS C:\Users\ilman\Downloads\PBO 2> & C:/Users/ilman/PBO 2/akar kuadrat.py
Masukkan angka: 89
Akar kuadrat dari 89.0 adalah 9.433981132056603
○ PS C:\Users\ilman\Downloads\PBO 2> 
```

7. Class Perhitungan Pangkat

```
class PerhitunganPangkat:
def hitung_pangkat(self, angka, pangkat):
return angka ** pangkat
```

#Membuat objek dari class PerhitunganPangkat

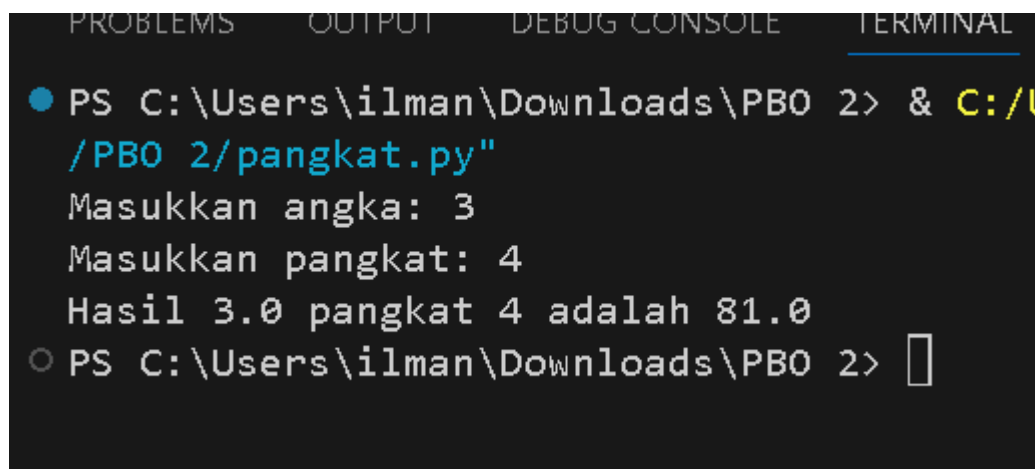
```
pangkat = PerhitunganPangkat()
```

#Memasukkan nilai angka dan pangkat

```
angka = float(input("Masukkan angka: "))
pangkatnya = int(input("Masukkan pangkat: "))
```

#Menghitung dan menampilkan hasil pangkat

```
hasil = pangkat.hitung_pangkat(angka, pangkatnya)
print(f'Hasil {angka} pangkat {pangkatnya} adalah {hasil}')
```



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
● PS C:\Users\ilman\Downloads\PBO 2> & C:/Users/ilman/PBO 2/pangkat.py
Masukkan angka: 3
Masukkan pangkat: 4
Hasil 3.0 pangkat 4 adalah 81.0
○ PS C:\Users\ilman\Downloads\PBO 2> 
```

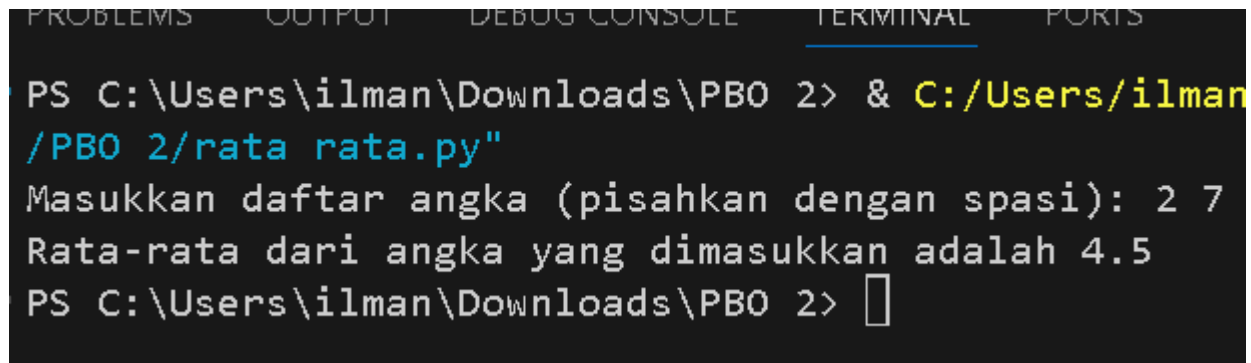
8. Class Perhitungan Rata - Rata

```
class PerhitunganRataRata:
    def hitung_rata_rata(self, daftar_angka):
        if len(daftar_angka) == 0:
            return "Daftar angka tidak boleh kosong"
        return sum(daftar_angka) / len(daftar_angka)
```

