

TUGAS INDIVIDU GRAFIKA KOMPUTER

“Implementasi Algoritma Bresenham dan Midpoint untuk Menggambar Lingkaran ”

Disusun untuk Memenuhi Tugas MataKuliah Grafika Komputer

Dosen Pengampu :

- 1) Febi Eka Febriansyah, M.T.
- 2) Wartariyus, S.Kom., M.T.I.
- 3) Putut Aji Nalendro, S.Pd., M.Pd.



Disusun Oleh:

Nama : Novilia Azizah

NPM : 2413025044

Kelas : PTI 24B

**PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
JURUSAN PENDIDIKAN MATEMATIKA DAN ILMU PENGETAHUAN ALAM
FAKULTAS KEGURUAN DAN ILMU PENDIDIKAN
UNIVERSITAS LAMPUNG**

2025

I. Pendahuluan

Algoritma Bresenham dan Midpoint merupakan dua metode yang digunakan untuk menggambar lingkaran pada layar berbasis piksel secara efisien. Keduanya menggunakan pendekatan perhitungan bilangan bulat, sehingga lebih cepat dibandingkan metode yang menggunakan operasi trigonometri atau akar kuadrat. Algoritma Bresenham memanfaatkan simetri delapan bagian lingkaran untuk menentukan titik-titik piksel, sedangkan algoritma Midpoint menggunakan titik tengah sebagai acuan untuk memilih piksel terdekat dengan jalur lingkaran.

Kedua algoritma ini memiliki keunggulan dalam hal kecepatan dan akurasi dalam menggambar lingkaran. Bresenham lebih sederhana secara logika, sementara Midpoint memberikan hasil yang lebih halus pada beberapa kasus. Keduanya sering digunakan dalam pengembangan grafika komputer dan menjadi dasar dalam pembelajaran algoritma rasterisasi bentuk dua dimensi.

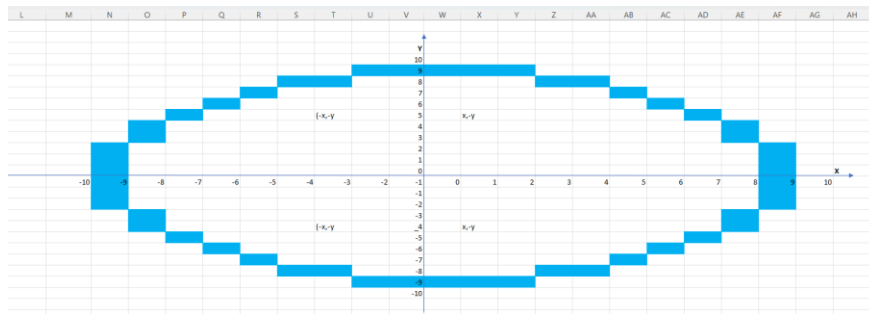
II. Pembahasan

A. Lingkaran Bresenham

1. Tabel

NAMA : NOVILIA AZIZAH							
NPM : 2413025044							
TUGAS TABEL LINGKARAN BRASENHAM							
X	Y	d	x_0+x, y_0+y	x_0-x, y_0-y	x_0+x, y_0-y	x_0+y, y_0+x	x_0-y, y_0-x
0	9	-15	10+0, 10+5	10-0, 10+5	10+0, 10-5	10+5, 10+0	10-5, 10-0
1	9	-5	10+1, 10+9	10-1, 10+9	10+1, 10-9	10+9, 10+1	10-9, 10-1
2	9	9	10+2, 10+9	10-2, 10+9	10+2, 10-9	10+9, 10+2	10-9, 10-2
3	8	-1	10+3, 10+8	10-3, 10+8	10+3, 10-8	10+8, 10+3	10-8, 10-3
4	8	21	10+4, 10+8	10-4, 10+8	10+4, 10-8	10+8, 10+4	10-8, 10-4
5	7	23	10+5, 10+7	10-5, 10+7	10+5, 10-7	10+7, 10+5	10-7, 10-5
6	6	33	10+6, 10+6	10-6, 10+6	10+6, 10-6	10+6, 10+6	10-6, 10-6
6	6						
7	5						
8	4						
8	3						
int x-0		$d=3-2*r$			jika $d < 0$ maka y tetap		
int y-0					$d = d+4*x+6$		
					x selalu inc ++		
					jika $d > 0$ maka y--		
					$d = d+4*(x-y)+10$		

