

MODUL 3

KOMPUTER GRAFIK 2D
LINGKARAN, ELLIPS DAN ATRIBUTE GARIS

D4 TEKNIK INFORMATIKA
JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA
POLITEKNIK NEGERI BANDUNG



MAHASISWA 000 | KOMPUTER GRAFIK | FEBRUARI, 26 2025

CONTENTS

LINGKARAN I

MIDPOINT CIRCLE ALGORITHM 2

ELLIPS 5

MIDPOINT ELLIPS ALGORITHM 6

TASK PRAKTIKUM I 3

PENGUMPULAN I 7

LINGKARAN

Lingkaran merupakan bentuk dasar yang biasa digunakan untuk membuat gambar atau objek yang kompleks, seperti dekorasi, batik, dan lain-lain. Pada paket library komputer grafik, sebagian atau lingkaran penuh dapat dibuat menggunakan sebuah prosedur. Secara general, prosedur tersebut dapat menghasilkan garis dan ellips.

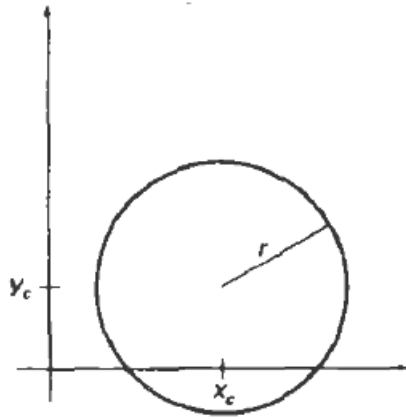


Figure 3-12
Circle with center coordinates (x_c, y_c) and radius r .

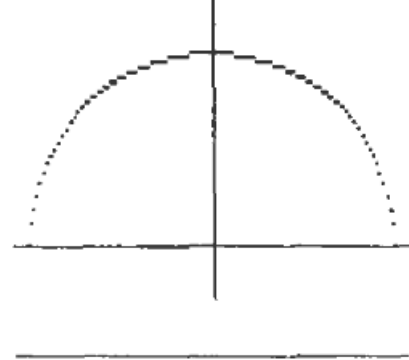


Figure 3-13
Positive half of a circle plotted with Eq. 3-25 and with $(x_c, y_c) = (0, 0)$.

Persamaan Lingkaran

$$(x - x_c)^2 + (y - y_c)^2 = r^2$$

$$y = y_c \pm \sqrt{r^2 - (x_c - x)^2}$$

Persamaan Lingkaran Polar Coordinates

$$x = x_c + r \cos \theta$$

$$y = y_c + r \sin \theta$$

Step size :

$$\frac{1}{r}$$

Komputasi lingkaran dapat direduksi karena lingkaran adalah bentuk yang simetris. Setiap kuadran memiliki bentuk bagian lingkaran yang sama.

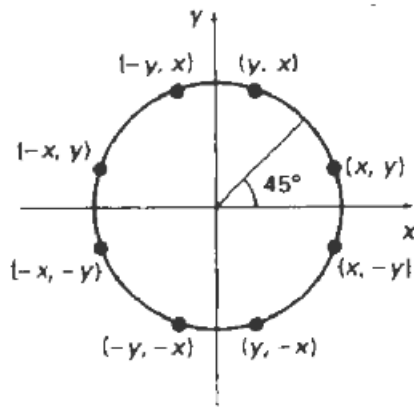


Figure 3-14
Symmetry of a circle.
Calculation of a circle point
(x, y) in one octant yields the
circle points shown for the
other seven octants.

Algoritma paling efisien berdasarkan kalkulasi incremental dari decision parameter, seperti bersenham line algorithm, yang hanya membutuhkan operasi integer sederhana. Algoritma line bersenham diadaptasi untuk pembentukan lingkaran dengan melakukan setup decision parameter untuk mencari pixel terdekat untuk mendapatkan lingkaran untuk setiap sampling step.

Metode untuk mendapatkan jarak secara langsung pada lingkaran, adalah dengan melakukan pengecekan posisi tengah dari dua pixel untuk menentukan apakah titik “midpoint” ini ada pada dalam atau luar lingkaran. Metode midpoint dapat diaplikasikan untuk bentuk-bentuk conics.

MIDPOINT CIRCLE ALGORITHM

Seperti pada line bersenham, tujuan dari algoritma untuk mendapatkan sampling titik dan menentukan pixel terdekat pada setiap step. Pada lingkaran, untuk setiap r dan titik posisi center (x_c, y_c) . Algoritma dimulai dengan mengkalkulasi posisi pixel dalam jalur lingkaran yang memiliki titik tengah origin $(0,0)$. Lalu untuk setiap posisi (x, y) yang dikalkulasi dipindahkan pada posisi yang benar dengan menambahkan x_c pada x dan y_c pada y .

