

Penerapan Algoritma Decrease and Conquer untuk Memperkirakan Alternatif Orientasi Gerak dalam Pengiriman Paket

Aurelius Marcel Candra - 13519198
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
13519198@std.stei.itb.ac.id

Kebutuhan jasa pengiriman paket pada zaman ini yang terus meningkat berimbas pada kompleksitas penentuan rute perjalanan yang paling efisien yang meningkat pula. Permasalahannya ini serupa dengan Travelling Salesperson Problem sehingga mengalami kendala yang sama, yaitu belum adanya algoritma yang “murah” dalam biaya komputasi untuk memecahkan persoalan ini. Dengan didasari kendala demikian, persoalan dapat dicoba diselesaikan dengan pendekatan lain, yang walaupun masih dalam ranah aproksimasi, tetapi menjamin kecepatan komputasi, yaitu dengan memanfaatkan prinsip algoritma *Decrease and Conquer*.

Kata Kunci: *decrease and conquer*, orientasi, titik, aproksimasi

I. PENDAHULUAN

Aktivitas pengiriman paket sudah dikenal sejak dahulu, dengan bentuk paling sederhananya berupa pengiriman surat. Beberapa tahun belakangan ini, jasa pengiriman paket kian populer dikarenakan kemudahan dan keandalan yang ditawarkannya, ditambah dengan kondisi pandemi sekarang yang mempersempit kemungkinan masyarakat untuk melakukan aktivitas berbelanja seperti kondisi normal. Dengan adanya jasa pengiriman paket, seseorang tidak perlu lagi repot-repot memperhatikan aspek higienis dengan cermat dan mengikuti protokol kesehatan ketika bepergian. Cukup dengan menggunakan aplikasi toko online, pilih dan pesan sebuah barang yang dibutuhkan, lakukan pembayaran dengan metode online pula, dan barang yang dipesan tadi akan datang dengan sendirinya ke rumah pemesan, dengan perantara jasa pengiriman paket tadi tentunya.



Gambar 1: Jasa Pengiriman Paket

Sumber: <https://rde.stanford.edu/studenthousing/mail-deliveries>

Proses pengiriman paket dengan batasan satu barang saja tampaknya sederhana, namun menjadi permasalahan baru jika jumlah paket yang perlu diantarakan bertambah banyak secara signifikan dan perlu diantarakan dalam satu kali perjalanan. Masalah serupa sudah pernah diutarakan sebelumnya, yaitu *Travelling Salesperson Problem* (TSP). TSP secara sederhana dapat dijabarkan sebagai problema menentukan rute perjalanan untuk melewati setiap kota tepat sekali dan *salesperson* ini dapat kembali ke kota pertama dia memulai perjalanannya. Pencarian rute perjalanan terpendek menjadi basis solusi, baik dalam konteks TSP maupun pengiriman paket.

Walaupun TSP sudah dikenal sebagai persoalan klasik, namun algoritma penyelesaian TSP yang “murah” dalam waktu komputasi masih belum ditemukan sampai sekarang. Konsep murah dalam hal ini diartikan sebagai kompleksitas waktu algoritma yang dalam bentuk polinomial. Konsekuensinya, persoalan yang serupa dengan TSP, seperti pengiriman paket ini juga belum dapat diselesaikan secara efisien.

Dalam implementasi secara nyata, terkait pengiriman paket, kendala mengenai rute terpendek ini dapat dirangkum menjadi lebih sederhana dalam bentuk pertanyaan “Dari mana perjalanan dimulai dan di mana perjalanan berakhir” atau dengan kata lain mencari orientasi perjalanan yang akan ditempuh. Dalam pertanyaan sederhana tersebut, terdapat pemangkasan pada aspek *salesperson* ini harus kembali ke tempat awal dia berangkat seperti pada TSP. Batasan ini digunakan untuk menyederhanakan permasalahan sekaligus merepresentasikan kendala umum yang sering muncul ketika dihadapkan pada persoalan yang kompleks, yaitu dari titik mana sebaiknya solusi mulai dicari. Dengan demikian, melalui makalah ini, penulis mencoba menggunakan pendekatan lain yang lebih sederhana dalam rangka menghasilkan alternatif solusi sebagai panduan dalam menginisiasi sebuah rute perjalanan pengiriman paket.

