

LAPORAN TUGAS KECIL 2 IF2211
STRATEGI ALGORITMA

*Membangun Kurva Bézier dengan Algoritma Titik Tengah berbasis
Divide and Conquer*



Dosen Pengampu: Dr. Nur Ulfa Maulidevi, S.T, M.Sc

Disusun oleh:

Panji Sri Kuncara Wisma (13522028)

Haikal Assyauqi (13522052)

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG

2024

DAFTAR ISI

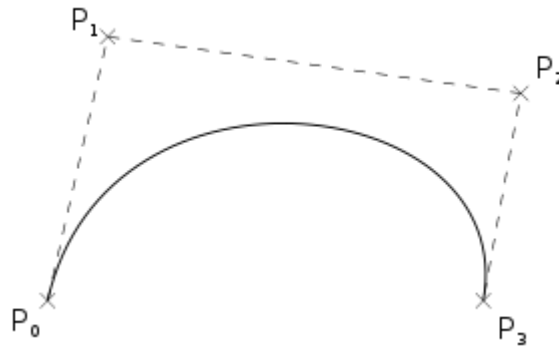
DAFTAR ISI.....	1
PENGECEKAN PROGRAM.....	2
DESKRIPSI MASALAH.....	3
A. Algoritma Divide and Conquer Tiga Titik atau Dua Garis.....	4
B. Algoritma Brute Force.....	5
C. Source Code.....	6
D. Input dan Output.....	10
E. Analisis Brute Force dan Divide and Conquer.....	15
F. Algoritma Divide and Conquer untuk N titik ($N > 3$).....	15
G. Repository.....	16

PENGECEKAN PROGRAM

Poin	Ya	Tidak
1. Program berhasil dijalankan.	✓	
2. Program dapat melakukan visualisasi kurva Bézier.	✓	
3. Solusi yang diberikan program optimal.	✓	
4. [Bonus] Program dapat membuat kurva untuk n titik kontrol.	✓	
5. [Bonus] Program dapat melakukan visualisasi proses pembuatan kurva.	✓	

Tabel 1. Tabel Pengecekan Program

DESKRIPSI MASALAH



Gambar 1. Kurva Bézier Kubik

(Sumber: https://id.wikipedia.org/wiki/Kurva_B%C3%A9zier)

Kurva Bézier adalah kurva halus yang sering digunakan dalam desain grafis, animasi, dan manufaktur. Kurva ini dibuat dengan menghubungkan beberapa titik kontrol, yang menentukan bentuk dan arah kurva. Cara membuatnya cukup mudah, yaitu dengan menentukan titik-titik kontrol dan menghubungkannya dengan kurva. Kurva Bézier memiliki banyak kegunaan dalam kehidupan nyata, seperti pen tool, animasi yang halus dan realistis, membuat desain produk yang kompleks dan presisi, dan membuat font yang indah dan unik. Keuntungan menggunakan kurva Bézier adalah kurva ini mudah diubah dan dimanipulasi, sehingga dapat menghasilkan desain yang presisi dan sesuai dengan kebutuhan.

Sebuah kurva Bézier didefinisikan oleh satu set titik kontrol P_0 sampai P_n , dengan n disebut order ($n = 1$ untuk linier, $n = 2$ untuk kuadrat, dan seterusnya). Titik kontrol pertama dan terakhir selalu menjadi ujung dari kurva, tetapi titik kontrol antara (jika ada) umumnya tidak terletak pada kurva. Pada gambar 1 diatas, titik kontrol pertama adalah P_0 , sedangkan titik kontrol terakhir adalah P_3 . Titik kontrol P_1 dan P_2 disebut sebagai titik kontrol antara yang tidak terletak dalam kurva yang terbentuk.

Tugas yang diberikan adalah mengimplementasikan pembuatan kurva Bézier dengan algoritma titik tengah berbasis divide and conquer

A. Algoritma Divide and Conquer Tiga Titik atau Dua Garis

Rekursif adalah konsep utama yang digunakan untuk mengimplementasikan algoritma divide and conquer untuk tiga titik atau dua garis. Misalkan terdapat tiga buah titik yaitu titik kontrol awal, titik kontrol antara dan titik kontrol akhir, maka :

- Kasus Basis (iterasi == 1)

Kasus basis terjadi ketika iterasi sama dengan satu. Berikut adalah hal yang terjadi pada kasus basis:

- a. Dicari terlebih dahulu beberapa titik yaitu tengah_1 yang berada di tengah garis yang menghubungkan titik kontrol awal dan titik kontrol antara, tengah_2 yang berada di tengah garis yang menghubungkan titik kontrol antara dan titik kontrol akhir
- b. Setelah mendapatkan tengah_1 dan tengah_2, dicari titik baru lagi yaitu tengah_dari_tengah yang berada di tengah garis yang menghubungkan titik tengah_1 dengan tengah_2
- c. Pengecekan array yang berisi titik kurva bezier dilakukan
- d. Apabila array kosong, titik kontrol awal, tengah_dari_tengah, titik kontrol akhir, secara berurutan ditambahkan ke dalam array sebagai elemen baru
- e. Apabila array tidak kosong, ada dua kondisi. Jika elemen terakhir array nilainya sama dengan titik kontrol awal maka yang ditambahkan ke dalam array hanya titik tengah_dari_tengah dan titik kontrol akhir. Jika elemen terakhir array nilainya tidak sama dengan titik kontrol awal, maka titik kontrol awal, tengah_dari_tengah, dan titik kontrol akhir secara berurutan ditambahkan ke dalam array sebagai elemen baru
- f. Titik titik yang berada di dalam array dihubungkan oleh suatu garis dan jadilah kurva bezier

- Kasus rekursif (iterasi > 1)

Apabila iterasi lebih besar dari satu maka pembentukan kurva bezier akan dibagi menjadi dua bagian, yaitu bagian sebelah kiri titik kontrol antara dan bagian sebelah kanan titik kontrol antara.

