

LAPORAN PRAKTIKUM MATA KULIAH GRAFIKA KOMPUTER

“Implementasi Algoritma Pembentukan Kurva Menggunakan HTML”



Dosen Mata Kuliah :

Febi Eka Febriansyah, M.T.

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

Putut Aji Nalendro, M.Pd.

Disusun Oleh :

Nama : Raehan

NPM : 2413025033

PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI

FAKULTAS KEGURUAN DAN ILMU PENDIDIKAN

UNIVERSITAS LAMPUNG

2025

A. Kurva Bézier Kubik

Kurva Bézier kubik adalah salah satu bentuk paling umum dari kurva Bézier yang banyak digunakan dalam grafika komputer, karena mampu menghasilkan lengkungan yang halus dan fleksibel dengan kontrol yang relatif sederhana. Kurva ini ditentukan oleh empat titik kendali, yaitu P_0 , P_1 , P_2 , dan P_3 . Titik P_0 dan P_3 adalah titik-titik ujung kurva (awal dan akhir), sedangkan P_1 dan P_2 berfungsi sebagai penarik arah lengkung (titik kontrol) yang memengaruhi bentuk kurva namun tidak harus dilalui oleh kurva itu sendiri.

Secara matematis, kurva Bézier kubik dinyatakan dengan persamaan parametrik sebagai berikut:

$$B(t) = (1-t)^3P_0 + 3(1-t)^2tP_1 + 3(1-t)t^2P_2 + t^3P_3,$$

dengan t merupakan parameter yang bergerak dari 0 hingga 1. Persamaan ini merupakan kombinasi dari polinomial bernilai empat yang menginterpolasi titik awal dan akhir, serta menginterpolasi arah kurva berdasarkan posisi titik-titik kontrol.

Kurva Bézier kubik memiliki banyak penerapan dalam berbagai bidang, terutama yang berkaitan dengan grafika komputer, desain, dan animasi. Dalam desain grafis vektor, kurva ini digunakan untuk menggambar bentuk-bentuk kompleks yang halus dan terkontrol, seperti logo, ikon, dan ilustrasi digital. Selain itu, dalam pemrograman antarmuka pengguna (UI/UX), kurva Bézier kubik juga digunakan untuk mengatur animasi transisi yang halus, seperti perpindahan tombol atau efek pergerakan elemen antar layar, misalnya melalui CSS transition dengan fungsi seperti `cubic-bezier(x1, y1, x2, y2)` pada pengembangan web.

B. Implementasi Kurva Bézier Kubik pada HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Raehan - Kurva Bezier Cubic – Pohon, Awan, dan Semak</title>
```

```
<style>
  canvas {
    border: 2px solid #87CEEB;
  }
  h1 {
    text-align: center;
  }
</style>
</head>
<body>
  <h1>POHON, AWAN, DAN SEMAK</h1>
  <canvas id="canvas" width="800" height="800"></canvas>
  <a href="index.html">Kembali ke home</a>

  <script>
    const canvas = document.getElementById("canvas");
    const ctx = canvas.getContext("2d");

    // Fungsi menggambar kurva Bezier
    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();
      ctx.closePath();
    }
  </script>
</body>
</html>
```

```

    ctx.stroke();
  }

  // Daun 1
  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, "red");

  // Daun 2
  const Daun2_points = Daun1_points.map(point => ({
    x: point.x - 200,
    y: point.y - 120
  }));
  drawBezierCurve(ctx, Daun2_points, "purple");

  // Semak 1
  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, "skyblue");

```

```
// Semak 2
const Semak2_points = Semak1_points.map(point => ({
  x: point.x + 190,
  y: point.y - 10
}));
drawBezierCurve(ctx, Semak2_points, "green");

// Semak 3
const Semak3_points = Semak1_points.map(point => ({
  x: point.x - 150,
  y: point.y - 80
}));
drawBezierCurve(ctx, Semak3_points, "black");

// Pohon 1
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, "tan");

// Pohon 2
const branchPoints2 = branchPoints1.map(point => ({
  x: point.x - 200,
  y: point.y - 120
}));
drawBezierCurve(ctx, branchPoints2, "tan");

