

**LAPORAN TUGAS**  
**ALGORITMA DAN STRUKTUR DATA**  
**PEKAN 3**



**Disusun oleh:**  
**Mhd. Farhan Lubis**  
**L200220277**  
**C**

**TEKNIK INFORMATIKA**  
**FAKULTAS KOMUNIKASI DAN INFORMATIKA**  
**UNIVERSITAS MUHAMMADIYAH SURAKARTA**  
**2023/2024**

## DAFTAR ISI

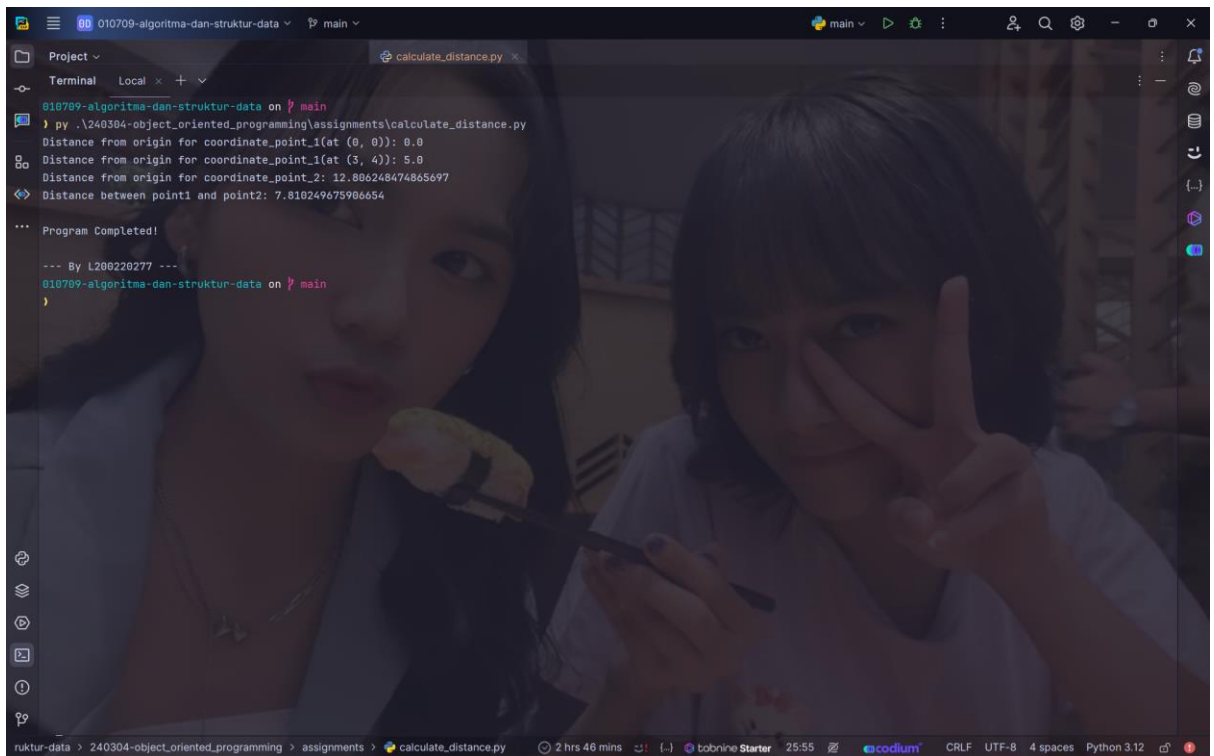
DAFTAR ISI.....	2
TUGAS .....	3
1. Calculate Distance .....	3
2. Inheritance.....	4

