LAPORAN TUGAS KECIL 1 IF2211 STRATEGI ALGORITMA

SEMESTER II TAHUN 2022/2023

Mencari Pasangan Titik Terdekat dengan 3D dengan Algoritma *Divide and Conquer*

Disusun oleh :

Alex Sander **13521061**

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

BANDUNG

2022

DAFTAR ISI

BAB I DESKRIPSI MASALAH..3

BAB II ALGORITMA..4

BAB III SOURCE CODE..5

BAB IV TEST CASE..16

BAB V TABEL...27

Link Repository..28

BAB I

DESKRIPSI MASALAH

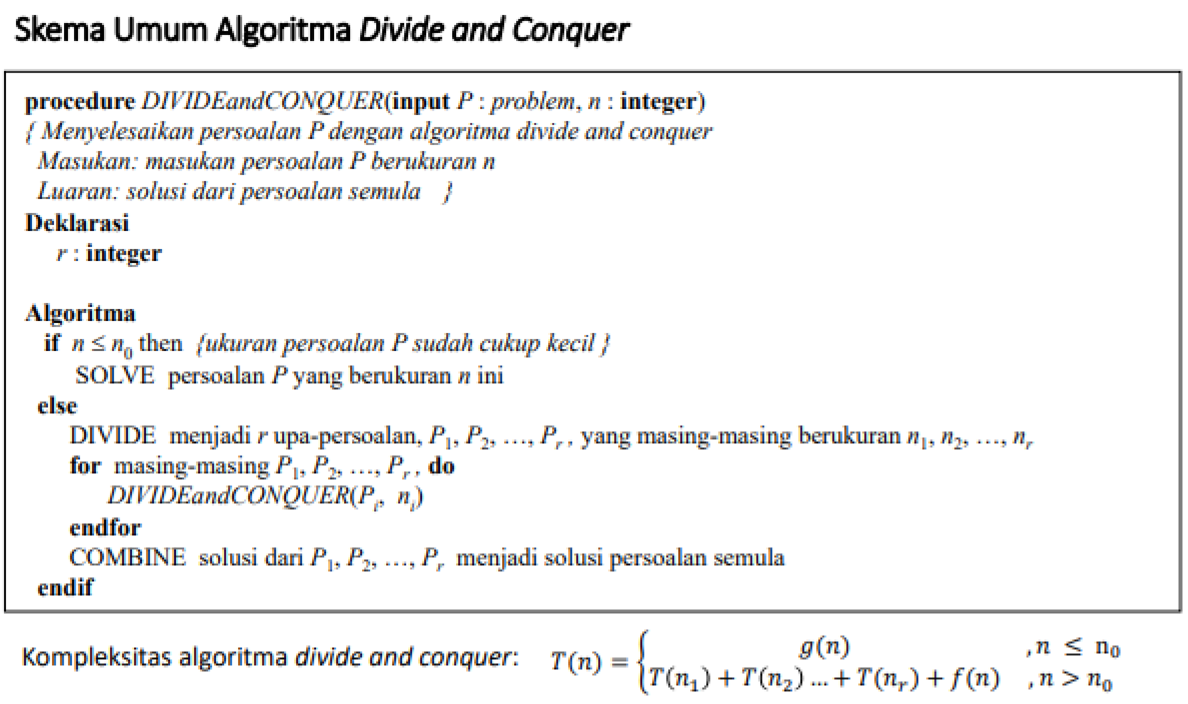
Mencari sepasang titik terdekat dengan Algoritma Divide and Conquer sudah dijelaskan di dalam kuliah. Persoalan tersebut dirumuskan untuk titik pada bidang datar (2D). Pada Tucil 2 kali ini Anda diminta mengembangkan algoritma mencari sepasang titik terdekat pada bidang 3D. Misalkan terdapat n buah titik pada ruang 3D. Setiap titik P di dalam ruang dinyatakan dengan koordinat P = (x, y, z). Carilah sepasang titik yang mempunyai jarak terdekat satu sama lain. Jarak dua buah titk $P_1 = (x_1, y_1, z_1)$ dan $P_2 = (x_2, y_2, z_2)$ dihitung dengan rumus Euclidean berikut:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

Buatlah program dalam dalam Bahasa C/C++/Java/Python/Golang/Ruby/Perl (pilih salah satu) untuk mencari sepasang titik yang jaraknya terdekat datu sama lain dengan menerapkan algoritma divide and conquer untuk penyelesaiannya, dan perbandingannya dengan Algoritma Brute Force.

BAB II
ALGORITMA

Algoritma yang digunakan merupakan aplikasi dari algoritma *divide and conquer*. *Divide and conquer* terdiri atas 2 konsep utama, *divide*, dimana persoalan akan dibagi menjadi beberapa upa-persoalan yang memiliki kemiripan dengan persoalan semula namun dengan ukuran yang lebih kecil, dan *conquer*, dimana tiap upa-persoalan yang dibentuk akan diselesaikan secara rekursif apabila ukuran masih besar dan secara langsung apabila ukuran sudah cukup kecil.



Skema Umum Algoritma *Divide and Conquer*

```
procedure DIVIDEandCONQUER(input P : problem, n : integer)
{ Menyelesaikan persoalan P dengan algoritma divide and conquer
  Masukan: masukan persoalan P berukuran n
  Luaran: solusi dari persoalan semula   }
Deklarasi
   r : integer

Algoritma
  if $n \le n_0$ then  {ukuran persoalan P sudah cukup kecil }
      SOLVE  persoalan P yang berukuran n ini
  else
      DIVIDE  menjadi r upa-persoalan, $P_1, P_2, \ldots, P_r$ , yang masing-masing berukuran $n_1, n_2, \ldots, n_r$
      for  masing-masing $P_1, P_2, \ldots, P_r$ , do
          DIVIDEandCONQUER($P_i$, $n_i$)
      endfor
      COMBINE  solusi dari $P_1, P_2, \ldots, P_r$  menjadi solusi persoalan semula
  endif
```

Kompleksitas algoritma *divide and conquer*: $T(n) = \begin{cases} g(n) & , n \le n_0 \\ T(n_1) + T(n_2) \ldots + T(n_r) + f(n) & , n > n_0 \end{cases}$

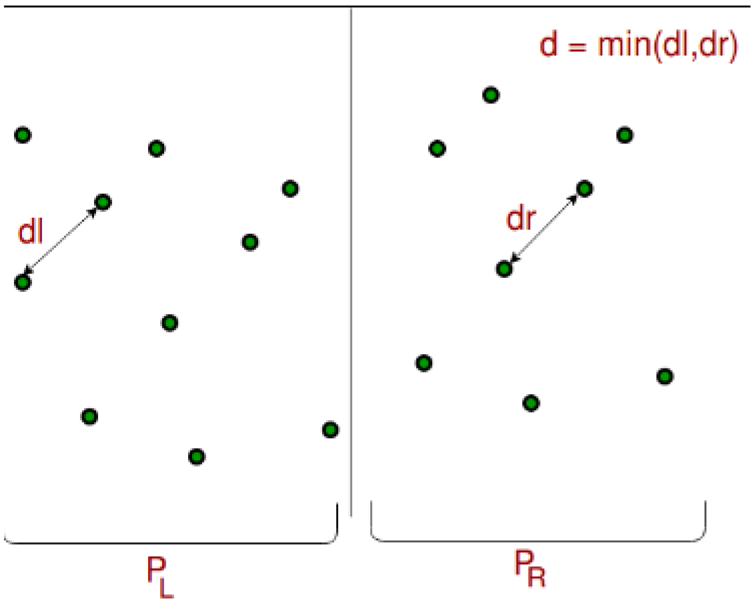
Dikutip dari : https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Divide-and-Conquer-(2021)-Bagian1.pdf

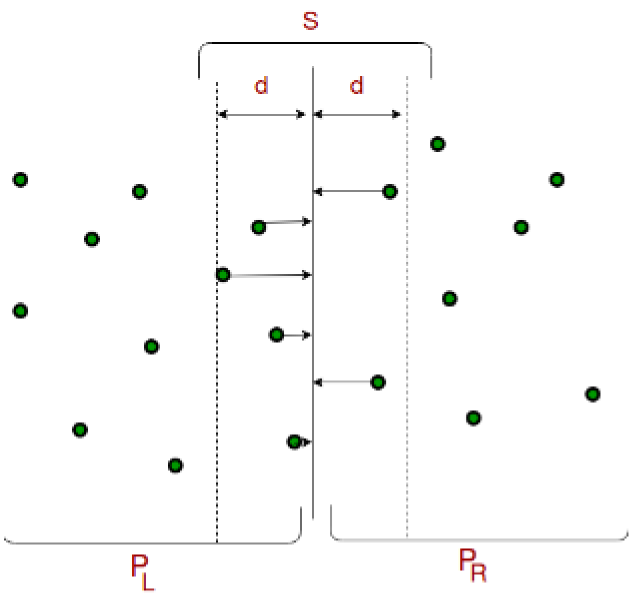
Persoalan mencari pasangan titik terdekat dari sekumpulan titik menggunakan 2 algoritma dan 1 rumus dasar. Rumus dasar yang digunakan adalah rumus untuk mencari jarak antara dua titik, yaitu *euclidean distance formula* yang dirumuskan sebagai berikut.

$$d = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

dengan n adalah jumlah dimensi, x dan y adalah titik yang diamati.

Algoritma *divide and conquer* untuk persoalan mencari pasangan titik terdekat dari sekumpulan titik adalah titik-titik akan dibagi menjadi 2 bagian sama besar hingga upa-persoalan yang muncul menjadi setengah besarnya. Setelah dibagi menjadi 3, dengan 2 bagian sama besar, program akan mencari jarak terdekat antara dua titik dari kedua bagian tersebut dan 1 bagian berada di tengah/median kumpulan titik awal untuk memperhitungkan jarak terdekat yang muncul dari interaksi sisi kanan dan sisi kiri. Berikut ilustrasi dari algoritma:





Dikutip dari : https://www.geeksforgeeks.org/closest-pair-of-points-using-divide-and-conquer-algorithm/

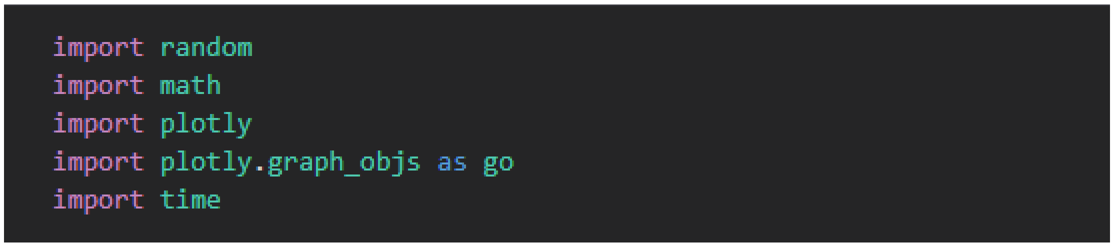
Algoritma sorting dianggap menggunakan *quicksort*, dengan pertimbangkan fungsi yang digunakan merupakan hasil *import* library python. Algoritma *quicksort* adalah algoritma pengurutan sekumpulan data yang terkenal dan tercepat. *Quicksort* merupakan salah satu aplikasi algorita *divide and conquer* yang paling efisien.

BAB III

SOURCE CODE

Bahasa yang digunakan untuk persoalan ini adalah Python. Berikut adalah *source code* hasil implementasi persoalan dalam file main.ipynb.

