

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

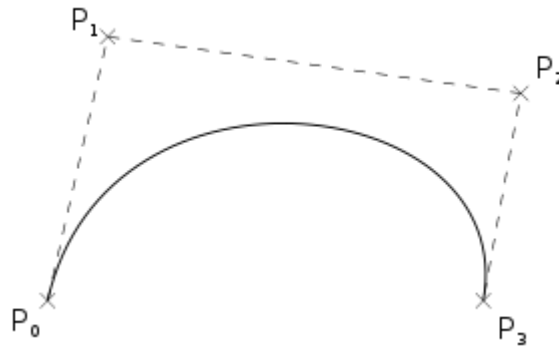
DAFTAR ISI.....	1
PENGECEKAN PROGRAM.....	2
DESKRIPSI MASALAH.....	3
A. Algoritma Divide and Conquer Tiga Titik atau Dua Garis.....	4
B. Algoritma Brute Force.....	5
C. Source Code.....	6
D. Input dan Output.....	10
E. Analisis Brute Force dan Divide and Conquer.....	15
F. Algoritma Divide and Conquer 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].	✓	
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 kedalam 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

```

10 > @ dncd_andongwary > @ plot_kurva_no_animasi
11 import matplotlib.pyplot as plt
12
13 def titik_tengah(titik_awal, titik_akhir):
14     hasil = []
15     x_tengah = (titik_awal[0] + titik_akhir[0]) / 2
16     y_tengah = (titik_awal[1] + titik_akhir[1]) / 2
17     hasil.append(x_tengah)
18     hasil.append(y_tengah)
19     return hasil
20
21 def kurva_bezier(titik_1, titik_2, titik_3, iterasi, arr, kontrol1, kontrol2, kontrol3, total_iterasi_animasi, pakai_animasi):
22     if iterasi == 1:
23         tengah_1 = titik_tengah(titik_1, titik_2)
24         tengah_2 = titik_tengah(titik_2, titik_3)
25         tengah_dari_tengah = titik_tengah(tengah_1, tengah_2)
26
27         if arr == []:
28             arr.append(titik_1)
29             arr.append(tengah_dari_tengah)
30             arr.append(titik_3)
31         else:
32             if arr[-1] == titik_1:
33                 arr.append(tengah_dari_tengah)
34                 arr.append(titik_3)
35             else:
36                 arr.append(titik_1)
37                 arr.append(tengah_dari_tengah)
38                 arr.append(titik_3)
39
40             if pakai_animasi == 'Y':
41                 plt.clf()
42                 plt.plot(kontrol1, kontrol2, kontrol3)
43                 total_iterasi_animasi.append(1)
44                 plot_kurva(arr)
45             else:
46                 tengah_1 = titik_tengah(titik_1, titik_2)
47                 tengah_2 = titik_tengah(titik_2, titik_3)
48                 tengah_dari_tengah = titik_tengah(tengah_1, tengah_2)
49
50                 #ini buat bagian kiri
51                 kurva_bezier(titik_1, tengah_1, tengah_dari_tengah, iterasi-1, arr, kontrol1, kontrol2, kontrol3, total_iterasi_animasi, pakai_animasi)
52
53                 #ini buat bagian kanan
54                 kurva_bezier(tengah_dari_tengah, tengah_2, titik_3, iterasi-1, arr, kontrol1, kontrol2, kontrol3, total_iterasi_animasi, pakai_animasi)
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

```

101 def plot_kurva(arr):
102     titik_x = []
103     titik_y = []
104     for elemen in arr:
105         titik_x.append(elemen[0])
106         titik_y.append(elemen[1])
107     plt.title("Kurva Bezier")
108     plt.plot(titik_x, titik_y, marker = 'o', label = 'kurva bezier')
109     plt.legend()
110     plt.pause(0.5)
111
112 def plot_kurva_no_animasi(arr):
113     titik_x = []
114     titik_y = []
115     for elemen in arr:
116         titik_x.append(elemen[0])
117         titik_y.append(elemen[1])
118     plt.title("Kurva Bezier")
119     plt.plot(titik_x, titik_y, marker = 'o', label = 'kurva bezier')
120     plt.legend()
121
122 def plot_kontrol(titik_awal, titik_tengah, titik_akhir):
123     x_kontrol = [titik_awal[0], titik_tengah[0], titik_akhir[0]]
124     y_kontrol = [titik_awal[1], titik_tengah[1], titik_akhir[1]]
125     plt.plot(x_kontrol, y_kontrol, linestyle="--", marker = 'o', label = 'titik kontrol')
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

