

**GRAFIKA KOMPUTER**  
**“ALGORITMA PEMBENTUKAN KURVA”**



**Dosen Pengampu:**

Febi Eka Febriansyah, M.T.

Wartariyus, S.Kom., M.T.I.

Putut Aji Nalendro, S.Pd., M.Pd.

**Disusun Oleh:**

Nama: Nabila Fatma Sari

NPM: 2453025003

Kelas: PTI 24 A

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

**2025**

## **PENGERTIAN KURVA BEZIER CUBIC**

Kurva Bezier Cubic adalah jenis kurva berparameter yang digunakan dalam grafika komputer dan bidang terkait.

Kurva bezier cubic didefinisikan oleh 4 titik kendali: P0 (titik awal), P1 dan P2 (titik kendali), dan P3 (titik akhir). Kurva ini dimulai dari P0, bergerak menuju P1, kemudian berbelok menuju P2, dan akhirnya tiba di P3. Titik P1 dan P2 tidak harus dilalui oleh kurva, mereka hanya memberikan informasi arah.

## **RUMUS MATEMATIKA**

Rumus matematika untuk Kurva Bezier cubic dinyatakan dengan persamaan parametris sebagai berikut:

$$\mathbf{B}(t) = (1-t)^3\mathbf{P}_0 + 3(1-t)^2t\mathbf{P}_1 + 3(1-t)t^2\mathbf{P}_2 + t^3\mathbf{P}_3$$

## **KURVA BEZIER CUBIC BANYAK DIGUNAKAN:**

1. Desain Grafis: Membuat kurva halus yang dapat diperbesar tanpa kehilangan resolusi.
2. Animasi: Mengatur kecepatan objek bergerak dari titik A ke titik B dengan lancar.
3. Robotik: Membuat gerakan mesin yang halus untuk mengurangi gesekan dan meningkatkan efisiensi.

## **KELEBIHAN MENGGUNAKAN KURVA BEZIER CUBIC:**

1. Fleksibilitas: Kurva bezier cubic dapat digunakan untuk menciptakan berbagai bentuk kurva, dari yang sederhana hingga yang kompleks.
2. Mudah dikombinasikan: Kurva bezier cubic dapat mudah dikombinasikan untuk menciptakan bentuk yang lebih kompleks, memungkinkan pengguna untuk membuat desain yang lebih dinamis dan menarik.
3. Penggunaan yang luas: Kurva bezier cubic banyak digunakan dalam berbagai bidang seperti desain grafis, animasi, robotik, CAD (Computer-Aided Design), sehingga memudahkan integrasi dengan berbagai alat dan perangkat lunak.

4. Kurva halus: Kurva bezier cubic dapat menghasilkan kurva yang sangat halus dan mulus, bahkan ketika diperbesar. Ini membuat ideal untuk aplikasi desain grafis dan animasi.

### CONTOH SEDERHANA DALAM JAVASCRIPT

```
function cubicBezier(t, P0, P1, P2, P3) {  
    const x = Math.pow(1 - t, 3) * P0.x +  
        3 * Math.pow(1 - t, 2) * t * P1.x +  
        3 * (1 - t) * Math.pow(t, 2) * P2.x +  
        Math.pow(t, 3) * P3.x;  
  
    const y = Math.pow(1 - t, 3) * P0.y +  
        3 * Math.pow(1 - t, 2) * t * P1.y +  
        3 * (1 - t) * Math.pow(t, 2) * P2.y +  
        Math.pow(t, 3) * P3.y;  
  
    return {x, y};  
}  
  
const P0 = {x: 10, y: 100};  
const P1 = {x: 100, y: 0};  
const P2 = {x: 200, y: 200};  
const P3 = {x: 300, y: 100};  
  
const point = cubicBezier(0.5, P0, P1, P2, P3);  
console.log(point);
```

## CODE PROGRAM PENGGUNAAN KURVA BEZIER CUBIC

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-
scale=1.0">

    <title>Kurva Bezier Cubic - Dua Awan dan Pohon</title>

</head>

<body>

    <h1 align="center">MEMBUAT POHON DAN AWAN</h1>

    <canvas id="canvas" width="500" height="500" style="border: 2px
solid #000000;"></canvas>

    <script>

        const canvas = document.getElementById("canvas");

        const ctx = canvas.getContext("2d");