II. DASAR TEORI

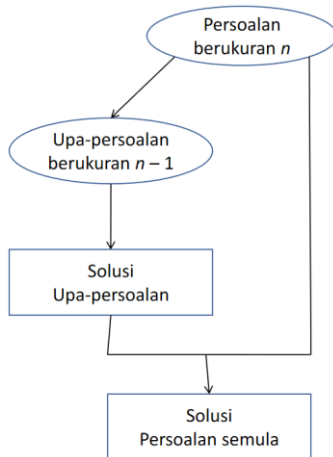
A. Algoritma Decrease and Conquer

Algoritma Decrease and Conquer adalah sebuah strategi algoritma yang mencoba menyelesaikan suatu persoalan dengan memecah persoalan tersebut menjadi bagian-bagian yang lebih kecil kemudian memilih satu dari banyak subpersoalan untuk melanjutkan penelusuran solusi atau penyelesaian persoalan utama. Selain jumlah subpersoalan yang diproses, aspek lain yang membedakan algoritma ini dengan algoritma *Divide and Conquer* adalah tidak adanya tahap *Combine*. Tahap ini terdapat pada algoritma Divide and Conquer sebagai tahap komplemen terhadap tahap pemrosesan seluruh subpersoalan supaya diperoleh satu solusi yang utuh, sedangkan pada Decrease and Conquer proses ini tidak dilakukan sehingga tahap Combine tidak diperlukan.

Algoritma Decrease and Conquer, dapat dipecah menjadi dua tahapan, yaitu tahap Decrease dan tahap Conquer. Tahap Decrease adalah tahap untuk memecah persoalan menjadi dua atau lebih subpersoalan yang lebih kecil. Sedangkan tahap Conquer diartikan sebagai tahap untuk memproses satu subpersoalan yang dapat dilakukan baik dengan metode rekursif maupun iteratif.

Pada algoritma Decrease and Conquer, terdapat tiga varian berdasarkan cara mereduksi persoalannya, yaitu:

1. *Decrease by a constant*: Varian ini mereduksi instans persoalan berdasarkan nilai konstanta yang sama setiap iterasi. Nilai konstanta yang umum digunakan adalah satu. Kompleksitas waktu varian ini sama dengan algoritma *Brute Force*. Contoh varian ini adalah *Topological Sorting* dan algoritma untuk menelusuri permutasi atau subhimpunan.

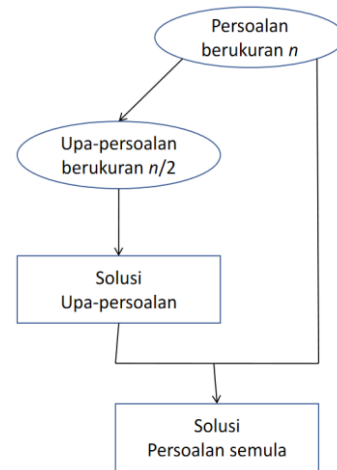


Gambar 2: Alur Decrease by a Constant

Sumber: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Decrease-and-Conquer-2021-Bagian2.pdf>

2. *Decrease by a constant factor*: Varian ini akan mereduksi instans persoalan berdasarkan nilai faktor konstanta yang sama untuk setiap iterasi. Nilai faktor konstanta yang umum digunakan adalah dua. Kompleksitas waktu varian ini adalah logaritmik dengan basis menyesuaikan nilai faktor yang digunakan.

