

LAPORAN
Komputer Grafik
Pertemuan 3 - 4
Bentuk Dasar dengan Transformasi



Disusun Oleh :
Yoga Nizar Habibulloh
201511064
D3-3B

PROGRAM STUDI D3-TEKNIK INFORMATIKA
JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA
POLITEKNIK NEGERI BANDUNG
2022

DAFTAR ISI

PERTEMUAN 2 BENTUK DASAR

Task 1 Fungsi Operasi Matrix dan Primitif	3
Task 2 Bentuk 2D.....	6
Task 3 Pola Lingkaran dengan Rotasi	7
Task 4 Pola Ellips Dan Rotasi.....	8
(Bunga 4 Kelopak dan Bunga 8 Kelopak)	8
Lesson Learn	10

Task 1 Fungsi Operasi Matrix dan Primitif

Pertama – tama saya membuat sebuah fungsi untuk melakukan operasi – operasi pada matriks, diantaranya adalah dengan membuat sebuah variabel baru untuk menampung matriks, dalam contoh ini saya membuat matriks dengan ordo 3x3 , dengan inisiasi awal bernilai 0. Tertera pada gambar 1 dibawah ini.

```
extends Node2D  
  
var theMatrix = [[0,0,0],[0,0,0],[0,0,0]]
```

Gambar 1

Selanjutnya, membuat fungsi matriks untuk operasi matriks identitas. Dengan membuat sebuah variabel i dan j untuk menampung baris dan kolom pada matriks. Dan membuat sebuah perulangan *for* untuk mengisi indeks dari matriks tersebut, kemudian membuat sebuah pengkondisian *if else* untuk kondisi jika nilai indeks i sama dengan j maka ordo akan diisi nilai 1 / true, jika indeks i tidak sama dengan j maka akan diisi dengan nilai 0 atau false. Tertera pada gambar 2 dibawah ini

```
func matrix3x3SetIdentity(m) :  
>| for i in range(3) :  
>| >| for j in range(3) :  
>| >| >| if i == j :  
>| >| >| m[i][j] = 1  
>| >| >| else :  
>| >| >| m[i][j] = 0>|
```

Gambar 2

Selanjutnya, membuat fungsi matriks untuk operasi matriks perkalian, dengan membuat fungsi bersama dengan parameter (a,b) dan ordo 3x3, Pertama – tama membuat sebuah variable untuk menampung indeks ordo dari matriks tersebut, kemudian melakukan perulangan pada variabel r dan c. perkalian dilakukan pada matriks a, dan menyimpan hasil perkalian matriks pada matriks b. tertera pada gambar 3 dibawah ini.

```
func matrix3x3PreMultiply(a,b) :  
>| var tmp = [[0,0,0],[0,0,0],[0,0,0]]  
>|  
>| for r in range(3) :  
>| >| for c in range(3) :  
>| >| >| tmp[r][c] = a[r][0] * b[0][c] + a[r][1] * b[1][c] + a[r][2] * b[2][c]  
>| >|  
>| for r in range(3) :  
>| >| for c in range(3) :  
>| >| >| b[r][c] = tmp[r][c]
```

Gambar 3

Selanjutnya, membuat fungsi matriks untuk operasi matriks pertambahan, dengan membuat fungsi bersama dengan parameter (a,b). terdapat variabel r dan c sebagai penampung dari nilai matriks pertama, dan menjumlahkannya pada matriks kedua, matriks a dengan indeks r dan c ditambahkan dengan matriks b dengan indeks r dan c. tertera pada gambar 4 dibawah ini

```
func matrix3x3Add(a,b) :
    for r in range(3) :
        for c in range(3) :
            a[r][c] += b[r][c]
```

Gambar 4

Selanjutnya, membuat fungsi untuk operasi matriks pengurangan. Untuk fungsi pengurangan matriks kurang lebih secara struktur sama dengan struktur operasi penambahan pada matriks, perbedaan nya terletak pada operasi pengurangannya saja. Tertera pada gambar 5 dibawah ini.

```
func matrix3x3Subtract(a,b) :
    for r in range(3) :
        for c in range(3) :
            a[r][c] -= b[r][c]
```