#Membuat objek dari class PerhitunganRataRata
rata_rata = PerhitunganRataRata()

#Memasukkan daftar angka yang akan dihitung rata-ratanya
daftar_angka = list(map(float, input("Masukkan daftar angka (pisahkan dengan spasi):
").split()))

#Menghitung dan menampilkan hasil rata-rata
hasil = rata_rata.hitung_rata_rata(daftar_angka)
print(f"Rata-rata dari angka yang dimasukkan adalah {hasil}")



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\ilman\Downloads\PB0 2> & C:/Users/ilman/PB0 2/rata rata.py
Masukkan daftar angka (pisahkan dengan spasi): 2 7
Rata-rata dari angka yang dimasukkan adalah 4.5
PS C:\Users\ilman\Downloads\PB0 2> 
```

9. Class Perhitungan Diskon

```
class PerhitunganDiskon:
    def hitung_diskon(self, harga_awal, persen_diskon):
        jumlah_diskon = (persen_diskon / 100) * harga_awal
        harga_setelah_diskon = harga_awal - jumlah_diskon
        return harga_setelah_diskon
```

#Membuat objek dari class PerhitunganDiskon
diskon = PerhitunganDiskon()

```
#Memasukkan harga awal dan persentase diskon
harga_awal = float(input("Masukkan harga awal: "))
persen_diskon = float(input("Masukkan persentase diskon: "))
```

```
#Menghitung dan menampilkan harga setelah diskon
hasil = diskon.hitung_diskon(harga_awal, persen_diskon)
print(f"Harga setelah diskon adalah {hasil}")
```

```
● PS C:\Users\ilman\Downloads\PBO 2> & C:/
  /PBO 2/diskon.py"
  Masukkan harga awal: 56000
  Masukkan persentase diskon: 10
  Harga setelah diskon adalah 50400.0
○ PS C:\Users\ilman\Downloads\PBO 2> █
```

10. Class Perhitungan Fibonacci

```
class PerhitunganFibonacci:
    def hitungfibonacci(self, n):
        if n <= 0:
            return "Input harus lebih besar dari 0"
        elif n == 1:
            return 0
        elif n == 2:
            return 1
        else:
            a, b = 0, 1
            for i in range(n - 2):
                a, b = b, a + b
            return b
```

```
#Membuat objek dari class PerhitunganFibonacci
fibonacci = PerhitunganFibonacci()
```

```
#Memasukkan bilangan Fibonacci yang ingin dihitung
n = int(input("Masukkan bilangan Fibonacci ke-n: "))
```

```
#Menghitung dan menampilkan hasil Fibonacci
hasil = fibonacci.hitung_fibonacci(n)
```

```
print(f"Bilangan Fibonacci ke-{n} adalah {hasil}")
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
● PS C:\Users\ilman\Downloads\PBO 2> & C:
/PBO 2/fibonacci.py"
Masukkan bilangan Fibonacci ke-n: 8
Bilangan Fibonacci ke-8 adalah 13
○ PS C:\Users\ilman\Downloads\PBO 2> █
```

B. Buatlah 10 contoh class dengan tema perhitungan fisika

1. Class Perhitungan Kecepatan

```
class PerhitunganKecepatan:
    @staticmethod
    def hitung_kecepatan(jarak, waktu):
        if waktu == 0:
            raise ValueError("Waktu tidak boleh nol!")
        return jarak / waktu

#Program utama
def main():
    try:
        # Input jarak dan waktu
        jarak = float(input("Masukkan jarak (dalam meter): "))
        waktu = float(input("Masukkan waktu (dalam detik): "))
```

