Optimalisasi Rute Pengumpulan Sampah Menggunakan Algoritma Program Dinamis

Novelya Putri Ramadhani - 13522096
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): 13522096@std.stei.itb.ac.id

***Abstract*—Optimasi rute pengumpulan sampah merupakan salah satu tantangan utama dalam pengelolaan sampah perkotaan. Makalah ini mengimplementasikan Algoritma Program Dinamis untuk meminimalkan total waktu perjalanan truk sampah dengan mempertimbangkan kapasitas truk dan waktu perjalanan antar tempat sampah. Algoritma ini memecah masalah menjadi submasalah yang lebih kecil dan membangun solusi optimal dengan prinsip optimalitas Bellman. Melalui simulasi pada berbagai skenario, hasil menunjukkan bahwa algoritma ini efektif dalam menghasilkan rute optimal untuk pengumpulan sampah pada skala kecil hingga menengah. Namun, kompleksitas waktu dan ruang meningkat secara eksponensial dengan jumlah tempat sampah, membatasi aplikasi pada skala yang lebih besar.**

Keywords—optimalisasi rute, pengumpulan sampah, Algoritma Program Dinamis, Vehicle Routing Problem.

I. PENDAHULUAN

Pengelolaan sampah yang efektif dan efisien merupakan salah satu tantangan utama yang dihadapi oleh kota-kota besar di seluruh dunia. Seiring dengan pertumbuhan populasi dan urbanisasi, volume sampah yang dihasilkan semakin meningkat, menuntut sistem pengelolaan yang lebih baik untuk mengurangi dampak lingkungan dan meningkatkan kualitas hidup masyarakat. Salah satu aspek penting dalam pengelolaan sampah adalah optimasi rute pengumpulan sampah. Rute pengumpulan yang optimal dapat mengurangi biaya operasional, menghemat bahan bakar, dan mengurangi emisi gas rumah kaca.

Optimalisasi rute pengumpulan sampah adalah masalah kompleks yang dapat dikategorikan sebagai Vehicle Routing Problem (VRP) yang tujuan utamanya adalah menentukan rute optimal bagi truk pengumpul sampah yang memiliki kapasitas terbatas untuk mengunjungi sejumlah tempat sampah dan kembali ke tempat pembuangan akhir. Penyelesaian masalah ini memerlukan pendekatan yang mempertimbangkan berbagai kendala seperti kapasitas truk, waktu perjalanan antar tempat sampah, dan jumlah tempat sampah yang harus dikunjungi.

Algoritma Program Dinamis adalah salah satu metode yang efektif untuk menyelesaikan masalah optimasi rute pengumpulan sampah. Algoritma ini memungkinkan pembagian masalah menjadi submasalah yang lebih kecil dan lebih sederhana, serta memanfaatkan hasil submasalah yang telah diselesaikan untuk membangun solusi optimal secara keseluruhan. Dalam konteks pengumpulan sampah, algoritma ini dapat digunakan untuk menentukan urutan kunjungan tempat sampah dan rute perjalanan yang meminimalkan waktu atau biaya operasional.

Makalah ini bertujuan untuk mengembangkan model optimasi rute pengumpulan sampah dengan menggunakan Algoritma Program Dinamis. Model ini akan mempertimbangkan berbagai faktor penting seperti kapasitas truk, waktu perjalanan, dan volume sampah di setiap tempat sampah. Dengan mengaplikasikan model ini, diharapkan dapat diperoleh solusi rute pengumpulan yang lebih efisien dan efektif, sehingga dapat memberikan kontribusi positif terhadap pengelolaan sampah yang lebih baik di kota-kota besar.

