

TUGAS AKHIR - EF234801

REDUCED-COST VOGEL METHOD (RVM) BERBASIS REDUKSI ZERO-POINT UNTUK KONSTRUKSI INITIAL BASIC FEASIBLE SOLUTION PADA BALANCED TRANSPORTATION PROBLEM

MUHAMMAD RAYYAAN FATIKHAHUR RAKHIM

NRP 5025221047

Dosen Pembimbing

Dr. Bilqis Amaliah, . S.Kom., M.Kom.

NIP 197509142001122002

Ilham Gurat Adillion, S.Kom., M.Eng.

NIP 1995202411009

Program Studi Strata 1 (S1) Teknik Informatika

Departemen Teknik Informatika

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya

2026

Halaman ini sengaja dikosongkan



TUGAS AKHIR - EF234801

**REDUCED-COST VOGEL METHOD (RVM) BERBASIS
REDUKSI ZERO-POINT UNTUK KONSTRUKSI INITIAL
BASIC FEASIBLE SOLUTION PADA BALANCED
TRANSPORTATION PROBLEM**

MUHAMMAD RAYYAAN FATIKHAHUR RAKHIM

NRP 5025221047

Dosen Pembimbing

Dr. Bilqis Amaliah,. S.Kom., M.Kom.

NIP 197509142001122002

Ilham Gurat Adillion, S.Kom., M.Eng.

NIP 1995202411009

Program Studi Strata 1 (S1) Teknik Informatika

Departemen Teknik Informatika

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya

2026

Halaman ini sengaja dikosongkan



FINAL PROJECT - EF234801

**REDUCED-COST VOGEL METHOD (RVM) BASED ON
ZERO-POINT REDUCTION FOR INITIAL BASIC FEASIBLE
SOLUTION CONSTRUCTION IN THE BALANCED
TRANSPORTATION PROBLEM**

MUHAMMAD RAYYAAN FATIKHAHUR RAKHIM

NRP 5025221047

Advisor

Dr. Bilqis Amaliah,. S.Kom., M.Kom.

NIP 197509142001122002

Ilham Gurat Adillion, S.Kom., M.Eng.

NIP 1995202411009

Undergraduate Study Program of Informatics Engineering

Department of Informatics Engineering

Faculty of Intelligent Electrical and Informatics Technology

Institut Teknologi Sepuluh Nopember

Surabaya

2026

This page is intentionally left blank

LEMBAR PENGESAHAN

REDUCED-COST VOGEL METHOD (RVM) BERBASIS REDUKSI ZERO-POINT UNTUK KONSTRUKSI INITIAL BASIC FEASIBLE SOLUTION PADA BALANCED TRANSPORTATION PROBLEM

TUGAS AKHIR

Diajukan untuk memenuhi salah satu syarat
memperoleh gelar Sarjana Komputer pada
Program Studi S-1 Teknik Informatika
Departemen Teknik Informatika
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember

Oleh: **Muhammad Rayyaan Fatikhahur Rakhim**
NRP. 5025221047

Disetujui oleh Tim Penguji Tugas Akhir:

1.Dr. Bilqis Amaliah,. S.Kom., M.Kom.

2.Ilham Gurat Adillion, S.Kom., M.Eng.

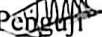
3.Prof. Dr. Eng. Nanik Suciati, S.Kom., M.Kom.

4.Dr. Eng. Muhamad Hilmil Muchtar Aditya Pradana, S.Kom., M.Sc.


Pembimbing


Ko-pembimbing


Penguji


Penguji

SURABAYA
1 Juli 2026

Halaman ini sengaja dikosongkan

APPROVAL SHEET

REDUCED-COST VOGEL METHOD (RVM) BASED ON ZERO-POINT REDUCTION FOR INITIAL BASIC FEASIBLE SOLUTION CONSTRUCTION IN THE BALANCED TRANSPORTATION PROBLEM

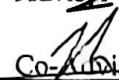
FINAL PROJECT

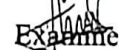
Submitted to fulfill one of the requirements
for obtaining a Bachelor of Computer Science degree at
Undergraduate Study Program of Informatics Engineering
Department of Informatics Engineering
Faculty of Intelligent Electrical and Informatics Technology
Institut Teknologi Sepuluh Nopember

By: **Muhammad Rayyaan Fatikhahur Rakhim**
NRP. 5025221047

Approved by Final Project Examiner Team:

1. Dr. Bilqis Amaliah., S.Kom., M.Kom.
2. Ilham Gurat Adillion, S.Kom., M.Eng.
3. Prof. Dr. Eng. Nanik Suciati, S.Kom., M.Kom.
4. Dr. Eng. Muhamad Hilmi Muchtar Aditya Pradana, S.Kom., M.Sc.


Adviser

Co-Adviser

Examiner

Examiner

SURABAYA
July 1, 2026

This page is intentionally left blank

PERNYATAAN ORISINALITAS

Yang bertanda tangan dibawah ini:

Nama Mahasiswa / NRP : Muhammad Rayyaan Fatikhahur Rakhim / 5025221047
Departemen : Teknik Informatika
Dosen Pembimbing / NIP : Dr. Bilqis Amaliah, S.Kom., M.Kom. / 197509142001122002
Dosen Ko-Pembimbing / NIP : Ilham Gurat Adillion, S.Kom., M.Eng. / 1995202411009

Dengan ini menyatakan bahwa Tugas Akhir dengan judul "*REDUCED-COST VOGEL METHOD (RVM) BERBASIS REDUKSI ZERO-POINT UNTUK KONSTRUKSI INITIAL BASIC FEASIBLE SOLUTION PADA BALANCED TRANSPORTATION PROBLEM*" adalah hasil karya sendiri, bersifat orisinal, dan ditulis dengan mengikuti kaidah penulisan ilmiah.

Bilamana di kemudian hari ditemukan ketidaksesuaian dengan pernyataan ini, maka saya bersedia menerima sanksi sesuai dengan ketentuan yang berlaku di Institut Teknologi Sepuluh Nopember.

Mengetahui
Dosen Pembimbing

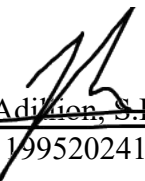

Dr. Bilqis Amaliah, S.Kom., M.Kom.
NIP. 197509142001122002

Surabaya, 1 Juli 2026

Mahasiswa


Muhammad Rayyaan Fatikhahur Rakhim
NRP. 5025221047

Dosen Ko-Pembimbing


Ilham Gurat Adillion, S.Kom., M.Eng.
NIP. 1995202411009

Halaman ini sengaja dikosongkan

STATEMENT OF ORIGINALITY

The undersigned below:

Name of student / NRP : Muhammad Rayyaan Fatikhahur Rakhim / 5025221047
Department : Informatics Engineering
Advisor / NIP : Dr. Bilqis Amaliah, S.Kom., M.Kom. / 197509142001122002
Co-Advisor / NIP : Ilham Gurat Adillion, S.Kom., M.Eng. / 1995202411009

Hereby declared that the Final Project with the title of "REDUCED-COST VOGEL METHOD (RVM) BASED ON ZERO-POINT REDUCTION FOR INITIAL BASIC FEASIBLE SOLUTION CONSTRUCTION IN THE BALANCED TRANSPORTATION PROBLEM" is the result of my own work, is original, and is written by following the rules of scientific writing.

If in future there is a discrepancy with this statement, then I am willing to accept sanctions in accordance with provisions that apply at Institut Teknologi Sepuluh Nopember.

Acknowledged by

Advisor



Dr. Bilqis Amaliah, S.Kom., M.Kom.
NIP. 197509142001122002

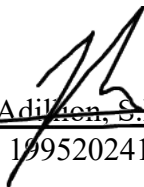
Surabaya, July 1, 2026

Student



Muhammad Rayyaan Fatikhahur Rakhim
NRP. 5025221047

Co-advisor



Ilham Gurat Adillion, S.Kom., M.Eng.
NIP. 1995202411009

This page is intentionally left blank

PERNYATAAN KODE ETIK PENGGUNAAN AI GENERATIF

Code of Conduct Statement: Generative AI or AI-Assisted Usage

Saya yang bertanda tangan di bawah ini:

I, the undersigned:

Nama Mahasiswa / NRP : Muhammad Rayyaan Fatikhahur Rakhim / 5025221047
Program Studi : Teknik Informatika
Judul Tugas Akhir : *REDUCED-COST VOGEL METHOD (RVM) BERBASIS REDUKSI ZERO-POINT UNTUK KONSTRUKSI INITIAL BASIC FEASIBLE SOLUTION PADA BALANCED TRANSPORTATION PROBLEM*

dengan ini menyatakan bahwa pada Tugas Akhir dengan judul di atas tersebut:

hereby declare that in the Final Project with the above title:

No.	Pernyataan / Statement	(✓)
1	Saya hanya menggunakan AI generatif sebagai alat bantu untuk memperbaiki tata bahasa. AI generatif tidak digunakan untuk membuat isi Tugas Akhir. <i>I only used generative AI as a tool to improve the readability or language of the text in my Final Project. It was not used to generate a complete text of my work.</i>	✓
2	Saya telah memeriksa dan/atau memperbaiki seluruh bagian dari Tugas Akhir saya yang dibantu oleh AI generatif agar sesuai dengan baku mutu penulisan karya ilmiah. <i>I have reviewed and refined all aspects of my work that generative AI assists with, ensuring it adheres to the standards of academic writing.</i>	✓
3	Saya tidak menggunakan AI generatif untuk pembuatan data primer, grafik dan/atau tabel pada Tugas Akhir saya. <i>I did not use generative AI to generate primary data, figures, and/or tables in my work.</i>	✓
4	Saya telah memberikan atribusi/pengakuan terhadap alat AI yang digunakan, secara rinci pada suatu bagian pada lampiran. <i>I have acknowledged the use of generative AI in any part of the work in the specific appendix page.</i>	✓
5	Saya memastikan tidak ada plagiarisme, termasuk hal yang berasal dari penggunaan AI generatif. <i>I have ensured that there is no plagiarism issue in the work, including any parts generated by AI.</i>	✓

Surabaya, Juli 2026

Mahasiswa



Muhammad Rayyaan Fatikhahur Rakhim
NRP. 5025221047

Halaman ini sengaja dikosongkan

ABSTRAK

REDUCED-COST VOGEL METHOD (RVM) BERBASIS REDUKSI ZERO-POINT UNTUK KONSTRUKSI INITIAL BASIC FEASIBLE SOLUTION PADA BALANCED TRANSPORTATION PROBLEM

Nama Mahasiswa / NRP: Muhammad Rayyaan Fatikhahur Rakhim / 5025221047

Departemen : Teknik Informatika - FTEIC - ITS

Dosen Pembimbing : Dr. Bilqis Amaliah, S.Kom., M.Kom.

Dosen Ko-pembimbing : Ilham Gurat Adillion, S.Kom., M.Eng.

Abstrak

Transportation Problem (TP) merupakan permasalahan optimisasi fundamental dalam *supply chain management* yang bertujuan meminimalkan biaya distribusi dengan kendala *supply* dan *demand*. Solusi diperoleh melalui dua tahap, yaitu penentuan *Initial Basic Feasible Solution* (IBFS) menggunakan metode heuristik, kemudian dioptimasi menggunakan *Modified Distribution method*. *Vogel's Approximation Method* (VAM) merupakan metode IBFS klasik terbaik, namun *penalty*-nya dihitung dari matriks biaya asli sehingga prioritas alokasi dapat menjadi kurang tepat. Penelitian ini mengembangkan *Reduced-cost Vogel Method* (RVM), yaitu metode konstruksi IBFS satu-langkah yang menghitung *penalty* Vogel pada matriks hasil reduksi *zero-point*. Reduksi tersebut membentuk pasangan *dual* layak sehingga entri matriks menjadi *reduced cost*, dan *penalty* yang dihitung di atasnya menjadi ukuran *regret* yang berlandaskan teori dualitas, sementara metode tetap deterministik dan berjalan dalam satu lintasan tanpa langkah penyempurnaan. Evaluasi pada 42 kasus *benchmark* menunjukkan RVM mencapai 38 kasus optimal (90,48%) dengan deviasi rata-rata 0,40%, melampaui tujuh metode pembanding pada kondisi pengujian identik, yaitu VAM (52,38%), TOCM-MT (61,90%), BCE (69,05%), JHM-M (73,81%), SSM (73,81%), CSM (78,57%), dan RBSM (80,95%). Studi ablasi terhadap 15 fungsi *penalty* mengonfirmasi bahwa keunggulan utama berasal dari reduksi *zero-point*, bukan bentuk fungsi *penalty*. Pengujian tambahan pada 667 kasus di luar *benchmark* utama beserta metode pembanding kedelapan (*Improved Heuristic Weight*, IHW) menunjukkan bahwa keunggulan RVM berkorelasi dengan kepadatan *twin cost* pada struktur data yang diujikan, sejalan dengan *No Free Lunch Theorem*. Dengan demikian, RVM menyediakan solusi awal berkualitas tinggi dengan biaya komputasi yang rendah dan landasan matematis yang kuat.

Kata Kunci: *Balanced Transportation Problem, Initial Basic Feasible Solution, Vogel's Approximation Method, Zero-Point Reduction, RVM.*

Halaman ini sengaja dikosongkan

ABSTRACT

REDUCED-COST VOGEL METHOD (RVM) BASED ON ZERO-POINT REDUCTION FOR INITIAL BASIC FEASIBLE SOLUTION CONSTRUCTION IN THE BALANCED TRANSPORTATION PROBLEM

Student Name / NRP: Muhammad Rayyaan Fatikhahur Rakhim / 5025221047

Department : Informatics Engineering - ELECTICS - ITS

Supervisor : Dr. Bilqis Amaliah, S.Kom., M.Kom.

Co-supervisor : Ilham Gurat Adillion, S.Kom., M.Eng.

Abstract

The Transportation Problem (TP) is a fundamental optimization problem in *supply chain management* that aims to minimize distribution costs subject to *supply* and *demand* constraints. Solutions are obtained in two stages: determining the Initial Basic Feasible Solution (IBFS) using a heuristic method, then optimizing it using the Modified Distribution method. Vogel's Approximation Method (VAM) is the best classical IBFS method, but its penalty is computed from the original cost matrix, which can make allocation priorities inaccurate. This research develops the Reduced-cost Vogel Method (RVM), a single-pass IBFS construction method that computes the Vogel penalty on a zero-point reduced matrix. The reduction forms a feasible dual pair so that the matrix entries become reduced costs, and the penalty computed on it becomes a regret measure grounded in duality theory, while the method remains deterministic and runs in a single pass without any refinement step. Evaluation on 42 benchmark cases shows that RVM reaches 38 optimal cases (90.48%) with an average deviation of 0.40%, surpassing seven comparison methods under identical testing conditions, namely VAM (52.38%), TOCM-MT (61.90%), BCE (69.05%), JHM-M (73.81%), SSM (73.81%), CSM (78.57%), and RBSM (80.95%). An ablation study of 15 penalty functions confirms that the main advantage comes from the zero-point reduction rather than the penalty form. Additional testing on 667 cases beyond the main benchmark, together with an eighth comparison method (Improved Heuristic Weight, IHW), shows that RVM's advantage correlates with the density of tied costs in the tested data structure, consistent with the No Free Lunch Theorem. RVM therefore provides a high-quality initial solution at a low computational cost with a strong mathematical foundation.

Keywords: *Balanced Transportation Problem, Initial Basic Feasible Solution, Vogel's Approximation Method, Zero-Point Reduction, RVM.*

This page is intentionally left blank

KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat Allah SWT atas rahmat, hidayah, dan karunia-Nya sehingga penulis dapat menyelesaikan laporan Tugas Akhir ini sebagai salah satu syarat memperoleh gelar Sarjana Komputer pada Departemen Teknik Informatika, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember.

Dalam pengerjaan dan penyusunan Tugas Akhir ini, penulis menyadari bahwa hasil yang dicapai tidak lepas dari bimbingan, arahan, motivasi, dukungan, serta bantuan dari berbagai pihak. Oleh karena itu, penulis ingin menyampaikan penghargaan dan ucapan terima kasih yang sebesar-besarnya kepada:

1. Ayah, Ibu, serta keluarga tercinta atas kasih sayang, doa, dan dukungan moral maupun materiil yang tiada henti bagi penulis.
2. Ibu Dr. Bilqis Amaliah, S.Kom., M.Kom. selaku Dosen Pembimbing I dan Bapak Ilham Gurat Adillion, S.Kom., M.Eng. selaku Dosen Pembimbing II atas bimbingan, masukan berharga, dan kesabaran dalam mengarahkan penulis hingga laporan ini terselesaikan.
3. Prof. Dr. Eng. Nanik Suciati, S.Kom., M.Kom. Dr. Eng. Muhamad Hilmil Muchtar Aditya Pradana, S.Kom., M.Sc. Dosen Penguji atas kritik, saran, dan masukan yang membangun demi penyempurnaan Tugas Akhir ini.
4. Bapak Prof.Ir.Ary Mazharuddin Shiddiqi, S.Kom., M.Comp.Sc., Ph.D. selaku Kepala Departemen Teknik Informatika ITS, beserta seluruh dosen, tenaga pendidik, dan staf di dalamnya, atas ilmu, didikan, dan pelayanan akademis selama masa perkuliahan penulis.
5. Ibu Siska Arifiani, S.Kom., M.Kom., Bapak Hadziq Fabroyir, S.Kom., Ph.D., dan Ibu Dini Adni Navastara, S.Kom., M.Sc. atas bimbingan dan dukungannya yang turut membentuk berbagai pencapaian penulis, termasuk keberhasilan dalam sejumlah kompetisi.
6. Bapak Sundar Pichai (CEO Google) dan Bapak Dario Amodei (CEO Anthropic) atas inovasi teknologi kecerdasan buatan dan mesin pencarian yang membantu penulis dalam riset dan penyusunan naskah Tugas Akhir ini.
7. Sahabat, teman, rekan, dan kolega baik di kehidupan sehari-hari, perkuliahan, maupun kantor, atas dukungan, motivasi, dan kebersamaan yang berkesan bagi penulis selama perkuliahan dan penyusunan Tugas Akhir ini.

Akhir kata, penulis menyadari bahwa laporan Tugas Akhir ini masih jauh dari sempurna. Semoga laporan ini dapat memberikan manfaat teoretis maupun praktis, serta menjadi sumbangsih positif bagi perkembangan ilmu pengetahuan.

Surabaya, Juli 2026

Muhammad Rayyaan Fatikhahur Rakhim

Halaman ini sengaja dikosongkan

DAFTAR ISI

LEMBAR PENGESAHAN	i
APPROVAL SHEET	iii
PERNYATAAN ORISINALITAS	v
STATEMENT OF ORIGINALITY	vii
PERNYATAAN KODE ETIK PENGGUNAAN AI GENERATIF	ix
ABSTRAK	xi
ABSTRACT	xiii
KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xxi
DAFTAR TABEL	xxiii
DAFTAR KODE SEMU	xxvii
DAFTAR SINGKATAN	xxix
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan	3
1.4 Batasan Masalah	3
1.5 Manfaat	4
1.5.1 Manfaat Teoritis	4
1.5.2 Manfaat Praktis	4
2 TINJAUAN PUSTAKA	5
2.1 Hasil Penelitian Terdahulu	5
2.1.1 Perkembangan Metode IBFS Berbasis Biaya Peluang	5

2.1.2	Studi Komparatif dan Pengujian Statistik	6
2.1.3	Sintesis dan <i>Research Gap</i>	7
2.1.4	Peta Jalan Penelitian.....	8
2.2	Dasar Teori	9
2.2.1	Transportation Problem	9
2.2.2	Initial Basic Feasible Solution	10
2.2.3	Dualitas dan Reduced Cost	10
2.2.4	Total Opportunity Cost Matrix	11
2.2.5	Reduksi Zero-Point	11
2.2.6	Penalti Vogel sebagai Ukuran Regret	13
2.2.7	Metrik Evaluasi Kualitas IBFS	14
2.2.8	No Free Lunch Theorem	14
3	METODOLOGI	17
3.1	Metode yang Digunakan	17
3.1.1	Notasi dan Inisialisasi	17
3.1.2	Reduksi Zero-Point pada Submatriks Aktif.....	17
3.1.3	Perhitungan Penalty Vogel	18
3.1.4	Pemilihan Baris atau Kolom dan Alokasi	18
3.1.5	<i>Tie-Breaking</i>	19
3.1.6	Kriteria Berhenti dan Penanganan Degenerasi	24
3.1.7	Analisis Kelayakan, Terminasi, dan Kompleksitas	24
3.1.8	Catatan Implementasi	25
3.1.9	Pseudocode	26
3.1.10	Penjelasan Umum Tahapan Algoritma	27
3.1.11	Penyelesaian Lengkap Langkah demi Langkah	28
3.1.11.1	Ilustrasi <i>Tie-Breaking</i>	33
3.1.12	Flowchart Metodologi	33
3.2	Bahan dan Peralatan yang Digunakan.....	35
3.2.1	Dataset	35
3.2.1.1	Format Instance	39
3.2.1.2	Ground Truth Solusi Optimal.....	40
3.2.2	Dataset Uji Tambahan (Dataset Uji 2 dan Dataset Uji 3)	40
3.2.2.1	Dataset Uji 2 dengan Variasi Ukuran Kasus	40
3.2.2.2	Dataset Uji 3 dengan Matriks Sintetis Skala Besar	40

3.2.2.3	Format Data dan Verifikasi	41
3.2.3	Karakteristik Struktur Biaya Dataset Uji 2 dan Dataset Uji 3	42
3.2.4	Metrik Evaluasi	43
3.2.4.1	Accuracy Percentage	43
3.2.4.2	Deviation Percentage	43
3.2.4.3	Percentage of Correctness	44
3.2.4.4	Improvement Percentage	44
3.2.4.5	Frequency of the Number of the Optimal Cost Solutions	44
3.2.4.6	Optimality Iterations	45
3.2.5	Metode Pembandingan Tambahan IHW	45
3.2.6	Desain Studi Ablasi	45
3.2.7	Perangkat Keras	48
3.2.8	Perangkat Lunak	48
3.2.8.1	Compiler dan IDE	48
3.2.8.2	Python Libraries	48
3.2.8.3	C Standard Libraries	48
4	HASIL DAN PEMBAHASAN	51
4.1	Hasil Penelitian	51
4.1.1	Rincian Hasil Biaya Solusi per Kasus	51
4.1.2	Hasil RVM dan Perbandingan Agregat dengan Metode Terpublikasi	52
4.1.3	Matriks Evaluasi Menyeluruh	55
4.1.3.1	Improvement Percentage	56
4.1.3.2	FNOCS menurut Ukuran Matriks	57
4.1.3.3	Optimality Iterations	58
4.1.4	Studi Ablasi Fungsi Penalty	60
4.1.5	Analisis Degenerasi	61
4.2	Pembahasan	61
4.2.1	Analisis Kasus Keunggulan RVM atas RBSM	61
4.2.2	Analisis Faktor Kasus Tidak Optimal	65
4.2.3	Sintesis Komparasi Karakteristik Matriks	70
4.3	Hasil pada Dataset Uji Tambahan (Dataset Uji 2 dan Dataset Uji 3)	71
4.3.1	Akurasi dan Deviasi pada Dataset Uji 2	71
4.3.2	Akurasi dan Deviasi pada Dataset Uji 3	71
4.3.3	Analisis Runtime dan Kompleksitas Komputasi	72

4.3.4	Karakteristik Struktur Biaya dan Performa RVM	73
4.3.5	Verifikasi Silang Performa IHW pada Dataset Uji 1	74
5	KESIMPULAN DAN SARAN	77
5.1	Kesimpulan	77
5.2	Saran	77
	DAFTAR PUSTAKA	79
	DAFTAR LAMPIRAN	83
	LAMPIRAN	83
	Lampiran A: Kode Sumber Inti RVM	83
	Lampiran B: Hasil Lengkap Studi Ablasi	84
	Lampiran C: Matriks Lengkap Kasus Uji	85
	Lampiran D: Ringkasan Dataset Uji Tambahan	103
	BIODATA PENULIS	107

DAFTAR GAMBAR

Gambar 2.1	Peta Jalan Penelitian Metode IBFS pada <i>Transportation Problem</i> . . .	9
Gambar 3.1	Flowchart Metode RVM	34
Gambar 4.1	Perbandingan Akurasi Antarmetode	54
Gambar 4.2	Perbandingan Deviasi Rata-Rata Antarmetode	55
Gambar 4.3	Pertumbuhan Runtime Menurut Jumlah Kolom pada Dataset Uji 3 ..	73
Gambar 4.4	Perbandingan Akurasi RVM dan IHW pada Dataset Uji 1, Dataset Uji 2, dan Dataset Uji 3	75

Halaman ini sengaja dikosongkan

DAFTAR TABEL

Tabel 2.1	Ringkasan Metode IBFS, Keterbatasan, dan Posisi RVM	8
Tabel 3.1	Contoh Ilustrasi Pemilihan Lini dan Alokasi	19
Tabel 3.2	Contoh Ilustrasi <i>tie-breaking</i> oleh MRV (Kasus 4×3)	20
Tabel 3.3	Contoh Ilustrasi <i>tie-breaking</i> oleh Prioritas Baris (Kasus 3×3).....	21
Tabel 3.4	Contoh Ilustrasi <i>tie-breaking</i> oleh Prioritas Indeks (Kasus 3×2)	21
Tabel 3.5	Ilustrasi <i>tie-breaking</i> Sel oleh Alokasi Terbesar	22
Tabel 3.6	Ilustrasi <i>tie-breaking</i> Sel oleh Biaya Asli Terkecil	22
Tabel 3.7	Ilustrasi <i>tie-breaking</i> Sel oleh Prioritas Indeks	23
Tabel 3.8	Justifikasi Matematis Tiap Tahapan RVM.....	26
Tabel 3.9	Kasus n11 (3×4), $Z^* = 2460$	29
Tabel 3.10	Iterasi 1 – Matriks Biaya Asli c_{ij}	29
Tabel 3.11	Iterasi 1 – Matriks Tereduksi $\bar{c}_{ij}$ dan Penalti	29
Tabel 3.12	Iterasi 2 – Matriks Biaya Asli c_{ij}	30
Tabel 3.13	Iterasi 2 – Matriks Tereduksi $\bar{c}_{ij}$ dan Penalti	30
Tabel 3.14	Iterasi 3 – Matriks Biaya Asli c_{ij}	30
Tabel 3.15	Iterasi 3 – Matriks Tereduksi $\bar{c}_{ij}$ dan Penalti	30
Tabel 3.16	Iterasi 4 – Matriks Biaya Asli c_{ij}	31
Tabel 3.17	Iterasi 4 – Matriks Tereduksi $\bar{c}_{ij}$ dan Penalti	31
Tabel 3.18	Iterasi 5 – Matriks Biaya Asli c_{ij}	31
Tabel 3.19	Iterasi 5 – Matriks Tereduksi $\bar{c}_{ij}$ dan Penalti	32
Tabel 3.20	Alokasi Akhir RVM pada Kasus n11	32
Tabel 3.21	Sebaran Ukuran Matriks pada 42 Kasus Uji	36
Tabel 3.22	Sumber, Ukuran, Nilai Optimal, dan Peran Seluruh Kasus Uji	36
Tabel 3.23	Instance n01 (3×3), $Z^* = 5600$	39
Tabel 3.24	Instance n02 (3×4), $Z^* = 28$	39
Tabel 3.25	Ground Truth Solusi Optimal Seluruh Kasus Uji.....	40
Tabel 3.26	Rata-Rata <i>Twin-Cost Ratio</i> per Himpunan Data	42
Tabel 3.27	Definisi 15 Varian Fungsi <i>Penalty</i> yang Diuji	47
Tabel 3.28	Spesifikasi Perangkat Keras	48
Tabel 3.29	Python Libraries	48
Tabel 3.30	C Standard Libraries	49

