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 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

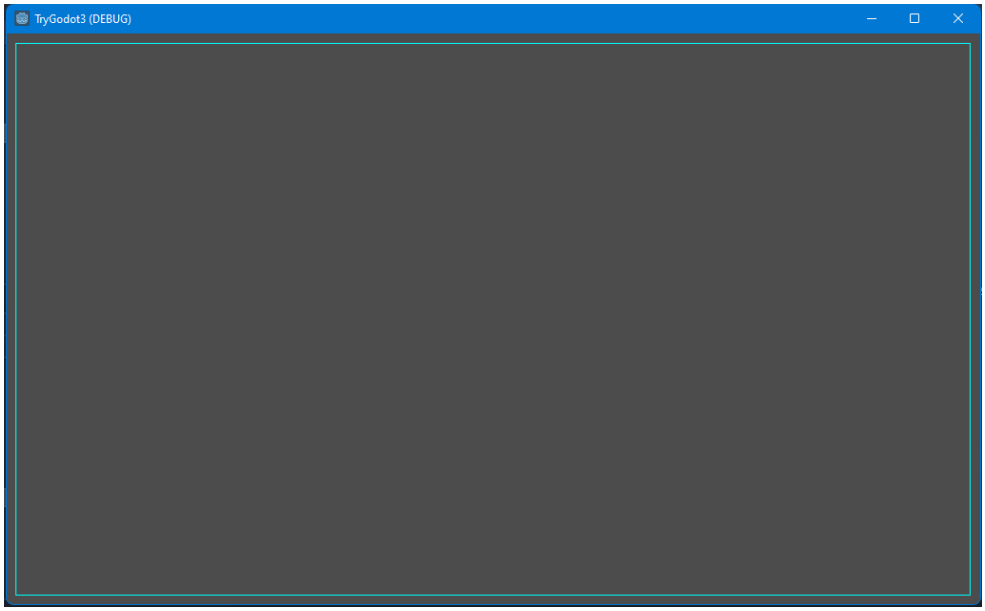
A. Pertanyaan 1

Buatlah bingkai 10px

a) Dapatkan koordinat pembangun bingkai (titik A, B, C, D) A ~ kiri atas, B kanan atas, C kiri bawah, dan D kanan bawah.

b) 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’

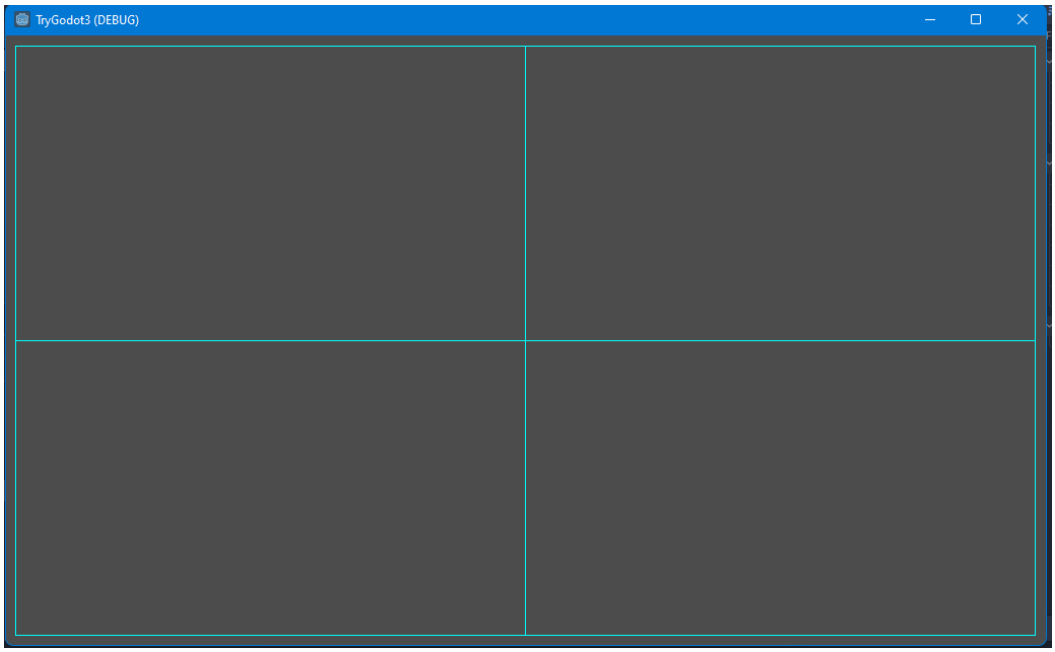
B. Pertanyaan 2

Buatlah cartesian

a.) Dapatkan koordinat pembangun kartesian dengan mendapatkan titik tengah dari width dan height.

b.) Lalu disambungkan menggunakan algoritma pengembangan garis seperti lineDDA atau Bersenham untuk membangun kartesian.

