

**LAPORAN GRAFIKA KOMPUTER**  
**ALGORITMA PEMBENTUKAN KURVA**

Disusun guna memenuhi tugas mata kuliah Grafika Komputer Dosen Pengampu

**Bapak Febi Eka Febriansyah, M.T.,**

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

**Putut Aji Nalendro, M.PD.**



**Disusun Oleh:**

Nama: Lulu Saputri

NPM: 2413025017

Kelas: PTI 24A

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

**2025**

## A. Pengertian Kurva Bezier cubic

Kurva Bezier Kubik adalah jenis kurva Bezier orde tiga yang dibentuk menggunakan empat titik kontrol, yaitu:

- Titik awal ( $P_0$ )
- Dua titik pengendali di tengah ( $P_1$  dan  $P_2$ )
- Titik akhir ( $P_3$ )

Kurva ini merupakan bentuk paling umum dari kurva Bezier karena menawarkan kombinasi ideal antara fleksibilitas dan kemudahan penggunaan.

Kurva Bezier Kubik dimodelkan secara matematis menggunakan persamaan parametris yang menghasilkan lintasan halus dan terkontrol. Nilai parameter  $t$  yang berada dalam rentang  $0 \leq t \leq 1$  digunakan untuk menghitung posisi titik di sepanjang kurva.

Jadi, Kurva Bezier Kubik adalah kurva lengkung yang dikontrol oleh 4 titik dan sering digunakan dalam bidang grafis komputer, seperti desain font, grafik vektor, dan animasi, karena mampu menghasilkan bentuk yang lentur dan presisi.

## B. Rumus Parametrik

$$B(t) = (1-t)^3 \cdot P_0 + 3(1-t)^2 t \cdot P_1 + 3(1-t)t^2 \cdot P_2 + t^3 \cdot P_3$$

- $t$  berada di antara 0 dan 1
- Setiap  $t$  menghasilkan satu titik di sepanjang kurva

## C. Keunggulan Kurva Bézier Kubik

1. Memungkinkan kontrol bentuk kurva yang presisi karena dimulai dari titik awal ( $P_0$ ) dan berakhir di titik akhir ( $P_3$ ), sehingga cocok untuk membuat jalur atau bentuk yang diinginkan secara akurat.
2. Memberikan fleksibilitas dalam desain visual, karena dua titik kontrol ( $P_1$  dan  $P_2$ ) secara dinamis mengatur lengkungan tanpa harus dilalui oleh kurva itu sendiri.
3. Mudah diarahkan dan dimodifikasi, karena arah awal dan akhir kurva dapat dikendalikan dengan memanipulasi garis  $P_0-P_1$  dan  $P_2-P_3$ .
4. Aman dan stabil dalam ruang desain, karena kurva selalu berada di dalam area terbatas (convex hull) dari empat titik kontrol, mencegah bentuk kurva keluar dari jalur yang diharapkan.
5. Dapat ditransformasi dengan bebas tanpa kehilangan bentuk asli, karena kurva ini mendukung transformasi affine seperti translasi (geser), rotasi (putar), dan penskalaan (perbesar/perkecil), sangat ideal untuk animasi dan desain responsif.

#### D. Kode Program

```
<!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>

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

<body>

  <canvas id="canvas" width ="500" height="500" style="border: 2px
solid #379b82 ;"></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
        );
      }
    }
  </script>
</body>
</html>
```

```

    }

    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, "#B0E0E6"); //Powder Blue, Biru pastel, doft
sekali

// Menggambar Awan Kedua (geser ke atas)
const cloud2_points = [
    {x: 465, y: 150},
    {x: 350, y: 150},
    {x: 300, 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, "#B0E0E6"); //Powder Blue, Biru pastel, doft
sekali
```

```
  const cloud3_points = cloud2_points.map(point => ({x:
point.x - 350, y: point.y - 80}));
  drawBezierCurve(ctx, cloud3_points, "#B0E0E6"); //Powder Blue, Biru pastel, doft
sekali
```