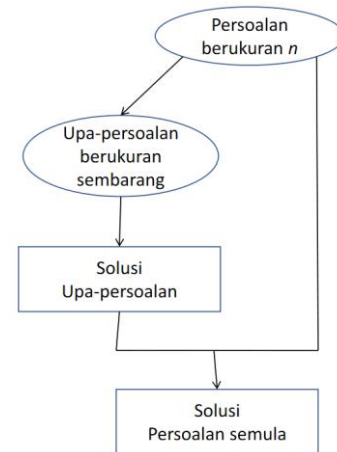
Contoh varian ini adalah *Binary Search* dan *Fake-coin Problem*.



Gambar 3: Alur Decrease by a Constant Factor

Sumber: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Decrease-and-Conquer-2021-Bagian2.pdf>

3. *Decrease by a variable size*: Varian ini akan mereduksi instans persoalan dengan ukuran yang berbeda untuk setiap iterasi. Kompleksitas waktu varian ini bergantung pada ukuran pemecahan persoalan untuk setiap iterasi. Contoh varian ini adalah *Interpolation Search* dan algoritma Euclid.



Gambar 4: Alur Decrease by a Variable Size

Sumber: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Decrease-and-Conquer-2021-Bagian2.pdf>

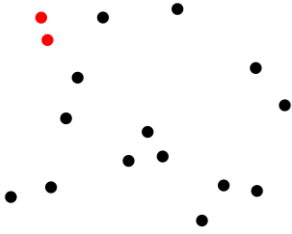
Algoritma Decrease and Conquer juga dapat dikategorikan dengan pendekatan implementasinya, yaitu:

1. *Top-down approach*: Pendekatan ini selalu dibentuk dalam pola rekursif.
2. *Bottom-up approach*: Pendekatan ini umumnya berbentuk pola iterative yang dimulai dari satu instans persoalan besar hingga direduksi menjadi subpersoalan yang berukuran cukup kecil sehingga mudah diselesaikan. [2]

III. IMPLEMENTASI

A. Definisi dan Cakupan Persoalan

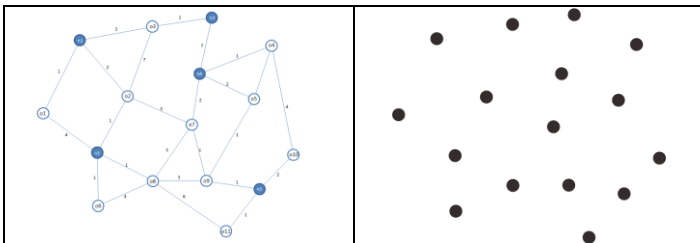
Persoalan yang dapat diselesaikan dengan algoritma Decrease and Conquer pada makalah ini memiliki batasan berupa sebaran titik-titik dalam jumlah yang terbatas pada bidang dua dimensi. Dengan analogi lain, persoalan yang dipakai di sini serupa pada persoalan mencari pasangan titik terdekat pada himpunan titik-titik secara acak; lebih jelasnya seperti pada gambar berikut.



Gambar 5: Contoh *Closest Pair Problem*

Sumber: https://en.wikipedia.org/wiki/Closest_pair_of_points_problem

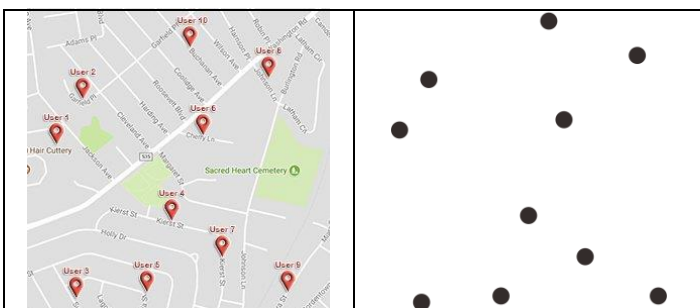
Bentukan persoalan tidak eksklusif hanya mencakup murni kumpulan titik-titik saja, tetapi juga meliputi hasil konversi dari bentuk lain, seperti graf atau sebuah peta berisi informasi himpunan alamat-alamat tujuan, menjadi bentuk kumpulan titik-titik seperti gambar di sebelumnya. Sebagai contoh sebagai berikut:



Gambar 6 & 7: Konversi Graf TSP
Sumber:

Kiri: <https://small-homes-decor.blogspot.com/2017/11/travelling-salesman.html>

Kanan: Dokumen penulis



Gambar 8 & 9: Konversi Peta Berisi Beberapa Titik Tujuan
Sumber:

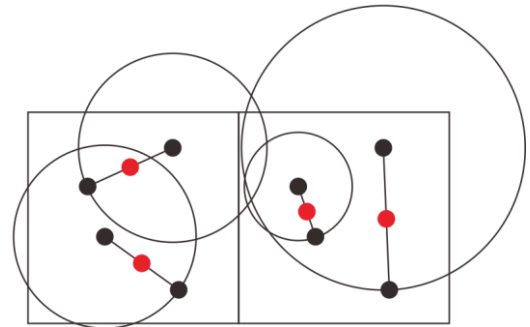
Kiri: <https://www.pushpendra.net/how-to-automatic-update-multiple-marker-position-on-google-map-dynamically-from-database-like-tracking-system/>

Kanan: Dokumen penulis

B. Strategi Penyelesaian

Langkah-langkah yang diterapkan terhadap persoalan hingga diperoleh solusi adalah sebagai berikut:

1. Titik awal perjalanan ditandai dengan warna atau simbol tertentu untuk memudahkan identifikasi. Abaikan titik awal ini dari langkah 2 sampai langkah 5.
2. Hubungkan setiap dua titik terdekat yang terdapat dalam persoalan. Jika jumlah titik ganjil, biarkan satu titik tidak memperoleh pasangan. Urutan titik yang akan dicari pasangan titik terdekatnya bisa menggunakan pendekatan terurut, misalnya dari titik dengan posisi teratas hingga terbawah atau dari kiri ke kanan, atau pendekatan acak namun memprioritaskan pasangan titik yang membentuk garis terpendek.



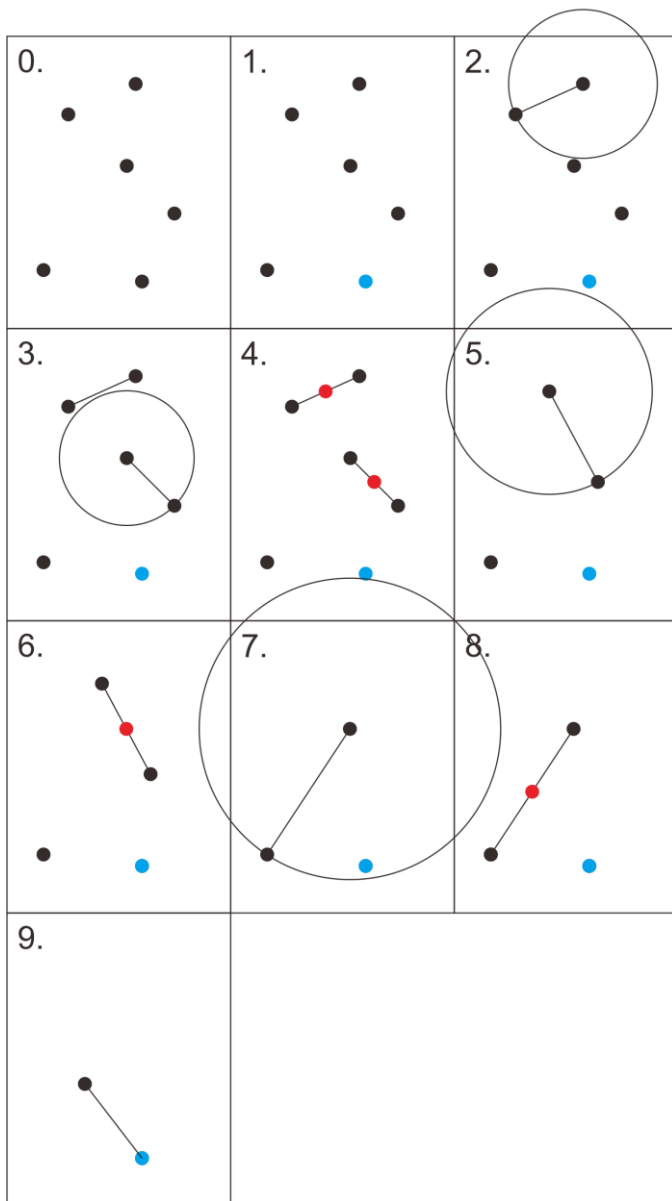
Gambar 10:

Kiri: Pendekatan Terurut

Kanan: Pendekatan Acak

Sumber: Dokumen penulis

3. Untuk setiap pasangan titik (garis) yang terbentuk, tandai titik tengah garis yang menghubungkan sebagai titik baru.
4. Kelompokkan titik-titik tengah tadi beserta satu titik yang tidak mendapat pasangan sebelumnya (jika ada) menjadi satu himpunan baru.
5. Ulangi langkah 2 sampai 4 untuk setiap kali himpunan titik baru yang terbentuk. Pengulangan ini berhenti ketika titik dalam himpunan tersebut berjumlah satu.
6. Hubungkan titik awal perjalanan pada langkah pertama dengan satu titik yang tersisa tadi. Garis yang terbentuk adalah aproksimasi orientasi gerak perjalanan.



Gambar 11: Langkah Strategi Penyelesaian
Sumber: Dokumen penulis

simplifikasi dan menjadi lebih sederhana. Secara matematis, bentuk reduksi yang dilakukan pada strategi penyelesaian ini adalah sebagai berikut:

$$N' = \lceil N/2 \rceil$$

Dengan:

- N' adalah jumlah titik pada subpersoalan (iterasi berikutnya)
- N adalah jumlah titik pada persoalan sekarang

Pada tahap Conquer, persoalan mencari pasangan titik terdekat menjadi persoalan menarik tersendiri, tetapi tidak menjadi fokus dalam pembahasan makalah ini. Dikarenakan pengerjaan makalah ini tidak didukung dengan komputasi nyata, pencarian pasangan titik terdekat ini dilakukan secara manual dan dengan alat bantu berupa aplikasi editor grafis; contohnya, pada pengerjaan makalah ini, digunakan aplikasi CorelDraw X8.

Metode yang digunakan untuk mencari pasangan titik terdekat adalah pemindaian melingkar atau *Radial Scanning*. Metode ini terinspirasi dari mekanisme kerja jangka untuk membentuk lingkaran dengan variasi radius yang terus bertambah sehingga menyerupai cara kerja radar pada lingkup area dua dimensi. Pemindaian dilakukan dengan menentukan satu titik sebagai pangkal atau pusat lingkaran dan lingkaran yang dibentuk dapat ditambah radiusnya hingga beririsan dengan titik lain. Titik pertama yang beririsan dengan sisi lingkaran merupakan titik terdekat terhadap titik pangkal tersebut. [3]

IV. PENGUJIAN DAN ANALISIS

A. Hasil Percobaan

Keterangan gambar:

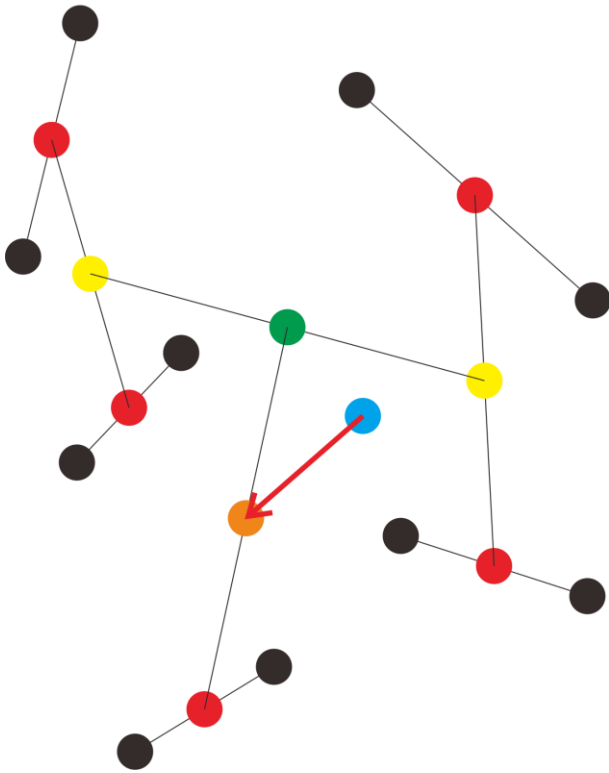
- Titik hitam: titik pada persoalan
- Titik biru: titik awal
- Titik merah: titik pada iterasi 1
- Titik kuning: titik pada iterasi 2
- Titik hijau: titik pada iterasi 3
- Titik jingga: titik pada iterasi 4
- Garis berarah merah: solusi persoalan

C. Korespondensi Strategi Penyelesaian Terhadap Algoritma Decrease and Conquer

Seperti dijelaskan pada dasar teori, algoritma Decrease and Conquer memiliki dua komponen tahapan, yaitu tahap Decrease dan tahap Conquer. Pada strategi penyelesaian yang dijabarkan di bagian sebelumnya, secara garis besar, tahap Decrease terdefinisi pada langkah 4, sedangkan tahap Conquer terdefinisi pada langkah 2 dan 3. Langkah 6 dapat dikategorikan secara khusus dan terpisah sebagai tahapan untuk memperoleh solusi keseluruhan dari algoritma ini.

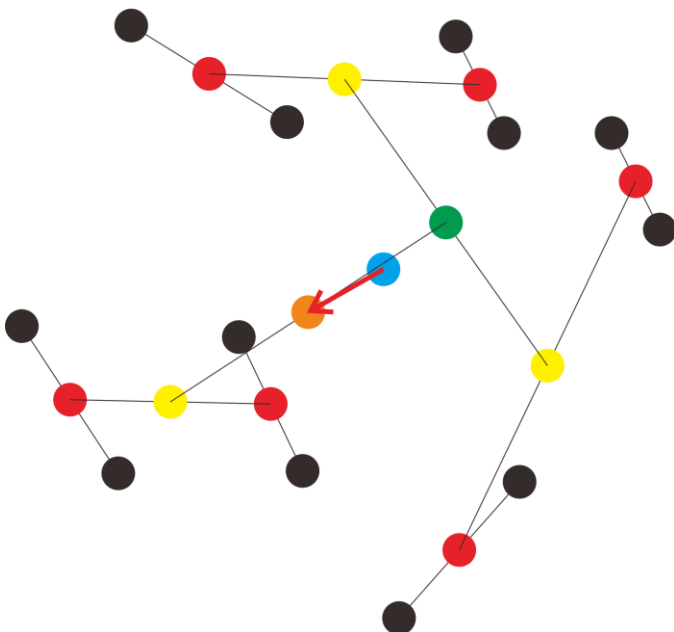
Tahap Decrease pada strategi penyelesaian ini tidak membagi persoalan menjadi subpersoalan-subpersoalan yang lebih kecil, namun lebih tepatnya melakukan reduksi pada setiap iterasi sehingga persoalan terus-menerus mengalami

- Percobaan 1 (himpunan titik sembarang, menggunakan pencarian pasangan titik dengan pendekatan acak):



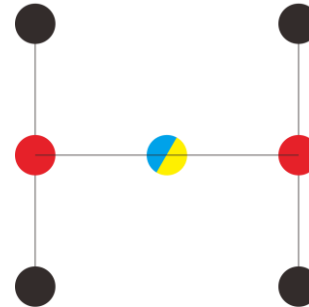
Gambar 12: Penyelesaian Himpunan Titik Sembarang
Sumber: Dokumen penulis

- Percobaan 2 (himpunan titik sembarang, menggunakan pencarian pasangan titik dengan pendekatan terurut dari posisi teratar hingga terbawah):