Berikut adalah hasil output dari pertanyaan 2:



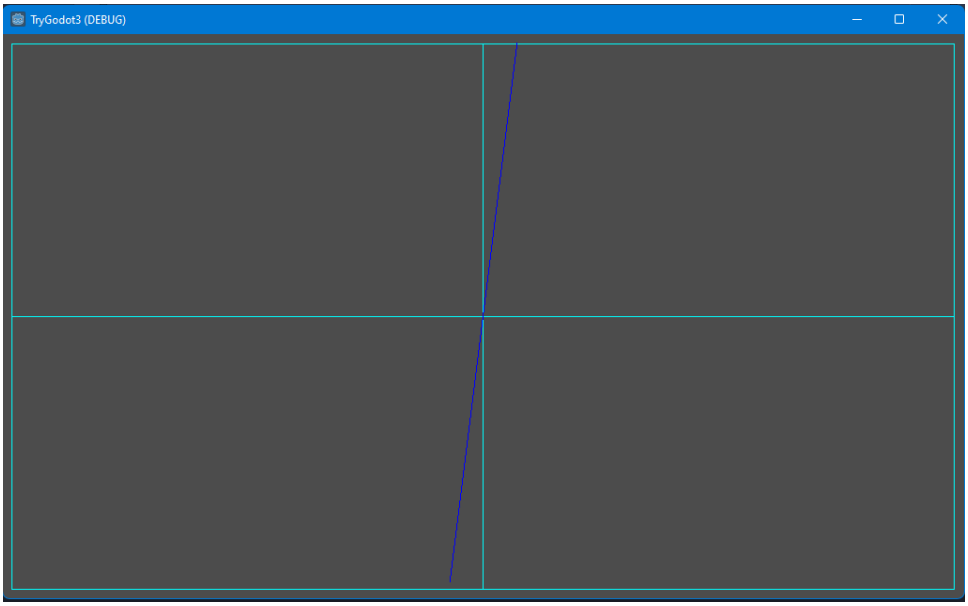
Berikut adalah scripnya

```
5 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

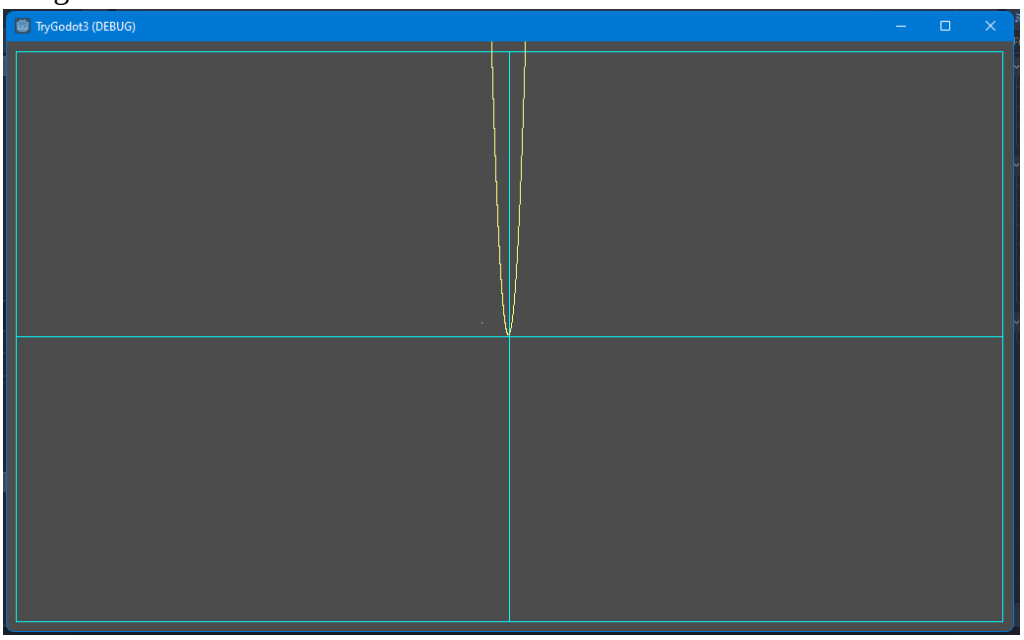
Berikut adalah hasil output dari pertanyaan 3:

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)
```