- dnc.py

```

10 > dnc.py > Ngasis_BForce
11 import matplotlib.pyplot as plt
12 import bruteForce
13 import time
14
15 def midpoint(x1,y1,x2,y2):
16     koordinat = []
17     koordinat.append((x1+x2) / 2)
18     koordinat.append((y1+y2) / 2)
19     return koordinat
20
21 def list_midpoint(list_of_titik):
22     daftar_midpoint = []
23     for i in range(len(list_of_titik)-1):
24         daftar_midpoint.append(midpoint(list_of_titik[i][0], list_of_titik[i+1][0], list_of_titik[i+1][1], list_of_titik[i+1][1]))
25     return daftar_midpoint
26
27 def slicing(list, awal, akhir):
28     array = []
29     for i in range(awal,akhir):
30         array.append(list[i])
31     return array
32
33 def expandarray(array1, array2):
34     for i in range(len(array2)):
35         array1.append(array2[i])
36     return array1
37
38 def animasi(array,xawal,yawal):
39     plt.clf()
40     plt.scatter([i[0] for i in array],[i[1] for i in array])
41     plt.title("Kurva Bezier")
42     plt.plot([i[0] for i in array],[i[1] for i in array], marker = 'o', label = 'kurva bezier')
43     plt.plot(xawal, yawal, marker = 'o', label = 'titik kontrol')
44     plt.legend()
45     plt.pause(0.5)
46
47 def Ngasis_BForce(algo):
48     titikawal = [] #titik p0
49     titikakhir = [] #titik p3
50     titikbantu = [] #semua titik kontrol + titik kurva
51     titikbezier = [] # titik kurva
52     xawal = [] # x titik yang masuk di awal
53     yawal = [] # y titik yang masuk di awal
54
55     n = int(input("Masukkan banyak titik: "))
56     garis = n-1
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```



```

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()
77
78     if algo == '2':
79         total_iterasi_animasi = 0
80         iterasi = int(input("Masukkan iterasi: "))
81         pakai_animasi = input("Inginkan menggunakan animasi (Y/N): ")
82         mulai = time.time()
83         for i in range(iterasi):
84             titikbezier = []
85             titikdikpakai = 0
86             titikbantutemp = titikbantu
87             titikbantu = []
88             while (titikdikpakai < len(titikbantutemp)-1):
89                 temp = []
90                 awal = titikdikpakai
91                 titikdikpakai += garis
92                 # print(titikbantutemp[awal:titikdikpakai+1])
93                 kumpulantitik = slicing(titikbantutemp, awal, titikdikpakai)
94                 kumpulantitik = list_midpoint(kumpulantitik)
95                 ulang = 0
96                 while len(kumpulantitik) > 1:
97                     temp.insert(ulang, kumpulantitik[0])
98                     temp.insert(len(temp)-ulang, kumpulantitik[len(kumpulantitik)-1])
99                     ulang += 1
100                 kumpulantitik = list_midpoint(kumpulantitik)
101                 temp.insert(ulang, kumpulantitik[0])
102                 temp.append(titikbantutemp[titikdikpakai])
103                 titikbantu = expandarray(titikbantu, temp)
104                 titikbantu.insert(0, titikawal[0])
105                 if pakai_animasi == 'Y':
106                     for i in range(0, len(titikbantu), garis):

```