Pada bagian lingkaran kuadran I mulai dari $x = 0$, $x = y$ nilai slope bervariasi dari 0 sampai -1 . Maka pergerakan unit step sesuai arah x positif dan menggunakan decision parameter untuk menentukan dua posisi yang mungkin lebih dekat dengan jalur lingkaran untuk setiap langkah.

$$f^{\circ}(x, y) = x^2 + y^2 - r^2$$

Jika point ada didalam interior lingkaran, fungsi lingkaran negative, dan sebaliknya.

$$f^{\circ}(x, y) = \begin{cases} 0, & \text{if } (x, y) \text{ is inside the boundary} \\ 0, & \text{if } (x, y) \text{ is on the boundary} \\ 0, & \text{if } (x, y) \text{ is outside the boundary} \end{cases}$$

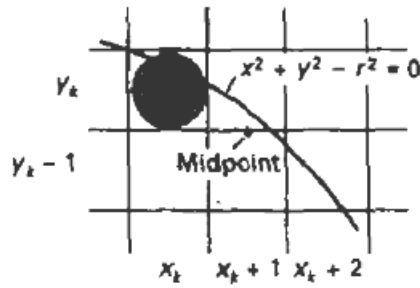


Figure 3-15
Midpoint between candidate pixels at sampling position $x_k + 1$ along a circular path.

$$p_k = f \circ (x_{k+1}, y_k - \frac{1}{2})$$

$$(x + 1)^2 + \left(y_k - \frac{1}{2}\right)^2 - r^2$$

Jika $p_k < 0$ midpoint terletak pada bagian dalam lingkaran dan y_k lebih dekat pada batas lingkaran. Sebaliknya $y_k - 1$ lebih dekat pada batas lingkaran.

Decision parameter selanjutnya didapatkan menggunakan kalkulasi incremental integer, yaitu $x_{k+1} + 1 = x_k + 2$

$$p_{k+1} = f \circ (x_{k+1} + 1, y_{k+1} - \frac{1}{2})$$

$$[(x_k + 1) + 1]^2 + \left(y_{k+1} - \frac{1}{2}\right)^2 - r^2$$

$$p_{k+1} = p_k + 2(x_k + 1) + (y_{k+1}^2 - y_k^2) - (y_{k+1} - y_k) + 1$$

Dimana, y_{k+1} antara y_k atau $y_k - 1$, bergantung pada tanda dari p_k

Evaluasi $2x_{k+1}$ dan $2y_{k+1}$ didapatkan secara inkemental menggunakan.

$$2x_{k+1} = 2x_k + 2$$

$$2y_{k+1} = 2y_k - 2$$

Pada titik awal $(0, r)$, decision parameter pertama didapatkan dengan melakukan evaluasi pada fungsi lingkaran dengan nilai $(x_0, y_0) = (0, r)$:

$$p_o = f \circ (1, r - \frac{1}{2})$$

$$1 + \left(r - \frac{1}{2}\right)^2 - r^2$$

$$\frac{5}{4} - r$$

Midpoint Circle Algorithm

1. Input radius r and circle center (x_c, y_c) , and obtain the first point on the circumference of a circle centered on the origin as

$$(x_0, y_0) = (0, r)$$

2. Calculate the initial value of the decision parameter as

$$p_0 = \frac{5}{4} - r$$

3. At each x_k position, starting at $k = 0$, perform the following test: If $p_k < 0$, the next point along the circle centered on $(0, 0)$ is (x_{k+1}, y_k) and

$$p_{k+1} = p_k + 2x_{k+1} + 1$$

Otherwise, the next point along the circle is $(x_k + 1, y_k - 1)$ and

$$p_{k+1} = p_k + 2x_{k+1} + 1 - 2y_{k+1}$$

where $2x_{k+1} = 2x_k + 2$ and $2y_{k+1} = 2y_k - 2$.

4. Determine symmetry points in the other seven octants.
5. Move each calculated pixel position (x, y) onto the circular path centered on (x_c, y_c) and plot the coordinate values:

$$x = x + x_c \quad y = y + y_c$$

6. Repeat steps 3 through 5 until $x \geq y$.

```

#include "device.h"

void circleMidpoint (int xCenter, int yCenter, int radius)
{
    int x = 0;
    int y = radius;
    int p = 1 - radius;
    void circlePlotPoints (int, int, int, int);

    /* Plot first set of points */
    circlePlotPoints (xCenter, yCenter, x, y);

    while (x < y) {
        x++;
        if (p < 0)
            p += 2 * x + 1;
        else {
            y--;
            p += 2 * (x - y) + 1;
        }
        circlePlotPoints (xCenter, yCenter, x, y);
    }
}

void circlePlotPoints (int xCenter, int yCenter, int x, int y)
{
    setPixel (xCenter + x, yCenter + y);
    setPixel (xCenter - x, yCenter + y);
    setPixel (xCenter + x, yCenter - y);
    setPixel (xCenter - x, yCenter - y);
    setPixel (xCenter + y, yCenter + x);
    setPixel (xCenter - y, yCenter + x);
    setPixel (xCenter + y, yCenter - x);
    setPixel (xCenter - y, yCenter - x);
}

```

Implementasi Circle Algorithm menggunakan C# pada bentukdasar.cs
--

Silakan lengkapi

Contoh Pemanggilan Lingkaran yang akan digunakan pada karya I

ELLIPS

Untuk menggambarkan sebuah ellips, kita dapat mengadopsi pola penggambaran lingkaran. Ellips bisa juga disebut lingkaran yang pipih, dengan modifikasi lingkaran yang memiliki dimensi vertikal dan horizontal yang berbeda atau disebut major dan minor axes.