Tabel 4.1	Biaya Solusi Seluruh Metode pada 42 Kasus Uji.....	51
Tabel 4.2	Hasil RVM pada 42 Kasus Uji.....	52
Tabel 4.3	Perbandingan RVM dengan Metode IBFS Terpublikasi.....	53
Tabel 4.4	Matriks Evaluasi Menyeluruh Antarmetode (AP, DP, PC).....	56
Tabel 4.5	Rincian Selisih Biaya dan Status Per-Kasus RVM terhadap Metode Pembanding	56
Tabel 4.6	Improvement Percentage (IP) RVM terhadap Tiap Metode Pembanding	57
Tabel 4.7	Pengelompokan Ukuran Matriks.....	58
Tabel 4.8	Rincian Jumlah Pivot MODI (OI) Seluruh Metode pada 42 Kasus Uji ..	58
Tabel 4.9	Optimality Iterations (OI) Tiap Metode pada 42 Kasus Uji	59
Tabel 4.10	Hasil Studi Ablasi 15 Fungsi <i>Penalty</i> pada Benchmark 42 Kasus	60
Tabel 4.11	Kasus n03 (5×5), $Z^* = 1102$	62
Tabel 4.12	Matriks Tereduksi Kasus n03 pada Iterasi 1	62
Tabel 4.13	Komparasi Langkah Kunci RVM dan RBSM pada Kasus n03	63
Tabel 4.14	Kasus n16 (3×4), $Z^* = 1510$	64
Tabel 4.15	Matriks Tereduksi Kasus n16 pada Iterasi 1	64
Tabel 4.16	Matriks Tereduksi Kasus n17 pada Iterasi 1	64
Tabel 4.17	Kasus n17 dan n25 (3×4)	65
Tabel 4.18	Karakteristik Kasus yang Tidak Mencapai Optimal	65
Tabel 4.19	Kasus n05 (4×4), $Z^* = 410$	66
Tabel 4.20	Perbandingan Alokasi RVM dan Optimal pada n05	66
Tabel 4.21	Status Alokasi Setelah Iterasi 1 RVM vs RBSM pada Kasus n05	67
Tabel 4.22	Snapshot Matriks Tereduksi Kasus n10 pada Iterasi 2	68
Tabel 4.23	Kasus n10 (3×4), $Z^* = 910$	68
Tabel 4.24	Perbandingan Alokasi RVM dan Optimal pada n10	68
Tabel 4.25	Matriks Tereduksi Kasus n33 pada Iterasi 1	69
Tabel 4.26	Kasus n33 (3×3), $Z^* = 1250$	69
Tabel 4.27	Perbandingan Alokasi RVM dan Optimal pada n33	69
Tabel 4.28	Komparasi Karakteristik Matriks pada Kondisi Optimal vs Tidak Optimal	70
Tabel 4.29	Hasil Sembilan Metode pada Dataset Uji 2 (27 Kasus)	71
Tabel 4.30	Hasil Sembilan Metode pada Dataset Uji 3 (640 Kasus)	72
Tabel 4.31	Rata-Rata Runtime (ms) Menurut Jumlah Kolom pada Dataset Uji 3 ...	72
Tabel 4.32	Twin-Cost Ratio dan AP RVM pada Ketiga Himpunan Data	73
Tabel 4.33	Perbandingan RVM dan IHW pada Ketiga Himpunan Data	74
Tabel B.1	Hasil Studi Ablasi 15 Fungsi Penalti pada 42 Kasus.....	84

Tabel C.1	Instance n03 (5×5), $Z^* = 1102$	85
Tabel C.2	Instance n04 (3×5), $Z^* = 1580$	85
Tabel C.3	Instance n05 (4×4), $Z^* = 410$	85
Tabel C.4	Instance n06 (4×4), $Z^* = 7750$	85
Tabel C.5	Instance n07 (3×4), $Z^* = 1160$	86
Tabel C.6	Instance n09 (4×4), $Z^* = 111$	86
Tabel C.7	Instance n10 (3×4), $Z^* = 910$	86
Tabel C.8	Instance n12 (3×3), $Z^* = 425$	86
Tabel C.9	Instance n13 (3×3), $Z^* = 4525$	86
Tabel C.10	Instance n14 (3×4), $Z^* = 920$	87
Tabel C.11	Instance n15 (3×4), $Z^* = 156$	87
Tabel C.12	Instance n16 (3×4), $Z^* = 1510$	87
Tabel C.13	Instance n17 (3×4), $Z^* = 743$	87
Tabel C.14	Instance n18 (3×4), $Z^* = 1650$	87
Tabel C.15	Instance n21 (3×3), $Z^* = 4205$	88
Tabel C.16	Instance n22 (3×4), $Z^* = 12075$	88
Tabel C.17	Instance n23 (3×4), $Z^* = 23$	88
Tabel C.18	Instance n24 (3×4), $Z^* = 3460$	88
Tabel C.19	Instance n25 (3×4), $Z^* = 809$	88
Tabel C.20	Instance n26 (4×5), $Z^* = 490$	89
Tabel C.21	Instance n27 (5×5), $Z^* = 59356$	89
Tabel C.22	Instance n28 (4×4), $Z^* = 2090$	89
Tabel C.23	Instance n29 (3×4), $Z^* = 3857$	89
Tabel C.24	Instance n30 (3×3), $Z^* = 465$	89
Tabel C.25	Instance n31 (3×4), $Z^* = 1480$	90
Tabel C.26	Instance n32 (4×5), $Z^* = 1780$	90
Tabel C.27	Instance n33 (3×3), $Z^* = 1250$	90
Tabel C.28	Instance n34 (4×5), $Z^* = 510$	90
Tabel C.29	Instance n35 (3×4), $Z^* = 280$	90
Tabel C.30	Instance S1 (5×5), $Z^* = 18855$	91
Tabel C.31	Instance S2 (5×5), $Z^* = 153500$	91
Tabel C.32	Instance S3 (11×12), $Z^* = 1044180$	91
Tabel C.33	Instance S4 (11×12), $Z^* = 92212$	92
Tabel C.34	Instance S5 (11×12), $Z^* = 116020$	92

Tabel C.35	Instance R1 (2×94), $Z^* = 12928429$, Supply $S_1=308252$, $S_2=138801$; ditranspos per kolom tujuan	92
Tabel C.36	Instance R2 (2×94), $Z^* = 13657053$, Supply $S_1=341015$, $S_2=138738$; ditranspos per kolom tujuan	95
Tabel C.37	Instance R3 (2×94), $Z^* = 12296587$, Supply $S_1=280110$, $S_2=138801$; ditranspos per kolom tujuan	97
Tabel C.38	Instance R4 (2×94), $Z^* = 14117586$, Supply $S_1=350910$, $S_2=142165$; ditranspos per kolom tujuan	99
Tabel C.39	Instance R5 (2×94), $Z^* = 14017419$, Supply $S_1=359746$, $S_2=138801$; ditranspos per kolom tujuan	101
Tabel D.1	Ringkasan 27 Kasus Uji Dataset Uji 2	104
Tabel D.2	Ringkasan 64 Kelompok Parameter Dataset Uji 3 (640 Kasus, 10 Rep- likasi per Kelompok)	104

DAFTAR KODE SEMU

Kode Semu 1	Reduksi Zero-Point Sekuensial	26
Kode Semu 2	Perhitungan Penalti Vogel Tereduksi dan Pemilihan Lini	27
Kode Semu 3	Algoritma Utama Reduced-cost Vogel Method (RVM).....	27

Halaman ini sengaja dikosongkan

DAFTAR SINGKATAN

Singkatan	Kepanjangan
AOCM	<i>Average Opportunity Cost Method</i>
AP	<i>Accuracy Percentage</i>
BCE	<i>Bilqis-Chastine-Erma Method</i>
CSM	<i>Cost-Supply Method</i>
DP	<i>Deviation Percentage</i>
DSM	<i>Direct Sum Method</i>
ER	<i>Excess Row</i>
EXP-ICHM	<i>Exponential and Inverse Contra-Harmonic Mean</i>
FA	<i>Feasible Allocation</i>
FLC	<i>First Least Cost</i>
FNOCS	<i>Frequency of the Number of the Optimal Cost Solutions</i>
GVAM	<i>Goyal's Vogel's Approximation Method</i>
IAM	<i>Incessant Allocation Method</i>
IBFS	<i>Initial Basic Feasible Solution</i>
IHW	<i>Improved Heuristic Weight</i>
IP	<i>Improvement Percentage</i>
IVAM	<i>Improved Vogel's Approximation Method</i>
JHM-M	<i>Juman-Hoque Method – Modified</i>
KSAM	<i>Karagül-Aproksimasi Supply-Demand Method</i>
LCM	<i>Least Cost Method</i>
MC	<i>Minimum Cost</i>
MODI	<i>Modified Distribution Method</i>
NWCM	<i>Northwest Corner Method</i>
OI	<i>Optimality Iterations</i>
PC	<i>Percentage of Correctness</i>
RAM	<i>Russell's Approximation Method</i>
RBSM	<i>Rihan-Bilqis-Saikhu Method</i>
RVM	<i>Reduced-cost Vogel Method</i>
SSM	<i>Supply Selection Method</i>
TOCM	<i>Total Opportunity Cost Matrix</i>
TOCM-MT	<i>Total Opportunity Cost Matrix – Minimal Total</i>
TP	<i>Transportation Problem</i>
VAM	<i>Vogel's Approximation Method</i>

Halaman ini sengaja dikosongkan

BAB I

PENDAHULUAN

1.1 Latar Belakang

Transportation Problem (TP) merupakan salah satu model optimisasi fundamental dalam *Operations Research* yang pertama kali dirumuskan oleh Hitchcock pada tahun 1941 dan dikembangkan lebih lanjut oleh Koopmans pada tahun 1949. Permasalahan ini bertujuan meminimalkan total biaya distribusi produk dari sejumlah m sumber (*supply*) ke sejumlah n tujuan (*demand*) dengan memenuhi seluruh batasan kapasitas dan permintaan. Penyelesaian TP umumnya dilakukan melalui dua fase, yaitu memperoleh *Initial Basic Feasible Solution* (IBFS) menggunakan metode heuristik, kemudian melakukan optimisasi iteratif dengan *Modified Distribution* (MODI) *method* hingga mencapai solusi optimal (Dantzig, 1963). Kualitas IBFS sangat menentukan efisiensi fase optimisasi karena solusi awal yang lebih dekat ke optimal memerlukan lebih sedikit iterasi MODI, bahkan pada kondisi tertentu dapat langsung menghasilkan solusi optimal tanpa iterasi tambahan (Kacher dan Singh, 2021).

Berbagai metode IBFS klasik telah dikembangkan sejak tahun 1950-an, termasuk *North-west Corner Method* (NWCM), *Least Cost Method* (LCM), *Vogel's Approximation Method* (VAM), dan *Russell's Approximation Method* (RAM). Di antara metode-metode tersebut, VAM secara konsisten dilaporkan menghasilkan IBFS yang paling mendekati optimal di antara metode klasik (Sultan, 1988; Mathirajan dan Meenakshi, 2004). Keunggulan VAM bersumber dari penggunaan *penalty*, yaitu selisih antara dua biaya terkecil pada setiap baris dan kolom, sebagai ukuran biaya peluang (*opportunity cost*) yang timbul apabila alokasi tidak dilakukan pada sel termurah. Akan tetapi, *penalty* VAM dihitung langsung dari matriks biaya asli, sehingga ketika baris atau kolom memiliki skala biaya yang berbeda-beda, prioritas alokasi dapat menjadi kurang tepat dan menghasilkan solusi yang menyimpang dari optimal (Korukoğlu dan Balli, 2011).

Untuk mengatasi keterbatasan tersebut, sejumlah penelitian mengusulkan transformasi matriks biaya sebelum *penalty* dihitung. Kirca dan Şatir (1990) memperkenalkan *Total Opportunity Cost Matrix* (TOCM) yang mengubah setiap biaya menjadi biaya peluang relatif terhadap biaya minimum baris dan kolomnya. Pandian dan Natarajan (2010) mengembangkan *Zero Point Method* yang mereduksi matriks dengan mengurangi biaya minimum setiap baris lalu setiap kolom, sehingga setiap baris dan kolom memuat sedikitnya satu sel bernilai nol. Transformasi semacam ini berakar pada teori dualitas pemrograman linear, yakni nilai-nilai minimum yang dikurangkan berperan sebagai variabel *dual*, dan entri matriks tereduksi merupakan *reduced cost* yang menyatakan biaya peluang setiap sel relatif terhadap solusi *dual* tersebut (Kuhn, 1955; Bazaraa et al., 2010). Dengan demikian, alokasi dapat diarahkan ke sel yang benar-benar paling menguntungkan secara relatif, bukan sekadar yang termurah secara nominal.

Pengembangan metode IBFS berbasis biaya peluang terus berlanjut secara intensif hingga lima tahun terakhir. Amaliah et al. (2019) mengusulkan *Total Opportunity Cost Matrix – Minimal Total* (TOCM-MT) yang memperhalus aturan pemilihan pada matriks TOCM. Karagül dan Şahin (2020) mengusulkan metode aproksimasi berbobot *supply-demand* (KSAM) yang jauh

lebih cepat daripada VAM, sedangkan Babu et al. (2020) memperbaiki VAM melalui *improved VAM* (IVAM). Pada garis penelitian yang dipelopori Juman dan Hoque (2015), Amaliah et al. (2022a) mengembangkan *Bilqis–Chastine–Erma method* (BCE) dan Amaliah et al. (2022b) mengusulkan *Supply Selection Method* (SSM) yang mencapai akurasi 76,19% pada *benchmark* terstandar; garis ini dilanjutkan oleh Bunyamin et al. (2026b) melalui *Rihan–Bilqis–Saikhu Method* (RBSM) yang mencapai 85,71%. Bunyamin et al. (2026a) juga mengusulkan *Cost-Supply Method* (CSM), yaitu pengembangan SSM yang menambahkan variabel *Cost-Supply* (hasil kali total biaya baris dengan *supply*-nya) untuk mengenali baris berdampak besar, dan mencapai akurasi 78,57% pada *benchmark* 42 kasus yang sama. Wireko et al. (2025) juga mengusulkan *Maximum Range Method* sebagai alternatif terbaru.

Meskipun demikian, studi komprehensif oleh Mathirajan et al. (2022) yang mengevaluasi 34 metode IBFS pada 640 *random instances* mengungkapkan bahwa tidak ada satu pun metode tunggal yang konsisten mencapai solusi optimal di seluruh kasus. Fenomena ini sejalan dengan *No Free Lunch* (NFL) *Theorem* (Wolpert dan Macready, 1997), yang menyatakan bahwa tidak ada algoritma optimisasi yang superior untuk seluruh kelas permasalahan. Implikasinya, banyak metode IBFS yang dilaporkan unggul sesungguhnya hanya unggul pada *benchmark* terbatas milik penelitiannya sendiri dan dapat mengalami penurunan performa pada kasus yang lebih beragam. Oleh karena itu, klaim keunggulan suatu metode IBFS perlu diuji secara empiris pada himpunan kasus yang luas dan diverifikasi terhadap solusi optimal.

Berdasarkan tinjauan tersebut, terdapat peluang penelitian yang belum dieksplorasi. Reduksi *zero-point* terbukti efektif menonjolkan struktur biaya peluang yang berlandaskan dualitas, sementara *penalty* Vogel terbukti efektif sebagai kriteria prioritas alokasi. Namun, sepanjang penelusuran penulis, belum ada metode IBFS yang secara eksplisit menghitung *penalty* Vogel di atas matriks hasil reduksi *zero-point* dalam satu prosedur konstruksi satu-langkah yang deterministik, lengkap dengan landasan matematisnya. Sebagian besar metode terkini, termasuk keluarga BCE–SSM–RBSM, mengandalkan transformasi biaya peluang yang dipadukan dengan aturan alokasi khusus dan langkah penyesuaian *excess unit*, sehingga ruang kombinasi antara reduksi *zero-point* dan *penalty* Vogel murni masih terbuka untuk dieksplorasi.

Penelitian ini mengusulkan *Reduced-cost Vogel Method* (RVM), yaitu metode konstruksi IBFS satu-langkah untuk *balanced transportation problem*. Pada setiap langkah, RVM mereduksi submatriks aktif melalui pengurangan biaya minimum baris dan kolom untuk membentuk *reduced cost* berbasis dualitas, menghitung *penalty* Vogel pada matriks tereduksi tersebut sebagai ukuran *regret*, memilih baris atau kolom dengan *penalty* terbesar, lalu mengalokasikan sebesar mungkin pada sel bernilai nol terkecil. Prosedur ini bersifat deterministik, berjalan dalam satu lintasan tanpa langkah penyempurnaan, dan berhenti setelah paling banyak $m + n - 1$ alokasi. Evaluasi pada 42 kasus uji menunjukkan RVM menghasilkan solusi optimal pada 38 kasus (90,48%) dengan deviasi rata-rata 0,40%, melampaui VAM, TOCM-MT, BCE, JHM-M, SSM, CSM, dan RBSM. Sebagai sebuah heuristik, RVM tidak menjamin optimalitas pada seluruh kasus, namun hasil empiris menunjukkan kombinasi reduksi *zero-point* dan *penalty* Vogel mampu memberikan solusi awal yang berkualitas tinggi dengan biaya komputasi yang rendah. Menindaklanjuti rekomendasi Mathirajan et al. (2022) untuk menguji metode IBFS pada himpunan kasus yang lebih luas, penelitian ini juga melaporkan pengujian tambahan pada 667 kasus di luar *benchmark* utama, serta menyertakan *Improved Heuristic Weight* (IHW) sebagai metode pembandingan kedelapan, untuk memeriksa sejauh mana keunggulan RVM konsisten pada karakteristik struktur biaya yang lebih beragam (Subbab 3.2.2–4.3).

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, pengembangan metode *Initial Basic Feasible Solution* (IBFS) yang efisien dan tangguh masih menjadi tantangan yang terbuka. Oleh karena itu, terdapat beberapa permasalahan utama yang diformulasikan untuk dijawab melalui tahapan-tahapan penelitian ini. Rumusan masalah tersebut adalah sebagai berikut.

1. Bagaimana merancang metode konstruksi IBFS satu-langkah yang mengombinasikan reduksi *zero-point* dengan *penalty* Vogel untuk *balanced transportation problem*, beserta landasan matematis pada setiap tahapannya?
2. Bagaimana performa RVM dibandingkan dengan metode IBFS terpublikasi (VAM, TOCM-MT, BCE, JHM-M, SSM, CSM, dan RBSM) berdasarkan **enam metrik evaluasi performa**, yaitu *Accuracy Percentage* (AP), *Deviation Percentage* (DP), *Percentage of Correctness* (PC), *Improvement Percentage* (IP), *Frequency of the Number of the Optimal Cost Solutions* (FNOCS), dan *Optimality Iterations* (OI)?
3. Bagaimana pengaruh pemilihan fungsi *penalty* terhadap kualitas solusi yang dihasilkan RVM, dan apakah sumber keunggulan berasal dari reduksi *zero-point* atau dari bentuk *penalty*?

1.3 Tujuan

Sejalan dengan rumusan masalah di atas, penelitian ini secara spesifik berfokus pada pengembangan RVM serta pembuktian empiris performanya. Oleh karena itu, sasaran utama yang ingin dicapai melalui pengerjaan tugas akhir ini adalah sebagai berikut.

1. Mengembangkan RVM, metode konstruksi IBFS satu-langkah berbasis reduksi *zero-point* dan *penalty* Vogel, beserta justifikasi matematis setiap tahapannya.
2. Mengevaluasi performa RVM terhadap tujuh metode IBFS terpublikasi secara komprehensif berdasarkan **enam metrik evaluasi performa** (AP, DP, PC, IP, FNOCS, dan OI).
3. Menganalisis pengaruh fungsi *penalty* terhadap kualitas solusi RVM melalui studi ablasi 15 varian penalti.

1.4 Batasan Masalah

Agar penelitian dapat berjalan secara fokus, terarah, dan dapat dievaluasi secara mendalam, diperlukan pembatasan ruang lingkup permasalahan. Batasan ini diterapkan guna meminimalisasi bias pengujian akibat variabel-variabel di luar fokus eksperimen. Adapun batasan masalah pada penelitian ini mencakup hal-hal berikut.

1. *Balanced Transportation Problem* dengan total *supply* sama dengan total *demand* ($\sum a_i = \sum b_j$); kasus tak seimbang berada di luar cakupan penelitian ini.
2. Biaya transportasi berupa bilangan bulat non-negatif.
3. IBFS dihasilkan melalui satu kali konstruksi tanpa langkah penyempurnaan (*refinement*).

4. Verifikasi optimalitas dilakukan terhadap solusi optimal yang diperoleh melalui pemrograman linear.
5. Metode pembandingan adalah metode IBFS yang telah terpublikasi, yaitu VAM, TOCM-MT, BCE, JHM-M, SSM, CSM, dan RBSM.
6. Evaluasi utama dilakukan pada 42 kasus uji *benchmark* (32 kasus literatur, 5 kasus sintetis, 5 *real-world*); pengujian tambahan pada himpunan kasus yang lebih besar dan metode pembandingan kedelapan (IHW) dilaporkan terpisah sebagai pengujian ketahanan (*robustness*) di luar cakupan utama ini (Subbab 3.2.2 dan 4.3).
7. Metrik evaluasi berfokus pada enam metrik standar kualitas IBFS, yaitu *Accuracy Percentage* (AP), *Deviation Percentage* (DP), *Percentage of Correctness* (PC), *Improvement Percentage* (IP), *Frequency of the Number of the Optimal Cost Solutions* (FNOCS), dan *Optimality Iterations* (OI).

1.5 Manfaat

Dari tugas akhir ini, diharapkan dapat dihasilkan kontribusi yang signifikan, baik dari segi pengembangan algoritma maupun aplikasinya pada masalah transportasi. Manfaat yang diperoleh diharapkan mencakup dua aspek utama, yaitu manfaat teoritis dan manfaat praktis, yang dijabarkan sebagai berikut.

1.5.1 Manfaat Teoritis

1. Memberikan kontribusi berupa metode konstruksi IBFS baru yang memadukan reduksi *zero-point* dengan *penalty* Vogel, lengkap dengan landasan dualitas dan *reduced cost*.
2. Memberikan analisis empiris komprehensif mengenai hubungan antara reduksi biaya peluang dan kualitas IBFS berdasarkan enam metrik evaluasi terstandar pada *transportation problem*.

1.5.2 Manfaat Praktis

1. Menyediakan metode deterministik yang sederhana dan cepat untuk memperoleh solusi awal TP yang berkualitas tinggi.
2. Menyediakan solusi awal yang dapat dimanfaatkan sebagai inisialisasi bagi *solver* TP guna mengurangi jumlah iterasi optimisasi.

BAB II

TINJAUAN PUSTAKA

2.1 Hasil Penelitian Terdahulu

2.1.1 Perkembangan Metode IBFS Berbasis Biaya Peluang

Kirca dan Şatir (1990) memperkenalkan *Total Opportunity Cost Matrix* (TOCM) yang dipublikasikan dalam *Journal of the Operational Research Society*. TOCM mentransformasi matriks biaya menjadi penjumlahan biaya peluang baris dan kolom, kemudian menerapkan VAM di atasnya. Pada pengujian terhadap 480 *instance*, TOCM mengungguli VAM pada 372 kasus dan mencapai optimal pada 134 kasus yang gagal dioptimalkan VAM (Mathirajan dan Meenakshi, 2004). Hasil ini menjadi bukti awal bahwa transformasi biaya peluang meningkatkan kualitas IBFS secara substansial.

Pandian dan Natarajan (2010) mengembangkan *Zero Point Method* yang mereduksi matriks biaya dengan mengurangi biaya minimum setiap baris lalu setiap kolom hingga terbentuk sel bernilai nol pada setiap baris dan kolom. Alokasi kemudian diarahkan ke sel bernilai nol. Pendekatan ini berakar pada metode Hungarian (Kuhn, 1955) dan menjadi dasar penting bagi penonjolan struktur biaya relatif melalui reduksi.

Juman dan Hoque (2015) mengusulkan heuristik IBFS yang dipublikasikan dalam *Applied Soft Computing*. Metode ini mengalokasikan permintaan ke biaya terkecil pada kolom, lalu memindahkan kelebihan unit (*excess unit*) dari biaya terkecil pertama ke biaya terkecil kedua. Evaluasi pada kasus literatur menunjukkan 88,9% kasus mencapai optimal. Mekanisme pemindahan *excess unit* ini menjadi dasar bagi keluarga metode yang dikembangkan setelahnya.

Amaliah et al. (2019) mengusulkan *Total Opportunity Cost Matrix – Minimal Total* (TOCM-MT) dalam *Egyptian Informatics Journal*. TOCM-MT memperhalus aturan pemilihan sel pada matriks TOCM dengan mempertimbangkan total minimum baris dan kolom sebagai *tie-breaking*. Metode ini menunjukkan peningkatan akurasi dibanding TOCM dan VAM pada kasus-kasus literatur.

Karagül dan Şahin (2020) mengusulkan metode aproksimasi baru (KSAM) dalam *Journal of King Saud University – Engineering Sciences*. KSAM membobot biaya dengan rasio *supply* dan *demand*. Metode ini unggul pada 17 dari 24 kasus uji dan jauh lebih cepat dari VAM, meskipun deviasi rata-ratanya sedikit lebih tinggi dibanding VAM. Temuan ini menegaskan bahwa keunggulan suatu metode bersifat multidimensi, mencakup akurasi sekaligus efisiensi.

Babu et al. (2020) mengembangkan *Improved VAM* (IVAM) dalam *OPSEARCH*. IVAM memperbaiki kelemahan VAM pada penanganan seri *penalty*. Pada analisis statistik terhadap 1.500 kasus acak dalam lima belas dimensi, IVAM menurunkan total biaya pada 71,8% kasus dibanding VAM, terutama pada kasus berukuran besar.

Amaliah et al. (2022a) mengembangkan *Bilqis–Chastine–Erma method* (BCE) dalam *Journal of King Saud University – Computer and Information Sciences*. BCE menyempurnakan pendekatan pemindahan *excess unit* dengan mempertimbangkan biaya terkecil pertama dan kedua

secara lebih cermat. Pada *benchmark* terstandar, BCE mencapai optimal pada 29 dari 42 kasus (69,05%).

Amaliah et al. (2022b) mengusulkan *Supply Selection Method* (SSM) dalam *Expert Systems with Applications*. SSM memilih *supply* berdasarkan biaya peluang total, lalu mengalokasikannya ke *demand* berbiaya terendah. SSM mencapai optimal pada 32 dari 42 kasus (76,19%) dengan deviasi rata-rata 1,31%, menjadikannya salah satu metode konstruksi terkuat sebelum penelitian ini.

Wireko et al. (2025) mengusulkan *Maximum Range Method* dalam *Results in Control and Optimization*. Metode ini menggunakan rentang maksimum biaya pada baris dan kolom sebagai kriteria prioritas, dan mampu menangani kasus seimbang maupun tak seimbang. Metode ini menunjukkan bahwa eksplorasi bentuk *penalty* alternatif masih aktif dilakukan.

Bunyamin et al. (2026a) mengusulkan *Cost-Supply Method* (CSM) dalam *Engineering, Technology & Applied Science Research*. CSM memperluas SSM dengan memperkenalkan variabel *Cost-Supply* (CS), yaitu hasil kali total biaya baris (TC) dengan *supply* barisnya, untuk mengenali baris berbiaya tinggi sekaligus berkapasitas besar ("*high-impact rows*") yang berisiko mendorong alokasi ke sel mahal apabila ditunda. Pada *benchmark* 42 kasus yang sama dengan penelitian ini, CSM dilaporkan mencapai optimal pada 33 kasus (78,57%) dengan deviasi rata-rata 0,87%, serta secara statistik signifikan lebih baik daripada VAM ($p = 0,015$) namun tidak signifikan dibanding JHM, TOCM-MT, BCE, dan SSM pada uji-t berpasangan publikasinya sendiri.

