**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](#_Toc160994968)

[TUGAS 3](#_Toc160994969)

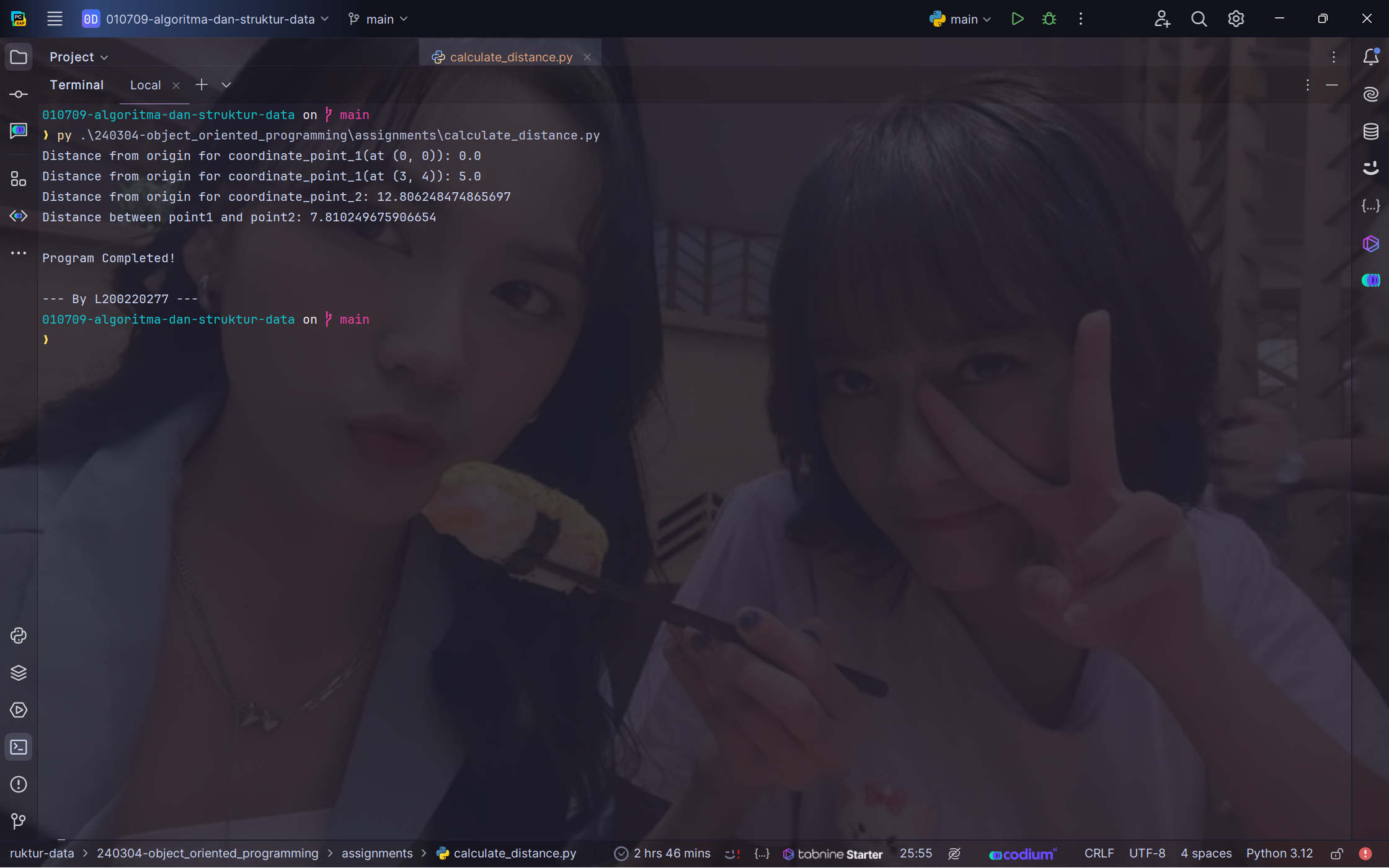
[1. Calculate Distance 3](#_Toc160994970)

[2. Inherintance 4](#_Toc160994971)

# TUGAS

## 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 ---") |



## Inherintance

|  |
| --- |
| *# 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 ---") |