Gambar 5

Selanjutnya, membuat sebuah fungsi scale, rotasi, translasi, transformasi. Fungsi – fungsi yang telah dibuat tersebut adalah untuk pengimplementasian pada bentuk dasar / shape. tertera pada gambar 6 dan 7 dibawah ini.

```
extends "res://064_Pertemuan_3_4_task1/matrix.gd"

var width
var height
var center_x
var center_y

# Called when the node enters the scene tree for the first time.
func _ready():
    pass

func _init():
    width = ProjectSettings.get("display/window/size/width")
    height = ProjectSettings.get("display/window/size/height")
    center_x = width/2
    center_y = height/2

func put_pixel(x, y, color):
    draw_primitive(PoolVector2Array([Vector2(x, y)]), PoolColorArray([color]), PoolVector2Array())

func put_pixel_2(x,y,color) :
    return draw_primitive(PoolVector2Array([Vector2(x, y)]), PoolColorArray([color]), PoolVector2Array())

func translate2 (tx : int, ty : int, matrix_temp) :
    var m = [[0,0,0], [0,0,0], [0,0,0]]

    matrix3x3SetIdentity(m)
    m[0][2] = tx
    m[1][2] = ty
    matrix3x3PreMultiply(m, matrix_temp)

func scale2(sx : float, sy : float, refpt : Vector2, matrix_temp) :
    var m = [[0,0,0],[0,0,0],[0,0,0]]

    matrix3x3SetIdentity(m)
    m[0][0] = sx
    m[0][2] = (1 - sx) * refpt.x
```

Gambar 6

```

Online Docs Search
71 m[1][1] = sy
72 m[1][2] = (1 - sy) * refpt.y
73 matrix3x3PreMultiply(m, matrix_temp)
74
75 func rotate2(a : float, refpt : Vector2, matrix_temp) :
76     var m = [[0,0,0],[0,0,0],[0,0,0]]
77
78     matrix3x3SetIdentity(m)
79     a = deg2rad(a)
80     m[0][0] = cos(a)
81     m[0][1] = -sin(a)
82     m[0][2] = refpt.x * (1 - cos(a)) + refpt.y * sin(a)
83     m[1][0] = sin(a)
84     m[1][1] = cos(a)
85     m[1][2] = refpt.y * (1 - cos(a)) - refpt.x * sin(a)
86     matrix3x3PreMultiply(m, matrix_temp)
87
88 func transformPoints2(npts : int, pts, matrix_temp) :
89     var tmp
90     var temp = []
91     temp.append_array(pts)
92
93     for k in range(npts) :
94         tmp = matrix_temp[0][0] * pts[k].x + matrix_temp[0][1] * pts[k].y + matrix_temp[0][2]
95         temp[k].y = matrix_temp[1][0] * pts[k].x + matrix_temp[1][1] * pts[k].y + matrix_temp[1][2]
96         temp[k].x = tmp
97     pts.resize(0)
98     pts.append_array(temp)
99
100 func rotate2_segi_lima(a: float, refpt: Vector2):
101     var m=[[0,0,0],[0,0,0],[0,0,0]]
102
103     matrix3x3SetIdentity(m)
104     a = deg2rad(a)
105     m[0][0] = cos(a)
106     m[0][1] = -sin(a)
107     m[0][2] = refpt.x * (1 - cos(a)) + refpt.y * sin(a)
108     m[1][0] = sin(a)
109     m[1][1] = cos(a)
110     m[1][2] = refpt.y * (1 - cos(a)) - refpt.x * sin(a)
111     matrix3x3PreMultiply(m, theMatrix)

