

PERTEMUAN 3 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

DAFTAR ISI.....	2
DAFTAR REFERENSI.....	3
A. Pertanyaan 1	4
B. Pertanyaan 2	4
C. Pertanyaan 3	5
D. Pertanyaan 4	9
E. Pertanyaan 5	12
F. Pertanyaan 6	14
G. Pertanyaan 7	16
LESSON LEARN.....	20
CURHAT DOSEN	20

DAFTAR REFERENSI

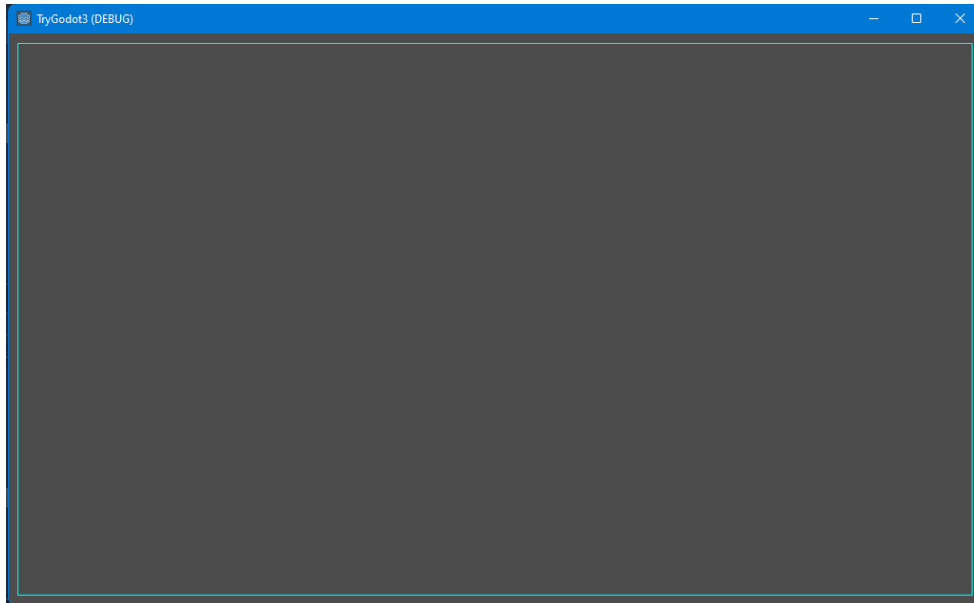
- <https://docs.godotengine.org/en/3.5/index.html>
- <https://github.com/godotengine/godot-demo-projects>
- <https://docs.godotengine.org/en/3.5/community/tutorials.html>
- https://www.youtube.com/watch?v=ZUPBoqC_X_o
- YouTube Channel : Kelas Terbuka – Godot Tutorial
- https://docs.godotengine.org/en/stable/tutorials/2d/custom_drawing_in_2d.html
- https://docs.godotengine.org/en/stable/tutorials/2d/custom_drawing_in_2d.html

A. Pertanyaan 1

Buatlah bingkai 10px

- Dapatkan koordinat pembangun bingkai (titik A, B, C, D) A ~ kiri atas, B kanan atas, C kiri bawah, dan D kanan bawah.
- Lalu disambungkan menggunakan algoritma pengembangan garis seperti lineDDA atau Bersenham.

Berikut adalah hasil output dari pertanyaan 1:



Berikut adalah scriptnya

```
#Saya membuat frame dengan margin 10
func frame():
    > #Garis horizontal
    > lineBres(10, 10, get_viewport().size.x-10, 10, warna)
    > lineBres(10, get_viewport().size.y-10, get_viewport().size.x-10, get_viewport().size.y-10, warna)
    > #Garis vertikal
    > lineDDA(10, 10, 10, get_viewport().size.y-10, warna )
    > lineDDA(get_viewport().size.x-10, 10, get_viewport().size.x-10, get_viewport().size.y-10, warna )
```

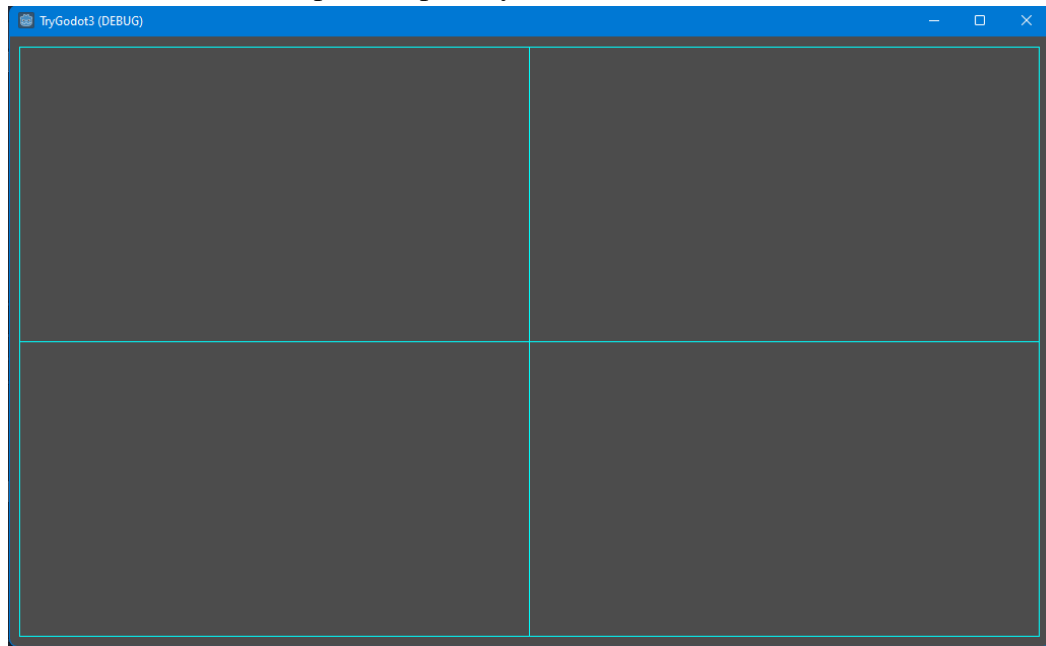
Disini digunakan algoritma bresenham dan juga untuk mengambil titik dari windows dengan menggunakan 'get_viewport'

B. Pertanyaan 2

Buatlah cartesian

- Dapatkan koordinat pembangun kartesian dengan mendapatkan titik tengah dari width dan height.
- Lalu disambungkan menggunakan algoritma pengembangan garis seperti lineDDA atau Bersenham untuk membangun kartesian.