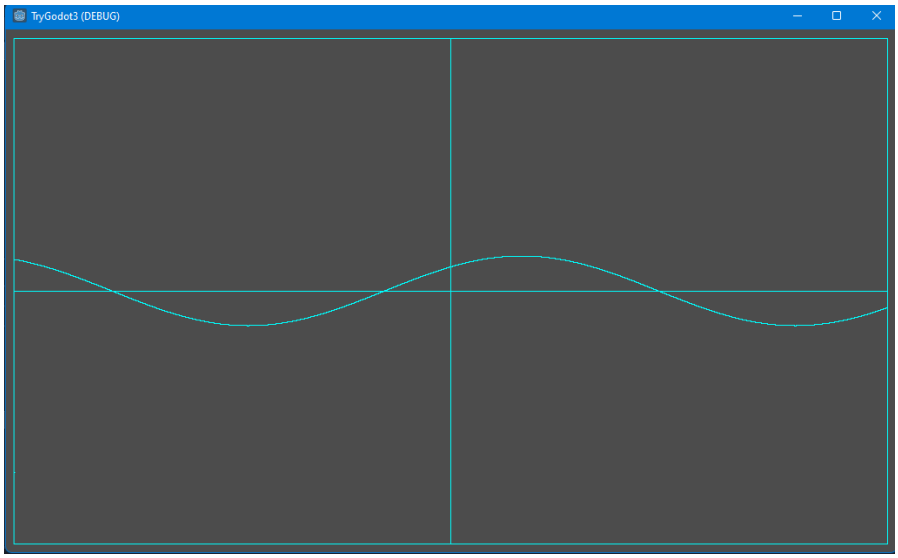
b. Fungsi Kuadrat



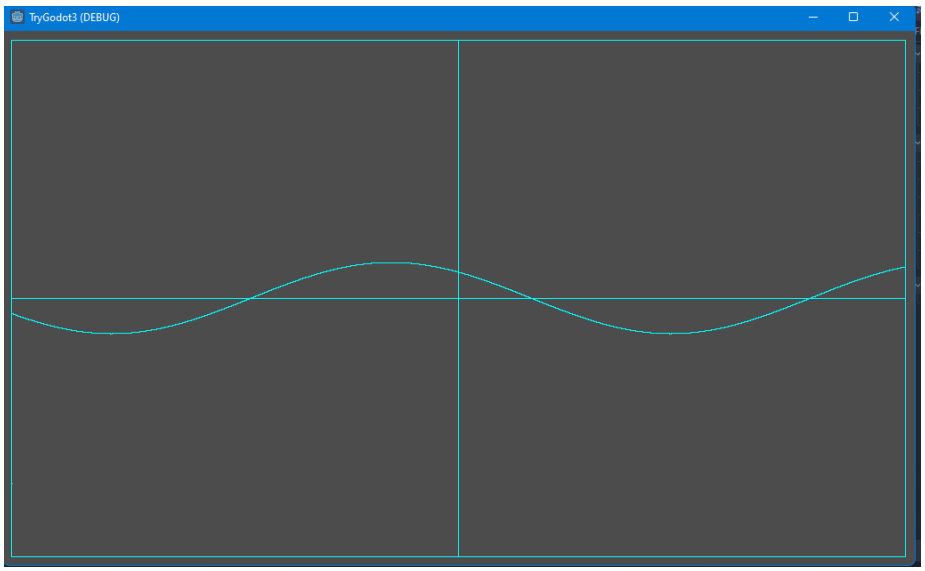

```
func fungsi_kuadrat(x_koef, konstanta, color):
    var width = get_viewport().size.x
    var height = get_viewport().size.y
    var y = pow(x_koef,2) + (3 * x_koef) + konstanta

    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