- a. Dicari terlebih dahulu beberapa titik yaitu *tengah_1* yang berada di tengah garis yang menghubungkan titik kontrol awal dan titik kontrol antara, *tengah_2* yang berada di tengah garis yang menghubungkan titik kontrol antara dan titik kontrol akhir
- b. Untuk mengurus sub bagian sebelah kiri, fungsi kurva bezier dipanggil kembali dengan titik titik kontrol yang baru. Titik kontrol awal tetap menjadi titik kontrol awal, *tengah_1* bertugas sebagai titik kontrol antara, dan *tengah_dari_tengah* bertugas sebagai titik kontrol akhir. Kemudian nilai iterasi dikurangi dengan satu
- c. Untuk mengurus sub bagian sebelah kanan, fungsi kurva bezier dipanggil kembali dengan titik titik kontrol yang baru. Titik *tengah_dari_tengah* bertugas sebagai *titik_kontrol_awal*, *tengah_2* bertugas sebagai titik kontrol antara, dan titik kontrol akhir tetap bertugas sebagai titik kontrol akhir. Kemudian nilai iterasi dikurangi dengan satu
- d. Proses b dan c adalah proses divide, setelah mencapai basis akan terjadi proses conquer dan merge sesuai yang sudah dijelaskan pada kasus basis
- e. Proses b dan c akan terus dilakukan hingga mencapai basis

- **KETERANGAN PENTING TAMBAHAN**

Ketika melakukan plot kurva bezier, untuk masing-masing sub kurva bezier terdapat jeda waktu 0.5 detik yang terjadi untuk keperluan animasi. Oleh karena itu dalam perhitungan waktu total atau *runtime* program adalah waktu akhir - waktu awal - (total galat yang terjadi * 0.5) detik.

B. Algoritma Brute Force

Pada algoritma brute force, kurva bezier didapatkan dengan rumus yang telah diberikan pada spesifikasi, yang mana tiap rumus berbeda tiap jumlah titik, beberapa rumus di antaranya:

1. Jumlah titik sama dengan 3

$$R_0 = B(t) = (1-t)^2 P_0 + (1-t)t P_1 + t^2 P_2, \quad t \in [0, 1]$$

2. Jumlah titik sama dengan 4

$$S_0 = B(t) = (1-t)^3 P_0 + 3(1-t)^2 t P_1 + 3(1-t)t^2 P_2 + t^3 P_3, \quad t \in [0, 1]$$

3. Jumlah titik sama dengan 5

$$T_0 = B(t) = (1-t)^4 P_0 + 4(1-t)^3 t P_1 + 6(1-t)^2 t^2 P_2 + 4(1-t)t^3 P_3 + t^4 P_4, \quad t \in [0, 1]$$

Dari 3 rumus di atas, kita dapat menurunkannya menjadi rumus umum yang dapat digunakan untuk mencari titik ke-n, rumusnya sebagai berikut

$$B(t) = xC_0 \cdot (1-t)^{x-1} \cdot P_0 + xC_1 \cdot (1-t)^{x-2} \cdot P_1 + \dots + xC_n \cdot t^{x-1} \cdot P_{x-1}$$

Dimana x = jumlah titik

Dari rumus yang diturunkan, didapatkan rumus kurva bezier, cara untuk mendapatkan kurva bezier dengan rumus tersebut adalah:

1. Tentukan jumlah iterasi
2. Agar keakuratan gambar pada kurva bezier brute force sama dengan kurva bezier dnc, maka jumlah titik yang dibentuk sama dengan $2^{\text{iterasi}} - 1$
3. Karena range t dari 0 sampai 1 maka tiap titik dibentuk dengan $t = 1/(2^{\text{iterasi}} - 1)$, yang t nya akan terus menaik hingga $t = 1$

C. Source Code

- bruteforce.py

```
src > bruteforce.py factorial
1 def factorial (num) :
2     angka = 1
3     for i in range(1,num + 1) :
4         angka *= i
5     return angka
6
7 def combination (leftnum, rightnum) :
8     return round(factorial(leftnum) / (factorial(rightnum) * factorial(leftnum-rightnum)))
9
10 def bezierbruteforce(array, t) :
11     # Array = banyak titik
12     titik = []
13     garis = len(array)-1 # jumlah garis
14     x = 0
15     y = 0
16     for i in range(garis+1) :
17         x += round(combination(garis,i) * ((1-t)**(garis-i)) * (t**i) * array[i][0], 3)
18         y += round(combination(garis,i) * ((1-t)**(garis-i)) * (t**i) * array[i][1], 3)
19     titik.append(round(x,3))
20     titik.append(round(y,3))
21     return titik
22
```