Gambar 1.1 Pengangkutan Sampah
(sumber: https://wartapemeriksa.bpk.go.id/wp-content/uploads/2022/07/sampah.jpg)

II. TEORI DASAR

A. *Pengelolaan Sampah Perkotaan*

Pengelolaan sampah adalah rangkaian kegiatan yang diperlukan untuk mengelola sampah sejak dari penghasilannya hingga pembuangan akhirnya. Kegiatan ini meliputi pengumpulan, pengangkutan, pemrosesan, daur ulang, dan pembuangan limbah. Dalam konteks perkotaan, pengelolaan sampah yang efektif sangat penting untuk menjaga kebersihan, kesehatan masyarakat, dan kelestarian lingkungan. Salah satu komponen kritis dari pengelolaan sampah adalah pengumpulan sampah, yang memerlukan perencanaan rute yang efisien untuk mengurangi biaya operasional dan dampak lingkungan.

B. *Vehicle Routing Problem (VRP)*

Vehicle Routing Problem (VRP) adalah masalah optimasi kombinatorial yang bertujuan untuk menentukan rute optimal bagi sejumlah kendaraan yang harus mengunjungi serangkaian lokasi dengan beberapa kendala tertentu. VRP pertama kali diperkenalkan oleh Dantzig dan Ramser pada tahun 1959 dalam konteks pengiriman minyak dengan truk tangki [3]. Masalah ini memiliki banyak varian, termasuk Capacitated VRP (CVRP), VRP with Time Windows (VRPTW), dan VRP with Pickup and Delivery (VRPPD). VRP secara umum dapat diformulasikan sebagai berikut:

- Input
    1) Sebuah graf berarah G = (V, E) dengan himpunan simpul V yang merepresentasikan depot dan pelanggan, serta himpunan sisi E yang merepresentasikan jalan antar simpul.
    2) Sebuah matriks jarak atau waktu perjalanan $D = [d_{ij}]$ untuk setiap pasangan simpul (i, j).
    3) Kapasitas kendaraan Q dan permintaan pelanggan q_i.
- Tujuan

  Meminimalkan total jarak atau waktu perjalanan kendaraan dengan memenuhi semua permintaan pelanggan tanpa melebihi kapasitas kendaraan.

C. *Program Dinamis*

Program Dinamis adalah metode pemecahan masalah dengan cara menguraikan solusi menjadi sekumpulan tahapan (stages) yang saling berkaitan. Metode ini digunakan untuk menyelesaikan persoalan optimasi (maksimisasi atau minimisasi) dengan memanfaatkan prinsip optimalitas Bellman, yang menyatakan bahwa solusi optimal dari suatu masalah dapat dibangun dari solusi optimal submasalah-submasalah nya [1]. Program dinamis biasanya diterapkan pada masalah yang memiliki struktur rekursif dan overlapping subproblems.

- Karakteristik Program Dinamis
    1) Overlapping Subproblems

       Masalah dapat dipecah menjadi submasalah yang lebih kecil, dan submasalah tersebut dipecahkan secara berulang kali.

    2) Optimal Substructure

       Solusi optimal dari masalah utama dapat diperoleh dengan mengkombinasikan solusi optimal dari submasalah-submasalah nya.

- Langkah-langkah Program Dinamis
    1) Definisikan Status (State)

       Mendefinisikan status yang merepresentasikan solusi dari submasalah.

    2) Identifikasi Basis (Base Case)

       Menentukan solusi dari sub masalah yang paling kecil.

    3) Rumuskan Relasi Rekurens (Recurrence Relation)

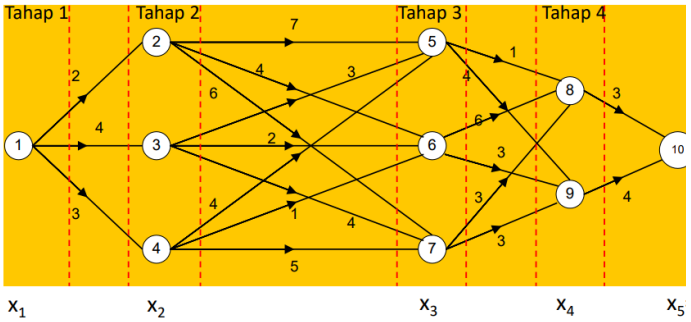
       Menyusun persamaan yang menghubungkan solusi dari sub masalah yang lebih kecil dengan masalah utama.