Berikut adalah hasil output dari pertanyaan 2:



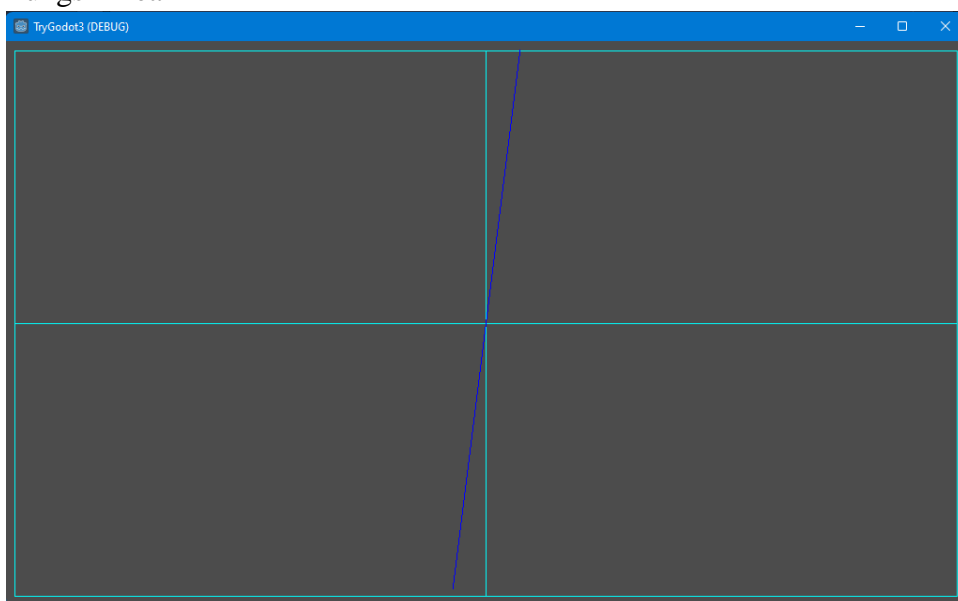
Berikut adalah scripnya

```
5 v func kartesian():  
6     lineBres(10, get_viewport().size.y/2, get_viewport().size.x-10, get_viewport().size.y/2, warna)  
7     lineDDA(get_viewport().size.x/2, 10, get_viewport().size.x/2, get_viewport().size.y-10, warna)
```

C. Pertanyaan 3

Berikut adalah hasil output dari pertanyaan 3:

a. Fungsi linear

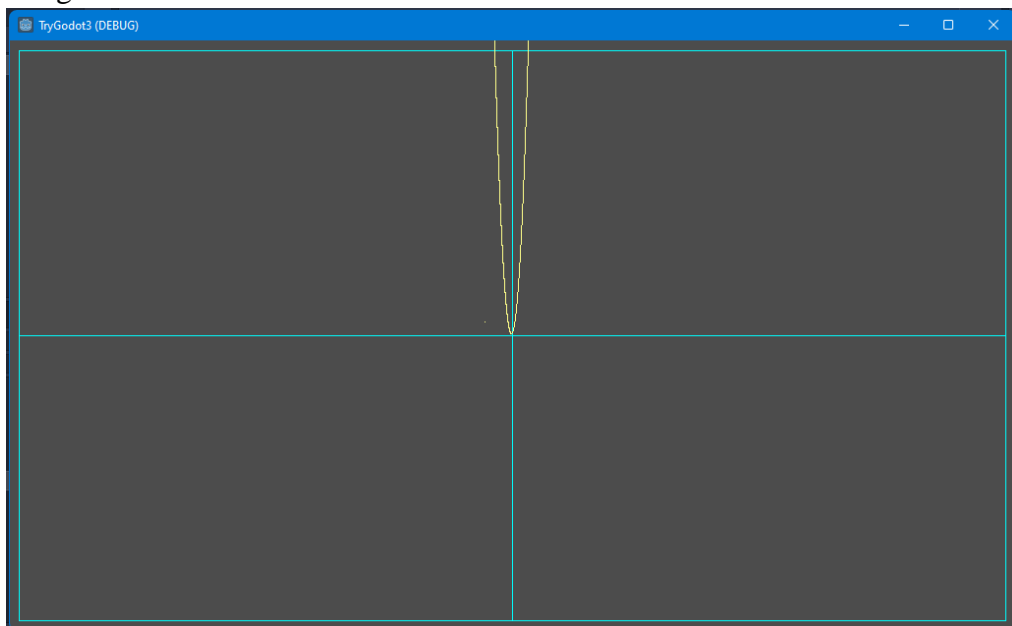


```

func fungsi_linear(x_koef, konstanta, color):
    >| var width = get_viewport().size.x
    >| var height = get_viewport().size.y
    >| # x = 0 // titik tengah
    >| var x = width/2
    >| # y = 0 // titik tengah
    >| var y = height/2
    >|
    >| var x_kebawah
    >| var y_kebawah
    >|
    >| #Selama di dalam frame
    >| while (x >= 10 && y < (height-10) && x < width-10 && y > 10):
    >|
    >| >| x += 0.01
    >| >| y = height/2 + (width/2 - x) * x_koef - konstanta
    >| >| put_pixel(x, y, color)
    >|
    >|
    >| >| x_kebawah = (width/2) - (x - width/2)
    >| >| y_kebawah = height/2 + (width/2 - x_kebawah) * x_koef - konstanta
    >| >| put_pixel(x_kebawah, y_kebawah, color)
    >|

```

b. Fungsi Kuadrat



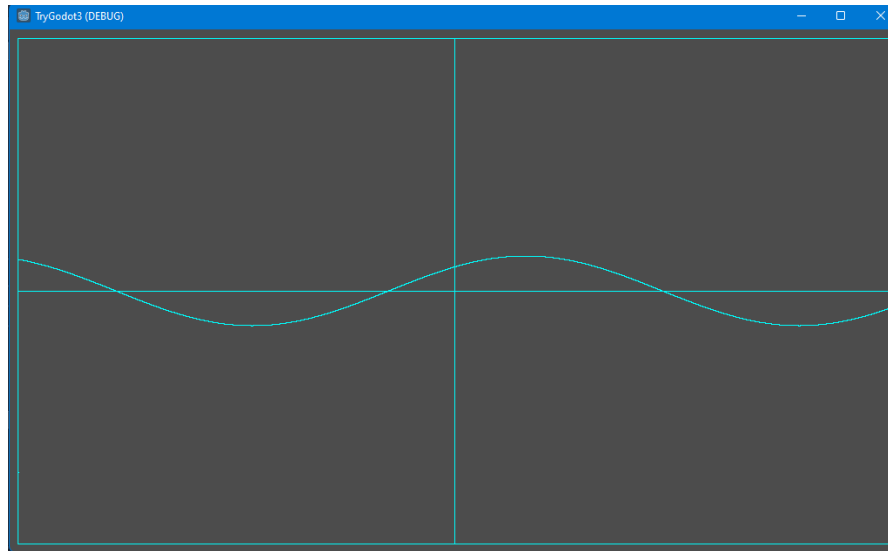
```

func fungsi_kuadrat(x_koef, konstanta, color):
    >| var width = get_viewport().size.x
    >| var height = get_viewport().size.y
    >| var y = pow(x_koef,2) + (3 * x_koef) + konstanta
    >|
    >| x_koef = x_koef - 30
    >|
    >| while (x_koef < 30):
    >| >| put_pixel(width/2 + x_koef, height/2 - y, color)
    >| >| x_koef+= 0.01
    >| >| y = pow(x_koef,2) + (3 * x_koef) + konstanta
    >|

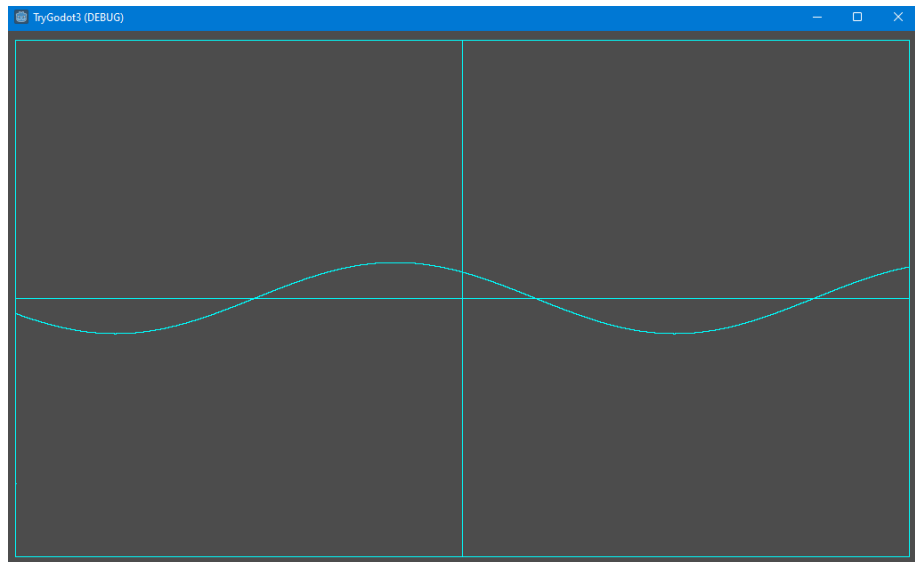
```

