

**PERTEMUAN 4 GODOT**  
**LAPORAN PRAKTIKUM**

Disusun untuk memenuhi tugas Mata Kuliah Komputer Grafik

Disusun oleh  
Nazwa Fitriyani Zahra      211511051



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

## DAFTAR ISI

<b>DAFTAR ISI.....</b>	<b>2</b>
<b>DAFTAR REFERENSI.....</b>	<b>3</b>
<b>A. Task 1.....</b>	<b>4</b>
<b>B. Task 1.....</b>	<b>5</b>
<b>C. Task 1.....</b>	<b>5</b>
<b>D. Task 1.....</b>	<b>7</b>
<b>KOTRETAN.....</b>	<b>9</b>
<b>LESSON LEARN.....</b>	<b>10</b>
<b>CURHAT DOSEN .....</b>	<b>10</b>

## **DAFTAR REFERENSI**

- <https://docs.godotengine.org/en/3.5/index.html>
- <https://github.com/godotengine/godot-demo-projects>
- <https://docs.godotengine.org/en/3.5/community/tutorials.html>
- [https://www.youtube.com/watch?v=ZUPBoqC\\_X\\_o](https://www.youtube.com/watch?v=ZUPBoqC_X_o)
- YouTube Channel : Kelas Terbuka – Godot Tutorial
- [https://docs.godotengine.org/en/stable/tutorials/2d/custom\\_drawing\\_in\\_2d.html](https://docs.godotengine.org/en/stable/tutorials/2d/custom_drawing_in_2d.html)
- [https://docs.godotengine.org/en/stable/tutorials/2d/custom\\_drawing\\_in\\_2d.html](https://docs.godotengine.org/en/stable/tutorials/2d/custom_drawing_in_2d.html)
- [https://docs.godotengine.org/en/3.1/tutorials/math/matrices\\_and\\_transforms.html](https://docs.godotengine.org/en/3.1/tutorials/math/matrices_and_transforms.html)

## A. Task 1

Berikut operasi matrix di godot engine

```
24 #Task 1 : Operasi Matrix
25 #Tambahkan primitif.gd memiliki fungsi operasi matrix Tambah, Kurang, Kali, Matrix Identitas
26 func tambahMatriks (a:Array, b:Array):
27     pass
28     var result = [0,0]
29     result[0] = a[0] + b[0]
30     result[1] = a[1] + b[1]
31     print(result)
32     return result
33
34 func kurangMatriks (a:Array, b:Array):
35     pass
36     var result = [0,0]
37     result[0] = a[0] - b[0]
38     result[1] = a[1] - b[1]
39     print(result)
40     return result
41
42 func kaliMatriks (a:Array, b:Array):
43     pass
44     var result
45
46     result = [a[0][0]*b[0][0]+a[0][1]*b[1][0], a[1][0]*b[0][0]+a[1][1]*b[1][0]]
47     print(result)
48     return result
49
50 func matriksIdentitas():
51     pass
52     var result = [[1,0],[0,1]]
53     print(result)
54     return result
55
```

Hasil operasi :

```
18 func _draw():
19     tambahMatriks([0,5], [1,2])
20     kurangMatriks([0,5], [1,2])
21     kaliMatriks ([[1,1],[1,2]], [2,2])
22     matriksIdentitas()
23
>

Output:
--- Debugging process started ---
Godot Engine v3.5.stable.official.991bb6ac7 - https://godotengine.org
OpenGL ES 2.0 Renderer: NVIDIA GeForce 840M/PCIe/SSE2

[1, 7]
[-1, 3]
[4, 6]
[[1, 0], [0, 1]]
```