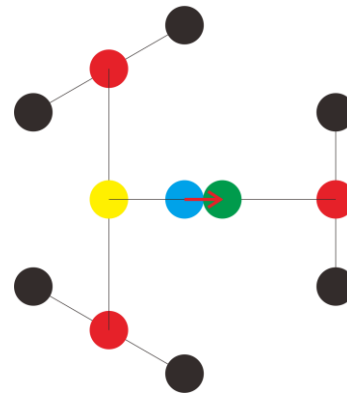
Gambar 13: Penyelesaian Himpunan Titik Sembarang
Sumber: Dokumen penulis

- Percobaan 3 (himpunan titik sudut persegi):



Gambar 14: Penyelesaian Himpunan Titik Sudut Persegi
Sumber: Dokumen penulis

- Percobaan 4 (himpunan titik sudut heksagon):



Gambar 15: Penyelesaian Himpunan Titik Sudut Heksagon
Sumber: Dokumen penulis

B. Analisis Hasil Percobaan

Percobaan 1 dan 2 mencoba mencari orientasi gerak perjalanan pada himpunan titik sembarang dengan percobaan 1 diawali jumlah titik (termasuk titik awal) 11 dan percobaan 2 berjumlah 13. Kedua percobaan memiliki kesamaan jumlah iterasi, yaitu sebanyak 4 kali, dengan bukti perhitungan matematisnya seperti berikut:

Iterasi	Percobaan 1	Percobaan 2
0 (awal)	11	13
1	$\lceil (11 - 1) / 2 \rceil = 5$	$\lceil (13 - 1) / 2 \rceil = 6$
2	$\lceil 5 / 2 \rceil = 3$	$\lceil 6 / 2 \rceil = 3$
3	$\lceil 3 / 2 \rceil = 2$	$\lceil 3 / 2 \rceil = 2$
4	$\lceil 2 / 2 \rceil = 1$	$\lceil 2 / 2 \rceil = 1$

Kesamaan lain yang dapat diperhatikan dari percobaan 1 dan 2 adalah bagaimana arah dari orientasi solusi yang terbentuk. Lebih jelas, berdasarkan gambar hasil percobaan, dapat dilihat bahwa orientasi (panah merah) pada kedua gambar memiliki pangkal dari wilayah yang cenderung memiliki banyak titik berdekatan atau bisa disebut kompleks dan berujung ke wilayah yang jarang terdapat titik berdekatan.

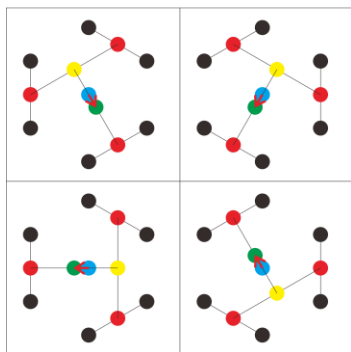
Percobaan 3 dan 4 mencoba mencari orientasi gerak perjalanan pada himpunan titik yang merupakan titik sudut dari sebuah bangun dua dimensi. Percobaan 3 menggunakan basis persegi, sedangkan percobaan 4 menggunakan basis heksagon. Kedua percobaan memiliki perbedaan jumlah iterasi, dengan bukti perhitungan matematisnya seperti berikut:

Iterasi	Percobaan 3	Percobaan 4
0 (awal)	5	7
1	$\lceil (5 - 1) / 2 \rceil = 2$	$\lceil (7 - 1) / 2 \rceil = 3$
2	$\lceil 2 / 2 \rceil = 1$	$\lceil 3 / 2 \rceil = 2$
3		$\lceil 2 / 2 \rceil = 1$

Percobaan 4 juga memiliki kesamaan dengan percobaan sebelumnya yang menyatakan orientasi solusi yang terbentuk cenderung berawal dari suatu kompleks titik dan berakhir pada wilayah yang jarang titik. Sedangkan pada percobaan 3 tidak ada orientasi yang terbentuk karena titik akhir dan titik awal identik.