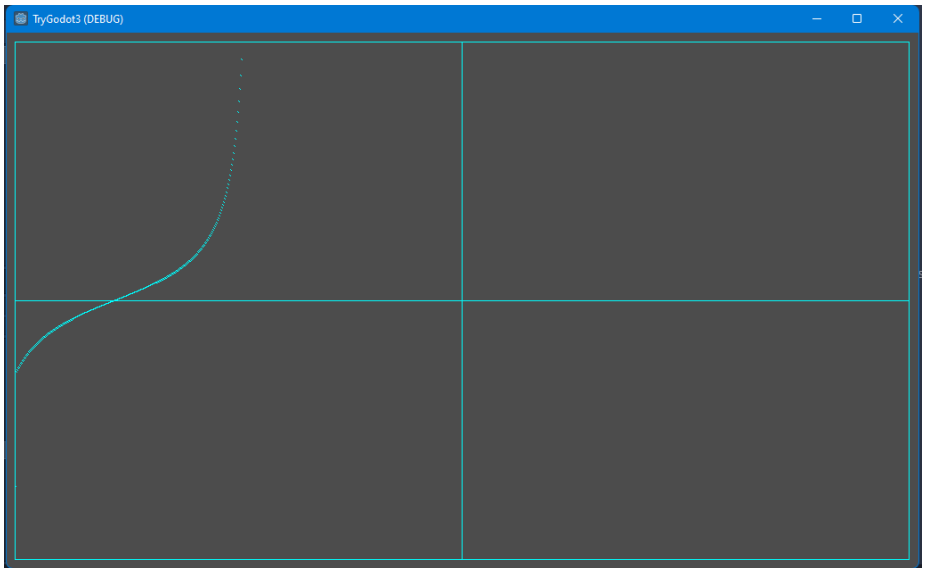
- Sin



- Cos



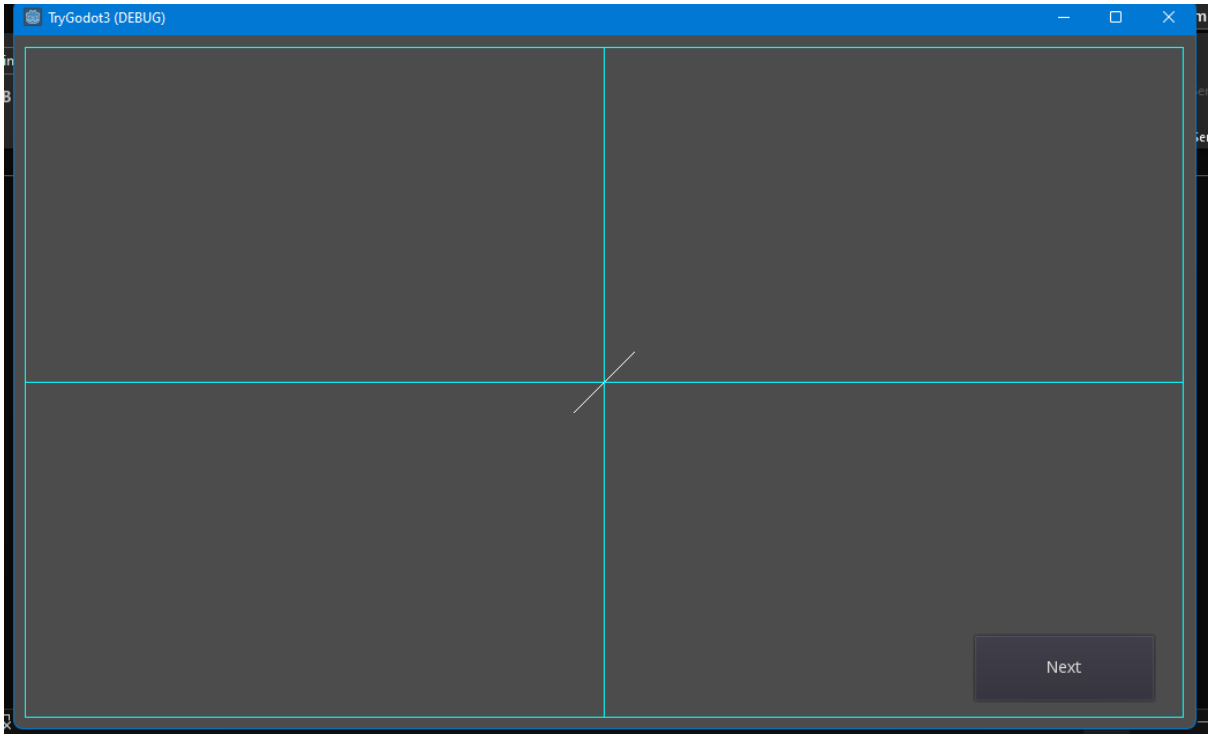
- Tan

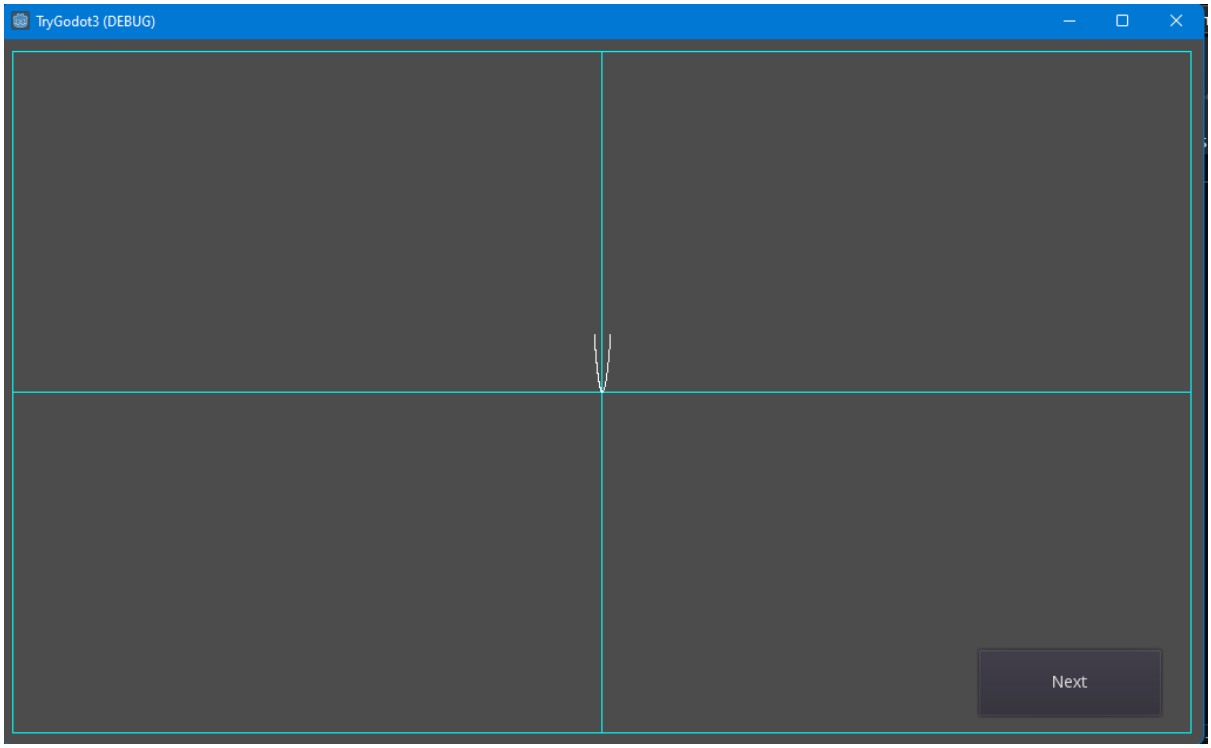


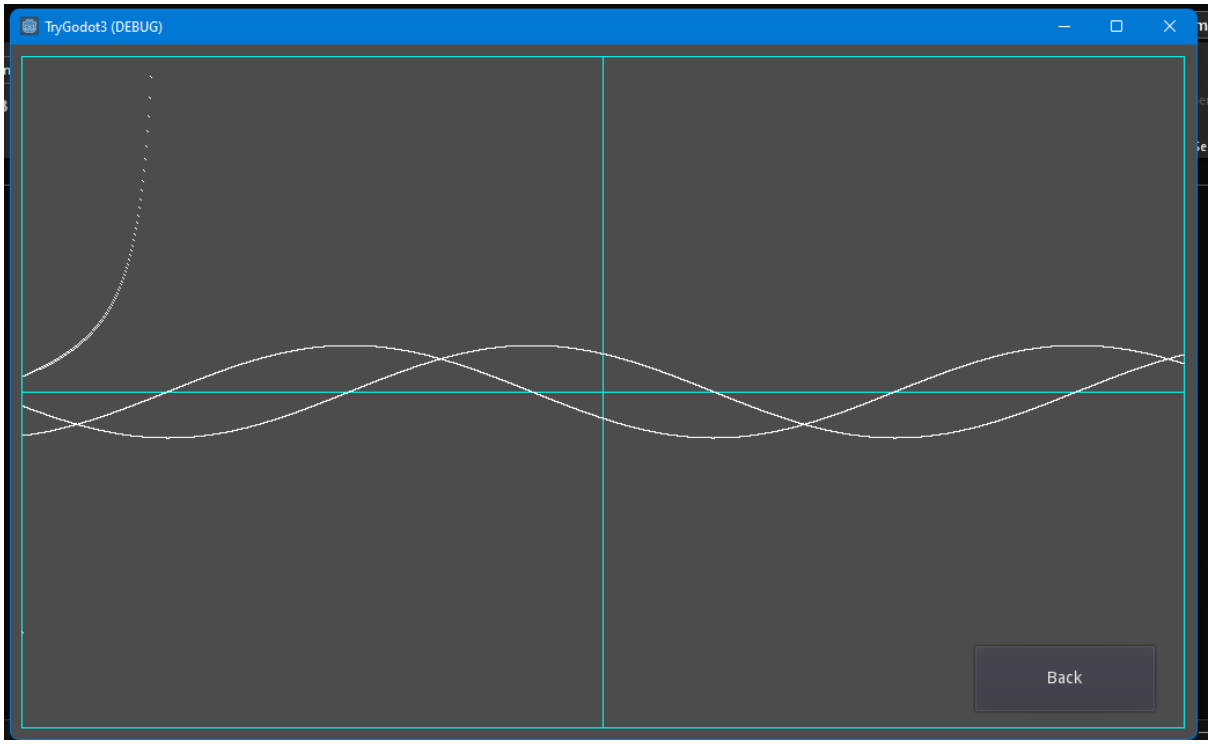
```
func fungsi_trigonometri(tipe ,x, colors):
    var width = get_viewport().size.x
    var height = get_viewport().size.y
    var xa = 10
    var ya = (width-10)/2
    var y; var xb; var yb;