- divide_and_conquer.py

```

nc > @ dnc_and_animation @ plot_kurva_no_animasi
1 import matplotlib.pyplot as plt
2
3 def titik_tengah(titik_awal, titik_akhir):
4     hasil = []
5     x_tengah = (titik_awal[0] + titik_akhir[0]) / 2
6     y_tengah = (titik_awal[1] + titik_akhir[1]) / 2
7     hasil.append(x_tengah)
8     hasil.append(y_tengah)
9     return hasil
10
11 def kurva_bezier(titik_1, titik_2, titik_3, iterasi, arr, kontrol1, kontrol2, kontrol3, total_iterasi_animasi, pakai_animasi):
12     if iterasi == 1:
13         tengah_1 = titik_tengah(titik_1, titik_2)
14         tengah_2 = titik_tengah(titik_2, titik_3)
15         tengah_dari_tengah = titik_tengah(tengah_1, tengah_2)
16
17         if arr == []:
18             arr.append(titik_1)
19             arr.append(tengah_dari_tengah)
20             arr.append(titik_3)
21         else:
22             if arr[-1] == titik_1:
23                 arr.append(tengah_dari_tengah)
24                 arr.append(titik_3)
25             else:
26                 arr.append(titik_1)
27                 arr.append(tengah_dari_tengah)
28                 arr.append(titik_3)
29             if pakai_animasi == 'Y':
30                 plt.clf()
31                 plt.plot(kontrol1, kontrol2, kontrol3)
32                 total_iterasi_animasi.append(1)
33                 plot_kurva(arr)
34             else:
35                 tengah_1 = titik_tengah(titik_1, titik_2)
36                 tengah_2 = titik_tengah(titik_2, titik_3)
37                 tengah_dari_tengah = titik_tengah(tengah_1, tengah_2)
38
39                 #ini buat bagian kiri
40                 kurva_bezier(titik_1, tengah_1, tengah_dari_tengah, iterasi-1, arr, kontrol1, kontrol2, kontrol3, total_iterasi_animasi, pakai_animasi)
41
42                 #ini buat bagian kanan
43                 kurva_bezier(tengah_dari_tengah, tengah_2, titik_3, iterasi-1, arr, kontrol1, kontrol2, kontrol3, total_iterasi_animasi, pakai_animasi)
44

```

```

45 def buat_kurva(arr):
46     titik_x = []
47     titik_y = []
48     for elemen in arr:
49         titik_x.append(elemen[0])
50         titik_y.append(elemen[1])
51     plt.title("Kurva Bezier")
52     plt.plot(titik_x, titik_y, marker = 'o', label = 'kurva bezier')
53     plt.legend()
54     plt.pause(0.5)
55
56 def plot_kurva_no_animasi(arr):
57     titik_x = []
58     titik_y = []
59     for elemen in arr:
60         titik_x.append(elemen[0])
61         titik_y.append(elemen[1])
62     plt.title("Kurva Bezier")
63     plt.plot(titik_x, titik_y, marker = 'o', label = 'kurva bezier')
64     plt.legend()
65
66
67 def plot_kontrol(titik_awal, titik_tengah, titik_akhir):
68     x_kontrol = [titik_awal[0], titik_tengah[0], titik_akhir[0]]
69     y_kontrol = [titik_awal[1], titik_tengah[1], titik_akhir[1]]
70     plt.plot(x_kontrol, y_kontrol, linestyle="--", marker = 'o', label = 'titik kontrol')
71

```

- dnc.py

```

nc > @ dnc.py @ Ngaris_BForce
1 import matplotlib.pyplot as plt
2 import bruteForce
3 import time
4
5 def midpoint(x1,y1,x2,y2):
6     koordinat = []
7     koordinat.append((x1+x2) / 2)
8     koordinat.append((y1+y2) / 2)
9     return koordinat
10
11 def list_midpoint(list_of_titik):
12     daftar_midpoint = []
13     for i in range(len(list_of_titik)-1):
14         daftar_midpoint.append(midpoint(list_of_titik[i][0], list_of_titik[i][1], list_of_titik[i+1][0], list_of_titik[i+1][1]))
15     return daftar_midpoint
16
17 def slicing(list, awal, akhir):
18     array = []
19     for i in range(awal,akhir):
20         array.append(list[i])
21     return array
22
23 def expandarray(array1, array2):
24     for i in range(len(array2)):
25         array1.append(array2[i])
26     return array1
27
28 def animasi(array,xawal,yawal):
29     plt.clf()
30     plt.scatter([i[0] for i in array],[i[1] for i in array])
31     plt.title("Kurva Bezier")
32     plt.plot([i[0] for i in array],[i[1] for i in array], marker = 'o', label = 'kurva bezier')
33     plt.plot(xawal, yawal, marker = 'o', label = 'titik kontrol')
34     plt.legend()
35     plt.pause(0.5)
36
37
38 def Ngaris_BForce(algo):
39     titikawal = [] #titik p0
40     titikakhir = [] #titik pn
41     titikbantu = [] #semua titik kontrol + titik kurva
42     titikbezier = [] # titik kurva
43     xawal = [] # x titik yang masuk di awal
44     yawal = [] # y titik yang masuk di awal
45
46     n = int(input("Masukkan banyak titik: "))
47     garis = n-1

```



```

src > dncpy > Ngaris_BForce
26
27
28 def Ngaris_BForce(algo):
29     titikawal = [] # titik PO
30     titikakhir = [] # titik PO
31     titikbantu = [] # semua titik kontrol + titik kurva
32     titikbezier = [] # titik kurva
33     xawal = [] # x titik yang masuk di awal
34     yawal = [] # y titik yang masuk di awal
35
36     n = int(input("Masukkan banyak titik: "))
37     garis = n-1
38
39     for i in range(n):
40         titik = []
41         if i == 0:
42             titik.append(float(input("input x: ")))
43             xawal.append(titik[-1])
44             titik.append(float(input("input y: ")))
45             yawal.append(titik[-1])
46             titikawal.append(titik)
47             titikbantu.append(titik)
48         elif i == n-1:
49             titik.append(float(input("input x: ")))
50             xawal.append(titik[-1])
51             titik.append(float(input("input y: ")))
52             yawal.append(titik[-1])
53             titikakhir.append(titik)
54             titikbantu.append(titik)
55         else:
56             titik.append(float(input("input x: ")))
57             xawal.append(titik[-1])
58             titik.append(float(input("input y: ")))
59             yawal.append(titik[-1])
60             titikbantu.append(titik)
61
62     if algo == '3':
63         x = []
64         x.append(titikawal[0][0])
65         y = []
66         y.append(titikawal[0][1])
67         iterasi = int(input("Masukkan iterasi: "))
68         mulai = time.time()
69         iterasi = 2*iterasi - 1
70         t = 1/iterasi
71         while t < 1:
72             titik = bruteforce.bezierbruteforce(titikbantu, t)
73             x.append(titik[0])
74             y.append(titik[1])
75             t += 1/iterasi
76         akhir = time.time()