## B. Task 1

Berikut operasi transformasi yang telah dibuat

```
59 #Task 2 : Operasi Matrix
60 #Buatlah gdscrip (primitif.gd) memiliki fungsi-fungsi berikut: scale, rotation, translation
61 func scale(titik:Array, center:Array, scaleVar):
62     var result
63     result = kurangMatriks(titik, center)
64     result = kaliMatriks(result,scaleVar)
65     result = tambahMatriks(result, center)
66     return result
67
68 func rotationIdentity(deg):
69     var result = [[cos(deg2rad(deg)), -sin(deg2rad(deg))],
70     [sin(deg2rad(deg)), cos(deg2rad(deg))]]
71     return result
72
73 func rotation(titik:Array, center:Array, deg):
74     var result
75     result = kurangMatriks(titik, center)
76     result = kaliMatriks(rotationIdentity(deg),result)
77     result = tambahMatriks(result,center)
78     print(result)
79     return result
80
81 func translation(a:Array, b:Array):
82     var result = tambahMatriks(a,b)
83     return result
84
```

## C. Task 1

Berikut fungsi bentuk menggunakan transformasi

```
16 #pass
17 func persegi(titik_awal: Vector2, panjang_sisi):
18     var res = PoolVector2Array()
19     point_point(titik_awal.x, titik_awal.y, titik_awal.x + panjang_sisi, titik_awal.y) #sisi atas
20     point_point(titik_awal.x, titik_awal.y + panjang_sisi, titik_awal.x + panjang_sisi, titik_awal.y + panjang_sisi) #sisi bawah
21     point_point(titik_awal.x, titik_awal.y, titik_awal.x, titik_awal.y + panjang_sisi) #sisi kiri
22     point_point(titik_awal.x + panjang_sisi, titik_awal.y, titik_awal.x + panjang_sisi, titik_awal.y + panjang_sisi) #sisi kanan
23     return res
24
25 func persegi_panjang(titik_awal: Vector2, panjang, lebar):
26     var res = PoolVector2Array()
27     point_dash(titik_awal.x, titik_awal.y, titik_awal.x + panjang, titik_awal.y) #sisi atas
28     point_dash(titik_awal.x, titik_awal.y + lebar, titik_awal.x + panjang, titik_awal.y + lebar) #sisi bawah
29     point_dash(titik_awal.x, titik_awal.y, titik_awal.x, titik_awal.y + lebar) #sisi kiri
30     point_dash(titik_awal.x + panjang, titik_awal.y, titik_awal.x + panjang, titik_awal.y + lebar) #sisi kanan
31     return res
32
33 func segitiga_siku_siku(titik_awal: Vector2, alas, tinggi):
34     var res = PoolVector2Array()
35     dash_dash(titik_awal.x, titik_awal.y, titik_awal.x, titik_awal.y + tinggi) #sisi kiri
36     dash_dash(titik_awal.x, titik_awal.y + tinggi, titik_awal.x + alas, titik_awal.y + tinggi) #sisi bawah
37     dash_dash(titik_awal.x, titik_awal.y, titik_awal.x + alas, titik_awal.y + tinggi) #sisi kanan
38     return res
39
40 func trapesium_siku_siku(titik_awal: Vector2, atas, tinggi, bawah):
41     var res = PoolVector2Array()
42     dash_dash(titik_awal.x, titik_awal.y, titik_awal.x + atas, titik_awal.y) #sisi atas