        // Fungsi untuk Menggambar Kurva Bezier Cubic

        function drawBezierCurve(ctx, points, color) {

            ctx.beginPath();

            ctx.moveTo(points[0].x, points[0].y);

            ctx.strokeStyle = color;

            ctx.fillStyle = color;

            for (let i = 1; i < points.length - 2; i += 3) {

                ctx.bezierCurveTo(

                    points[i].x, points[i].y,
```

```

        points[i + 1].x, points[i + 1].y,
        points[i + 2].x, points[i + 2].y
    );
}

ctx.fill(); // Mengisi Warna

ctx.closePath();

ctx.stroke();
}

// Menggambar Awan Pertama
const cloud1_points = [
    {x: 370, y: 100},
    {x: 265, y: 100},
    {x: 200, y: 105},
    {x: 268, y: 75},
    {x: 250, y: 80},
    {x: 300, y: 30},
    {x: 330, y: 70},
    {x: 370, y: 70},
    {x: 350, y: 100}
];

drawBezierCurve(ctx, cloud1_points, "lightblue");

// Menggambar Awan Kedua
const cloud2_points = [
    {x: 466, y: 150},

```

```

        {x: 360, y: 150},
        {x: 310, y: 155},
        {x: 368, y: 125},
        {x: 350, y: 130},
        {x: 400, y: 80},
        {x: 430, y: 120},
        {x: 465, y: 120},
        {x: 450, y: 150}
    ];

    drawBezierCurve(ctx, cloud2_points, "lightblue");

    const cloud3_points = cloud2_points.map(point => ({x:
point.x - 350, y: point.y - 80}));

    drawBezierCurve(ctx, cloud3_points, "lightblue");

    // Gambar Daun Pertama
    const Daun1_points = [
        {x: 280, y: 320},
        {x: 190, y: 340},
        {x: 200, y: 225},
        {x: 240, y: 200},
        {x: 280, y: 190},
        {x: 280, y: 220},
        {x: 280, y: 205},
        {x: 270, y: 200},
        {x: 360, y: 120},

```

```
{x: 380, y: 220},  
  
{x: 510, y: 210},  
  
{x: 410, y: 410},  
  
{x: 295, y: 310},  
  
{x: 250, y: 145},  
  
{x: 370, y: 110},  
  
{x: 400, y: 180},  
  
{x: 430, y: 180},  
  
{x: 460, y: 250},  
  
{x: 380, y: 300},  
  
{x: 340, y: 310},  
  
{x: 290, y: 300},  
  
{x: 250, y: 320},  
  
{x: 280, y: 190},  
  
{x: 200, y: 150},  
  
{x: 370, y: 160},  
  
{x: 240, y: 90}  
  
];
```

```
drawBezierCurve(ctx, Daun1_points, "green");
```

```
// Gambar Daun Kedua
```

```
const Daun2_points = Daun1_points.map(point => ({x: point.x  
- 230, y: point.y - 100}));
```

```
drawBezierCurve(ctx, Daun2_points, "yellow");
```

```

// Gambar Semak Semak Pertama

const Semak1_points = [

    {x: 150, y: 410},

    {x: 180, y: 320},

    {x: 200, y: 430},

    {x: 210, y: 370},

    {x: 240, y: 390},

    {x: 255, y: 400},

    {x: 210, y: 375},

    {x: 370, y: 440},

    {x: 150, y: 400},

    {x: 150, y: 410}

];

drawBezierCurve(ctx, Semak1_points, "green");

// Gambar Semak Semak Kedua

const Semak2_points = Semak1_points.map(point => ({x:
point.x - 150, y: point.y - 5}));

drawBezierCurve(ctx, Semak2_points, "green");

// Gambar Semak Semak Ketiga

const Semak3_points = Semak1_points.map(point => ({x:
point.x - 90, y: point.y - 50}));

drawBezierCurve(ctx, Semak3_points, "green");

// Menggambar Pohon Pertama

```



```

const branchPoints1 = [
    {x: 310, y: 480},
    {x: 310, y: 320},
    {x: 310, y: 310},
    {x: 300, y: 290},
    {x: 180, y: 185},
    {x: 330, y: 345},
    {x: 325, y: 245},
    {x: 330, y: 360},
    {x: 460, y: 190},
    {x: 350, y: 310},
    {x: 350, y: 470},
    {x: 350, y: 370},
    {x: 350, y: 480}
];

drawBezierCurve(ctx, branchPoints1, "brown");

