KOMPUTER GRAFIK

Minggu 4

Oleh :

Nuno Alwi Azimah 201524057

POLITEKNIK NEGERI BANDUNG

2021

Daftar Isi

Task 1

Primitif.gd

```
func tambahMatrix(first:Array, second:Array):
	var r1 = first.size()
	var c1 = first[0].size()
	var r2 = second.size()
	var c2 = second[0].size()
	var result = []

	if r1 == r2 and c1 == c2:
		for i in range(r1):
			result.append([])
			for j in range(c2):
				result[i].append(0)

		for i in range(r1):
			for j in range(c1):
				result[i][j] = first[i][j] + second[i][j]

	return result

func kurangMatrix(first:Array, second:Array):
	var r1 = first.size()
	var c1 = first[0].size()
	var r2 = second.size()
	var c2 = second[0].size()
	var result = []

	if r1 == r2 and c1 == c2:
		for i in range(r1):
			result.append([])
			for j in range(c2):
				result[i].append(0)

		for i in range(r1):
			for j in range(c1):
				result[i][j] = first[i][j] - second[i][j]

	return result
func kaliMatrix(first:Array, second:Array):
	var r1 = first.size()
	var c1 = first[0].size()
	var r2 = second.size()
	var c2 = second[0].size()
	var result = []

	if c1 == r2:
		for i in range(r1):
```

```
                result.append([])
                for j in range(c2):
                        result[i].append(0)

        for i in range(r1):
                for j in range(c2):
                        for k in range(c1):
                                result[i][j] += first[i][k] * second[k][j];
return result
```

Task 2

Primitif.gd

```
func scaleVector2(point:Vector2, scaleFactor:int, tikpus:Vector2=Vector2(0,
0)):
	var matrikRumus = [[scaleFactor,0], [0, scaleFactor]]
	var matrikPoint = [[point.x - tikpus.x], [point.y - tikpus.y]]
	var matrikTikpus = [[tikpus.x], [tikpus.y]]
	var res = tambahMatrix(kaliMatrix(matrikRumus, matrikPoint),
matrikTikpus)

	point.x = res[0][0]
	point.y = res[1][0]

	return point

func rotasiVector2(point:Vector2, derajat:float, tikpus:Vector2=Vector2(0, 0)):
	derajat = deg2rad(derajat)

	var matrikRumus = [[cos(derajat), -sin(derajat)], [sin(derajat),
cos(derajat)]]
	var matrikPoint = [[point.x - tikpus.x], [point.y - tikpus.y]]
	var matrikTikpus = [[tikpus.x], [tikpus.y]]
	var res = tambahMatrix(kaliMatrix(matrikRumus, matrikPoint),
matrikTikpus)

	point.x = res[0][0]
	point.y = res[1][0]

	return point


func translasiVector2(point:Vector2, a:int, b:int):
	point.x += a
	point.y += b
	return point
```