Terkait hasil percobaan 3 ini, orientasi yang tidak terbentuk dapat diartikan bahwa orientasi perjalanan ke arah mana pun selalu sama efisiensinya. Dengan mengandalkan hipotesis ini, percobaan 4 membawa penulis pada pertanyaan lain, yaitu, "Mengapa hasil percobaan 4 tidak sama dengan hasil percobaan 3?".

Dasar pertimbangannya adalah kedua percobaan menggunakan basis poligon yang teratur atau lebih spesifiknya jarak antara setiap titik sudut dengan pusatnya selalu sama, seperti lingkaran. Maka, idealnya, bangun yang teratur, sebagaimana pada percobaan 3 dan 4, tidak memiliki orientasi. Kemudian, penulis menyadari bahwa bentuk orientasi yang dapat dibentuk tidak hanya satu, tetapi dimungkinkan bentukan lain yang bergantung pada bagaimana proses pembentukan pasangan titik untuk setiap iterasinya; contohnya sebagai berikut:



Gambar 16: Variasi Orientasi Solusi Percobaan 4
Sumber: Dokumen penulis

Dengan demikian, orientasi sebagai solusi yang terbentuk tidak mutlak hanya satu dan tidak dapat dipastikan benar. Solusi yang terbentuk ini sangat bergantung pada proses iterasi pembentukan pasangan titik. Walaupun tidak dapat diperoleh solusi tunggal dan tidak selalu benar, orientasi solusi yang terbentuk masih dapat digunakan sebatas perkiraan atau hasil aproksimasi terhadap persoalan.

Dikaitkan dengan kompleksitas waktu, strategi penyelesaian ini sesuai dengan dasar teori yang menyatakan bahwa varian *Decrease by a constant factor* memiliki kompleksitas logaritmik. Hal ini dibuktikan dengan jumlah iterasi yang perlu dilakukan untuk memperoleh solusi mengikuti klasifikasi jumlah titik (tidak meliputi titik awal) pada persoalan berdasarkan nilai 2^n . Lebih jelasnya digambarkan pada tabel berikut:

Jumlah titik	1	2	3-4	5-8	9-16	17-32
Iterasi	0	1	2	3	4	5

V. KESIMPULAN

Berdasarkan percobaan yang sudah dilakukan, dapat ditarik kesimpulan bahwa strategi penyelesaian ini belum berhasil menghasilkan satu solusi tunggal yang dijamin kebenarannya, tetapi hasil yang diperoleh masih dapat digunakan dalam ukuran "nilai" perkiraan. Dengan kompleksitas waktu logaritmik ($\log n$), strategi penyelesaian ini setidaknya unggul dalam kecepatan sehingga dapat membantu dalam pengambilan keputusan yang lebih cepat ketika meginisiasi sebuah perjalanan pengiriman paket.

UCAPAN TERIMA KASIH

Puji syukur kepada Tuhan Yang Maha Esa atas berkat dan rahmat-Nya sehingga makalah ini dapat diselesaikan dengan baik dan tepat waktu. Penulis menyampaikan terima kasih kepada Dr. Ir. Rinaldi Munir, M.T., atas bimbingannya selama satu semester dan penyediaan website yang berisi sumber materi dan daftar makalah sebagai sumber inspirasi. Penulis juga menyampaikan terima kasih kepada pemilik sumber-sumber referensi sehingga dapat mendukung isi dan kebenaran informasi dari makalah ini.

REFERENSI

- [1] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Decrease-and-Conquer-2021-Bagian1.pdf>. Diakses pada 9 Mei 2021.
- [2] <https://www.geeksforgeeks.org/decrease-and-conquer/>. Diakses pada 9 Mei 2021.
- [3] [https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Makalah/Makalah-Matdis-2020%20\(191\).pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Makalah/Makalah-Matdis-2020%20(191).pdf). Diakses pada 10 Mei 2021.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 11 Mei 2021

Aurelius Marcel Candra - 13519198