```

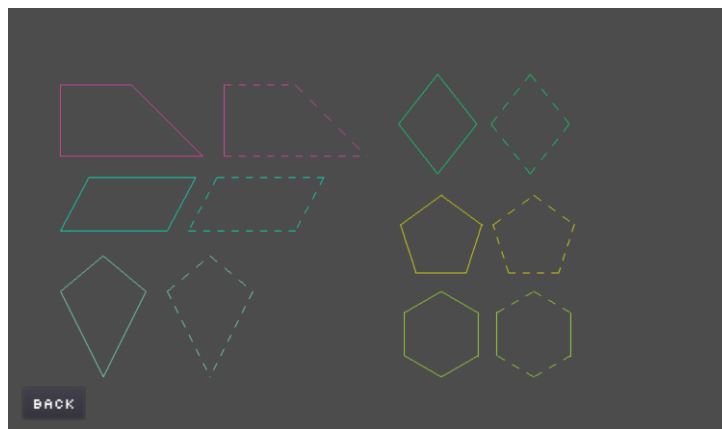
Gambar 7

Task 2 Bentuk 2D

Pada task 2 ini, saya membuat fungsi untuk membuat sebuah bentuk / shape sederhana, mereferensikan dari pertemuan 2 sebelumnya tentang bagaimana membuat sebuah garis yang akan membentuk sebuah bentuk shape sederhana. Tertera pada Gambar 8, dan 9 dibawah ini

```
1 extends "res://064_Pertemuan_3_4_task2/shape.gd"
2
3 func _ready():
4     pass
5
6 func _process(delta):
7     pass
8
9 func _draw():
10    trapesium_siku(Vector2(75,110),100,100,200,Color("#FF44BF"))
11    trapesium_siku(Vector2(305,110),100,100,200,Color("#FF44BF"),true)
12
13    jajar_genjang(Vector2(75,240),40,150,75,Color("#00FFD1"))
14    jajar_genjang(Vector2(255,240),40,150,75,Color("#00FFD1"),true)
15
16    layang_layang(Vector2(75,400),120,50,120,Color.aquamarine)
17    layang_layang(Vector2(225,400),120,50,120,Color.aquamarine,true)
18
19    ketupat(Vector2(550,165),110,140,Color.springgreen)
20    ketupat(Vector2(680,165),110,140,Color.springgreen,true)
21
22    segi_lima(Vector2(610,325),60,Color.yellow)
23    segi_lima(Vector2(740,325),60,Color.yellow,true)
24
25    segi_enam(Vector2(610,460),60,Color.greenyellow)
26    segi_enam(Vector2(740,460),60,Color.greenyellow,true)
27
```