```

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

```

```

src > dncpy > Ngaris_BForce
38 def Ngaris_BForce(algo):
39
40     if algo == '3':
41         x = []
42         x.append(titikawal[0][0])
43         y = []
44         y.append(titikawal[0][1])
45         iterasi = int(input("Masukkan iterasi: "))
46         mulai = time.time()
47         iterasi = 2*iterasi - 1
48         t = 1/iterasi
49         while t < 1:
50             titik = bruteforce.bezierbruteforce(titikbantu, t)
51             x.append(titik[0])
52             y.append(titik[1])
53             t += 1/iterasi
54         akhir = time.time()
55
56     if algo == '2':
57         total_iterasi_animasi = 0
58         iterasi = int(input("Masukkan iterasi: "))
59         pakai_animasi = input("Inginkan menggunakan animasi (Y/N): ")
60         mulai = time.time()
61         for i in range(iterasi):
62             titikbezier = []
63             titikdikpakai = 0
64             titikbantutemp = titikbantu
65             titikbantu = []
66             while (titikdikpakai < len(titikbantutemp)-1):
67                 temp = []
68                 awal = titikdikpakai
69                 titikdikpakai += garis
70                 # print(titikbantutemp[awal:titikdikpakai+1])
71                 kumpulantitik = slicing(titikbantutemp, awal, titikdikpakai)
72                 kumpulantitik = list_midpoint(kumpulantitik)
73                 ulang = 0
74                 while len(kumpulantitik) > 1:
75                     temp.insert(ulang, kumpulantitik[0])
76                     temp.insert(len(temp)-ulang, kumpulantitik[len(kumpulantitik)-1])
77                     ulang += 1
78                 kumpulantitik = list_midpoint(kumpulantitik)
79                 temp.insert(ulang, kumpulantitik[0])