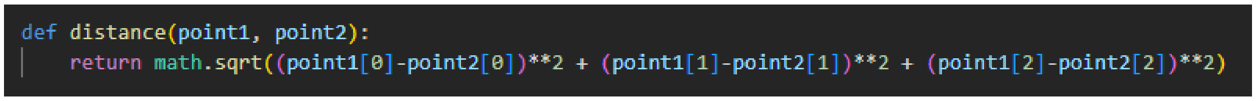
1. **Library**



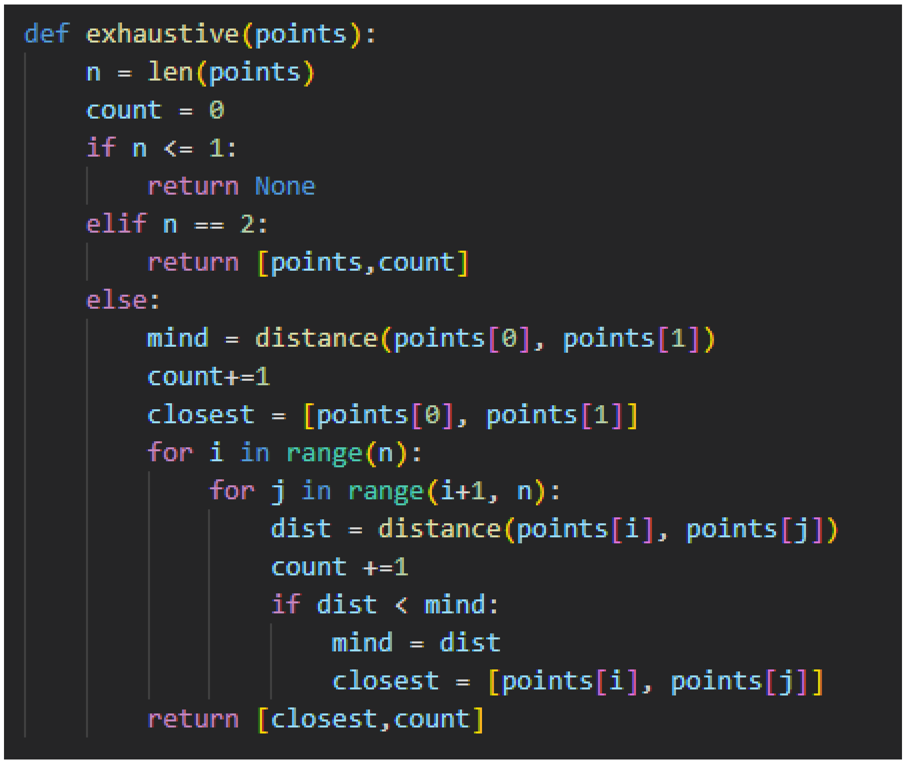
```python
import random
import math
import plotly
import plotly.graph_objs as go
import time
```

Library yang digunakan adalah *random* untuk memunculkan titik-titik *random* yang akan digunakan nantinya, *math* untuk menghitung jarak *euclidian* antara satu titik dengan titik lainnya, *plotly* untuk memvisualisasikan titik-titik dalam suatu *graph* tiga dimesional yang interaktif, dan *time* untuk mengukur waktu eksekusi program.

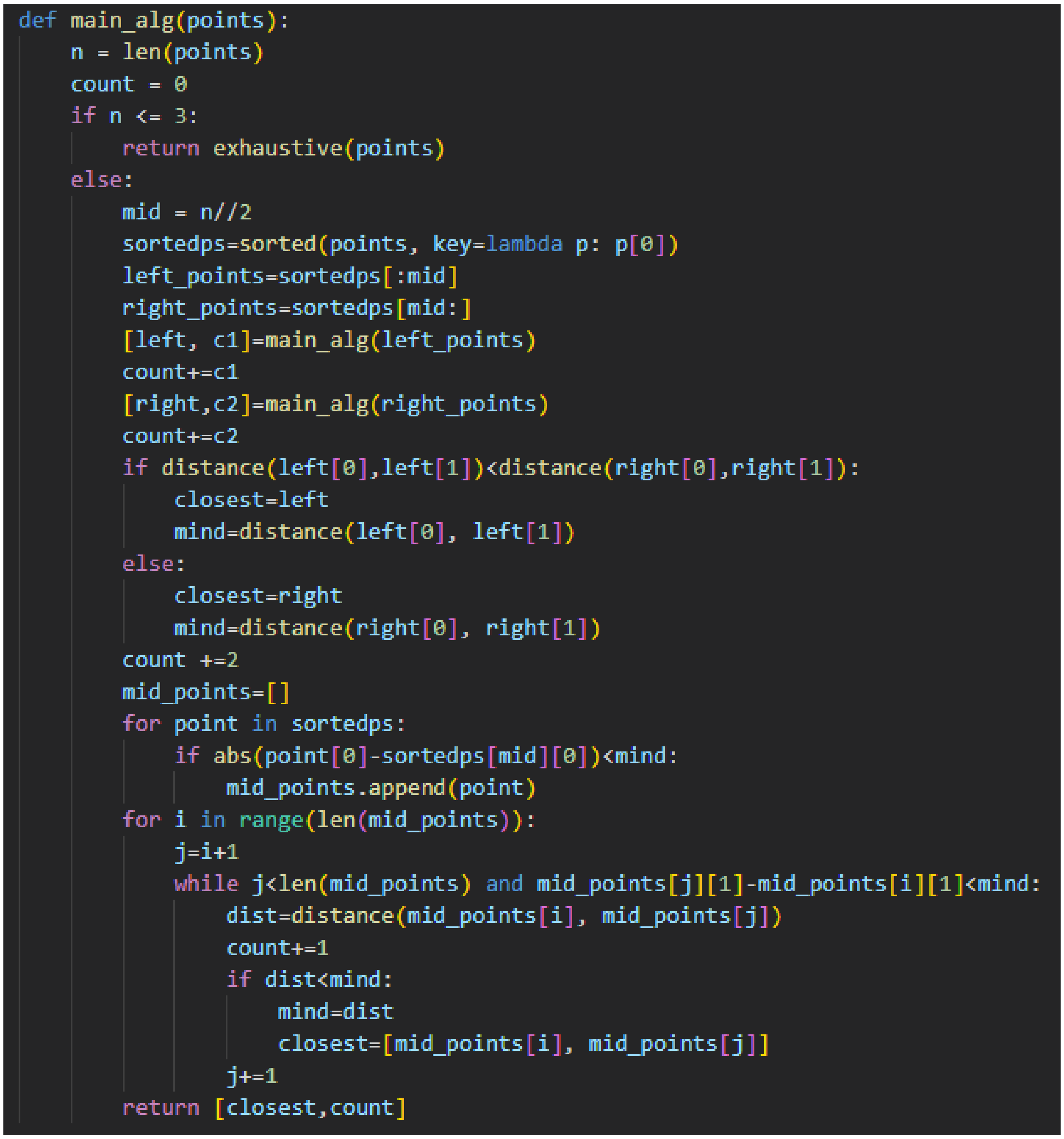
2. **Fungsi**



```python
def distance(point1, point2):
    return math.sqrt((point1[0]-point2[0])**2 + (point1[1]-point2[1])**2 + (point1[2]-point2[2])**2)
```

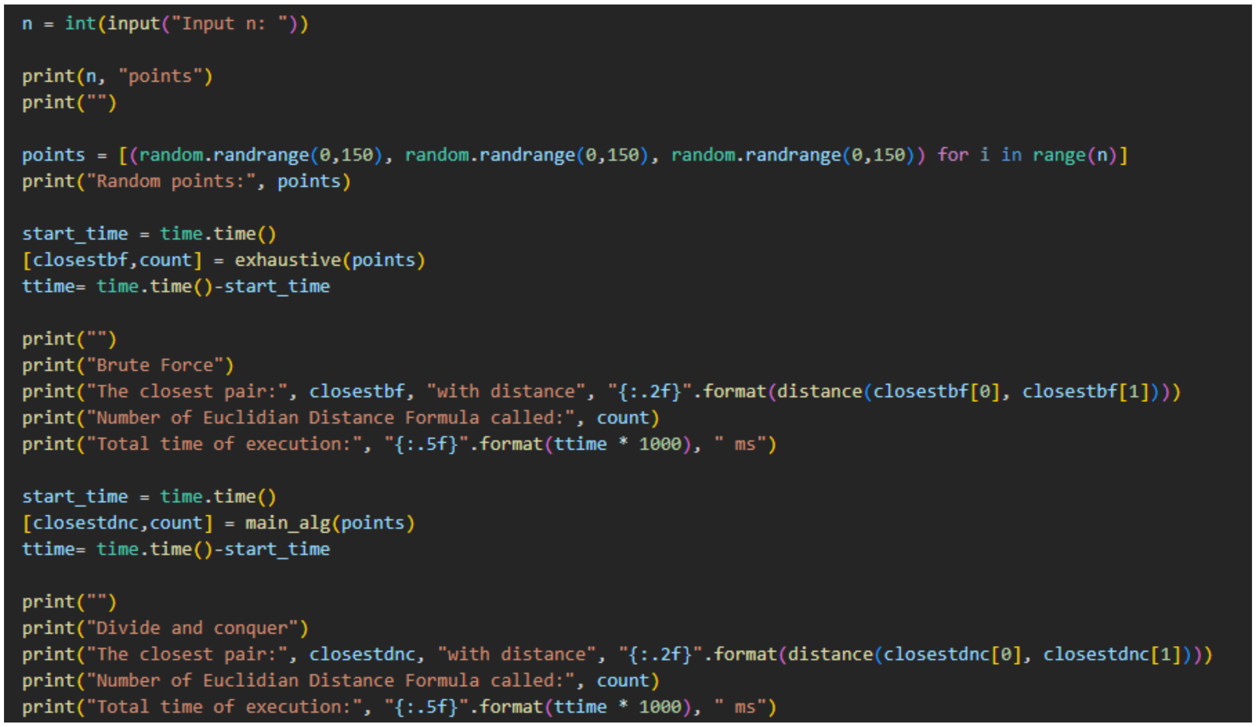


```python
def exhaustive(points):
    n = len(points)
    count = 0
    if n <= 1:
        return None
    elif n == 2:
        return [points,count]
    else:
        mind = distance(points[0], points[1])
        count+=1
        closest = [points[0], points[1]]
        for i in range(n):
            for j in range(i+1, n):
                dist = distance(points[i], points[j])
                count +=1
                if dist < mind:
                    mind = dist
                    closest = [points[i], points[j]]
        return [closest,count]
```



```
def main_alg(points):
    n = len(points)
    count = 0
    if n <= 3:
        return exhaustive(points)
    else:
        mid = n//2
        sortedps=sorted(points, key=lambda p: p[0])
        left_points=sortedps[:mid]
        right_points=sortedps[mid:]
        [left, c1]=main_alg(left_points)
        count+=c1
        [right,c2]=main_alg(right_points)
        count+=c2
        if distance(left[0],left[1])<distance(right[0],right[1]):
            closest=left
            mind=distance(left[0], left[1])
        else:
            closest=right
            mind=distance(right[0], right[1])
        count +=2
        mid_points=[]
        for point in sortedps:
            if abs(point[0]-sortedps[mid][0])<mind:
                mid_points.append(point)
        for i in range(len(mid_points)):
            j=i+1
            while j<len(mid_points) and mid_points[j][1]-mid_points[i][1]<mind:
                dist=distance(mid_points[i], mid_points[j])
                count+=1
                if dist<mind:
                    mind=dist
                    closest=[mid_points[i], mid_points[j]]
                j+=1
        return [closest,count]
```