Bunyamin et al. (2026b) mengusulkan *Rihan–Bilqis–Saikhu Method* (RBSM) dalam *MethodsX*. RBSM memperluas SSM dengan memasukkan total biaya-*supply* setiap baris dan memperhalus aturan realokasi kelebihan *supply*. Pada *benchmark* 42 kasus yang sama dengan penelitian ini, RBSM dilaporkan mencapai optimal pada 36 kasus (85,71%) dengan deviasi rata-rata 0,58% pada publikasinya (sebagai artikel *in-press*, angka ini belum dapat diverifikasi secara independen pada versi terbit), menjadi metode konstruksi terbaik yang dilaporkan sebelum penelitian ini. Namun peningkatan RBSM atas pendahulunya (SSM) tergolong tipis, sehingga ruang perbaikan akurasi konstruksi IBFS masih terbuka.

Boah et al. (2026) memperkenalkan *refinement* VAM berbasis *penalti fraksional* dalam *AppliedMath*, yang mengganti *regret* aditif $s_2 - s_1$ dengan rasio kontras multiplikatif s_2/s_1 antara dua biaya terkecil pada tiap baris dan kolom. Pendekatan ini bersifat *scale-invariant* sehingga lebih peka terhadap perbedaan biaya relatif; pada 34 kasus *benchmark* serta instans besar (hingga 10×20) dilaporkan mencapai rata-rata *gap* optimalitas 2,78% dibanding 5,22% untuk VAM klasik. Namun, sebagaimana VAM, penalti tersebut tetap dihitung pada *matriks biaya asli* tanpa membentuk *reduced cost* yang layak-*dual*.

2.1.2 Studi Komparatif dan Pengujian Statistik

Mathirajan et al. (2022) melakukan studi evaluatif berskala besar dalam *OPSEARCH*, membandingkan 34 metode IBFS pada 640 *random instances* berukuran 3×3 hingga 10×10 . Temuan utamanya adalah tidak ada metode tunggal yang konsisten optimal; metode terbaik menunjukkan *loss of optimality* 0–2% dengan performa yang bervariasi menurut karakteristik *instance*. Studi ini menegaskan pentingnya pengujian pada himpunan kasus yang luas dan beragam.

2.1.3 Sintesis dan *Research Gap*

Berdasarkan tinjauan tersebut, dapat disintesis beberapa hal. Pertama, transformasi biaya peluang, baik melalui TOCM maupun reduksi *zero point*, secara konsisten meningkatkan kualitas IBFS dibanding alokasi pada matriks biaya asli. Kedua, garis penelitian terkini yang paling produktif, yaitu Juman dan Hoque (2015) → BCE (2022a) → SSM (2022b) → RBSM (2026), menyempurnakan *aturan alokasi dan pemindahan excess unit* di atas matriks biaya, bukan menyempurnakan transformasi biayanya. Ketiga, *penalty* Vogel terbukti menjadi kriteria prioritas yang efektif, sementara reduksi *zero point* terbukti efektif menonjolkan struktur biaya relatif yang berlandaskan dualitas.

Dari sintesis ini muncul tiga *gap research* yang konsisten pada metode terdahulu. (i) Garis penelitian alokasi-donasi (JHM, BCE, SSM, RBSM) menghitung kriteria langsung dari *matriks biaya asli*, sehingga perbandingan antarbaris/kolom rentan terhadap perbedaan skala biaya nominal; bahkan *refinement* VAM terbaru yang secara eksplisit mengejar *scale-invariance* (Boah et al., 2026) masih beroperasi pada biaya asli melalui rasio, bukan pada *reduced cost* berbasis dualitas. (ii) Metode yang memakai transformasi biaya peluang, yaitu TOCM dan turunannya, menggunakan bentuk *aditif* yang menjumlahkan dua biaya peluang dari matriks asli sehingga tidak membentuk pasangan *dual* yang layak secara murni. (iii) Hampir seluruh metode bersifat algoritmik tanpa *justifikasi matematis* yang mengaitkan kriteria pemilihannya dengan teori dualitas dan *complementary slackness*.

Gap research yang teridentifikasi adalah belum adanya metode konstruksi IBFS yang menghitung *penalty* Vogel langsung di atas matriks hasil reduksi *zero point* sekuensial dalam satu lintasan deterministik, sekaligus disertai justifikasi matematis melalui teori dualitas. RVM mengisi ketiga *gap research* tersebut sekaligus, yakni bekerja pada *reduced cost* yang netral terhadap skala (mengatasi *gap* i), memakai reduksi sekuensial yang membentuk *dual* layak (mengatasi *gap* ii), dan dilandasi proposisi-proposisi dualitas pada Subbab 2.2.5–2.2.6 (mengatasi *gap* iii). Keterbaruan (*novelty*) RVM dapat dirumuskan dalam empat poin yang membedakannya dari seluruh metode terdahulu. *Pertama*, RVM merupakan satu-satunya metode yang menghitung *penalty* Vogel langsung di atas matriks *reduced cost* hasil reduksi *zero point* sekuensial; VAM menghitung *penalty* pada biaya asli, sedangkan keluarga TOCM menghitungnya pada biaya peluang aditif yang bukan *reduced cost* layak-*dual*. *Kedua*, reduksi yang dipakai bersifat *sekuensial* (baris kemudian kolom) sehingga setiap nilai nol yang dihasilkan adalah sel *tight* yang memenuhi *complementary slackness*, berbeda dari penjumlahan aditif TOCM yang mengabaikan biaya asli dan kehilangan makna dualitas. *Ketiga*, kriteria pemilihannya dilandasi *justifikasi matematis* eksplisit melalui teori dualitas (Subbab 2.2.5–2.2.6), bukan sekadar aturan algoritmik empiris sebagaimana garis JHM–BCE–SSM–RBSM yang menyempurnakan aturan donasi tanpa landasan dualitas. *Keempat*, penanganan serinya berbasis *nilai struktural*, bukan sekadar indeks. Pemilihan lini saat *penalty* seri memakai heuristik *Most Constrained Variable* (MRV) dari *Constraint Satisfaction Problem*, sehingga setiap keputusan RVM dapat dipertanggungjawabkan secara struktural, tidak seperti VAM yang *tie-breaking*nya sembarang dan tidak menambah daya pembeda (Subbab 3.1.5). Kombinasi keempat hal tersebut belum pernah disatukan oleh metode konstruksi IBFS mana pun sebelumnya, dan inilah kontribusi utama penelitian ini.

Tabel 2.1 merangkum posisi RVM terhadap metode-metode utama beserta *gap research* yang melatarinya. Berbeda dari baris-baris lain yang melaporkan akurasi (AP) hasil publikasinya, baris RVM tidak melaporkan AP, sebab pada tahap tinjauan pustaka ini RVM belum diuji, melainkan menjelaskan *bagaimana* RVM dirancang untuk mengatasi ketiga *gap research* di

atas; hasil pengujian empiris RVM disajikan tersendiri pada Bab IV.

Tabel 2.1 Ringkasan Metode IBFS, Keterbatasan, dan Posisi RVM

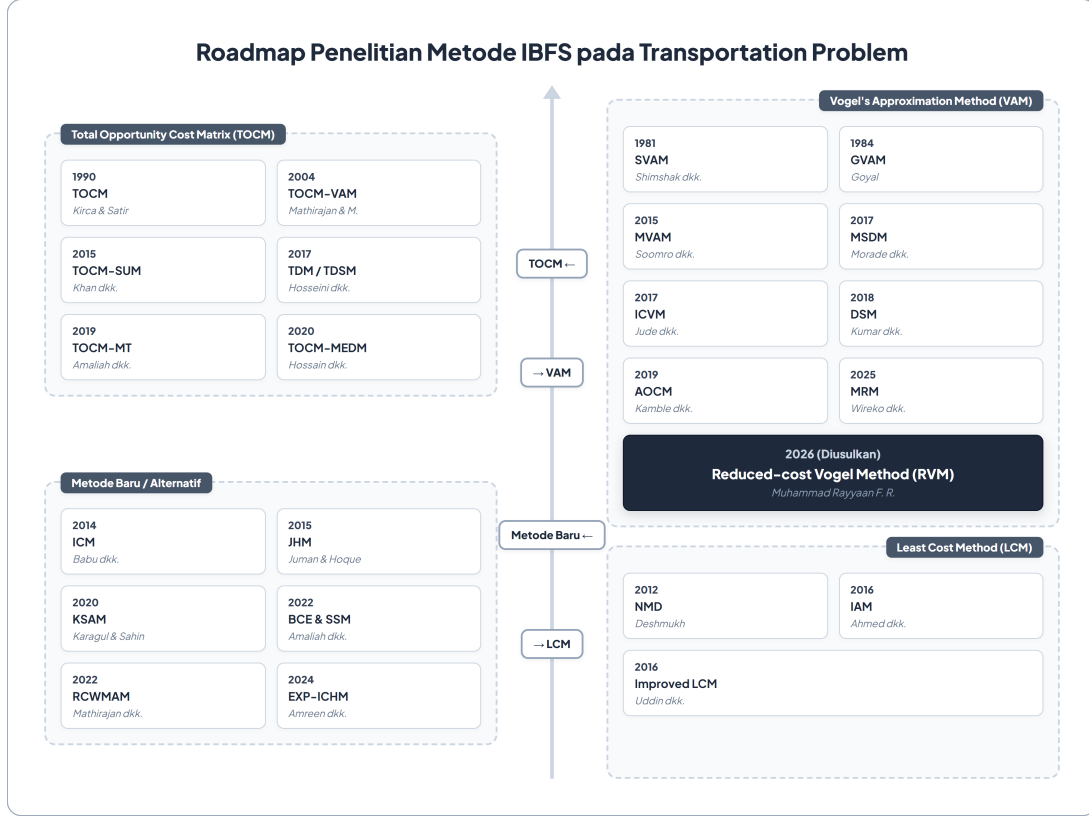
Metode	Tahun	Ide inti	Gap Research	AP (%)
VAM	1958	<i>Penalty</i> pada matriks biaya asli	Perbandingan biaya nominal; rentan salah prioritas saat skala antarbaris berbeda	52,38
TOCM	1990	Biaya peluang aditif baris + kolom	Penjumlahan ganda c_{ij} ; tidak membentuk <i>dual</i> layak	—
JHM-M	2015	Pemindahan <i>excess unit</i> FLC ke SLC	Bekerja pada biaya asli; aturan pemindahan kompleks	64,29*
TOCM-MT	2019	TOCM + <i>tie-breaking</i> minimal total	Transformasi aditif; tanpa landasan dualitas	61,90
KSAM	2020	Pembobotan biaya oleh rasio <i>supply/demand</i>	Heuristik tanpa landasan dualitas	—
IVAM	2020	Perbaikan penanganan seri VAM	Tetap pada matriks biaya asli	—
BCE	2022	Penyempurnaan aturan donasi	Pada biaya asli; tanpa transformasi reduksi	69,05
SSM	2022	Seleksi <i>supply</i> berbasis biaya peluang	Transformasi tak berbasis reduksi <i>zero-point</i>	76,19
CSM	2026	SSM + variabel <i>Cost-Supply</i> ($TC \times supply$)	Menyempurnakan seleksi baris berdampak besar; tanpa justifikasi dualitas	78,57
RBSM	2026	SSM + total biaya- <i>supply</i>	Menyempurnakan seleksi, bukan transformasi; tanpa justifikasi dualitas	85,71
RVM	2026	<i>Penalty</i> Vogel pada matriks reduksi <i>zero-point</i> sekuensial	Mengatasi ketiga gap: (i) <i>reduced cost</i> netral skala; (ii) reduksi sekuensial <i>dual</i> layak; (iii) justifikasi matematis eksplisit	Bab IV

Tanda “—” menunjukkan metode tidak diuji pada *benchmark* 42 kasus yang sama sehingga akurasi tidak diperbandingkan langsung. Tanda “*” pada JHM-M menunjukkan angka AP yang dilaporkan publikasi aslinya (Juman dan Hoque, 2015) pada himpunan kasus literatur miliknya, bukan hasil reproduksi pada 42 kasus *benchmark* penelitian ini; reproduksi JHM-M pada *benchmark* yang sama dilaporkan terpisah pada Tabel 4.3 Bab IV (73,81%), sehingga kedua angka tidak boleh disamakan secara langsung.

2.1.4 Peta Jalan Penelitian

Gambar 2.1 memetakan perkembangan metode konstruksi IBFS pada TP ke dalam empat garis keturunan berdasarkan karakteristik transformasi biayanya. Garis pertama, kelompok VAM, berbasis penalti selisih biaya terkecil dan mencakup VAM klasik (1958), SVAM (1981), GVAM (1984), MVAM (2015), DSM (2018), AOCM (2019), hingga MRM (2025); RVM yang diusulkan penelitian ini bertumpu sebagai pengembangan terbaru pada garis ini, memindahkan perhitungan penalti ke ruang *reduced cost* hasil reduksi *zero-point* sekuensial. Garis kedua, kelompok LCM, berbasis pencarian biaya minimum langsung, meliputi NMD (2012), IAM (2016), dan *Improved LCM* (2016). Garis ketiga, kelompok TOCM, mentransformasi biaya menjadi biaya peluang aditif, dipelopori TOCM (1990) dan diteruskan TOCM-VAM (2004),

TOCM-MT (2019), serta TOCM-MEDM (2020). Garis keempat mencakup metode dengan kriteria alokasi non-klasik yang tidak sepenuhnya mengikuti pola tiga garis di atas, seperti ICM (2014), JHM (2015), KSAM (2020), BCE dan SSM (2022), RCWMAM (2022), serta EXP-ICHM (2024).



Gambar 2.1 Peta Jalan Penelitian Metode IBFS pada *Transportation Problem*

2.2 Dasar Teori

2.2.1 Transportation Problem

Transportation Problem (TP) merupakan kasus khusus pemrograman linear yang dirumuskan oleh Hitchcock (1941) dan dikembangkan Koopmans (1949). Diberikan m sumber dan n tujuan, formulasinya adalah

$$\text{Minimasi } Z = \sum_{i=1}^m \sum_{j=1}^n c_{ij}x_{ij} \quad (2.1)$$

dengan kendala

$$\sum_{j=1}^n x_{ij} = a_i, \quad i = 1, \dots, m, \quad (2.2)$$

$$\sum_{i=1}^m x_{ij} = b_j, \quad j = 1, \dots, n, \quad (2.3)$$

$$x_{ij} \geq 0, \quad \forall i, j, \quad (2.4)$$

dengan c_{ij} biaya per unit, x_{ij} jumlah unit dari sumber i ke tujuan j , a_i kapasitas *supply*, dan b_j permintaan *demand*. TP disebut *balanced* apabila $\sum_i a_i = \sum_j b_j$, dan solusi basisnya memiliki tepat $m + n - 1$ variabel basis (Dantzig, 1963).

2.2.2 Initial Basic Feasible Solution

Initial Basic Feasible Solution (IBFS) adalah solusi awal yang memenuhi seluruh kendala dengan paling banyak $m + n - 1$ alokasi basis dan menjadi titik awal prosedur optimisasi lanjutan. Metode klasik untuk memperolehnya meliputi *Northwest Corner Method* (mengabaikan biaya), *Least Cost Method* (memilih biaya terkecil), serta *Vogel's Approximation Method* (VAM) yang menggunakan *penalty* sebagai biaya peluang (Reinfeld dan Vogel, 1958) dan umumnya menghasilkan IBFS terbaik di antara metode klasik (Sultan, 1988). *Russell's Approximation Method* (Russell, 1969) menggunakan ukuran relatif terhadap biaya maksimum baris dan kolom.

Secara struktural, karena $\sum_i a_i = \sum_j b_j$, matriks kendala TP memiliki *rank* $m + n - 1$, sehingga solusi basis layak memuat tepat $m + n - 1$ variabel basis. Himpunan sel basis membentuk *spanning tree* pada graf bipartit sumber–tujuan, yaitu tidak memuat *loop* (siklus tertutup), yang menjamin keterhubungan sekaligus ketakhadiran redundansi. Apabila sebuah alokasi menghabiskan *supply* dan *demand* secara bersamaan, jumlah sel basis positif menjadi kurang dari $m + n - 1$ dan solusi disebut *degenerate*; kondisi ini tidak memengaruhi kelayakan, namun memerlukan penambahan sel basis bernilai nol bila dilanjutkan dengan prosedur optimisasi (Taha, 2017). Kualitas IBFS diukur dari kedekatan biayanya terhadap biaya optimal, karena IBFS yang lebih baik mempersingkat fase optimisasi.

2.2.3 Dualitas dan Reduced Cost

Landasan matematis RVM terletak pada teori dualitas TP. Secara intuitif, dualitas dapat dipahami sebagai upaya memberi *shadow price* pada setiap sumber dan tujuan: u_i adalah *shadow price* yang melekat pada sumber i dan v_j *shadow price* yang melekat pada tujuan j , sedemikian rupa sehingga jumlah kedua nilai itu tidak pernah melebihi biaya pengiriman riil c_{ij} pada rute mana pun. Selisih antara biaya riil dan jumlah dua *shadow price* itulah yang kemudian disebut *reduced cost* pada Persamaan (2.6). Setiap kendala *supply* dan *demand* berasosiasi dengan variabel *dual* u_i dan v_j ini. Permasalahan *dual* dari TP adalah

$$\text{Maksimasi } \sum_{i=1}^m u_i a_i + \sum_{j=1}^n v_j b_j \quad \text{dengan kendala} \quad u_i + v_j \leq c_{ij}, \quad \forall i, j. \quad (2.5)$$

Untuk pasangan *dual* layak (u, v) yang memenuhi $u_i + v_j \leq c_{ij}$, didefinisikan *reduced cost*

$$\bar{c}_{ij} = c_{ij} - u_i - v_j \geq 0, \quad (2.6)$$

yang menyatakan biaya peluang sel (i, j) relatif terhadap *dual* tersebut. Hubungan antara primal dan *dual* ditegakkan oleh dua teorema fundamental. *Teorema dualitas lemah* menyatakan bahwa untuk sembarang solusi primal layak x dan *dual* layak (u, v) berlaku

$$\sum_i u_i a_i + \sum_j v_j b_j \leq \sum_{i,j} c_{ij} x_{ij}, \quad (2.7)$$

sehingga nilai *dual* layak apa pun merupakan batas bawah biaya optimal. *Teorema dualitas kuat* menjamin bahwa pada solusi optimal kedua nilai tersebut sama. Konsekuensinya, *teorema*

complementary slackness menyatakan bahwa solusi layak x bersifat optimal jika dan hanya jika

$$x_{ij} > 0 \implies \bar{c}_{ij} = c_{ij} - u_i - v_j = 0, \quad \forall i, j, \quad (2.8)$$

yaitu seluruh alokasi positif terjadi pada sel *tight* dengan *reduced cost* nol (Bazaraa et al., 2010). Persamaan (2.8) merupakan dasar uji optimalitas sekaligus alasan teoretis mengapa alokasi sebaiknya diarahkan ke sel dengan *reduced cost* bernilai nol atau mendekati nol, yakni sel demikian adalah satu-satunya yang boleh aktif pada solusi optimal. Reduksi *zero-point* pada RVM justru membangun pasangan (u, v) layak sehingga sel *tight* dapat dikenali sejak awal konstruksi.

2.2.4 Total Opportunity Cost Matrix

TOCM diuraikan secara khusus pada subbab ini karena menjadi pembanding utama bagi reduksi *zero-point* RVM pada Subbab 2.2.5: kedua pendekatan sama-sama mengubah biaya asli menjadi biaya peluang sebelum *penalty* dihitung, namun berbeda secara fundamental dalam *cara* dan *urutan* transformasinya, sebagaimana ditunjukkan pada Subbab 2.2.5 bahwa kedua transformasi menghasilkan matriks yang tidak ekuivalen meski tampak serupa. TOCM tetap relevan dibahas meskipun bukan dasar langsung RVM, sebab garis penelitian TOCM–TOCM–MT mewakili *gap research* (ii) pada Subbab 2.1 yang ingin diatasi RVM.

Total Opportunity Cost Matrix (TOCM) dari Kirca dan Şatir (1990) membentuk biaya peluang dengan menjumlahkan dua komponen yang dihitung *serentak* dari matriks asli,

$$c_{ij}^{\text{TOCM}} = \left(c_{ij} - \min_k c_{ik} \right) + \left(c_{ij} - \min_l c_{lj} \right). \quad (2.9)$$

Kedua komponen, yaitu biaya peluang baris dan biaya peluang kolom, sama-sama dihitung dari matriks biaya asli lalu dijumlahkan.

2.2.5 Reduksi Zero-Point

Reduksi *zero-point* merupakan inti transformasi pada RVM dan berakar pada metode Hungarian (Kuhn, 1955) serta *Zero Point Method* (Pandian dan Natarajan, 2010). Berbeda dengan TOCM yang menjumlahkan dua biaya peluang secara *serentak*, reduksi *zero-point* dilakukan *sekuensial* dalam dua tahap yang masing-masing memiliki makna dualitas.

Tahap 1 (reduksi baris). Untuk setiap baris dikurangkan biaya minimumnya,

$$c_{ij}^{(1)} = c_{ij} - u_i, \quad u_i = \min_j c_{ij}. \quad (2.10)$$

Pemilihan u_i sebagai minimum baris bukanlah pilihan sembarang: u_i adalah nilai *dual* baris terbesar yang masih menjaga seluruh entri baris tetap non-negatif, sehingga sel termurah pada baris itu memperoleh *reduced cost* nol. Dengan kata lain, tahap ini menyatakan setiap biaya sebagai biaya peluang *relatif terhadap pilihan terbaik dalam barisnya*.

Tahap 2 (reduksi kolom). Dari matriks hasil tahap pertama, dikurangkan biaya minimum setiap kolom,

$$\bar{c}_{ij} = c_{ij}^{(1)} - v_j, \quad v_j = \min_i c_{ij}^{(1)}. \quad (2.11)$$

Tahap ini melengkapi pasangan (u, v) menjadi *dual* layak karena menjamin $u_i + v_j \leq c_{ij}$ untuk seluruh sel. Akibatnya, matriks tereduksi $\bar{C} = [\bar{c}_{ij}]$ tepat berisi *reduced cost* pada Persamaan

(2.6), dengan sel bernilai nol pada setiap baris dan kolom. *Matriks baru ini bukanlah permasalahan yang berbeda*, melainkan permasalahan yang sama yang dinyatakan ulang dalam ruang biaya peluang; sel bernilai nol adalah sel *tight* yang menurut *complementary slackness* merupakan kandidat alokasi optimal. Sifat-sifat berikut menjamin validitas transformasi tersebut.

Dual feasibility (kelayakan dual) berarti pasangan nilai (u, v) tidak melanggar kendala $u_i + v_j \leq c_{ij}$ pada sel mana pun (Persamaan (2.5)); pasangan yang demikian disebut *dual* layak. Proposisi berikut menunjukkan bahwa (u, v) hasil reduksi *zero-point* selalu memenuhi syarat ini.

Proposisi 2.1 (Kelayakan dual hasil reduksi). *Pasangan (u, v) dengan $u_i = \min_j c_{ij}$ dan $v_j = \min_i (c_{ij} - u_i)$ merupakan solusi dual layak, yaitu $u_i + v_j \leq c_{ij}$ untuk setiap i, j , dan entri matriks tereduksi memenuhi $\bar{c}_{ij} = c_{ij} - u_i - v_j \geq 0$.*

Bukti. Menurut definisi, $v_j = \min_i (c_{ij} - u_i) \leq c_{ij} - u_i$ untuk setiap i , sehingga $u_i + v_j \leq c_{ij}$. Akibatnya $\bar{c}_{ij} = c_{ij} - u_i - v_j \geq 0$ untuk seluruh sel, sehingga (u, v) memenuhi kendala *dual* pada Persamaan (2.5). $\square$

Proposisi 2.1 menegaskan bahwa reduksi tidak sekadar menghasilkan angka non-negatif, melainkan membangun batas bawah *dual* yang sah bagi biaya optimal (lihat Persamaan (2.7)). Sebagai ilustrasi numerik, tinjau submatriks S_1, S_2 terhadap D_1, D_2 dari kasus n11, yaitu $\begin{pmatrix} 10 & 2 \\ 12 & 7 \end{pmatrix}$.

Reduksi baris ($u = (2, 7)$) memberikan $\begin{pmatrix} 8 & 0 \\ 5 & 0 \end{pmatrix}$, lalu reduksi kolom ($v = (5, 0)$) memberikan $\bar{C} = \begin{pmatrix} 3 & 0 \\ 0 & 0 \end{pmatrix}$. Sel (S_2, D_1) dan kedua sel kolom D_2 menjadi *tight*, dan dapat diperiksa bahwa $u_i + v_j \leq c_{ij}$ pada setiap sel, misalnya $u_1 + v_1 = 7 \leq 10$ dan $u_2 + v_1 = 12 = c_{21}$.

Preservation (pelestarian) di sini berarti alokasi x mana pun yang *paling optimal* pada matriks biaya asli akan tetap menjadi alokasi yang *paling optimal* pula pada matriks tereduksi, dengan hanya nilai total biayanya yang bergeser oleh suatu konstanta, bukan urutan baik-buruknya antarlokasi. Proposisi berikut membuktikan klaim ini secara formal.

Proposisi 2.2 (Pelestarian himpunan solusi optimal). *Pada balanced TP, mengurangi konstanta u_i dari setiap baris dan v_j dari setiap kolom matriks biaya tidak mengubah himpunan solusi optimal; nilai fungsi objektif hanya bergeser sebesar konstanta $\sum_i u_i a_i + \sum_j v_j b_j$.*

Bukti. Untuk sembarang solusi layak x berlaku $\sum_j x_{ij} = a_i$ dan $\sum_i x_{ij} = b_j$, sehingga

$$\sum_{i,j} \bar{c}_{ij} x_{ij} = \sum_{i,j} (c_{ij} - u_i - v_j) x_{ij} = Z(x) - \sum_i u_i a_i - \sum_j v_j b_j.$$

Karena suku $\sum_i u_i a_i + \sum_j v_j b_j$ konstan dan tidak bergantung pada x , fungsi objektif tereduksi hanya berbeda dari $Z(x)$ oleh konstanta, sehingga himpunan pemberi nilai minimum keduanya identik. $\square$

Proposisi 2.2 menjelaskan *mengapa boleh* bekerja pada matriks tereduksi, yakni optimum permasalahan tidak berubah. Proposisi ini berlaku untuk solusi optimal eksak; RVM adalah heuristik konstruktif yang *tidak* mewarisi jaminan optimalitas tersebut, sehingga reduksi hanya menyediakan struktur pemandu, sedangkan optimalitas hasil tetap persoalan empiris.

Proposisi 2.3 (Reduksi cukup satu putaran). *Setelah reduksi baris diikuti reduksi kolom, setiap baris dan setiap kolom matriks tereduksi memuat sedikitnya satu entri nol, dan putaran reduksi berikutnya tidak mengubah matriks.*

Bukti. Setelah reduksi baris, setiap baris i memuat entri nol pada kolom minimumnya k_i , yaitu $c_{ik_i}^{(1)} = 0$, dan semua entri non-negatif. Reduksi kolom mengurangi $v_j = \min_i c_{ij}^{(1)} \geq 0$, sehingga setiap kolom memperoleh entri nol. Untuk entri nol pada (i, k_i) , karena $c_{ik_i}^{(1)} = 0$ dan semua entri non-negatif, maka $v_{k_i} = 0$; akibatnya kolom k_i tidak tergeser dan entri tersebut tetap nol. Dengan demikian setiap baris tetap memuat nol sementara setiap kolom memperoleh nol. Karena setiap baris dan kolom telah memuat nol, minimum baris dan kolom pada putaran berikutnya bernilai nol, sehingga matriks tidak berubah. $\square$

Proposisi 2.3 menjelaskan *mengapa cukup satu putaran* (baris lalu kolom) tanpa iterasi tambahan. Perbedaan dengan TOCM bersifat fundamental. Pada TOCM, kedua biaya peluang dihitung serentak dari matriks asli, sedangkan pada reduksi *zero-point* biaya peluang kolom dihitung dari matriks yang *telah* direduksi baris. Sebagai ilustrasi, untuk matriks $\begin{pmatrix} 4 & 6 \\ 7 & 5 \end{pmatrix}$, Persamaan (2.9) menghasilkan $\begin{pmatrix} 0 & 3 \\ 5 & 0 \end{pmatrix}$, sedangkan reduksi *zero-point* menghasilkan $\begin{pmatrix} 0 & 2 \\ 2 & 0 \end{pmatrix}$. Kedua transformasi karenanya tidak ekuivalen.

