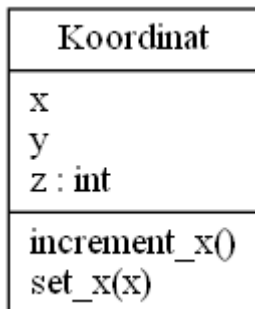


- Aisyah Tiara Pratiwi
- 121450074
- Kelas RC

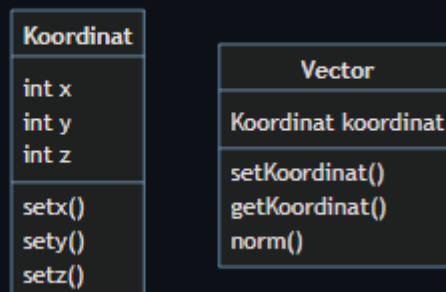
▼ Programming Exercise

- Buatlah class diagram dari code python berikut ini:

```
class Koordinat:
    def __init__(self,x,y):
        self.z = 0
        self.x = x
        self.y = y
    def set_x(self,x):
        self.x = x
    def increment_x(self):
        self.x+=1
    def __str__(self):
        return f"x : {self.x} , y:{self.y} , z:{self.z}"
```



- Implementasikan class diagram berikut menjadi program python!



```

from numpy import sqrt
class Koordinat:
    def __init__(self, x, y, z):
        self.x = x
        self.y = y
        self.z = z

    def setx(self, X):
        self.x = X
        return self.x
    def sety(self, Y):
        self.y = Y
        return self.y

    def setz(self, Z):
        self.z = Z
        return self.z

class Vector(Koordinat):

    def setKoordinat(self, X, Y, Z):
        self.x = X
        self.y = Y
        self.z = Z

    def getKoordinat(self):
        print(f" Koordinat Vektor adalah: {self.x, self.y, self.z}")

    def norm(self):
        Norm = sqrt(self.x**2 + self.y**2 + self.z**2)
        return Norm
koor = Vector(3,6,9)
koor.getKoordinat()
koor.norm()

```

```

    Koordinat Vektor adalah: (3, 6, 9)
    11.224972160321824

```

- fungsi `norm()` adalah fungsi untuk menghitung norm dari sebuah vector yaitu $\text{norm} = \sqrt{x^2 + y^2 + z^2}$. Buatlah fungsi untuk menghitung norm dari vector berdasarkan class yang telah anda buat!

```

class Koordinat:
    def __init__(self,x,y,z):
        self.x = x
        self.y = y
        self.z = z

```

```

def setx(self, x):
    self.x = x

def sety(self, y):
    self.y = y

def setz(self, z):
    self.z = z

class vektor(Koordinat):
    def setKoordinat(self, Koordinat):
        self.Koordinat = Koordinat

    def getKoordinat(self):
        return self.Koordinat

    def norm(self):
        print((self.x**2 + self.y**2 + self.z**2)*0.5)

a = vektor(3,6,9)
a.norm()

63.0

```

- Euclidian Distance adalah jarak antara 2 vector dan didefinisikan sebagai

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (x_1 - x_2)^2}$$

buatlah program untuk menghitung euclidian distance antara 2 vector berdasarkan class yang telah anda buat!

```

import math

class Nilai:
    pass

first = Nilai()
second = Nilai()
first.x = float(input('Koordinat x dari titik pertama:'))
first.y = float(input('Koordinat y dari titik pertama:'))

second.x = float(input('Koordinat x dari titik kedua:'))
second.y = float(input('Koordinat y dari titik kedua:'))

def dist(x1,y1,x2,y2):
    dist = math.sqrt((x2 - x1) * (x2 - x1) + (y2 - y1) * (y2 - y1))

```

```
return dist
```

```
dist = dist(first.x, first.y, second.x, second.y)
```

```
print("Jarak euclidian antara 2 vector :",dist)
```



Koordinat x dari titik pertama:5
 Koordinat y dari titik pertama:6
 Koordinat x dari titik kedua:3
 Koordinat y dari titik kedua:4
 Jarak euclidian antara 2 vector : 2.8284271247461903

- Perhatikan tabel berikut

Vector	x	y	z	kelas
A	0	1	1	0
B	2	2	2	1
C	1.5	1.2	1.5	1
D	10	9	8	0
E	5	5	5	0
F	8	8	8	0
G	6	6	1	0
H	3	3	3	?

Salah satu metode dalam data science untuk melakukan klasifikasi adalah dengan menggunakan k-nearest neighbor, dimana cara menentukan label kelas adalah dengan cara melihat tetangga dengan euclidian distance terdekat. Tentukan kelas dari vector H dengan cara:

- Tentukan 3 tetangga paling dekat dari H
- dari 3 tetangga tersebut, pilih kelas yang paling banyak muncul
- Jadikan kelas yang paling banyak muncul tersebut sebagai kelas dari vector H

Implementasikan algoritma tersebut menggunakan program python dengan paradigma object-oriented programming yang telah anda pelajari!

```
class Koordinat:
    def __init__(self, x, y, z):
        self.x = x
        self.y = y
        self.z = z

    def setx(self):
        return self.x

    def sety(self):
        return self.y
```

```

def setz(self):
    return self.z

class Vector:
    def getKoordinat(self,setx,sety,setz):
        self.setx = setx
        self.sety = sety
        self.setz = setz

    def setKoordinat(self,k):
        koordinat = (k.setx(), k.sety(), k.setz())
        return koordinat

    def norm(self,a,b):
        sum = 0
        self.a = self.setKoordinat(a)
        self.b = self.setKoordinat(b)
        for n in range (3):
            sum = sum + (self.a[n]-self.b[n])**2
        return sum**0.5

a = Koordinat( 0 , 1 , 1 )
b = Koordinat( 2 , 2 , 2 )
c = Koordinat( 1.5 , 1.2, 1.5 )
d = Koordinat( 10 , 9 , 8 )
e = Koordinat( 5 , 5 , 5 )
f = Koordinat( 8 , 8 , 8 )
g = Koordinat( 6 , 6 , 1 )
h = Koordinat( 3 , 3 , 3 )
v = Vector()
titik = [a,b,c,d,e,f,g,h]

output = []
for n in range (len(titik)):
    panjang = v.norm(titik[n],titik[7])
    output.append(panjang)
    #print('Jarak titik H ke titik',titik[n],panjang)

print(output)
output.sort()
print("tetangga paling dekat dengan H yaitu:", output[1])
print("tetangga kedua paling dekat dengan H yaitu:", output[2])
print("tetangga ketiga paling dekat dengan H yaitu:", output[3])

[4.123105625617661, 1.7320508075688772, 2.7820855486487113, 10.488088481701515, 3.464101
tetangga paling dekat dengan H yaitu: 1.7320508075688772
tetangga kedua paling dekat dengan H yaitu: 2.7820855486487113
tetangga ketiga paling dekat dengan H yaitu: 3.4641016151377544

```

Produk berbayar Colab - [Batalan kontrak di sini](#)