Terdapat 3 fungsi utama dalam program ini, yaitu *distance*, *exhaustive*, dan *main_alg*. Fungsi *distance* berfungsi untuk mengembalikan jarak antara dua titik *point1* dan *point2*. Fungsi *exhaustive* akan mencari jarak terdekat antara 2 titik dalam sekupulan set titik secara *brute force*. Fungsi *main_alg* akan mencari jarak terdekat antara 2 titik dalam sekumpulan set titik dengan menggunakan konsep *divide and conquer*.



```
n = int(input("Input n: "))

print(n, "points")
print("")

points = [(random.randrange(0,150), random.randrange(0,150), random.randrange(0,150)) for i in range(n)]
print("Random points:", points)

start_time = time.time()
[closestbf,count] = exhaustive(points)
ttime= time.time()-start_time

print("")
print("Brute Force")
print("The closest pair:", closestbf, "with distance", "{:.2f}".format(distance(closestbf[0], closestbf[1])))
print("Number of Euclidian Distance Formula called:", count)
print("Total time of execution:", "{:.5f}".format(ttime * 1000), " ms")

start_time = time.time()
[closestdnc,count] = main_alg(points)
ttime= time.time()-start_time

print("")
print("Divide and conquer")
print("The closest pair:", closestdnc, "with distance", "{:.2f}".format(distance(closestdnc[0], closestdnc[1])))
print("Number of Euclidian Distance Formula called:", count)
print("Total time of execution:", "{:.5f}".format(ttime * 1000), " ms")
```

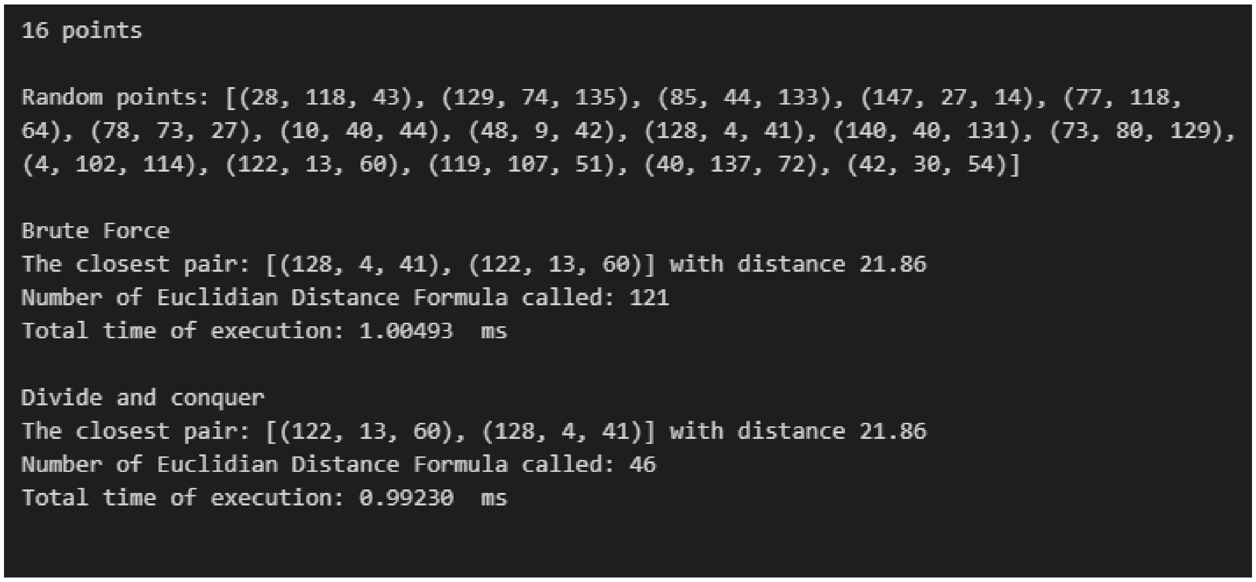
Semua fungsi akan disusun pada bagian utama, dimana akan memunculkan hasil berdasarkan 2 jenis algoritma, *exhaustive* dan *divide and conquer*.

Pemunculan titik-titik secara acak dilakukan menggunakan library *random*. Perlu diketahui bahwa angka yang dimunculkan hanya akan berada diinterval $0 < x < 150$. Jika ingin mengubah intervalnya, dapat dilakukan dengan mengubah parameter *random.randrange(a,b)* dengan a adalah batas bawah dan b adalah batas atas intervalnya.

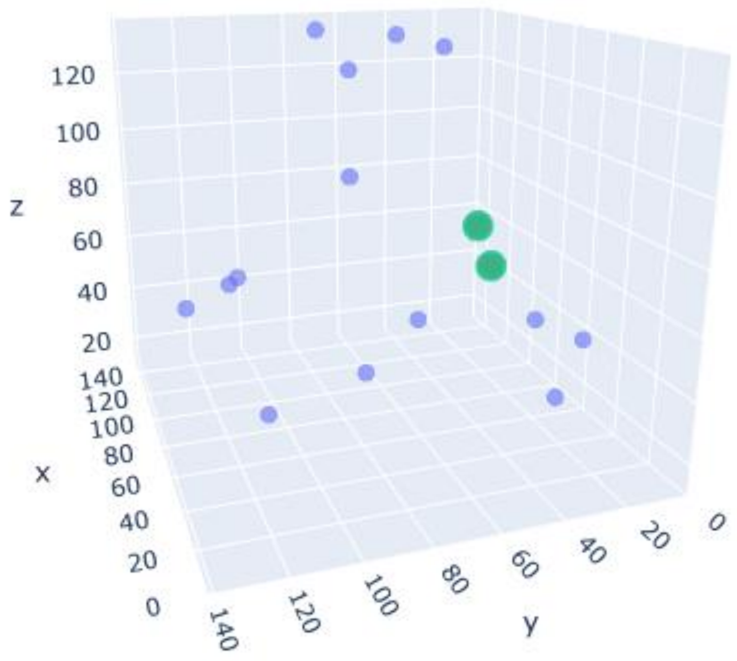
BAB IV

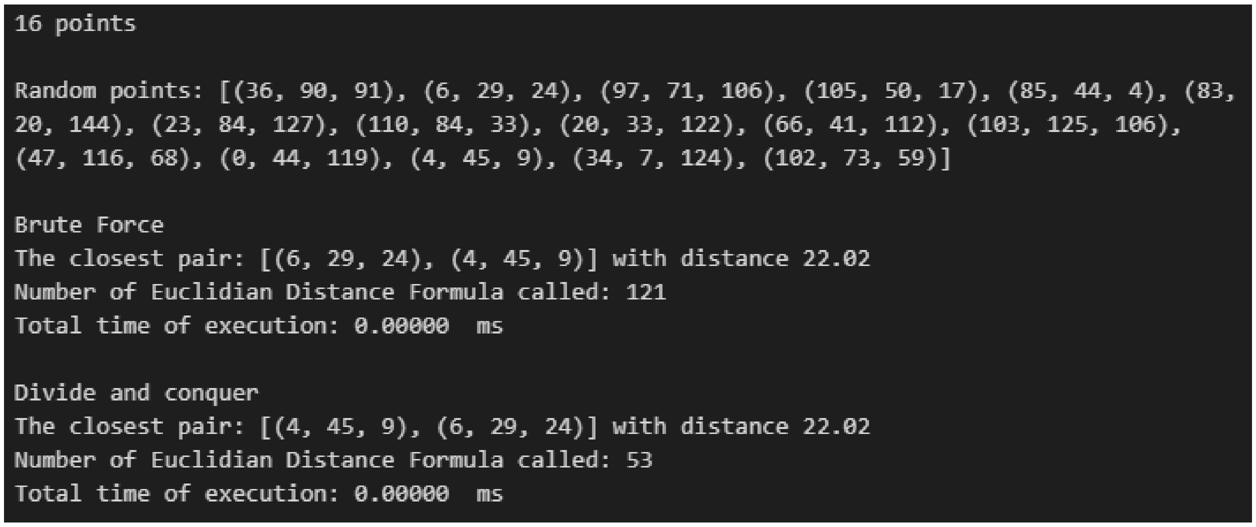
TEST CASE

1. n = 16

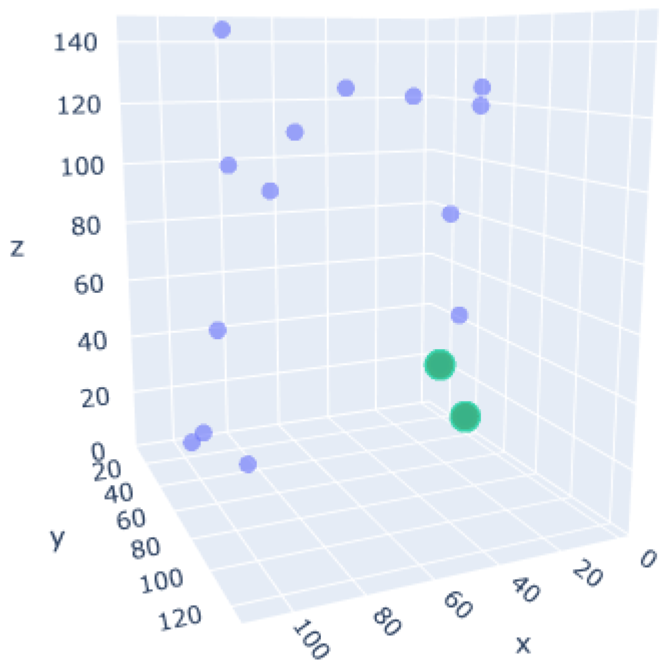


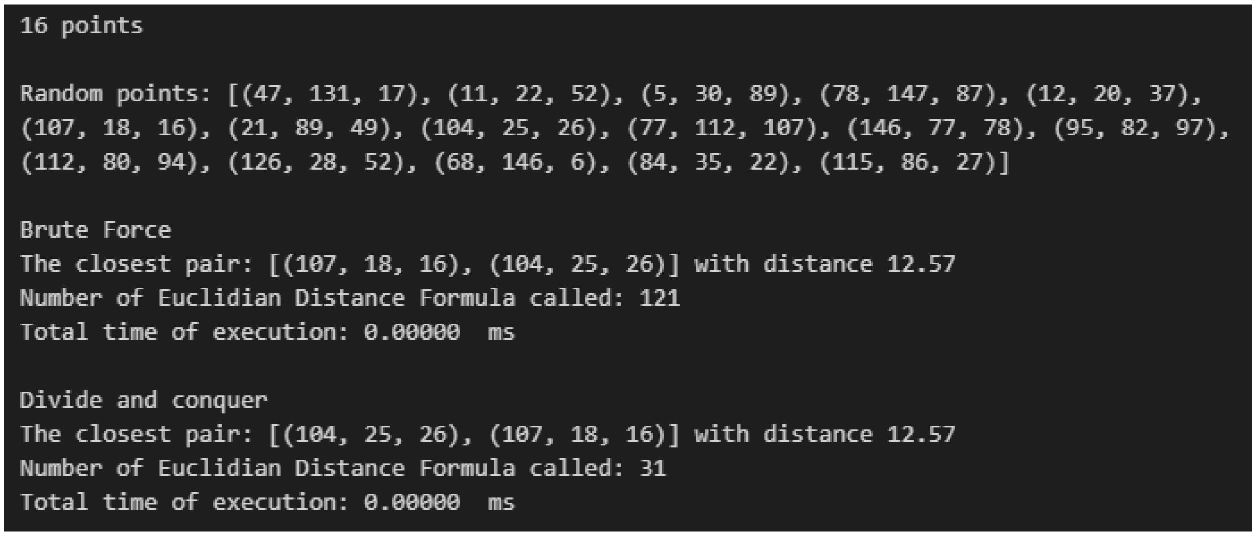
```
16 points

Random points: [(28, 118, 43), (129, 74, 135), (85, 44, 133), (147, 27, 14), (77, 118,
64), (78, 73, 27), (10, 40, 44), (48, 9, 42), (128, 4, 41), (140, 40, 131), (73, 80, 129),
(4, 102, 114), (122, 13, 60), (119, 107, 51), (40, 137, 72), (42, 30, 54)]

Brute Force
The closest pair: [(128, 4, 41), (122, 13, 60)] with distance 21.86
Number of Euclidian Distance Formula called: 121
Total time of execution: 1.00493  ms

Divide and conquer
The closest pair: [(122, 13, 60), (128, 4, 41)] with distance 21.86
Number of Euclidian Distance Formula called: 46
Total time of execution: 0.99230  ms
```