2.2.6 Penalti Vogel sebagai Ukuran Regret

Setelah matriks tereduksi terbentuk, RVM menggunakan *penalty* Vogel sebagai kriteria prioritas. Untuk setiap baris atau kolom, dengan $s_1 \leq s_2$ dua nilai *reduced cost* terkecil pada lini tersebut, *penalty* didefinisikan

$$p = s_2 - s_1. \quad (2.12)$$

Secara konseptual, *penalty* adalah ukuran *regret*: semakin besar selisih antara pilihan terbaik dan pilihan terbaik kedua pada suatu lini, semakin besar pula “kerugian” yang akan ditanggung apabila pilihan terbaik itu sampai tidak terambil. Sebagai gambaran, bila suatu baris memiliki dua biaya tereduksi terkecil 0 dan 8, menunda alokasi pada baris itu berisiko kehilangan opsi bernilai 0 dan terpaksa berpindah ke opsi bernilai 8. *Regret* sebesar 8 inilah yang dikuantifikasi oleh *penalty*. Lini dengan *regret* besar karena itu perlu dilayani lebih dahulu, sebelum opsi termurahnya hilang diambil lini lain (Reinfeld dan Vogel, 1958; Russell, 1969). Pada matriks hasil reduksi *zero-point*, setiap lini menurut Proposisi 2.3 memuat entri nol, sehingga $s_1 = 0$ dan

$$p = s_2 = \text{reduced cost terkecil kedua pada lini tersebut.} \quad (2.13)$$

Interpretasinya menjadi tajam, yakni *penalty* pada lini adalah biaya peluang marginal apabila sel *tight* (ber-*reduced cost* nol) pada lini itu tidak segera dialokasikan. Mengalokasikan terlebih dahulu pada lini dengan *penalty* terbesar berarti meminimalkan *regret* yang terakumulasi, sehingga alokasi cenderung tetap berada pada sel dengan *reduced cost* nol, yaitu sel yang menurut *complementary slackness* dipilih oleh solusi optimal. Inilah justifikasi matematis mengapa *penalty* berbentuk selisih dua terkecil dipilih, dan mengapa *penalty* tersebut diterapkan di atas matriks tereduksi, bukan matriks biaya asli. Pembuktian empiris atas pilihan ini diberikan melalui studi ablasi pada Bab IV.

2.2.7 Metrik Evaluasi Kualitas IBFS

Klaim keunggulan suatu metode IBFS hanya bermakna apabila diukur dengan metrik yang konsisten dipakai lintas penelitian, sehingga hasilnya dapat dibandingkan secara adil. Penelitian ini mengadopsi enam metrik yang masing-masing menjawab pertanyaan berbeda tentang kualitas IBFS: *Accuracy Percentage* (AP), *Deviation Percentage* (DP), *Percentage of Correctness* (PC), *Improvement Percentage* (IP), *Frequency of the Number of Optimal Cost Solutions* (FNOCS), dan *Optimality Iterations* (OI).

Amaliah et al. (2019) memperkenalkan AP sebagai ukuran frekuensi keberhasilan mutlak, yaitu persentase kasus yang IBFS-nya identik dengan solusi optimal hasil pemrograman linear. Tinjauan pada Subbab 2.1 memperlihatkan bahwa garis penelitian JHM–BCE–SSM–RBSM secara konsisten melaporkan AP sebagai metrik utama; Amaliah et al. (2022a) melaporkan BCE optimal pada 29 dari 42 kasus (69,05%), Amaliah et al. (2022b) melaporkan SSM optimal pada 32 dari 42 kasus (76,19%), Bunyamin et al. (2026a) melaporkan CSM optimal pada 33 dari 42 kasus (78,57%), dan Bunyamin et al. (2026b) melaporkan RBSM optimal pada 36 dari 42 kasus (85,71%).

AP saja tidak cukup untuk menilai kualitas solusi pada kasus yang *tidak* mencapai optimal, sebab AP memberi bobot yang sama pada kegagalan kecil dan kegagalan besar. Juman dan Hoque (2015) merumuskan DP sebagai jarak relatif biaya IBFS terhadap biaya optimal, melengkapi AP dengan menangkap *seberapa jauh* suatu metode menyimpang ketika gagal optimal, prinsip yang sejalan dengan ukuran *loss of optimality* pada studi banding 34 metode oleh Mathirajan et al. (2022). Wireko et al. (2025) selanjutnya mengusulkan PC, didefinisikan $PC = 100\% - DP$, yang menyajikan DP dalam skala kedekatan terhadap optimal searah dengan AP (semakin tinggi semakin baik), memudahkan pembacaan tabel komparatif yang memuat kedua metrik sekaligus.

Ketiga metrik di atas bersifat agregat terhadap seluruh kasus, sehingga belum menunjukkan seberapa besar kontribusi suatu metode baru relatif terhadap *masing-masing* metode pembandingan secara terpisah. Amaliah et al. (2022a, 2022b) mengisi kebutuhan ini melalui IP, yaitu persentase penurunan biaya IBFS suatu metode dibandingkan satu metode pembandingan tertentu per kasus; metrik sejenis juga dipakai Bunyamin et al. (2026a) untuk menunjukkan bahwa CSM mengungguli VAM pada 15 dari 42 kasus (36%). Babu et al. (2020) mengusulkan FNOCS untuk merinci jumlah kasus optimal menurut ukuran matriks, bukan hanya agregat keseluruhan, karena metode yang unggul secara agregat dapat saja lemah pada subset ukuran tertentu.

Kelima metrik di atas menilai kualitas IBFS sesaat setelah konstruksi, tanpa memperhitungkan manfaatnya bagi fase optimisasi lanjutan. Penelitian ini melengkapinya dengan OI, yaitu banyaknya iterasi prosedur *stepping stone/Modified Distribution* (MODI) yang diperlukan untuk membawa suatu IBFS ke solusi optimal (Charnes & Cooper, 1954; Dantzig, 1963; Taha, 2017). Semakin kecil OI suatu metode, semakin dekat IBFS-nya terhadap struktur optimal, sehingga OI menjadi ukuran tidak langsung atas manfaat praktis IBFS sebagai titik awal optimisasi, yang relevan khususnya bagi RVM yang salah satu motivasinya adalah mempersingkat fase optimisasi lanjutan (Subbab 1.1). Keenam metrik ini membentuk protokol evaluasi yang dipakai konsisten pada Bab III dan Bab IV.

2.2.8 No Free Lunch Theorem

Wolpert dan Macready (1997) membuktikan *No Free Lunch* (NFL) *Theorem*: dirata-ratakan pada seluruh kemungkinan permasalahan, performa semua algoritma optimisasi identik. Imp-

likasinya untuk IBFS, tidak ada metode tunggal yang optimal untuk seluruh karakteristik kasus, sehingga klaim keunggulan perlu diuji pada himpunan kasus yang cukup banyak dan diverifikasi terhadap solusi optimal.

NFL *Theorem* relevan langsung bagi penelitian ini sebagai pengingat metodologis, bukan sekadar teori abstrak: ia menjadi alasan mengapa RVM diuji pada 42 kasus yang beragam (literatur, sintetik, *real-world*, Subbab 3.2) bukan hanya kasus yang menguntungkan RVM, mengapa klaim keunggulan RVM pada Bab IV selalu disandingkan dengan enam metrik yang saling melengkapi (Subbab 3.2.4) ketimbang satu angka tunggal, dan mengapa kasus-kasus tempat RVM tidak mencapai optimal tetap dianalisis secara eksplisit (Subbab 4.2.2) ketimbang disembunyikan demi narasi keunggulan universal.

Halaman ini sengaja dikosongkan

BAB III

METODOLOGI

3.1 Metode yang Digunakan

Penelitian ini mengembangkan *Reduced-cost Vogel Method* (RVM), yaitu metode konstruksi IBFS satu-langkah untuk *balanced transportation problem*. RVM merupakan pengembangan dari VAM dengan satu modifikasi inti, yaitu *penalty* Vogel tidak dihitung dari matriks biaya asli melainkan dari matriks hasil reduksi *zero-point*. Reduksi tersebut menonjolkan struktur biaya peluang sehingga prioritas alokasi menjadi lebih tepat. Metode ini bersifat deterministik, berjalan dalam satu lintasan, dan tidak menggunakan langkah penyempurnaan apa pun setelah solusi layak terbentuk.

3.1.1 Notasi dan Inisialisasi

Diberikan *balanced* TP dengan matriks biaya $C = [c_{ij}]$ berukuran $m \times n$, vektor *supply* $a = [a_i]$, dan vektor *demand* $b = [b_j]$ dengan $\sum_i a_i = \sum_j b_j$. RVM memelihara himpunan baris aktif $I \subseteq \{1, \dots, m\}$ dan himpunan kolom aktif $J \subseteq \{1, \dots, n\}$, yang pada inisialisasi memuat seluruh baris dan kolom. Istilah *line* (disingkat *lini* pada pembahasan selanjutnya) dipakai sebagai sebutan umum untuk satu baris atau satu kolom, tanpa membedakan keduanya, karena banyak aturan RVM (misalnya perhitungan *penalty* dan pemilihan *tie-breaking*) berlaku identik pada baris maupun kolom. *Submatriks aktif* adalah submatriks C yang dibatasi pada baris I dan kolom J . Nilai *supply* dan *demand* sisa diperbarui setiap kali alokasi dilakukan, dan sebuah baris atau kolom dinonaktifkan ketika sisa kapasitasnya habis. Seluruh alokasi x_{ij} diinisialisasi nol.

3.1.2 Reduksi Zero-Point pada Submatriks Aktif

Pada setiap langkah, submatriks aktif direduksi dalam dua tahap. Tahap pertama mengurangi biaya minimum setiap baris aktif,

$$c_{ij}^{(1)} = c_{ij} - u_i, \quad u_i = \min_{j \in J} c_{ij}, \quad i \in I, \quad (3.1)$$

dan tahap kedua mengurangi biaya minimum setiap kolom aktif dari hasil tahap pertama,

$$\bar{c}_{ij} = c_{ij}^{(1)} - v_j, \quad v_j = \min_{i \in I} c_{ij}^{(1)}, \quad j \in J. \quad (3.2)$$

Kedua tahap ini memiliki makna dualitas yang menjelaskan *mengapa* urutannya demikian (Subbab 2.2.3). Pengurangan minimum baris menetapkan u_i sebagai nilai *dual* baris terbesar yang menjaga seluruh entri tetap non-negatif, sehingga sel termurah pada barisnya memperoleh *reduced cost* nol; tahap ini menyatakan biaya sebagai biaya peluang relatif terhadap pilihan terbaik dalam barisnya. Pengurangan minimum kolom kemudian melengkapi pasangan (u, v) menjadi *dual* layak, yaitu $u_i + v_j \leq c_{ij}$, sehingga entri matriks tereduksi tepat menjadi *reduced cost* $\bar{c}_{ij} = c_{ij} - u_i - v_j \geq 0$ pada Persamaan (2.6). *Mengapa terbentuk matriks baru*: matriks tereduksi bukan permasalahan yang berbeda, melainkan permasalahan yang sama yang dinyatakan

ulang dalam ruang biaya peluang dengan himpunan solusi optimal yang lestari (Proposisi 2.2). *Mengapa cukup satu putaran*: berdasarkan Proposisi 2.3, setelah reduksi baris lalu kolom, setiap baris dan kolom aktif dijamin memuat sedikitnya satu entri $\bar{c}_{ij} = 0$ dan putaran berikutnya tidak mengubah matriks. Reduksi ini berakar pada metode Hungarian (Kuhn, 1955) dan *Zero Point Method* (Pandian dan Natarajan, 2010). Matriks tereduksi $\bar{C} = [\bar{c}_{ij}]$ inilah yang menjadi dasar perhitungan *penalty*.

3.1.3 Perhitungan Penalty Vogel

Untuk setiap baris aktif $i \in I$, dimisalkan $s_1^{(i)} \leq s_2^{(i)}$ adalah dua nilai terkecil dari $\{\bar{c}_{ij} : j \in J\}$. *Penalty* baris didefinisikan sebagai selisih keduanya,

$$p_i^{\text{row}} = s_2^{(i)} - s_1^{(i)}. \quad (3.3)$$

Karena setiap baris aktif memuat entri nol pada matriks tereduksi, maka $s_1^{(i)} = 0$ sehingga $p_i^{\text{row}} = s_2^{(i)}$, yaitu nilai tereduksi terkecil kedua pada baris tersebut. *Mengapa penalty berbentuk selisih dua terkecil*: berdasarkan Subbab 2.2.6, *penalty* merupakan ukuran *regret*, yaitu biaya peluang marginal yang ditanggung apabila lini tersebut terpaksa beralih dari sel *tight* (ber-reduced cost nol) ke pilihan terbaik keduanya. Pada matriks tereduksi, nilai ini tepat sama dengan *reduced cost* terkecil kedua, sehingga lini dengan *penalty* besar adalah lini yang paling mahal untuk ditunda alokasinya (Reinfeld dan Vogel, 1958; Russell, 1969). *Penalty* kolom p_j^{col} dihitung dengan cara yang serupa pada $\{\bar{c}_{ij} : i \in I\}$. Apabila suatu baris atau kolom aktif hanya memiliki satu sel tersisa, *penalty*-nya ditetapkan sama dengan nilai sel tunggal tersebut.

3.1.4 Pemilihan Baris atau Kolom dan Alokasi

Baris atau kolom dengan *penalty* terbesar dipilih sebagai lini alokasi. Apabila terdapat lebih dari satu lini dengan *penalty* sama, baris diprioritaskan sebelum kolom, dan indeks terkecil dipilih lebih dahulu. Pada lini terpilih, alokasi dilakukan ke sel dengan biaya tereduksi terkecil, yaitu sel bernilai $\bar{c}_{ij} = 0$. *Mengapa dialokasikan ke sel bernilai nol*: menurut teorema *complementary slackness* (Subbab 2.2.3), solusi optimal hanya menempatkan alokasi positif pada sel ber-reduced cost nol; sel *tight* ini merupakan pilihan yang konsisten secara *dual* sehingga alokasi padanya menjaga solusi tetap dekat ke struktur optimal. Jumlah yang dialokasikan adalah

$$q = \min\{a_i, b_j\}, \quad (3.4)$$

sehingga $x_{ij} = q$. Apabila terdapat lebih dari satu sel bernilai nol pada lini terpilih, sel dipilih berdasarkan urutan kriteria: alokasi terbesar q , kemudian biaya asli c_{ij} terkecil, dan terakhir indeks terkecil.

Sebagai ilustrasi pemilihan lini dan alokasi, tinjau contoh matriks tereduksi pada Tabel 3.1. Pada submatriks aktif berukuran 2×3 ini, nilai pada sel menyatakan biaya tereduksi ($\bar{c}_{ij}$).

Tabel 3.1 Contoh Ilustrasi Pemilihan Lini dan Alokasi

	D_1	D_2	D_3	Supply	Penalty Baris
S_1	2	0	4	30	2
S_2	0	3	0	40	0
Demand	20	20	30		
Penalty Kolom	2	3	4		

Berdasarkan Tabel 3.1, langkah-langkah keputusan RVM adalah sebagai berikut:

1. Penentuan penalti

Untuk setiap baris dan kolom aktif, dihitung selisih antara nilai tereduksi terkecil kedua dan terkecil pertama. Sebagai contoh, pada kolom D_3 , dua nilai terkecil adalah 0 dan 4, sehingga penaltinya adalah $4 - 0 = 4$.

2. Pemilihan lini

Lini dengan penalti terbesar terpilih untuk alokasi. Dalam contoh ini, kolom D_3 memiliki penalti terbesar (4) dibandingkan dengan baris S_1 (2), baris S_2 (0), kolom D_1 (2), dan kolom D_2 (3), sehingga kolom D_3 terpilih.

3. Pemilihan sel

Pada kolom terpilih D_3 , dicari sel yang memiliki nilai biaya tereduksi nol ($\bar{c}_{ij} = 0$), yaitu sel (S_2, D_3).

4. Alokasi

Kuantitas alokasi dihitung berdasarkan batas minimum kapasitas yaitu $q = \min\{a_2, b_3\} = \min\{40, 30\} = 30$. Alokasi sebesar 30 dilakukan pada sel (S_2, D_3), dan kolom D_3 dinonaktifkan karena permintaannya telah terpenuhi sepenuhnya.

3.1.5 Tie-Breaking

Tie-breaking, yaitu aturan pemecah kesamaan nilai (seri) ketika lebih dari satu lini atau sel sama-sama memenuhi kriteria terbaik, menentukan kualitas sekaligus keterulangan hasil karena RVM bersifat deterministik dan satu lintasan. Prinsip yang dianut adalah bahwa *setiap tie-breaking harus bertumpu pada nilai yang relevan dengan struktur persoalan, bukan sekadar urutan penelusuran data*. *tie-breaking* yang hanya bersandar pada indeks, yaitu lini atau sel mana yang kebetulan ditemukan lebih dahulu, tidak menambah daya pembeda apa pun terhadap VAM klasik yang memang kerap dikritik karena penanganan serinya sembarang. Karena itu indeks dalam RVM ditempatkan sebagai jaring penjamin determinisme paling akhir, bukan sebagai kriteria utama. Seri dapat muncul pada dua tingkat yang menjawab pertanyaan berbeda, sehingga kriteria pemecahnya pun ikut berbeda.

Tingkat pertama: pemilihan lini (baris atau kolom). Pertanyaan pada tingkat ini adalah *lini mana yang paling mendesak untuk dilayani lebih dahulu*. Kriteria utamanya tetap *penalty* terbesar. Ketika beberapa lini berbagi *penalty* maksimum yang sama, RVM memilih lini **paling terbatas**, yaitu lini dengan jumlah sel ber-*reduced cost* nol paling sedikit. Banyaknya sel ber-*reduced cost* nol mencerminkan banyaknya opsi berbiaya-peluang nol yang masih dimiliki lini tersebut; makin sedikit jumlahnya, makin sempit ruang geraknya. Kriteria ini mengadopsi heuristik *Most Constrained Variable*, yang juga dikenal sebagai *Minimum Remaining*

Values (MRV), dari ranah *Constraint Satisfaction Problem* (Haralick & Elliott, 1980; Russell & Norvig, 2021). Heuristik ini menetapkan bahwa entitas dengan pilihan tersisa paling sedikit sebaiknya ditangani lebih dahulu (prinsip *fail-first*). Pada konteks transportasi, lini dengan sedikit opsi berisiko kehilangan satu-satunya opsi murah jika *supply* atau *demand* penyangganya lebih dahulu terpakai oleh lini lain. Melayaninya terlebih dahulu, selagi opsi itu masih tersedia, mencegah keterpaksaan beralih ke sel mahal pada langkah-langkah akhir, sedangkan lini yang fleksibel (memiliki banyak sel nol) masih dapat menunggu. Jika kriteria MRV pun masih seri, baris didahulukan atas kolom sebagai konvensi simetris karena *transportation problem* simetris terhadap transpos baris dan kolom, dan indeks terkecil dipakai sebagai pemutus terakhir.

Untuk memperjelas penanganan seri tingkat pertama, di bawah ini ditunjukkan ilustrasi terisolasi untuk masing-masing kriteria *tie-breaking* pada RVM:

Kasus 1. Tie-Breaking oleh MRV (Tingkat Keterdesakan)

Kasus ini menunjukkan bagaimana kriteria MRV memutus kesamaan penalti dengan memilih lini yang paling terbatas (jumlah sel nol paling sedikit). Ditinjau submatriks aktif berukuran 4×3 pada Tabel 3.2. Di sini, seluruh sel aktif memiliki penalti 0, sehingga terdapat situasi seri untuk seluruh baris dan kolom aktif.

Tabel 3.2 Contoh Ilustrasi *tie-breaking* oleh MRV (Kasus 4×3)

	D_1	D_2	D_3	Supply	Penalti	MRV	Keputusan
S_1	0	0	0	15	0	3	Eliminasi (MRV=3)
S_2	0	0	0	20	0	3	Eliminasi (MRV=3)
S_3	0	0	0	25	0	3	Eliminasi (MRV=3)
S_4	0	0	4	30	0	2	Terpilih (MRV)
Demand	30	30	30				
Penalti	0	0	0				
MRV	4	4	3				
Keputusan	Eliminasi	Eliminasi	Eliminasi				

Berdasarkan Tabel 3.2, seluruh baris dan kolom memiliki penalti maksimal 0. Jumlah sel nol pada tiap baris dan kolom aktif adalah: baris S_1, S_2, S_3 masing-masing 3 sel nol; baris S_4 hanya 2 sel nol karena sel $(S_4, D_3) = 4$; kolom D_1 dan D_2 masing-masing 4 sel nol; kolom D_3 3 sel nol.

Sesuai heuristik MRV, lini yang paling terbatas (jumlah sel nol paling sedikit) dipilih lebih dahulu. Nilai MRV minimum di antara seluruh lini adalah 2, hanya dimiliki oleh baris S_4 , sehingga baris ini terpilih secara unik tanpa memerlukan aturan prioritas tambahan.

Kasus 2. Tie-Breaking oleh Prioritas Baris

Kasus ini menunjukkan bagaimana prioritas baris memutus kesamaan ketika terdapat nilai MRV terkecil yang sama antara baris dan kolom. Ditinjau submatriks aktif berukuran 3×3 pada Tabel 3.3.

Tabel 3.3 Contoh Ilustrasi *tie-breaking* oleh Prioritas Baris (Kasus 3×3)

	D_1	D_2	D_3	Supply	Penalti	MRV	Keputusan
S_1	0	0	0	15	0	3	Eliminasi (MRV=3)
S_2	0	0	4	20	0	2	Terpilih (Baris)
S_3	0	0	0	25	0	3	Eliminasi (MRV=3)
Demand	10	20	30				
Penalti	0	0	0				
MRV	3	3	2				
Keputusan	Eliminasi	Eliminasi	Kalah Prior.				

Berdasarkan Tabel 3.3, seluruh lini memiliki penalti 0. Evaluasi MRV menghasilkan nilai terkecil 2, yang dimiliki oleh baris S_2 dan kolom D_3 . Karena terjadi seri pada nilai MRV minimum, aturan prioritas RVM mendahulukan baris dibandingkan kolom. Dengan demikian, baris S_2 terpilih sebagai lini alokasi, sedangkan kolom D_3 dinyatakan kalah prioritas.

Kasus 3. *Tie-Breaking* oleh Prioritas Indeks

Kasus ini menunjukkan bagaimana prioritas indeks memutus kesamaan ketika beberapa lini sejenis (sesama baris atau sesama kolom) memiliki nilai MRV terkecil yang sama. Ditinjau submatriks aktif berukuran 3×2 pada Tabel 3.4.

Tabel 3.4 Contoh Ilustrasi *tie-breaking* oleh Prioritas Indeks (Kasus 3×2)

	D_1	D_2	Supply	Penalti	MRV	Keputusan
S_1	0	0	15	0	2	Terpilih (Indeks)
S_2	0	0	20	0	2	Kalah Indeks
S_3	0	0	25	0	2	Kalah Indeks
Demand	30	30				
Penalti	0	0				
MRV	3	3				
Keputusan	Eliminasi	Eliminasi				

Berdasarkan Tabel 3.4, seluruh lini memiliki penalti 0. Hasil perhitungan MRV terkecil adalah 2, yang dimiliki bersama oleh baris S_1 , S_2 , dan S_3 . Karena terjadi seri pada sesama baris, maka prioritas indeks terkecil diterapkan, sehingga baris S_1 terpilih karena indeksnya paling kecil.

Tingkat kedua: pemilihan sel (penentuan pasangan) pada lini terpilih. Setelah satu lini ditetapkan, pertanyaannya berubah menjadi *pada sel mana, yakni dipasangkan dengan kolom atau baris yang mana, alokasi sebaiknya ditempatkan.* Beberapa sel pada lini terpilih dapat sama-sama bernilai $\bar{c}_{ij} = 0$. Pada tingkat ini kriteria MRV tidak lagi relevan, karena yang dibandingkan bukan fleksibilitas antarl lini melainkan sel-sel di dalam satu lini yang fleksibilitasnya sudah tetap. Digunakan tiga *tie-breaking* berurutan: (i) alokasi terbesar $q = \min\{a_i, b_j\}$, (ii) biaya asli terkecil c_{ij} , (iii) prioritas indeks terkecil. Di bawah ini ditunjukkan ilustrasi ter-

isolasi untuk masing-masing kriteria *tie-breaking* tingkat kedua menggunakan submatriks aktif berukuran 2×2 :

Kasus 1. Tie-Breaking Sel oleh Alokasi Terbesar (q)

Kasus ini menunjukkan pemilihan sel berbiaya tereduksi nol berdasarkan kuantitas alokasi terbesar. Ditinjau submatriks aktif berukuran 2×2 pada Tabel 3.5. Misalkan baris S_1 telah terpilih sebagai lini alokasi.

Tabel 3.5 Ilustrasi *tie-breaking* Sel oleh Alokasi Terbesar

	D_1	D_2	Supply
S_1	0	0	15
S_2	0	0	10
Demand	5	20	
Alokasi (q)	5	15	
Keputusan Sel	Eliminasi	Terpilih	

Berdasarkan Tabel 3.5, baris terpilih S_1 memiliki dua kandidat sel berbiaya tereduksi nol, yaitu (S_1, D_1) dan (S_1, D_2) . Pemilihan sel dilakukan sebagai berikut:

1. Perhitungan alokasi kandidat pertama
Untuk sel (S_1, D_1) , kapasitas *supply* adalah 15 dan *demand* adalah 5, sehingga alokasi maksimum yang dapat dilakukan adalah $q = \min\{15, 5\} = 5$.
2. Perhitungan alokasi kandidat kedua
Untuk sel (S_1, D_2) , kapasitas *supply* adalah 15 dan *demand* adalah 20, sehingga alokasi maksimum yang dapat dilakukan adalah $q = \min\{15, 20\} = 15$.
3. Pemilihan alokasi terbesar
Karena alokasi $q = 15$ pada sel (S_1, D_2) lebih besar dibandingkan alokasi $q = 5$ pada sel (S_1, D_1) , maka sel (S_1, D_2) secara unik terpilih untuk alokasi.

Kasus 2. Tie-Breaking Sel oleh Biaya Asli Terkecil (c_{ij})

Kasus ini menunjukkan bagaimana biaya asli digunakan untuk memutus seri ketika kapasitas alokasi dari kandidat sel bernilai sama besar. Ditinjau submatriks aktif berukuran 2×2 pada Tabel 3.6. Misalkan baris S_1 telah terpilih sebagai lini alokasi. Nilai di dalam sel menyatakan biaya tereduksi $\bar{c}_{ij}$, sedangkan nilai dalam tanda kurung menyatakan biaya asli c_{ij} pada sel bersangkutan.

Tabel 3.6 Ilustrasi *tie-breaking* Sel oleh Biaya Asli Terkecil

	D_1	D_2	Supply
S_1	0 (12)	0 (4)	10
S_2	0	0	15
Demand	10	15	
Alokasi (q)	10	10	
Biaya Asli (c_{ij})	12	4	
Keputusan Sel	Eliminasi	Terpilih	

Berdasarkan Tabel 3.6, baris terpilih S_1 memiliki dua kandidat sel berbiaya tereduksi nol, yaitu (S_1, D_1) dan (S_1, D_2) . Pemilihan sel dilakukan sebagai berikut:

1. Perbandingan alokasi maksimum

Untuk sel (S_1, D_1) , alokasi maksimum adalah $q = \min\{10, 10\} = 10$. Untuk sel (S_1, D_2) , alokasi maksimum adalah $q = \min\{10, 15\} = 10$. Karena alokasi kedua kandidat bernilai sama ($q = 10$), terjadi seri pada kriteria alokasi.

2. Perbandingan biaya asli

Untuk memutus kesamaan alokasi, dibandingkan nilai biaya asli (c_{ij}) masing-masing kandidat. Biaya asli sel (S_1, D_1) adalah $c_{11} = 12$, sedangkan biaya asli sel (S_1, D_2) adalah $c_{12} = 4$.