80                 temp.append(titikbantutemp[titikdikpakai])
81                 titikbantu = expandarray(titikbantu, temp)
82                 titikbantu.insert(0, titikawal[0])
83                 if pakai_animasi == 'Y':
84                     for i in range(0, len(titikbantu), garis):
85                         titikbezier.append(titikbantu[i])
86                         animasi(titikbezier, xawal, yawal)
87                     total_iterasi_animasi += 1
88                 else:
89                     for i in range(0, len(titikbantu), garis):
90                         titikbezier.append(titikbantu[i])
91
92     x = []
93     y = []
94     for i in range(len(titikbezier)):
95         x.append(titikbezier[i][0])
96         y.append(titikbezier[i][1])
97     akhir = time.time()
98
99     plt.clf()
100     plt.title("Kurva Bezier")
101     plt.xlabel(x, marker = 'o', label = 'kurva bezier')
102     plt.ylabel(y, marker = 'o', label = 'titik kontrol')
103     plt.plot(xawal, yawal, marker = 'o', label = 'titik kontrol')
104     plt.legend()
105     if algo == '2':
106         # Mengurangi pengurangan karena ada galat ketika menampilkan animasi
107         if pakai_animasi == 'Y':
108             runtime = (akhir - mulai - (total_iterasi_animasi*0.5)) * 1000
109         else:
110             runtime = (akhir - mulai) * 1000
111     else:
112         runtime = (akhir - mulai) * 1000
113     print("waktu program berjalan: {:.2f} milliseconds".format(runtime))
114     plt.show()

```

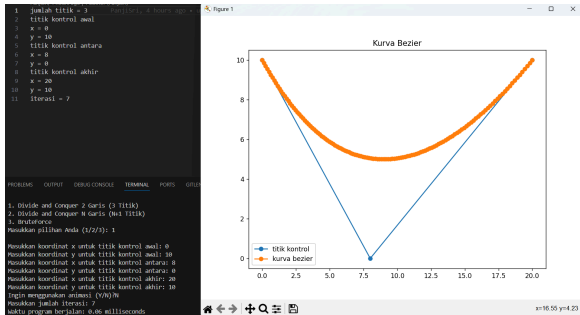
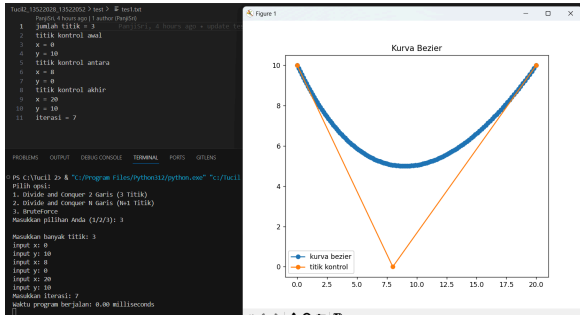
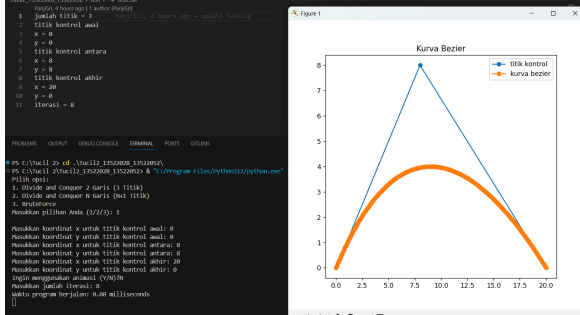
- main.py

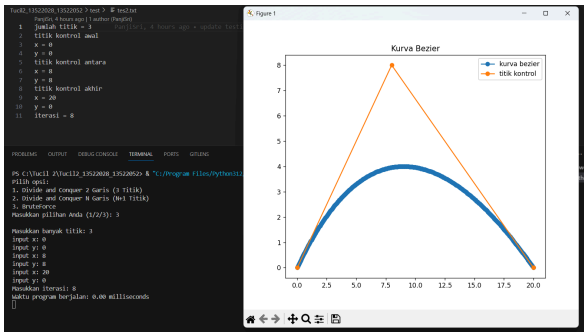
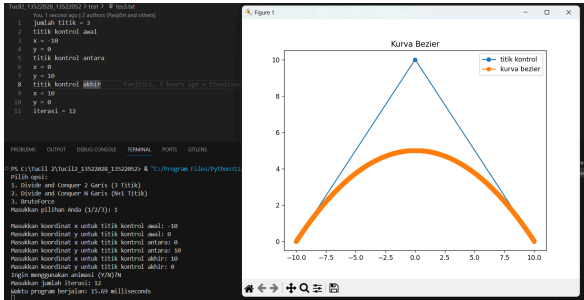
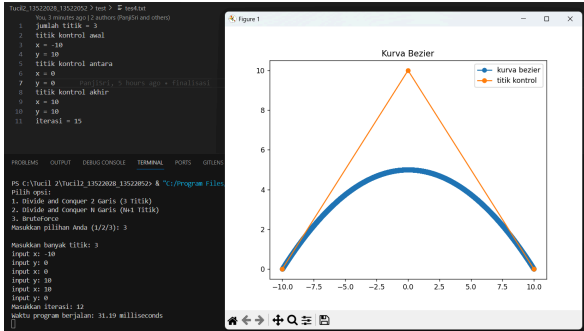
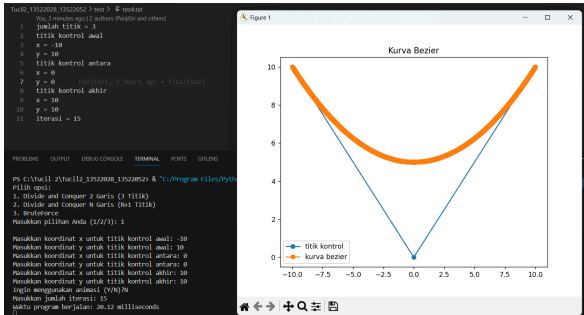
```