// Menggambar Pohon Kedua

const branchPoints2 = branchPoints1.map(point => ({x:
point.x - 200, y: point.y - 120}));

drawBezierCurve(ctx, branchPoints2, "brown");

// Menambahkan Event Listener untuk Resize Canvas

window.addEventListener("resize", () => {
    canvas.width = window.innerWidth;

```

```
        canvas.height = window.innerHeight;

        drawBezierCurve(ctx, cloud1_points, "lightblue");
        drawBezierCurve(ctx, cloud2_points, "lightblue");
        drawBezierCurve(ctx, cloud3_points, "lightblue");
        drawBezierCurve(ctx, Daun1_points, "green");
        drawBezierCurve(ctx, Daun2_points, "yellow");
        drawBezierCurve(ctx, Semak1_points, "green");
        drawBezierCurve(ctx, Semak2_points, "green");
        drawBezierCurve(ctx, Semak3_points, "green");
        drawBezierCurve(ctx, branchPoints1, "brown");
        drawBezierCurve(ctx, branchPoints2, "brown");

    });

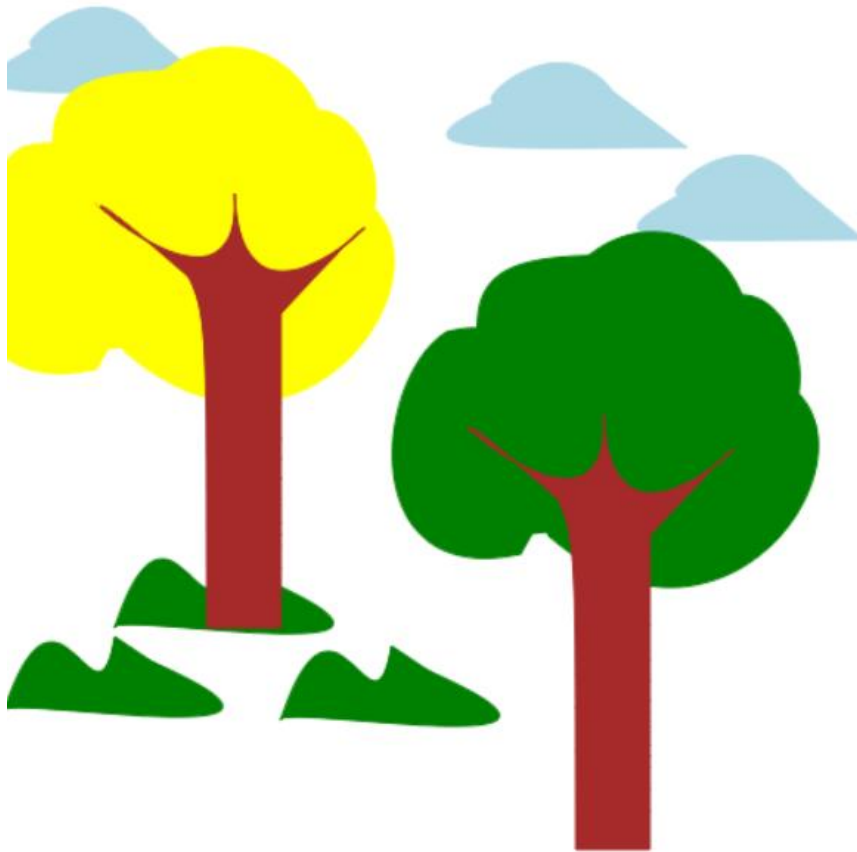
</script>

<a href="index.html">Kembali ke home</a>

</body>

</html>
```

## **HASIL DARI CODE PROGRAM**



## PENJELASAN CODE PROGRAM

### 1. Struktur Dasar HTML

```
1 <!DOCTYPE html>
2 <html lang="en">
```

- Mendefinisikan dokumen sebagai HTML5
- Lang="en" menyatakan bahwa bahasa utama halaman ini adalah bahasa Inggris

```
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Kurva Bezier Cubic - Dua Awan dan Pohon</title>
7 </head>
```

- Meta charset="UTF-8" memastikan teks dapat menampilkan karakter internasional dengan benar

- Viewport digunakan agar tampilan responsif di berbagai perangkat
- <title> adalah judul browser