c. Fungsi Trigonometri

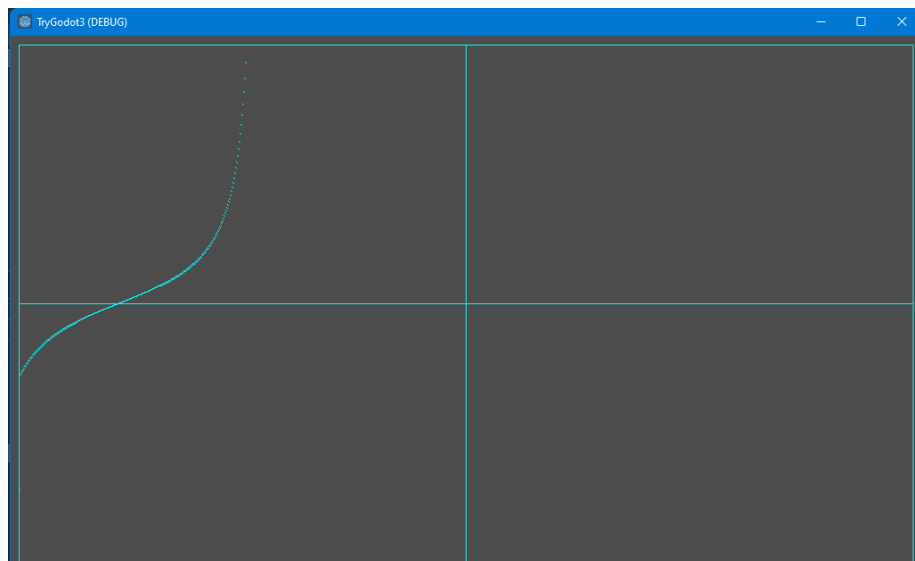
- Sin



- Cos



- Tan



```

✓ func fungsi_trigonometri(tipe ,x, colors):
  » var width = get_viewport().size.x
  » var height = get_viewport().size.y
  » var xa = 10
  » var ya = (width-10)/2
  » var y; var xb; var yb;
  »
✓ » while (xa >= 10 && ya < (height-10) && xa < width-10 && ya > 10):
✓ »   if tipe == "cos":
  »   »   » y = height/2 - (cos(x) * 40)
  »   »   » xb = xa + 1
  »   »   » yb = y
  »   »   » lineBres(xa,ya,xb,yb,colors)
  »   »   » x = x + 0.01
  »   »   » xa = xb
  »   »   » ya = yb
✓ »   elif tipe == "sin":
  »   »   » y = height/2 - (sin(x) * 40)
  »   »   » xb = xa + 1
  »   »   » yb = y
  »   »   » lineBres(xa,ya,xb,yb,colors)
  »   »   » x = x + 0.01
  »   »   » xa = xb
  »   »   » ya = yb
✓ »   elif tipe == "tan":
  »   »   » y = height/2 - (tan(x) * 40)
  »   »   » xb = xa + 1
  »   »   » yb = y
  »   »   » lineBres(xa,ya,xb,yb,colors)

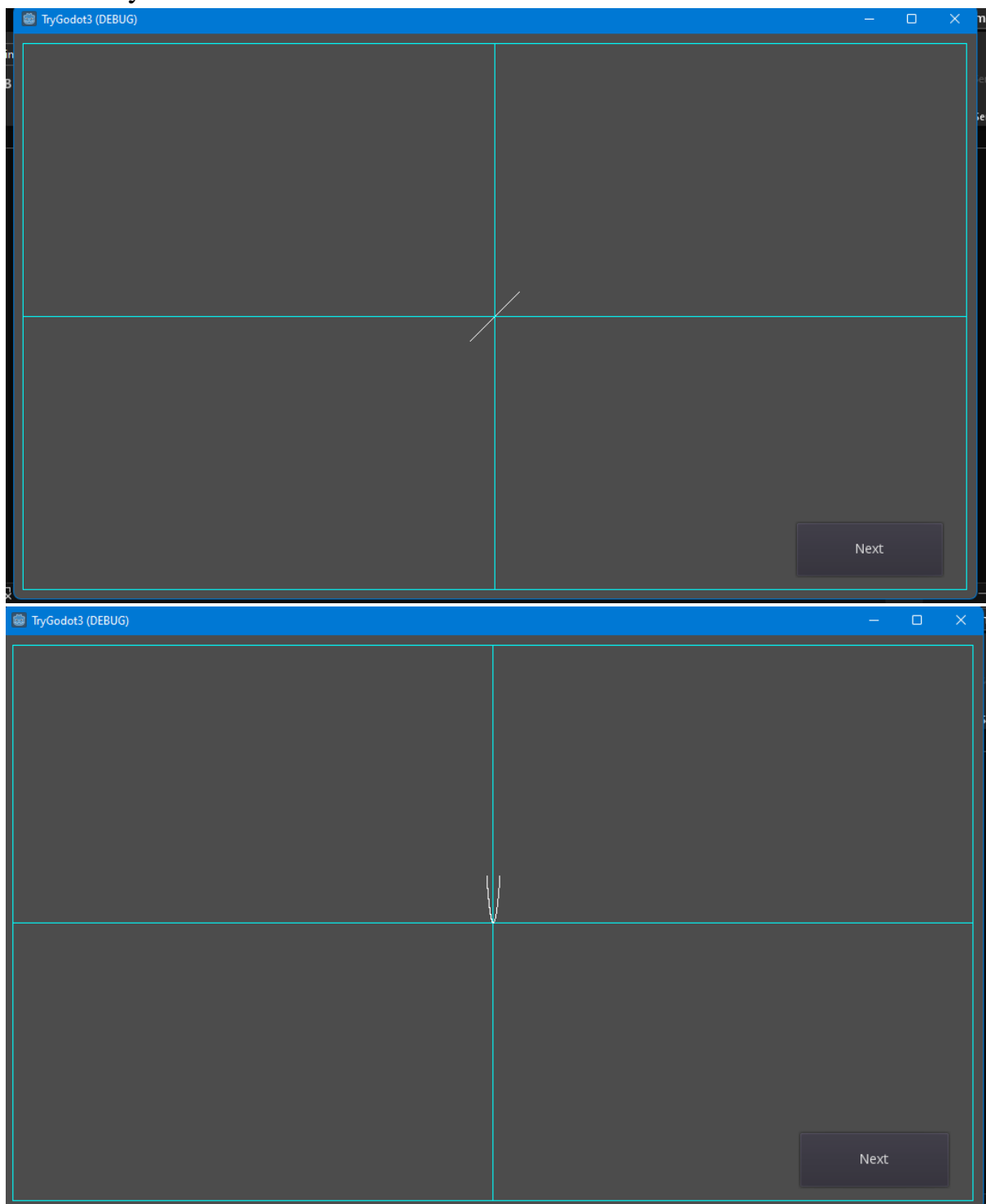
```