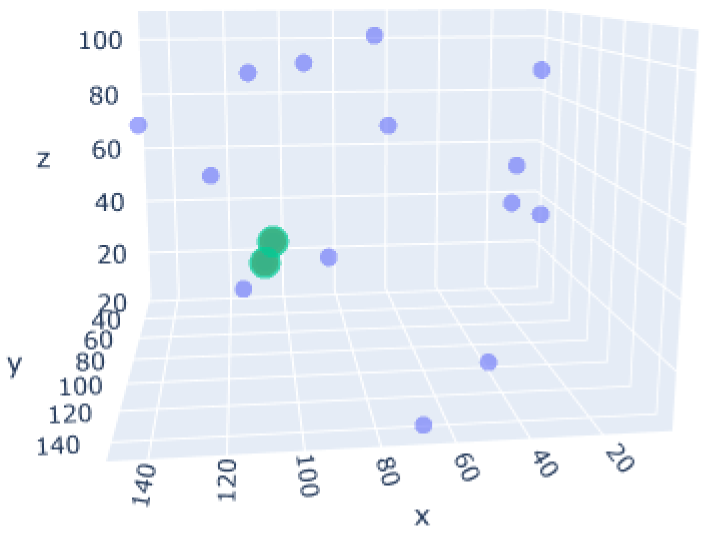


```
16 points

Random points: [(36, 90, 91), (6, 29, 24), (97, 71, 106), (105, 50, 17), (85, 44, 4), (83,
20, 144), (23, 84, 127), (110, 84, 33), (20, 33, 122), (66, 41, 112), (103, 125, 106),
(47, 116, 68), (0, 44, 119), (4, 45, 9), (34, 7, 124), (102, 73, 59)]

Brute Force
The closest pair: [(6, 29, 24), (4, 45, 9)] with distance 22.02
Number of Euclidian Distance Formula called: 121
Total time of execution: 0.00000  ms

Divide and conquer
The closest pair: [(4, 45, 9), (6, 29, 24)] with distance 22.02
Number of Euclidian Distance Formula called: 53
Total time of execution: 0.00000  ms
```



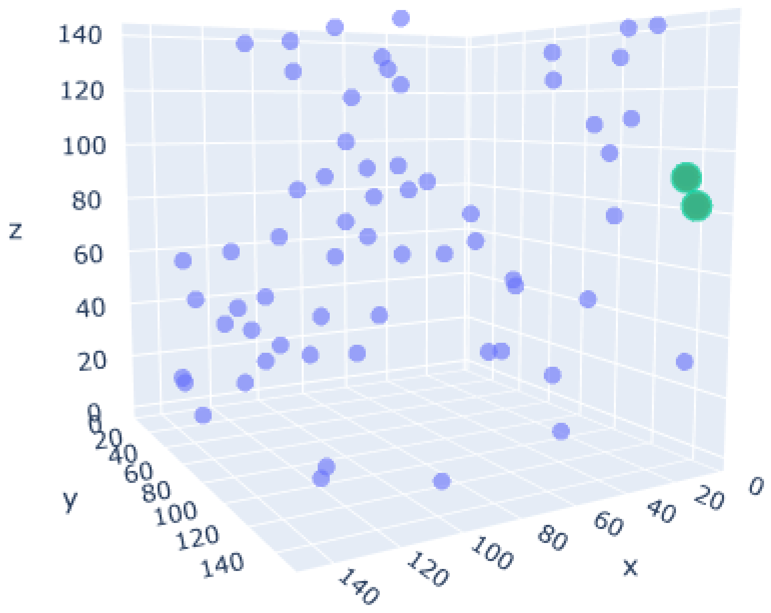


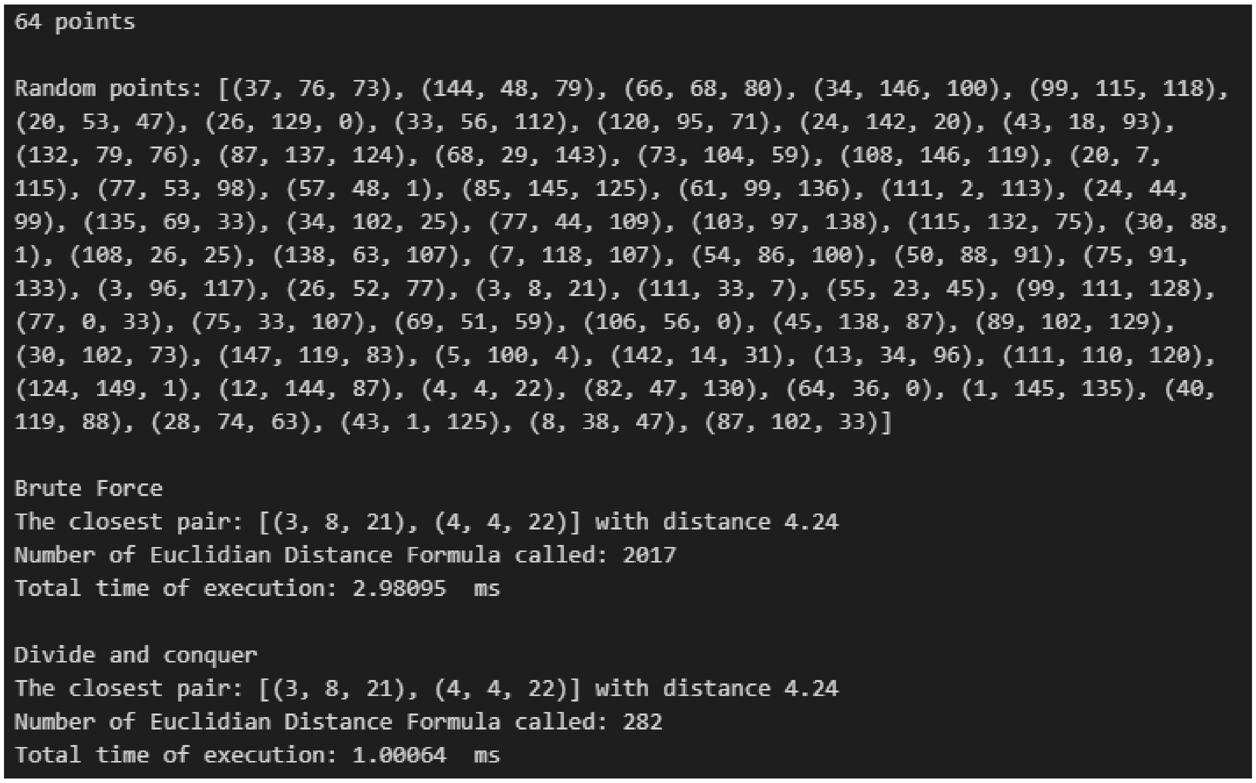
```
16 points

Random points: [(47, 131, 17), (11, 22, 52), (5, 30, 89), (78, 147, 87), (12, 20, 37),
(107, 18, 16), (21, 89, 49), (104, 25, 26), (77, 112, 107), (146, 77, 78), (95, 82, 97),
(112, 80, 94), (126, 28, 52), (68, 146, 6), (84, 35, 22), (115, 86, 27)]

Brute Force
The closest pair: [(107, 18, 16), (104, 25, 26)] with distance 12.57
Number of Euclidian Distance Formula called: 121
Total time of execution: 0.00000  ms

Divide and conquer
The closest pair: [(104, 25, 26), (107, 18, 16)] with distance 12.57
Number of Euclidian Distance Formula called: 31
Total time of execution: 0.00000  ms
```