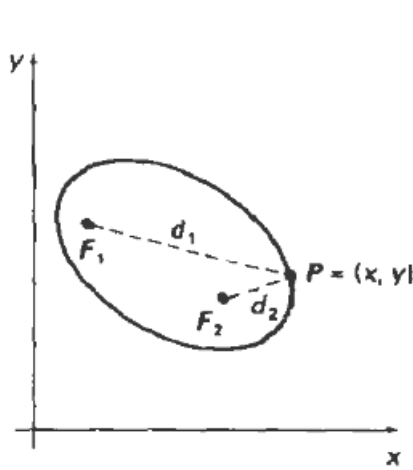


Figure 3-17
Ellipse generated about foci F_1 and F_2 .

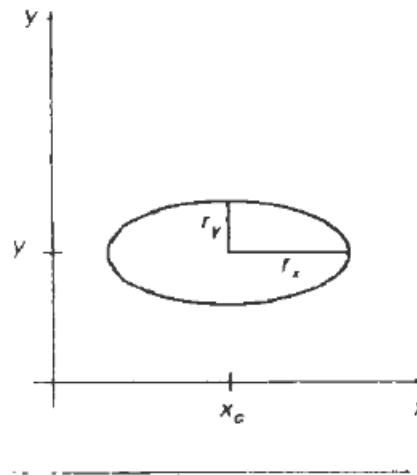


Figure 3-18
Ellipse centered at (x_c, y_c) with semimajor axis r_x and semiminor axis r_y .

Ellips dapat didefinisikan sebagai Kumpulan dari titik yang sedemikian hingga jumlah dari jarak antara dua titik tetap (foci atau fixed position) adalah sama untuk seluruh titik. Jika jarak dari dua foci dari sebuah titik $P = (x, y)$ pada sebuah ellips diberi label d_1 dan d_2 , maka persamaan ellips dapat dinyatakan sebagai:

$$d_1 + d_2 = \text{constant}$$

Eksprsi dari jarak d_1 dan d_2 dalam foci $F_1 = (x_1, y_1)$ dan $F_2 = (x_2, y_2)$ maka didapatkan :

$$\sqrt{((x - x_1)^2 + (y - y_1)^2)} + \sqrt{((x - x_2)^2 + (y - y_2)^2)} = \text{constant}$$

Persamaan ellips lain ketika posisi mayor axes dan minor axes pada posisi standar dapat dinyatakan sebagai:

$$\left(\frac{x - x_c}{r_x}\right)^2 + \left(\frac{y - y_c}{r_y}\right)^2 = 1$$

Dengan menggunakan koordinat polar ellips pada posisi standar dapat dinyatakan dengan:

$$x = x_c + r_x \cos \theta$$

$$y = y_c + r_y \sin \theta$$

MIDPOINT ELLIPS ALGORITHM

Seperti pada penggambaran lingkaran, Proses sampling pada satu unit koordinat (x atau y) dan menentukan nilai integer terdekat yang sesuai dengan jalur garis dari koordinat lain.

Metode midpoint ellips diaplikasikan pada quadran I untuk dua bagian yaitu Region I (Mayor axes) dan Region II (Minor Axis)

Steps atau iterasi dilakukan pada arah x jika $m < 1$ sebaliknya iterasi dilakukan pada arah y jika $m > 1$. Titik awal pada posisi $(0, r_y)$ dan arah mengikuti jarum jam atau (CW) pada kuadran I ellips. Perubahan unit step dari x ke unit step y

ketika $m < -1$. Lalu, mirip dengan lingkaran ada property simetris pada penggambaran ellips, secara parallel untuk dua region.

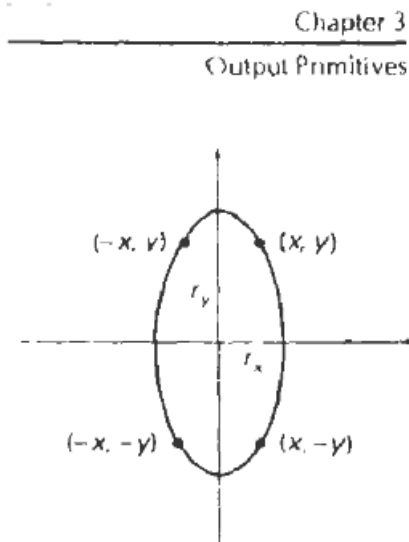


Figure 3-19
Symmetry of an ellipse
Calculation of a point (x, y)
in one quadrant yields the
ellipse points shown for the
other three quadrants.

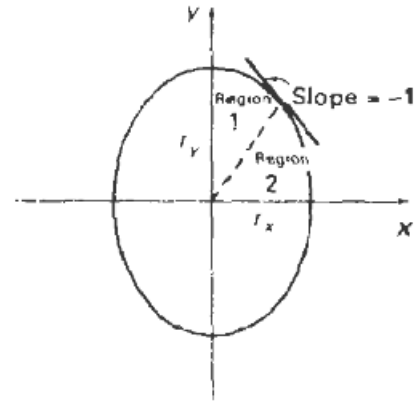


Figure 3-20
Ellipse processing regions.
Over region 1, the magnitude
of the ellipse slope is less
than 1; over region 2, the
magnitude of the slope is
greater than 1.

