**Laporan Kecil 2 IF2211 Strategi Algoritma**

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

Description automatically generated

Disusun oleh:

Ahmad Ghulam Ilham (13521118)

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung

Tahun 2022/2023

# BAB I

**Deskripsi Tugas**

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 P1 = (x1, y1, z1) dan P2 = (x2, y2, z2) dihitung dengan rumus Euclidean berikut:

A picture containing text, watch

Description automatically generated

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.

Masukan program:

* n
* titik-titik (dibangkitkan secara acak) dalam koordinat (x, y, z)

Luaran program:

* sepasang titik yang jaraknya terdekat dan nilai jaraknya
* banyaknya operasi perhitungan rumus Euclidian
* waktu riil dalam detik (spesifikasikan komputer yang digunakan)

# BAB II

**Landasan Teori**

Divide and Conquer dulunya adalah strategi militer yang dikenal dengan nama divide ut imperes. Sekarang strategi tersebut menjadi strategi fundamental di dalam ilmu komputer dengan nama Divide and Conquer.

* Divide: membagi persoalan menjadi beberapa upa-persoalan yang memiliki kemiripan dengan persoalan semula namun berukuran lebih kecil (idealnya berukuran hampir sama)
* Conquer (solve): menyelesaikan masing-masing upa-persoalan (secara langsung jika sudah berukuran kecil atau secara rekursif jika masih berukuran besar)
* Combine: mengabungkan solusi masing-masing upa-persoalan sehingga membentuk solusi persoalan semula

Obyek persoalan yang dibagi : masukan (input) atau instances persoalan yang berukuran n seperti:

* tabel (larik),
* matriks,
* eksponen,
* polinom,
* dll, bergantung persoalannya.

Tiap-tiap upa-persoalan memiliki karakteristik yang sama (the same type) dengan karakteristik persoalan semula sehingga metode Divide and Conquer lebih natural diungkapkan dalam skema rekursif.

Skema Umum Algoritma Divide and Conquer adalah sebagai berikut.

Graphical user interface, text, application

Description automatically generated

# BAB III

**Aplikasi Algoritma Divide and Conquer**

Algoritma divide and conquer yang digunakan pada implementasi program tucil ini adalah sebagai berikut:

Inisialisasi:

1. Mengurutkan terlebih dahulu himpunan titik berdasarkan komponen absis (x) dari masing-masing titik.

Divide and Conquer:

1. Ambil sebuah bidang melebar sepanjang sumbu y dan sumbu z dengan posisi absis adalah median dari komponen absis semua titik.
2. Bagi himpunan titik P menjadi dua bagian, yaitu P1 (bagian kiri bidang datar) dan P2 (bagian kanan bidang datar).
3. Secara rekursif, kembali lakukan langkah 2 dan 3 sampai mencapai basis.
4. Jika sudah mencapai basis, yaitu ketika jumlah titik dalam himpunan P kurang dari sama dengan 3, gunakan Euclidean Distance untuk mencari jarak terdekat antara titik-titik tersebut.

Combine:

1. Setelah didapat jarak terdekat dari masing-masing himpunan titik (P1 dan P2), akan dibuat sebuah strip yang berisi titik-titik dengan jarak ke bidang datar pemisah himpunan sama dengan jarak terdekat antara P1 atau jarak terdekat antara P2 (digunakan yang lebih kecil).
2. Melakukan perhitungan Euclidean Distance pada titik-titik dalam strip
3. Menggabungkan solusi dengan memilih jarak terdekat antara pasangan titik pada P1, pasangan titik pada P2, atau pasangan titik dalam strip.

Untuk memenuhi algoritma divide and conquer tersebut, berikut adalah tipe data, fungsi, dan prosedur yang digunakan dalam tucil ini.