43     dash_dash(titik_awal.x, titik_awal.y + tinggi, titik_awal.x + bawah + round(abs(bawah-atas)/2), titik_awal.y + tinggi) #sisi bawah
44     dash_dash(titik_awal.x, titik_awal.y, titik_awal.x, titik_awal.y + tinggi) #sisi kiri
45     dash_dash(titik_awal.x + atas, titik_awal.y, titik_awal.x + bawah, titik_awal.y + tinggi) #sisi kanan
46     return res
47
```

```

47
48 ~ func jajargenjang(titik_awal: Vector2, panjang, tinggi, geser):
49   > var res = PoolVector2Array()
50   > dash_dash(titik_awal.x, titik_awal.y, titik_awal.x + panjang, titik_awal.y) #sisi atas
51   > dash_dash(titik_awal.x - geser, titik_awal.y + tinggi, (titik_awal.x + panjang - geser), titik_awal.y + tinggi) #sisi bawah
52   > res.append_array(lineDDA(titik_awal.x, titik_awal.y, (titik_awal.x - geser), titik_awal.y + tinggi)) #sisi kiri
53   > res.append_array(lineDDA(titik_awal.x + panjang, titik_awal.y, ((titik_awal.x + panjang) - geser), titik_awal.y + tinggi)) #sisi kanan
54   > return res
55
56 #Ketupat tanpa rotasi
57 #func ketupattanparotasi(titik_awal: Vector2, diagonal):
58   > #var res = PoolVector2Array()
59   > #point_dash(titik_awal.x, titik_awal.y, titik_awal.x, titik_awal.y + diagonal)
60   > #point_dash(titik_awal.x - (diagonal/2), titik_awal.y + (diagonal/2), titik_awal.x + (diagonal/2), titik_awal.y + (diagonal/2))
61   > #res.append_array(lineDDA(titik_awal.x, titik_awal.y, titik_awal.x - (diagonal/2), titik_awal.y + (diagonal/2)))
62   > #res.append_array(lineDDA(titik_awal.x, titik_awal.y, titik_awal.x + (diagonal/2), titik_awal.y + (diagonal/2)))
63   > #res.append_array(lineDDA(titik_awal.x - (diagonal/2), titik_awal.y + (diagonal/2), titik_awal.x, titik_awal.y + diagonal))
64   > #res.append_array(lineDDA(titik_awal.x + (diagonal/2), titik_awal.y + (diagonal/2), titik_awal.x, titik_awal.y + diagonal))
65   > #return res
66
67 ~ func layanglayang (titik_awal: Vector2, diagonal_a, diagonal_b):
68   > var res = PoolVector2Array()
69   > point_dash(titik_awal.x, titik_awal.y, titik_awal.x, titik_awal.y + diagonal_a)
70   > point_dash(titik_awal.x - (diagonal_b/2), titik_awal.y + (diagonal_a/3), titik_awal.x + (diagonal_b/2), titik_awal.y + (diagonal_a/3))
71   > res.append_array(bold(titik_awal.x, titik_awal.y, titik_awal.x - (diagonal_b/2), titik_awal.y + (diagonal_a/3)))
72   > res.append_array(bold(titik_awal.x, titik_awal.y, titik_awal.x + (diagonal_b/2), titik_awal.y + (diagonal_a/3)))
73   > res.append_array(bold(titik_awal.x - (diagonal_b/2), titik_awal.y + (diagonal_a/3), titik_awal.x, titik_awal.y + diagonal_a))
74   > res.append_array(bold(titik_awal.x + (diagonal_b/2), titik_awal.y + (diagonal_a/3), titik_awal.x, titik_awal.y + diagonal_a))
75   > return res
76   >
77