Didefinisikan fungsi ellips pada sebuah titik $(x_c, y_c) = (0,0)$ sebagai berikut:

$$f_{\text{ellipse}}(x, y) = r_y^2 x^2 + r_x^2 y^2 - r_x^2 r_y^2$$

Yang mengikuti syarat-syarat sebagai berikut:

$$f_{\text{ellipse}}(x, y) \begin{cases} 0, \text{ if } (x, y) \text{ is inside the ellipse boundary} \\ 0, \text{ if } (x, y) \text{ is on the ellipse boundary} \\ 0, \text{ if } (x, y) \text{ is outside the ellipse boundary} \end{cases}$$

fungsi ellipse berfungsi sebagai decision parameter pada algoritma midpoint. Untuk setiap posisi sampling, pemilihan pixel selanjutnya pada jalur ellips tergantung pada kondisi dari fungsi ellips yang dievaluasi pada midpoint untuk dua kandidat pixel.

Ellipse slope atau m didapatkan dari :

$$\frac{dy}{dx} = \frac{-2r_y^2 x}{2r_x^2 y}$$

Pada boundary antara region 1 dan region 2, nilai dari $\frac{dy}{dx} = -1$ dan $2r_y^2 x = 2r_x^2 y$

Maka, pergantian region terjadi saat

Figure 3-21 is a graph illustrating the midpoint between candidate pixels at sampling position $x_k + 1$ along an elliptical path. The graph shows a grid with horizontal and vertical lines. The horizontal axis is labeled x_k and $x_k + 1$. The vertical axis is labeled y_k and $y_k - 1$. A cluster of pixels is shown at the intersection of x_k and y_k . A point labeled "midpoint" is located at the intersection of $x_k + 1$ and $y_k - 1$. An elliptical path is drawn, passing through the cluster of pixels and the midpoint. The equation of the path is given as $r_y^2 x^2 + r_x^2 y^2 - r_x^2 r_y^2 = 0$.

$$\begin{aligned} p1_k &= f_{ellipse}\left(x_k + 1, y_k - \frac{1}{2}\right) \\ r_y^2(x_k + 1)^2 + r_x^2\left(y_k - \frac{1}{2}\right) - r_x^2 r_y^2 \\ y_{k+1} &= \begin{cases} jikap1_k < 0, y_k \\ jikap1_k > 0, y_k - 1 \end{cases} \\ p1_{k+1} &= f_{ellipse}\left(x_{k+1} + 1, y_{k+1} - \frac{1}{2}\right) \\ r_y^2[(x_k + 1) + 1]^2 + r_x^2\left(y_{k+1} - \frac{1}{2}\right)^2 - r_x^2 r_y^2 \end{aligned}$$
$$p1_{k+1} = p1_k + 2r_y^2(x_k + 1) + r_y^2 + r_x^2 \left[\left(y_{k+1} - \frac{1}{2} \right)^2 - \left(y_k - \frac{1}{2} \right)^2 \right]$$
$$increrement = \begin{cases} 2r_y^2 x_{k+1} + r_y^2, jikap1_k < 0 \\ 2r_y^2 x_{k+1} + r_y^2 - 2r_x^2 y_{k+1}, jikap1_k > 0 \end{cases}$$
$$p1_0 = f_{ellipse} \left(1, r_y - \frac{1}{2} \right)$$

$$r_y^2 + r_x^2 \left(r_y - \frac{1}{2}\right)^2 - r_x^2 r_y^2$$

Atau

$$p1_0 = r_y^2 - r_x^2 r_y^2 + \frac{1}{4} r_x^2$$

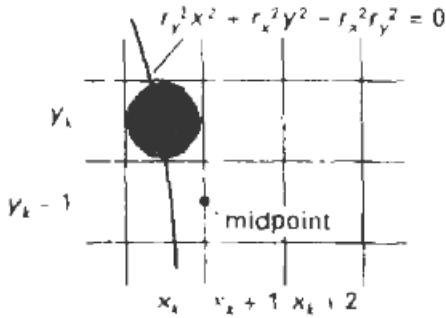


Figure 3-22
Midpoint between candidate pixels at sampling position $y_k - 1$ along an elliptical path.

Untuk region 2, sampling unit step terjadi pada arah y negatif, dan midpoint diambil diantara dua titik kandidat horizontal pixel untuk setiap step. Maka decision parameter dievaluasi sbb:

$$p2_k = f_{ellipse} \left(x_k + \frac{1}{2}, y_k - 1 \right)$$

$$r_y^2 \left(x_k + \frac{1}{2} \right)^2 + r_x^2 (y_k - 1)^2 - r_x^2 r_y^2$$

$$x_{k+1} = \begin{cases} \text{jika } p2_k > 0, x_k \\ \text{jika } p2_k \leq 0, x_{k+1} \end{cases}$$

$$p2_{k+1} = f_{ellipse} \left(x_{k+1} + \frac{1}{2}, y_{k+1} - 1 \right)$$

$$r_y^2 \left[x_{k+1} + \frac{1}{2} \right]^2 + r_x^2 [(y_k - 1) - 1]^2 - r_x^2 r_y^2$$

Atau

$$p2_{k+1} = p2_k + 2r_x^2 (y_k - 1) + r_x^2 + r_y^2 \left[\left(x_{k+1} + \frac{1}{2} \right)^2 - \left(x_k + \frac{1}{2} \right)^2 \right]$$

Para region 2, nilai pertama adalah (x_0, y_0) diambil berdasarkan posisi terakhir dari region 1 dan decision parameter pertama region ke 2 adalah.

$$p2_0 = f_{ellipse} \left(x_0 + \frac{1}{2}, y_0 - 1 \right)$$

$$r_y^2 \left(x_0 + \frac{1}{2} \right)^2 + r_x^2 (y_0 - 1)^2 - r_x^2 r_y^2$$

Untuk menyederhanakan perhitungan, titik pertama dapat diambil pada $(r_x, 0)$. Unit step akan berjalan dengan arah y positif sampai dengan posisi terakhir dari Region 1.