    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

```gdscript
extends "res://TCSN/line.gd"

# Declare member variables here. Examples:
# var a = 2
# var b = "text"
var limit = titikTengah.y
var limit2 = titikTengah.y

# Called when the node enters the scene tree for the first time.
func _ready():
    pass # Replace with function body.
func _process(delta):
    pass
    limit = limit-1
    if (titikTengah.y+limit == 24):
        limit = 0
    limit2 = limit2+1
    if (titikTengah.y+limit2 == 24):
        limit = 0


    update()

func fungsi_linear(m,c):
    pass

    var xa # titikX = 0
    var ya # titikY = 0
    var xb
    var yb

```

```
func fungsi_linear(m,c):
    pass

    var xa # titikX = 0
    var ya # titikY = 0
    var xb
    var yb

    xa = titikTengah.x
    ya = titikTengah.y-c
    while ((xa <= windowSizeX-24) && (xa >= 24) && (ya >= limit)): # steps = 20
        xa += gradientFinderX(m)
        ya -= gradientFinderY(m)
        print ("titik xa ", xa, " titik ya ", ya)
        put_pixel(xa,ya,Color(1,1,1,1))

    xb = titikTengah.x
    yb = titikTengah.y-c
    while ((xb <= windowSizeX-24) && (xb >= 24) && (yb <= limit2)): # steps = 20
        xb -= gradientFinderX(m)+c
        yb += gradientFinderY(m)+c
        put_pixel(xb,yb,Color(1,1,1,1))

func _draw():
    frame()
    kartesian()
    fungsi_linear(1,0)
```