// Awan 1
const cloud1_points = [
```

```

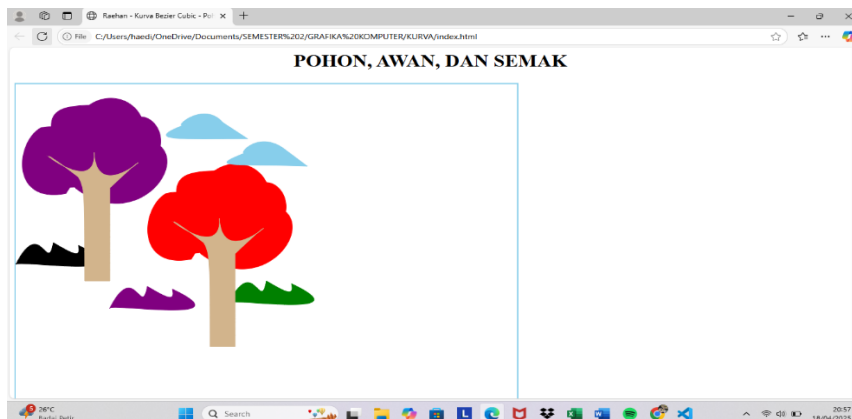
        {x: 370, y: 100}, {x: 265, y: 100}, {x: 200, y: 105},
        {x: 268, y: 75}, {x: 250, y: 80}, {x: 300, y: 30},
        {x: 330, y: 70}, {x: 370, y: 70}, {x: 370, y: 100}
    ];
    drawBezierCurve(ctx, cloud1_points, "skyblue");

    // Awan 2
    const cloud2_points = [
        {x: 465, y: 150}, {x: 350, y: 145}, {x: 300, y: 155},
        {x: 368, y: 125}, {x: 350, y: 135}, {x: 385, y: 80},
        {x: 430, y: 120}, {x: 400, y: 120}, {x: 450, y: 150}
    ];
    drawBezierCurve(ctx, cloud2_points, "skyblue");

    // Resize canvas (optional)
    window.addEventListener("resize", () => {
        canvas.width = window.innerWidth;
        canvas.height = window.innerHeight;
        // (Redraw bisa ditambahkan di sini kalau mau responsif)
    });
</script>
</body>
</html>

```

Output dari kode HTML di atas sebagai berikut :



Kode ini membuat sebuah tampilan grafis menggunakan elemen `<canvas>` HTML5 yang berisi gambar **pohon, awan, dan semak**. Gambar-gambar tersebut dibuat menggunakan kurva Bézier kubik melalui fungsi `ctx.bezierCurveTo()` di dalam JavaScript. Setiap objek (daun, semak, pohon, dan awan) direpresentasikan sebagai serangkaian titik yang digunakan sebagai titik kendali untuk menghasilkan lengkungan yang halus. Fungsi `drawBezierCurve()` digunakan untuk menggambar kurva dengan menerima parameter berupa konteks canvas (`ctx`), daftar titik, dan warna. Objek-objek tersebut kemudian digambar satu per satu dengan memanipulasi titik-titik kontrol dan menerapkan translasi posisi untuk membuat variasi bentuk yang berbeda. Kode ini juga mencakup fitur responsif (opsional), di mana ukuran canvas akan disesuaikan secara otomatis saat jendela browser diubah ukurannya.

C. Kesimpulan

Melalui praktikum ini, dapat disimpulkan bahwa kurva Bézier kubik merupakan metode yang sangat efektif untuk membentuk objek grafis dengan bentuk lengkung yang halus dan fleksibel. Dengan menggunakan elemen `<canvas>` dalam HTML dan fungsi `bezierCurveTo()` pada JavaScript, kita dapat mengimplementasikan berbagai bentuk visual seperti pohon, awan, dan semak secara dinamis dan artistik. Pendekatan ini memungkinkan manipulasi titik-titik kontrol untuk menciptakan bentuk-bentuk kompleks yang terstruktur, serta memberikan pemahaman yang lebih dalam mengenai prinsip dasar grafika komputer, khususnya dalam pembuatan kurva dan objek dua dimensi.