Midpoint Ellipse Algorithm

1. Input r_x , r_y , and ellipse center (x_c, y_c) , and obtain the first point on an ellipse centered on the origin as

$$(x_0, y_0) = (0, r_y)$$

2. Calculate the initial value of the decision parameter in region 1 as

$$p1_0 = r_y^2 - r_x^2 r_y + \frac{1}{4} r_x^2$$

3. At each x_k position in region 1, starting at $k = 0$, perform the following test: If $p1_k < 0$, the next point along the ellipse centered on $(0, 0)$ is (x_{k+1}, y_k) and

$$p1_{k+1} = p1_k + 2r_y^2 x_{k+1} + r_y^2$$

Otherwise, the next point along the circle is $(x_k + 1, y_k - 1)$ and

$$p1_{k+1} = p1_k + 2r_y^2 x_{k+1} - 2r_x^2 y_{k+1} + r_y^2$$

with

$$2r_y^2 x_{k+1} = 2r_y^2 x_k + 2r_y^2, \quad 2r_x^2 y_{k+1} = 2r_x^2 y_k - 2r_x^2$$

and continue until $2r_y^2 x \geq 2r_x^2 y$.

and continue until $2r_y^2x = 2r_x^2y$.

4. Calculate the initial value of the decision parameter in region 2 using the last point (x_0, y_0) calculated in region 1 as

$$p2_0 = r_y^2 \left(x_0 + \frac{1}{2} \right)^2 + r_x^2 (y_0 - 1)^2 - r_x^2 r_y^2$$

5. At each y_k position in region 2, starting at $k = 0$, perform the following test: If $p2_k > 0$, the next point along the ellipse centered on $(0, 0)$ is $(x_k, y_k - 1)$ and

$$p2_{k+1} = p2_k - 2r_x^2 y_{k+1} + r_x^2$$

Otherwise, the next point along the circle is $(x_k + 1, y_k - 1)$ and

$$p2_{k+1} = p2_k + 2r_y^2 x_{k+1} - 2r_x^2 y_{k+1} + r_x^2$$

using the same incremental calculations for x and y as in region 1.

6. Determine symmetry points in the other three quadrants.
7. Move each calculated pixel position (x, y) onto the elliptical path centered on (x_c, y_c) and plot the coordinate values:

$$x = x + x_c \quad y = y + y_c$$

8. Repeat the steps for region 1 until $2r_y^2x \geq 2r_x^2y$.

```

#include "device.h"

#define ROUND(a) ((int)(a+0.5))

void ellipseMidpoint (int xCenter, int yCenter, int Rx, int Ry)
{
    int Rx2 = Rx*Rx;
    int Ry2 = Ry*Ry;
    int twoRx2 = 2*Rx2;
    int twoRy2 = 2*Ry2;
    int p;
    int x = 0;
    int y = Ry;
    int px = 0;
    int py = twoRx2 * y;
    void ellipsePlotPoints (int, int, int, int);

    /* Plot the first set of points */
    ellipsePlotPoints (xCenter, yCenter, x, y);

    /* Region 1 */
    p = ROUND (Ry2 - (Rx2 * Ry) + (0.25 * Rx2));
    while (px < py) {
        x++;
        px += twoRy2;
        if (p < 0)
            p += Ry2 + px;
        else {
            y--;
            py -= twoRx2;
            p += Ry2 + px - py;
        }
        ellipsePlotPoints (xCenter, yCenter, x, y);
    }

    /* Region 2 */
    p = ROUND (Ry2*(x+0.5)*(x+0.5) + Rx2*(y-1)*(y-1) - Rx2*Ry2);
    while (y > 0) {
        y--;
        py -= twoRx2;
        if (p > 0)
            p += Rx2 - py;
        else {
            x++;
            px += twoRy2;
            p += Rx2 - py + px;
        }
    }
}

```

```

    }
    ellipsePlotPoints (xCenter, yCenter, x, y);
}

void ellipsePlotPoints (int xCenter, int yCenter, int x, int y)
{
    setPixel (xCenter + x, yCenter + y);
    setPixel (xCenter - x, yCenter + y);
    setPixel (xCenter + x, yCenter - y);
    setPixel (xCenter - x, yCenter - y);
}

```

Implementasi Ellips Algorithm pada C# pada bentukdasar.cs

Silakan lengkapi

Contoh Pemanggilan Ellips pada karya I

TASK PRAKTIKUM

TASK 1: IMPLEMENTASI MIDPOINT LINGKARAN DAN MIDPOINT GARIS DAN LENGKAPI BENTUK DASAR

1. Lanjutkan kode minggu lalu dan rename menjadi [KG2025_2A_D4_2023]_Modul3_000, modifikasi scene karya I dan karya 2 sesuai dengan tugas modul 3.
2. Amati Algoritma Midpoint untuk menggambar lingkaran dan ellips
3. Implementasi Algoritma tersebut dan buatlah fungsi lingkaran / circle dan ellips / ellipse pada class bentukdasar.cs
4. Karya 2 pada pertemuan minggu lalu dipindahkan menjadi karya I, selain Persegi, Persegi Panjang, Segitiga Siku-Siku, dan Trapesium Siku-Siku, tambahkan Lingkaran dan Ellips di sembarang kuadran