// 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, "#8A9A5B"); //Desert Sand, Hijau lumut, kalem  
dan earthy
```

```
// Gambar Daun Kedua  
const Daun2_points = Daun1_points.map(point => ( {x: point.x - 200, y: point.y -  
120}));
```

```
drawBezierCurve(ctx, Daun2_points, "#A9BA9D"); //Laurel Green, Hijau keabu-  
abuan kalem
```

```
// Gambar Semak Semak Pertama  
const Semak1_points = [  
{x: 150, y: 410},  
{x: 190, y: 320},  
{x: 200, y: 430},  
{x: 210, y: 370},  
{x: 240, y: 390},  
{x: 255, y: 400},  
{x: 250, y: 375},  
{x: 370, y: 440},  
{x: 150, y: 400},  
{x: 150, y: 410}  
];
```

```
drawBezierCurve(ctx, Semak1_points, "#93C572"); //pistachio, Hijau kacang  
pistachio yang cerah
```

```
// Gambar Semak Semak Kedua

const Semak2_points = Semak1_points.map(point => ({x:
point.x - 190, y: point.y - 10}));

drawBezierCurve(ctx, Semak2_points, "#93C572"); //pistachio, Hijau kacang
pistachio yang cerah
```

```
// Gambar Semak Semak Ketiga

const Semak3_points = Semak1_points.map(point => ({x:
point.x - 150, y: point.y - 80}));

drawBezierCurve(ctx, Semak3_points, "#93C572"); //pistachio, Hijau kacang
pistachio yang cerah
```

```
// 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, "#EDC9AF"); //Desert Sand, Coklat pastel
terang
```

```
// Menggambar Pohon Kedua

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

drawBezierCurve(ctx, branchPoints2, "#EDC9AF"); //Desert Sand, Coklat pastel
terang

// Menambahkan Event Listener untuk Resize Canvas
window.addEventListener("resize", () => {
  canvas.width = window.innerWidth;
  canvas.height = window.innerHeight;
  drawBezierCurve(ctx, cloud1_points, "Powder Blue");
  drawBezierCurve(ctx, cloud2_points, "Powder Blue");
  drawBezierCurve(ctx, branchPoints1, "Desert Sand");
  drawBezierCurve(ctx, branchPoints2, "Desert Sand");
  drawBezierCurve(ctx, Daun1_points, "Desert Sand");
  drawBezierCurve(ctx, Daun2_points, "Laurel Green");
  drawBezierCurve(ctx, Semak1_points, "pistachio");
  drawBezierCurve(ctx, Semak2_points, "pistachio");
  drawBezierCurve(ctx, Semak3_points, "pistachio");
});
</script>
<a href="index.html"> kembali ke home</a>
</body>
</html>
```



## E. Hasil dari Programnya:



## F. Kesimpulan

Kurva Bézier Kubik adalah salah satu teknik penting dalam pemrograman grafis yang memungkinkan pembuatan bentuk lengkung yang halus dan fleksibel. Dengan menggunakan empat titik kontrol, kurva ini sangat efektif dalam membentuk berbagai objek visual seperti awan, daun, semak, dan batang pohon secara realistis dan estetik.

Melalui kode program berbasis HTML dan JavaScript dengan elemen `<canvas>`, pengguna dapat menggambar pemandangan alam dengan komponen berbentuk kurva menggunakan fungsi `bezierCurveTo()`. Ini menunjukkan bahwa pemahaman konsep matematis kurva Bézier dapat diterapkan secara langsung dalam visualisasi grafis komputer, terutama pada desain vektor dan animasi.

**G. Link Video:** [https://youtu.be/ngxQ-DPisVE?si=6yMcIsxttd\\_oESAi](https://youtu.be/ngxQ-DPisVE?si=6yMcIsxttd_oESAi)