Gambar 8



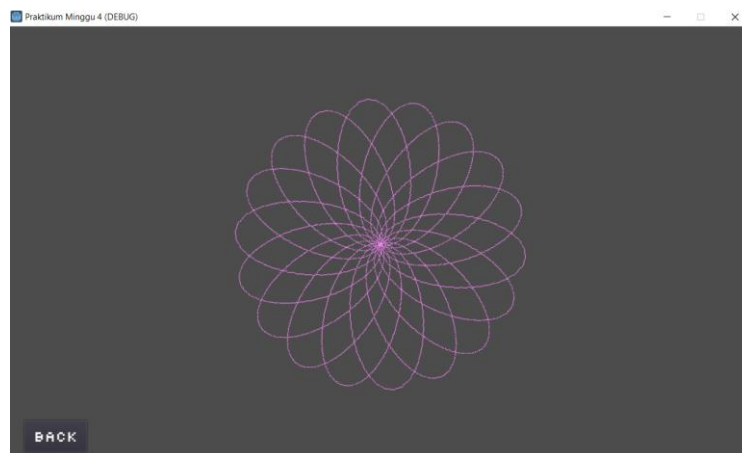
Gambar 9

Task 3 Pola Lingkaran dengan Rotasi

Selanjutnya membuat sebuah fungsi overlapping lingkaran. Pertama – tama saya membuat sebuah fungsi ellipse midpoint sesuai dengan referensi yang tertera pada ppt praktikum di google classroom, selanjutnya membuat sebuah fungsi translasi, rotasi serta transformasi untuk perubahan arah bentuk shape yang terjadi pada lingkaran ellips tersebut. Kemudian membuat sebuah variabel insiasi i untuk nilai awal yang selanjutnya akan dimasukan kedalam proses perulangan while dengan kondisi $j \leq 20$, (angka kondisi dalam looping while dapat dirubah sesuai dengan kondisi yang disesuaikan) selanjutnya didalam proses looping selanjutnya menambahkan variabel inisiasi i dengan nilai 0 yang kemudian di masukan kedalam loopin while dengan kondisi $i < \text{ukuran size dari indeks } i$, selanjutnya terdapat proses iterasi penambahan pada variabel I yang dengan aturan i ditambah nilai nya dengan 1, dan j ditambah nilainya dengan 1. Terteta pada gambar 10 dan 11 dibawah ini.

```
func overlapping(center: Vector2) :  
    #print(ident)  
    var temp = ellipseMidpoint_2(512,300,100,50, Color.green)  
    var ident = [[0,0,0],[0,0,0],[0,0,0]]  
    var point  
  
    matrix3x3SetIdentity(ident)  
    translate2(100,0,ident)  
    rotate2(45, Vector2(512,300), ident)  
    transformPoints2(temp.size(), temp, ident)  
  
    var j = 1  
    while(j <= 20) :  
        >| matrix3x3SetIdentity(ident)  
        >| rotate2(20, Vector2(512,300), ident)  
        >| transformPoints2(temp.size(), temp, ident)  
  
        >| var i = 0  
        >| while(i < temp.size()) :  
        >| >| put_pixel(temp[i].x,temp[i].y,Color.violet)  
        >| >| i += 1  
        >|  
        >| j+=1
```

Gambar 10



Gambar 11

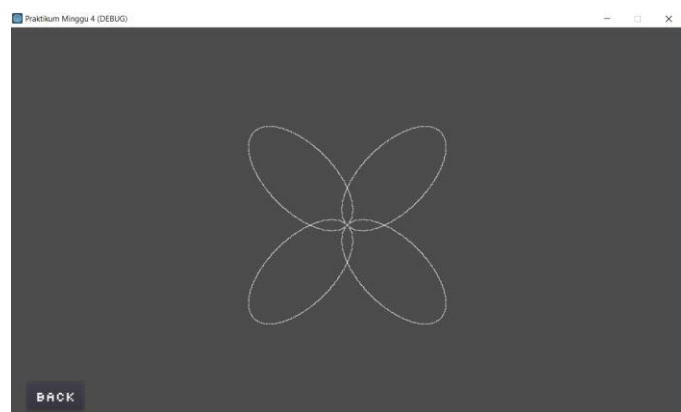
Task 4 Pola Ellips Dan Rotasi

(Bunga 4 Kelopak dan Bunga 8 Kelopak)

Selanjutnya membuat sebuah fungsi untuk membuat fungsi untuk membuat bunga dengan 4 kelopak, pertama tama seperti rotasi lingkaran yang sebelumnya sebagai inisiasi saya menggunakan fungsi ellipse midpoint yang ada pada ppt praktikum yang pada google classroom. Selanjutnya membuat variabel inisiasi i dengan nilai 1 dan melakukan pengulangan while terhadap j dengan kondisi $j \leq 4$ untuk membuat kelopak yang berjumlah 4, dikarenakan iterasi akan berhenti pada 4 pada saat variabel inisiasi i yang nilai awalnya 0 akan mengalami increment penambahan dengan nilai 1. Dan variabel inisiasi j mengalami increment +1. Tertera pada gambar 12 dan 13 dibawah berikut.

```
func kelopak4(center: Vector2) :
>| #print(ident)
>| var temp = ellipseMidpoint_2(512,300,100,50, Color.cyan)
>| var ident = [[0,0,0],[0,0,0],[0,0,0]]
>| var point
>|
>| matrix3x3SetIdentity(ident)
>| translate2(100,0,ident)
>| rotate2(45, Vector2(512,300), ident)
>| transformPoints2(temp.size(), temp, ident)
>|
>| var j = 1
>| while(j <= 4) :
>| >| matrix3x3SetIdentity(ident)
>| >| rotate2(90, Vector2(512,300), ident)
>| >| transformPoints2(temp.size(), temp, ident)
>|
>| >| var i = 0
>| >| while(i < temp.size()) :
>| >| >| put_pixel(temp[i].x,temp[i].y,Color.white)
>| >| >| i += 1
>| >|
>| >| j+=1
```