```
        # Hitung kecepatan
        kecepatan = PerhitunganKecepatan.hitung_kecepatan(jarak, waktu)
        print(f"Kecepatan: {kecepatan} m/s")

    except ValueError as e:
        print(e)
```



```
if name == "main":
```

```
main()
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\ilman\Downloads\PBO 2> &
  lman/Downloads/PBO 2/kecepatan.py"
Masukkan jarak (dalam meter): 34
Masukkan waktu (dalam detik): 56
Kecepatan: 0.6071428571428571 m/s
PS C:\Users\ilman\Downloads\PBO 2> 
```

2. Class Perhitungan Gaya

```
class PerhitunganGaya:
```

```
@staticmethod
```

```
def hitung_gaya(massa, percepatan):
```

```
    return massa * percepatan
```

```
#Program utama
```

```
def main():
```

```
    try:
```

```
        # Input massa dan percepatan
```

```
        massa = float(input("Masukkan massa (dalam kilogram): "))
```

```
        percepatan = float(input("Masukkan percepatan (dalam meter per detik kuadrat): "))
```

```
        # Hitung gaya
```

```
        gaya = PerhitunganGaya.hitung_gaya(massa, percepatan)
```

```
        print(f"Gaya yang dihasilkan: {gaya} Newton")
```

```
    except ValueError as e:
```

```
        print("Input tidak valid, pastikan angka yang dimasukkan benar.")
```

```
if name == "main":
```

```
main()
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
● PS C:\Users\ilman\Downloads\PBO 2> & C:/Users/ilman/AppD
  lman/Downloads/PBO 2/gaya.py"
  Masukkan massa (dalam kilogram): 78
  Masukkan percepatan (dalam meter per detik kuadrat): 54
  Gaya yang dihasilkan: 4212.0 Newton
○ PS C:\Users\ilman\Downloads\PBO 2> █
```

3. Class Perhitungan Energi Kinetik

```
class PerhitunganEnergiKinetik:
    @staticmethod
    def hitung_energi_kinetik(massa, kecepatan):
        return 0.5 * massa * kecepatan ** 2
```

Program utama

```
def main():
    try:
        # Input massa dan kecepatan
        massa = float(input("Masukkan massa (dalam kilogram): "))
        kecepatan = float(input("Masukkan kecepatan (dalam meter per detik): "))
```

```
        # Hitung energi kinetik
        energi_kinetik = PerhitunganEnergiKinetik.hitung_energi_kinetik(massa,
        kecepatan)
        print(f"Energi Kinetik: {energi_kinetik} Joule")

    except ValueError as e:
        print("Input tidak valid, pastikan angka yang dimasukkan benar.")
```

```
if name == "main":
    main()
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
● PS C:\Users\ilman\Downloads\PBO 2> & C:/Users/ilman/Downloads/PBO 2/energi_kinetik.py"
Masukkan massa (dalam kilogram): 67
Masukkan kecepatan (dalam meter per detik): 5
Energi Kinetik: 837.5 Joule
○ PS C:\Users\ilman\Downloads\PBO 2> █
```

4. Class Perhitungan Energi Potensial

```
class PerhitunganEnergiPotensial:
    @staticmethod
    def hitung_energi_potensial(massa, ketinggian, gravitasi=9.8):
        return massa * gravitasi * ketinggian

#Program utama
def main():
    try:
        # Input massa dan ketinggian
        massa = float(input("Masukkan massa (dalam kilogram): "))
        ketinggian = float(input("Masukkan ketinggian (dalam meter): "))
```

```
        # Hitung energi potensial
        energi_potensial =
        PerhitunganEnergiPotensial.hitung_energi_potensial(massa, ketinggian)
        print(f"Energi Potensial: {energi_potensial} Joule")

    except ValueError as e:
        print("Input tidak valid, pastikan angka yang dimasukkan benar.")
```