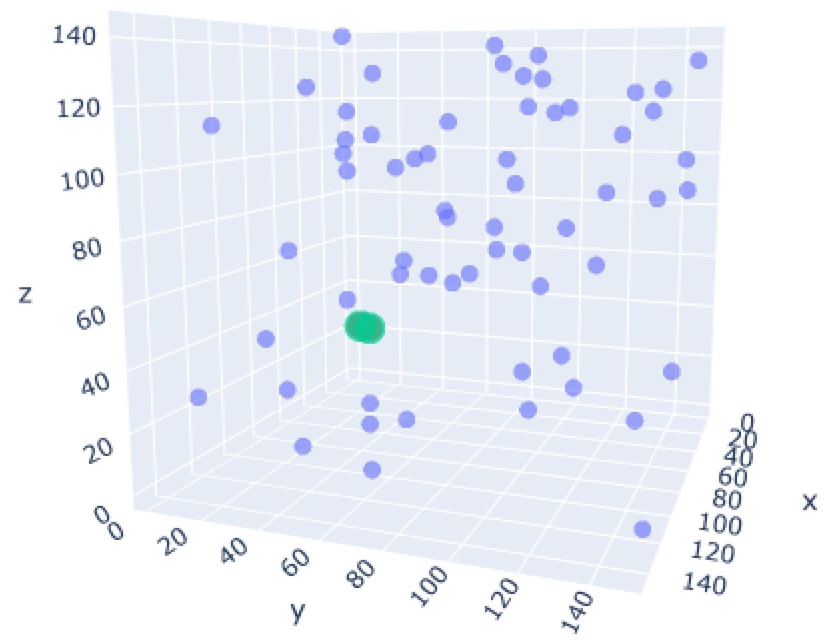
2. n = 64

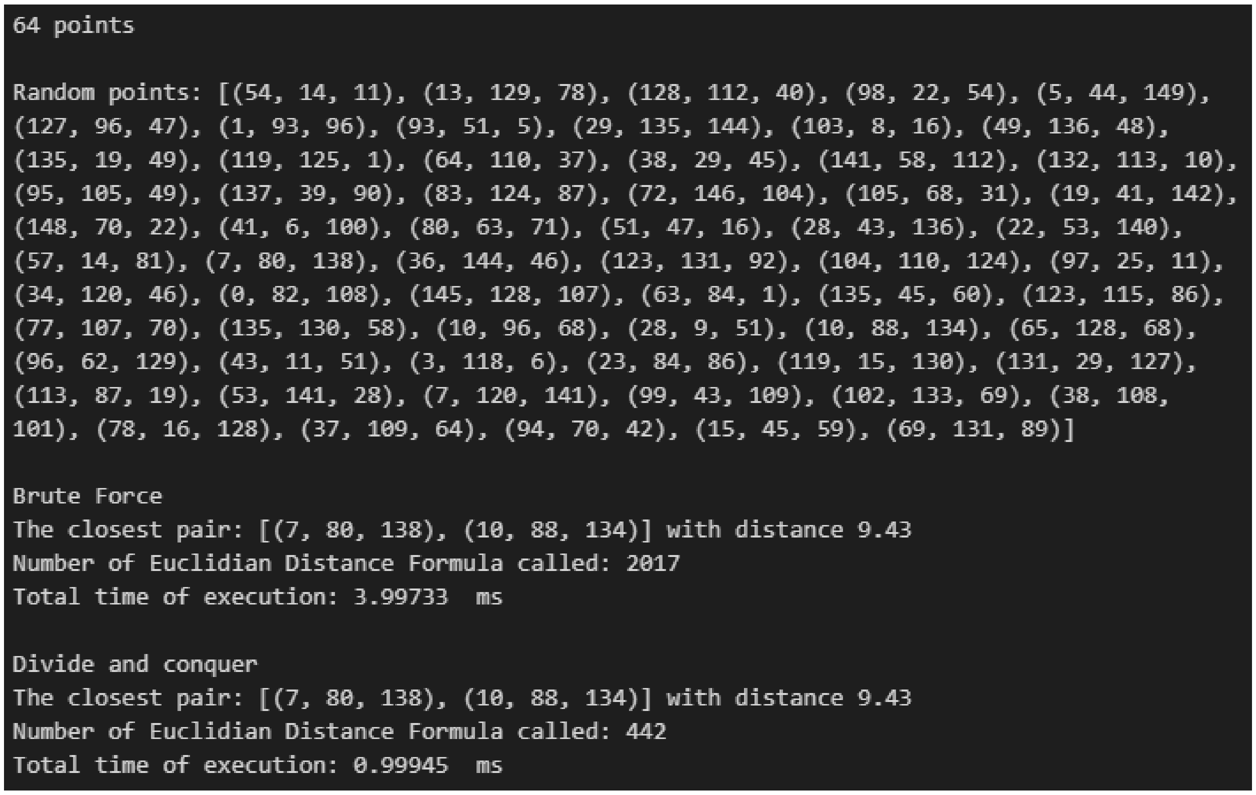
```
64 points

Random points: [(111, 36, 21), (143, 40, 18), (70, 51, 132), (146, 68, 14), (24, 70, 49),
(65, 56, 122), (20, 38, 60), (100, 101, 86), (120, 32, 41), (40, 149, 137), (38, 91, 51),
(103, 58, 27), (15, 103, 46), (115, 145, 137), (140, 81, 69), (32, 103, 123), (100, 74,
140), (10, 77, 12), (77, 106, 67), (137, 18, 14), (90, 126, 0), (77, 25, 56), (85, 68,
69), (64, 60, 84), (22, 142, 139), (122, 88, 134), (10, 146, 32), (49, 132, 11), (97, 80,
117), (94, 28, 83), (120, 107, 4), (126, 54, 19), (42, 13, 90), (125, 84, 73), (109, 17,
29), (48, 97, 30), (113, 55, 31), (100, 61, 41), (119, 133, 97), (130, 70, 134), (119,
140, 57), (90, 92, 126), (4, 138, 91), (138, 146, 96), (119, 15, 32), (127, 118, 5), (99,
79, 76), (15, 116, 77), (28, 13, 83), (10, 107, 98), (3, 142, 82), (10, 19, 69), (57, 4,
11), (100, 119, 70), (67, 117, 36), (18, 108, 108), (12, 76, 134), (132, 6, 56), (5, 112,
110), (142, 114, 60), (120, 122, 104), (128, 8, 41), (7, 108, 131), (119, 83, 125)]

Brute Force
The closest pair: [(4, 138, 91), (3, 142, 82)] with distance 9.90
Number of Euclidian Distance Formula called: 2017
Total time of execution: 4.00615  ms

Divide and conquer
The closest pair: [(3, 142, 82), (4, 138, 91)] with distance 9.90
Number of Euclidian Distance Formula called: 375
Total time of execution: 0.98944  ms
```



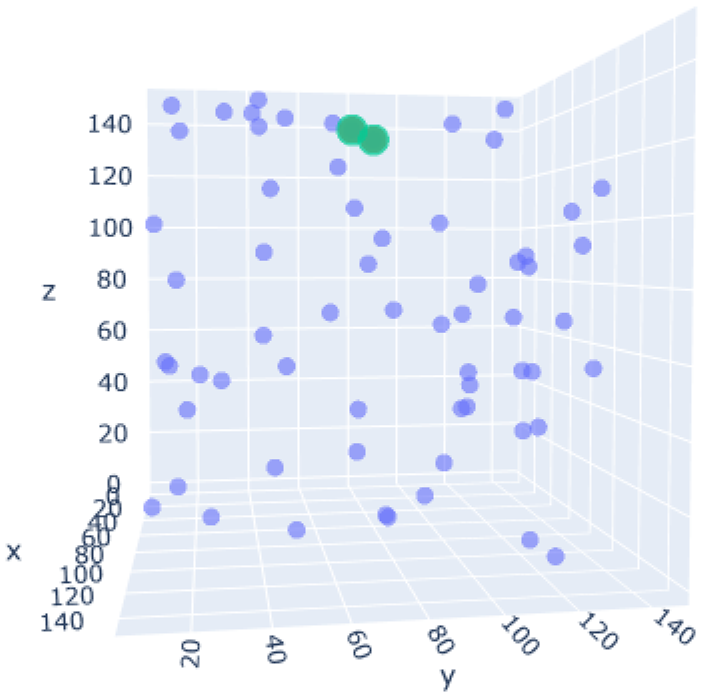


```
64 points

Random points: [(37, 76, 73), (144, 48, 79), (66, 68, 80), (34, 146, 100), (99, 115, 118),
(20, 53, 47), (26, 129, 0), (33, 56, 112), (120, 95, 71), (24, 142, 20), (43, 18, 93),
(132, 79, 76), (87, 137, 124), (68, 29, 143), (73, 104, 59), (108, 146, 119), (20, 7,
115), (77, 53, 98), (57, 48, 1), (85, 145, 125), (61, 99, 136), (111, 2, 113), (24, 44,
99), (135, 69, 33), (34, 102, 25), (77, 44, 109), (103, 97, 138), (115, 132, 75), (30, 88,
1), (108, 26, 25), (138, 63, 107), (7, 118, 107), (54, 86, 100), (50, 88, 91), (75, 91,
133), (3, 96, 117), (26, 52, 77), (3, 8, 21), (111, 33, 7), (55, 23, 45), (99, 111, 128),
(77, 0, 33), (75, 33, 107), (69, 51, 59), (106, 56, 0), (45, 138, 87), (89, 102, 129),
(30, 102, 73), (147, 119, 83), (5, 100, 4), (142, 14, 31), (13, 34, 96), (111, 110, 120),
(124, 149, 1), (12, 144, 87), (4, 4, 22), (82, 47, 130), (64, 36, 0), (1, 145, 135), (40,
119, 88), (28, 74, 63), (43, 1, 125), (8, 38, 47), (87, 102, 33)]

Brute Force
The closest pair: [(3, 8, 21), (4, 4, 22)] with distance 4.24
Number of Euclidian Distance Formula called: 2017
Total time of execution: 2.98095  ms

Divide and conquer
The closest pair: [(3, 8, 21), (4, 4, 22)] with distance 4.24
Number of Euclidian Distance Formula called: 282
Total time of execution: 1.00064  ms
```