4) Hitung Nilai Optimal secara Iteratif

Menggunakan tabel untuk menghitung dan menyimpan solusi dari submasalah-submasalah hingga mencapai solusi dari masalah utama.

5) Rekonstruksi Solusi Optimal

Menentukan urutan pengambilan keputusan yang menghasilkan solusi optimal berdasarkan tabel yang telah dihitung.



Gambar 2.1 Contoh Pemodelan Persoalan

(sumber: https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Program-Dinamis-2020-Bagian1.pdf)

D. Algoritma Program Dinamis untuk VRP

Algoritma program dinamis dapat diterapkan pada VRP dengan memecah masalah utama menjadi submasalah yang lebih kecil, di mana setiap submasalah merepresentasikan rute pengumpulan sampah dari sejumlah tempat sampah dengan menggunakan satu atau lebih rute kendaraan. Pendekatan ini dapat dijelaskan sebagai berikut:

1. Definisikan Status (State)
   - $dp[i][j]$: Waktu minimum untuk mengumpulkan semua sampah dari subset j menggunakan i rute.
   - j: Representasi bitmask dari tempat sampah yang sudah dikunjungi.
2. Tahap (Stage)

   Setiap tahap merepresentasikan pengambilan sampah dari satu tempat sampah.
3. Basis (Base Case)

   $dp[0][0]$: Tidak ada waktu yang dibutuhkan untuk mengunjungi 0 tempat sampah dengan 0 rute.
4. Relasi Rekuren (Recurrence)
   - Jika kapasitas truk tidak penuh setelah menambah sampah dari tempat sampah k:

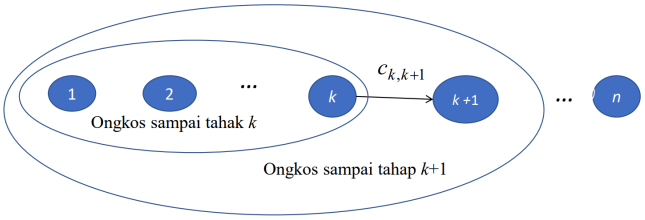
     $$dp[i][j \cup \{k\}] = min(dp[i][j \cup \{k\}], dp[i][j] + W[previous][k])$$
   - Jika kapasitas truk penuh dan harus kembali ke TPS:

     $$dp[i+1][j \cup \{k\}] = min(dp[i+1][j \cup \{k\}], dp[i][j] + W[previous][TPS] + W[TPS][k])$$

   dengan W[i][j] adalah waktu tempuh dari i ke j.

E. Prinsip Optimalitas Bellman

Prinsip optimalitas Bellman menyatakan bahwa solusi optimal dari suatu masalah dapat diperoleh dengan menggabungkan solusi optimal dari submasalah-submasalah nya. Prinsip ini merupakan dasar dari algoritma program dinamis dan memastikan bahwa solusi yang ditemukan adalah optimal. Dalam konteks VRP, prinsip ini berarti bahwa rute optimal untuk mengumpulkan sampah dapat dibangun dengan menggabungkan rute optimal dari subset tempat sampah yang lebih kecil.



Gambar 2.2 Prinsip Optimalitas
(sumber: https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Program-Dinamis-2020-Bagian1.pdf)

III. PEMBAHASAN

A. Deskripsi Persoalan

Pengelolaan sampah yang efisien dan efektif merupakan tantangan utama di kota-kota besar. Salah satu komponen utama dalam pengelolaan sampah adalah pengumpulan sampah. Tujuan utama dalam optimasi rute pengumpulan sampah adalah meminimalkan total waktu perjalanan truk pengumpul sampah sambil memastikan semua sampah diambil tanpa melebihi kapasitas truk.