3. Pemilihan biaya asli terkecil

Karena biaya asli sel (S_1, D_2) bernilai 4 (lebih kecil dari 12), maka sel (S_1, D_2) terpilih untuk menerima alokasi.

Kasus 3. Tie-Breaking Sel oleh Prioritas Indeks Terkecil

Kasus ini menunjukkan bagaimana indeks kolom terkecil digunakan sebagai jaring penjamin terakhir ketika alokasi dan biaya asli kedua sel sama-sama bernilai seri. Ditinjau submatriks aktif berukuran 2×2 pada Tabel 3.7. Misalkan baris S_1 telah terpilih sebagai lini alokasi. Nilai di dalam sel menyatakan biaya tereduksi $\bar{c}_{ij}$, sedangkan nilai dalam tanda kurung menyatakan biaya asli c_{ij} pada sel bersangkutan.

Tabel 3.7 Ilustrasi *tie-breaking* Sel oleh Prioritas Indeks

	D_1	D_2	Supply
S_1	0 (8)	0 (8)	10
S_2	0	0	15
Demand	10	15	
Alokasi (q)	10	10	
Biaya Asli (c_{ij})	8	8	
Indeks Kolom	1	2	
Keputusan Sel	Terpilih	Kalah Indeks	

Berdasarkan Tabel 3.7, baris terpilih S_1 memiliki dua kandidat sel berbiaya tereduksi nol, yaitu (S_1, D_1) dan (S_1, D_2) . Pemilihan sel dilakukan sebagai berikut:

1. Perbandingan alokasi maksimum

Untuk sel (S_1, D_1) , alokasi maksimum adalah $q = \min\{10, 10\} = 10$. Untuk sel (S_1, D_2) , alokasi maksimum adalah $q = \min\{10, 15\} = 10$. Terjadi seri pada alokasi ($q = 10$).

2. Perbandingan biaya asli

Biaya asli sel (S_1, D_1) adalah $c_{11} = 8$ dan biaya asli sel (S_1, D_2) adalah $c_{12} = 8$. Terjadi seri pada kriteria biaya asli ($c_{ij} = 8$).

3. Pemilihan indeks kolom terkecil

Karena terjadi kesamaan pada alokasi dan biaya asli, prioritas kolom dengan indeks terkecil digunakan. Kolom D_1 memiliki indeks kolom lebih kecil dari kolom D_2 (indeks 1 vs 2), sehingga sel (S_1, D_1) terpilih untuk menerima alokasi.

Alasan kedua tingkat memakai kriteria berbeda. Perbedaan ini muncul karena objek yang dibandingkan pada kedua tingkat memang berbeda, bukan karena rancangan yang tidak konsisten. Pada tingkat pertama, kandidat yang bersaing adalah lini-lini utuh, sehingga ukuran yang tepat adalah tingkat keterdesakan tiap lini yang tercermin dari jumlah opsi tersisa (MRV). Pada tingkat kedua, lini sudah terpilih dan kandidatnya adalah sel-sel pasangan di dalam lini yang sama, sehingga fleksibilitas tidak lagi membedakan; yang relevan adalah dampak penempatan, yaitu besar alokasi sebagai ukuran kemajuan dan biaya asli sebagai pengaman. Dengan kata lain, tingkat pertama menentukan lini mana yang dilayani lebih dahulu berdasarkan keterdesakan, sedangkan tingkat kedua menentukan sel mana di dalam lini tersebut yang diisi berdasarkan kemajuan dan biaya.

Seluruh keputusan seri tetap diarahkan ke sel ber-*reduced cost* nol. Berapa pun banyaknya seri yang muncul, alokasi tidak pernah keluar dari himpunan sel yang konsisten dengan *complementary slackness*; aturan *tie-breaking* hanya menentukan urutan pengisian di antara sel-sel yang sama-sama optimal-konsisten. Inilah sebabnya pada kasus dengan banyak seri sekalipun solusi yang dihasilkan tetap dekat dengan struktur optimal, meskipun pada sebagian kecil kasus pilihan yang optimal secara lokal tidak menuju optimum global (dibahas pada Subbab 4.2.2).

3.1.6 Kriteria Berhenti dan Penanganan Degenerasi

Setelah alokasi, sisa *supply* dan *demand* diperbarui, yaitu $a_i \leftarrow a_i - q$ dan $b_j \leftarrow b_j - q$. Baris i dinonaktifkan apabila $a_i = 0$, dan kolom j dinonaktifkan apabila $b_j = 0$. Proses berulang hingga tidak ada lagi baris atau kolom aktif. Karena setiap alokasi menonaktifkan sedikitnya satu baris atau kolom, RVM berhenti setelah paling banyak $m + n - 1$ alokasi.

Pada kasus tertentu, sisa *supply* dan *demand* dapat mencapai nol secara bersamaan dalam satu alokasi, sehingga sebuah baris dan sebuah kolom dinonaktifkan sekaligus. Akibatnya, jumlah sel basis positif yang dihasilkan menjadi kurang dari $m + n - 1$, yang berarti solusi memiliki lebih sedikit “sel terisi” daripada yang idealnya dibutuhkan untuk menggambarkan strukturnya secara unik. Sebagai gambaran, pada TP berukuran 3×4 (semestinya $3 + 4 - 1 = 6$ sel basis), apabila salah satu iterasi menghabiskan *supply* dan *demand* secara bersamaan, hanya 5 sel yang terisi; kondisi inilah yang disebut *degenerasi*. Degenerasi merupakan fenomena standar pada metode konstruksi IBFS dan tidak memengaruhi kelayakan maupun biaya solusi. Alokasi yang dihasilkan tetap sah dan optimal atau mendekati optimal sebagaimana mestinya. Pengaruhnya baru terasa apabila solusi hendak dilanjutkan dengan prosedur optimisasi seperti MODI, yang *mensyaratkan* basis berjumlah tepat $m + n - 1$ sel untuk menghitung nilai *dual* u_i, v_j pada Persamaan (2.5) secara unik (Subbab 2.2.3); pada kondisi ini, basis cukup dilengkapi dengan sel tambahan bernilai alokasi nol (dipilih agar tidak membentuk siklus pada graf bipartit sumber–tujuan, Subbab 2.2) hingga jumlahnya tepat $m + n - 1$, tanpa mengubah biaya total.

3.1.7 Analisis Kelayakan, Terminasi, dan Kompleksitas

Proposisi berikut menjamin bahwa RVM selalu menghasilkan solusi yang valid dalam waktu polinomial.

Proposisi 3.1 (Kelayakan dan terminasi). *Untuk balanced TP, RVM berhenti setelah paling*

banyak $m + n - 1$ iterasi dan menghasilkan solusi layak dengan paling banyak $m + n - 1$ alokasi positif.

Bukti. Setiap iterasi mengalokasikan $q = \min\{a_i, b_j\}$, sehingga sedikitnya satu dari sisa *supply* a_i atau *demand* b_j menjadi nol, dan lini bersangkutan dinonaktifkan. Karena terdapat m baris dan n kolom, dan iterasi terakhir menonaktifkan satu baris dan satu kolom sekaligus (sebab total *supply* sama dengan total *demand*), jumlah iterasi tidak melebihi $m + n - 1$. Setiap alokasi mengurangi *supply* dan *demand* dengan jumlah yang sama, dan karena permasalahan seimbang, seluruh *supply* dan *demand* terpenuhi tepat ketika proses berhenti, sehingga solusi yang dihasilkan layak. Banyaknya alokasi positif sama dengan banyaknya iterasi, yaitu paling banyak $m + n - 1$. $\square$

Dari sisi kompleksitas, setiap iterasi melakukan reduksi *zero-point* dan perhitungan *penalty* pada submatriks aktif yang berukuran paling besar $m \times n$, masing-masing memerlukan waktu $O(mn)$. Karena terdapat paling banyak $m + n - 1$ iterasi, kompleksitas waktu keseluruhan adalah $O(mn(m + n))$, sedangkan kompleksitas ruang adalah $O(mn)$ untuk menyimpan matriks biaya dan alokasi. Kompleksitas polinomial ini setara dengan VAM dan metode konstruksi sejenis, sehingga RVM tetap efisien untuk kasus berukuran besar seperti *instance* 2×94 pada penelitian ini.

3.1.8 Catatan Implementasi

Implementasi memelihara himpunan baris dan kolom aktif menggunakan penanda boolean, dan submatriks aktif tidak disalin melainkan ditelusuri melalui penanda tersebut untuk menghemat memori. Reduksi dan *penalty* dihitung ulang setiap iterasi karena penonaktifan lini mengubah nilai minimum baris dan kolom. Seluruh aturan *tie-breaking* diimplementasikan secara deterministik. Pada pemilihan lini, *penalty* terbesar lalu lini paling terbatas (MRV), dengan baris-dulu dan indeks terkecil hanya sebagai pemutus akhir; pada pemilihan sel, alokasi terbesar lalu biaya asli terkecil. Dengan demikian keluaran metode tunggal dan dapat direproduksi. Sebagai verifikasi silang, implementasi dalam bahasa C dibandingkan dengan implementasi acuan dalam Python pada seluruh kasus uji, dan keduanya menghasilkan biaya yang identik. Tabel 3.8 merangkum landasan matematis setiap tahapan inti RVM, yang menegaskan bahwa metode ini bukan prosedur *ad hoc* melainkan dibangun dari teori dualitas *transportation problem*.

Tabel 3.8 Justifikasi Matematis Tiap Tahapan RVM

Tahapan	Mengapa demikian	Landasan
Kurangi min baris (u_i)	u_i adalah <i>dual</i> baris terbesar yang menjaga entri non-negatif; sel termurah baris menjadi ber- <i>reduced cost</i> nol	Dualitas TP, Persamaan (2.5); Kuhn (1955)
Kurangi min kolom (v_j)	Melengkapi (u, v) menjadi <i>dual</i> layak ($u_i + v_j \leq c_{ij}$); entri menjadi <i>reduced cost</i>	Persamaan (2.6); Bazaraa et al. (2010)
Bekerja pada matriks baru	Permasalahan sama dinyatakan ulang; himpunan solusi optimal lestari	Proposisi 2.2
Cukup satu putaran	Tiap baris dan kolom dijamin memuat nol; putaran berikutnya tak mengubah	Proposisi 2.3
$Penalty = s_2 - s_1$	Ukuran <i>regret</i> ; pada matriks tereduksi sama dengan <i>reduced cost</i> terkecil kedua	Subbab 2.2.6; Reinfield dan Vogel (1958)
Alokasi ke sel nol	Sel <i>tight</i> ($\bar{c}_{ij} = 0$) adalah kandidat optimal	<i>Complementary slackness</i> , Subbab 2.2.3

3.1.9 Pseudocode

Prosedur kerja RVM dibagi menjadi tiga tahap utama yang saling terintegrasi: reduksi matriks sekuensial (Kode Semu 1), perhitungan penalti Vogel tereduksi serta aturan *tie-breaking* (Kode Semu 2), dan prosedur alokasi utama RVM (Kode Semu 3).

Kode Semu 1 Reduksi Zero-Point Sekuensial

Require: Matriks biaya aktif C , himpunan baris aktif I , himpunan kolom aktif J

Ensure: Matriks *reduced cost* $\bar{C}$

```

1: for all  $i \in I$  do
2:    $u_i \leftarrow \min_{j \in J} c_{ij}$  {Hitung biaya minimum baris}
3: end for
4: for all  $j \in J$  do
5:    $v_j \leftarrow \min_{i \in I} (c_{ij} - u_i)$  {Hitung biaya minimum kolom tereduksi}
6: end for
7: for all  $i \in I, j \in J$  do
8:    $\bar{c}_{ij} \leftarrow c_{ij} - u_i - v_j$  {Bentuk reduced cost berbasis dualitas}
9: end for
10: return  $\bar{C}$ 

```

Kode Semu 2 Perhitungan Penalti Vogel Tereduksi dan Pemilihan Lini

Require: Matriks *reduced cost* $\bar{C}$, himpunan I, J

Ensure: Lini terpilih L^* dan sel kandidat (i^*, j^*)

- 1: Hitung penalti p_i^{row} (selisih dua nilai $\bar{c}_{ij}$ terkecil pada baris i) untuk seluruh $i \in I$
 - 2: Hitung penalti p_j^{col} (selisih dua nilai $\bar{c}_{ij}$ terkecil pada kolom j) untuk seluruh $j \in J$
 - 3: $P^* \leftarrow \max(\{p_i^{\text{row}}\} \cup \{p_j^{\text{col}}\})$
 - 4: $L^* \leftarrow$ Lini dengan penalti P^* (Pemutus seri: pilih lini dengan q terbanyak/MRV, lalu baris-dulu, lalu indeks terkecil)
 - 5: $(i^*, j^*) \leftarrow$ Sel pada L^* dengan $\bar{c}_{ij} = 0$ (Pemutus seri: pilih q terbesar, lalu c_{ij} nominal terkecil, lalu indeks terkecil)
 - 6: **return** $L^*, (i^*, j^*)$
-

Kode Semu 3 Algoritma Utama Reduced-cost Vogel Method (RVM)

Require: *Balanced TP*: matriks biaya C , *supply* a , *demand* b

Ensure: IBFS berupa matriks alokasi x

- 1: $x_{ij} \leftarrow 0$ untuk semua i, j
 - 2: $I \leftarrow \{1, \dots, m\}; \quad J \leftarrow \{1, \dots, n\}$
 - 3: **while** $I \neq \emptyset$ **and** $J \neq \emptyset$ **do**
 - 4: $\bar{C} \leftarrow \text{ReduksiZeroPoint}(C, I, J)$ {Memanggil Kode Semu 1}
 - 5: $L^*, (i, j) \leftarrow \text{PilihLiniDanSel}(\bar{C}, I, J)$ {Memanggil Kode Semu 2}
 - 6: $q \leftarrow \min\{a_i, b_j\}; \quad x_{ij} \leftarrow q$
 - 7: $a_i \leftarrow a_i - q; \quad b_j \leftarrow b_j - q$
 - 8: **if** $a_i = 0$ **then**
 - 9: $I \leftarrow I \setminus \{i\}$
 - 10: **end if**
 - 11: **if** $b_j = 0$ **then**
 - 12: $J \leftarrow J \setminus \{j\}$
 - 13: **end if**
 - 14: **end while**
 - 15: **return** x
-

3.1.10 Penjelasan Umum Tahapan Algoritma

Sebelum ditelusuri pada satu kasus konkret di Subbab 3.1.11, tahapan eksekusi RVM (Kode Semu 1–3) diringkas secara umum berikut.

1. **Inisialisasi.** Menerima masukan berupa matriks biaya $C_{m \times n}$, vektor *supply* $a = [a_i]$, dan vektor *demand* $b = [b_j]$, lalu memeriksa keseimbangan $\sum a_i = \sum b_j$. Jika seimbang, seluruh baris $I = \{1, \dots, m\}$ dan kolom $J = \{1, \dots, n\}$ ditetapkan aktif dan matriks alokasi x_{ij} diinisialisasi nol.
2. **Reduksi *zero-point* sekuensial.** Submatriks aktif direduksi dalam dua sub-tahap. Untuk tiap baris aktif i dihitung $u_i = \min_{j \in J} c_{ij}$ lalu dikurangkan, $c_{ij} \leftarrow c_{ij} - u_i$; untuk tiap kolom aktif j dihitung $v_j = \min_{i \in I} c_{ij}$ dari hasil reduksi baris, lalu dikurangkan, $\bar{c}_{ij} \leftarrow c_{ij} - v_j$. Matriks $\bar{C} = [\bar{c}_{ij}]$ yang dihasilkan menjamin tiap baris dan kolom aktif memuat sedikitnya satu sel nol.

3. **Perhitungan penalti.** Untuk tiap baris dan kolom aktif dihitung penalti $p = s_2 - s_1$, yaitu selisih nilai *reduced cost* terkecil kedua dan terkecil pertama pada matriks tereduksi. Karena $s_1 = 0$ (dijamin langkah sebelumnya), penalti tepat sama dengan nilai terkecil kedua pada lini tersebut.
4. **Pemilihan lini.** Lini (baris atau kolom) dengan penalti maksimum $p_{\max}$ dipilih sebagai lini alokasi. Ketika lebih dari satu lini berbagi $p_{\max}$ yang sama (*tie-breaking*, Subbab 3.1.5), aturan hierarki berikut diterapkan berurutan sampai seri terpecah:
 - (a) Lini dengan penalti $p = p_{\max}$ menjadi kandidat. Jika hanya satu lini yang memenuhi, lini itu langsung terpilih dan proses berhenti di sini.
 - (b) Jika masih ada lebih dari satu kandidat, lini paling terbatas menurut MRV (*Most Constrained Variable*), yaitu lini dengan jumlah sel ber-*reduced cost* nol paling sedikit, yang menang.
 - (c) Jika MRV masih sama di antara kandidat yang tersisa, baris diprioritaskan atas kolom.
 - (d) Jika seri masih terjadi juga pada tingkat ini (selalu antara sesama baris atau sesama kolom, karena tingkat sebelumnya sudah memisahkan baris dari kolom), lini dengan indeks terkecil menjadi pemenang akhir.

Keempat tingkat ini bersifat berjenjang: setiap tingkat hanya dievaluasi jika tingkat sebelumnya belum berhasil memecah seri.
5. **Pemilihan sel.** Pada lini terpilih, hanya sel dengan $\bar{c}_{ij} = 0$ yang menjadi kandidat alokasi. Jika lebih dari satu sel memenuhi syarat ini, *tie-breaking* tingkat kedua diterapkan berurutan: kuantitas alokasi $q = \min\{a_i, b_j\}$ terbesar dipilih lebih dulu; jika masih seri, biaya asli c_{ij} terkecil yang menang; jika masih seri juga, indeks terkecil menjadi penentu akhir.
6. **Eksekusi alokasi.** Alokasi $x_{ij} = q$ ditetapkan pada sel terpilih, sisa kapasitas diperbarui ($a_i \leftarrow a_i - q$ dan $b_j \leftarrow b_j - q$), dan lini yang kapasitasnya mencapai nol dinonaktifkan dari himpunan aktif.
7. **Pemeriksaan kondisi berhenti.** Jika masih ada baris atau kolom aktif, algoritma kembali ke tahap reduksi (tahap 2) pada submatriks aktif yang baru; jika tidak ada lagi baris maupun kolom aktif, matriks alokasi x dikembalikan sebagai IBFS akhir.

3.1.11 Penyelesaian Lengkap Langkah demi Langkah

Untuk memperlihatkan cara kerja RVM secara menyeluruh, bagian ini menelusuri penyelesaian kasus n11 berukuran 3×4 iterasi demi iterasi. Matriks biaya awal, *supply*, dan *demand* ditunjukkan pada Tabel 3.9, dengan solusi optimal $Z^* = 2460$. Pada setiap iterasi, reduksi *zero-point* dilakukan dalam dua sub-langkah *berurutan*: pertama setiap entri dikurangi minimum barisnya ($c_{ij} - u_i$), kemudian hasilnya dikurangi minimum kolomnya ($-v_j$). Penting ditegaskan bahwa kedua pengurangan ini bersifat *sekuensial* (minimum kolom dihitung dari matriks yang sudah direduksi baris), bukan penjumlahan dua biaya peluang yang dihitung serentak dari matriks asli seperti pada TOCM; perbedaan ini berdampak nyata pada hasil, sebagaimana ditunjukkan di akhir bagian ini.

Tabel 3.9 Kasus n11 (3×4), $Z^* = 2460$

	D_1	D_2	D_3	D_4	Supply
S_1	10	2	20	22	60
S_2	12	7	9	40	100
S_3	4	14	16	32	40
Demand	20	60	60	60	

Iterasi 1. Pada awal iterasi ini seluruh baris dan kolom aktif, sehingga submatriks aktif identik dengan matriks pada Tabel 3.9. Iterasi ini ditelusuri secara rinci sebagai acuan bagi iterasi-iterasi berikutnya, yang mengikuti prosedur identik namun disajikan lebih ringkas. Reduksi baris memberi $u(S_1) = \min\{10, 2, 20, 22\} = 2$, $u(S_2) = \min\{12, 7, 9, 40\} = 7$, $u(S_3) = \min\{4, 14, 16, 32\} = 4$; reduksi kolom atas hasilnya (sekuensial, bukan serentak dari matriks asli) memberi $v(D_1) = 0$, $v(D_2) = 0$, $v(D_3) = 2$, $v(D_4) = 20$. Tabel 3.10 menyajikan biaya asli beserta *supply/demand*, sedangkan Tabel 3.11 menyajikan hasil reduksi $\bar{c}_{ij} = c_{ij} - u_i - v_j$ dengan kolom u_i dan baris v_j di tepinya, ditambah kolom **Penalti** (baris) di kanan dan baris **Penalti** (kolom) di bawah – keduanya dihitung langsung dari nilai tereduksi pada tabel yang sama, tanpa tabel terpisah.

Tabel 3.10 Iterasi 1 – Matriks Biaya Asli c_{ij}

	D_1	D_2	D_3	D_4	Supply
S_1	10	2	20	22	60
S_2	12	7	9	40	100
S_3	4	14	16	32	40
Demand	20	60	60	60	

Tabel 3.11 Iterasi 1 – Matriks Tereduksi $\bar{c}_{ij}$ dan Penalti

	D_1	D_2	D_3	D_4	Penalti (baris)
S_1	8	0	16	0	0
S_2	5	0	0	13	0
S_3	0	10	10	8	8
Penalti (kolom)	5	0	10	8	

Sel bercetak tebal pada Tabel 3.11 menandai $\bar{c}_{ij} = 0$, hasil verifikasi Proposisi 2.3: setiap baris dan setiap kolom memuat sedikitnya satu sel bernilai nol, penanda kelayakan *dual* sesuai Proposisi 2.1. Penalti maksimum bernilai 10 dan hanya dimiliki oleh kolom D_3 ; karena tidak ada lini lain yang menyamai nilai ini, D_3 langsung terpilih sebagai lini alokasi tanpa memerlukan aturan *tie-breaking* (Subbab 3.1.5). Pada kolom D_3 , sel ber- $\bar{c}_{ij} = 0$ hanya (S_2, D_3) , sehingga terpilih tanpa seri. Kuantitas alokasi $q = \min\{a_{S_2}, b_{D_3}\} = \min\{100, 60\} = 60$ dialokasikan pada $x_{S_2, D_3} = 60$; sisa *supply* S_2 menjadi $100 - 60 = 40$ (baris tetap aktif), sedangkan *demand* D_3 menjadi $60 - 60 = 0$ sehingga kolom D_3 dinonaktifkan.

Iterasi 2. Baris dan kolom aktif tersisa $\{S_1, S_2, S_3\} \times \{D_1, D_2, D_4\}$ dengan sisa *supply* $S_1 = 60$, $S_2 = 40$ ($100 - 60$), $S_3 = 40$, dan *demand* $D_1 = 20$, $D_2 = 60$, $D_4 = 60$. Reduksi baris:

$u(S_1) = 2, u(S_2) = 7, u(S_3) = 4$; reduksi kolom: $v(D_1) = 0, v(D_2) = 0, v(D_4) = 20$. Tabel 3.12 dan 3.13 menyajikan matriks asli serta matriks tereduksi dan penaltinya.

Tabel 3.12 Iterasi 2 – Matriks Biaya Asli c_{ij}

	D_1	D_2	D_4	Supply
S_1	10	2	22	60
S_2	12	7	40	40
S_3	4	14	32	40
Demand	20	60	60	

Tabel 3.13 Iterasi 2 – Matriks Tereduksi $\bar{c}_{ij}$ dan Penalti

	D_1	D_2	D_4	Penalti (baris)
S_1	8	0	0	0
S_2	5	0	13	5
S_3	0	10	8	8
Penalti (kolom)	5	0	8	

Penalti maksimum bernilai 8, dimiliki bersama oleh baris S_3 dan kolom D_4 . Karena kriteria MRV pada keduanya juga sama-sama bernilai 1 (sel nol tunggal), aturan prioritas baris-dulu (Subbab 3.1.5) memutuskan: S_3 terpilih. Pada baris S_3 , satu-satunya sel bernilai nol adalah (S_3, D_1) . Dialokasikan $q = \min\{40, 20\} = 20$; sisa *supply* S_3 menjadi $40 - 20 = 20$ (tetap aktif), sedangkan *demand* D_1 menjadi $20 - 20 = 0$, sehingga D_1 dinonaktifkan.

Iterasi 3. Baris dan kolom aktif tersisa $\{S_1, S_2, S_3\} \times \{D_2, D_4\}$ dengan sisa *supply* $S_1 = 60$, $S_2 = 40$, $S_3 = 20$, dan *demand* $D_2 = 60$, $D_4 = 60$. Reduksi baris: $u(S_1) = 2, u(S_2) = 7, u(S_3) = 14$; reduksi kolom: $v(D_2) = 0, v(D_4) = 18$. Tabel 3.14 dan 3.15 menyajikan matriks asli serta matriks tereduksi dan penaltinya.

Tabel 3.14 Iterasi 3 – Matriks Biaya Asli c_{ij}

	D_2	D_4	Supply
S_1	2	22	60
S_2	7	40	40
S_3	14	32	20
Demand	60	60	

Tabel 3.15 Iterasi 3 – Matriks Tereduksi $\bar{c}_{ij}$ dan Penalti

	D_2	D_4	Penalti (baris)
S_1	0	2	2
S_2	0	15	15
S_3	0	0	0
Penalti (kolom)	0	2	

Baris S_2 memiliki penalti terbesar (15) secara tunggal, karena alternatif kedua S_2 jauh lebih

mahal (15) dibanding sel *tight*-nya di D_2 , sehingga S_2 langsung terpilih tanpa seri. Satu-satunya sel nol pada baris S_2 adalah (S_2, D_2) . Dialokasikan $q = \min\{40, 60\} = 40$; *supply* S_2 menjadi $40 - 40 = 0$ sehingga S_2 dinonaktifkan, sedangkan *demand* D_2 menjadi $60 - 40 = 20$ (tetap aktif).

Iterasi 4. Baris dan kolom aktif tersisa $\{S_1, S_3\} \times \{D_2, D_4\}$ dengan sisa *supply* $S_1 = 60$, $S_3 = 20$, dan *demand* $D_2 = 20$, $D_4 = 60$. Reduksi baris: $u(S_1) = 2$, $u(S_3) = 14$; reduksi kolom: $v(D_2) = 0$, $v(D_4) = 18$. Tabel 3.16 dan 3.17 menyajikan matriks asli serta matriks tereduksi dan penaltinya.

Tabel 3.16 Iterasi 4 – Matriks Biaya Asli c_{ij}

	D_2	D_4	Supply
S_1	2	22	60
S_3	14	32	20
Demand	20	60	

Tabel 3.17 Iterasi 4 – Matriks Tereduksi $\bar{c}_{ij}$ dan Penalti

	D_2	D_4	Penalti (baris)
S_1	0	2	2
S_3	0	0	0
Penalti (kolom)	0	2	

Baris S_1 dan kolom D_4 seri pada nilai 2; karena baris didahulukan atas kolom (Subbab 3.1.5), S_1 terpilih. Pada baris S_1 , satu-satunya sel nol adalah (S_1, D_2) . Dialokasikan $q = \min\{60, 20\} = 20$; *demand* D_2 menjadi $20 - 20 = 0$ sehingga D_2 dinonaktifkan, sedangkan *supply* S_1 menjadi $60 - 20 = 40$ (tetap aktif).

Iterasi 5. Tersisa $\{S_1, S_3\} \times \{D_4\}$ dengan sisa *supply* $S_1 = 40$, $S_3 = 20$, dan *demand* $D_4 = 60$. Karena hanya satu kolom aktif, reduksi baris menyamakan setiap baris dengan biaya satu-satunya selnya sehingga seluruh sel menjadi nol; reduksi kolom tidak mengubah apa pun lagi karena $v(D_4) = 0$. Karena S_1 dan S_3 masing-masing hanya memiliki satu sel tersisa, penaltinya ditetapkan sama dengan nilai sel tunggal tersebut (Subbab 2.2.5), yaitu nol untuk keduanya; kolom D_4 juga bernilai nol. Tabel 3.18 dan 3.19 menyajikan matriks asli serta matriks tereduksi dan penaltinya.