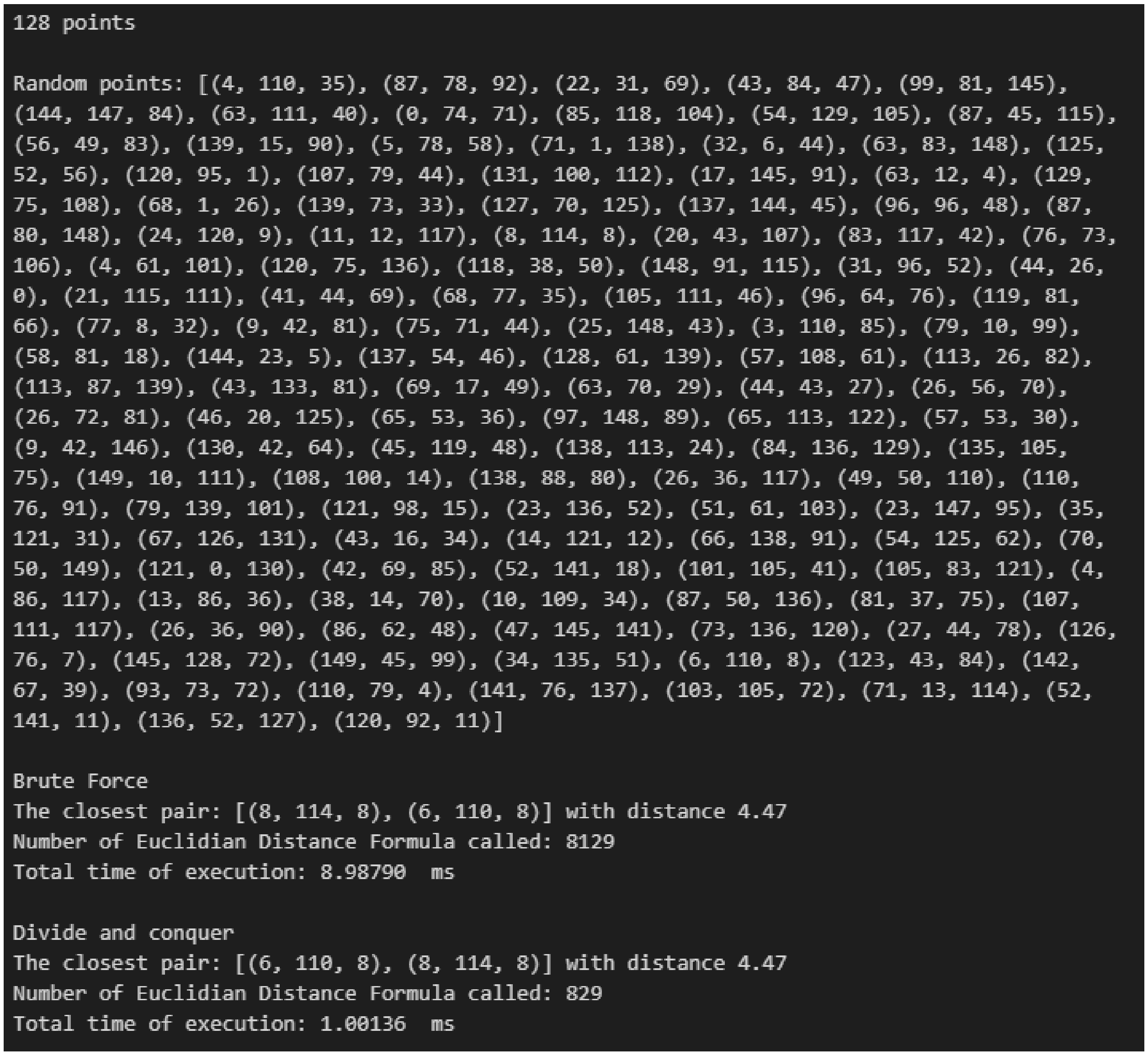




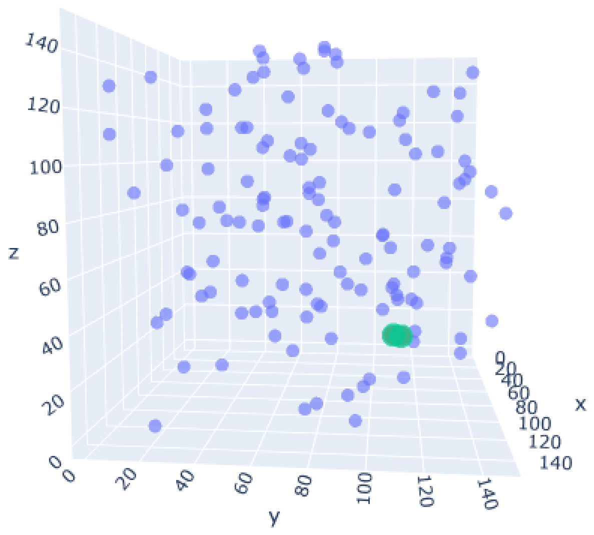
```
64 points

Random points: [(54, 14, 11), (13, 129, 78), (128, 112, 40), (98, 22, 54), (5, 44, 149),
(127, 96, 47), (1, 93, 96), (93, 51, 5), (29, 135, 144), (103, 8, 16), (49, 136, 48),
(135, 19, 49), (119, 125, 1), (64, 110, 37), (38, 29, 45), (141, 58, 112), (132, 113, 10),
(95, 105, 49), (137, 39, 90), (83, 124, 87), (72, 146, 104), (105, 68, 31), (19, 41, 142),
(148, 70, 22), (41, 6, 100), (80, 63, 71), (51, 47, 16), (28, 43, 136), (22, 53, 140),
(57, 14, 81), (7, 80, 138), (36, 144, 46), (123, 131, 92), (104, 110, 124), (97, 25, 11),
(34, 120, 46), (0, 82, 108), (145, 128, 107), (63, 84, 1), (135, 45, 60), (123, 115, 86),
(77, 107, 70), (135, 130, 58), (10, 96, 68), (28, 9, 51), (10, 88, 134), (65, 128, 68),
(96, 62, 129), (43, 11, 51), (3, 118, 6), (23, 84, 86), (119, 15, 130), (131, 29, 127),
(113, 87, 19), (53, 141, 28), (7, 120, 141), (99, 43, 109), (102, 133, 69), (38, 108,
101), (78, 16, 128), (37, 109, 64), (94, 70, 42), (15, 45, 59), (69, 131, 89)]

Brute Force
The closest pair: [(7, 80, 138), (10, 88, 134)] with distance 9.43
Number of Euclidian Distance Formula called: 2017
Total time of execution: 3.99733  ms

Divide and conquer
The closest pair: [(7, 80, 138), (10, 88, 134)] with distance 9.43
Number of Euclidian Distance Formula called: 442
Total time of execution: 0.99945  ms
```