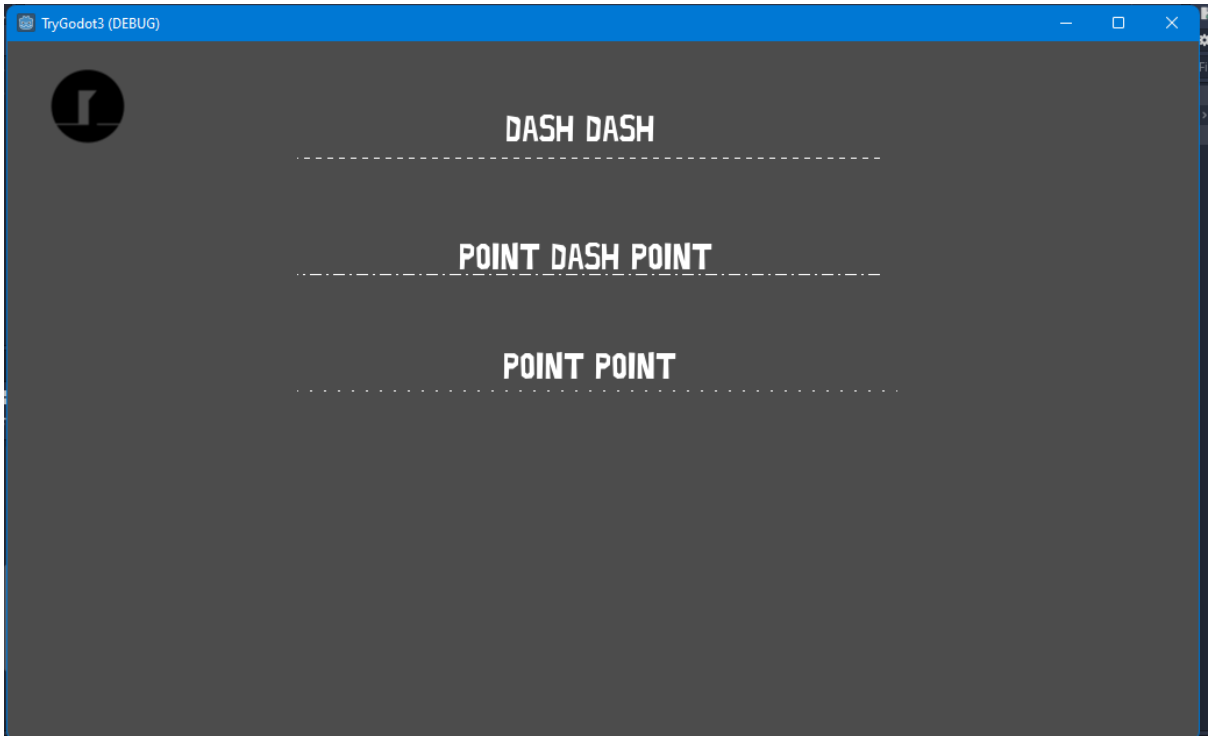
Kuadrat

```
extends "res://TCSN/line.gd"
var x = 0
var y = 0
var limit = 0

# Declare member variables here. Examples:
# var a = 2
# var b = "text"

func _process(delta):
    pass
    limit = limit+1
    if (titikTengah.y-limit == 24):
        limit = 0
    update()


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

func fungsi_kuadrat(a,b,c):
    var x = 0
    var y = 0
    pass
    # ax^2 + bx + c
    # a = gradien; b = variabel; c = konstanta
    while (((x+10 <= windowSizeX-10) && (x+24 >= 10)) &&
    (titikTengah.y-y >= titikTengah.y-limit)):
        y = (a*x*x)-(b*x)+c
        put_pixel(titikTengah.x-x,titikTengah.y-y,Color(1,1,1,1))

        x = x+0.01

    x = 0
    y = 0
    while (((x+10 <= windowSizeX-10) && (x+10 >= 10)) &&
    (titikTengah.y-y >= titikTengah.y-limit)):

        y = (a*x*x)-(b*x)+c
        put_pixel(titikTengah.x+x,titikTengah.y-y,Color(1,1,1,1))
        x = x+0.01

func _draw():
    pass
    frame()
    kartesian()
    fungsi_kuadrat(1,0,0)

# Called every frame. 'delta' is the elapsed time since the previous frame.
#func _process(delta):
#    pass


func _on_Button_pressed():
    get_tree().change_scene("res://TCSN/TrigonometriAnimasi.tscn")
```

Trigonometri

```gdscript
extends "res://TCSN/line.gd"



# Declare member variables here. Examples:
# var a = 2
# var b = "text"
var value = 0;

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


# Called every frame. 'delta' is the elapsed time since the previous frame.
#func _process(delta):
#    pass

func _process(delta):
    value -= 3 * delta;
    update()

func _draw():
    frame()
    kartesian()
    fungsi_trigonometri('sin', value, Color(1,1,1,1))
    fungsi_trigonometri('cos', value, Color(1,1,1,1))
    fungsi_trigonometri('tan', value, Color(1,1,1,1))


func _on_Button_pressed():
    get_tree().change_scene('res://TCSN/line_menu.tscn')

```

E. Pertanyaan 5



Cara membuat variasi garis adalah dengan memodifikasi Increment titiknya

Dash Dash

```
func Dash_dash(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 % 5 == 0):
            x += 5
        x += xIncrement
        y += yIncrement
        put_pixel(round(x), round(y), color)
```

Point Dash Point

```
func point_dash(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 = round(dx/ float(steps))
    yIncrement = round(dy/ float(steps))
    put_pixel(round(x), round(y), color)

    for k in steps:
        if(int(x) % 10 == 0):
            x+= 4
            tampung = x
        #if(int(x) % 2 == 4):
            #x+= 20
        if(x == tampung + 2):
            x+= 4
        x += xIncrement
        y += yIncrement
        put_pixel(round(x), round(y), color)
```

Point Point

```gdscript
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.