```

```

67 ~ func layanglayang (titik_awal: Vector2, diagonal_a, diagonal_b):
68   > var res = PoolVector2Array()
69   > point_dash(titik_awal.x, titik_awal.y, titik_awal.x, titik_awal.y + diagonal_a)
70   > point_dash(titik_awal.x - (diagonal_b/2), titik_awal.y + (diagonal_a/3), titik_awal.x + (diagonal_b/2), titik_awal.y + (diagonal_a/3))
71   > res.append_array(bold(titik_awal.x, titik_awal.y, titik_awal.x - (diagonal_b/2), titik_awal.y + (diagonal_a/3)))
72   > res.append_array(bold(titik_awal.x, titik_awal.y, titik_awal.x + (diagonal_b/2), titik_awal.y + (diagonal_a/3)))
73   > res.append_array(bold(titik_awal.x - (diagonal_b/2), titik_awal.y + (diagonal_a/3), titik_awal.x, titik_awal.y + diagonal_a))
74   > res.append_array(bold(titik_awal.x + (diagonal_b/2), titik_awal.y + (diagonal_a/3), titik_awal.x, titik_awal.y + diagonal_a))
75   > return res
76   >
77
78 ~ func segilima(pusat:Array, panjangDiagonal):
79   > var titikPusat = pusat
80   > var titik1 = [titikPusat[0]-panjangDiagonal/2,titikPusat[1]]