```

8 <body>
9   <h1 align="center">MEMBUAT POHON DAN AWAN</h1>
10  <canvas id="canvas" width="500" height="500" style="border: 2px solid black;"></canvas>

```

- Judul utama halaman, ditampilkan di tengah
- <canvas> adalah area gambar. Lebar dan tinggi ditentukan 500px dan diberi garis tepi hitam

## 2. Menggambar dengan Kurva Bezier

```

13 const canvas = document.getElementById("canvas");
14 const ctx = canvas.getContext("2d");

```

- Mengambil elemen canvas dan membuat context 2D nya untuk menggambar grafik 2D

### a. Fungsi drawBezierCurve()

```

17 function drawBezierCurve(ctx, points, color) {
18   ctx.beginPath();
19   ctx.moveTo(points[0].x, points[0].y);
20   ctx.strokeStyle = color;
21   ctx.fillStyle = color;
22   for (let i = 1; i < points.length - 2; i += 3) {
23     ctx.bezierCurveTo(
24       points[i].x, points[i].y,
25       points[i + 1].x, points[i + 1].y,
26       points[i + 2].x, points[i + 2].y
27     );
28   }
29   ctx.fill(); // Mengisi Warna
30   ctx.closePath();
31   ctx.stroke();
32 }

```

- Fungsi ini menerima:
  - ctx: context 2D canvas
  - points: array koordinat titik
  - color: warna untuk fill dan garis
- Kurva bezier memerlukan 4 titik:

- Satu titik awal (moveTo)
- Tiga titik selanjutnya (bezierCurveTo)
- Fungsi ini mengambil array titik dan menggambar kurva berdasarkan kelipatan 3 titik kontrol

## b. Menggambar Awan

```

35     const cloud1_points = [
36         {x: 370, y: 100},
37         {x: 265, y: 100},
38         {x: 200, y: 105},
39         {x: 268, y: 75},
40         {x: 250, y: 80},
41         {x: 300, y: 30},
42         {x: 330, y: 70},
43         {x: 370, y: 70},
44         {x: 350, y: 100}
45     ];
46     drawBezierCurve(ctx, cloud1_points, "lightblue");

```

- Titik – titik dalam array membentuk kurva awan pertama berwarna biru muda

```

48     // Menggambar Awan Kedua
49     const cloud2_points = [
50         {x: 466, y: 150},
51         {x: 360, y: 150},
52         {x: 310, y: 155},
53         {x: 368, y: 125},
54         {x: 350, y: 130},
55         {x: 400, y: 80},
56         {x: 430, y: 120},
57         {x: 465, y: 120},
58         {x: 450, y: 150}
59     ];
60     drawBezierCurve(ctx, cloud2_points, "lightblue");

```

- Awan kedua, bentuk dan lokasi nya berbeda

```

62     const cloud3_points = cloud2_points.map(point => ({x: point.x - 350, y: point.y - 80}));
63     drawBezierCurve(ctx, cloud3_points, "lightblue");

```

- Awan ketiga dibentuk dengan menggeser awan kedua ke kiri atas

### c. Menggambar Daun

```
66     const Daun1_points = [  
67         {x: 280, y: 320},  
68         {x: 190, y: 340},  
69         {x: 200, y: 225},  
70         {x: 240, y: 200},  
71         {x: 280, y: 190},  
72         {x: 280, y: 220},  
73         {x: 280, y: 205},  
74         {x: 270, y: 200},  
75         {x: 360, y: 120},  
76         {x: 380, y: 220},  
77         {x: 510, y: 210},  
78         {x: 410, y: 410},  
79         {x: 295, y: 310},  
80         {x: 250, y: 145},  
81         {x: 370, y: 110},  
82         {x: 400, y: 180},  
83         {x: 430, y: 180},  
84         {x: 460, y: 250},  
85         {x: 380, y: 300},  
86         {x: 340, y: 310},  
87         {x: 290, y: 300},  
88         {x: 250, y: 320},  
89         {x: 280, y: 190},  
90         {x: 200, y: 150},  
91         {x: 370, y: 160},  
92         {x: 240, y: 90}  
93     ];  
94  
95     drawBezierCurve(ctx, Daun1_points, "green");
```

- Daun pertama dengan banyak lekukan, dengan warna hijau