Gambar 12

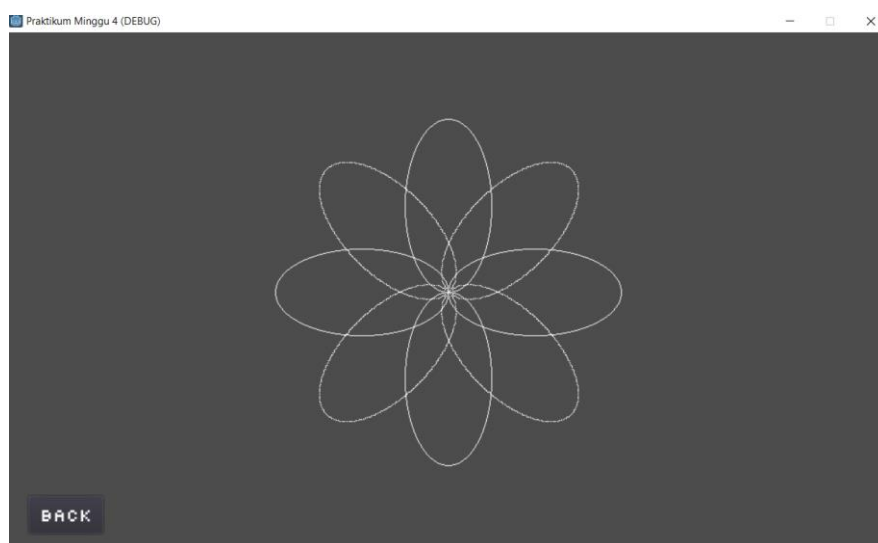


Gambar 13

Selanjutnya membuat sebuah fungsi untuk membuat fungsi untuk membuat bunga dengan 8 kelopak, pertama tama seperti rotasi lingkaran yang sebelumnya sebagai inisiasi saya menggunakan fungsi ellipse midpoint yang ada pada ppt praktikum yang pada google classroom. Selanjutnya membuat variabel inisiasi i dengan nilai 1 dan melakukan pengulangan while terhadap j dengan kondisi $j \leq 8$ untuk membuat kelopak yang berjumlah 8, dikarenakan iterasi akan berhenti pada 8 pada saat variabel inisiasi i yang nilai awalnya 0 akan mengalami increment penambahan dengan nilai 1. Dan variabel inisiasi j mengalami increment +1. Tertera pada gambar 14 dan 15 dibawah berikut.

```
func kelopak8(center: Vector2) :
>I #print(ident)
>I var temp = ellipseMidpoint_2(512,300,100,50, Color.yellow)
>I var ident = [[0,0,0],[0,0,0],[0,0,0]]
>I var point
>I
>I matrix3x3SetIdentity(ident)
>I translate2(100,0,ident)
>I rotate2(45, Vector2(512,300), ident)
>I transformPoints2(temp.size(), temp, ident)
>I
>I var j = 1
>I while(j <= 8) :
>I >I matrix3x3SetIdentity(ident)
>I >I rotate2(45, Vector2(512,300), ident)
>I >I transformPoints2(temp.size(), temp, ident)
>I
>I >I var i = 0
>I >I while(i < temp.size()) :
>I >I >I put_pixel(temp[i].x,temp[i].y,Color.white)
>I >I >I i += 1
>I >I
>I i+=1
```

Gambar 14



Gambar 15

Lesson Learn

Pada pengimplementasian task ini, saya mempelajari bagaimana penerapan dari midpoint untuk ellipse dan circle. Beberapa bentuk kombinasi seperti kelopak bunga dan rotasi lingkaran, selain itu saya mempelajari bagaimana operasi aritmatika sederhana dari matriks, mulai dari perkalian matriks, penjumlahan matriks, pengurangan matriks, dan perkalian matriks.