

**PERTEMUAN 2 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. Pertanyaan 1 .....</b>	<b>4</b>
<b>B. Pertanyaan 2 .....</b>	<b>4</b>
<b>C. Pertanyaan 3 .....</b>	<b>5</b>
<b>D. Pertanyaan 4 .....</b>	<b>9</b>
<b>E. Pertanyaan 5 .....</b>	<b>12</b>
<b>LESSON LEARN.....</b>	<b>14</b>

## **DAFTAR REFERENSI**

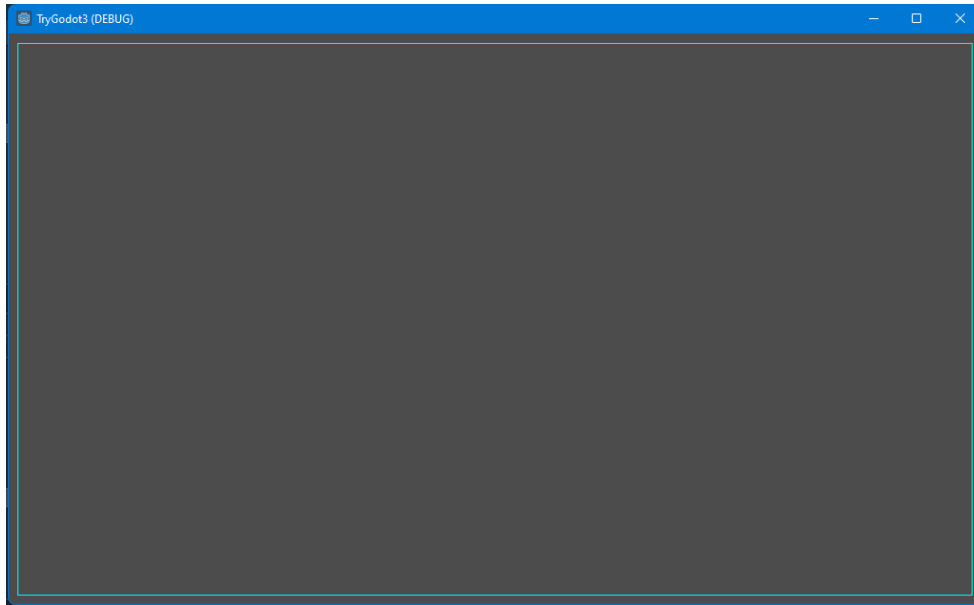
- <https://docs.godotengine.org/en/3.5/index.html>
- <https://github.com/godotengine/godot-demo-projects>
- <https://docs.godotengine.org/en/3.5/community/tutorials.html>
- [https://www.youtube.com/watch?v=ZUPBoqC\\_X\\_o](https://www.youtube.com/watch?v=ZUPBoqC_X_o)
- YouTube Channel : Kelas Terbuka – Godot Tutorial