```

```

src > dncpy > Ngaris_BForce
28 def Ngaris_BForce(algo):
29
30     if algo == '3':
31         x = []
32         x.append(titikawal[0][0])
33         y = []
34         y.append(titikawal[0][1])
35         iterasi = int(input("Masukkan iterasi: "))
36         mulai = time.time()
37         iterasi = 2*iterasi - 1
38         t = 1/iterasi
39         while t < 1:
40             titik = bruteforce.bezierbruteforce(titikbantu, t)
41             x.append(titik[0])
42             y.append(titik[1])
43             t += 1/iterasi
44         akhir = time.time()
45
46     if algo == '2':
47         total_iterasi_animasi = 0
48         iterasi = int(input("Masukkan iterasi: "))
49         pakai_animasi = input("Inginkan menggunakan animasi (Y/N): ")
50         mulai = time.time()
51         for i in range(iterasi):
52             titikbezier = []
53             titikdikpakai = 0
54             titikbantutemp = titikbantu
55             titikbantu = []
56             while (titikdikpakai < len(titikbantutemp)-1):
57                 temp = []
58                 awal = titikdikpakai
59                 titikdikpakai += garis
60                 # print(titikbantutemp[awal:titikdikpakai+1])
61                 kumpulantitik = slicing(titikbantutemp, awal, titikdikpakai)
62                 kumpulantitik = list_midpoint(kumpulantitik)
63                 ulang = 0
64                 while len(kumpulantitik) > 1:
65                     temp.insert(ulang, kumpulantitik[0])
66                     temp.insert(len(temp)-ulang, kumpulantitik[len(kumpulantitik)-1])
67                     ulang += 1
68                     kumpulantitik = list_midpoint(kumpulantitik)
69                     temp.insert(ulang, kumpulantitik[0])
70                     temp.remove(titikbantutemp[titikdikpakai])
71                     titikbantu = expandarray(titikbantu, temp)
72                     titikbantu.insert(0, titikawal[0])
73                 if pakai_animasi == 'Y':
74                     for i in range(0, len(titikbantu), garis):

```

```

src > dncpy > Ngaris_BForce
28 def Ngaris_BForce(algo):
29     # print(titikbantutemp[awal:titikdikpakai+1])
30     kumpulantitik = slicing(titikbantutemp, awal, titikdikpakai)
31     kumpulantitik = list_midpoint(kumpulantitik)
32     ulang = 0
33     while len(kumpulantitik) > 1:
34         temp.insert(ulang, kumpulantitik[0])
35         temp.insert(len(temp)-ulang, kumpulantitik[len(kumpulantitik)-1])
36         ulang += 1
37         kumpulantitik = list_midpoint(kumpulantitik)
38         temp.insert(ulang, kumpulantitik[0])
39         temp.remove(titikbantutemp[titikdikpakai])
40         titikbantu = expandarray(titikbantu, temp)
41         titikbantu.insert(0, titikawal[0])
42     if pakai_animasi == 'Y':
43         for i in range(0, len(titikbantu), garis):
44             titikbezier.append(titikbantu[i])
45             animasi(titikbezier, xawal, yawal)
46         total_iterasi_animasi += 1
47     else:
48         for i in range(0, len(titikbantu), garis):
49             titikbezier.append(titikbantu[i])
50
51     x = []
52     y = []
53     for i in range(len(titikbezier)):
54         x.append(titikbezier[i][0])
55         y.append(titikbezier[i][1])
56     akhir = time.time()
57
58     plt.clf()
59     plt.title("Kurva Bezier")
60     plt.xlabel(x, marker = 'o', label = 'kurva bezier')
61     plt.ylabel(y, marker = 'o', label = 'titik kontrol')
62     plt.legend()
63     if algo == '2':
64         # Mengurangi pengurangan karena ada galat ketika menampilkan animasi
65         if pakai_animasi == 'Y':
66             runtime = (akhir - mulai - (total_iterasi_animasi*0.5)) * 1000
67         else:
68             runtime = (akhir - mulai) * 1000
69     else:
70         runtime = (akhir - mulai) * 1000
71     print("Waktu program berjalan: {:.2f} milliseconds".format(runtime))
72     plt.show()