```
if name == "main":
    main()
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
● PS C:\Users\ilman\Downloads\PBO 2> & C:/User
  ilman/Downloads/PBO 2/energi potensial.py"
  Masukkan massa (dalam kilogram): 56
  Masukkan ketinggian (dalam meter): 54
  Energi Potensial: 29635.200000000004 Joule
○ PS C:\Users\ilman\Downloads\PBO 2> █
```

5. Class Perhitungan Usaha

```
import math
```

```
class PerhitunganUsaha:
```

```
    @staticmethod
```

```
    def hitung_usaha(gaya, jarak, sudut):
```

```
        # Menghitung usaha dengan rumus  $W = F d \cos(\theta)$ 
```

```
        sudut_rad = math.radians(sudut) # Mengubah sudut dari derajat ke radian
```

```
        usaha = gaya * jarak * math.cos(sudut_rad)
```

```
        return usaha
```

```
#Program utama
```

```
def main():
```

```
    try:
```

```
        # Input gaya, jarak, dan sudut
```

```
        gaya = float(input("Masukkan gaya (dalam Newton): "))
```

```
        jarak = float(input("Masukkan jarak (dalam meter): "))
```

```
        sudut = float(input("Masukkan sudut antara arah gaya dan gerakan (dalam derajat): "))
```

```
        # Hitung usaha
```

```
        usaha = PerhitunganUsaha.hitung_usaha(gaya, jarak, sudut)
```

```
        print(f"Usaha yang dilakukan: {usaha} Joule")
```

```
    except ValueError as e:
```

```
        print("Input tidak valid, pastikan angka yang dimasukkan benar.")
```

```
if name == "main":
```

```
    main()
```

```
● PS C:\Users\ilman\Downloads\PBO 2> & C:/Users/ilman/AppData/Local/
  man/Downloads/PBO 2/usaha.py"
  Masukkan gaya (dalam Newton): 56
  Masukkan jarak (dalam meter): 67
  Masukkan sudut antara arah gaya dan gerakan (dalam derajat): 34
  Usaha yang dilakukan: 3110.5489722265165 Joule
○ PS C:\Users\ilman\Downloads\PBO 2> █
```

6. Class Perhitungan Daya

```
class PerhitunganDaya:
    @staticmethod
    def hitung_daya(usaha, waktu):
        if waktu == 0:
            raise ValueError("Waktu tidak boleh nol!")
        return usaha / waktu

#Program utama
def main():
    try:
        # Input usaha dan waktu
        usaha = float(input("Masukkan usaha (dalam Joule): "))
        waktu = float(input("Masukkan waktu (dalam detik): "))

        # Hitung daya
        daya = PerhitunganDaya.hitung_daya(usaha, waktu)
        print(f"Daya yang dihasilkan: {daya} Watt")

    except ValueError as e:
        print(e)

if name == "main":
    main()
```

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORT
● PS C:\Users\ilman\Downloads\PBO 2> & C:/Users/
  ilman/Downloads/PBO 2/daya.py"
  Masukkan usaha (dalam Joule): 56
  Masukkan waktu (dalam detik): 67
  Daya yang dihasilkan: 0.835820895522388 Watt
○ PS C:\Users\ilman\Downloads\PBO 2> 
```

7. Class Perhitungan Tegangan

```
class PerhitunganTegangan:
```

```
    @staticmethod
```

```
    def hitung_tegangan(arus, resistansi):
```

```
        return arus * resistansi
```

```
    #Program utama
```

```
def main():
```

```
    try:
```

```
        # Input arus dan resistansi
```

```
        arus = float(input("Masukkan arus (dalam Ampere): "))
```

```
        resistansi = float(input("Masukkan resistansi (dalam Ohm): "))
```

```
        # Hitung tegangan
```

```
        tegangan = PerhitunganTegangan.hitung_tegangan(arus, resistansi)
```

```
        print(f"Tegangan: {tegangan} Volt")
```

```
    except ValueError as e:
```

```
        print("Input tidak valid, pastikan angka yang dimasukkan benar.")
```

```
if name == "main":
```

```
    main()
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
● PS C:\Users\ilman\Downloads\PBO 2> & C:\Users\ilman\Downloads\PBO 2/tegangan.py"
Masukkan arus (dalam Ampere): 67
Masukkan resistansi (dalam Ohm): 87
Tegangan: 5829.0 Volt
○ PS C:\Users\ilman\Downloads\PBO 2> █
```