## 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 )
```

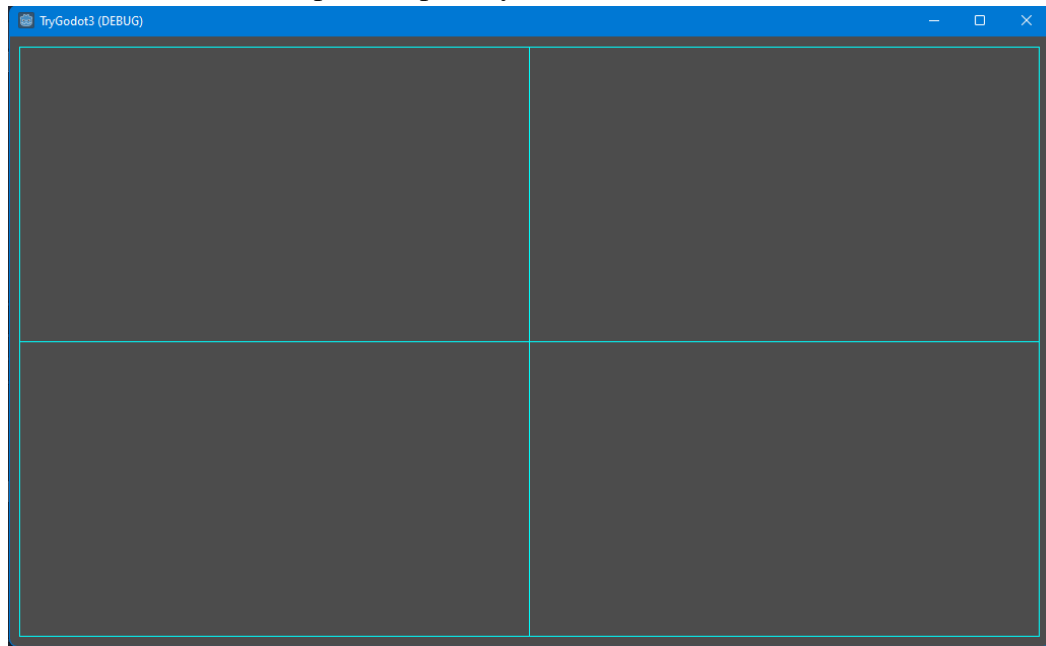
Disini digunakan algoritma bresenham dan juga untuk mengambil titik dari windows dengan menggunakan 'get\_viewport'

## 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:



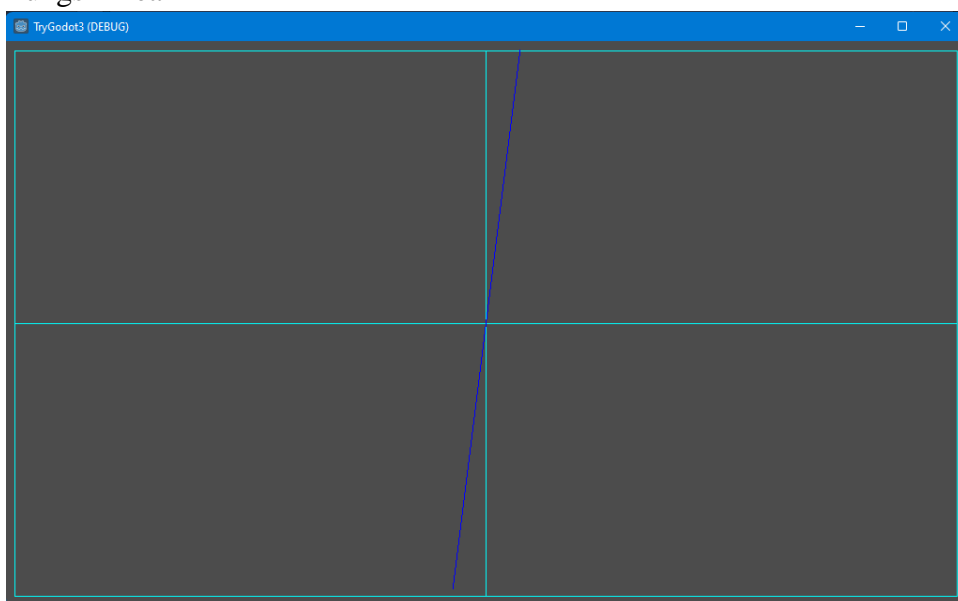
Berikut adalah scripnya