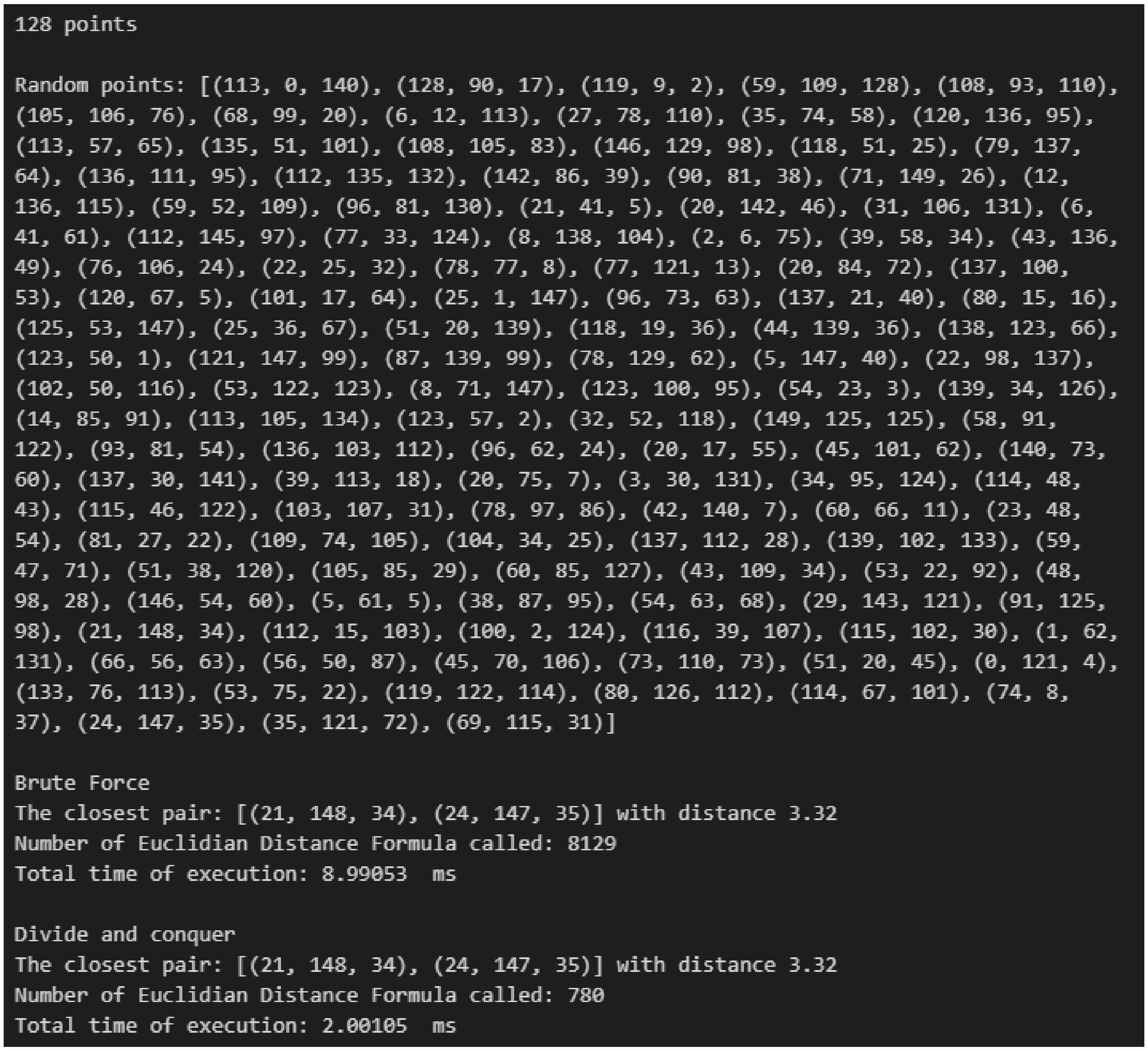


3. n = 128

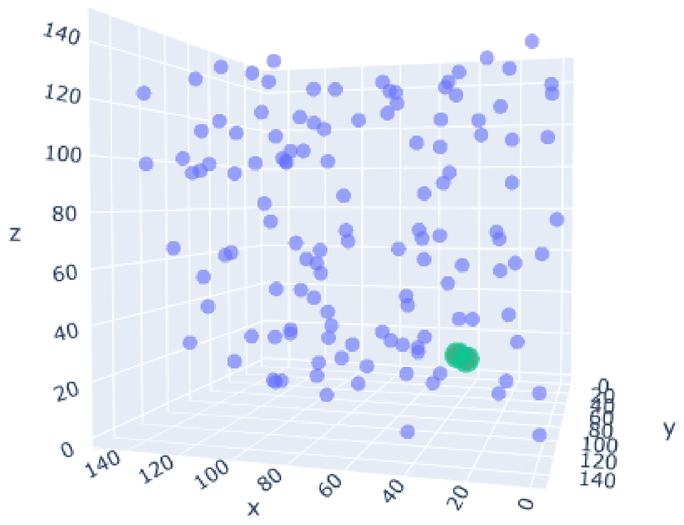


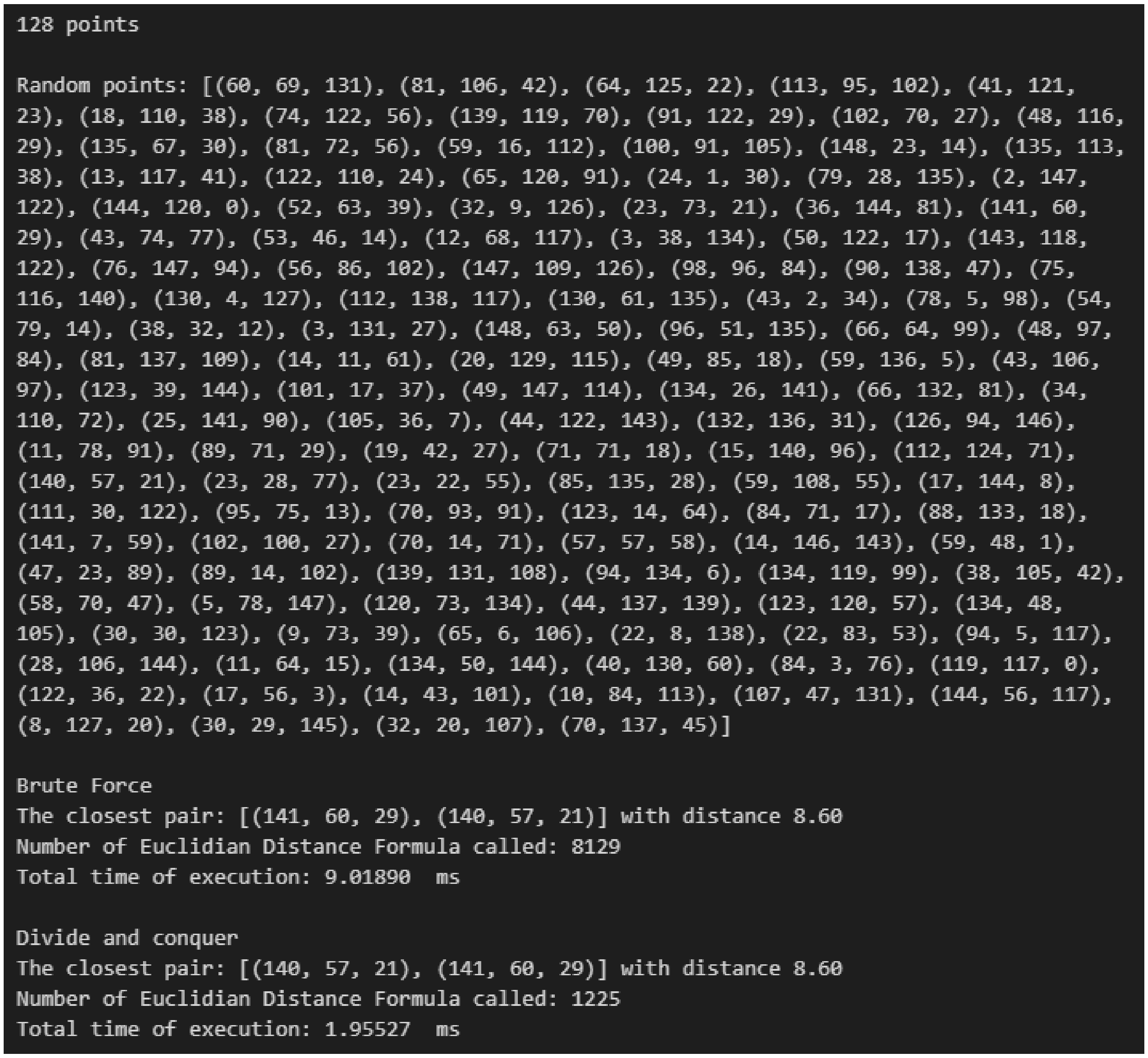
```
128 points

Random points: [(4, 110, 35), (87, 78, 92), (22, 31, 69), (43, 84, 47), (99, 81, 145),
(144, 147, 84), (63, 111, 40), (0, 74, 71), (85, 118, 104), (54, 129, 105), (87, 45, 115),
(56, 49, 83), (139, 15, 90), (5, 78, 58), (71, 1, 138), (32, 6, 44), (63, 83, 148), (125,
52, 56), (120, 95, 1), (107, 79, 44), (131, 100, 112), (17, 145, 91), (63, 12, 4), (129,
75, 108), (68, 1, 26), (139, 73, 33), (127, 70, 125), (137, 144, 45), (96, 96, 48), (87,
80, 148), (24, 120, 9), (11, 12, 117), (8, 114, 8), (20, 43, 107), (83, 117, 42), (76, 73,
106), (4, 61, 101), (120, 75, 136), (118, 38, 50), (148, 91, 115), (31, 96, 52), (44, 26,
0), (21, 115, 111), (41, 44, 69), (68, 77, 35), (105, 111, 46), (96, 64, 76), (119, 81,
66), (77, 8, 32), (9, 42, 81), (75, 71, 44), (25, 148, 43), (3, 110, 85), (79, 10, 99),
(58, 81, 18), (144, 23, 5), (137, 54, 46), (128, 61, 139), (57, 108, 61), (113, 26, 82),
(113, 87, 139), (43, 133, 81), (69, 17, 49), (63, 70, 29), (44, 43, 27), (26, 56, 70),
(26, 72, 81), (46, 20, 125), (65, 53, 36), (97, 148, 89), (65, 113, 122), (57, 53, 30),
(9, 42, 146), (130, 42, 64), (45, 119, 48), (138, 113, 24), (84, 136, 129), (135, 105,
75), (149, 10, 111), (108, 100, 14), (138, 88, 80), (26, 36, 117), (49, 50, 110), (110,
76, 91), (79, 139, 101), (121, 98, 15), (23, 136, 52), (51, 61, 103), (23, 147, 95), (35,
121, 31), (67, 126, 131), (43, 16, 34), (14, 121, 12), (66, 138, 91), (54, 125, 62), (70,
50, 149), (121, 0, 130), (42, 69, 85), (52, 141, 18), (101, 105, 41), (105, 83, 121), (4,
86, 117), (13, 86, 36), (38, 14, 70), (10, 109, 34), (87, 50, 136), (81, 37, 75), (107,
111, 117), (26, 36, 90), (86, 62, 48), (47, 145, 141), (73, 136, 120), (27, 44, 78), (126,
76, 7), (145, 128, 72), (149, 45, 99), (34, 135, 51), (6, 110, 8), (123, 43, 84), (142,
67, 39), (93, 73, 72), (110, 79, 4), (141, 76, 137), (103, 105, 72), (71, 13, 114), (52,
141, 11), (136, 52, 127), (120, 92, 11)]

Brute Force
The closest pair: [(8, 114, 8), (6, 110, 8)] with distance 4.47
Number of Euclidian Distance Formula called: 8129
Total time of execution: 8.98790  ms

Divide and conquer
The closest pair: [(6, 110, 8), (8, 114, 8)] with distance 4.47
Number of Euclidian Distance Formula called: 829
Total time of execution: 1.00136  ms
```



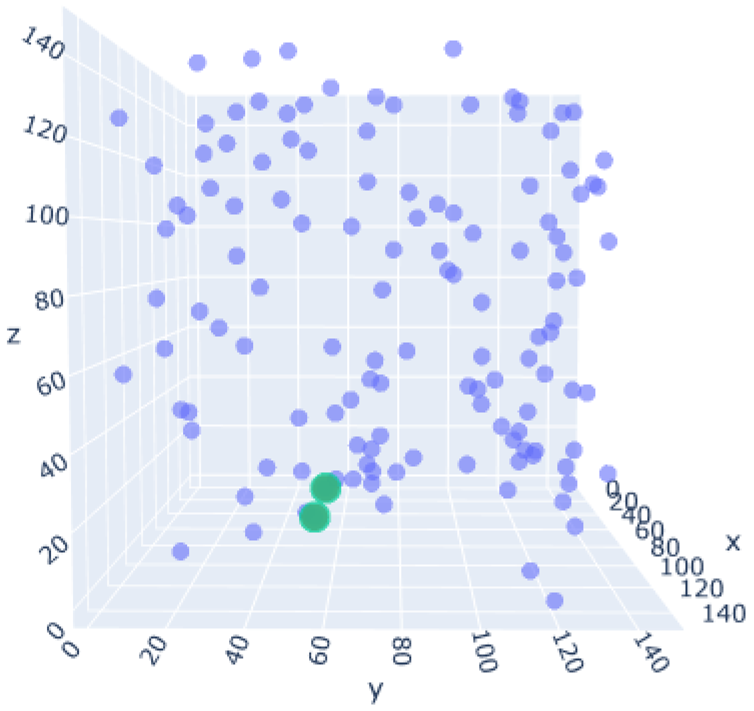


```
128 points

Random points: [(113, 0, 140), (128, 90, 17), (119, 9, 2), (59, 109, 128), (108, 93, 110),
(105, 106, 76), (68, 99, 20), (6, 12, 113), (27, 78, 110), (35, 74, 58), (120, 136, 95),
(113, 57, 65), (135, 51, 101), (108, 105, 83), (146, 129, 98), (118, 51, 25), (79, 137,
64), (136, 111, 95), (112, 135, 132), (142, 86, 39), (90, 81, 38), (71, 149, 26), (12,
136, 115), (59, 52, 109), (96, 81, 130), (21, 41, 5), (20, 142, 46), (31, 106, 131), (6,
41, 61), (112, 145, 97), (77, 33, 124), (8, 138, 104), (2, 6, 75), (39, 58, 34), (43, 136,
49), (76, 106, 24), (22, 25, 32), (78, 77, 8), (77, 121, 13), (20, 84, 72), (137, 100,
53), (120, 67, 5), (101, 17, 64), (25, 1, 147), (96, 73, 63), (137, 21, 40), (80, 15, 16),
(125, 53, 147), (25, 36, 67), (51, 20, 139), (118, 19, 36), (44, 139, 36), (138, 123, 66),
(123, 50, 1), (121, 147, 99), (87, 139, 99), (78, 129, 62), (5, 147, 40), (22, 98, 137),
(102, 50, 116), (53, 122, 123), (8, 71, 147), (123, 100, 95), (54, 23, 3), (139, 34, 126),
(14, 85, 91), (113, 105, 134), (123, 57, 2), (32, 52, 118), (149, 125, 125), (58, 91,
122), (93, 81, 54), (136, 103, 112), (96, 62, 24), (20, 17, 55), (45, 101, 62), (140, 73,
60), (137, 30, 141), (39, 113, 18), (20, 75, 7), (3, 30, 131), (34, 95, 124), (114, 48,
43), (115, 46, 122), (103, 107, 31), (78, 97, 86), (42, 140, 7), (60, 66, 11), (23, 48,
54), (81, 27, 22), (109, 74, 105), (104, 34, 25), (137, 112, 28), (139, 102, 133), (59,
47, 71), (51, 38, 120), (105, 85, 29), (60, 85, 127), (43, 109, 34), (53, 22, 92), (48,
98, 28), (146, 54, 60), (5, 61, 5), (38, 87, 95), (54, 63, 68), (29, 143, 121), (91, 125,
98), (21, 148, 34), (112, 15, 103), (100, 2, 124), (116, 39, 107), (115, 102, 30), (1, 62,
131), (66, 56, 63), (56, 50, 87), (45, 70, 106), (73, 110, 73), (51, 20, 45), (0, 121, 4),
(133, 76, 113), (53, 75, 22), (119, 122, 114), (80, 126, 112), (114, 67, 101), (74, 8,
37), (24, 147, 35), (35, 121, 72), (69, 115, 31)]

Brute Force
The closest pair: [(21, 148, 34), (24, 147, 35)] with distance 3.32
Number of Euclidian Distance Formula called: 8129
Total time of execution: 8.99053  ms

Divide and conquer
The closest pair: [(21, 148, 34), (24, 147, 35)] with distance 3.32
Number of Euclidian Distance Formula called: 780
Total time of execution: 2.00105  ms
```