81   > var titik2 = rotation(titik1,titikPusat,72)
82   > for i in 5:
83     > pass
84     > lineDDA(titik1[0],titik1[1],titik2[0],titik2[1])
85     > titik1 = titik2
86     > titik2 = rotation(titik1,titikPusat,72)
87   >
88 ~ func segiEnam(pusat:Array, panjangDiagonal):
89   > var titikPusat = pusat
90   > var titik1 = [titikPusat[0]-panjangDiagonal/2,titikPusat[1]]
91   > var titik2 = rotation(titik1,titikPusat,60)
92   > for i in 6:
93     > pass
94     > lineDDA(titik1[0],titik1[1],titik2[0],titik2[1])
95     > titik1 = titik2
96     > titik2 = rotation(titik1,titikPusat,60)
97   >
98   >
99

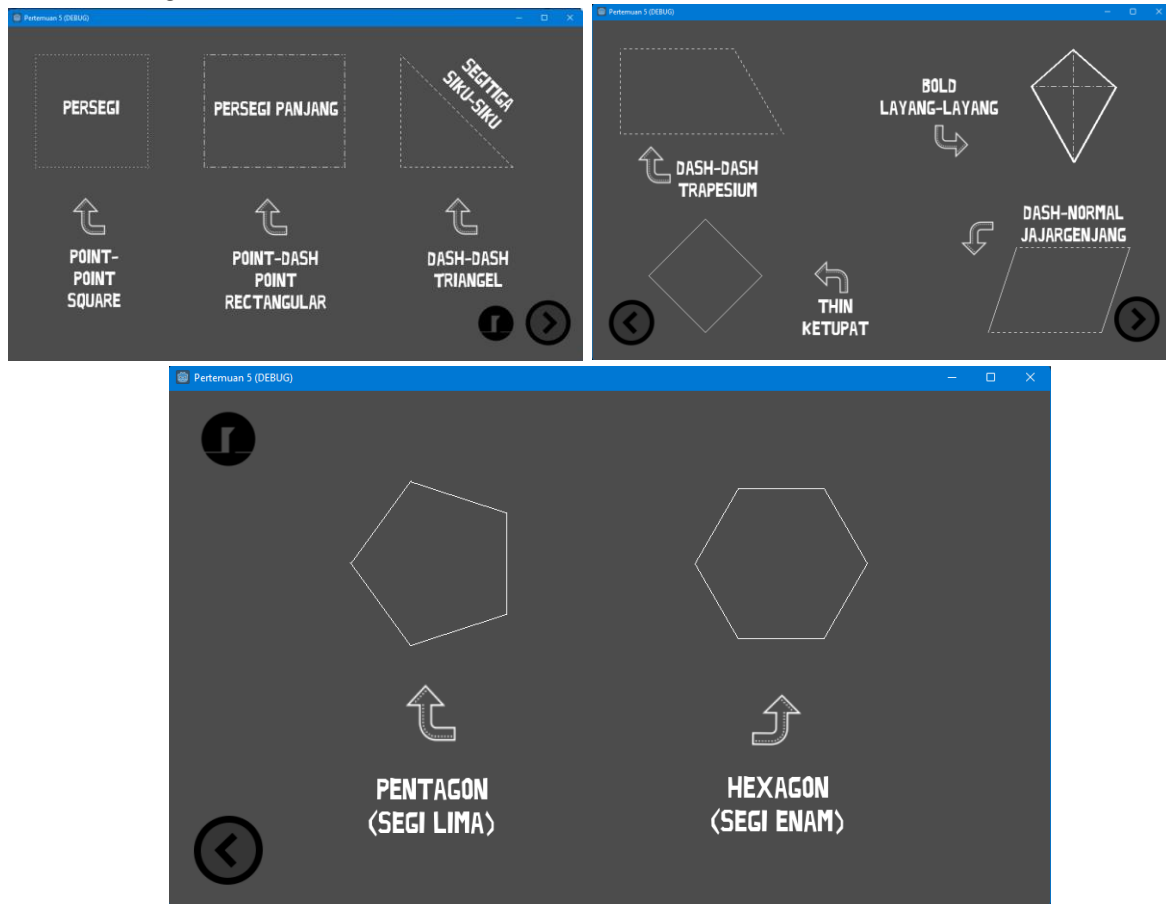
```

```

98 #Ketupat dengan rotasi
99 ~ func ketupat(pusat:Array, panjangDiagonal):
100   > var titikPusat = pusat
101   > var titik1 = [titikPusat[0]-panjangDiagonal/2,titikPusat[1]]
102   > var titik2 = rotation(titik1,titikPusat,90)
103   > for i in 4:
104     > pass
105     > lineDDA(titik1[0],titik1[1],titik2[0],titik2[1])
106     > titik1 = titik2
107     > titik2 = rotation(titik1,titikPusat,90)
108   >
109

