

IF2211 Strategi Algoritma

IMPLEMENTASI ALGORITMA DIVIDE AND CONQUER DALAM MENCARI PASANGAN TITIK TERDEKAT 3D

Laporan Tugas Kecil II

Disusun untuk memenuhi tugas mata kuliah Strategi Algoritma

Pada Semester 2 (dua) Tahun Akademik 2022/2023



Disusun Oleh:

Fakhri Muhammad Mahendra 13521045

Vanessa Rebecca Wiyono 13521151

**PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG**

**BANDUNG
2023**

DAFTAR ISI

BAB I: DESKRIPSI MASALAH	3
1.1 Algoritma <i>Divide and Conquer</i> pada <i>Closest Pair Problem</i>	3
BAB II: SOURCE PROGRAM	4
BAB III: TEST CASE	4
BAB IV: DAFTAR PUSTAKA	4
LAMPIRAN	4

BAB I DESKRIPSI MASALAH

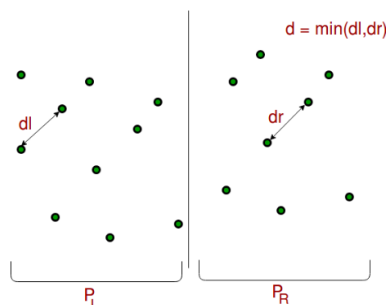
Algoritma *Divide and Conquer* pada *Closest Pair Problem*

Algoritma *Divide and Conquer* memecahkan suatu permasalahan dengan cara memecah atau membaginya menjadi beberapa bagian kecil sehingga akan lebih mudah untuk diselesaikan. Oleh karena itu, algoritma ini sering digunakan dalam memecahkan persoalan yang rumit. Berikut adalah langkah-langkah algoritma *Divide and Conquer* secara garis besar:

1. Divide : membagi masalah menjadi beberapa masalah yang lebih kecil
2. Conquer : Menyelesaikan setiap masalah kecil dan mendapatkan solusinya
3. Combine : menggabungkan solusi dari setiap masalah kecil untuk mendapat solusi secara keseluruhan

Closest Pair Problem merupakan masalah dalam komputasi geometri yang mencari pasangan titik terdekat diantara kumpulan titik dalam ruang n-dimensi. Pada umumnya, permasalahan ini digunakan dalam menyelesaikan masalah seperti navigasi, pencarian rute terpendek, dan pengenalan pola dalam data. Terdapat berbagai macam algoritma yang dapat digunakan untuk penyelesaiannya, dan salah satu alternatif algoritma yang dapat digunakan adalah algoritma *Divide and Conquer*. Berikut adalah langkah-langkahnya:

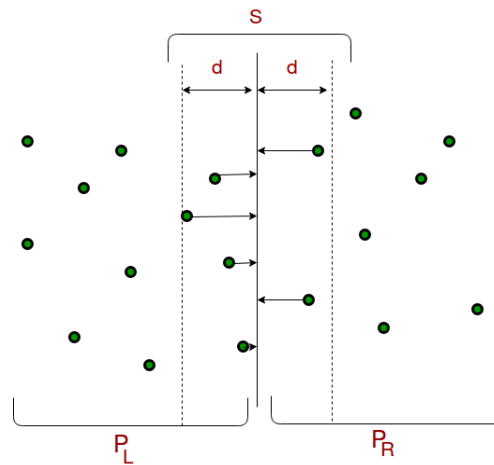
1. Urutkan titik-titik dalam himpunan terurut sesuai koordiat x
2. Pecah himpunan menjadi dua bagian yang sama besar dengan bidang vertical yang melalui titik median x
3. Selesaikan masalah secara rekursif pada dua himpunan titik yang lebih kecil dan dalam setiap rekursinya, jarak terdekat dari kedua himpunan subtitik diambil



Sumber Gambar:

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

4. Tentukan strip yang terletak disebelah kiri dan kanan bidang median dengan lebar dua kali jarak terdekat yang ditemukan pada langkah sebelumnya



Sumber Gambar:

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

5. Urutkan titik-titik dalam strip tersebut berdasarkan koordinat y dan periksa jarak antara setiap pasang titik berdekatan
6. Jika terdapat pasangan titik yang jaraknya lebih kecil dari jarak terdekat yang telah ditemukan pada langkah sebelumnya, maka jarak terdekat diperbarui
7. Tentukan dua bidang yang terletak pada jarak terdekat yang telah ditemukan pada langkah sebelumnya, dengan garis yang sejajar dengan sumbu z. Himpunan titik kemudian dibagi berdasarkan dua bidang ini
8. Selesaikan masalah secara rekursif pada dua himpunan titik yang lebih kecil. Dalam setiap rekursi, ambil jarak terdekat dari kedua himpunan subtitik
9. Kembalikan jarak terdekat beserta titiknya

Jika ditinjau berdasarkan kompleksitas waktunya, maka penggunaan algoritma divide and conquer untuk masalah closest pair dalam ruang 3D adalah $O(n \log^2 n)$, dimana n merupakan banyak titik dalam himpunan. kompleksitas waktu $O(n \log^2 n)$ berasal dari dua tahap rekursif, dimana setiap tahapnya memiliki kompleksitas waktu $O(n \log n)$ dan setiap titik dalam strip hanya diperiksa dengan paling banyak oleh 6 titik lainnya. Berbeda dengan kompleksitas dalam ruang 3D, kompleksitas waktu yang dihasilkan dalam ruang n -D adalah $O(n \log^{n-1} n)$. Hal ini disebabkan oleh jumlah perbandingan yang semakin banyak ketika dimensi ruang meningkat sehingga dilakukan tahap rekursif sebanyak n kali, dimana setiap tahap memiliki kompleksitas waktu $O(n \log n)$ dan setiap titik dalam strip hanya diperiksa dengan paling banyak $2^{(n-1)}$ titik lainnya.

BAB II SOURCE PROGRAM

BAB III TEST CASE

BAB IV DAFTAR PUSTAKA

GeeksforGeeks, url: <https://www.geeksforgeeks.org/divide-and-conquer/>

Munir, Rinaldi. Algoritma Divide and Conquer (Bagian 1). 2023. url: [https :/ / informatika . stei.itb.ac . id / ~rinaldi . munir / Stmik / 2020 - 2021 / Algoritma-Divide -and-Conquer-\(2021\)-Bagian1.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Divide-and-Conquer-(2021)-Bagian1.pdf)

Munir, Rinaldi. Algoritma Divide and Conquer (Bagian 1). 2023. url: [https :/ / informatika . stei.itb.ac . id / ~rinaldi . munir / Stmik / 2020 - 2021 / Algoritma-Divide -and-Conquer-\(2021\)-Bagian2.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Divide-and-Conquer-(2021)-Bagian2.pdf)

Munir, Rinaldi. Algoritma Divide and Conquer (Bagian 1). 2023. url: [https :/ / informatika . stei.itb.ac . id / ~rinaldi . munir / Stmik / 2020 - 2021 / Algoritma-Divide -and-Conquer-\(2021\)-Bagian3.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Divide-and-Conquer-(2021)-Bagian3.pdf)

Munir, Rinaldi. Algoritma Divide and Conquer (Bagian 1). 2023. url: [https :/ / informatika . stei.itb.ac . id / ~rinaldi . munir / Stmik / 2020 - 2021 / Algoritma-Divide -and-Conquer-\(2021\)-Bagian4.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Divide-and-Conquer-(2021)-Bagian4.pdf)

USCSB, url: <https://sites.cs.ucsb.edu/~suri/cs235/ClosestPair.pdf>

LAMPIRAN

Repository GitHub:

https://github.com/vanessrw/Tucil2_13521045_13521151

Checklist Table

Poin	Ya	Tidak
1. Program berhasil dikompilasi tanpa ada kesalahan		

2. Program berhasil <i>running</i>		
3. Program dapat menerima masukan dan menuliskan luaran		
4. Luaran program sudah benar (solusi <i>closest pair</i> benar)		
5. Bonus 1 dikerjakan		
6. Bonus 2 dikerjakan		