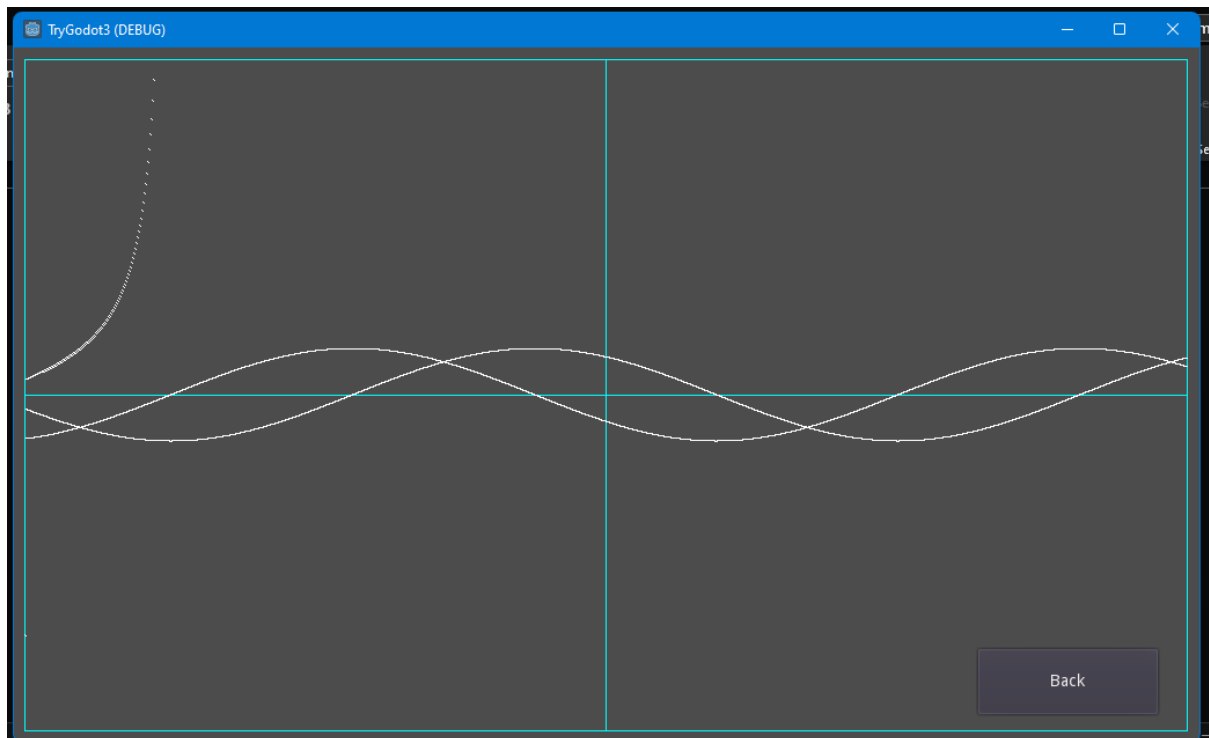
```

04 »   »   » x = x + 0.01
05 »   »   » xa = xb
06 »   »   » ya = yb
07

```


D. Pertanyaan 4





Untuk membuat animasi/garisnya bergerak caranya adalah bisa dengan menggunakan `process()` delta

Linear

```

1  extends "res://TCSN/line.gd"
2
3  # Declare member variables here. Examples:
4  # var a = 2
5  # var b = "text"
6  var limit = titikTengah.y
7  var limit2 = titikTengah.y
8
9  # Called when the node enters the scene tree for the first time.
10 func _ready():
11     pass # Replace with function body.
12 func _process(delta):
13     pass
14     limit = limit-1
15     if (titikTengah.y+limit == 24):
16         limit = 0
17         limit2 = limit2+1
18     if (titikTengah.y+limit2 == 24):
19         limit = 0
20
21
22     update()
23
24 func fungsi_linear(m,c):
25     pass
26
27     var xa # titikX = 0
28     var ya # titikY = 0
29     var xb
30     var yb
31
  
```

```

23
24 ~ func fungsi_linear(m,c):
25     pass
26
27     var xa # titikX = 0
28     var ya # titikY = 0
29     var xb
30     var yb
31
32     xa = titikTengah.x
33     ya = titikTengah.y-c
34 ~ while ((xa <= windowSizeX-24) && (xa >= 24) && (ya >= limit)): # steps = 20
35     xa += gradientFinderX(m)
36     ya -= gradientFinderY(m)
37     print ("titik xa ", xa, " titik ya ", ya)
38     put_pixel(xa,ya,Color(1,1,1))
39
40     xb = titikTengah.x
41     yb = titikTengah.y-c
42 ~ while ((xb <= windowSizeX-24) && (xb >= 24) && (yb <= limit2)): # steps = 20
43     xb -= gradientFinderX(m)+c
44     yb += gradientFinderY(m)+c
45     put_pixel(xb,yb,Color(1,1,1))
46
47 ~ func _draw():
48     frame()
49     kartesian()
50     fungsi_linear(1,0)
51

```

Kuadrat

```

1 extends "res://TCSM/line.gd"
2 var x = 0
3 var y = 0
4 var limit = 0
5
6 # Declare member variables here. Examples:
7 # var a = 2
8 # var b = "text"
9
10 ~ func _process(delta):
11     pass
12     limit = limit+1
13 ~ if (titikTengah.y-limit == 24):
14     limit = 0
15     update()
16
17
18 # Called when the node enters the scene tree for the first time.
19 ~ func _ready():
20     pass # Replace with function body.
21
22 ~ func fungsi_kuadrat(a,b,c):
23     var x = 0
24     var y = 0
25     pass
26     # ax^2 + bx + c
27     # a = gradien; b = variabel; c = konstanta
28     while (((x+10 <= windowSizeX-10) && (x+24 >= 10)) &&
29         (titikTengah.y-y >= titikTengah.y-limit)):
30         y = (a*x*x)-(b*x)+c
31         put_pixel(titikTengah.x-x,titikTengah.y-y,Color(1,1,1))
32
33         x = x+0.01
34