## TUGAS

### 1. Calculate Distance

```
# Class ini digunakan untuk memberikan nilai x dan y
# untuk menentukan titik koordinat
# kemudian dapat menemukan jarak anatar titik
# koordinatnya
class CalculateDistance:
    # Konstruktor ini menginisialisasi variabel instance
    # x dan y dengan nilai argumen
    def __init__(self, x, y):
        self.x = x
        self.y = y
    # Mendefinisikan method bernama set_location yang
    # mengambil tiga argumen yaitu self, x, dan y.
    # kemudian variabel x dan y di set dari nilai x dan
    # y dari argumen
    def set_location(self, x, y):
        self.x = x
        self.y = y
    # Mendefinisikan method bernama distance_from_origin
    # yang digunakan untuk
    # Menghitung jarak titik dari titik asal menggunakan
    # rumus jarak Euclidean.
    def distance_from_origin(self):
        return ((self.x ** 2) + (self.y ** 2)) ** 0.5
    # Mendefinisikan method bernama distance yang
    # digunakan untuk
    # menghitung jarak antara 2 titik menggunakan rumus
    # jarak Euclidean.
    def distance(self, other_point):
        return ((self.x - other_point.x) ** 2 + (self.y
- other_point.y) ** 2) ** 0.5

coordinate_point_1 = CalculateDistance(0, 0)
print(f"Distance from origin for coordinate_point_1(at
({coordinate_point_1.x}, {coordinate_point_1.y})): ",
coordinate_point_1.distance_from_origin())
coordinate_point_1.set_location(3, 4)
coordinate_point_2 = CalculateDistance(8, 10)
print(f"Distance from origin for coordinate_point_1(at
({coordinate_point_1.x}, {coordinate_point_1.y})): ",
coordinate_point_1.distance_from_origin())
print("Distance from origin for coordinate_point_2:",
coordinate_point_2.distance_from_origin())
print("Distance between point1 and point2:",
coordinate_point_1.distance(coordinate_point_2))
print("\nProgram Completed!\n\n--- By L200220277 ---")
```



## 2. Inheritance

```
# Class ini berfungsi sebagai blueprint untuk
# membuat objek vehicle yang mempunyai
# atribut name, color, year_production, capacity, dan
# mileage dan method set_capacity
class Vehicle:
    # Konstruktor ini menginisialisasi variabel instance
    # name, color, year_production, capacity, dan
    # mileage dengan nilai argumen
    def __init__(self, name, color, year_production,
capacity, mileage):
        self.name = name
        self.color = color
        self.year_production = year_production
        self.capacity = capacity
        self.mileage = mileage
    # Mendefinisikan method bernama set_capacity yang
    # memiliki argumen self, capacity yang digunakan
    # untuk
    # mengubah nilai capacity
    def set_capacity(self, capacity):
        self.capacity = capacity

# Class ini berfungsi sebagai blueprint untuk objek
# truck yang merupakan
# inherited class dari vehicle sehingga memiliki atribut
# dan method yang di turunkan
```

```

class Truck(Vehicle):
    pass

# instansiasi objek truck yang merupakan inherited class
# dari vehicle
truck = Truck(name="Tesla Truck", color="Gray",
year_production=2024, capacity=9000, mileage=50)
print("Name:", truck.name)
print("Color:", truck.color)
print("Year of Production:", truck.year_production)
print("Capacity:", truck.capacity)
print("Mileage:", truck.mileage)
truck.set_capacity(10000)
print("Updated Capacity:", truck.capacity)
print("\nProgram Completed!\n\n--- By L200220277 ---")

```

```

Project: 010709-algoritma-dan-struktur-data
Terminal: Local
inherentface.py
010709-algoritma-dan-struktur-data on main
> py .\240304-object_oriented_programming\assignments\inherentface.py
Name: Tesla Truck
Color: Gray
Year of Production: 2024
Capacity: 9000
Mileage: 50
*** Updated Capacity: 10000

Program Completed!

--- By L200220277 ---
010709-algoritma-dan-struktur-data on main
>

```

-dan-struktur-data > 240304-object\_oriented\_programming > assignments > inherentface.py 2 hrs 46 mins bobnine Starter 29:26 codium CRLF UTF-8 4 spaces Python 3.12