```

- main.py

```

src > manpy > @ main
1 from divide_and_conquer import *
2 from dvc import *
3 import time
4
5 def input_titik():
6     titik = []
7     for i in range(3):
8         if i == 0:
9             x = float(input("Masukkan koordinat x untuk titik kontrol awal: "))
10            y = float(input("Masukkan koordinat y untuk titik kontrol awal: "))
11        elif i == 1:
12            x = float(input("Masukkan koordinat x untuk titik kontrol antara: "))
13            y = float(input("Masukkan koordinat y untuk titik kontrol antara: "))
14        else:
15            x = float(input("Masukkan koordinat x untuk titik kontrol akhir: "))
16            y = float(input("Masukkan koordinat y untuk titik kontrol akhir: "))
17        titik.append((x, y))
18    return titik
19
20 def pilihan():
21     while True:
22         print("Pilih opsi:")
23         print("1. Divide and Conquer 2 Garis (3 Titik)")
24         print("2. Divide and Conquer N Garis (N+1 Titik)")
25         print("3. BruteForce")
26
27         pilihan = input("Masukkan pilihan Anda (1/2/3): ")
28
29         if pilihan in ['1', '2', '3']:
30             break
31         else:
32             print("Pilihan tidak valid. Silakan masukkan 1, 2, atau 3.")
33
34     return pilihan
35
36 def main():
37     algo = pilihan()
38     print("")
39     if algo == '1':
40         titik = input_titik()
41         pakai_animasi = input("Ingin menggunakan animasi (Y/N)? ")
42         iterasi = int(input("Masukkan jumlah iterasi: "))
43         arr = []
44         total_iterasi_animasi = []
45         mulai = time.time()
46         kurva_bezier(titik[0], titik[1], titik[2], iterasi, arr, titik[0], titik[1], titik[2], total_iterasi_animasi, pakai_animasi)
47         selesai = time.time()
48         if pakai_animasi == 'Y':
49             print("Waktu program berjalan: {:.2f} milliseconds".format((selesai - mulai - (len(total_iterasi_animasi) * 0.5)) * 1000))
50         else:
51             plot_kontrol(titik[0], titik[1], titik[2])
52             plot_kurva_bezier(arr)
53             print("Waktu program berjalan: {:.2f} milliseconds".format((selesai - mulai) * 1000))
54             plt.show()
55
56     elif algo == '2' or algo == '3':
57         ngaris_Bforce(algo)
58
59 if __name__ == "__main__":
60     main()

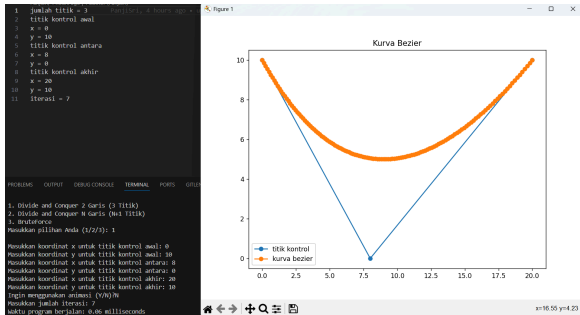
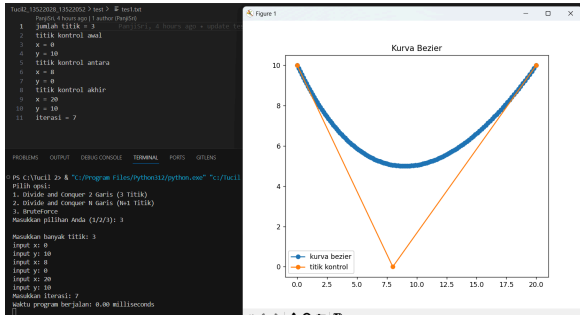
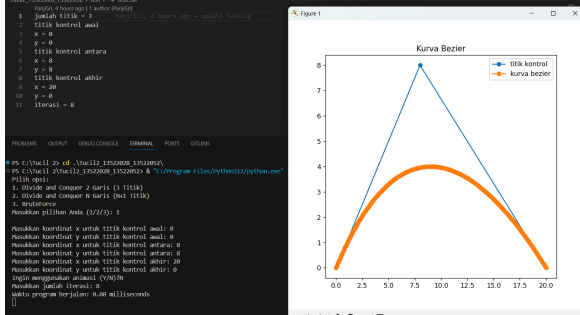
```