```

## Hasil running



## D. Task 1

Berikut adalah pembuatan bunga kelopak 4 dan 8

```

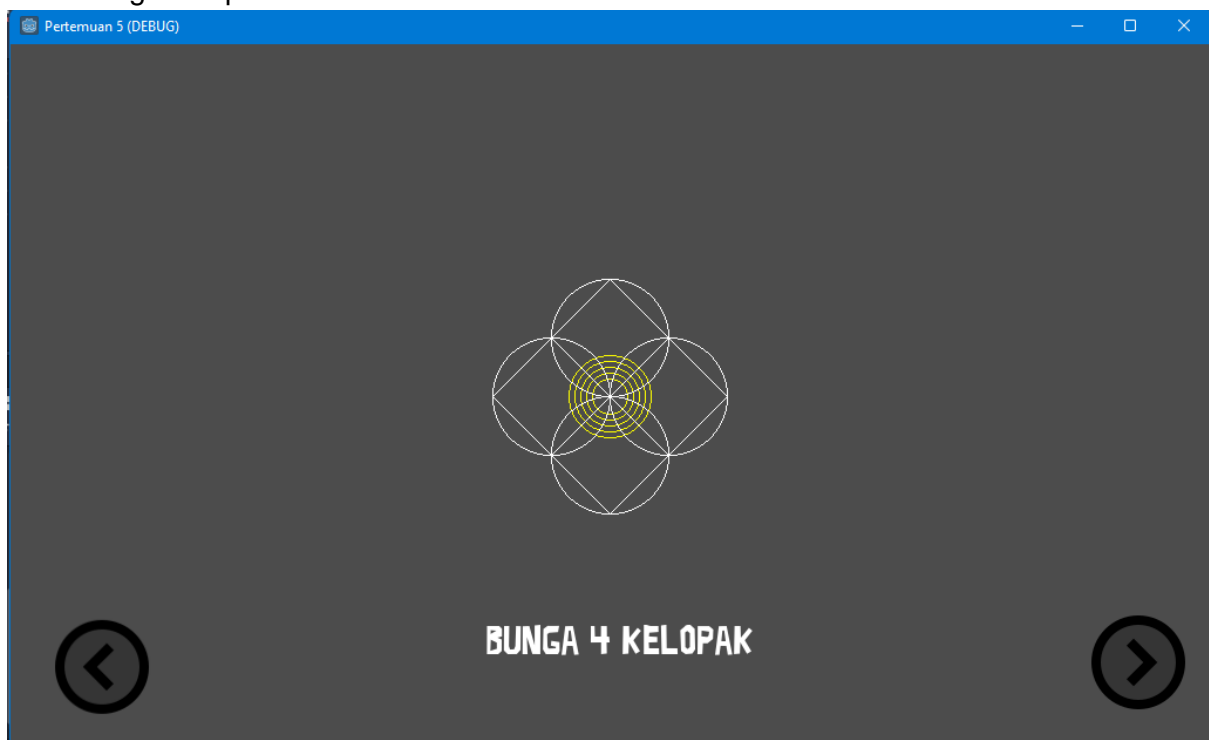
17
18 < func Kelopak(titikPusat:Array, diagonal, besarKelopak, jumlahKelopak):
19 > pass
20 > var putaran = 360/jumlahKelopak
21 > var titikPutar = [titikPusat[0],titikPusat[1]-diagonal/2] # [50,25]
22 > var rotationTitikPutar = rotation(titikPutar,titikPusat,putaran) # []
23 > for i in jumlahKelopak:
24 > > #circleMidPoint(titikPutar[0],titikPutar[1],besarKelopak,Color(1,1,1,1))
25 > > circleMidPoint(titikPutar[0],titikPutar[1],besarKelopak,Color(1,1,1,1))
26 > > titikPutar = rotationTitikPutar
27 > > rotationTitikPutar = rotation(rotationTitikPutar,titikPusat,putaran)
28 > >
29 < func Kelopak1(titikPusat:Array, diagonal, besarKelopak, jumlahKelopak):
30 > pass
31 > var res = PoolVector2Array()
32 > var putaran = 360/jumlahKelopak
33 > var titikPutar = [titikPusat[0],titikPusat[1]-diagonal/2] # [50,25]
34 > var rotationTitikPutar = rotation(titikPutar,titikPusat,putaran) # []
35 > for i in jumlahKelopak:
36 > > res.append_array(lineDDA(titikPutar[0], titikPutar[1], titikPutar[0] - (diagonal/2), titikPutar[1] + (diagonal/2)))
37 > > res.append_array(lineDDA(titikPutar[0], titikPutar[1], titikPutar[0] + (diagonal/2), titikPutar[1] + (diagonal/2)))
38 > > res.append_array(lineDDA(titikPutar[0] - (diagonal/2), titikPutar[1] + (diagonal/2), titikPutar[0], titikPutar[1] + diagonal))
39 > > res.append_array(lineDDA(titikPutar[0] + (diagonal/2), titikPutar[1] + (diagonal/2), titikPutar[0], titikPutar[1] + diagonal))
40 > > titikPutar = rotationTitikPutar
41 > > rotationTitikPutar = rotation(rotationTitikPutar,titikPusat,putaran)
42 > >