```

```

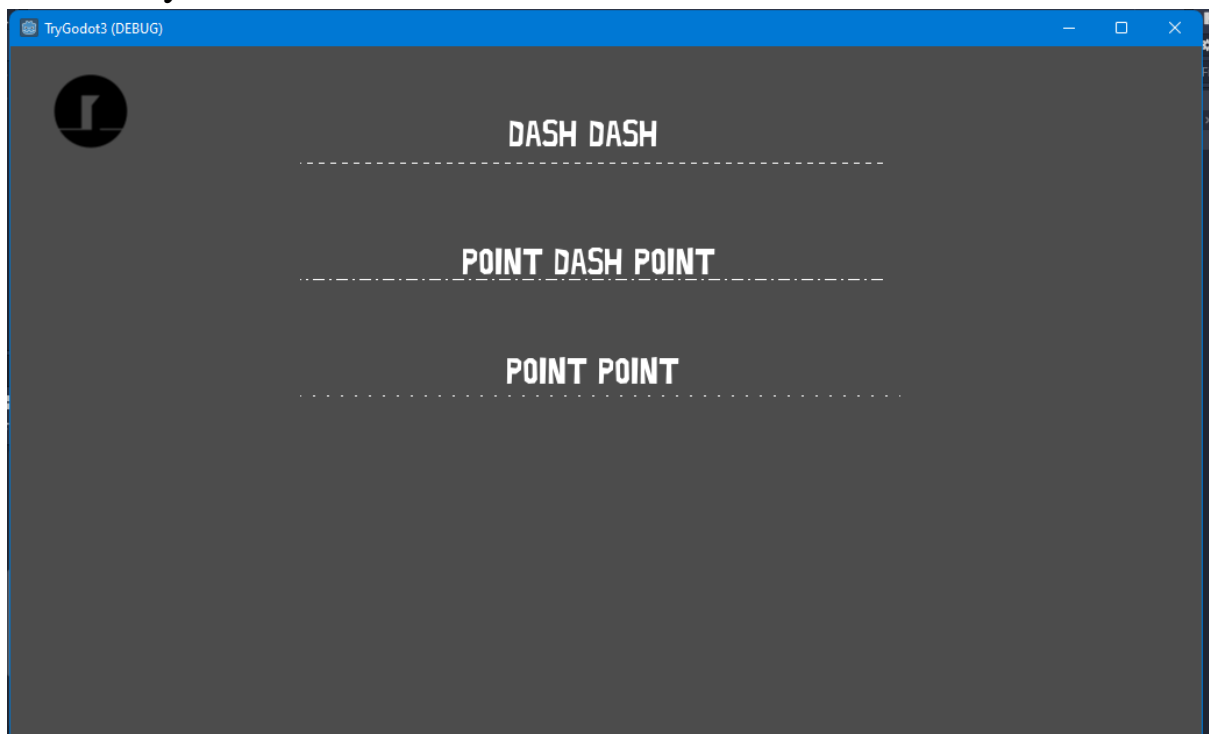
35     x = 0
36     y = 0
37     while (((x+10 <= windowSizeX-10) && (x+10 >= 10)) &&
38         (titikTengah.y-y >= titikTengah.y-limit)):
39
40         y = (a*x*x)-(b*x)+c
41         put_pixel(titikTengah.x+x,titikTengah.y-y,Color(1,1,1))
42         x = x+0.01
43
44 ~ func _draw():
45     pass
46     frame()
47     kartesian()
48     fungsi_kuadrat(1,0,0)
49
50 # Called every frame, 'delta' is the elapsed time since the previous frame.
51 #func _process(delta):
52     # pass
53
54
55 ~ func _on_Button_pressed():
56     get_tree().change_scene("res://TCSM/TrigonometriAnimasi.tscn")
57

```

Trigonometri

```
1 extends "res://TCSN/line.gd"
2
3
4 # Declare member variables here. Examples:
5 # var a = 2
6 # var b = "text"
7 var value = 0;
8
9 # Called when the node enters the scene tree for the first time.
10 func _ready():
11     pass # Replace with function body.
12
13
14 # Called every frame, 'delta' is the elapsed time since the previous frame.
15 func _process(delta):
16     pass
17
18 func _process(delta):
19     value -= 3 * delta;
20     update()
21
22 func _draw():
23     frame()
24     kartesian()
25     fungsi_trigonometri('sin', value, Color(1,1,1))
26     fungsi_trigonometri('cos', value, Color(1,1,1))
27     fungsi_trigonometri('tan', value, Color(1,1,1))
28
29
30 func _on_Button_pressed():
31     get_tree().change_scene('res://TCSN/line_menu.tscn')
32
```

E. Pertanyaan 5



Cara membuat variasi garis adalah dengan memodifikasi Increment titiknya

Dash Dash

```
22 ~ func Dash_dash(xa, ya, xb, yb, color):
23 ~     var dx = xb - xa
24 ~     var dy = yb - ya
25 ~     var steps
26 ~     var xIncrement
27 ~     var yIncrement
28 ~     var x = xa
29 ~     var y = ya
30 ~     var tampung
31 ~
32 ~     if (abs(dx) > abs(dy)) :
33 ~         steps = abs(dx)
34 ~     else :
35 ~         steps = abs(dy)
36 ~
37 ~     xIncrement = dx/ float(steps)
38 ~     yIncrement = dy/ float(steps)
39 ~     put_pixel(round(x), round(y), color)
40 ~
41 ~     for k in steps:
42 ~         if(k % 5 == 0):
43 ~             x += 5
44 ~             x += xIncrement
45 ~             y += yIncrement
46 ~             put_pixel(round(x), round(y), color)
47 ~
```

Point Dash Point

```
48 ~ func point_dash(xa, ya, xb, yb, color):
49 ~     var dx = xb - xa
50 ~     var dy = yb - ya
51 ~     var steps
52 ~     var xIncrement
53 ~     var yIncrement
54 ~     var x = xa
55 ~     var y = ya
56 ~     var tampung
57 ~
58 ~     if (abs(dx) > abs(dy)) :
59 ~         steps = abs(dx)
60 ~     else :
61 ~         steps = abs(dy)
62 ~
63 ~     xIncrement = round(dx/ float(steps))
64 ~     yIncrement = round(dy/ float(steps))
65 ~     put_pixel(round(x), round(y), color)
66 ~
67 ~     for k in steps:
68 ~         if(int(x) % 10 == 0):
69 ~             x += 4
70 ~             tampung = x
71 ~             #if(int(x) % 2 == 4):
72 ~                 #x+= 20
73 ~         if(x == tampung + 2):
74 ~             x += 4
75 ~             x += xIncrement
76 ~             y += yIncrement
77 ~             put_pixel(round(x), round(y), color)
78 ~
```

Point Point

```
func Point_point(xa, ya, xb, yb, color):
    var dx = xb - xa
    var dy = yb - ya
    var steps
    var xIncrement
    var yIncrement
    var x = xa
    var y = ya
    var tampung
    if (abs(dx) > abs(dy)) :
        steps = abs(dx)
    else :
        steps = abs(dy)
    xIncrement = dx / float(steps)
    yIncrement = dy / float(steps)
    put_pixel(round(x), round(y), color)
    for k in steps:
        if(k % 2 == 0):
            x += 10
            x += xIncrement
            y += yIncrement
            put_pixel(round(x), round(y), color)
```

F. Pertanyaan 6



Dibuat dengan menggunakan Vector2 yang mana titiknya tersebut disimpan dalam sebuah array dulu.