Task 3

Shape.gd

```
func draw_ketupat(length:int, pos:Vector2, color:Color=Color.red, thick:int=1,
dot:int=1, dash:bool=false):
	var point = Vector2(length, 0)
	var a
	var b
	for i in range(4):
		a = translasiVector2(point, pos.x, pos.y)
		point = rotasiVector2(point, 90)
		b = translasiVector2(point, pos.x, pos.y)
		line_bresenham_2(a, b,  color, thick, dash)

func draw_jajargenjang(length:int, pos:Vector2, color:Color=Color.red,
thick:int=1, dot:int=1, dash:bool=false):
	var point = Vector2(length, length)
	var a
	var b
	var pindah = 0
	for i in range(4):
		a = translasiVector2(point, pos.x + pindah, pos.y)

		if (i + 1) % 2 != 0:
			point = rotasiVector2(point, 90)
			b = translasiVector2(point, pos.x + pindah, pos.y)

		else:
			point = rotasiVector2(point, 180)
			b = translasiVector2(point, pos.x + pindah, pos.y)

			point = Vector2(-length, -length)
			pindah = 2 * length
		line_bresenham_2(a, b,  color, thick, dash)

func draw_layang2(length:int, pos:Vector2, color:Color=Color.red, thick:int=1,
dot:int=0, dash:bool=false):
	var point = Vector2(0, -length)
	var a
	var b
	var temp = point
	for i in range(4):
		a = translasiVector2(point, pos.x, pos.y)
		point = rotasiVector2(temp, 90)
		temp = point
		b = translasiVector2(point, pos.x, pos.y)
		if i==1:
			b = translasiVector2(b, 0, length)
			point = translasiVector2(point, 0, length)
```

```
                line_dda_2(a, b, color, thick, dot, dash)

func draw_trapesium(length:int, pos:Vector2, color:Color=Color.red, thick:int=1,
dot:int=0, dash:bool=false):
        var point = Vector2(0, -length)
        var a
        var b
        for i in range(4):
                a = translasiVector2(point, pos.x, pos.y)
                point = rotasiVector2(point, -90)
                b = translasiVector2(point, pos.x, pos.y)

                if (i+1)%2 == 0:
                        b = translasiVector2(b, 0, length)
                        point = translasiVector2(point, 0, length)

                line_dda_2(a, b, color, thick, dot, dash)

func draw_segilima(length:int, pos:Vector2, color:Color=Color.red, thick:int=1,
dot:int=0, dash:bool=false):
        var point = Vector2(length, length)
        var a
        var b
        for i in range(360/5):
                a = translasiVector2(point, pos.x, pos.y)
                point = rotasiVector2(point, 360/5)
                b = translasiVector2(point, pos.x, pos.y)
                line_dda_2(a, b, color, thick, dot, dash)

func draw_segienam(length:int, pos:Vector2, color:Color=Color.red, thick:int=1,
dot:int=0, dash:bool=false):
        var point = Vector2(length, length)
        var a
        var b
        for i in range(360/6):
                a = translasiVector2(point, pos.x, pos.y)
                point = rotasiVector2(point, 360/6)
                b = translasiVector2(point, pos.x, pos.y)
                line_dda_2(a, b, color, thick, dot, dash)

func draw_lingkaran(pos:Vector2, r:int, color:Color):
        var xCenter:int = pos.x
        var yCenter:int = pos.y
        var x:int = 0
        var y = r
        var p = 1 - r

        circlePlotPoints(xCenter, yCenter, x, y, color)

        while x < y:
```

```
		x += 1
		if p < 0:
			p += 2 * x + 1
		else:
			y -= 1
			p += 2 * (x - y) + 1
		circlePlotPoints(xCenter, yCenter, x, y, color)

func circlePlotPoints(xCenter:int, yCenter:int, x:int, y:int, color:Color):
	put_pixel(xCenter + x, yCenter + y, color)
	put_pixel(xCenter - x, yCenter + y, color)
	put_pixel(xCenter + x, yCenter - y, color)
	put_pixel(xCenter - x, yCenter - y, color)
	put_pixel(xCenter + y, yCenter + x, color)
	put_pixel(xCenter - y, yCenter + x, color)
	put_pixel(xCenter + y, yCenter - x, color)
	put_pixel(xCenter - y, yCenter - x, color)

func draw_ellipse(pos:Vector2, Rx:int, Ry:int, color:Color, tipe:String="thick"):
	var xCenter:int = pos.x
	var yCenter:int = pos.y
	var Rx2 = Rx*Rx
	var Ry2 = Ry*Ry
	var twoRx2 = 2*Rx2
	var twoRy2 = 2*Ry2
	var p
	var x = 0
	var y = Ry
	var px = 0
	var py = twoRx2*y
	var i = x

	ellipsePlotPoints(xCenter,yCenter,x,y, color)
	#Region1
	p = round(Ry2 - (Rx2 * Ry) + (0.25 * Rx2))
	while (px < py):
		x = x + 1
		i = i + 1
		px += twoRy2
		if (p < 0):
			p += Ry2 + px
		else :
			y = y - 1
			py -= twoRx2
			p += Ry2 + px - py
		ellipsePlotPoints(xCenter,yCenter,x,y, color)

	#Region2
	p = round(Ry2 * (x+0.5) * (x + 0.5) + Rx2 * (y-1) * (y-1) - Rx2 * Ry2)
	while (y > 0):
```

```
                    y = y - 1
                    i = i + 1
                    py -= twoRx2
                    if (p > 0):
                            p += Rx2 - py
                    else :
                            x = x + 1
                            px += twoRy2
                            p += Rx2 - py + px
                    ellipsePlotPoints(xCenter,yCenter,x,y, color)

func ellipsePlotPoints(xCenter:int, yCenter:int, x:int, y:int, color:Color):
        put_pixel(xCenter + x, yCenter + y, color)
        put_pixel(xCenter - x, yCenter + y, color)
        put_pixel(xCenter + x, yCenter - y, color)
        put_pixel(xCenter - x, yCenter - y, color)
```