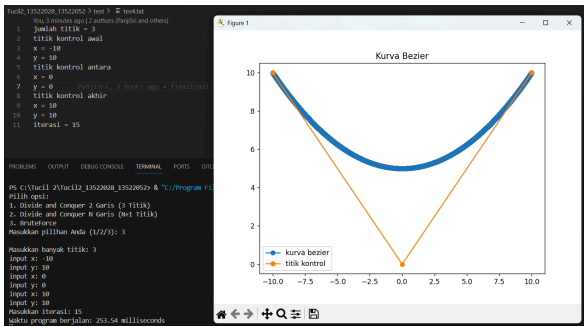
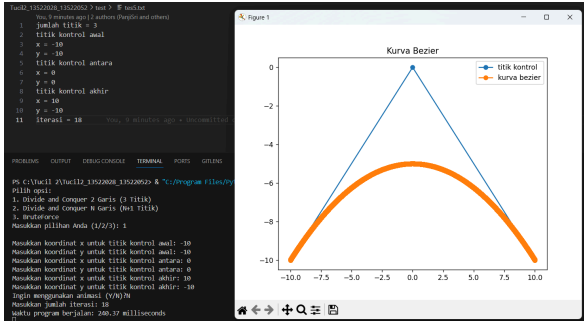
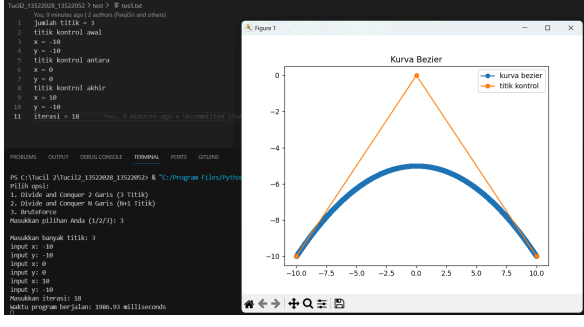
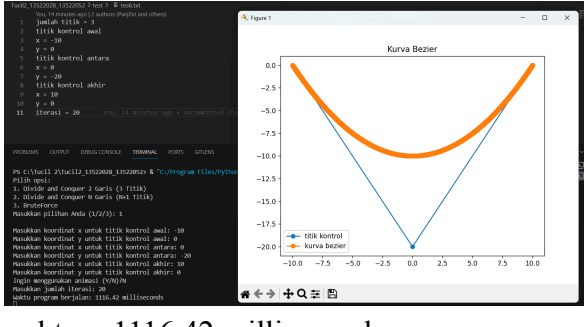
src > manpy > @ main
1 from divide_and_conquer import *
2 from dvc import *
3 import time
4
5 def input_titik():
6     titik = []
7     for i in range(3):
8         if i == 0:
9             x = float(input("Masukkan koordinat x untuk titik kontrol awal: "))
10            y = float(input("Masukkan koordinat 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()