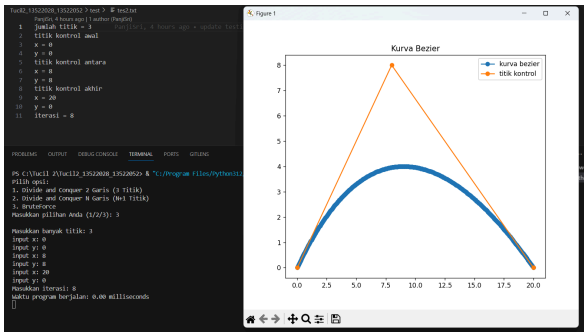
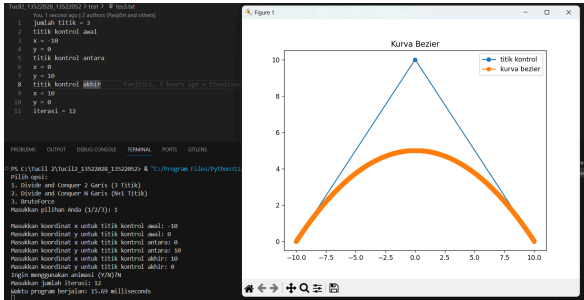
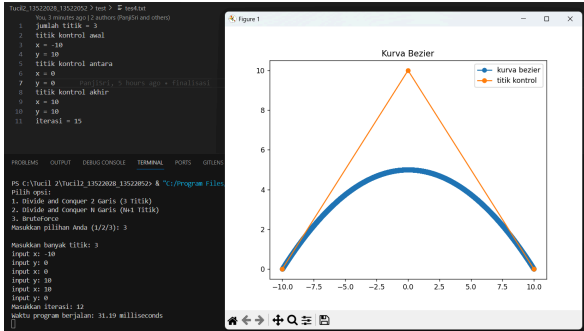
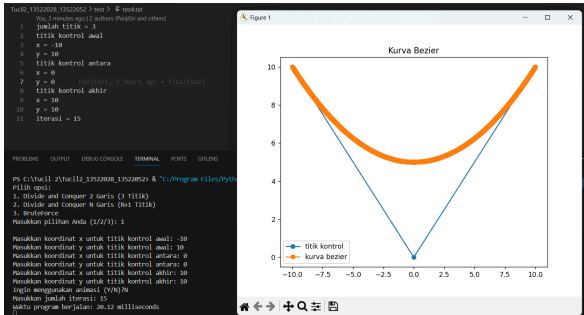
```

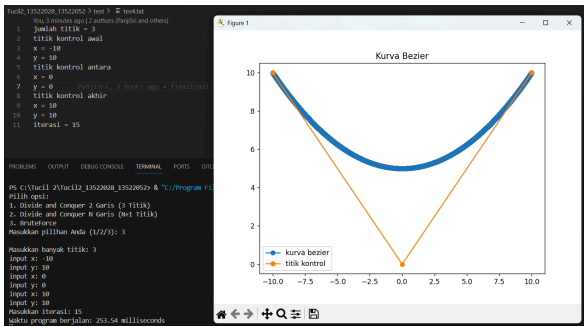
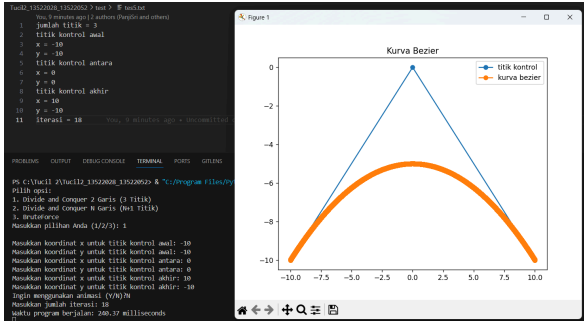
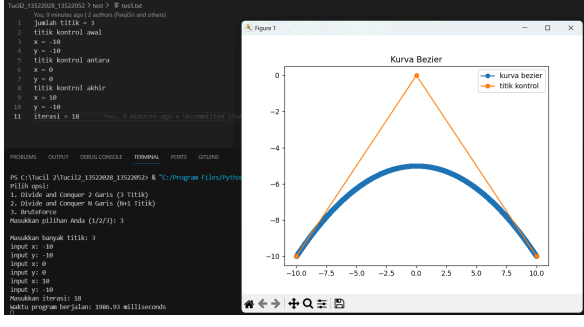
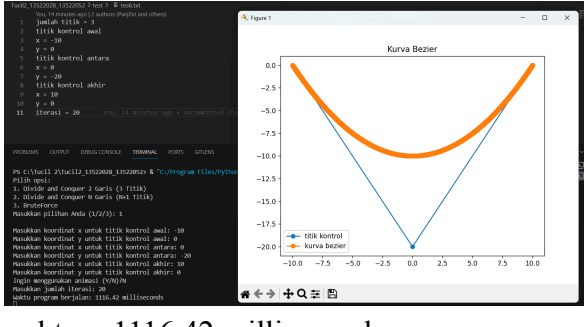
40 def main():
41     algo = pilihan()
42     print("")
43     if algo == '1':
44         titik = input_titik()
45         pakai_animasi = input("Ingin menggunakan animasi (Y/N)? ")
46         iterasi = int(input("Masukkan jumlah iterasi: "))
47         arr = []
48         total_iterasi_animasi = []
49         mulai = time.time()
50         kurva_bezier(titik[0], titik[1], titik[2], iterasi, arr, titik[0], titik[1], titik[2], total_iterasi_animasi, pakai_animasi)
51         selesai = time.time()
52         if pakai_animasi == 'Y':
53             print("Waktu program berjalan: {:.2f} milliseconds".format((selesai - mulai - (len(total_iterasi_animasi) * 0.5)) * 1000))
54         else:
55             plot_kontrol(titik[0], titik[1], titik[2])
56             plot_kurva_bezier(arr)
57             print("Waktu program berjalan: {:.2f} milliseconds".format((selesai - mulai) * 1000))
58             plt.show()
59
60     elif algo == '2' or algo == '3':
61         ngaris_Bforce(algo)
62
63 if __name__ == "__main__":
64     main()

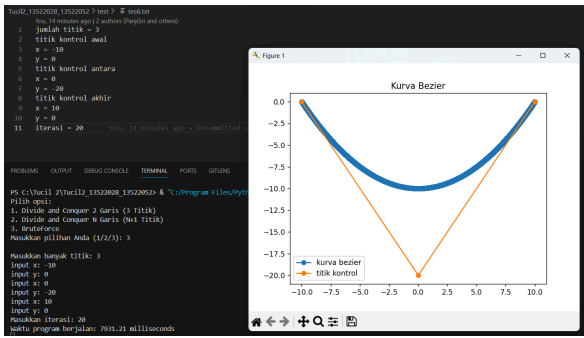
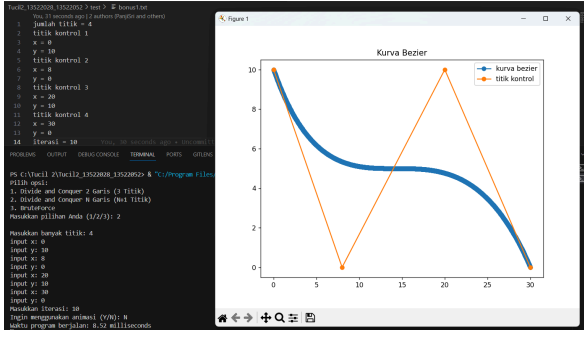
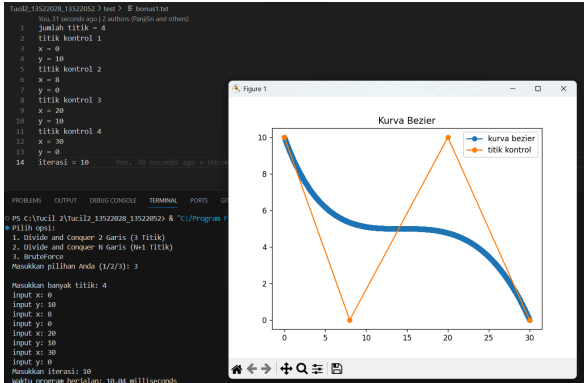
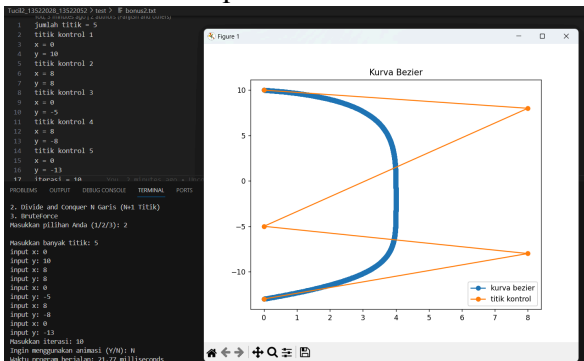
```