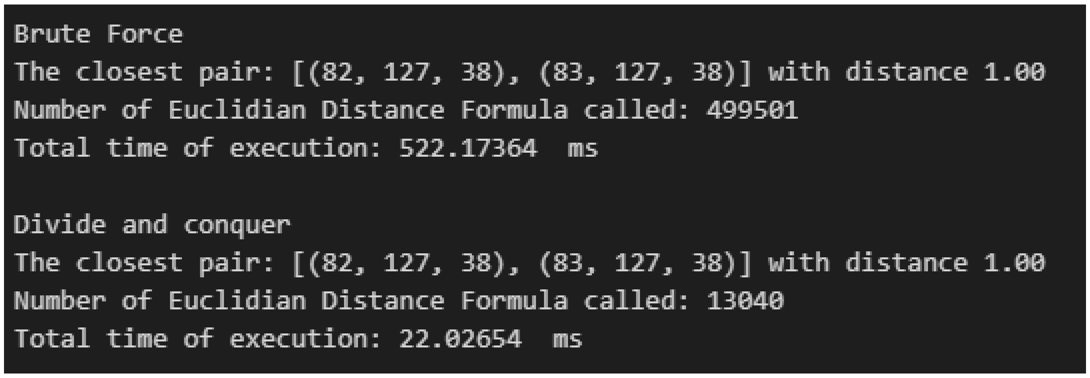




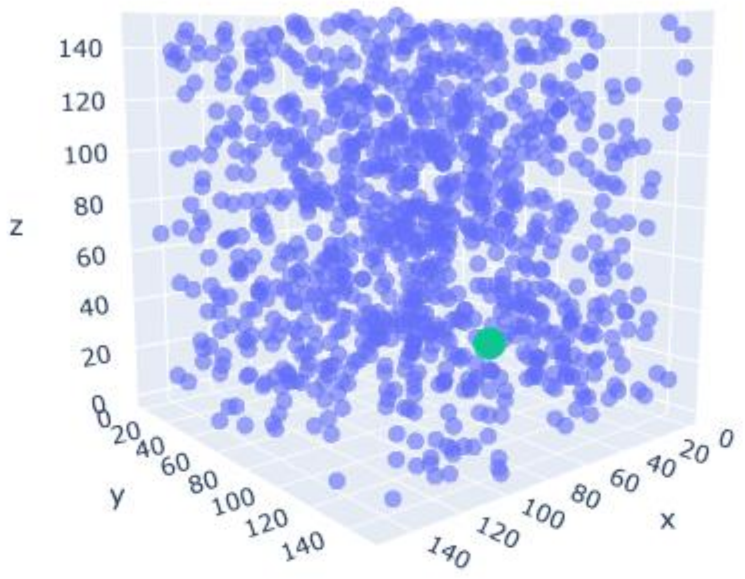
```
128 points

Random points: [(60, 69, 131), (81, 106, 42), (64, 125, 22), (113, 95, 102), (41, 121,
23), (18, 110, 38), (74, 122, 56), (139, 119, 70), (91, 122, 29), (102, 70, 27), (48, 116,
29), (135, 67, 30), (81, 72, 56), (59, 16, 112), (100, 91, 105), (148, 23, 14), (135, 113,
38), (13, 117, 41), (122, 110, 24), (65, 120, 91), (24, 1, 30), (79, 28, 135), (2, 147,
122), (144, 120, 0), (52, 63, 39), (32, 9, 126), (23, 73, 21), (36, 144, 81), (141, 60,
29), (43, 74, 77), (53, 46, 14), (12, 68, 117), (3, 38, 134), (50, 122, 17), (143, 118,
122), (76, 147, 94), (56, 86, 102), (147, 109, 126), (98, 96, 84), (90, 138, 47), (75,
116, 140), (130, 4, 127), (112, 138, 117), (130, 61, 135), (43, 2, 34), (78, 5, 98), (54,
79, 14), (38, 32, 12), (3, 131, 27), (148, 63, 50), (96, 51, 135), (66, 64, 99), (48, 97,
84), (81, 137, 109), (14, 11, 61), (20, 129, 115), (49, 85, 18), (59, 136, 5), (43, 106,
97), (123, 39, 144), (101, 17, 37), (49, 147, 114), (134, 26, 141), (66, 132, 81), (34,
110, 72), (25, 141, 90), (105, 36, 7), (44, 122, 143), (132, 136, 31), (126, 94, 146),
(11, 78, 91), (89, 71, 29), (19, 42, 27), (71, 71, 18), (15, 140, 96), (112, 124, 71),
(140, 57, 21), (23, 28, 77), (23, 22, 55), (85, 135, 28), (59, 108, 55), (17, 144, 8),
(111, 30, 122), (95, 75, 13), (70, 93, 91), (123, 14, 64), (84, 71, 17), (88, 133, 18),
(141, 7, 59), (102, 100, 27), (70, 14, 71), (57, 57, 58), (14, 146, 143), (59, 48, 1),
(47, 23, 89), (89, 14, 102), (139, 131, 108), (94, 134, 6), (134, 119, 99), (38, 105, 42),
(58, 70, 47), (5, 78, 147), (120, 73, 134), (44, 137, 139), (123, 120, 57), (134, 48,
105), (30, 30, 123), (9, 73, 39), (65, 6, 106), (22, 8, 138), (22, 83, 53), (94, 5, 117),
(28, 106, 144), (11, 64, 15), (134, 50, 144), (40, 130, 60), (84, 3, 76), (119, 117, 0),
(122, 36, 22), (17, 56, 3), (14, 43, 101), (10, 84, 113), (107, 47, 131), (144, 56, 117),
(8, 127, 20), (30, 29, 145), (32, 20, 107), (70, 137, 45)]

Brute Force
The closest pair: [(141, 60, 29), (140, 57, 21)] with distance 8.60
Number of Euclidian Distance Formula called: 8129
Total time of execution: 9.01890  ms

Divide and conquer
The closest pair: [(140, 57, 21), (141, 60, 29)] with distance 8.60
Number of Euclidian Distance Formula called: 1225
Total time of execution: 1.95527  ms
```

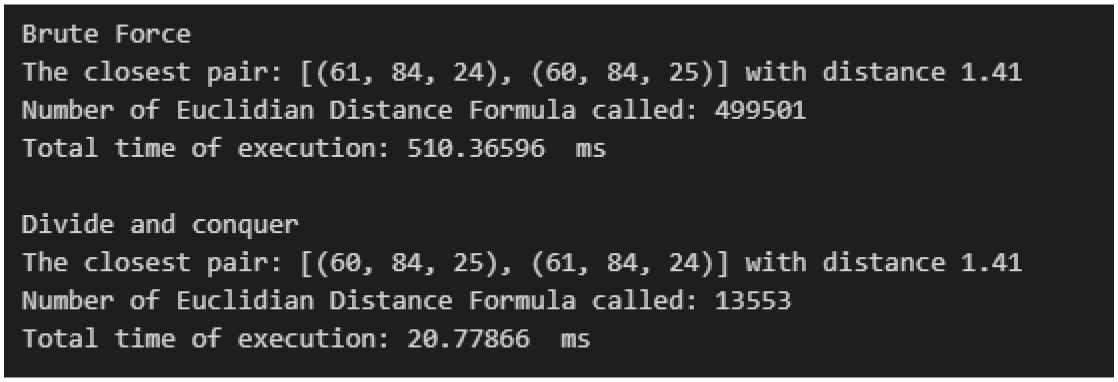


4. n = 1000

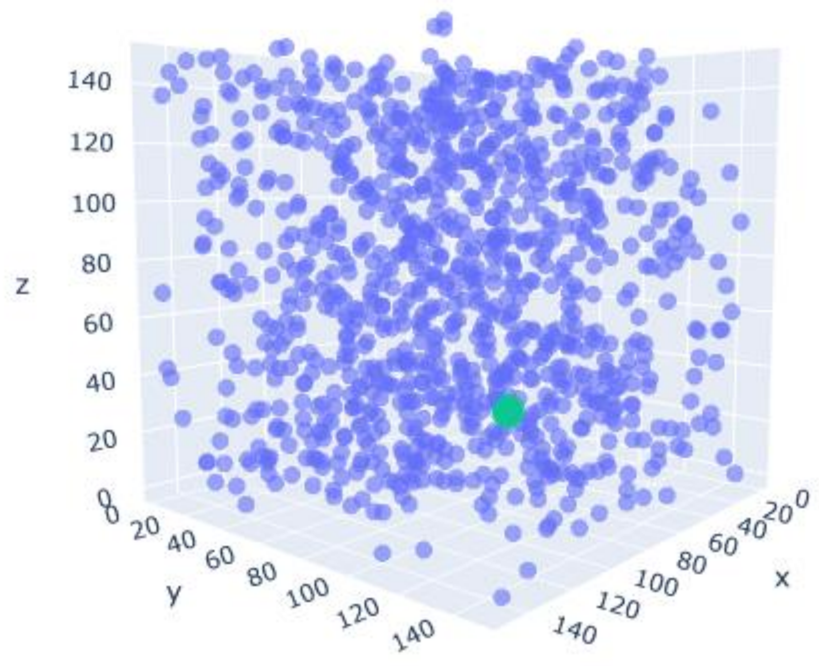


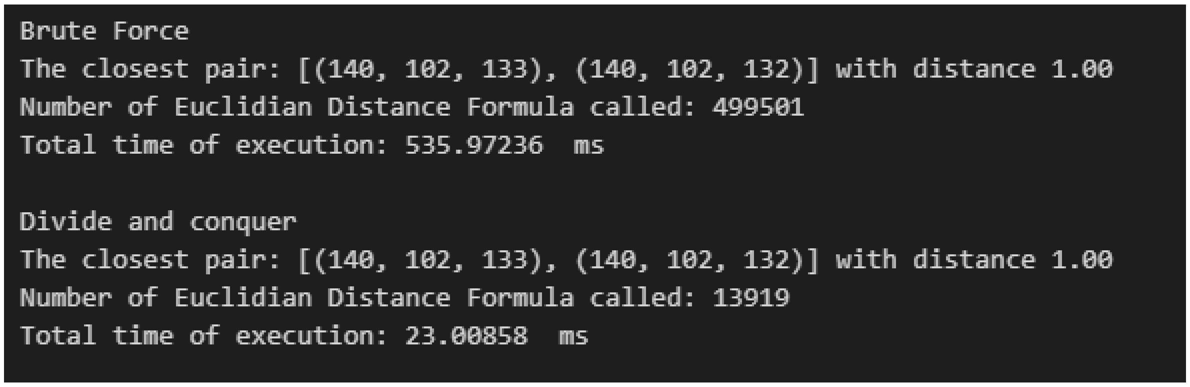
```
Brute Force
The closest pair: [(82, 127, 38), (83, 127, 38)] with distance 1.00
Number of Euclidian Distance Formula called: 499501
Total time of execution: 522.17364  ms

Divide and conquer
The closest pair: [(82, 127, 38), (83, 127, 38)] with distance 1.00
Number of Euclidian Distance Formula called: 13040
Total time of execution: 22.02654  ms
```



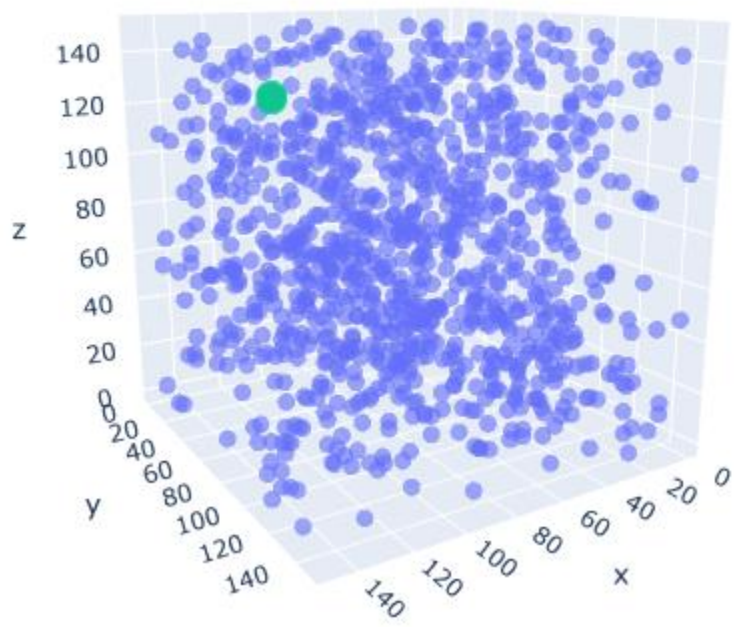


```
Brute Force
The closest pair: [(61, 84, 24), (60, 84, 25)] with distance 1.41
Number of Euclidian Distance Formula called: 499501
Total time of execution: 510.36596  ms

Divide and conquer
The closest pair: [(60, 84, 25), (61, 84, 24)] with distance 1.41
Number of Euclidian Distance Formula called: 13553
Total time of execution: 20.77866  ms
```





```
Brute Force
The closest pair: [(140, 102, 133), (140, 102, 132)] with distance 1.00
Number of Euclidian Distance Formula called: 499501
Total time of execution: 535.97236  ms

Divide and conquer
The closest pair: [(140, 102, 133), (140, 102, 132)] with distance 1.00
Number of Euclidian Distance Formula called: 13919
Total time of execution: 23.00858  ms
```



z
140
120
100
80
60
40
20
0
y
0
20
40
60
80
100
120
140
x
0
20
40
60
80
100
120
140

BAB V

TABEL

Poin	YA	TIDAK
1. Program berhasil dikompilasi tanpa kesalahan	✓	
2. Program berhasil *running*	✓	
3. Program dapat menerima masukan dan menuliskan luaran	✓	
4. Luaran program sudah benar (solusi *closest pair* benar)	✓	
5. Bonus 1 dikerjakan	✓	
6. Bonus 2 dikerjakan		✓

LINK REPOSITORY

https://github.com/maximatey/Tucil2_13521061