```

Untuk kelopak 4

```
15 func _process(delta):
16     # pass
17     func _draw():
18         var ukuran = 35
19         for i in 5:
20             circleMidPoint(get_viewport().size.x/2, get_viewport().size.y/2, ukuran, Color(1,1,0,1))
21             ukuran = ukuran - 5
22         #
23         Kelopak([get_viewport().size.x/2, get_viewport().size.y/2],100, 50, 4)
24         Kelopak1([get_viewport().size.x/2, get_viewport().size.y/2 - 50],100, 10, 4)
25     #
26
27 func _on_Button3_pressed():
28     pass # Replace with function body.
29     get_tree().change_scene("res://TCSN/8 Kelopak.tscn")
30
31 func _on_Button4_pressed():
32     pass # Replace with function body.
33     get_tree().change_scene("res://TCSN/Main.tscn")
34
```

Hasil bunga kelopak 4

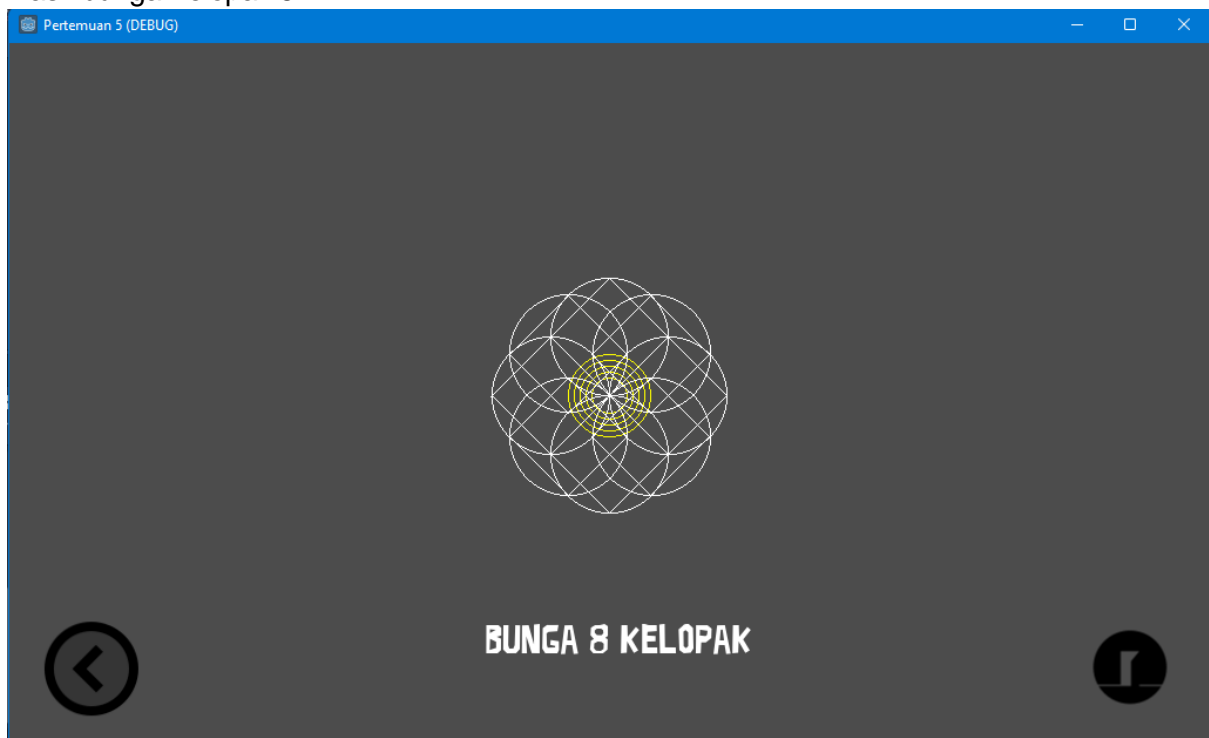




Untuk kelopak 8

```
16 # pass
17 func _draw():
18     var ukuran = 35
19     for i in 5 :
20         circleMidPoint(get_viewport().size.x/2, get_viewport().size.y/2, ukuran, Color(1,1,0,1))
21         ukuran = ukuran - 5
22     }
23     Kelopak([get_viewport().size.x/2, get_viewport().size.y/2], 100, 50, 8)
24     Kelopak1([get_viewport().size.x/2, get_viewport().size.y/2 - 50], 100, 10, 8)
25
26
27 func _on_Button3_pressed():
28     pass # Replace with function body.
29     get_tree().change_scene("res://TCSN/4 Kelopak.tscn")
30
31 func _on_Button_pressed():
32     pass # Replace with function body.
33     get_tree().change_scene("res://TCSN/Main.tscn")
34
```

Hasil bunga kelopak 8



## KOTRETAN

Disini yang perlu untuk dihitung adalah besaran sudut rotasi untuk bidang.

Untuk segi lima :

$$\text{Sudut} = 360 / 5 = 72$$

Untuk segi enam :

$$\text{Sudut} = 360 / 6 = 60$$

## LESSON LEARN

Yang saya pelajari antara lain adalah mengenai pengimplementasian matrix dalam godot. Disini digunakan array untuk pengimplementasiannya.

## CURHAT DOSEN

Untuk tugas godot ini terasa bingung karena kurang mengerti mengenai materi transformasi, sehingga dalam pengerjaanya pun memerlukan waktu lama untuk berfikir dan membuat pengerjaan ini terhambat.