Tipe data:

|  |  |  |
| --- | --- | --- |
| Nama | Atribut | Deskripsi |
| Point | Self.x  Self.y  Self.z | Tipe data titik pada ruang 3D |

Fungsi:

|  |  |  |  |
| --- | --- | --- | --- |
| Nama | Parameter Input | Parameter Output | Deskripsi |
| Bruteforce | Point[] | Float, Point, Point | Mencari pasangan titik terdekat 3D dengan algoritma brute force |
| Conquer | Point[] | Float, Point, Point | Mencari pasangan titik terdekat 3D dengan algoritma divide and conquer |
| findMidPoint | Point[] | Integer | Mencari posisi median dari himpunan titik |
| findLeft | Point[], Integer | Point[] | Mencari titik yang berada di himpunan bagian kiri |
| findRight | Point[], Integer | Point[] | Mencari titik yang berada di himpunan bagian kanan |
| findMidStrip | Point[], Point[] | Float | Mencari posisi tengah bidang datar strip |
| Divide | Point[] | Point[], Point[], Float | Membagi himpunan titik P menjadi dua, yaitu P1 (bagian kiri) dan P2 (bagian kanan) |
| makeStrip | Point[], Float, Float | Point[] | Membuat strip berisi titik yang berada di dekat strip |
| sortStrip | Point[] | Point[] | Mengurutkan titik dalam strip berdasarkan komponen ordinat (y) dengan menggunakan algoritma quicksort |
| getInteger | - | Integer | Validasi input nilai n adalah integer |
| getValid | - | Integer | Validasi input nilai n >= 2 |
| generatePoints | Integer | Point[] | Membangkitkan secara acak titik dalam koordinat (x, y, z) |
| sortPoints | Point[] | Point[] | Mengurutkan titik berdasarkan komponen absis (x) dengan mneggunakan algoritma quicksort |
| findDistance | Point, Point | Float | Mencari jarak antara dua buah titik pada bidang 3D dengan Euclidean Distance |
| findNonStrip | Point[] | Float, Point, Point | Mencari jarak terdekat di antara n buah titik |
| findInStrip | Point[], Point, Point, Float | Float, Point, Point | Mencari jarak terdekat antara pasangan titik dalam strip |
| getCount | - | Integer | Menghitung banyak operasi Euclidean Distance dilakukan selama program berjalan |

Prosedur:

|  |  |  |
| --- | --- | --- |
| Nama | Parameter Input | Deskripsi |
| printPoints | Point[] | Menampilkan titik-titik pada bidang 3D ke terminal |
| displayPoints | Point[], Point[] | Menggambarkan titik dalam bidang 3D |