```
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



```

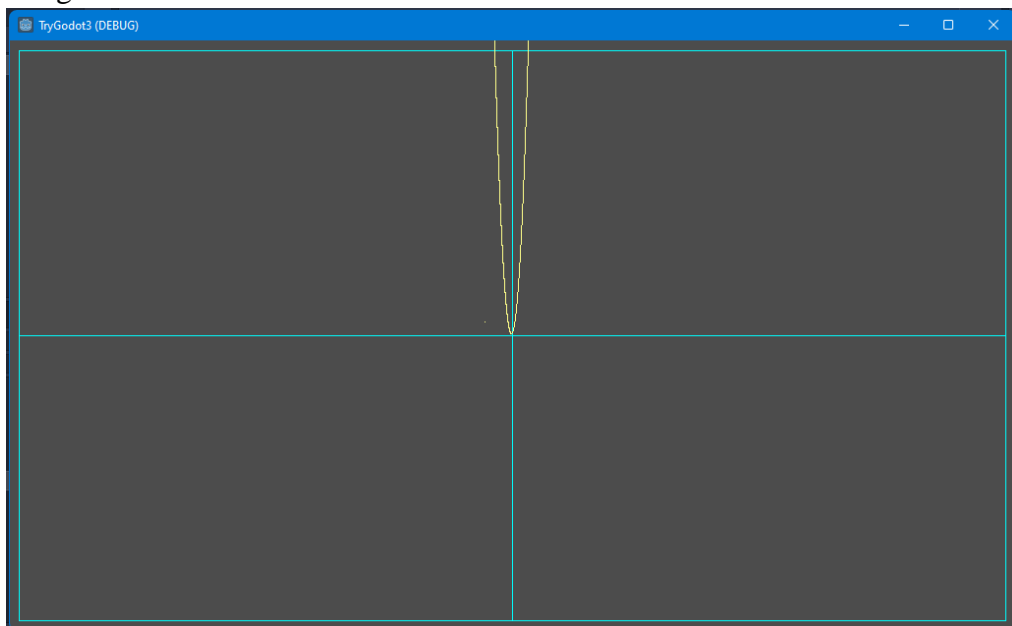
func fungsi_linear(x_koef, konstanta, color):
    >| var width = get_viewport().size.x
    >| var height = get_viewport().size.y
    >| # x = 0 // titik tengah
    >| var x = width/2
    >| # y = 0 // titik tengah
    >| var y = height/2
    >|
    >| var x_kebawah
    >| var y_kebawah

    >| #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



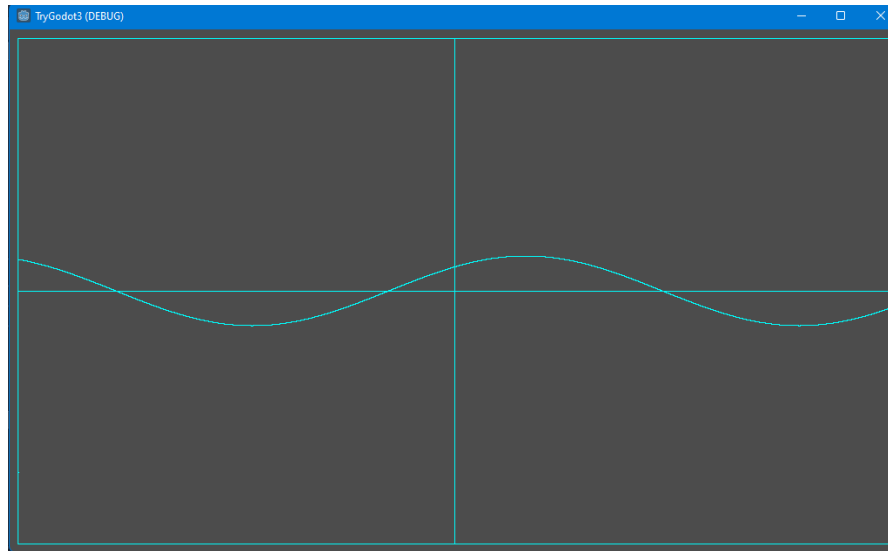
```