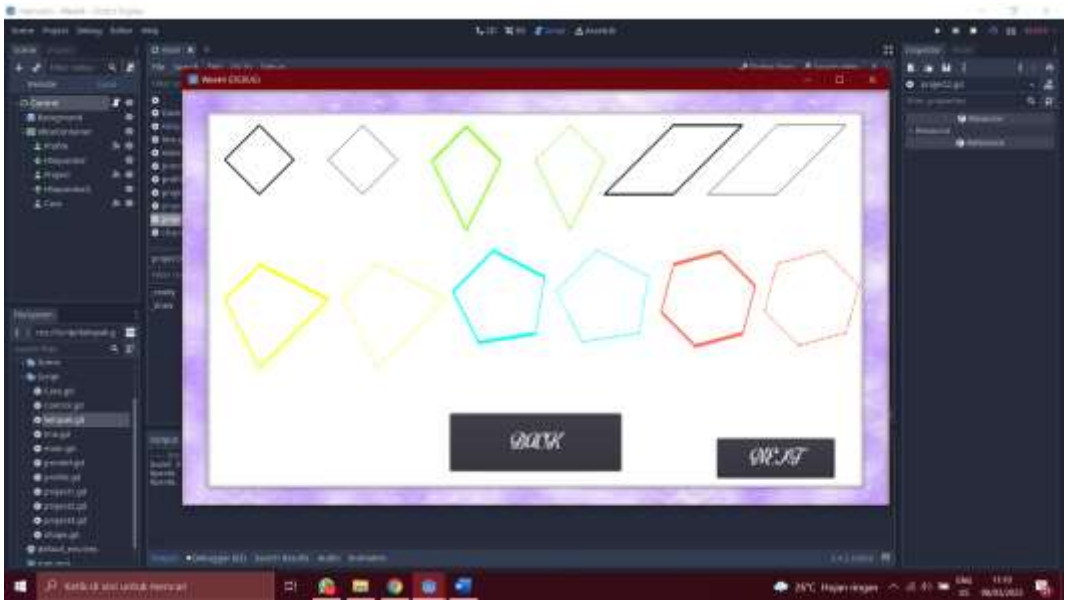
Task 4

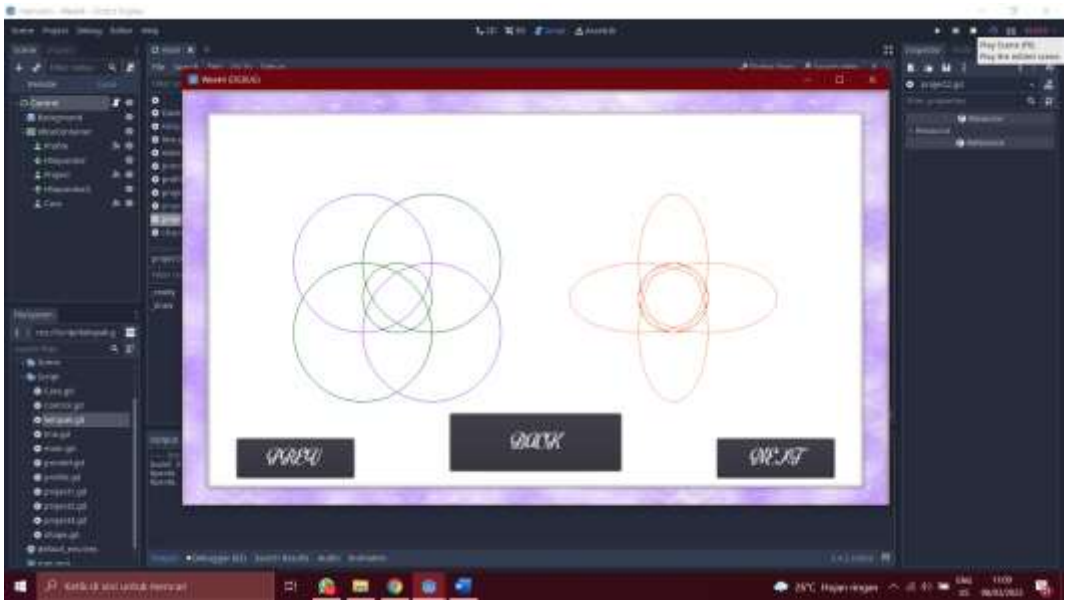
Kelopak.gd

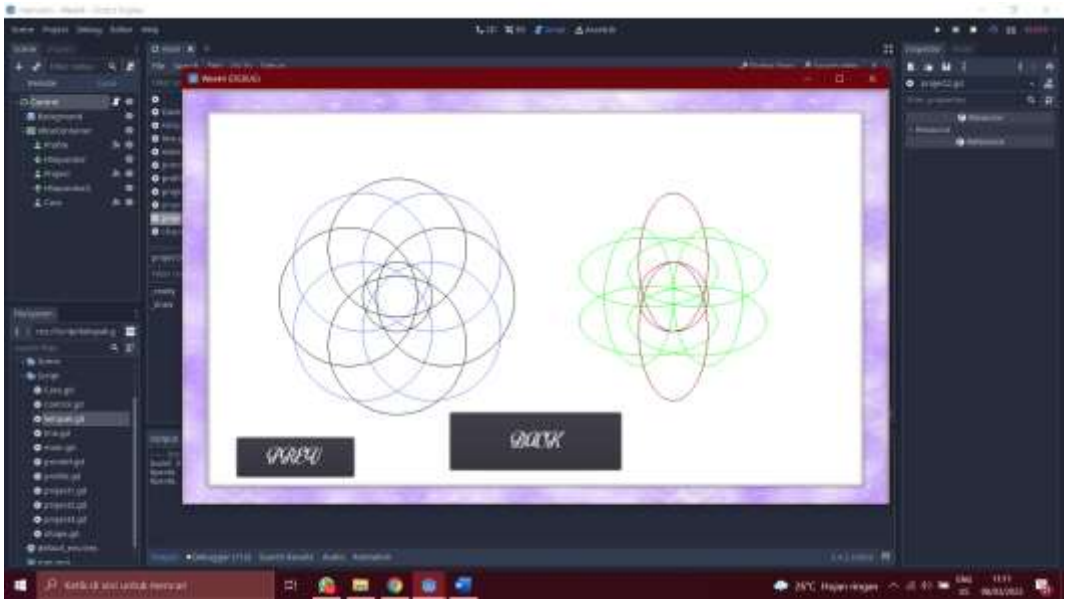
```
extends "res://Script/shape.gd"


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

func bunga_kelopak(n:int ,pos:Vector2, color:Color, color2:Color):
	var point = Vector2(50, 50)
	draw_lingkaran(pos, 50, color2)
	for i in range(n):
		point = rotasiVector2(point, 360/n)
		if (i+1)%2 == 0:
			draw_lingkaran(translasiVector2(point, pos.x, pos.y),
100, color)
		else:
			draw_lingkaran(translasiVector2(point, pos.x, pos.y),
100, color2)

func bunga_4_kelopak(tipus:Vector2, color:Color, color2:Color):
	var n = 4
	var point = Vector2(0, 50)
	draw_lingkaran(tipus, 50, color2)
	for i in range(n):
		point = rotasiVector2(point, 360/n)
		if (i+1)%2 == 0:
			draw_ellipse(translasiVector2(point, tipus.x, tipus.y), 50,
100, color)
		else:
			draw_ellipse(translasiVector2(point, tipus.x, tipus.y),
100, 50, color)

func bunga_8_kelopak(tipus:Vector2, color:Color, color2:Color):
	var n = 8
	var point = Vector2(0, 50)
	draw_lingkaran(tipus, 50, color2)
	for i in range(n):
		point = rotasiVector2(point, 360/n)
		if(i+1)%4 == 0:
			draw_ellipse(translasiVector2(point, tipus.x, tipus.y), 50,
100, color2)
		elif (i+1)%2 == 0:
			draw_ellipse(translasiVector2(point, tipus.x, tipus.y), 50,
100, color)
		else:
```

```
                draw_ellipse(translasiVector2(point, tipus.x, tipus.y),
100, 50, color)
```

Task 5







Leason Learn

What Went Well?
Saat mengerjakan pembuatan fungsi bentuk dasar dan juga kelopak bunga saya tidak mengalami kesulitan dan berjalan lancar