2. Diagram



B. Lingkaran Midpoint

1. Tabel

NOVILIA AZIZAH NPM : 2413025044

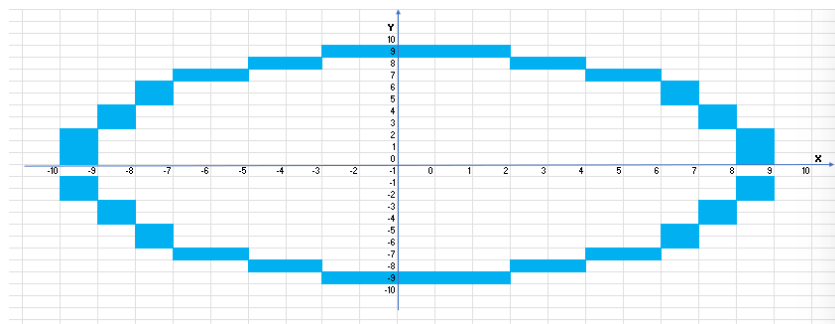
TUGAS TABEL LINGKARAN MINPOINT

X	Y	P	x_0+x, y_0+y	x_0-x, y_0+y	x_0+x, y_0-y	x_0-x, y_0-y	x_0+y, y_0+x	x_0-y, y_0+x	x_0+y, y_0-x	x_0-y, y_0-x
0	9	-8	10+0, 10+5	10-0, 10+5	10+0, 10-5	10-0, 10-5	10+5, 10+0	10-5, 10+0	10+5, 10-0	10-5, 10-0
1	9	-5	10+1, 10+9	10-1, 10+9	10+1, 10-9	10-1, 10-9	10+1, 10+9	10-9, 10+1	10+9, 10-1	10-9, 10-1
2	9	0	10+2, 10+9	10-2, 10+9	10+2, 10-9	10-2, 10-9	10+9, 10+2	10-9, 10+2	10+9, 10-2	10-9, 10-2
3	8	-9	10+3, 10+8	10-3, 10+8	10+3, 10-8	10-3, 10-8	10+8, 10+3	10-8, 10+3	10+8, 10-3	10-8, 10-3
4	8	0	10+4, 10+8	10-4, 10+8	10+4, 10-8	10-4, 10-8	10+8, 10+4	10-8, 10+4	10+8, 10-4	10-8, 10-4
5	7	-3	10+5, 10+7	10-5, 10+7	10+5, 10-7	10-5, 10-7	10+7, 10+5	10-7, 10+5	10+7, 10-5	10-7, 10-5
6	7	10	10+6, 10+7	10-6, 10+7	10+6, 10-7	10-6, 10-7	10+7, 10+6	10-7, 10+6	10+7, 10-6	10-7, 10-6
7	6	13	10+7, 10+6	10-7, 10+6	10+7, 10-6	10-7, 10-6	10+6, 10+7	10-6, 10+7	10+6, 10-7	10-6, 10-7
6	7									
7	6									
7	5									
8	4									
8	3									
9	2									
9	1									