Tabel 3.18 Iterasi 5 – Matriks Biaya Asli c_{ij}

	D_4	Supply
S_1	22	40
S_3	32	20
Demand	60	

Tabel 3.19 Iterasi 5 – Matriks Tereduksi $\bar{c}_{ij}$ dan Penalti

	D_4	Penalti (baris)
S_1	0	0
S_3	0	0
Penalti (kolom)	0	

Ketiga lini seri pada penalti 0, sehingga penentu turun ke MRV: baris S_1 dan S_3 sama-sama memiliki 1 sel nol (paling terbatas), sedangkan kolom D_4 memiliki 2 sel nol. MRV minimum (1) dimiliki bersama oleh S_1 dan S_3 yang keduanya sama-sama baris, sehingga pemutus terakhir adalah indeks terkecil dan S_1 terpilih. Karena hanya satu kolom aktif, satu-satunya sel pada baris S_1 adalah (S_1, D_4) . Dialokasikan $q = \min\{40, 60\} = 40$; *supply* S_1 menjadi $40 - 40 = 0$ sehingga S_1 dinonaktifkan, sedangkan *demand* D_4 menjadi $60 - 40 = 20$ (tetap aktif).

Iterasi 6. Tersisa satu baris dan satu kolom aktif, S_3 dan D_4 , dengan sisa *supply* $S_3 = 20$ dan *demand* $D_4 = 20$ yang persis sama besar. Karena hanya tersisa satu sel (S_3, D_4) pada biaya asli 32, sel itu otomatis terpilih tanpa perlu reduksi, penalti, atau *tie-breaking*. Dialokasikan $q = \min\{20, 20\} = 20$, sehingga *supply* S_3 dan *demand* D_4 sama-sama menjadi nol; baris dan kolom terakhir nonaktif bersamaan dan konstruksi berhenti. Tabel 3.20 merangkum alokasi akhir.

Tabel 3.20 Alokasi Akhir RVM pada Kasus n11

Sel	Alokasi	Biaya satuan	Biaya
(S_3, D_1)	20	4	80
(S_1, D_2)	20	2	40
(S_2, D_2)	40	7	280
(S_2, D_3)	60	9	540
(S_1, D_4)	40	22	880
(S_3, D_4)	20	32	640
Total			2460

Hasil RVM ($Z = 2460$) tepat sama dengan solusi optimal. Sebagai pembanding, VAM standar pada kasus yang sama menghasilkan $Z = 2980$ (deviasi 21,1%). Perbedaan kunci terletak pada penanganan baris S_2 . VAM, yang menghitung *penalty* dari matriks biaya asli, mengalokasikan seluruh *supply* S_1 sebesar 60 ke (S_1, D_2) karena biaya 2 tampak sangat menarik secara nominal; akibatnya D_2 terpenuhi penuh oleh S_1 dan S_2 terpaksa memenuhi D_4 melalui sel (S_2, D_4) yang berbiaya 40, menambah biaya sebesar $40 \times 40 = 1600$. Sebaliknya, RVM bekerja pada *reduced cost*. Pada iterasi 3, *penalty* S_2 yang besar (15) memaksa S_2 mengisi D_2 lebih dahulu, sehingga S_1 tersisa untuk D_4 dengan biaya 22 (bukan 40 oleh S_2). Reduksi *zero-point* dengan demikian mengoreksi kesalahan prioritas VAM yang timbul dari membandingkan biaya nominal antarbaris dengan skala berbeda, dan inilah inti keunggulan RVM yang dijelaskan secara teoretis pada Subbab 2.2.6.

Untuk menegaskan bahwa sifat *sekuensial* reduksi memang penting, dilakukan perbandingan dengan varian *aditif* yang menghitung biaya peluang baris dan kolom secara serentak dari matriks asli, yaitu $\bar{c}_{ij} = (c_{ij} - \min_k c_{ik}) + (c_{ij} - \min_l c_{lj})$, lalu menerapkan kerangka dan *penalty* yang sama. Pada kasus n11, varian aditif menghasilkan $Z = 2980$, yaitu identik dengan

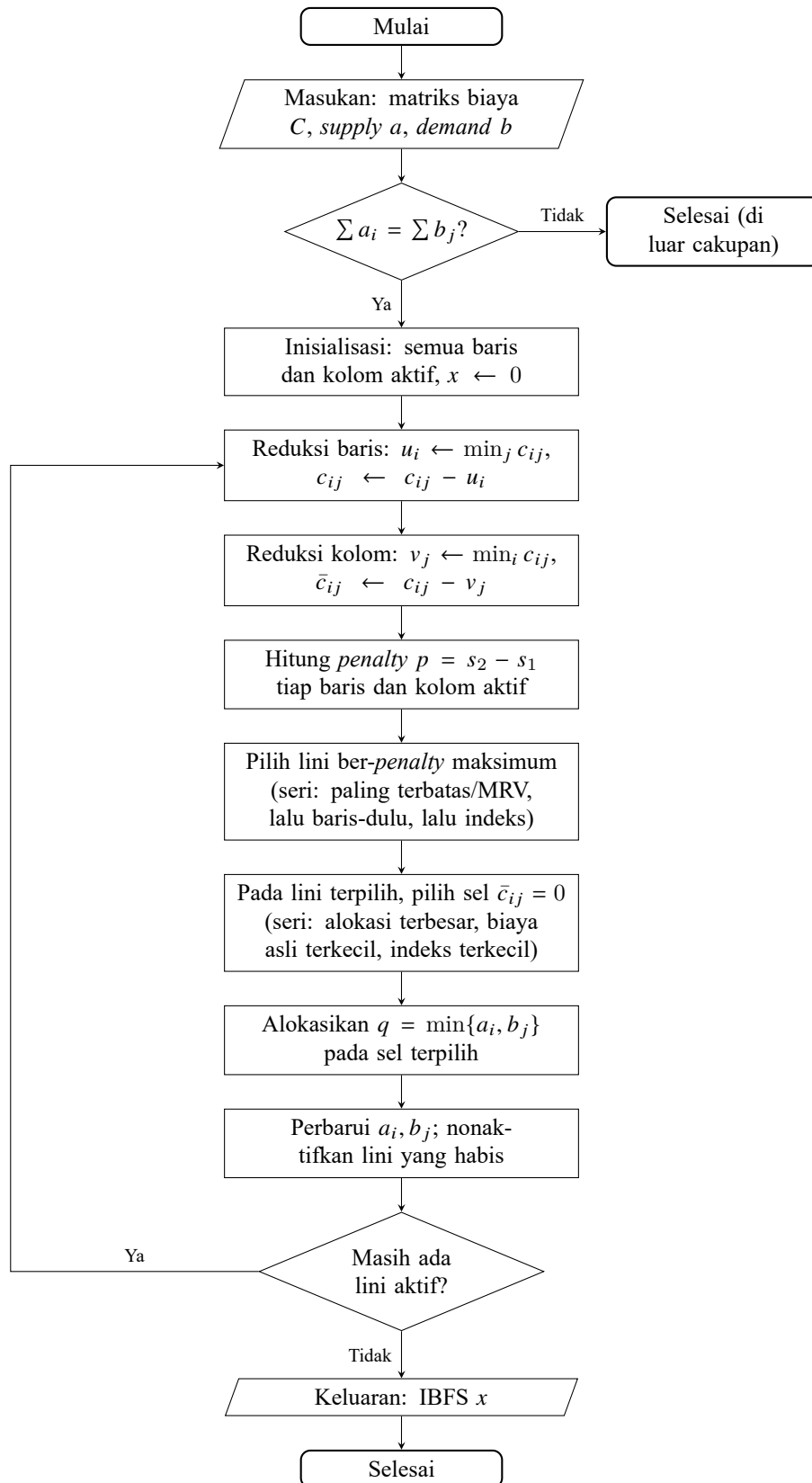
VAM dan menyimpang 21,1% dari optimal, sedangkan reduksi sekuensial menghasilkan 2460 yang optimal. Pada keseluruhan *benchmark*, varian aditif hanya mencapai 34 dari 42 kasus optimal, dibanding 38 dari 42 pada reduksi sekuensial. Hal ini mengonfirmasi bahwa keunggulan RVM bersumber pada reduksi *zero-point* yang sekuensial, yang membentuk *reduced cost* berlandaskan *dual* layak (Proposisi 2.1), bukan sekadar penjumlahan dua biaya peluang.

3.1.11.1 Ilustrasi *Tie-Breaking*

Untuk memperjelas peran aturan *tie-breaking* (Subbab 3.1.5), tinjau iterasi ketiga pada kasus n13, yang memunculkan seri pada tingkat kedua (pemilihan sel). Pada submatriks aktif, sisa *supply* $S_1 = 150$ sedangkan sisa *demand* $D_1 = 25$ dan $D_3 = 300$. Setelah reduksi, baris S_1 memiliki dua sel ber-*reduced cost* nol sekaligus, yaitu (S_1, D_1) dan (S_1, D_3) , sehingga kriteria *reduced cost* minimum tidak membedakan keduanya. *tie-breaking* pertama, yaitu alokasi terbesar, menjadi penentu: untuk (S_1, D_1) diperoleh $q = \min\{150, 25\} = 25$, sedangkan untuk (S_1, D_3) diperoleh $q = \min\{150, 300\} = 150$. Karena $150 > 25$, sel (S_1, D_3) dipilih. Keputusan ini menempatkan seluruh sisa *supply* S_1 pada satu sel *tight*, menonaktifkan baris lebih cepat, dan pada kasus ini menghasilkan solusi optimal $Z = 4525$. Contoh ini menegaskan bahwa *tie-breaking* bukan pemilihan sembarang, melainkan mekanisme yang mengarahkan seri ke sel optimal-konsisten yang memberi kemajuan terbesar.

3.1.12 Flowchart Metodologi

Alur kerja RVM secara keseluruhan divisualisasikan pada Gambar 3.1.



Gambar 3.1 Flowchart Metode RVM

Gambar 3.1 merangkum keseluruhan prosedur RVM yang telah dijabarkan pada Subbab 2.2.5–3.1.5 ke dalam satu alur tunggal. Alur dimulai dengan pemeriksaan keseimbangan $\sum a_i =$

$\sum b_j$; kasus tak seimbang berada di luar cakupan penelitian ini (Subbab 4.2) dan proses dihentikan tanpa konstruksi lebih lanjut. Apabila seimbang, inisialisasi menetapkan seluruh baris dan kolom sebagai aktif dengan matriks alokasi kosong, sebagaimana dirumuskan pada Subbab 2.2.5.

Blok perulangan (*loop*) pada flowchart, yaitu rangkaian proses dari reduksi baris hingga pembaruan *supply/demand* yang ditelusuri balik oleh cabang “Ya” pada keputusan *Masih ada lini aktif?*, dieksekusi berulang selama masih terdapat baris atau kolom aktif. Setiap kali *loop* ini ditelusuri sekali penuh, itu bersesuaian dengan satu iterasi sebagaimana dicontohkan langkah demi langkah pada Subbab 3.1.11: reduksi baris dan kolom membentuk matriks *reduced cost* (Persamaan (2.6)), penalti dihitung di atasnya, lini dengan penalti maksimum dipilih dengan aturan *tie-breaking* pada Subbab 3.1.5, sel ber-*reduced cost* nol pada lini itu diisi, dan kapasitas *supply/demand* diperbarui. Karena submatriks aktif menyusut setiap iterasi (sedikitnya satu baris atau kolom dinonaktifkan, Proposisi 3.1), keputusan *Masih ada lini aktif?* pada flowchart bersifat monoton menurun sehingga *loop* ini dijamin berhenti dalam paling banyak $m + n - 1$ putaran. Begitu tidak ada lagi baris maupun kolom aktif, flowchart keluar dari blok perulangan melalui cabang “Tidak” dan mengembalikan matriks alokasi x sebagai IBFS akhir.

3.2 Bahan dan Peralatan yang Digunakan

3.2.1 Dataset

Penelitian menggunakan 42 kasus uji *benchmark* yang merupakan himpunan yang sama dengan yang digunakan pada studi terbaru keluarga metode ini (Bunjamin et al., 2026a, 2026b), dan setiap *instance* literatur ditelusuri hingga sumber aslinya sebagaimana dirangkum pada Tabel 3.22. Pemilihan ini disengaja: karena RVM dibandingkan langsung dengan CSM dan RBSM sebagai dua metode konstruksi terkuat yang dilaporkan hingga saat ini, memakai *benchmark* yang persis sama memastikan selisih performa yang teramati murni berasal dari perbedaan algoritma, bukan dari kasus uji yang kebetulan menguntungkan salah satu metode. Menyusun himpunan kasus sendiri membuka celah bias tak disengaja, misalnya memilih kasus yang secara struktur cocok dengan cara kerja RVM, yang akan melemahkan validitas klaim keunggulannya. Beberapa *instance* yang pada sumber aslinya tidak seimbang diseimbangkan dengan menambahkan baris atau kolom *dummy* berbiaya nol tanpa mengubah nilai optimalnya.

Kasus terbagi menjadi tiga kategori yang saling melengkapi dari sisi asal-usul dan tujuan pengujiannya. Kategori pertama adalah kasus dari literatur sebanyak 32 *instances* dengan label n01 hingga n35 dan ukuran bervariasi dari 3×3 hingga 5×5 . Kasus-kasus inilah yang menjadi tolok ukur historis metode IBFS, sebab masing-masing pernah dipakai penelitian terdahulu sehingga hasil RVM dapat ditempatkan pada konteks yang sudah dikenal komunitas riset. Kategori kedua adalah kasus sintetik sebanyak 5 *instances* dengan label S1 hingga S5 berukuran 5×5 hingga 11×12 yang dirancang dengan karakteristik biaya ekstrem untuk menguji ketahanan metode pada situasi yang jarang muncul di kasus literatur berukuran kecil. Kategori ketiga adalah *real-world* sebanyak 5 *instances* dengan label R1 hingga R5 berukuran 2×94 yang berasal dari data distribusi nyata sebuah perusahaan, sehingga menguji perilaku metode pada matriks berdimensi sangat tidak simetris (dua sumber, sembilan puluh empat tujuan) yang tidak terwakili oleh kasus literatur maupun sintetik.

Setiap kasus literatur diambil satu *instance* dari tiap sumber, bukan seluruh *instance* yang tersedia pada sumber tersebut. Alasannya, banyak publikasi IBFS menyajikan beberapa kasus uji yang saling berukuran serupa untuk mengilustrasikan langkah metodenya, sehingga mema-

sukkan semuanya hanya akan menggandakan kasus berukuran sama tanpa menambah keragaman struktur biaya yang diuji. Sebagai contoh, kasus 3×3 n01 diambil dari Ramadan dan Ramadan (2012) karena *instance* tersebut telah menjadi contoh 3×3 yang berulang dipakai pada studi pembandingan sesudahnya, sehingga representatif sebagai wakil ukuran itu. Prinsip yang sama berlaku untuk sumber lain: yang dipilih adalah *instance* yang paling sering dirujuk atau yang struktur biayanya paling mewakili karakteristik ukuran matriksnya, bukan sekadar *instance* pertama yang muncul. Tabel 3.21 merangkum sebaran ukuran matriks pada seluruh 42 kasus untuk memperlihatkan cakupan dimensi yang diuji.

Tabel 3.21 Sebaran Ukuran Matriks pada 42 Kasus Uji

Ukuran	Jumlah	Kategori	Kasus
2×94	5	<i>Real-world</i>	R1–R5
3×3	6	Literatur	n01, n12, n13, n21, n30, n33
3×4	16	Literatur	n02, n07, n10, n11, n14–n18, dst.
3×5	1	Literatur	n04
4×4	4	Literatur	n05, n06, n09, n28
4×5	3	Literatur	n26, n32, n34
5×5	4	Literatur/Sintetik	n03, n27, S1, S2
11×12	3	Sintetik	S3, S4, S5
Total	42		

Sebaran pada Tabel 3.21 memperlihatkan bahwa kasus 3×4 mendominasi (16 dari 42) karena ukuran ini paling lazim dipakai literatur IBFS sebagai contoh baku, sedangkan kasus berukuran besar (5×5 ke atas) sengaja dilengkapi lewat kategori sintetik dan *real-world* agar pengujian tidak terbatas pada matriks kecil. Justru pada rentang ukuran besar inilah perbedaan kualitas antarmetode paling menonjol, sebagaimana ditunjukkan analisis FNOCS pada Subbab 4.1.3.

Tabel 3.22 Sumber, Ukuran, Nilai Optimal, dan Peran Seluruh Kasus Uji

Instance	Ukuran	Z^*	Sumber	Keterangan
n01	3×3	5600	(Ramadan & Ramadan, 2012)	Matriks biaya 3×3 dengan rentang nilai lebar (32–200)
n02	3×4	28	(Samuel, 2012)	Matriks biaya 3×4 skala kecil dengan entri <i>twin cost</i> dominan (0–4)
n03	5×5	1102	(Karagül & Şahin, 2020)	Matriks biaya 5×5 dengan sebaran biaya renggang (8–96)
n04	3×5	1580	(Kaur et al., 2018)	Matriks biaya 3×5 dengan kepadatan <i>twin cost</i> tinggi (3–8)
n05	4×4	410	(Hussein & Shiker, 2020)	Matriks biaya 4×4 dengan <i>twin-cost ratio</i> dominan (2–11)

Tabel 3.22 – *lanjutan*

Instance	Ukuran	Z^*	Sumber	Keterangan
n06	4×4	7750	(Wireko et al., 2025)	Matriks biaya 4×4 dengan variasi biaya rendah-sedang (0–25)
n07	3×4	1160	(Das et al., 2014)	Matriks biaya 3×4 tanpa entri <i>twin cost</i> sama sekali
n09	4×4	111	(Murugesan & Esakkiammal, 2020)	Matriks biaya 4×4 dengan rentang biaya sempit (1–8)
n10	3×4	910	(Amaliah et al., 2019)	Matriks biaya 3×4 dengan biaya minimum berulang
n11	3×4	2460	(Amaliah et al., 2019)	Matriks biaya 3×4 dengan biaya unik di setiap sel (2–40)
n12	3×3	425	(Amaliah et al., 2019)	Matriks biaya 3×3 dengan biaya unik di setiap sel (7–25)
n13	3×3	4525	(Juman & Hoque, 2015)	Matriks biaya 3×3 dengan kapasitas supply besar (600)
n14	3×4	920	(Juman & Hoque, 2015)	Matriks biaya 3×4 dengan <i>twin cost</i> padat (4–8)
n15	3×4	156	(Ahmed et al., 2016)	Matriks biaya 3×4 dengan variasi biaya sangat sempit (1–9)
n16	3×4	1510	(Amreen, 2024)	Matriks biaya 3×4 dengan sebaran biaya merata (5–25)
n17	3×4	743	(Gupta, 2020)	Matriks biaya 3×4 dengan sebaran biaya asimetris (8–70)
n18	3×4	1650	(Goyal, 1984)	Matriks biaya 3×4 dengan entri biaya nol (0–21)
n21	3×3	4205	(Amreen, 2024)	Matriks biaya 3×3 dengan rentang biaya tinggi (71–260)
n22	3×4	12075	(Rafi & Islam, 2020)	Matriks biaya 3×4 dengan volume supply besar (950)
n23	3×4	23	(Kaur et al., 2018)	Matriks biaya 3×4 dengan kapasitas supply sangat kecil (6)
n24	3×4	3460	(Juman & Hoque, 2015)	Matriks biaya 3×4 dengan sebaran biaya renggang

Tabel 3.22 – *lanjutan*

Instance	Ukuran	Z^*	Sumber	Keterangan
n25	3×4	809	(Juman & Hoque, 2015)	Matriks biaya 3×4 dengan biaya minimum sama dengan n17
n26	4×5	490	(Patel et al., 2017)	Matriks biaya 4×5 dengan sebaran biaya menengah
n27	5×5	59356	(Korukoğlu & Balli, 2011)	Matriks biaya 5×5 dengan rentang biaya lebar (4–199)
n28	4×4	2090	(Reza et al., 2019)	Matriks biaya 4×4 dengan supply melimpah (1340)
n29	3×4	3857	(Kousalya & Malarvizhi, 2016)	Matriks biaya 3×4 dengan sebaran biaya lebar (1–90)
n30	3×3	465	(Ahmed et al., 2015)	Matriks biaya 3×3 dengan rentang biaya sempit (8–21)
n31	3×4	1480	(Kirca & Şatir, 1990)	Matriks biaya 3×4 dengan biaya relatif merata
n32	4×5	1780	(Babu et al., 2020)	Matriks biaya 4×5 dengan tingkat <i>twin cost</i> tinggi
n33	3×3	1250	(Amreen & Venkateswarlu, 2024)	Matriks biaya 3×3 dengan <i>twin cost</i> dominan (3–10)
n34	4×5	510	(Sahito, 2021)	Matriks biaya 4×5 dengan biaya hampir seluruhnya kembar (4–9)
n35	3×4	280	(Amaliah et al., 2022b)	Matriks biaya 3×4 dengan rentang biaya rendah (1–15)
S1	5×5	18855	Sintetik	Matriks sintetis 5×5 dengan rentang biaya asimetris (9–36)
S2	5×5	153500	Sintetik	Matriks sintetis 5×5 dengan rentang biaya asimetris (20–110)
S3	11×12	1044180	Sintetik	Matriks sintetis 11×12 dengan variabilitas biaya tinggi (3–546)
S4	11×12	92212	Sintetik	Matriks sintetis 11×12 dengan <i>twin cost</i> padat (10–99)
S5	11×12	116020	Sintetik	Matriks sintetis 11×12 dengan variabilitas biaya tinggi (1–182)

Tabel 3.22 – *lanjutan*

Instance	Ukuran	Z^*	Sumber	Keterangan
R1	2×94	12928429	Perusahaan XYZ	Kasus alokasi logistik riil dengan matriks asimetris 2×94
R2	2×94	13657053	Perusahaan XYZ	Kasus alokasi logistik riil dengan matriks asimetris 2×94
R3	2×94	12296587	Perusahaan XYZ	Kasus alokasi logistik riil dengan matriks asimetris 2×94
R4	2×94	14117586	Perusahaan XYZ	Kasus alokasi logistik riil dengan matriks asimetris 2×94
R5	2×94	14017419	Perusahaan XYZ	Kasus alokasi logistik riil dengan matriks asimetris 2×94

3.2.1.1 Format Instance

Setiap *instance* disimpan dalam format matriks transportasi yang terdiri atas matriks biaya, vektor *supply*, dan vektor *demand*. Tabel 3.23 menunjukkan contoh *instance* berukuran 3×3 , sedangkan Tabel 3.24 menunjukkan contoh *instance* berukuran 3×4 . Kasus n11 yang dipakai sebagai contoh penyelesaian lengkap langkah demi langkah pada Subbab 3.1.11 ditunjukkan pada Tabel 3.9.

Tabel 3.23 Instance n01 (3×3), $Z^* = 5600$

	D_1	D_2	D_3	Supply
S_1	32	40	120	20
S_2	60	68	104	30
S_3	200	80	60	45
Demand	30	35	30	

Tabel 3.24 Instance n02 (3×4), $Z^* = 28$

	D_1	D_2	D_3	D_4	Supply
S_1	1	2	3	4	6
S_2	4	3	2	0	8
S_3	0	2	2	1	10
Demand	4	6	8	6	

Sisa 39 *instance* lainnya (29 kasus literatur selain n01, n02, dan n11; 5 kasus sintetik S1–S5; serta 5 *real-world* R1–R5) ditampilkan lengkap pada Lampiran C, agar seluruh data masukan *benchmark* dapat ditelusuri dan direproduksi tanpa merujuk berkas eksternal, tanpa membebani badan teks dengan puluhan tabel matriks.

3.2.1.2 Ground Truth Solusi Optimal

Setiap *instance* memiliki *ground truth* berupa nilai solusi optimal Z^* yang digunakan sebagai acuan evaluasi. Nilai-nilai tersebut diperoleh dengan menyelesaikan setiap kasus sebagai program linear hingga optimal. Tabel 3.25 menyajikan *ground truth* seluruh 42 kasus uji.

Tabel 3.25 Ground Truth Solusi Optimal Seluruh Kasus Uji

Case	Optimal	Case	Optimal	Case	Optimal
n01	5600	n16	1510	n32	1780
n02	28	n17	743	n33	1250
n03	1102	n18	1650	n34	510
n04	1580	n21	4205	n35	280
n05	410	n22	12075	S1	18855
n06	7750	n23	23	S2	153500
n07	1160	n24	3460	S3	1044180
n09	111	n25	809	S4	92212
n10	910	n26	490	S5	116020
n11	2460	n27	59356	R1	12928429
n12	425	n28	2090	R2	13657053
n13	4525	n29	3857	R3	12296587
n14	920	n30	465	R4	14117586
n15	156	n31	1480	R5	14017419

3.2.2 Dataset Uji Tambahan (Dataset Uji 2 dan Dataset Uji 3)

Selain 42 kasus uji utama (selanjutnya disebut Dataset Uji 1) pada Subbab 3.2.1, penelitian ini melengkapi pengujian dengan dua himpunan data tambahan yang **sengaja dipisahkan** dari Dataset Uji 1 dan tidak digabung ke dalam metrik AP/DP/PC/IP/FNOCS/OI pada Bab IV bagian utama. Pemisahan ini disengaja karena ketiga himpunan data punya tujuan pengujian yang berbeda: Dataset Uji 1 menguji RVM pada kasus yang telah mapan dipakai literatur dan studi pembandingan (Bunjamin dkk., 2026a, 2026b), sedangkan Dataset Uji 2 dan Dataset Uji 3 menguji ketahanan RVM pada variasi struktur biaya dan skala matriks yang lebih luas, termasuk terhadap metode pembandingan kedelapan yang tidak diikuti pada Dataset Uji 1.

3.2.2.1 Dataset Uji 2 dengan Variasi Ukuran Kasus

Dataset Uji 2 terdiri atas 27 *instance* tambahan berukuran 3×2 hingga 30×32 , jauh lebih beragam dibanding rentang ukuran Dataset Uji 1 (3×3 hingga 2×94 untuk *real-world*, namun didominasi ukuran kecil-menengah). Dataset Uji 2 disusun untuk menjembatani kekosongan ukuran menengah-besar (di atas 11×12) yang tidak terwakili Dataset Uji 1 selain oleh *real-world* R1–R5 yang justru berbentuk sangat tidak simetris (2×94).

3.2.2.2 Dataset Uji 3 dengan Matriks Sintetis Skala Besar

Dataset Uji 3 terdiri atas 640 *instance* sintetis berukuran 10×20 hingga 10×100 , terbagi ke dalam 64 kelompok parameter berlabel G10 sampai G73 (masing-masing 10 replikasi) yang memvariasikan rentang biaya R (selisih biaya maksimum-minimum, dari 20 hingga 1000) dan tingkat variasi *supply-demand* V (10%, 25%, 50%, atau 75% menyimpang dari rata-rata). Keenam

puluh empat kelompok adalah hasil kombinasi lengkap 4 ukuran matriks (10×20 , 10×40 , 10×60 , 10×100), 4 nilai R (20, 100, 500, 1000), dan 4 nilai V (10%, 25%, 50%, 75%), sehingga $4 \times 4 \times 4 = 64$ kelompok.

Data Dataset Uji 3 dibangkitkan dengan skrip Python yang disertakan sebagai berkas pendamping penelitian. Biaya tiap sel mengikuti distribusi seragam pada interval $[500 - R/2, 500 + R/2]$, sehingga R menentukan langsung lebar ruang nilai integer yang mungkin muncul. Pada kelompok dengan $R = 20$, seluruh biaya berada pada interval $[490, 510]$, hanya 21 nilai bulat berbeda untuk mengisi $10 \times 20 = 200$ sel, sehingga tumbukan nilai (*twin cost*) hampir pasti terjadi semata karena keterbatasan ruang nilai tersebut. Mekanisme yang sama berlaku pada 64 kelompok lainnya dengan lebar interval berbeda, dan inilah yang mendasari tingginya *twin-cost ratio* Dataset Uji 3 pada Subbab 3.2.3.