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

func put_pixel_all(dot: PoolVector2Array, color):
    for i in dot.size():
        put_pixel(dot[i].x, dot[i].y, color)
```

```

9 func lineDDA(xa : float, ya : float, xb : float, yb : float):
10     var dx = xb - xa
11     var dy = yb - ya
12     var steps
13     var xIncrement
14     var yIncrement
15     var x = xa
16     var y = ya
17     var res = PoolVector2Array()
18
19     if (abs(dx) > abs(dy)) :
20         steps = abs(dx)
21     else :
22         steps = abs(dy)
23
24     xIncrement = dx/ float(steps)
25     yIncrement = dy/ float(steps)
26     res.append(Vector2(round(x), round(y)))
27
28     for k in steps:
29         x += xIncrement
30         y += yIncrement
31         res.append(Vector2(round(x), round(y)))
32
33     #Dipanggil agar garis tampil
34     put_pixel_all(res, color)
35     return res
36

```

```

1 extends 'res://TCSD/primitif_array.gd'
2
3
4 # Declare member variables here. Examples:
5 # var a = 2
6 # var b = "text"
7
8 # Called when the node enters the scene tree for the first time.
9 func _ready():
10     pass # Replace with function body.
11
12 func persegi(titik_awal: Vector2, panjang_sisi):
13     var res = PoolVector2Array()
14     res.append_array(lineDDA(titik_awal.x, titik_awal.y, titik_awal.x + panjang_sisi, titik_awal.y)) #sisi atas
15     res.append_array(lineDDA(titik_awal.x, titik_awal.y + panjang_sisi, titik_awal.x + panjang_sisi, titik_awal.y + panjang_sisi)) #sisi bawah
16     res.append_array(lineDDA(titik_awal.x, titik_awal.y, titik_awal.x, titik_awal.y + panjang_sisi)) #sisi kiri
17     res.append_array(lineDDA(titik_awal.x + panjang_sisi, titik_awal.y, titik_awal.x + panjang_sisi, titik_awal.y + panjang_sisi)) #sisi kanan
18     return res
19
20 func persegi_panjang(titik_awal: Vector2, panjang, lebar):
21     var res = PoolVector2Array()
22     res.append_array(lineDDA(titik_awal.x, titik_awal.y, titik_awal.x + panjang, titik_awal.y)) #sisi atas
23     res.append_array(lineDDA(titik_awal.x, titik_awal.y + lebar, titik_awal.x + panjang, titik_awal.y + lebar)) #sisi bawah
24     res.append_array(lineDDA(titik_awal.x, titik_awal.y, titik_awal.x, titik_awal.y + lebar)) #sisi kiri
25     res.append_array(lineDDA(titik_awal.x + panjang, titik_awal.y, titik_awal.x + panjang, titik_awal.y + lebar)) #sisi kanan
26     return res
27
28 func segitiga_siku_siku(titik_awal: Vector2, alas, tinggi):
29     var res = PoolVector2Array()
30     res.append_array(lineDDA(titik_awal.x, titik_awal.y, titik_awal.x, titik_awal.y + tinggi)) #sisi kiri
31     res.append_array(lineDDA(titik_awal.x, titik_awal.y + tinggi, titik_awal.x + alas, titik_awal.y + tinggi)) #sisi bawah
32     res.append_array(lineDDA(titik_awal.x, titik_awal.y, titik_awal.x + alas, titik_awal.y + tinggi)) #sisi kanan
33     return res
34

```

```

36 func trapesium_siku_siku(titik_awal: Vector2, atas, tinggi, bawah):
37     var res = PoolVector2Array()
38     res.append_array(lineDDA(titik_awal.x, titik_awal.y, titik_awal.x + atas, titik_awal.y)) #sisi atas
39     res.append_array(lineDDA(titik_awal.x, titik_awal.y + tinggi, titik_awal.x + bawah, titik_awal.y + tinggi)) #sisi bawah
40     res.append_array(lineDDA(titik_awal.x, titik_awal.y, titik_awal.x, titik_awal.y + tinggi)) #sisi kiri
41     res.append_array(lineDDA(titik_awal.x + atas, titik_awal.y, titik_awal.x + bawah, titik_awal.y + tinggi)) #sisi kanan
42     return res
43
44 func trapesium_sama_kaki(titik_awal: Vector2, atas, tinggi, bawah):
45     var res = PoolVector2Array()
46     res.append_array(lineDDA(titik_awal.x, titik_awal.y, titik_awal.x + atas, titik_awal.y)) #sisi atas
47     res.append_array(lineDDA(titik_awal.x - ((bawah - atas)/2), titik_awal.y + tinggi, (titik_awal.x - ((bawah - atas)/2)) + bawah, titik_awal.y + tinggi)) #sisi bawah
48     res.append_array(lineDDA(titik_awal.x, titik_awal.y, (titik_awal.x - ((bawah - atas)/2)), titik_awal.y + tinggi)) #sisi kiri
49     res.append_array(lineDDA(titik_awal.x + atas, titik_awal.y, (titik_awal.x - ((bawah - atas)/2)) + bawah, titik_awal.y + tinggi)) #sisi kanan
50     return res
51
52 func jajargenjang(titik_awal: Vector2, panjang, tinggi, geser):
53     var res = PoolVector2Array()
54     res.append_array(lineDDA(titik_awal.x, titik_awal.y, titik_awal.x + panjang, titik_awal.y)) #sisi atas
55     res.append_array(lineDDA(titik_awal.x - geser, titik_awal.y + tinggi, (titik_awal.x + panjang - geser), titik_awal.y + tinggi)) #sisi bawah
56     res.append_array(lineDDA(titik_awal.x, titik_awal.y, (titik_awal.x - geser), titik_awal.y + tinggi)) #sisi kiri
57     res.append_array(lineDDA(titik_awal.x + panjang, titik_awal.y, (titik_awal.x + panjang - geser), titik_awal.y + tinggi)) #sisi kanan
58     return res
59