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
    >|

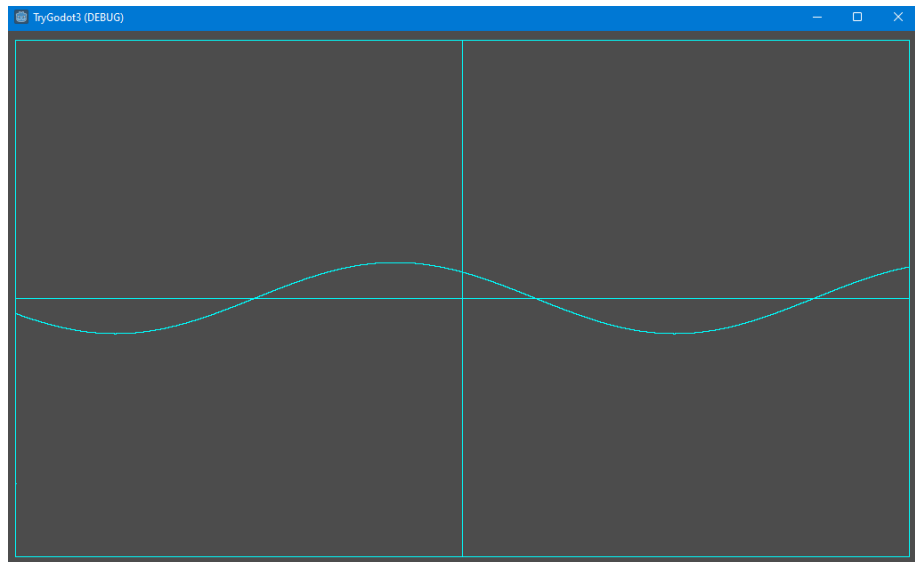
```

### c. Fungsi Trigonometri

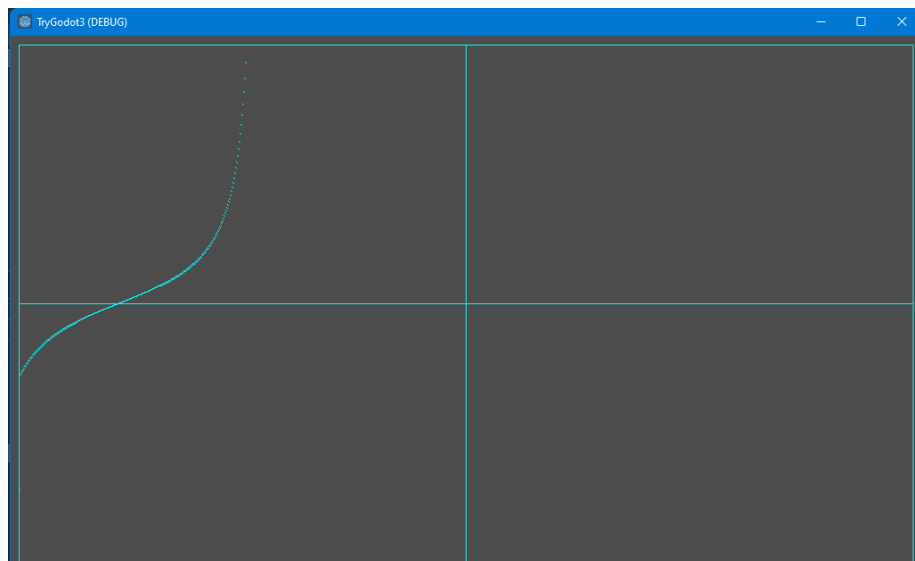
#### - Sin



#### - 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)