Supply dan *demand* dibangkitkan seragam pada rentang $\pm V\%$ dari rata-rata (100 unit per tujuan), lalu diseimbangkan (*balanced*) melalui prosedur perbaikan iteratif: karena *supply* dan *demand* awalnya dibangkitkan independen sehingga totalnya hampir pasti tidak sama, skrip menaikkan atau menurunkan nilai *demand* pada elemen terpilih acak sebesar satu satuan berulang kali hingga totalnya tepat sama dengan total *supply*, tanpa melanggar batas rentang V yang ditetapkan. Nilai optimal Z^* tiap *instance* diperoleh melalui pemrograman linear memakai pustaka *scipy* (fungsi *linprog*, metode *highs*), konsisten dengan prosedur *ground truth* Dataset Uji 1 (Subbab 3.2), disertai validasi bahwa nilai optimal yang diperoleh benar-benar bilangan bulat sesuai sifat unimodularitas TP *balanced*. Seluruh proses menggunakan *seed* acak tetap sehingga data dapat dibangkitkan ulang secara identik untuk keperluan reproduksi.

Tujuan Dataset Uji 3 adalah menguji ketahanan RVM pada skala matriks yang jauh lebih besar dari Dataset Uji 1 dan pada kepadatan *twin cost* (*tie*) yang sistematis tinggi, sebagaimana dijelaskan pada Subbab 3.2.3.

3.2.2.3 Format Data dan Verifikasi

Berbeda dari Dataset Uji 1 yang *ground truth*-nya disimpan terpisah pada *ground-truth.txt* (Subbab 3.2), setiap *instance* Dataset Uji 2 dan Dataset Uji 3 menyimpan nilai optimalnya langsung sebagai baris terakhir pada berkas yang sama, mengikuti format berkas asal data tersebut. Seluruh *instance* Dataset Uji 2 dan Dataset Uji 3 diverifikasi dapat dibaca benar oleh kesembilan metode (kedelapan metode pembandingan dan RVM) sebelum pengujian utama dijalankan; ditemukan dan diperbaiki dua kelas persoalan teknis pada tahap verifikasi ini. Pertama, sebagian berkas Dataset Uji 3 (397 dari 640) memuat penanda *byte order mark* (BOM) UTF-8 di awal berkas yang menggagalkan pembacaan *fscanf* pada seluruh implementasi C/C++, diperbaiki dengan menghapus BOM tanpa mengubah data numerik apa pun di dalamnya. Kedua, ditemukan cacat logika pada implementasi BCE, CSM, dan RBSM yang tidak pernah teramati pada Dataset Uji 1 karena kondisinya jarang terjadi pada matriks kecil: BCE mengalami *segmentation fault* akibat variabel penunjuk sel tidak diinisialisasi ketika tidak ada lagi baris *unmet* berkandidat valid, sedangkan CSM dan RBSM keliru menganggap kondisi seluruh baris langsung *satisfied* sejak alokasi awal (kasus paling ideal, bukan galat) sebagai masukan tidak valid. Ketiganya diperbaiki dengan penambahan penanganan kasus tepi minimal tanpa mengubah logika alokasi inti algoritmanya, dan perbaikan ini diverifikasi tidak mengubah satu pun hasil pada 42 kasus Dataset Uji 1.

3.2.3 Karakteristik Struktur Biaya Dataset Uji 2 dan Dataset Uji 3

Untuk memahami mengapa performa RVM berbeda antara Dataset Uji 1 dan data tambahan (Subbab 4.3), struktur biaya ketiga himpunan data dibandingkan melalui *twin-cost ratio*.

Konsep *twin cost* merujuk pada keberadaan entri biaya duplikat (nilai biaya yang sama) di dalam satu matriks biaya. Nilai biaya yang identik ini merupakan pemicu utama munculnya situasi seri (*tie*) saat menghitung penalti lini atau memilih sel berbiaya tereduksi nol. Pada matriks biaya dengan *twin-cost ratio* yang tinggi, RVM lebih sering mengaktifkan mekanisme *tie-breaking* (Subbab 3.1.5) untuk menentukan prioritas alokasi, alih-alih mengambil keputusan murni berdasarkan perbedaan biaya antarsel.

Secara formal, misalkan C adalah matriks biaya berukuran $m \times n$. Himpunan koordinat sel (i, j) yang memiliki nilai biaya berulang (kembar) pada matriks didefinisikan sebagai:

$$S_{\text{kembar}} = \{(i, j) \mid \exists (i', j') \neq (i, j) \text{ sedemikian sehingga } C_{i,j} = C_{i',j'}\} \quad (3.5)$$

twin-cost ratio (r_{kembar}) dihitung sebagai proporsi jumlah sel yang memiliki entri biaya berulang terhadap total sel dalam matriks:

$$r_{\text{kembar}} = \frac{|S_{\text{kembar}}|}{m \times n} \quad (3.6)$$

Sebagai ilustrasi, tinjau matriks biaya berukuran 3×3 berikut:

$$C = \begin{pmatrix} 10 & 20 & 30 \\ 40 & 10 & 50 \\ 60 & 70 & 20 \end{pmatrix} \quad (3.7)$$

Terdapat dua nilai biaya yang berulang, yaitu angka 10 (muncul dua kali) dan angka 20 (muncul dua kali). Sel-sel yang memuat biaya berulang tersebut adalah sel $(1, 1)$, $(2, 2)$, $(1, 2)$, dan $(3, 3)$, sehingga total terdapat 4 sel yang memiliki *twin cost*. Dengan demikian, *twin-cost ratio* matriks tersebut adalah $4/9 \approx 44,4\%$.

Rasio ini dipilih karena keputusan *tie-breaking* (Subbab 3.1.5) hanya aktif ketika terdapat entri biaya yang identik antarlani; makin tinggi *twin-cost ratio*, makin sering RVM bergantung pada aturan *tie-breaking* struktural alih-alih pada perbedaan biaya itu sendiri untuk menetapkan prioritas alokasi.

Tabel 3.26 merangkum *twin-cost ratio* rata-rata ketiga himpunan data.

Tabel 3.26 Rata-Rata *Twin-Cost Ratio* per Himpunan Data

Himpunan	Jumlah Kasus	<i>twin-cost ratio</i>
Dataset Uji 1 (literatur + sintetik + <i>real-world</i>)	42	52,3%
Dataset Uji 2 (kasus tambahan beragam ukuran)	27	62,4%
Dataset Uji 3 (sintetis skala besar)	640	98,6%

Dataset Uji 3 memiliki *twin-cost ratio* hampir dua kali lipat Dataset Uji 1, konsekuensi langsung dari mekanisme pembangkitannya (Subbab 3.2.2): biaya dibangkitkan seragam pada rentang R yang pada kelompok tertentu jauh lebih sempit dibanding jumlah sel matriks, sehingga banyak nilai bertumbukan semata karena keterbatasan ruang nilai integer yang tersedia, bukan karena struktur bisnis yang sesungguhnya seperti pada kasus literatur atau *real-world* Dataset

Uji 1. Dataset Uji 2 berada di antara keduanya, sejalan dengan variasi ukuran dan sumbernya yang lebih beragam dibanding Dataset Uji 3 yang seluruhnya sintetis seragam.

Pemeriksaan lebih lanjut pada 64 kelompok parameter Dataset Uji 3 menunjukkan bahwa jumlah replikasi RVM mencapai optimal per kelompok (dari 10 replikasi) tidak berkorelasi kuat dengan rentang biaya R (rata-rata 6,16 dari 10 pada kelompok $R < 250$ berbanding 6,31 dari 10 pada kelompok $R \geq 250$) maupun dengan tingkat variasi *supply-demand* V (6,12 berbanding 6,34 untuk $V < 0,3$ dan $V \geq 0,3$) maupun lebar matriks n (relatif stabil pada kisaran 6,1–6,4 dari $n = 20$ hingga $n = 100$). Ketidadaan korelasi ini justru memperkuat kesimpulan bahwa penurunan performa RVM pada Dataset Uji 3 bersumber dari kepadatan *twin cost* yang sistematis di seluruh himpunan data, bukan dari kombinasi parameter tertentu.

3.2.4 Metrik Evaluasi

Hasil dari setiap metode penyelesaian TP akan dievaluasi untuk memperoleh kesimpulan mengenai kinerja masing-masing metode. Evaluasi dilakukan dengan menghitung *Accuracy Percentage* (AP), *Deviation Percentage* (DP), *Percentage of Correctness* (PC), *Improvement Percentage* (IP), *Frequency of the Number of the Optimal Cost Solutions* (FNOCS), serta *Optimality Iterations* (OI). Uraian mengenai masing-masing ukuran kinerja dijelaskan sebagai berikut.

3.2.4.1 Accuracy Percentage

Accuracy Percentage (AP) merupakan ukuran efektivitas metode berdasarkan jumlah data uji yang berhasil mencapai solusi optimal dibandingkan seluruh data uji (Amaliah dkk., 2019). Nilai AP dihitung menggunakan

$$AP = \frac{\text{Jumlah data uji yang mencapai solusi optimal}}{\text{Jumlah seluruh data uji}} \times 100\%. \quad (3.8)$$

Metrik ini menjawab pertanyaan paling mendasar tentang sebuah metode IBFS, yaitu seberapa sering ia langsung menghasilkan solusi terbaik tanpa memerlukan optimisasi lanjutan. AP bersifat biner pada tiap kasus, artinya sebuah *instance* hanya dihitung apabila biaya IBFS-nya sama persis dengan biaya optimal, sehingga metrik ini menjadi ukuran yang tegas dan mudah dibandingkan lintas penelitian. Kelemahannya, AP memperlakukan seluruh kegagalan secara setara tanpa membedakan seberapa jauh sebuah solusi meleset, sehingga perlu dilengkapi oleh metrik deviasi berikutnya.

3.2.4.2 Deviation Percentage

Deviation Percentage (DP) merupakan ukuran penyimpangan nilai IBFS suatu metode terhadap solusi optimal yang diketahui (Juman & Hoque, 2015). Nilai DP dihitung menggunakan

$$DP = \frac{(\text{IBFS}) - (\text{Solusi Optimal})}{\text{Solusi Optimal}} \times 100\%. \quad (3.9)$$

Berbeda dari AP yang bersifat biner, DP menangkap besarnya penyimpangan secara kontinu, sehingga dua metode yang sama-sama gagal mencapai optimal tetap dapat dibedakan berdasarkan seberapa dekat masing-masing solusinya terhadap biaya minimum. Metrik ini penting karena dalam praktik, IBFS yang meleset tipis jauh lebih bernilai daripada yang meleset jauh, mengingat keduanya sama-sama akan disempurnakan oleh fase optimisasi lanjutan namun dengan be-

ban yang berbeda. Statistik deskriptif DP berupa rata-rata dan maksimum keduanya dilaporkan, karena rata-rata menggambarkan kualitas tipikal sedangkan maksimum mengungkap kasus terburuk yang mungkin dihadapi suatu metode. Solusi optimal Z^* diperoleh melalui pemrograman linear, bukan dari klaim publikasi sumber (Subbab 3.2.6), agar acuan deviasi seragam untuk seluruh metode.

3.2.4.3 Percentage of Correctness

Percentage of Correctness (PC) merupakan ukuran yang digunakan untuk menilai tingkat kedekatan solusi awal yang dihasilkan suatu metode terhadap solusi optimal (Wireko dkk., 2025). Ukuran ini diperoleh dari selisih antara 100% dengan DP. Secara matematis, *Percentage of Correctness* dirumuskan sebagai berikut.

$$PC = 100\% - DP. \quad (3.10)$$

Secara informasi, PC dan DP sepenuhnya setara karena satu adalah komplemen langsung dari yang lain, namun PC menyajikannya dalam arah yang searah dengan AP, yaitu semakin tinggi nilainya semakin baik kualitas metode. Penyajian searah ini memudahkan pembacaan tabel komparatif yang memuat AP dan DP sekaligus, sebab pembaca tidak perlu membalik interpretasi ketika berpindah antarkolom. PC disertakan mengikuti praktik pelaporan pada literatur IBFS terbaru agar hasil penelitian ini dapat dibandingkan langsung dengan angka yang dilaporkan studi sebelumnya.

3.2.4.4 Improvement Percentage

Improvement Percentage (IP) merupakan ukuran peningkatan kualitas solusi IBFS metode yang diusulkan dibandingkan dengan metode pembanding, dihitung secara per-pasangan untuk setiap metode pembanding pada setiap kasus uji (Amaliah dkk., 2022a, 2022b). Perhitungan IP mengacu pada

$$IP = \frac{(\text{IBFS}_{\text{Pembanding}}) - (\text{IBFS}_{\text{RVM}})}{\text{IBFS}_{\text{Pembanding}}} \times 100\%. \quad (3.11)$$

Nilai IP dapat positif (RVM lebih murah), negatif (RVM lebih mahal), atau nol (seri), dan dilaporkan sebagai rata-rata serta jumlah kasus positif/negatif/nol untuk setiap pasangan RVM–metode pembanding.

3.2.4.5 Frequency of the Number of the Optimal Cost Solutions

Frequency of the Number of the Optimal Cost Solutions (FNOCS) menunjukkan banyaknya data uji yang berhasil mencapai biaya optimal untuk tiap ukuran matriks (Md. A. Babu dkk., 2020). Berbeda dari AP yang menyajikan satu angka agregat untuk seluruh kasus, FNOCS memecah capaian optimal menurut dimensi matriks, sehingga dapat terlihat apakah keunggulan suatu metode merata di semua ukuran atau justru terpusat pada dimensi tertentu saja. Rincian per ukuran ini penting karena sebuah metode yang unggul secara agregat masih mungkin lemah pada subset ukuran tertentu, dan kelemahan seperti itu akan tersembunyi jika hanya dilihat dari AP keseluruhan. Tabel 4.7 menyajikan FNOCS, yang memuat ukuran matriks, jumlah data uji, serta jumlah hasil implementasi dari setiap metode yang mencapai solusi optimal.

3.2.4.6 Optimality Iterations

Optimality Iterations (OI) merupakan ukuran yang digunakan untuk mengetahui jumlah iterasi yang diperlukan suatu metode setelah IBFS diperoleh hingga mencapai solusi optimal menggunakan prosedur *Modified Distribution* (MODI)/*stepping stone* (Charnes & Cooper, 1954; Dantzig, 1963; Taha, 2017). Jumlah iterasi digunakan sebagai indikator efisiensi metode: semakin sedikit jumlah iterasi yang dibutuhkan, semakin baik kualitas solusi awal yang dihasilkan karena solusi tersebut sudah berada dekat dengan solusi optimal. OI dihitung dengan menjalankan MODI dari basis IBFS setiap metode (dilengkapi sel basis bernilai nol bila terjadi degenerasi, Subbab 3.1) hingga seluruh *reduced cost* non-basis taknegatif, dan mencacah banyaknya pivot yang terjadi; IBFS yang sudah optimal sejak awal memiliki $OI = 0$.

Metode pembanding adalah tujuh metode IBFS terpublikasi, yaitu VAM, TOCM-MT, BCE, JHM-M, SSM, CSM, dan RBSM. Seluruh metode diimplementasi ulang dan dijalankan pada 42 *instance* yang sama, dan angka komparatif berasal dari hasil eksekusi pada penelitian ini, bukan dari nilai terpublikasi, sehingga seluruh metode diukur pada kondisi yang identik. Reproduksi VAM, BCE, dan CSM menghasilkan jumlah optimal yang sama persis dengan publikasi (22, 29, dan 33 dari 42), yang memvalidasi *harness* pengujian. Sebagai catatan keterbatasan, sebagian implementasi pembanding memuat batas iterasi dan penyesuaian aturan *tie-breaking* pada kode sumber yang tersedia, sehingga angka yang dilaporkan mencerminkan implementasi yang diuji, bukan klaim universal atas metode tersebut.

3.2.5 Metode Pembanding Tambahan IHW

Selain tujuh metode pembanding pada Dataset Uji 1, pengujian pada Dataset Uji 2 dan Dataset Uji 3 menyertakan *Improved Heuristic Weight* (IHW) sebagai metode pembanding kedelapan. IHW dipilih karena representatif terhadap garis pendekatan yang berbeda dari VAM dan turunannya: alih-alih penalti berbasis selisih dua biaya terkecil, IHW membentuk kandidat alokasi dari tiga komponen sekaligus untuk tiap baris/kolom aktif, yaitu *Minimum Cost* (MC, biaya terkecil pada lini tersebut), *Feasible Allocation* (FA, kuantitas maksimum yang bisa dialokasikan pada sel MC), dan *Inverse Contra-Harmonic Mean* (ICHM, ukuran sebaran biaya pada lini itu). Ketiganya digabung menjadi $Weight = MC \times FA \times ICHM$, dan pada tiap iterasi lini dengan *Weight* terkecil dipilih untuk dialokasikan; aturan *tie-breaking* berjenjang: *Weight* sama, lalu MC lebih kecil, lalu FA lebih kecil, lalu baris diprioritaskan atas kolom, lalu indeks terkecil.

Kode sumber IHW yang diperoleh berupa cuplikan lampiran dengan sejumlah bagian sengaja dipotong untuk kebutuhan penyajian (penyeimbangan *supply-demand* tak seimbang dan pencetakan hasil evaluasi tidak disertakan, dan program dibaca lewat masukan interaktif alih-alih argumen baris perintah). Bagian yang dipotong tersebut dilengkapi mengikuti pola implementasi metode pembanding lain pada penelitian ini (I/O berbasis argumen berkas, keluaran tercetak dalam format yang dapat diurai otomatis), sementara logika inti algoritma (perhitungan *Weight* dan aturan *tie-breaking*) dipertahankan tidak berubah dari cuplikan asli.

3.2.6 Desain Studi Ablasi

Studi ablasi dirancang untuk mengisolasi sumber keunggulan RVM, yaitu apakah berasal dari reduksi *zero-point* atau dari bentuk fungsi *penalty*. Kerangka algoritma dipertahankan identik (reduksi, pemilihan lini, alokasi, serta aturan *tie-breaking* pada Subbab 3.1.5), sementara hanya fungsi *penalty* yang divariasikan. Misalkan s_1, s_2 adalah dua nilai tereduksi terkecil pada lini, dengan varian berbeda mensyaratkan nilai terkecil kedua tidak sama dengan nilai terkecil

pertama (duplikat dilewati, dilambangkan dengan superskrip bintang pada Tabel 3.27); l_1, l_2 adalah dua nilai tereduksi terbesar dengan varian berbeda yang sepadan; v_i menyatakan nilai tereduksi tiap sel aktif, $\bar{v}$ menyatakan rata-ratanya, dan k menyatakan jumlah sel aktif. Pada formula rata-rata geometris dan harmonik, perhitungan dibatasi pada sel-sel bernilai tak nol saja agar tidak terbagi oleh nol, mengingat Proposisi 2.3 menjamin setiap lini memuat sedikitnya satu nilai nol. Lini dengan tepat satu sel aktif tetap mengikuti aturan umum: penaltinya adalah nilai sel tunggal tersebut. Formulasi lengkap beserta interpretasi matematis dari 15 varian fungsi *penalty* yang diuji disajikan pada Tabel 3.27; formula pertama (selisih dua nilai terkecil dengan duplikat dihitung) adalah *penalty* yang dipakai RVM sendiri.

Tabel 3.27 Definisi 15 Varian Fungsi *Penalty* yang Diuji

Nama Penalti	Fungsi Penalti	Interpretasi Matematis
Selisih Dua Cost Terkecil dengan Nilai Sama Tetap Dihitung	$s_2 - s_1$	Selisih dua nilai tereduksi terkecil pada lini, diambil langsung dari urutan terkecil (<i>twin cell</i> ikut dihitung sebagai s_1 dan s_2). Formula ini dipakai RVM sebagai <i>penalty</i> utama.
Median Cost - Cost Terendah	$\text{med}(v) - s_1$	Selisih nilai median dan nilai terkecil pada lini.
Rata-rata Selisih Cost terhadap Cost Terendah	$\bar{v} - s_1$	Rata-rata selisih tiap sel terhadap nilai terendah pada lini; pada matriks tereduksi $s_1 = 0$ sehingga setara $\bar{v}$.
Rata-rata Cost	$\bar{v}$	Rata-rata aritmatika seluruh nilai tereduksi pada lini.
Cost Tertinggi - Cost Terendah	$l_1 - s_1$	Selisih nilai terbesar dan terkecil (rentang) pada lini.
Total Selisih Cost terhadap Cost Terendah	$\sum_i (v_i - s_1)$	Jumlah seluruh selisih sel terhadap nilai terendah pada lini, tanpa perataan.
Contra Harmonic Mean	$\frac{\sum_i v_i^2}{\sum_i v_i}$	Rata-rata kontra-harmonik; memberi bobot lebih pada nilai besar.
Geometric Mean	$(\prod_{v_i \neq 0} v_i)^{1/k^*}$	Rata-rata geometris atas sel taknol pada lini ($k^* =$ banyaknya sel taknol; 0 bila seluruh sel nol).
Standar Deviasi Cost	$\sqrt{\frac{1}{k} \sum_i (v_i - \bar{v})^2}$	Standar deviasi (penyebaran) nilai tereduksi pada lini.
Harmonic Mean	$\frac{k^*}{\sum_{v_i \neq 0} 1/v_i}$	Rata-rata harmonik atas sel taknol pada lini; 0 bila seluruh sel nol.
Selisih Dua Cost Terkecil yang Berbeda	$s_2^* - s_1$	Selisih nilai terkecil dengan nilai berbeda terkecil berikutnya (duplikat dilewati); 0 bila seluruh sel bernilai sama.
Selisih Dua Cost Terbesar yang Berbeda	$l_1 - l_2^*$	Selisih nilai terbesar dengan nilai berbeda terbesar berikutnya (duplikat dilewati); 0 bila seluruh sel bernilai sama.
Selisih Dua Cost Terbesar dengan Nilai Sama Tetap Dihitung	$l_1 - l_2$	Selisih dua nilai tereduksi terbesar pada lini, <i>twin cell</i> ikut dihitung sebagai l_1 dan l_2 .
Inverse Harmonic Mean	$\frac{1}{k^*} \sum_{v_i \neq 0} \frac{1}{v_i}$	Rata-rata aritmatika dari kebalikan nilai taknol pada lini.
Inverse Contra-Harmonic Mean	$\frac{\sum_i v_i}{\sum_i v_i^2}$	Kebalikan dari rata-rata kontra-harmonik.

Apabila berbagai bentuk *penalty* menghasilkan kualitas yang berdekatan dan seluruhnya jauh melampaui VAM standar, maka dapat disimpulkan bahwa sumber utama keunggulan adalah reduksi *zero-point*, bukan bentuk *penalty* tertentu. Hasil lengkap studi ablasi disajikan pada Bab IV.

3.2.7 Perangkat Keras

Penelitian ini dilaksanakan menggunakan laptop pribadi dengan spesifikasi yang ditunjukkan pada Tabel 3.28.

Tabel 3.28 Spesifikasi Perangkat Keras

Komponen	Spesifikasi
Laptop	Acer Aspire A514-54
Processor	Intel Core i5-1135G7 @ 2.40GHz
RAM	24 GB DDR4
Sistem Operasi	Windows 11 Home 64-bit

3.2.8 Perangkat Lunak

Penelitian ini menggunakan beberapa perangkat lunak untuk implementasi dan analisis.

3.2.8.1 Compiler dan IDE

Implementasi RVM dilakukan dalam bahasa C (standar C11) dan dikompilasi menggunakan GCC (MinGW). Proses pengembangan dan *debugging* dilakukan menggunakan Visual Studio Code sebagai IDE utama. Verifikasi kebenaran dan analisis hasil dilakukan dengan Python pada Google Colab.

3.2.8.2 Python Libraries

Verifikasi dan analisis hasil memerlukan beberapa *library* Python. Tabel 3.29 menunjukkan *library* yang digunakan beserta fungsinya.

Tabel 3.29 Python Libraries

Library	Fungsi
numpy	Komputasi numerik dan operasi matriks
scipy	Penyelesaian program linear (<i>ground truth</i>)
pandas	Manipulasi dan analisis data hasil
matplotlib	Visualisasi hasil pengujian

3.2.8.3 C Standard Libraries

Implementasi RVM dalam bahasa C memanfaatkan beberapa *standard library*. Tabel 3.30 menunjukkan *library* yang digunakan beserta fungsinya.

Tabel 3.30 C Standard Libraries

Library	Fungsi
<stdio.h>	Operasi input dan output
<stdlib.h>	Alokasi memori dan <i>sorting</i>
<string.h>	Pemrosesan argumen dan string
<math.h>	Fungsi matematika

Halaman ini sengaja dikosongkan

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Penelitian

Bagian ini menyajikan hasil pengujian empiris kinerja algoritma *Reduced Vogel's Method* (RVM) pada *benchmark* 42 kasus uji utama (Dataset Uji 1), yang terdiri atas 32 kasus literatur (*n01–n35*), 5 kasus sintetis (*S1–S5*), dan 5 *real-world* (*R1–R5*). Pengujian dilakukan dengan membandingkan RVM secara penuh terhadap tujuh metode IBFS terpublikasi (VAM, TOCM-MT, BCE, JHM-M, SSM, CSM, dan RBSM) untuk mengukur kualitas solusi awal, efisiensi iterasi optimisasi, serta ketahanan algoritma lintas dimensi matriks dan fungsi penalti.

4.1.1 Rincian Hasil Biaya Solusi per Kasus

Tabel 4.1 menyajikan data mentah biaya solusi seluruh delapan metode pada setiap kasus uji dari 42 kasus *benchmark* (kasus literatur *n01–n35* serta kasus sintetis dan *real-world* *S1–R5*). Dasar pemilihan ke-42 matriks kasus uji ini merujuk pada standar pengujian literatur terkini (sebagaimana telah dirinci pada Tabel 3.22), yang mana kasus-kasus tersebut sengaja dipilih karena mencakup variasi ukuran, tingkat degenerasi (kepadatan *twin cost*), dan rentang biaya ekstrem untuk menguji ketahanan (*robustness*) setiap algoritma secara komprehensif. Kolom **Optimal** menunjukkan acuan *ground truth* yang diverifikasi melalui pemrograman linear. Nilai yang dicetak tebal pada kolom metode mana pun menandakan bahwa metode tersebut berhasil mencapai biaya optimal pada kasus tersebut.