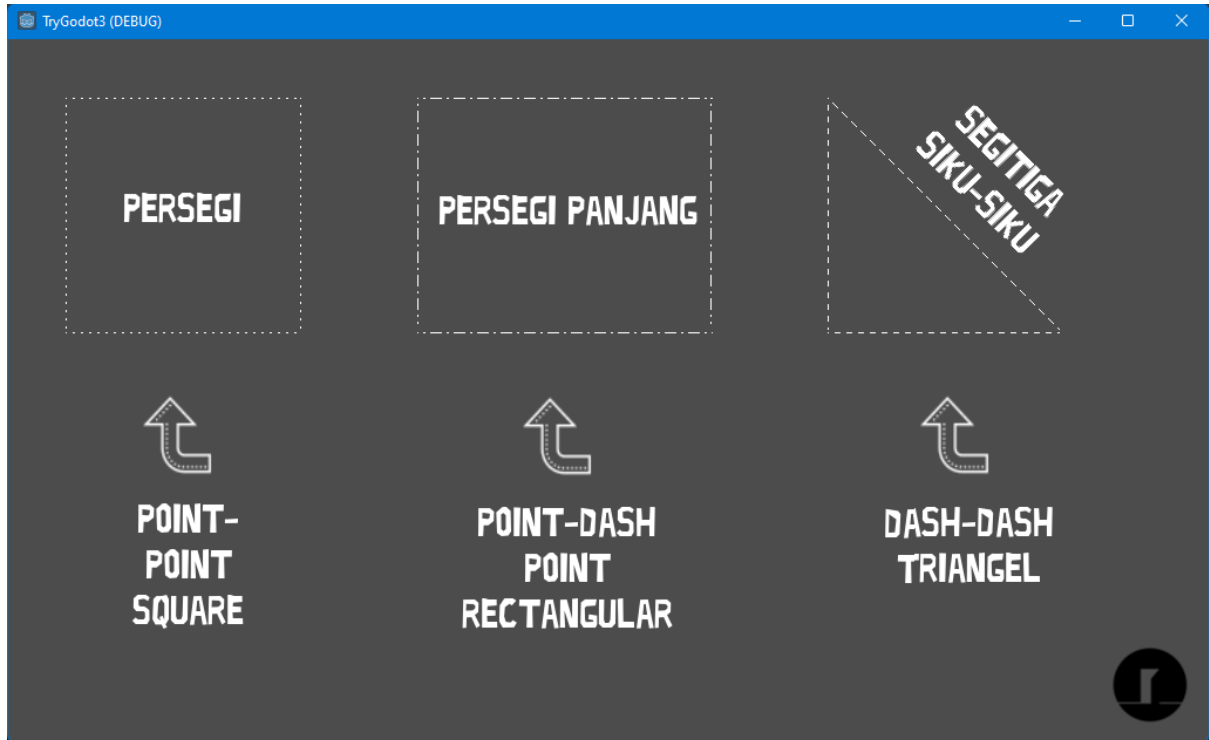
```

```

64 func _draw():
65     >| perseg(Vector2(50,50), 200)
66     >| perseg_panjang(Vector2(350,50),250 , 200)
67     >| segitiga_siku_siku(Vector2(700,50), 200, 200)
68     >| trapesium_siku_siku(Vector2(50,350), 200, 150, 250)
69     >| trapesium_sama_kaki(Vector2(375,350), 200, 150, 300)
70     >| jajargenjang(Vector2(700,350), 200, 150, 50)
71

```

G. Pertanyaan 7



Hasil penerapan antara pertanyaan 5 dan 6 dengan membentuk bidang menggunakan algoritma garis yang telah dimodifikasi

```

17 func put_pixel(x, y, color):
18     >| draw_primitive(PoolVector2Array([Vector2(x, y)]),
19     >| PoolColorArray([color]),
20     >| PoolVector2Array())
21
22 func put_pixel_all(dot: PoolVector2Array, color):
23     >| for i in dot.size():
24     >|     put_pixel(dot[i].x, dot[i].y, color)
25

```



```

27 #line_dda
28 func dash_dash(xa : float, ya : float, xb : float, yb : float):
29     var dx = xb - xa
30     var dy = yb - ya
31     var steps
32     var xIncrement
33     var yIncrement
34     var x = xa
35     var y = ya
36     var res = PoolVector2Array()
37
38
39     if (abs(dx) > abs(dy)) :
40         steps = abs(dx)
41     else :
42         steps = abs(dy)

```

```

43
44     xIncrement = dx/ float(steps)
45     yIncrement = dy/ float(steps)
46     res.append(Vector2(round(x), round(y)))
47     if(ya == yb):
48         for k in steps :
49             if(k % 5 == 0):
50                 x += 5
51                 x += xIncrement
52                 y += yIncrement
53                 res.append(Vector2(round(x), round(y)))
54             if(k > 95):
55                 break
56             print(k)
57     elif(xa == xb):
58         for k in steps :
59             if(k % 5 == 0):
60                 y += 5
61                 y += yIncrement
62                 res.append(Vector2(round(x), round(y)))
63             if(k > 95):
64                 break
65             print(k)

```

```

66     else:
67         for k in steps:
68             if(k % 5 == 0):
69                 x += 3
70                 y += 3
71                 x += xIncrement
72                 y += yIncrement
73                 put_pixel(round(x), round(y), color)
74             if(k > 120):
75                 break
76             print(steps)
77
78     #Dipanggil agar garis tampil
79     put_pixel_all(res, color)
80     return res

```

```
82 ▾ func point_dash(xa : float, ya : float, xb : float, yb : float):
```

```
83   ▸ var dx = xb - xa
```

```
84   ▸ var dy = yb - ya
```

```
85   ▸ var steps
```

```
86   ▸ var xIncrement
```

```
87   ▸ var yIncrement
```

```
88   ▸ var x = xa
```

```
89   ▸ var y = ya
```

```
90   ▸ var res = PoolVector2Array()
```

```
91   ▸ var tampung = 0
```

```
92   ▸
```

```
93 ▾ ▸ if (abs(dx) > abs(dy)) :
```

```
94   ▸   ▸ steps = abs(dx)
```

```
95 ▾ ▸ else :
```

```
96   ▸   ▸ steps = abs(dy)
```

```
97   ▸
```

```
98   ▸ xIncrement = dx/ float(steps)
```

```
99   ▸ yIncrement = dy/ float(steps)
```

```
100  ▸ res.append(Vector2(round(x), round(y)))
```

```
101 ▾ ▸ if(ya == yb):
```

```
102 ▾ ▸   ▸ for k in steps:
```

```
103 ▾ ▸     ▸   ▸ if(int(x) % 10 == 0):
```

```
104   ▸     ▸     ▸   ▸ x+= 4
```

```
105   ▸     ▸     ▸   ▸ tampung = x
```

```
106 ▾ ▸     ▸   ▸ if(x == tampung + 2):
```

```
107   ▸     ▸     ▸   ▸ x+= 4
```

```
108   ▸     ▸   ▸ x += xIncrement
```

```
109   ▸     ▸   ▸ y += yIncrement
```

```
110   ▸     ▸   ▸ res.append(Vector2(round(x), round(y)))
```

```
111 ▾ ▸     ▸   ▸ if(k > 145):
```

```
112   ▸     ▸     ▸   ▸ break
```

```
113 ▾ ▸   ▸ else :
```

```
114 ▾ ▸     ▸   ▸ for k in steps:
```

```
115 ▾ ▸       ▸   ▸ if(int(y) % 10 == 0):
```

```
116   ▸       ▸     ▸   ▸ y+= 4
```

```
117   ▸       ▸     ▸   ▸ tampung = y
```

```
118 ▾ ▸       ▸   ▸ if(y == tampung + 2):
```

```
119   ▸       ▸     ▸   ▸ y+= 4
```

```
120   ▸       ▸   ▸ y += yIncrement
```

```
121   ▸       ▸   ▸ res.append(Vector2(round(x), round(y)))
```

```
122 ▾ ▸       ▸   ▸ if(k > 110):
```

```
123   ▸       ▸     ▸   ▸ break
```

```
124   ▸ #Dipanggil agar garis tampil
```

```
125   ▸ put_pixel_all(res, color)
```

```
126   ▸ return res
```

```
func point_point(xa : float, ya : float, xb : float, yb : float):
```

```
▸ var dx = xb - xa
```

```
▸ var dy = yb - ya
```

```
▸ var steps
```

```
▸ var xIncrement
```

```
▸ var yIncrement
```

```
▸ var x = xa
```

```
▸ var y = ya
```

```
▸ var res = PoolVector2Array()
```

```
▸
```

```
▸ if (abs(dx) > abs(dy)) :
```

```
▸   ▸ steps = abs(dx)
```

```
▸ else :
```

```
▸   ▸ steps = abs(dy)
```

```
▸
```

```

xIncrement = dx/ float(steps)
yIncrement = dy/ float(steps)
res.append(Vector2(round(x), round(y)))

if(ya == yb):
    for k in steps :
        if(k % 2 == 0):
            x += 5
            x += xIncrement
            y += yIncrement
            res.append(Vector2(round(x), round(y)))
            if(k > 54):
                break
            print(k)
elif(xa == xb):
    for k in steps :
        if(k % 2 == 0):
            y += 5
            y += yIncrement
            res.append(Vector2(round(x), round(y)))
            if(k > 54):
                break
            print(k)