```
98     const Daun2_points = Daun1_points.map(point => ({x: point.x - 230, y: point.y - 100}));  
99     drawBezierCurve(ctx, Daun2_points, "yellow");
```

- Daun kedua adalah salinan daun pertama tapi digeser, dengan warna kuning

#### d. Menggambar Semak

```
102     const Semak1_points = [  
103         {x: 150, y: 410},  
104         {x: 180, y: 320},  
105         {x: 200, y: 430},  
106         {x: 210, y: 370},  
107         {x: 240, y: 390},  
108         {x: 255, y: 400},  
109         {x: 210, y: 375},  
110         {x: 370, y: 440},  
111         {x: 150, y: 400},  
112         {x: 150, y: 410}  
113     ];  
114  
115     drawBezierCurve(ctx, Semak1_points, "green");
```

- Semak pertama, bentuk menyerupai lengkungan

```
117     // Gambar Semak Semak Kedua  
118     const Semak2_points = Semak1_points.map(point => ({x: point.x - 150, y: point.y - 5}));  
119     drawBezierCurve(ctx, Semak2_points, "green");  
120  
121     // Gambar Semak Semak Ketiga  
122     const Semak3_points = Semak1_points.map(point => ({x: point.x - 90, y: point.y - 50}));  
123     drawBezierCurve(ctx, Semak3_points, "green");
```

- Semak kedua dan ketiga adalah semak pertama tapi digeser

#### e. Menggambar Pohon

```
126     const branchPoints1 = [  
127         {x: 310, y: 480},  
128         {x: 310, y: 320},  
129         {x: 310, y: 310},  
130         {x: 300, y: 290},  
131         {x: 180, y: 185},  
132         {x: 330, y: 345},  
133         {x: 325, y: 245},  
134         {x: 330, y: 360},  
135         {x: 460, y: 190},  
136         {x: 350, y: 310},  
137         {x: 350, y: 470},  
138         {x: 350, y: 370},  
139         {x: 350, y: 480}  
140     ];  
141  
142     drawBezierCurve(ctx, branchPoints1, "brown");
```

- Pohon pertama dibuat dari garis melengkuk menyerupai batang dan cabang, dengan berwarna coklat

```
145     const branchPoints2 = branchPoints1.map(point => ({x: point.x - 200, y: point.y - 120}));
146     drawBezierCurve(ctx, branchPoints2, "brown");
```

- Pohon kedua adalah versi pohon pertama yang digeser

#### f. Responsif saat Layar Diresize

```
149     window.addEventListener("resize", () => {
150         canvas.width = window.innerWidth;
151         canvas.height = window.innerHeight;
152         drawBezierCurve(ctx, cloud1_points, "lightblue");
153         drawBezierCurve(ctx, cloud2_points, "lightblue");
154         drawBezierCurve(ctx, cloud3_points, "lightblue");
155         drawBezierCurve(ctx, Daun1_points, "green");
156         drawBezierCurve(ctx, Daun2_points, "yellow");
157         drawBezierCurve(ctx, Semak1_points, "green");
158         drawBezierCurve(ctx, Semak2_points, "green");
159         drawBezierCurve(ctx, Semak3_points, "green");
160         drawBezierCurve(ctx, branchPoints1, "brown");
161         drawBezierCurve(ctx, branchPoints2, "brown");
162     });
```

- Event listener ini akan memicu ketika ukuran jendela browser berubah
- Saat terjadi perubahan ukuran, canvas.widht dan canvas.height diubah agar sesuai dengan ukuran jendela
- Setelah ukuran diubah, semua gambar di-*redward* karena mengganti ukuran canvas akan menghapus konten sebelumnya

### 3. Kembali ke Home

```
165     <a href="index.html">Kembali ke home</a>
```

- Sebuah tautan untuk kembali ke halaman utama, index.html



## **KESIMPULAN**

Kurva Bezier Cubic bisa dipakai untuk menggambar yang halus dan fleksibel, seperti awan, daun, pohon, dan semak – semak. Dengan 4 titik kendali, kita bisa atur bentuk kurva seskua hati. Di program ini, kurva bezier dipakai di canvas HTML untuk membuat objek – objek alam. Gambar juga otomatis mengikuti ukuran layar, jadi tampilannya tetap bagus di semua perangkat. Kurva ini cocok untuk membuat desain grafis atau animasi yang butuh bentuk – bentuk organik.