8. Class Perhitungan Resistansi

```
class PerhitunganResistansi:
    @staticmethod
    def hitung_resistansi(tegangan, arus):
        if arus == 0:
            raise ValueError("Arus tidak boleh nol!")
        return tegangan / arus
```

```
#Program Program utama
def main():
    try:
        # Input tegangan dan arus
        tegangan = float(input("Masukkan tegangan (dalam Volt): "))
        arus = float(input("Masukkan arus (dalam Ampere): "))
```

```
        # Hitung resistansi
        resistansi = PerhitunganResistansi.hitung_resistansi(tegangan, arus)
        print(f"Resistansi: {resistansi} Ohm")

    except ValueError as e:
        print(e)
```

```
if name == "main":
    main()
```

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL
● PS C:\Users\ilman\Downloads\PBO 2> &
  lman/Downloads/PBO 2/resistansi.py"
  Masukkan tegangan (dalam Volt): 78
  Masukkan arus (dalam Ampere): 76
  Resistansi: 1.0263157894736843 Ohm
○ PS C:\Users\ilman\Downloads\PBO 2> █
```

9. Class Perhitungan Laju Reaksi

```
class PerhitunganLajuReaksi:
    @staticmethod
    def hitung_laju_reaksi(perubahan_produk, perubahan_waktu):
        if perubahan_waktu == 0:
            raise ValueError("Perubahan waktu tidak boleh nol!")
        return perubahan_produk / perubahan_waktu
```

Program utama

```
def main():
    try:
        # Input perubahan konsentrasi produk dan waktu
        perubahan_produk = float(input("Masukkan perubahan konsentrasi produk (dalam mol/L): "))
        perubahan_waktu = float(input("Masukkan perubahan waktu (dalam detik): "))
```

```
        # Hitung laju reaksi
        laju_reaksi = PerhitunganLajuReaksi.hitung_laju_reaksi(perubahan_produk,
        perubahan_waktu)
        print(f"Laju reaksi: {laju_reaksi} mol/(L.s)")

    except ValueError as e:
        print(e)
```

```
if name == "main":
    main()
```



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
● PS C:\Users\ilman\Downloads\PBO 2> & C:/Users/ilman/AppD
lman/Downloads/PBO 2/laju.py"
Masukkan perubahan konsentrasi produk (dalam mol/L): 67
Masukkan perubahan waktu (dalam detik): 65
Laju reaksi: 1.0307692307692307 mol/(L.s)
○ PS C:\Users\ilman\Downloads\PBO 2> █
```

10. Class Perhitungan Frekuensi Gelombang

```
class PerhitunganFrekuensiGelombang:
    @staticmethod
    def hitung_frekuensi(kecepatan, panjang_gelombang):
        if panjang_gelombang == 0:
            raise ValueError("Panjang gelombang tidak boleh nol!")
        return kecepatan / panjang_gelombang
```

Program utama

```
def main():
    try:
        # Input kecepatan dan panjang gelombang
        kecepatan = float(input("Masukkan kecepatan gelombang (dalam meter per detik): "))
        panjang_gelombang = float(input("Masukkan panjang gelombang (dalam meter): "))
```

```
        # Hitung frekuensi
        frekuensi = PerhitunganFrekuensiGelombang.hitung_frekuensi(kecepatan,
        panjang_gelombang)
        print(f"Frekuensi gelombang: {frekuensi} Hertz")

    except ValueError as e:
        print(e)
```

```
if name == "main":
    main()
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

- PS C:\Users\ilman\Downloads\PBO 2> & C:/Users/ilman/AppData/Local/Programs/Python/Python39-64/Scripts/python.exe C:/Users/ilman/Downloads/PBO 2/frekuensi_gelombang.py
Masukkan kecepatan gelombang (dalam meter per detik): 54
Masukkan panjang gelombang (dalam meter): 45
Frekuensi gelombang: 1.2 Hertz
- PS C:\Users\ilman\Downloads\PBO 2>