# BAB IV

**Implementasi dan Pengujian**

Implementasi bruteforce.py

Text

Description automatically generated

Implementasi conquer.py

Text

Description automatically generated

Implementasi divide.py

Text

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

Implementasi input.py

**Text

Description automatically generated**

Implementasi point.py

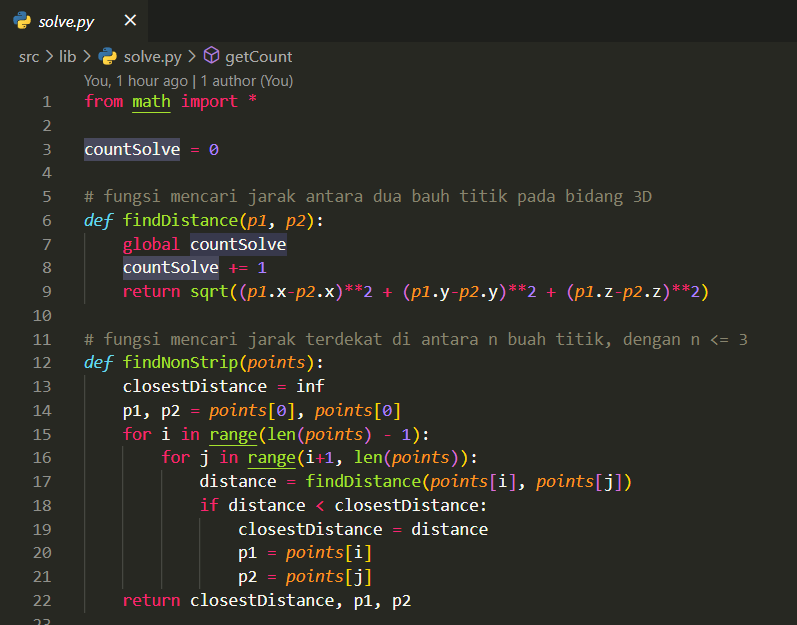
Text

Description automatically generated

Text

Description automatically generated

Implementasi solve.py



A screenshot of a computer

Description automatically generated with medium confidence

Implementasi main.py

Text

Description automatically generated

Text

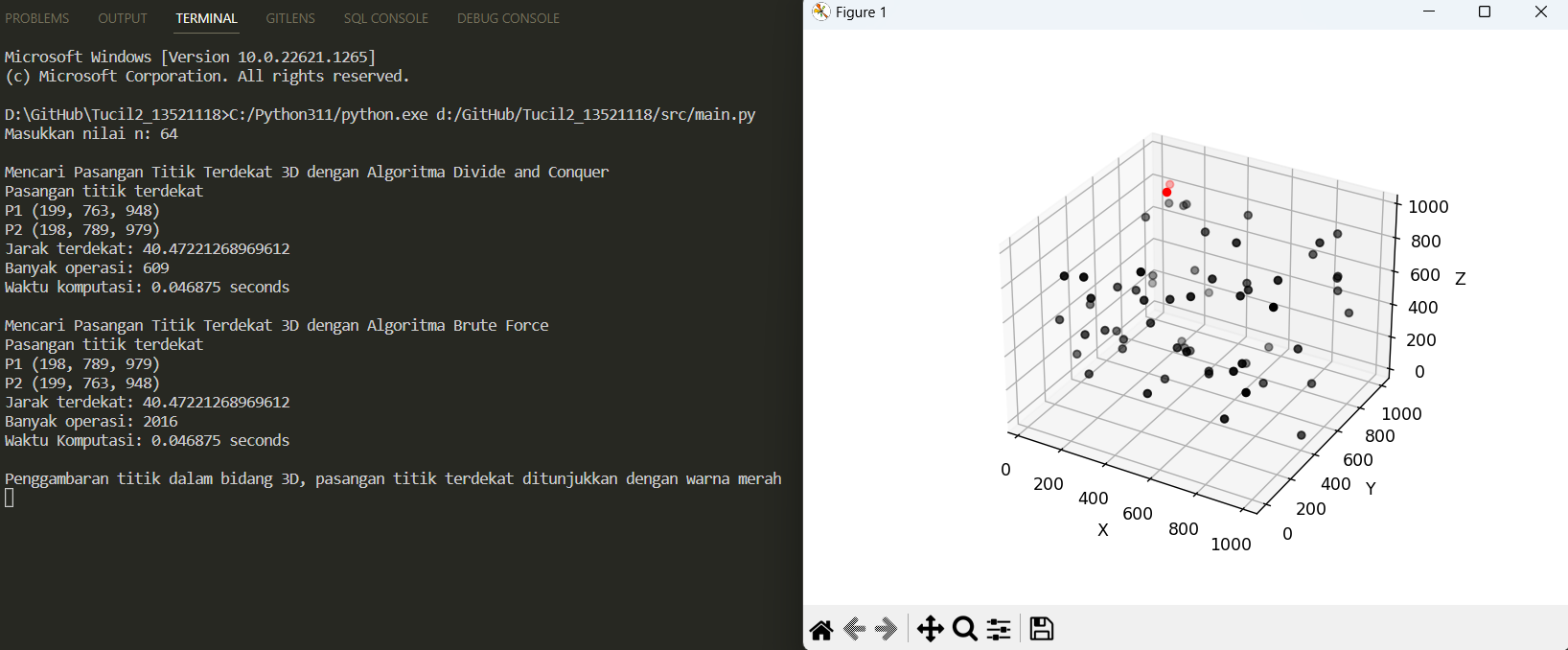
Description automatically generated

Hasil pengujian dengan n = 16

A picture containing graphical user interface

Description automatically generated

Hasil pengujian dengan n = 64



Hasil pengujian dengan n = 128

A picture containing graphical user interface

Description automatically generated

Hasil pengujian dengan n = 1000

Graphical user interface

Description automatically generated with low confidence

# BAB V

**Kesimpulan**

|  |  |  |
| --- | --- | --- |
| Poin | Ya | Tidak |
| 1. Program berhasil dikompilasi tanpa ada kesalahan | ✓ |  |
| 1. Program berhasil *running* | ✓ |  |
| 1. Program dapat menerima masukan dan menuliskan luaran | ✓ |  |
| 1. Luaran program sudah benar (solusi *closest pair* benar) | ✓ |  |
| 1. Bonus 1 dikerjakan | ✓ |  |
| 1. Bonus 2 dikerjakan |  | ✓ |

# DAFTAR PUSTAKA

Referensi

* <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Divide-and-Conquer-(2021)-Bagian1.pdfRepository>
* <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Divide-and-Conquer-(2021)-Bagian2.pdf>

Repository GitHub

<https://github.com/Agilham/Tucil2_13521118>