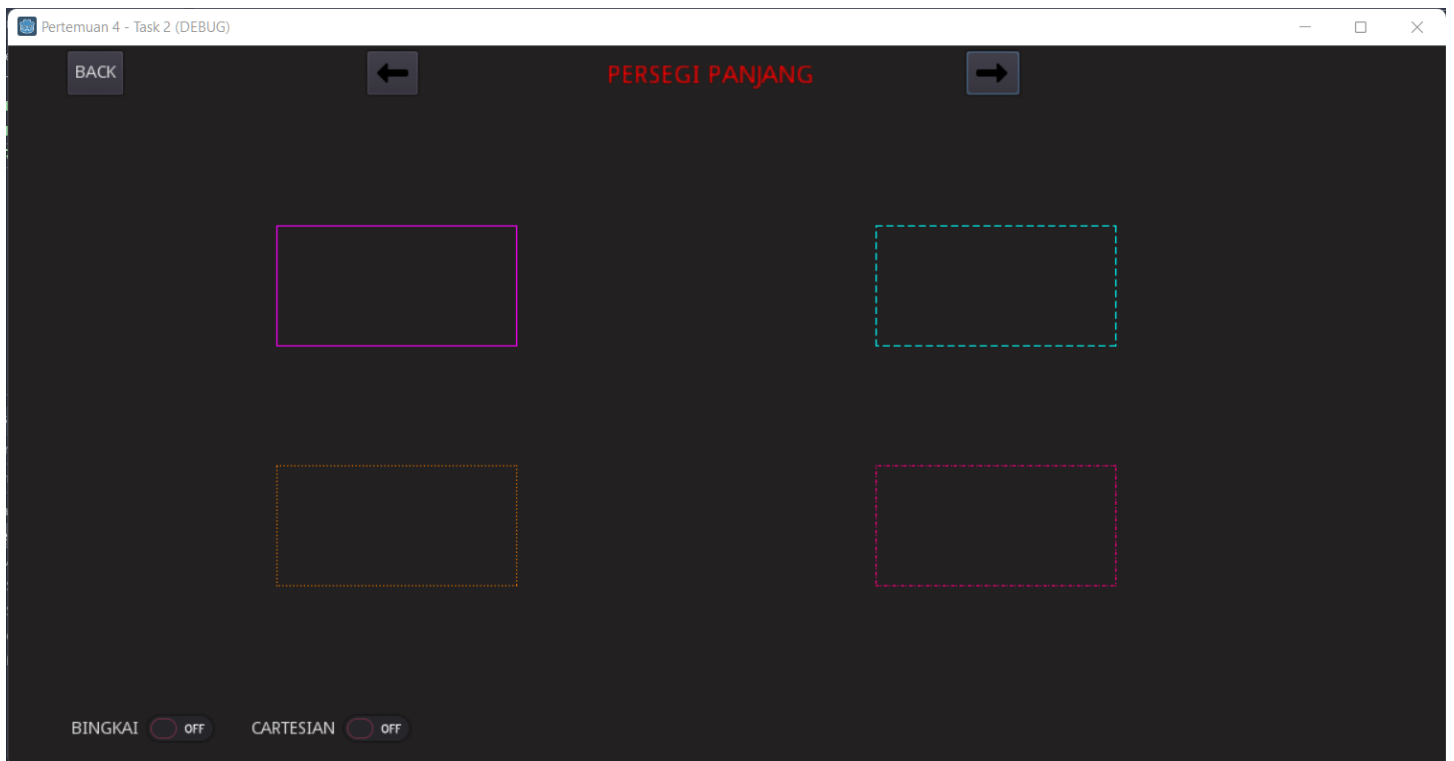
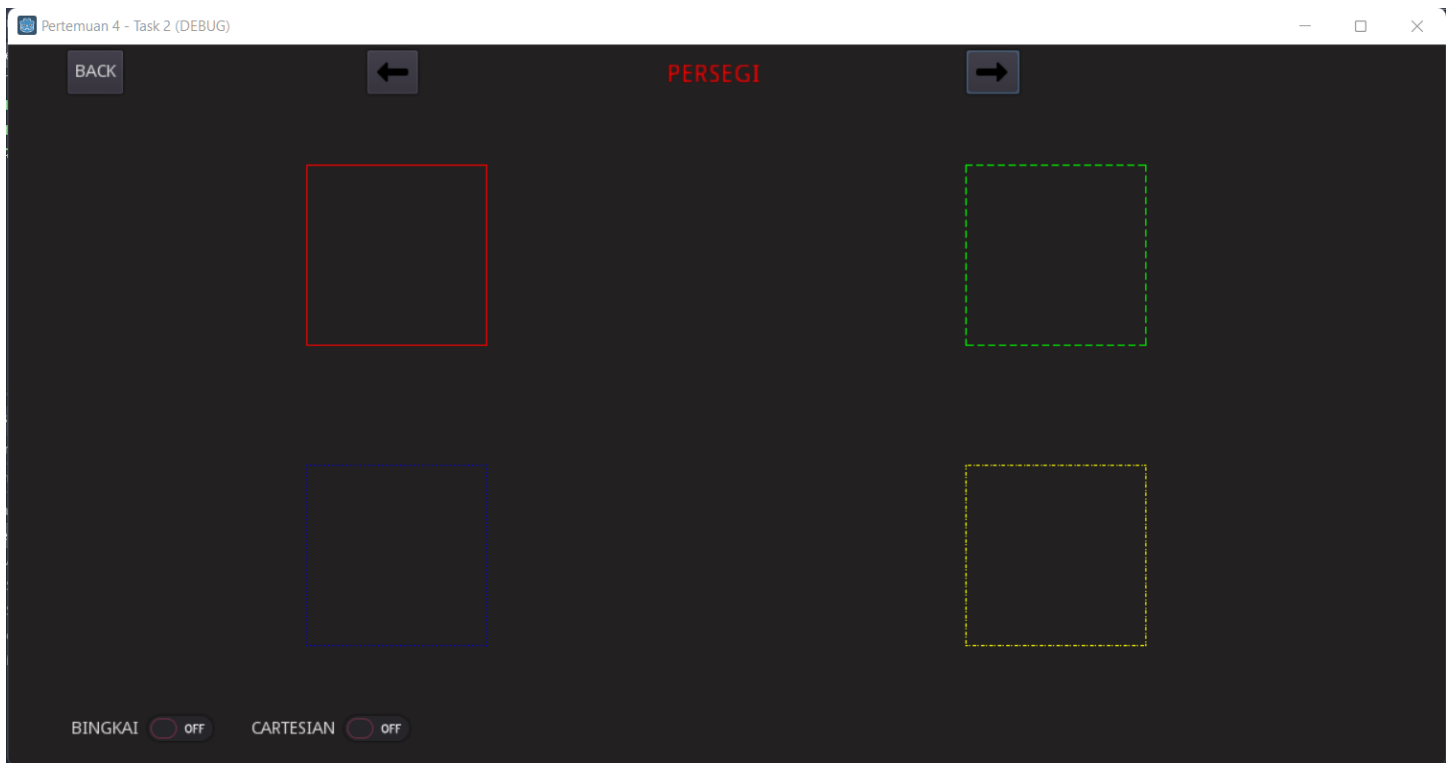
Lesson Learnt (Code, Print Screen Hasil Karya, dan Komentar)