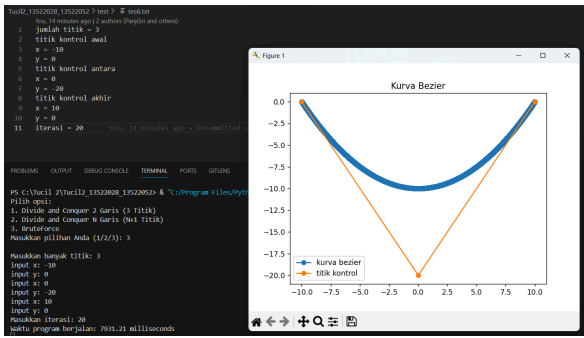
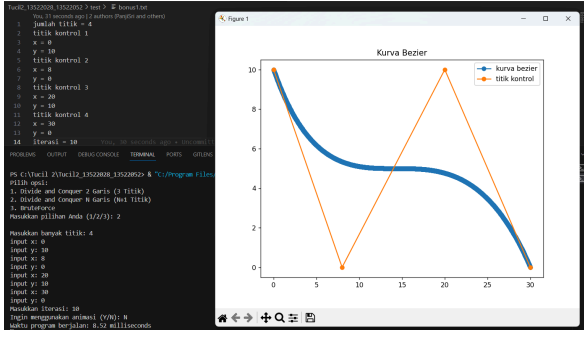
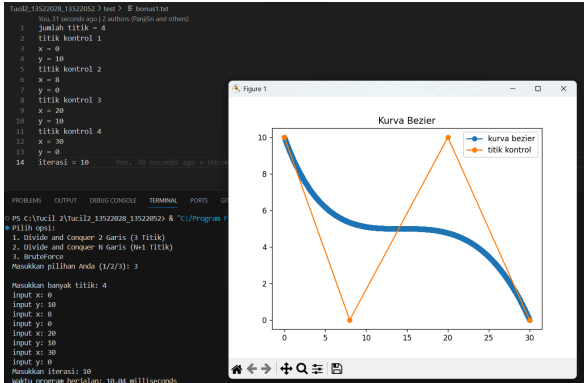
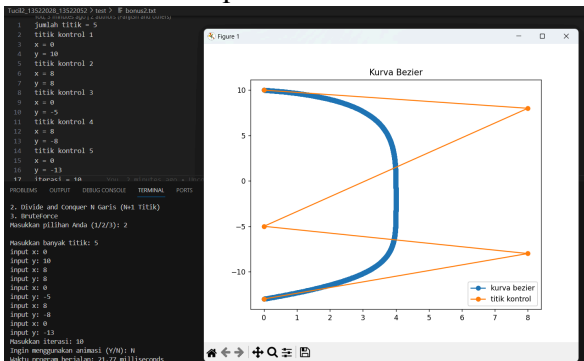
```

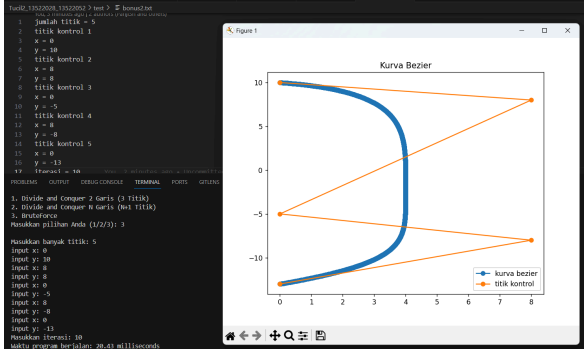
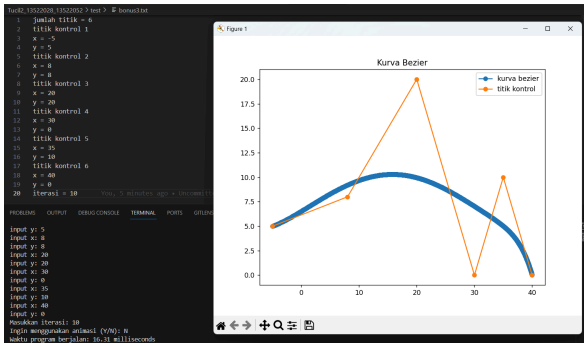
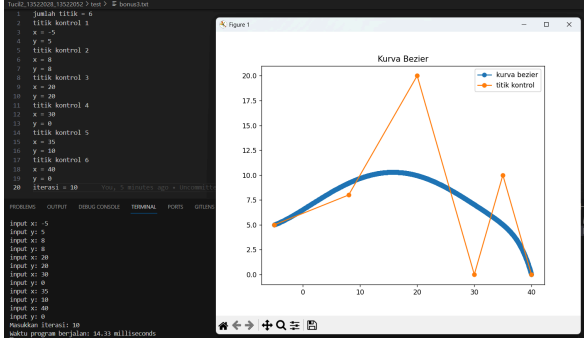
D. Input dan Output

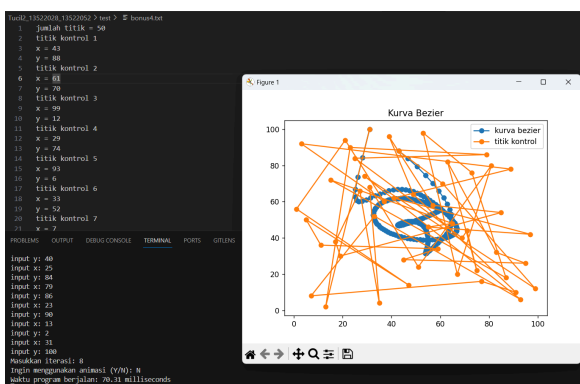
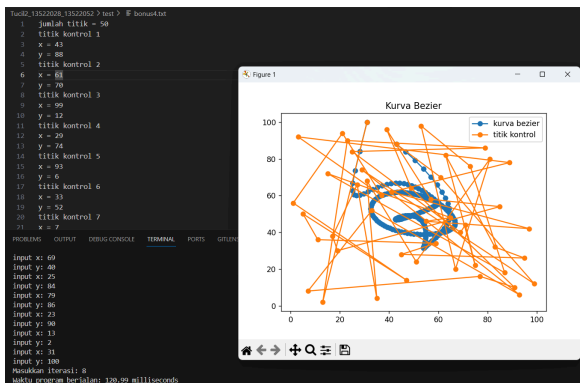
No.	Input	Output
1.	tes1.txt jumlah titik = 3 titik kontrol awal $x = 0$ $y = 10$ titik kontrol antara $x = 8$ $y = 0$ titik kontrol akhir $x = 20$ $y = 10$ iterasi = 7	<div>-Divide and Conquer</div>  <p>waktu : 0.06 milliseconds</p> <div>-BruteForce</div>  <p>waktu = 0.00 milliseconds</p>
2.	tes2.txt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

		 <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> $y = -5$ 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</p> <p>-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