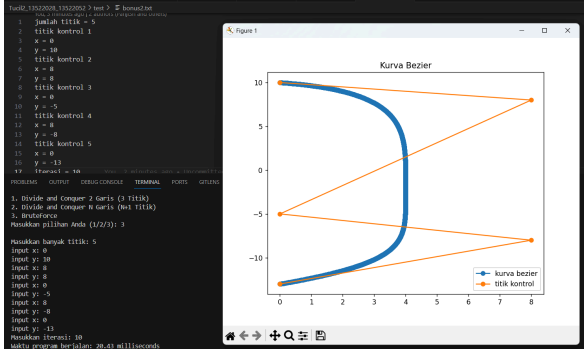
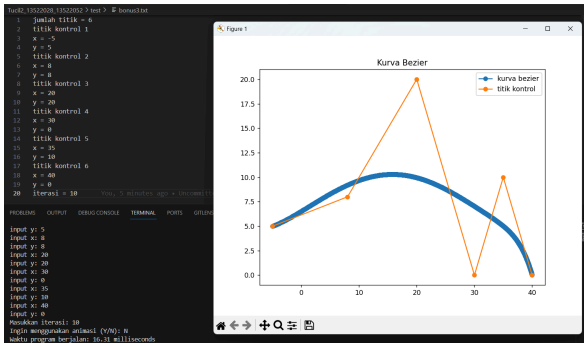
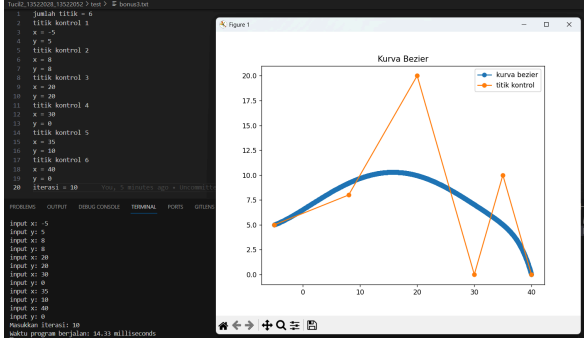
D. Input dan Output

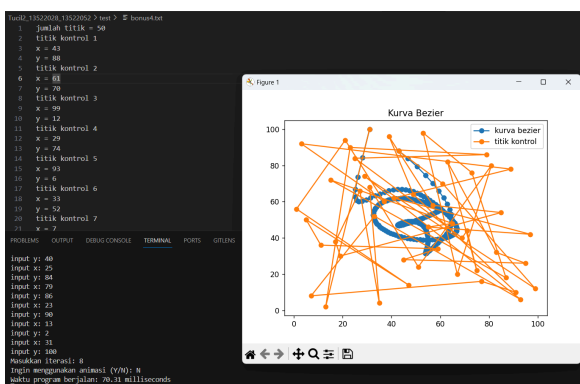
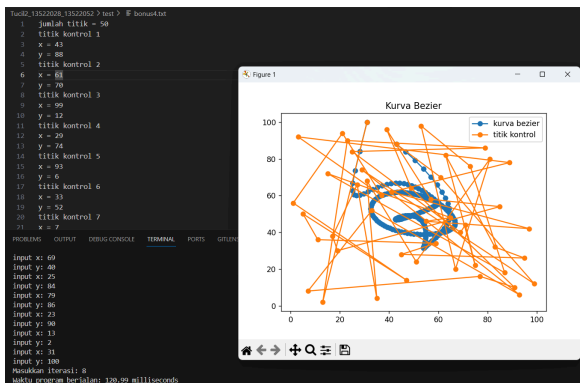
No.	Input	Output
1.	<p>tes1.txt</p> <p>jumlah titik = 3 titik kontrol awal $x = 0$ $y = 10$ titik kontrol antara $x = 8$ $y = 0$ titik kontrol akhir $x = 20$ $y = 10$ iterasi = 7</p>	<p>-Divide and Conquer</p>  <p>waktu : 0.06 milliseconds</p> <p>-BruteForce</p>  <p>waktu = 0.00 milliseconds</p>
2.	<p>tes2.txt</p> <p>jumlah titik = 3 titik kontrol awal $x = 0$ $y = 0$ titik kontrol antara $x = 8$ $y = 8$ titik kontrol akhir $x = 20$ $y = 0$ iterasi = 8</p>	<p>-Divide and Conquer</p>  <p>waktu = 0.00 milliseconds</p> <p>-BruteForce</p>

		 <p>waktu = 0.00 milliseconds</p>
3.	<p>Tes3.txt</p> <p>jumlah titik = 3 titik kontrol awal $x = -10$ $y = 0$ titik kontrol antara $x = 0$ $y = 10$ titik kontrol akhir $x = 10$ $y = 0$ iterasi = 12</p>	<p>-Divide and Conquer</p>  <p>waktu = 15.69 milliseconds</p> <p>-BruteForce</p>  <p>waktu = 31.19 milliseconds</p>
4.	<p>tes4.txt</p> <p>jumlah titik = 3 titik kontrol awal $x = -10$ $y = 10$ titik kontrol antara $x = 0$ $y = 0$ titik kontrol akhir $x = 10$ $y = 10$</p>	<p>-Divide and Conquer</p>  <p>waktu = 20.12 milliseconds</p> <p>-BruteForce</p>