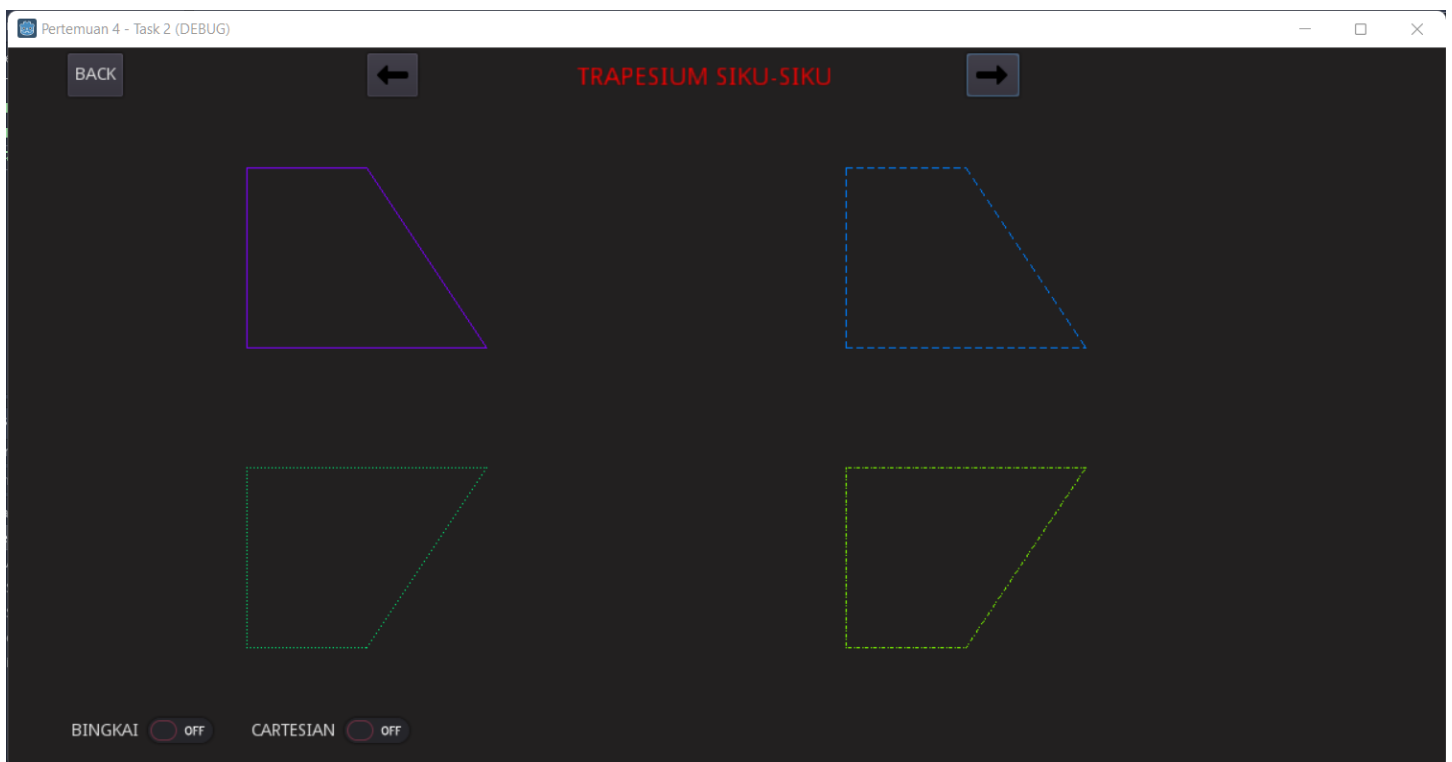
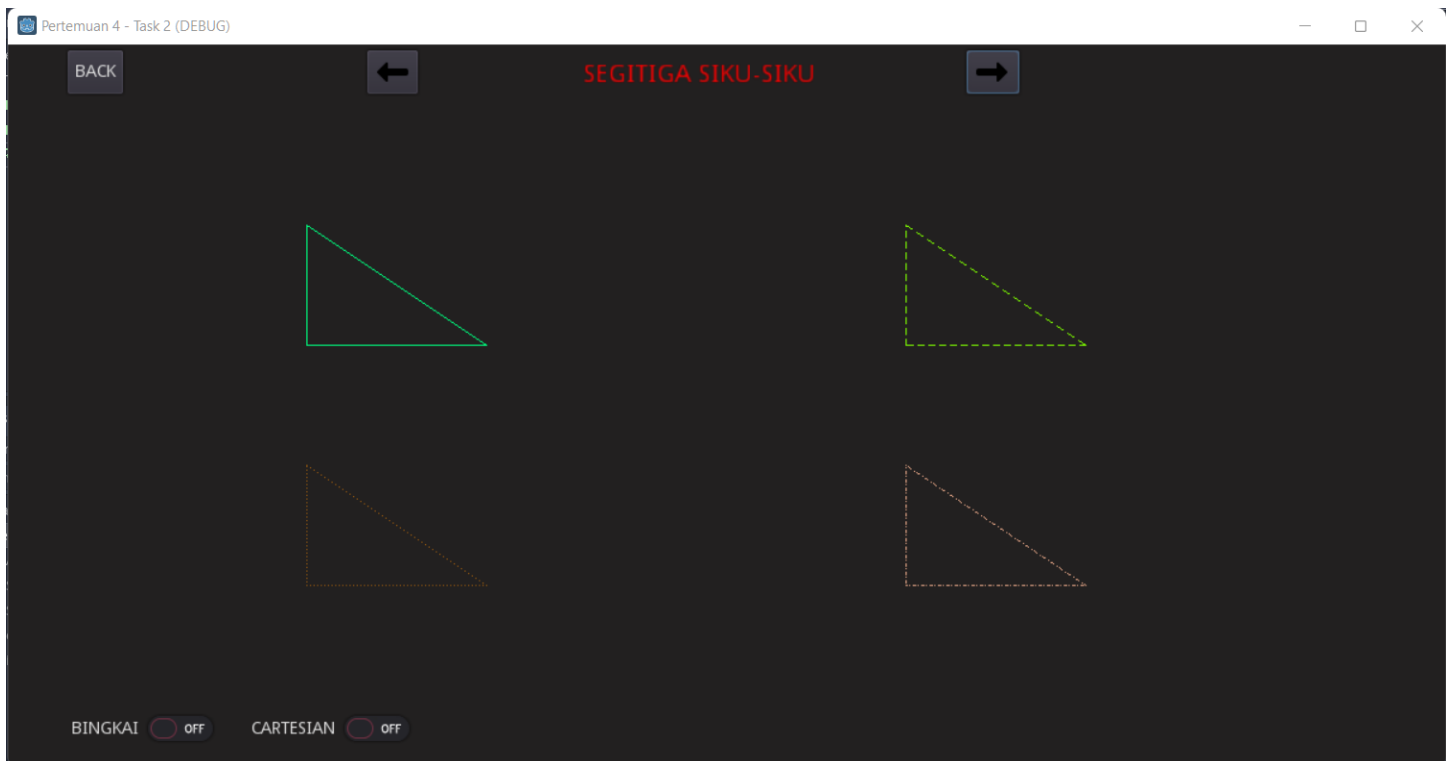
TASK 2: MODIFIKASI GARIS

1. Buatlah berbagai macam attribute garis seperti titik-titik, titik-garis-titik, garis-garis dan lain-lain lalu buatlah scene karya2 seperti berikut



2. Buatlah Scene baru, karya3 yang merupakan Modifikasi Bentuk Dasar Persegi, Persegi Panjang, Segitiga Siku-Siku, dan Trapesium Siku-Siku, Lingkaran dan Ellips yang attribute garisnya sesuai tipe yang diinginkan, normal, titik-titik, titik-garis-titik, garis-garis.
3. Contoh Scene sbb:





Lesson Learnt (Code, Print Screen Hasil Karya, dan Komentar)

PENGUMPULAN

Ikuti Format yang diberikan di Google Classroom.