#Dipanggil agar garis tampil
put_pixel_all(res, color)
return res

```

```

1 ~ func persegi(titik_awal: Vector2, panjang_sisi):
2   var res = PoolVector2Array()
3   point_point(titik_awal.x, titik_awal.y, titik_awal.x + panjang_sisi, titik_awal.y) #sisi atas
4   point_point(titik_awal.x, titik_awal.y + panjang_sisi, titik_awal.x + panjang_sisi, titik_awal.y + panjang_sisi) #sisi bawah
5   point_point(titik_awal.x, titik_awal.y, titik_awal.x, titik_awal.y + panjang_sisi) #sisi kiri
6   point_point(titik_awal.x + panjang_sisi, titik_awal.y, titik_awal.x + panjang_sisi, titik_awal.y + panjang_sisi) #sisi kanan
7   return res
8
9 ~ func persegi_panjang(titik_awal: Vector2, panjang, lebar):
10  var res = PoolVector2Array()
11  point_dash(titik_awal.x, titik_awal.y, titik_awal.x + panjang, titik_awal.y) #sisi atas
12  point_dash(titik_awal.x, titik_awal.y + lebar, titik_awal.x + panjang, titik_awal.y + lebar) #sisi bawah
13  point_dash(titik_awal.x, titik_awal.y, titik_awal.x, titik_awal.y + lebar) #sisi kiri
14  point_dash(titik_awal.x + panjang, titik_awal.y, titik_awal.x + panjang, titik_awal.y + lebar) #sisi kanan
15  return res
16
17 ~ func segitiga_siku_siku(titik_awal: Vector2, alas, tinggi):
18  var res = PoolVector2Array()
19  dash_dash(titik_awal.x, titik_awal.y, titik_awal.x, titik_awal.y + tinggi) #sisi kiri
20  dash_dash(titik_awal.x, titik_awal.y + tinggi, titik_awal.x + alas, titik_awal.y + tinggi) #sisi bawah
21  dash_dash(titik_awal.x, titik_awal.y, titik_awal.x + alas, titik_awal.y + tinggi) #sisi kanan
22  return res
23

```

```

1 ~ func _draw():
2   persegi(Vector2(50,50), 200)
3   persegi_panjang(Vector2(350,50),250 , 200)
4   segitiga_siku_siku(Vector2(700,50), 200, 200)
5

```

```

1 ~ func _ready():
2   $CanvasItem.draw()
3   $CanvasItem.draw()
4   $CanvasItem.draw()
5   $CanvasItem.draw()
6   $CanvasItem.draw()
7   $CanvasItem.draw()
8   $CanvasItem.draw()
9   $CanvasItem.draw()
10  $CanvasItem.draw()
11  $CanvasItem.draw()
12  $CanvasItem.draw()
13  $CanvasItem.draw()
14  $CanvasItem.draw()
15  $CanvasItem.draw()
16  $CanvasItem.draw()
17  $CanvasItem.draw()
18  $CanvasItem.draw()
19  $CanvasItem.draw()
20  $CanvasItem.draw()
21  $CanvasItem.draw()
22  $CanvasItem.draw()
23  $CanvasItem.draw()
24  $CanvasItem.draw()
25  $CanvasItem.draw()
26  $CanvasItem.draw()
27  $CanvasItem.draw()
28  $CanvasItem.draw()
29  $CanvasItem.draw()
30  $CanvasItem.draw()
31  $CanvasItem.draw()
32  $CanvasItem.draw()
33  $CanvasItem.draw()
34  $CanvasItem.draw()
35  $CanvasItem.draw()
36  $CanvasItem.draw()
37  $CanvasItem.draw()
38  $CanvasItem.draw()
39  $CanvasItem.draw()
40  $CanvasItem.draw()
41  $CanvasItem.draw()
42  $CanvasItem.draw()
43  $CanvasItem.draw()
44  $CanvasItem.draw()
45  $CanvasItem.draw()
46  $CanvasItem.draw()
47  $CanvasItem.draw()
48  $CanvasItem.draw()
49  $CanvasItem.draw()
50  $CanvasItem.draw()
51  $CanvasItem.draw()
52  $CanvasItem.draw()
53  $CanvasItem.draw()
54  $CanvasItem.draw()
55  $CanvasItem.draw()
56  $CanvasItem.draw()
57  $CanvasItem.draw()
58  $CanvasItem.draw()
59  $CanvasItem.draw()
60  $CanvasItem.draw()
61  $CanvasItem.draw()
62  $CanvasItem.draw()
63  $CanvasItem.draw()
64  $CanvasItem.draw()
65  $CanvasItem.draw()
66  $CanvasItem.draw()
67  $CanvasItem.draw()
68  $CanvasItem.draw()
69  $CanvasItem.draw()
70  $CanvasItem.draw()
71  $CanvasItem.draw()
72  $CanvasItem.draw()
73  $CanvasItem.draw()
74  $CanvasItem.draw()
75  $CanvasItem.draw()
76  $CanvasItem.draw()
77  $CanvasItem.draw()
78  $CanvasItem.draw()
79  $CanvasItem.draw()
80  $CanvasItem.draw()
81  $CanvasItem.draw()
82  $CanvasItem.draw()
83  $CanvasItem.draw()
84  $CanvasItem.draw()
85  $CanvasItem.draw()
86  $CanvasItem.draw()
87  $CanvasItem.draw()
88  $CanvasItem.draw()
89  $CanvasItem.draw()
90  $CanvasItem.draw()
91  $CanvasItem.draw()
92  $CanvasItem.draw()
93  $CanvasItem.draw()
94  $CanvasItem.draw()
95  $CanvasItem.draw()
96  $CanvasItem.draw()
97  $CanvasItem.draw()
98  $CanvasItem.draw()
99  $CanvasItem.draw()
100 $CanvasItem.draw()

```

LESSON LEARN

Setelah saya mengerjakan tugas ini saya mendapat banyak pelajaran seperti cara memodifikasi garis di dalam godot engine, dan menggunakan array dalam menyimpan nilai koordinat dari titik untuk membuat sebuah garis.

Hal yang saya pelajari juga adalah dalam menuliskan baris kode, harus efektif, karena pengalaman saya dapatkan berpengaruh pada saat merunning programnya jadi lebih lambat

Hasil temuan library : Vector2, append

CURHAT DOSEN

Ketika saya mengerjakan projek godot ini, seringkali not responding laptopnya dan ternyata karena pengaruh dari kodenya yang tidak efektif. Untuk tugas ini tidak terlupakan seperti yang minggu kemarin karena kebetulan tugas lainnya yang memiliki deadline dekat sudah saya kerjakan.