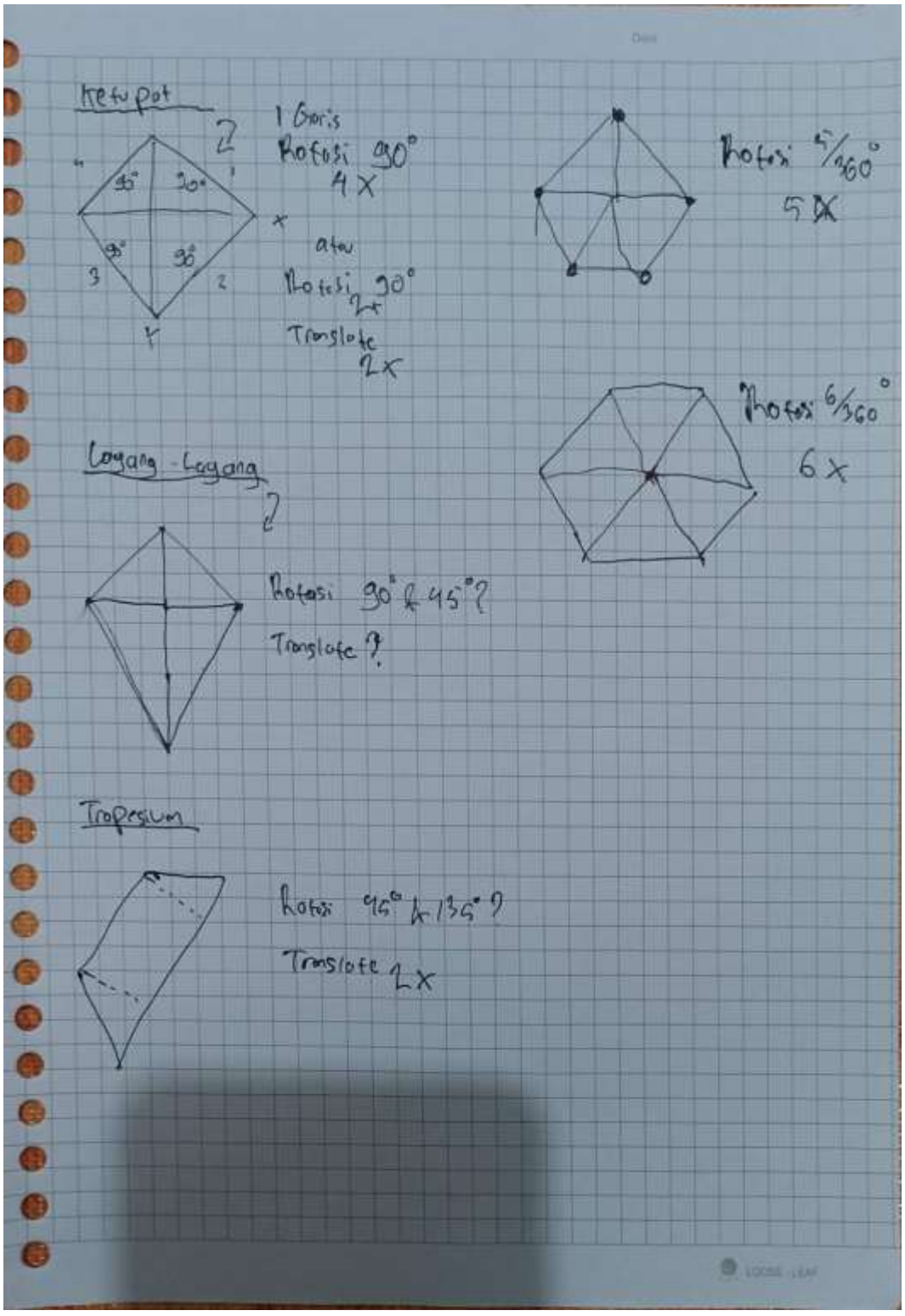
What didn’t go Well? Solutions?
Saat mengerjakan transformasi sedikit mengalami kesulitan dan solusinya melihat referensi dan juga tugas project yang sudah dikerjakan oleh teman, dan juga masih belum bisa memiringkan elips

What might have been better handled if done differently?
Karena tugas waktu yang saya kerjakan saja telat 1 minggu oleh karena itu sepertinya saya butuh keringanan waktu yang lebih untuk mengerjakan tugas ini.

What recommendations would you give to others who might be
Pelajari terlebih dahulu bentuk matrik dan juga transformasi

involved in future projects of a similar type?
Harus lebih menghargai waktu agar tidak terlambat untuk kedepannya

Lampiran


Ketupat
1 Garis
Rotasi 90°
4 X
atau
Rotasi 180°
2x
Translate
2X
Rotasi 5/360°
5X
Rotasi 6/360°
6x
Layang - Layang
Rotasi 90° & 45°?
Translate ?
Trapesium
Rotasi 45° & 135°?
Translate 2x

Referensi

Admin. (2021) Godot Docs:
https://docs.godotengine.org/en/stable

Aldila Ekasatya. (2022) ROTASI(PUTARAN)
https://www.academia.edu/20620812/ROTASI_PUTARAN_

Admin. (2018) sheetmath: contoh soal matriks
https://www.sheetmath.com/2016/08/contoh-soal-matriks-penjumlahan.html

Rizhka Tedi. (2020) Pengertian dan jenis jenis transformasi geometri
https://www.ruangguru.com/blog/pengertian-dan-jenis-jenis-transformasi-geometri