jika $p < 0$ maka y tetap
 $p = p + 2x + 1$

jika $p > 0$ maka y--
 $p = p + 2x - (x - y) + 1$

2. Diagram



C. Kode Program

```
Lingkaran > Untitled-2.html > html > body > script > titik
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8" />
5   <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
6   <title>Novilia azizah_2413025044</title>
7   <style>
8     canvas {
9       border: 1px solid #000000;
10      margin: 10px;
11    }
12  </style>
13 </head>
14 <body>
15   <h1 align="center">LINGKARAN BRESENHAM DAN MIDPOINT</h1>
16
17   <label>X: <input type="number" id="x" value="150"></label>
18   <label>Y: <input type="number" id="y" value="150"></label>
19   <label>Radius: <input type="number" id="rad" value="50"></label>
20   <label>Warna: <input type="color" id="warna" value="#E4AB16"></label>
21   <button onclick="buatGambar()">Gambar Lingkaran</button>
22
23   <br/><br/>
24   <canvas id="myCanvas" width="300" height="300"></canvas>
25   <canvas id="midpoint" width="300" height="300"></canvas>
26
27   <script>
28     let canvas = document.getElementById("myCanvas");
29     let ctx = canvas.getContext("2d");
30
31     function titik(x, y, warna) {
32       ctx.fillStyle = warna;
33       ctx.fillRect(x, y, 1, 1);
34     }
```

```
34   }
35
36   function gambarTitikSimetris(x0, y0, x, y, warna) {
37     titik(x0 + x, y0 + y, warna);
38     titik(x0 - x, y0 + y, warna);
39     titik(x0 + x, y0 - y, warna);
40     titik(x0 - x, y0 - y, warna);
41     titik(x0 + y, y0 + x, warna);
42     titik(x0 - y, y0 + x, warna);
43     titik(x0 + y, y0 - x, warna);
44     titik(x0 - y, y0 - x, warna);
45   }
46
47   function linkBre(x0, y0, r, warna) {
48     var d = 3 - 2 * r;
49     var x = 0, y = r;
50
51     while (x <= y) {
52       gambarTitikSimetris(x0, y0, x, y, warna);
53       if (d <= 0) {
54         d = d + 4 * x + 6;
55       } else {
56         d = d + 4 * (x - y) + 10;
57         y--;
58       }
59       x++;
60     }
61   }
62
63   function buatGambar() {
```

```

63 function buatGambar() {
64   ctx.clearRect(0, 0, canvas.width, canvas.height);
65   ctxmidpoint.clearRect(0, 0, midpoint.width, midpoint.height);
66
67   let x0 = parseInt(document.getElementById("x").value);
68   let y0 = parseInt(document.getElementById("y").value);
69   let r = parseInt(document.getElementById("rad").value);
70   let warna = document.getElementById("warna").value;
71
72   linkBre(x0, y0, r, warna);
73   drawmidpoint(x0, y0, r);
74 }
75
76 let midpoint = document.getElementById("midpoint");
77 let ctxmidpoint = midpoint.getContext("2d");
78
79 function drawmidpoint(x0, y0, r) {
80   let x = r;
81   let y = 0;
82   let d = r - 1;
83
84   while (x >= y) {
85     ctxmidpoint.fillRect(x0 + x, y0 + y, 1, 1);
86     ctxmidpoint.fillRect(x0 - x, y0 + y, 1, 1);
87     ctxmidpoint.fillRect(x0 + x, y0 - y, 1, 1);
88     ctxmidpoint.fillRect(x0 - x, y0 - y, 1, 1);
89     ctxmidpoint.fillRect(x0 + y, y0 + x, 1, 1);
90     ctxmidpoint.fillRect(x0 - y, y0 + x, 1, 1);
91     ctxmidpoint.fillRect(x0 + y, y0 - x, 1, 1);
92     ctxmidpoint.fillRect(x0 - y, y0 - x, 1, 1);

```

```

41   titik(x0 + y, y0 + x, warna);
42   titik(x0 - y, y0 + x, warna);
43   titik(x0 + y, y0 - x, warna);
44   titik(x0 - y, y0 - x, warna);
45 }
46
47 function linkBre(x0, y0, r, warna) {
48   var d = 3 - 2 * r;
49   var x = 0, y = r;
50
51   while (x <= y) {
52     gambarTitikSimetris(x0, y0, x, y, warna);
53     if (d <= 0) {
54       d = d + 4 * x + 6;
55     } else {
56       d = d + 4 * (x - y) + 10;
57       y--;
58     }
59     x++;
60   }
61 }
62
63 function buatGambar() {
64   ctx.clearRect(0, 0, canvas.width, canvas.height);
65   ctxmidpoint.clearRect(0, 0, midpoint.width, midpoint.height);
66
67   let x0 = parseInt(document.getElementById("x").value);
68   let y0 = parseInt(document.getElementById("y").value);
69   let r = parseInt(document.getElementById("rad").value);

```

```

91   ctxmidpoint.fillRect(x0 + y, y0 -
92   ctxmidpoint.fillRect(x0 - y, y0 -
93
94   if (d >= 2 * y) {
95     d -= 2 * y + 1;
96     y++;
97   } else if (d < 2 * (r - x)) {
98     d += 2 * x - 1;
99     x--;
100   } else {
101     d += 2 * (x - y - 1);
102     x--;
103     y++;
104   }
105 }
106 }
107 </script>
108 </body>
109 </html>
110

```

D. Output Kode Program



III. Kesimpulan

Algoritma Bresenham dan Midpoint merupakan dua metode efektif untuk menggambar lingkaran dalam grafika komputer. Keduanya bekerja dengan prinsip perhitungan bilangan bulat dan memanfaatkan simetri, sehingga menghasilkan proses yang cepat, efisien, dan akurat tanpa menggunakan operasi matematika kompleks.

Meskipun memiliki pendekatan yang sedikit berbeda, kedua algoritma sama-sama mampu menghasilkan visualisasi lingkaran yang baik pada layar piksel. Oleh karena itu, pemahaman dan penerapan kedua algoritma ini sangat penting dalam pengembangan aplikasi grafis maupun pembelajaran dasar komputer grafik.