Penelitian ini menggunakan pendekatan Algoritma Program Dinamis untuk memecahkan masalah optimasi rute pengumpulan sampah. Pertimbangan yang diambil meliputi beberapa faktor penting:

- Jumlah tempat sampah yang harus dikunjungi.
- Kapasitas truk pengumpul sampah.
- Waktu perjalanan antar tempat sampah dan antara tempat sampah dengan Tempat Pembuangan Sementara (TPS).

Masalah ini dimodelkan sebagai masalah Vehicle Routing Problem (VRP), dimana setiap tempat sampah memiliki volume sampah yang harus diangkut dan truk pengumpul sampah harus kembali ke TPS jika kapasitasnya penuh.

B. Implementasi Algoritma

Langkah-langkah implementasi algoritma program dinamis dapat dijelaskan sebagai berikut:

1. Status (State)

   Tempat sampah, dimisalkan TS, yang akan dikunjungi dan memiliki bobot berupa banyak sampah.

2. Tahap (Stage)

   Setiap tahap merepresentasikan pengambilan sampah dari satu tempat sampah.

3. Basis (Base Case)

   f(s) = W[0][i] yaitu waktu dari TPS ke tempat sampah.

4. Relasi Rekuren (Recurrence)
   - Jika kapasitas truk tidak penuh setelah menambah sampah dari tempat sampah k:

     $$fk(s) = min\{fk - 1(s) + W[k][k + 1]\}$$

   - Jika kapasitas truk penuh dan harus kembali ke TPS:

     $$fk(s) = min\{fk - 1(s) + W[k][k + 1] + W[0][k]\}$$

5. Perhitungan Dinamis
   - Untuk setiap tahap, periksa setiap subset ‘mask’ dan perbarui waktu perjalanan untuk setiap tempat sampah k yang belum dikunjungi.
   - Periksa apakah kapasitas truk cukup untuk mengambil sampah dari tempat sampah k. Jika tidak, tambahkan rute baru dan kembali ke TPS sebelum melanjutkan perjalanan.
6. Rekonstruksi Solusi Optimal

   Setelah menghitung nilai optimal dengan tabel program dinamis, rekonstruksi solusi optimal dilakukan dengan menelusuri kembali tabel untuk menentukan urutan pengambilan keputusan yang menghasilkan solusi optimal.

Misalkan terdapat truk sampah dengan kapasitas 8 yang akan mengangkut sampah dari 3 tempat sampah (TS1, TS2, TS3) yang mempunyai berat (4, 5, 3). Waktu tempuh antar tempat sampah dan antara tempat sampah dengan TPS direpresentasikan dalam bentuk matriks sebagai berikut:

W = [[0, 2, 3, 4],
[2, 0, 2, 3],
[3, 2, 0, 2],
[4, 3, 2, 0]]

dengan W[i][j] adalah waktu tempuh dari i ke j dan indeks 0 adalah TPS.

a) Tahap 1: Mengunjungi tempat sampah pertama

TABLE I. TAHAP 1

TS (tujuan)	TPS (asal)	
	Waktu	***Kapasitas Truk***
1	2	4
2	3	5
3	4	3

- Waktu dari TPS ke TS1 = 2
- Waktu dari TPS ke TS2 = 3
- Waktu dari TPS ke TS3 = 4

b) Tahap 2: Mengunjungi tempat sampah kedua

TABLE II. TAHAP 2

TS (tujuan)	**TS1**		**TS2**		**TS3**		**Solusi Optimal**
	Waktu	***Kapasitas Truk***	***Waktu***	***Kapasitas Truk***	***Waktu***	***Kapasitas Truk***	
1	∞	∞	8	4	7	7	TS3
2	7	5	∞	∞	6	8	TS3
3	5	7	5	8	∞	∞	TS1, 2

∞: tidak bisa dihitung

- Dari TS1, jika ke TS2, harus kembali ke TPS dulu karena kapasitas total melebihi 8. Waktu = 2 + 2 = 4. Lalu, untuk ke TS2, waktu minimal adalah 3 (langsung dari TPS). Total waktu = 4 + 3 = 7.
- Dari TS1, jika ke TS3, kapasitas total menjadi 7. Waktu = 2 + 3 = 5.