```

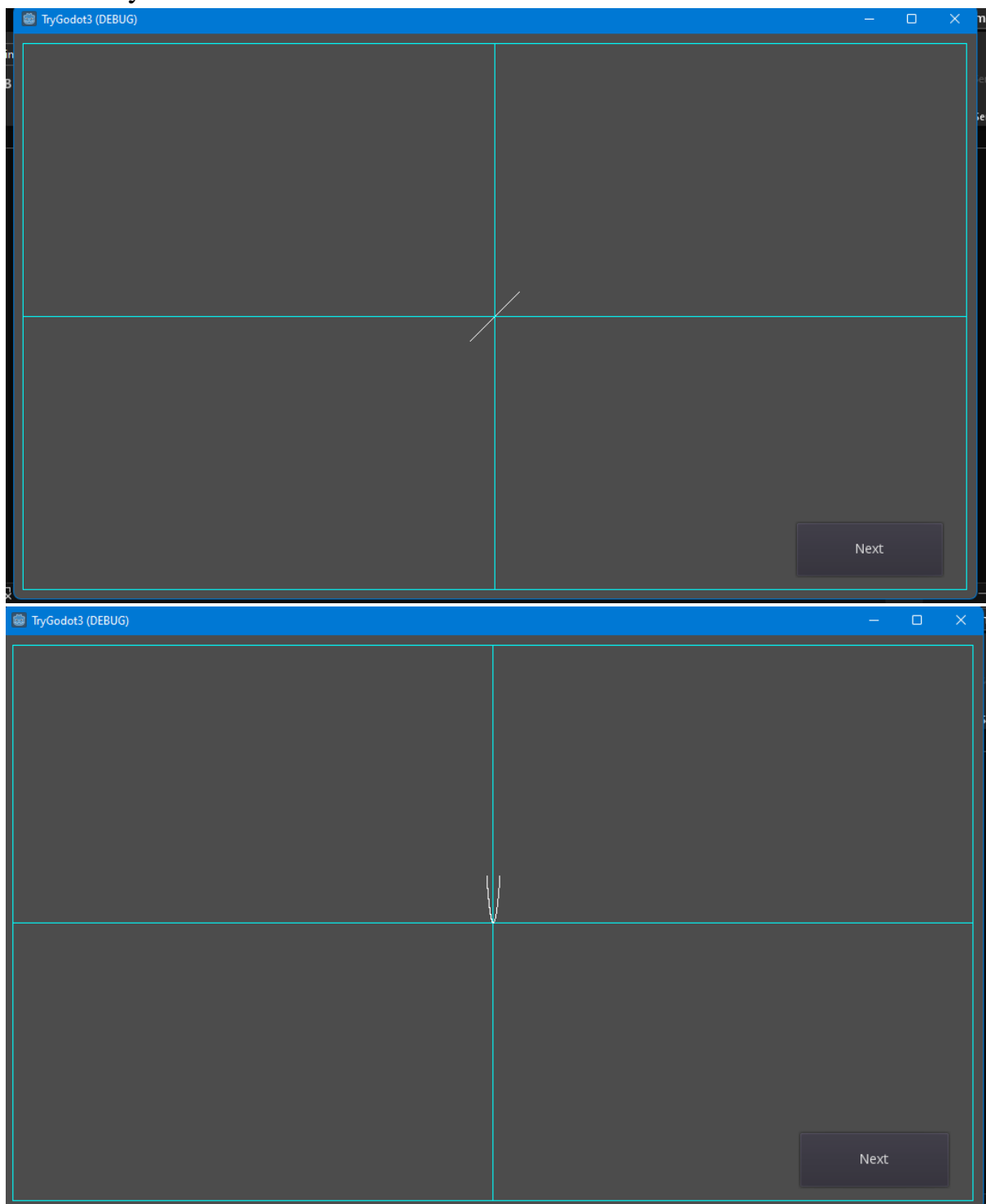
```

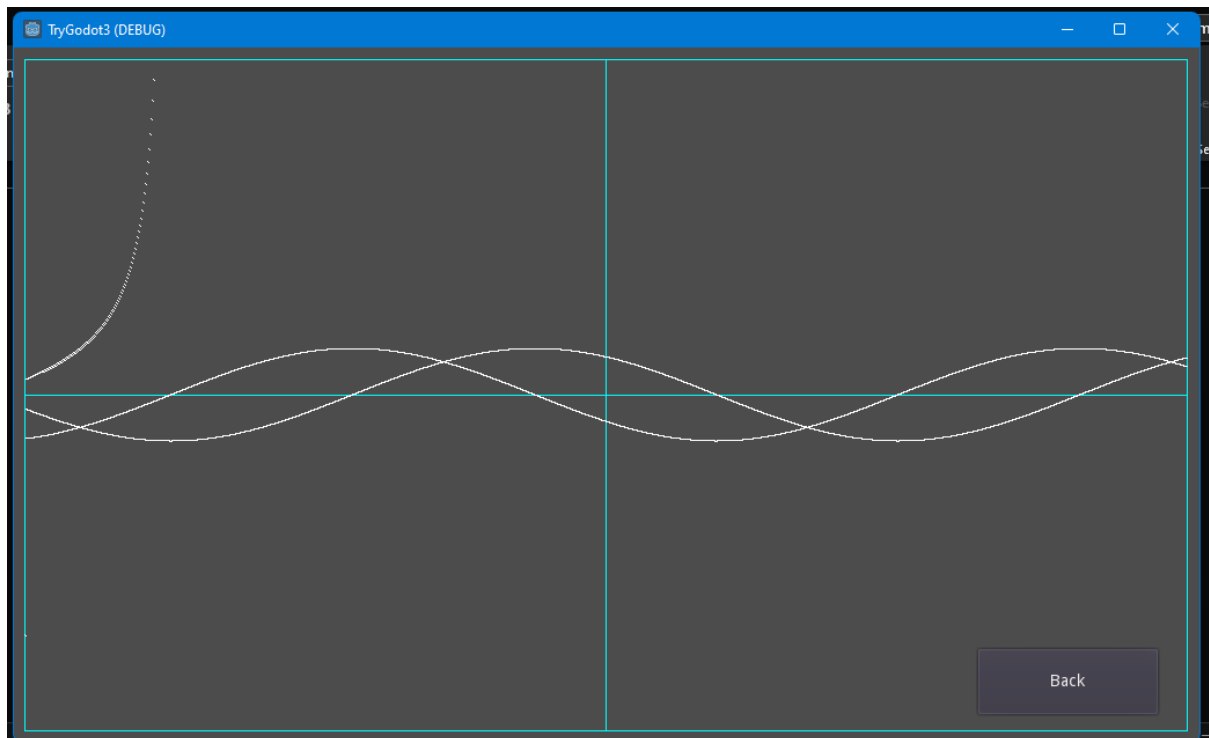
04 »   »   » x = x + 0.01
05 »   »   » xa = xb
06 »   »   » ya = yb
07