```gdscript
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)
```

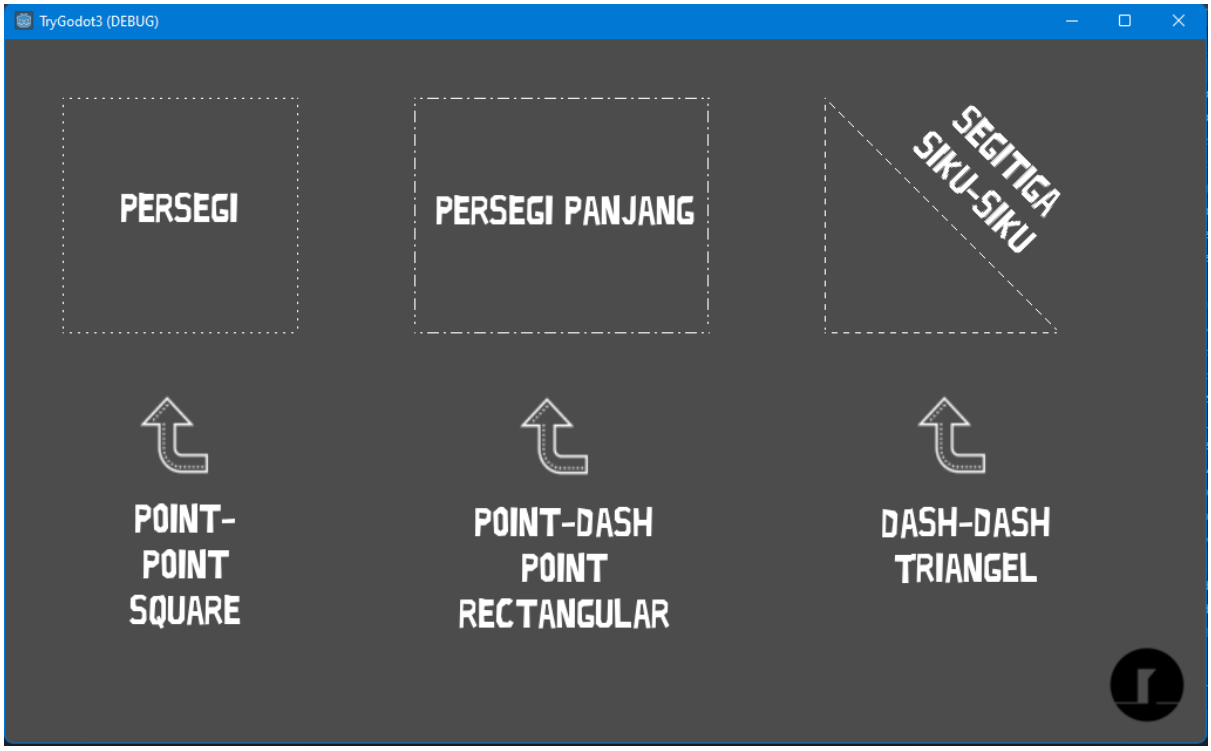
```gdscript
func lineDDA(xa : float, ya : float, xb : float, yb : float):
    var dx = xb - xa
    var dy = yb - ya
    var steps
    var xIncrement
    var yIncrement
    var x = xa
    var y = ya
    var res = PoolVector2Array()

    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)))

    for k in steps:
        x += xIncrement
        y += yIncrement
        res.append(Vector2(round(x), round(y)))
    #Dipanggil agar garis tampil
    put_pixel_all(res, color)
    return res
```

```gdscript
extends 'res://TCSN/primitif_array.gd'



# Declare member variables here. Examples:
# var a = 2
# var b = "text"

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

func persegi(titik_awal: Vector2, panjang_sisi):
    var res = PoolVector2Array()
    res.append_array(lineDDA(titik_awal.x, titik_awal.y, titik_awal.x + panjang_sisi, titik_awal.y)) #sisi atas
    res.append_array(lineDDA(titik_awal.x, titik_awal.y + panjang_sisi, titik_awal.x + panjang_sisi, titik_awal.y + panjang_sisi)) #sisi bawah
    res.append_array(lineDDA(titik_awal.x, titik_awal.y, titik_awal.x, titik_awal.y + panjang_sisi)) #sisi kiri
    res.append_array(lineDDA(titik_awal.x + panjang_sisi, titik_awal.y, titik_awal.x + panjang_sisi, titik_awal.y + panjang_sisi)) #sisi kanan
    return res

func persegi_panjang(titik_awal: Vector2, panjang, lebar):
    var res = PoolVector2Array()
    res.append_array(lineDDA(titik_awal.x, titik_awal.y, titik_awal.x + panjang, titik_awal.y)) #sisi atas
    res.append_array(lineDDA(titik_awal.x, titik_awal.y + lebar, titik_awal.x + panjang, titik_awal.y + lebar)) #sisi bawah
    res.append_array(lineDDA(titik_awal.x, titik_awal.y, titik_awal.x, titik_awal.y + lebar)) #sisi kiri
    res.append_array(lineDDA(titik_awal.x + panjang, titik_awal.y, titik_awal.x + panjang, titik_awal.y + lebar)) #sisi kanan
    return res

func segitiga_siku_siku(titik_awal: Vector2, alas, tinggi):
    var res = PoolVector2Array()
    res.append_array(lineDDA(titik_awal.x, titik_awal.y, titik_awal.x, titik_awal.y + tinggi)) #sisi kiri
    res.append_array(lineDDA(titik_awal.x, titik_awal.y + tinggi, titik_awal.x + alas, titik_awal.y + tinggi)) #sisi bawah
    res.append_array(lineDDA(titik_awal.x, titik_awal.y, titik_awal.x + alas, titik_awal.y + tinggi)) #sisi kanan
    return res
```