- Dari TS2,jika ke TS1, harus kembali ke TPS dulu karena kapasitas total melebihi 8. Waktu = 3 + 3 = 6. Lalu, untuk ke TS1, waktu minimal adalah 2 (langsung dari TPS). Total waktu = 6 + 2 = 8.
- Dari TS2, jika ke TS3, kapasitas total menjadi 8. Waktu = 3 + 2 = 5.
- Dari TS3 ke TS1, bisa langsung dengan total waktu 7.
- Dari TS3 ke TS2, bisa langsung dengan total waktu 5.

c) Tahap 3: Mengunjungi tempat sampah ketiga

TABLE III. TAHAP 3

TS (tujuan)	TS31		TS32		TS13		TS23	
	Waktu	*Kapasitas Truk*	*Waktu*	*Kapasitas Truk*	*Waktu*	*Kapasitas Truk*	*Waktu*	*Kapasitas Truk*
1	∞	∞	11	4	∞	∞	11	4
2	12	3	∞	∞	12	5	∞	∞
3	∞	∞	∞	∞	∞	∞	∞	∞

∞: tidak bisa dihitung

- Dari TS1, solusi optimal sebelumnya adalah dari TS3. Truk harus kembali ke TPS terlebih dahulu, sehingga total waktu minimal = 7 + 2 + 3 = 12
- Dari TS2, solusi optimal sebelumnya adalah dari TS3. Truk harus kembali ke TPS terlebih dahulu, sehingga total waktu minimal = 6 + 3 + 2 = 11
- Dari TS3, solusi optimal sebelumnya adalah dari TS1 atau TS2. Truk harus kembali ke TPS terlebih dahulu, sehingga total waktu minimal dari TS1 = 5 + 4 + 3 = 12 dan dari TS2 = 5 + 4 + 2 = 11

Didapatkan waktu terpendek untuk mengambil sampah dari ketiga tempat sampah adalah 11 dengan 2 kemungkinan rute, yaitu TPS - TS2 - TS3 - TPS - TS1 dan TPS - TS3 - TS2 - TPS - TS1.

C. Analisis Kompleksitas

- Kompleksitas Waktu

  Ada 2^n kemungkinan subset dari n tempat sampah. Ini adalah bagian eksponensial dari kompleksitas yang tidak bisa dihindari karena setiap kombinasi tempat sampah yang mungkin perlu diperiksa. Untuk setiap subset dipertimbangkan hingga K rute, dimana K adalah jumlah rute maksimal yang dibutuhkan. Dalam skenario terburuk, K bisa menjadi n jika truk harus kembali ke TPS setelah mengunjungi setiap tempat sampah. Untuk setiap kombinasi subset dan rute dilakukan iterasi untuk memperbarui status berdasarkan perjalanan ke setiap tempat sampah yang mungkin, melibatkan pemeriksaan waktu perjalanan dan kapasitas truk untuk memastikan apakah sampah dari tempat sampah selanjutnya bisa diambil atau truk harus kembali ke TPS. Oleh karena itu, algoritma ini memiliki kompleksitas $O(2^n . n . K)$ dengan n adalah jumlah tempat sampah dan K adalah jumlah rute maksimal.
- Kompleksitas Ruang

  Kompleksitas ruang dari algoritma ini adalah $O(2^n . K)$ untuk menyimpan tabel dinamis dp, yaitu hasil perhitungan untuk setiap subset dan rute yang mungkin.
- Hasil Analisis

