

Tugas 3 Grafika Komputer

Algoritma Pembentukan Lingkaran

Nama : Angga Bagus Susilo

Kelas : CD

NPM : 2217051064

1. Bresenham Algorithm

a) Source Code

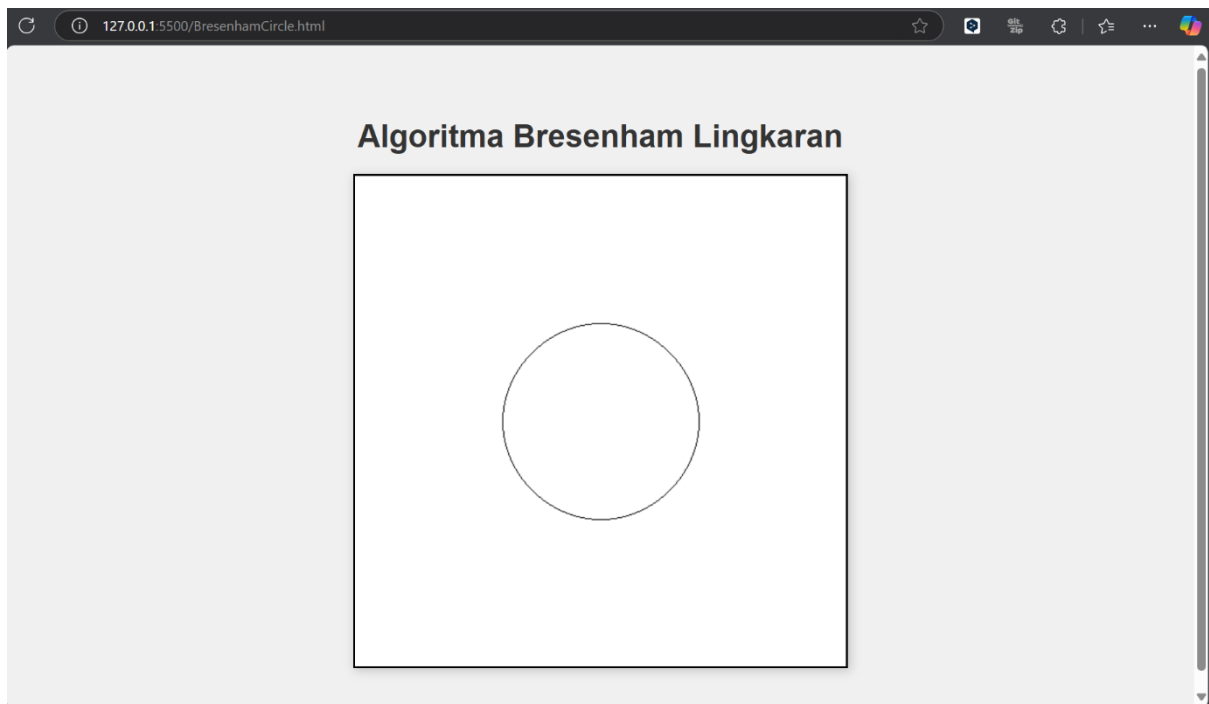
```
<> BresenhamCircle.html X
<> BresenhamCircle.html > ...
1  <!DOCTYPE html>
2  <html lang="id">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Algoritma Bresenham Lingkaran</title>
7      <style>
8          body {
9              display: flex;
10             justify-content: center;
11             align-items: center;
12             height: 100vh;
13             background-color: #f0f0f0;
14             font-family: Arial, sans-serif;
15         }
16         .container {
17             text-align: center;
18         }
19         canvas {
20             border: 2px solid black;
21             background-color: white;
22             box-shadow: 2px 2px 10px rgba(0, 0, 0, 0.2);
23         }
24         h1 {
25             margin-bottom: 20px;
26             color: #333;
27         }
28     </style>
29 </head>
```

```

30 <body>
31   <div class="container">
32     <h1>Algoritma Bresenham Lingkaran</h1>
33     <canvas id="canvas" width="500" height="500"></canvas>
34   </div>
35   <script>
36     function drawPixel(ctx, x, y) {
37       ctx.fillRect(x, y, 1, 1);
38     }
39
40     function circlePlotPoints(ctx, x0, y0, x, y) {
41       drawPixel(ctx, x0 + x, y0 + y);
42       drawPixel(ctx, x0 - x, y0 + y);
43       drawPixel(ctx, x0 + x, y0 - y);
44       drawPixel(ctx, x0 - x, y0 - y);
45       drawPixel(ctx, x0 + y, y0 + x);
46       drawPixel(ctx, x0 - y, y0 + x);
47       drawPixel(ctx, x0 + y, y0 - x);
48       drawPixel(ctx, x0 - y, y0 - x);
49     }
50
51     function drawCircleBresenham(x0, y0, r) {
52       const canvas = document.getElementById("canvas");
53       const ctx = canvas.getContext("2d");
54       ctx.fillStyle = "black";
55
56       let x = 0;
57       let y = r;
58       let d = 3 - 2 * r;
59
60       circlePlotPoints(ctx, x0, y0, x, y);
61
62       while (y >= x) {
63         x++;
64         if (d > 0) {
65           y--;
66           d = d + 4 * (x - y) + 10;
67         } else {
68           d = d + 4 * x + 6;
69         }
70         circlePlotPoints(ctx, x0, y0, x, y);
71       }
72     }
73
74     // Contoh penggunaan
75     drawCircleBresenham(250, 250, 100);
76   </script>
77 </body>
78 </html>