```gdscript
func trapesium_siku_siku(titik_awal: Vector2, atas, tinggi, bawah):
    var res = PoolVector2Array()
    res.append_array(lineDDA(titik_awal.x, titik_awal.y, titik_awal.x + atas, titik_awal.y)) #sisi atas
    res.append_array(lineDDA(titik_awal.x, titik_awal.y + tinggi, titik_awal.x + bawah, titik_awal.y + tinggi)) #sisi bawah
    res.append_array(lineDDA(titik_awal.x, titik_awal.y, titik_awal.x, titik_awal.y + tinggi)) #sisi kiri
    res.append_array(lineDDA(titik_awal.x + atas, titik_awal.y, titik_awal.x + bawah, titik_awal.y + tinggi)) #sisi kanan
    return res

func trapesium_sama_kaki(titik_awal: Vector2, atas, tinggi, bawah):
    var res = PoolVector2Array()
    res.append_array(lineDDA(titik_awal.x, titik_awal.y, titik_awal.x + atas, titik_awal.y)) #sisi atas
    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
    res.append_array(lineDDA(titik_awal.x, titik_awal.y, (titik_awal.x - ((bawah - atas)/2)), titik_awal.y + tinggi)) #sisi kiri
    res.append_array(lineDDA(titik_awal.x + atas, titik_awal.y, (titik_awal.x - ((bawah - atas)/2)) + bawah, titik_awal.y + tinggi)) #sisi kanan
    return res

func jajargenjang(titik_awal: Vector2, panjang, tinggi, geser):
    var res = PoolVector2Array()
    res.append_array(lineDDA(titik_awal.x, titik_awal.y, titik_awal.x + panjang, titik_awal.y)) #sisi atas
    res.append_array(lineDDA((titik_awal.x - geser), titik_awal.y + tinggi, (titik_awal.x + panjang - geser), titik_awal.y + tinggi)) #sisi bawah
    res.append_array(lineDDA(titik_awal.x, titik_awal.y, (titik_awal.x - geser), titik_awal.y + tinggi)) #sisi kiri
    res.append_array(lineDDA(titik_awal.x + panjang, titik_awal.y, ((titik_awal.x + panjang) - geser), titik_awal.y + tinggi)) #sisi kanan
    return res
```

```
64 func _draw():
65     persegi(Vector2(50,50), 200)
66     persegi_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
```

```
#line_dda
func dash_dash(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 % 5 == 0):
                x += 5
            x += xIncrement
            y += yIncrement
            res.append(Vector2(round(x), round(y)))
            if(k > 95):
                break
            print(k)
    elif(xa == xb) :
        for k in steps :
            if(k % 5 == 0):
                y += 5
            y += yIncrement
            res.append(Vector2(round(x), round(y)))
            if(k > 95):
                break
            print(k)
```

```
    else:
        for k in steps:
            if(k % 5 == 0):
                x += 3
                y += 3
            x += xIncrement
            y += yIncrement
            put_pixel(round(x), round(y), color)
            if(k > 120):
                break
            print(steps)

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

```
func point_dash(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()
    var tampung = 0

    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(int(x) % 10 == 0):
                x+= 4
                tampung = x
            if(x == tampung + 2):
                x+= 4
            x += xIncrement
            y += yIncrement
            res.append(Vector2(round(x), round(y)))
            if(k > 145):
                break
    else :
        for k in steps:
            if(int(y) % 10 == 0):
                y+= 4
                tampung = y
            if(y == tampung + 2):
                y+= 4
            y += yIncrement
            res.append(Vector2(round(x), round(y)))
            if(k > 110):
                break
    #Dipanggil agar garis tampil
    put_pixel_all(res, color)
    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
```

```
func persegi(titik_awal: Vector2, panjang_sisi):
    var res = PoolVector2Array()
    point_point(titik_awal.x, titik_awal.y, titik_awal.x + panjang_sisi, titik_awal.y) #sisi atas
    point_point(titik_awal.x, titik_awal.y + panjang_sisi, titik_awal.x + panjang_sisi, titik_awal.y + panjang_sisi) #sisi bawah
    point_point(titik_awal.x, titik_awal.y, titik_awal.x, titik_awal.y + panjang_sisi) #sisi kiri
    point_point(titik_awal.x + panjang_sisi, titik_awal.y, titik_awal.x + panjang_sisi, titik_awal.y + panjang_sisi) #sisi kanan
    return res

func persegi_panjang(titik_awal: Vector2, panjang, lebar):
    var res = PoolVector2Array()
    point_dash(titik_awal.x, titik_awal.y, titik_awal.x + panjang, titik_awal.y) #sisi atas
    point_dash(titik_awal.x, titik_awal.y + lebar, titik_awal.x + panjang, titik_awal.y + lebar) #sisi bawah
    point_dash(titik_awal.x, titik_awal.y, titik_awal.x, titik_awal.y + lebar) #sisi kiri
    point_dash(titik_awal.x + panjang, titik_awal.y, titik_awal.x + panjang, titik_awal.y + lebar) #sisi kanan
    return res

func segitiga_siku_siku(titik_awal: Vector2, alas, tinggi):
    var res = PoolVector2Array()
    dash_dash(titik_awal.x, titik_awal.y, titik_awal.x, titik_awal.y + tinggi) #sisi kiri
    dash_dash(titik_awal.x, titik_awal.y + tinggi, titik_awal.x + alas, titik_awal.y + tinggi) #sisi bawah
    dash_dash(titik_awal.x, titik_awal.y, titik_awal.x + alas, titik_awal.y + tinggi) #sisi kanan
    return res
```

```
func _draw():
    persegi(Vector2(50,50), 200)
    persegi_panjang(Vector2(350,50),250 , 200)
    segitiga_siku_siku(Vector2(700,50), 200, 200)
```

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 koodinat 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.