	<p>iterasi = 15</p>	 <p>waktu = 253.54 milliseconds</p>
5.	<p>tes5.txt</p> <p>jumlah titik = 3 titik kontrol awal $x = -10$ $y = -10$ titik kontrol antara $x = 0$ $y = 0$ titik kontrol akhir $x = 10$ $y = -10$ iterasi = 18</p>	<p>-Divide and Conquer</p>  <p>waktu = 240.37 milliseconds</p> <p>-BruteForce</p>  <p>waktu = 1896.93 milliseconds</p>
6.	<p>tes6.txt</p> <p>jumlah titik = 3 titik kontrol awal $x = -10$ $y = 0$ titik kontrol antara $x = 0$ $y = -20$ titik kontrol akhir $x = 10$ $y = 0$ iterasi = 20</p>	<p>-Divide and Conquer</p>  <p>waktu = 1116.42 milliseconds</p> <p>-BruteForce</p>

		 <p>waktu = 7931.21 milliseconds</p>
7	<p>bonus1.txt</p> <p>jumlah titik = 4 titik kontrol 1 $x = 0$ $y = 10$ titik kontrol 2 $x = 8$ $y = 0$ titik kontrol 3 $x = 20$ $y = 10$ titik kontrol 4 $x = 30$ $y = 0$ iterasi = 10</p>	<p>-Divide and Conquer</p>  <p>waktu = 8.52 milliseconds</p> <p>-BruteForce</p>  <p>waktu = 10.04 milliseconds</p>
8	<p>Bonus2.txt</p> <p>jumlah titik = 5 titik kontrol 1 $x = 0$ $y = 10$ titik kontrol 2 $x = 8$ $y = 8$ titik kontrol 3 $x = 0$</p>	<p>-Divide and Conquer</p>  <p>waktu = 31.67 milliseconds</p>

	<p> $y = -5$ titik kontrol 4 $x = 8$ $y = -8$ titik kontrol 5 $x = 0$ $y = -13$ iterasi = 10 </p>	<p>waktu = 21.77 milliseconds</p> <p>-BruteForce</p>  <p>waktu = 20.43 milliseconds</p>
9	<p>bonus3.txt</p> <p> jumlah titik = 6 titik kontrol 1 $x = -5$ $y = 5$ titik kontrol 2 $x = 8$ $y = 8$ titik kontrol 3 $x = 20$ $y = 20$ titik kontrol 4 $x = 30$ $y = 0$ titik kontrol 5 $x = 35$ $y = 10$ titik kontrol 6 $x = 40$ $y = 0$ iterasi = 10 </p>	<p>-Divide and Conquer</p>  <p>waktu = 16.31 milliseconds</p> <p>-BruteForce</p>  <p>waktu = 14.33 milliseconds</p>
10	<p>bonus4.txt</p> <p> jumlah titik = 50 titik kontrol 1 $x = 43$ $y = 88$ titik kontrol 2 $x = 61$ $y = 70$ titik kontrol 3 $x = 99$ </p>	<p>-Divide and Conquer</p>

	<p> $y = 12$ titik kontrol 4 $x = 29$ $y = 74$ titik kontrol 5 $x = 93$ $y = 6$ dst. (full titik dapat dilihat di folder test) iterasi = 8 </p>	 <p> Waktu = 70.31 milliseconds -BruteForce </p>  <p> Waktu = 120.99 milliseconds </p>
--	--	--

Tabel 2. Input dan Output Program

E. Analisis Brute Force dan Divide and Conquer

Dari 10 *test case* yang digunakan, *brute force* efektif digunakan ketika jumlah titik dan jumlah iterasi sedikit, hal ini dikarenakan kompleksitas waktu *divide and conquer* adalah $O(n \log n)$, sementara *brute force* $O(n!)$ jika dilihat dari jumlah titik, bisa dilihat pada bonus2.txt dan bonus3.txt, waktu yang dihasilkan *brute force* dan *divide and conquer* berbeda tipis, namun ketika titik diperbanyak sesuai bonus4.txt terjadi perbedaan waktu yang sangat signifikan.

F. Algoritma Divide and Conquer untuk N titik ($N > 3$)

Untuk menyelesaikan ini, kurva bezier n titik ini dibentuk secara iteratif, cara pembentukannya adalah:

1. Di awal, semua titik kontrol yang berjumlah n disimpan

2. Dari semua titik kontrol akan terus dilakukan looping sehingga terbentuk 1 titik bezier, dalam setiap looping, titik yang berada paling kanan dan paling kiri akan disimpan sebagai titik kontrol baru untuk iterasi berikutnya.
3. Jika titik kontrol berjumlah lebih besar dari n , dengan pola yang ditemukan, maka langkah 2 akan dilakukan dengan menggunakan n titik hingga titik kontrol dalam *array* digunakan seluruhnya

Contoh:

Jumlah titik = 4

Array titik kontrol = [P0,P1,P2,P3,P4,P5,P6]

Set 1 = [P0,P1,P2,P3]

Set 2 = [P3,P4,P5,P6]

Titik kontrol 2 set digabung dan akan menggantikan titik kontrol lama, lakukan hingga iterasi selesai.

4. Dengan menggunakan pola unik yang ditemukan yaitu dalam setiap titik kontrol, titik kelipatan $n-1$ (dengan asumsi titik awal dihitung sebagai titik ke-0) merupakan titik bezier, maka dari array titik kontrol dapat ditemukan kumpulan titik yang bisa membentuk kurva bezier

G. Repository

Link Repository dari Tugas Kecil 02 IF2211 Strategi Algoritma Panji Sri Kuncara Wisma dan Haikal Assyauqi adalah sebagai berikut.

https://github.com/PanjiSri/Tucil2_13522028_13522052.git