```

b) Output



2. Midpoint Algorithm

a) Source Code

```
MidPointCircle.html > ...
1 <!DOCTYPE html>
2 <html lang="id">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Algoritma Midpoint Lingkaran</title>
7   <style>
8     body {
9       display: flex;
10      justify-content: center;
11      align-items: center;
12      height: 100vh;
13      background-color: #f0f0f0;
14      font-family: Arial, sans-serif;
15    }
16    .container {
17      text-align: center;
18    }
19    canvas {
20      border: 2px solid black;
21      background-color: white;
22      box-shadow: 2px 2px 10px rgba(0, 0, 0, 0.2);
23    }
24    h1 {
25      margin-bottom: 20px;
26      color: #333;
27    }
28  </style>
29 </head>
```

```

30 <body>
31   <div class="container">
32     <h1>Algoritma Midpoint Lingkaran</h1>
33     <canvas id="canvas" width="500" height="500"></canvas>
34   </div>
35   <script>
36     function drawPixel(ctx, x, y) {
37       ctx.fillRect(x, y, 1, 1);
38     }
39
40     function circlePlotPoints(ctx, x0, y0, x, y) {
41       drawPixel(ctx, x0 + x, y0 + y);
42       drawPixel(ctx, x0 - x, y0 + y);
43       drawPixel(ctx, x0 + x, y0 - y);
44       drawPixel(ctx, x0 - x, y0 - y);
45       drawPixel(ctx, x0 + y, y0 + x);
46       drawPixel(ctx, x0 - y, y0 + x);
47       drawPixel(ctx, x0 + y, y0 - x);
48       drawPixel(ctx, x0 - y, y0 - x);
49     }
50
51     function drawCircleMidpoint(x0, y0, radius) {
52       const canvas = document.getElementById("canvas");
53       const ctx = canvas.getContext("2d");
54       ctx.fillStyle = "black";
55
56       let x = 0;
57       let y = radius;
58       let p = 1 - radius;
59
60       circlePlotPoints(ctx, x0, y0, x, y);
61
62       while (x < y) {
63         x++;
64         if (p < 0) {
65           p += 2 * x + 1;
66         } else {
67           y--;
68           p += 2 * (x - y) + 1;
69         }
70         circlePlotPoints(ctx, x0, y0, x, y);
71       }
72     }
73
74     // Contoh penggunaan
75     drawCircleMidpoint(250, 250, 100);
76   </script>
77 </body>
78 </html>

```

b) Output