```



## 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

```

```

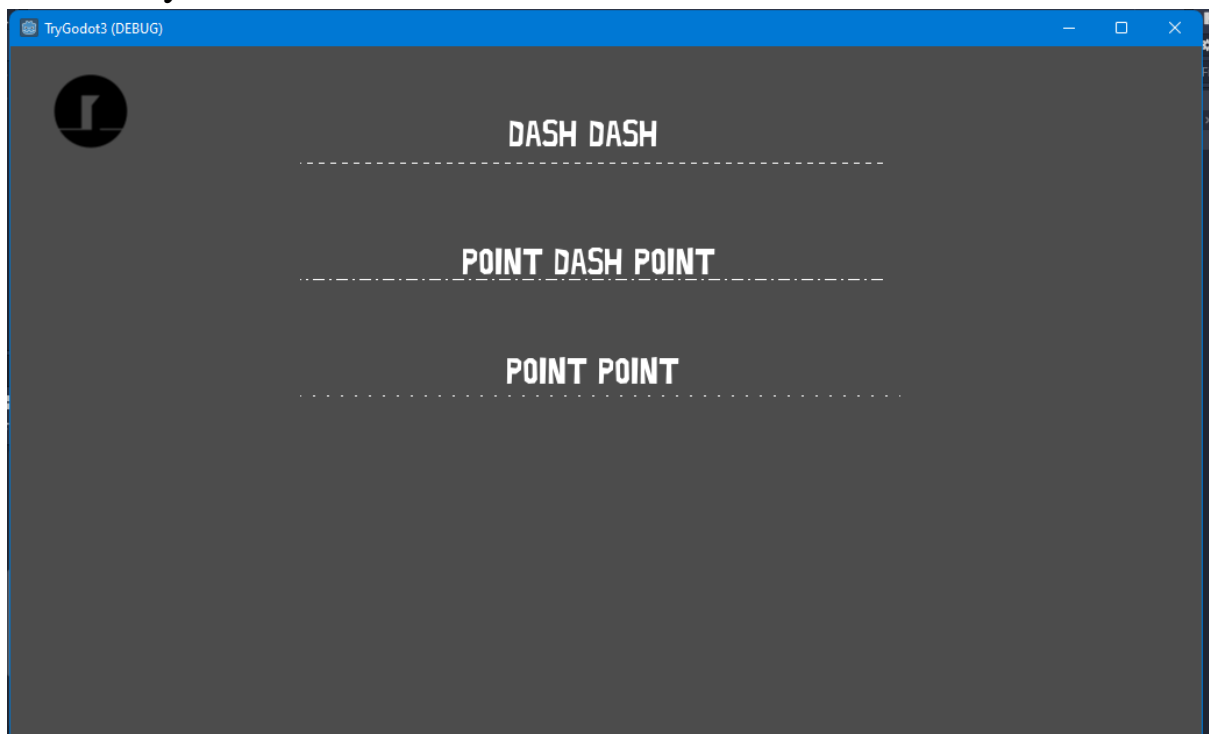
35     x = 0
36     y = 0
37     while (((x+10 <= windowSizeX-10) && (x+10 >= 10)) &&
38         (titikTengah.y-y >= titikTengah.y-limit)):
39
40         y = (a*x*x)-(b*x)+c
41         put_pixel(titikTengah.x+x,titikTengah.y-y,Color(1,1,1))
42         x = x+0.01
43
44 ~ func _draw():
45     pass
46     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)
```

## LESSON LEARN

Setelah saya mengerjakan tugas ini saya mendapat banyak pelajaran seperti cara memodifikasi garis di dalam godot engine

Hasil temuan library : parse, get\_window size, process (delta), \_draw, draw\_primitive, get\_viewport, get\_window\_size dan lainnya

## CURHAT DOSEN

Ketika saya mengerjakan projek godot ini, seringkali not responding.

Contohnya ketika saya sudah mendapatkan ide bagaimana logikanya, yang jadi masalah adalah bagaimana cara runningnya bila not responding terus, maka hasil logika tsb tidak bisa diuji apakah benar atau tidak seperti ketika ada 'while' duakali.

Kadang tidak bisa membagi waktu karena mengerjakan tugas tugas lainnya yang deadlinnya dekat sehingga tugas godot ini sering terlupakan dan malah mengerjakan h-1 sebelum mengumpulkan