  Pada masalah dengan skala kecil hingga menengah (misalnya, $n \leq 10$), algoritma ini memberikan solusi optimal dengan waktu komputasi yang dapat diterima. Untuk $n = 10$, perlu diperiksa $2^{10} = 1024$ subset, yang masih bisa dikelola oleh komputer modern dalam waktu yang wajar. Namun, pada masalah dengan skala besar (misalnya, $n > 15$), kompleksitas eksponensial dari algoritma ini menjadi kendala besar. Untuk $n = 15$, perlu diperiksa $2^{15} = 32768$ subset yang memerlukan waktu komputasi dan memori yang signifikan.
- Evaluasi

  Algoritma program dinamis memberikan solusi optimal dengan memastikan semua kemungkinan subset dan rute diperiksa secara sistematis. Namun, kompleksitas eksponensialnya menjadi kendala untuk skala besar. Keterbatasan utama adalah kompleksitas waktu dan ruang yang tinggi untuk masalah dengan banyak tempat sampah. Dalam aplikasi nyata dengan jumlah tempat sampah yang sangat besar, algoritma ini mungkin tidak praktis tanpa optimasi tambahan.

IV. PENUTUP

Pengelolaan sampah yang efisien adalah salah satu tantangan utama yang dihadapi oleh kota-kota besar. Optimasi rute pengumpulan sampah memainkan peran penting dalam mengurangi biaya operasional, menghemat bahan bakar, dan mengurangi emisi gas rumah kaca. Dalam penelitian ini, Algoritma Program Dinamis telah diterapkan untuk menyelesaikan masalah optimasi rute pengumpulan sampah

dengan mempertimbangkan kapasitas truk dan waktu perjalanan antar tempat sampah.

Algoritma ini memecah masalah utama menjadi submasalah yang lebih kecil, menggunakan prinsip optimalitas Bellman untuk membangun solusi optimal dari submasalah-submasalah tersebut. Hasil penelitian menunjukkan bahwa Algoritma Program Dinamis mampu memberikan solusi optimal untuk berbagai skenario pengujian, termasuk variasi jumlah tempat sampah dan kapasitas truk.

Namun, kompleksitas waktu dan ruang dari algoritma ini meningkat secara eksponensial dengan jumlah tempat sampah, sehingga menjadi kurang praktis untuk masalah skala besar. Oleh karena itu, untuk aplikasi nyata dengan jumlah tempat sampah yang sangat besar, diperlukan pendekatan lain seperti heuristik atau algoritma metaheuristik yang dapat memberikan solusi mendekati optimal dalam waktu yang lebih singkat.

UCAPAN TERIMA KASIH

Penulis ingin mengucapkan terima kasih kepada seluruh dosen pengampu IF2211 terutama Bu Dr. Nur Ulfa Maulidevi, S.T, M.Sc. sebagai dosen kelas 02 IF2211 Strategi Algoritma yang menerangkan materi terkait Program Dinamis sehingga saya bisa mengaplikasikannya pada tugas makalah ini. Terima kasih juga kepada Pak Dr. Ir. Rinaldi, M.T. yang telah menyediakan banyak sumber belajar bagi mahasiswa untuk mata kuliah Strategi Algoritma di website.

REFERENSI

[1] R. Munir, "Program Dinamis Bagian I," Bahan Kuliah IF2211 Strategi Algoritma, Program Studi Teknik Informatika, STEI-ITB, 2020.

[2] R. Munir, "Program Dinamis Bagian II," Bahan Kuliah IF2211 Strategi Algoritma, Program Studi Teknik Informatika, STEI-ITB, 2020.

[3] G. B. Dantzig and J. H. Ramser, "The Truck Dispatching Problem," Management Science, vol. 6, no. 1, pp. 80-91, 1959.

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, 12 Juni 2024

Novelya Putri Ramadhani 13522096