Tabel 4.1 Biaya Solusi Seluruh Metode pada 42 Kasus Uji

Kasus	Optimal	VAM	TOCM-MT	BCE	JHM-M	SSM	CSM	RBSM	RVM
n01	5.600	5.600	5.600	5.600	5.600	5.600	5.600	5.600	5.600
n02	28	28	28	28	28	28	28	28	28
n03	1.102	1.104	1.127	1.137	1.134	1.134	1.134	1.127	1.102
n04	1.580	1.580	1.580	1.580	1.625	1.625	1.580	1.625	1.580
n05	410	470	435	410	430	430	430	430	415
n06	7.750	8.350	7.750	8.350	7.750	7.750	7.750	7.750	7.750
n07	1.160	1.220	1.160	1.170	1.165	1.165	1.160	1.160	1.160
n09	111	114	111	111	111	111	111	111	111
n10	910	990	910	910	960	960	910	910	990
n11	2.460	2.980	2.500	2.460	2.500	2.500	2.460	2.460	2.460
n12	425	425	439	439	425	425	439	425	425
n13	4.525	5.125	5.225	4.525	4.525	4.525	4.525	4.525	4.525
n14	920	960	930	920	920	920	920	920	920
n15	156	156	156	156	156	156	156	156	156
n16	1.510	1.590	1.530	1.530	1.510	1.510	1.510	1.590	1.510
n17	743	779	743	743	743	743	779	779	743
n18	1.650	1.745	1.650	1.695	1.650	1.650	1.650	1.650	1.650
n21	4.205	4.205	4.205	4.429	4.205	4.205	4.429	4.205	4.205
n22	12.075	12.075	12.200	12.700	12.075	12.075	12.075	12.075	12.075
n23	23	23	23	23	23	23	23	23	23
n24	3.460	3.520	3.460	3.460	3.460	3.460	3.460	3.460	3.460
n25	809	849	809	809	809	809	849	849	809
n26	490	490	508	490	490	490	490	490	490
n27	59.356	68.804	59.356	64.499	59.356	59.356	59.356	59.356	59.356
n28	2.090	2.690	2.090	2.090	2.090	2.090	2.090	2.090	2.090
n29	3.857	3.857	3.857	3.857	3.857	3.857	3.857	3.857	3.857
n30	465	473	475	475	465	465	475	465	465
n31	1.480	1.480	1.480	1.480	1.480	1.480	1.480	1.480	1.480

Tabel 4.1 – lanjutan

Kasus	Optimal	VAM	TOCM-MT	BCE	JHM-M	SSM	CSM	RBSM	RVM
n32	1.780	1.780	1.780	1.780	1.820	1.820	1.780	1.780	1.780
n33	1.250	1.430	1.250	1.310	1.310	1.310	1.310	1.310	1.310
n34	510	510	510	510	510	510	510	510	510
n35	280	320	280	280	280	280	280	280	280
S1	18.855	18.855	18.855	19.035	20.925	20.925	18.855	18.855	18.855
S2	153.500	153.500	153.500	153.500	176.500	176.500	153.500	153.500	153.500
S3	1.044.180	1.044.180	1.044.180	1.044.180	1.044.180	1.044.180	1.044.180	1.044.180	1.044.180
S4	92.212	93.351	92.431	94.264	95.301	95.301	95.440	95.440	94.060
S5	116.020	116.020	116.020	116.020	116.020	116.020	116.020	116.020	116.020
R1	12.928.429	12.928.429	12.965.785	12.928.429	12.928.429	12.928.429	12.928.429	12.928.429	12.928.429
R2	13.657.053	13.657.053	13.700.907	13.657.053	13.657.053	13.657.053	13.657.053	13.657.053	13.657.053
R3	12.296.587	12.296.587	12.322.885	12.296.587	12.296.587	12.296.587	12.296.587	12.296.587	12.296.587
R4	14.117.586	14.117.586	14.166.940	14.117.586	14.117.586	14.117.586	14.117.586	14.117.586	14.117.586
R5	14.017.419	14.017.419	14.074.887	14.017.419	14.017.419	14.017.419	14.017.419	14.017.419	14.017.419

Tabel 4.1 memperlihatkan bahwa kegagalan tiap metode tersebar pada kasus yang berbeda-beda, bukan terkonsentrasi pada subset kasus yang sama; TOCM-MT misalnya gagal optimal pada n03, n22, dan n27, sedangkan RBSM justru optimal pada ketiganya tetapi gagal pada n16, n17, dan n25. Pola saling silang ini konsisten dengan *No Free Lunch Theorem* dan menjadi dasar studi kasus pada Subbab 4.2.1.

4.1.2 Hasil RVM dan Perbandingan Agregat dengan Metode Terpublikasi

Tabel 4.2 merangkum hasil khusus RVM pada 42 kasus uji beserta deviasinya terhadap solusi optimal. RVM mencapai solusi optimal pada 38 dari 42 kasus, dengan deviasi rata-rata 0,40%, median 0,00%, simpangan baku 1,55%, dan deviasi maksimum 8,79%. Median deviasi yang bernilai nol menunjukkan bahwa pada lebih dari separuh kasus, RVM langsung menghasilkan solusi optimal tanpa memerlukan iterasi optimisasi lanjutan.

Tabel 4.2 Hasil RVM pada 42 Kasus Uji

Kasus	Ukuran	Optimal	RVM	Deviasi (%)
n01	3×3	5600	5600	0,00
n02	3×4	28	28	0,00
n03	5×5	1102	1102	0,00
n04	3×5	1580	1580	0,00
n05	4×4	410	415	1,22
n06	4×4	7750	7750	0,00
n07	3×4	1160	1160	0,00
n09	4×4	111	111	0,00
n10	3×4	910	990	8,79
n11	3×4	2460	2460	0,00
n12	3×3	425	425	0,00
n13	3×3	4525	4525	0,00
n14	3×4	920	920	0,00
n15	3×4	156	156	0,00
n16	3×4	1510	1510	0,00
n17	3×4	743	743	0,00
n18	3×4	1650	1650	0,00
n21	3×3	4205	4205	0,00
n22	3×4	12075	12075	0,00
n23	3×4	23	23	0,00

Tabel 4.2 – *lanjutan*

Kasus	Ukuran	Optimal	RVM	Deviasi (%)
n24	3×4	3460	3460	0,00
n25	3×4	809	809	0,00
n26	4×5	490	490	0,00
n27	5×5	59356	59356	0,00
n28	4×4	2090	2090	0,00
n29	3×4	3857	3857	0,00
n30	3×3	465	465	0,00
n31	3×4	1480	1480	0,00
n32	4×5	1780	1780	0,00
n33	3×3	1250	1310	4,80
n34	4×5	510	510	0,00
n35	3×4	280	280	0,00
S1	5×5	18855	18855	0,00
S2	5×5	153500	153500	0,00
S3	11×12	1044180	1044180	0,00
S4	11×12	92212	94060	2,00
S5	11×12	116020	116020	0,00
R1	2×94	12928429	12928429	0,00
R2	2×94	13657053	13657053	0,00
R3	2×94	12296587	12296587	0,00
R4	2×94	14117586	14117586	0,00
R5	2×94	14017419	14017419	0,00

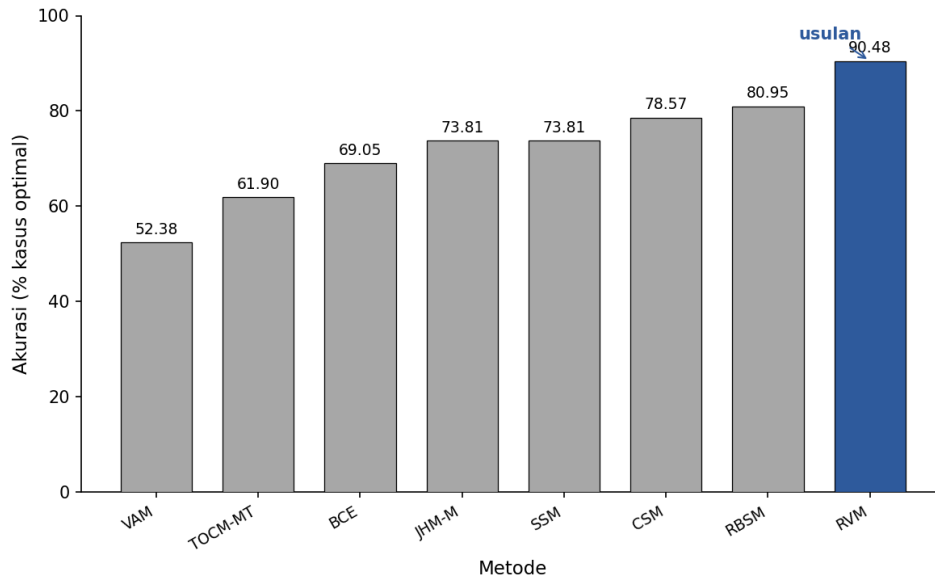
RVM hanya gagal mencapai optimal pada empat kasus, yaitu n05 (deviasi 1,22%), n10 (8,79%), n33 (4,80%), dan S4 (2,00%). Keempatnya merupakan kasus dengan banyak biaya bernilai sama atau berdekatan yang menyebabkan keputusan *tie-breaking* pada matriks tereduksi tidak selalu mengarah ke struktur optimal. Pembahasan lebih lanjut atas keempat kasus ini disajikan pada Subbab 4.2.

Dari hasil mentah seluruh metode pada Tabel 4.1, Tabel 4.3 merangkum perbandingan agregat tingkat akurasi (AP) dan deviasi rata-rata (DP) RVM terhadap tujuh metode IBFS perbandingan terpublikasi.

Tabel 4.3 Perbandingan RVM dengan Metode IBFS Terpublikasi

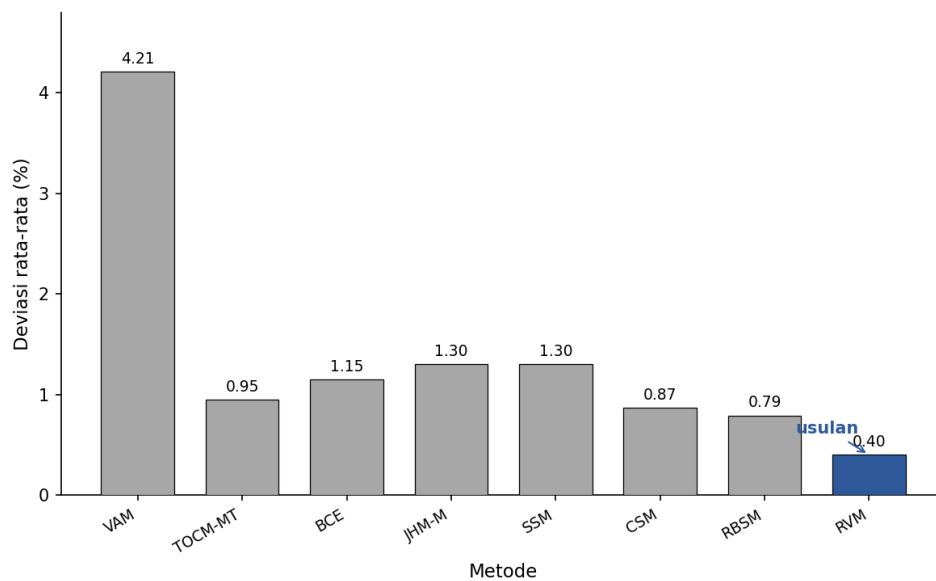
Metode	Optimal	Akurasi (%)	Deviasi rata-rata (%)
VAM	22	52,38	4,21
TOCM-MT	26	61,90	0,95
BCE	29	69,05	1,15
JHM-M	31	73,81	1,30
SSM	31	73,81	1,30
CSM	33	78,57	0,87
RBSM	34	80,95	0,79
RVM	38	90,48	0,40

RVM menempati posisi teratas pada kedua metrik. Dari sisi akurasi, RVM mencapai 90,48%, melampaui RBSM (80,95%) sebagai metode konstruksi terbaik berikutnya, dan jauh di atas VAM klasik (52,38%). Dari sisi deviasi rata-rata, RVM mencapai 0,40%, lebih rendah dari RBSM (0,79%) dan kurang dari sepersepuluh deviasi VAM (4,21%). Gambar 4.1 memvisualisasikan perbandingan akurasi, dan Gambar 4.2 memvisualisasikan perbandingan deviasi rata-rata.



Gambar 4.1 Perbandingan Akurasi Antarmetode

Gambar 4.1 memperlihatkan tren peningkatan akurasi yang konsisten dari metode klasik ke metode terkini. VAM yang hanya mengandalkan *penalty* pada matriks asli berada di posisi terendah. Penambahan transformasi biaya peluang pada TOCM-MT, JHM, dan BCE meningkatkan akurasi ke kisaran 62–74%. Penyempurnaan aturan alokasi pada SSM, CSM, dan RBSM mendorong akurasi lebih jauh ke 74–81%. RVM melampaui seluruhnya, yang menunjukkan bahwa memadukan reduksi *zero-point* berlandaskan dualitas dengan *penalty* Vogel memberikan dasar prioritas alokasi yang lebih tepat dibanding penyempurnaan aturan alokasi pada matriks biaya asli.



Gambar 4.2 Perbandingan Deviasi Rata-Rata Antarmetode

Gambar 4.2 menegaskan pola yang sama dari perspektif kualitas solusi. Urutan deviasi tidak selalu sejalan dengan urutan akurasi; TOCM-MT misalnya memiliki deviasi rata-rata (0,95%) yang lebih rendah dari SSM (1,30%) meskipun akurasinya lebih rendah, artinya ketika TOCM-MT gagal optimal, kegagalannya cenderung kecil. RVM unggul pada kedua dimensi sekaligus, yaitu paling sering optimal sekaligus paling kecil deviasinya ketika tidak optimal, yang mengindikasikan kualitas solusi yang konsisten.

Secara karakteristik, keberhasilan RVM mencapai tingkat optimal yang tinggi (38 dari 42 kasus) bersumber pada kemampuannya menyelaraskan proses pencarian dengan prinsip *complementary slackness*. Reduksi *zero-point* mentransformasi biaya nominal menjadi *reduced cost* yang secara efektif bertindak sebagai proksi bagi *dual bound*. Oleh karena itu, perhitungan penalti Vogel di atas matriks tereduksi ini menghasilkan nilai *regret* yang jauh lebih akurat dibandingkan metode klasik. Alokasi algoritma secara natural terarah untuk mengisi sel-sel *tight* yang memang merupakan bagian dari struktur basis optimal, dan hanya mengalami sedikit penyimpangan ketika matriks memiliki kepadatan *twin cost* (*twin-cost*) ekstrem yang memicu ambiguitas *tie-breaking* berantai.

4.1.3 Matriks Evaluasi Menyeluruh

Bagian ini merangkum performa seluruh metode menurut keenam metrik evaluasi pada Subbab 3.2.4: AP, DP, dan PC sebagai metrik agregat (Tabel 4.4); IP sebagai metrik perpasangan (Tabel 4.6); FNOCS dirinci menurut ukuran matriks untuk seluruh metode (Tabel 4.7); dan OI sebagai ukuran manfaat IBFS bagi fase optimisasi lanjutan (Tabel 4.9).

Tabel 4.4 Matriks Evaluasi Menyeluruh Antarmetode (AP, DP, PC)

Metode	AP (%)	DP (%)	PC (%)
VAM	52,38	4,21	95,79
TOCM-MT	61,90	0,95	99,05
BCE	69,05	1,15	98,85
JHM-M	73,81	1,30	98,70
SSM	73,81	1,30	98,70
CSM	78,57	0,87	99,13
RBSM	80,95	0,79	99,21
RVM	90,48	0,40	99,60

Pada ketiga metrik, RVM menempati posisi terbaik, yaitu akurasi tertinggi (90,48%), deviasi terendah (0,40%), dan PC tertinggi (99,60%). Nilai PC seluruh metode berada di atas 95%, yang menunjukkan bahwa meskipun akurasi (AP) bervariasi cukup lebar, secara biaya semua metode menghasilkan solusi yang relatif dekat ke optimal; perbedaan tipis pada PC (RBSM 99,21% vs RVM 99,60%) justru menegaskan bahwa keunggulan RVM konsisten baik pada frekuensi optimal maupun pada kedekatan biaya.

4.1.3.1 Improvement Percentage

Tabel 4.5 merinci selisih biaya aktual RVM terhadap masing-masing tujuh metode pembandingan untuk seluruh 42 kasus uji, dengan notasi angka +N menunjukkan RVM lebih murah *N* unit dari metode pembandingan (Positif), angka -N yang dicetak tebal menunjukkan RVM lebih mahal *N* unit (Negatif), dan angka 0 menunjukkan kedua metode menghasilkan biaya yang sama (Seri). Selanjutnya, Tabel 4.6 merangkum distribusi kasus Positif, Negatif, dan Nol tersebut beserta nilai rata-rata IP (Persamaan 3.11) untuk memberikan gambaran agregat per-pasangan. Metrik ini menjawab pertanyaan yang tidak dapat dijawab AP atau DP, yaitu seberapa besar keunggulan RVM ketika dibandingkan lawan satu metode tertentu, kasus demi kasus.

Tabel 4.5 Rincian Selisih Biaya dan Status Per-Kasus RVM terhadap Metode Pembandingan

Kasus	Biaya RVM	VAM	TOCM-MT	BCE	JHM-M	SSM	CSM	RBSM
n01	5.600	0	0	0	0	0	0	0
n02	28	0	0	0	0	0	0	0
n03	1.102	+2	+25	+35	+32	+32	+32	+25
n04	1.580	0	0	0	+45	+45	0	+45
n05	415	+55	+20	-5	+15	+15	+15	+15
n06	7.750	+600	0	+600	0	0	0	0
n07	1.160	+60	0	+10	+5	+5	0	0
n09	111	+3	0	0	0	0	0	0
n10	990	0	-80	-80	-30	-30	-80	-80
n11	2.460	+520	+40	0	+40	+40	0	0
n12	425	0	+14	+14	0	0	+14	0
n13	4.525	+600	+700	0	0	0	0	0
n14	920	+40	+10	0	0	0	0	0
n15	156	0	0	0	0	0	0	0
n16	1.510	+80	+20	+20	0	0	0	+80
n17	743	+36	0	0	0	0	+36	+36
n18	1.650	+95	0	+45	0	0	0	0
n21	4.205	0	0	+224	0	0	+224	0
n22	12.075	0	+125	+625	0	0	0	0
n23	23	0	0	0	0	0	0	0
n24	3.460	+60	0	0	0	0	0	0
n25	809	+40	0	0	0	0	+40	+40
n26	490	0	+18	0	0	0	0	0

Tabel 4.5 – *lanjutan*

Kasus	Biaya RVM	VAM	TOCM-MT	BCE	JHM-M	SSM	CSM	RBSM
n27	59.356	+9.448	0	+5.143	0	0	0	0
n28	2.090	+600	0	0	0	0	0	0
n29	3.857	0	0	0	0	0	0	0
n30	465	+8	+10	+10	0	0	+10	0
n31	1.480	0	0	0	0	0	0	0
n32	1.780	0	0	0	+40	+40	0	0
n33	1.310	+120	-60	0	0	0	0	0
n34	510	0	0	0	0	0	0	0
n35	280	+40	0	0	0	0	0	0
R1	12.928.429	0	+37.356	0	0	0	0	0
R2	13.657.053	0	+43.854	0	0	0	0	0
R3	12.296.587	0	+26.298	0	0	0	0	0
R4	14.117.586	0	+49.354	0	0	0	0	0
R5	14.017.419	0	+57.468	0	0	0	0	0
S1	18.855	0	0	+180	+2.070	+2.070	0	0
S2	153.500	0	0	0	+23.000	+23.000	0	0
S3	1.044.180	0	0	0	0	0	0	0
S4	94.060	-709	-1.629	+204	+1.241	+1.241	+1.380	+1.380
S5	116.020	0	0	0	0	0	0	0

Tabel 4.6 Improvement Percentage (IP) RVM terhadap Tiap Metode Pembanding

Metode Pembanding	Jumlah Kasus			Rata-rata IP (%)
	Positif	Negatif	Nol	
VAM	18	1	23	3,32
TOCM-MT	15	3	24	0,48
BCE	12	2	28	0,70
JHM-M	9	1	32	0,82
SSM	9	1	32	0,82
CSM	8	1	33	0,44
RBSM	7	1	34	0,37

Sebagaimana terlihat pada Tabel 4.5 dan 4.6, RVM lebih murah daripada VAM pada 18 dari 42 kasus dengan rata-rata IP 3,32%, sedangkan terhadap RBSM, metode konstruksi terkuat kedua, RVM lebih murah pada 7 kasus (n03, n04, n16, n17, n25, S1, S4) dengan rata-rata IP 0,37%, konsisten dengan kedekatan PC keduanya pada Tabel 4.4. Pada seluruh tujuh pasangan, jumlah kasus "Negatif" (RVM lebih mahal) tidak lebih dari tiga, dan seluruhnya berasal dari subset kasus kegagalan RVM yang sama (n05, n10, n33, S4; Subbab 4.2.2).

4.1.3.2 FNOCS menurut Ukuran Matriks

Tabel 4.7 merinci FNOCS seluruh delapan metode berdasarkan ukuran matriks, untuk memperlihatkan ketahanan tiap metode lintas dimensi. Rincian ini melengkapi angka AP agregat pada Tabel 4.4 dengan menunjukkan di ukuran mana tepatnya sebuah metode kuat atau lemah, sehingga keunggulan yang tampak menyeluruh pada AP dapat diuji apakah benar-benar merata atau hanya bertumpu pada ukuran matriks tertentu.

Tabel 4.7 Pengelompokan Ukuran Matriks

Ukuran	Jumlah Data Uji	Jumlah Optimal pada Metode							
		VAM	TOCM-MT	BCE	JHM-M	SSM	CSM	RBSM	RVM
2×94	5	5	0	5	5	5	5	5	5
3×3	6	3	3	2	5	5	2	5	5
3×4	16	6	12	12	13	13	14	13	15
3×5	1	1	1	1	0	0	1	0	1
4×4	4	0	3	3	3	3	3	3	3
4×5	3	3	2	3	2	2	3	3	3
5×5	4	2	3	1	1	1	3	3	4
11×12	3	2	2	2	2	2	2	2	2
Total	42	22	26	29	31	31	33	34	38

RVM mencapai optimal penuh pada ukuran 2×94 , 3×5 , 4×5 , dan 5×5 , serta hampir sempurna pada 3×4 (15 dari 16). Penurunan terjadi pada beberapa kasus 3×3 , 4×4 , dan 11×12 yang memuat banyak biaya bernilai sama, sejalan dengan empat kasus kegagalan pada Subbab 4.1.2. Pola FNOCS metode lain juga mengikuti karakteristik ukuran tertentu; VAM misalnya gagal total pada 4×4 (0 dari 4) namun optimal penuh pada seluruh kasus 2×94 , menunjukkan bahwa kelemahan tiap metode bersifat spesifik-ukuran, bukan seragam di semua dimensi.

4.1.3.3 Optimality Iterations

Tabel 4.8 merinci jumlah pivot MODI yang dibutuhkan oleh masing-masing metode pada seluruh 42 kasus uji, dengan angka pada tiap sel menyatakan banyaknya iterasi/pivot MODI yang diperlukan untuk membawa IBFS ke solusi optimal; angka 0 menandakan IBFS sudah optimal sejak awal, tanpa perlu pivot sama sekali. Selanjutnya, Tabel 4.9 merangkum rata-rata, nilai minimum, maksimum, serta frekuensi kasus $OI = 0$ untuk setiap metode. Metrik ini mengukur manfaat praktis IBFS sebagai titik awal optimisasi lanjutan, sebab semakin sedikit pivot yang dibutuhkan, semakin ringan beban komputasi perbaikan solusi.

Tabel 4.8 Rincian Jumlah Pivot MODI (OI) Seluruh Metode pada 42 Kasus Uji

Kasus	VAM	TOCM-MT	BCE	JHM-M	SSM	CSM	RBSM	RVM
n01	0	0	0	0	0	0	0	0
n02	0	0	0	0	0	0	0	0
n03	1	1	2	1	1	1	1	0
n04	0	0	0	1	1	0	1	0
n05	3	2	0	1	1	1	1	1
n06	1	0	1	0	0	0	0	0
n07	1	0	1	1	1	0	0	0
n09	1	0	0	0	0	0	0	0
n10	1	0	0	1	1	0	0	1
n11	1	1	0	1	1	0	0	0
n12	0	1	1	0	0	1	0	0
n13	1	2	0	0	0	0	0	0
n14	1	1	0	0	0	0	0	0
n15	0	0	0	0	0	0	0	0
n16	1	1	1	0	0	0	1	0
n17	1	0	0	0	0	1	1	0
n18	2	0	2	0	0	0	0	0
n21	0	0	1	0	0	1	0	0
n22	0	1	1	0	0	0	0	0
n23	0	0	0	0	0	0	0	0
n24	2	0	0	0	0	0	0	0
n25	1	0	0	0	0	1	1	0
n26	0	1	0	0	0	0	0	0
n27	3	0	3	0	0	0	0	0

Tabel 4.8 – *lanjutan*

Kasus	VAM	TOCM-MT	BCE	JHM-M	SSM	CSM	RBSM	RVM
n28	1	0	0	0	0	0	0	0
n29	0	0	0	0	0	0	0	0
n30	1	1	1	0	0	1	0	0
n31	0	0	0	0	0	0	0	0
n32	0	0	0	1	1	0	0	0
n33	2	0	2	2	2	2	2	2
n34	0	0	0	0	0	0	0	0
n35	1	0	0	0	0	0	0	0
R1	5	14	5	5	5	5	5	5
R2	3	11	3	3	3	3	3	3
R3	4	12	4	4	4	4	4	4
R4	3	15	3	3	3	3	3	3
R5	3	18	3	3	3	3	3	3
S1	0	0	1	2	2	0	0	0
S2	0	0	0	2	2	0	0	0
S3	9	9	9	9	9	9	9	9
S4	9	5	14	10	10	5	5	5
S5	9	9	9	9	9	9	9	9

Tabel 4.9 Optimality Iterations (OI) Tiap Metode pada 42 Kasus Uji

Metode	Rata-rata OI	Minimum	Maksimum	Kasus OI=0
RVM	1,07	0	9	31
RBSM	1,17	0	9	27
CSM	1,19	0	9	26
JHM-M	1,40	0	10	24
SSM	1,40	0	10	24
BCE	1,60	0	14	22
VAM	1,69	0	9	15
TOCM-MT	2,50	0	18	24

Sebagaimana diperlihatkan pada Tabel 4.8 dan 4.9, RVM membutuhkan OI rata-rata terendah (1,07) dan mencapai $OI = 0$ (IBFS sudah optimal sejak awal) pada 31 dari 42 kasus, terbanyak di antara seluruh metode. RVM hanya membutuhkan pivot pada 11 kasus, yang terdiri atas 4 kasus tidak optimal (n05: 1 pivot, n10: 1 pivot, n33: 2 pivot, S4: 5 pivot) serta 7 kasus sintetis dan *real-world* berukuran besar (S3: 9 pivot, S5: 9 pivot, R1: 5 pivot, R2: 3 pivot, R3: 4 pivot, R4: 3 pivot, R5: 3 pivot).

TOCM-MT menonjol sebagai pengecualian menarik: meskipun AP-nya (61,90%) tidak terburuk, rata-rata OI-nya tertinggi (2,50) dan membengkak hingga 11–18 pivot pada kelompok *real-world* R1–R5, jauh di atas RVM dan metode lain yang hanya membutuhkan 3–5 pivot pada kelompok kasus yang sama. AP dan OI mengukur dimensi yang berbeda: AP menilai kedekatan biaya IBFS terhadap optimal, sedangkan OI menilai kedekatan *struktur basis* IBFS terhadap basis optimal. Pada kasus berkepadatan *twin cost* tinggi (n33, S3, S5), seluruh delapan metode memerlukan jumlah pivot yang sama besar (berurutan 2, 9, dan 9), mengindikasikan banyaknya solusi optimal alternatif (*alternate optima*) yang membuat OI pada kasus-kasus tersebut lebih bergantung pada aturan *tie-breaking* prosedur MODI itu sendiri ketimbang pada metode pembentukan IBFS-nya.

4.1.4 Studi Ablasi Fungsi Penalty

Studi ablasi dilakukan dengan menguji 15 varian fungsi *penalty* (yang formulasi dan interpretasi matematisnya telah dijelaskan pada Tabel 3.27 di Subbab 3.2.6). Tabel 4.10 menyajikan hasil lengkap pengujian tersebut pada *benchmark* 42 kasus uji, diurutkan menurut tingkat akurasi. Notasi s_1 , s_2 , l_1 , l_2 , v_i , $\bar{v}$, k beserta varian berbedanya mengikuti definisi pada Subbab 3.2.6.

Tabel 4.10 Hasil Studi Ablasi 15 Fungsi *Penalty* pada Benchmark 42 Kasus

Nama Penalty	Fungsi Penalty	Optimal	Akurasi (%)	Deviasi (%)
Selisih Dua Cost Terkecil dengan Nilai Sama Tetap Dihitung	$s_2 - s_1$	38	90,48	0,40
Median Cost - Cost Terendah	$\text{med}(v) - s_1$	35	83,33	0,57
Rata-rata Selisih Cost terhadap Cost Terendah	$\bar{v} - s_1$	32	76,19	1,00
Rata-rata Cost	$\bar{v}$	32	76,19	1,00
Cost Tertinggi - Cost Terendah	$l_1 - s_1$	30	71,43	1,73
Total Selisih Cost terhadap Cost Terendah	$\sum_i (v_i - s_1)$	29	69,05	1,86
Contra Harmonic Mean	$\frac{\sum_i v_i^2}{\sum_i v_i}$	27	64,29	2,38
Geometric Mean	$\left(\prod_{v_i \neq 0} v_i\right)^{1/k^*}$	27	64,29	3,17
Standar Deviasi Cost	$\sqrt{\frac{1}{k} \sum_i (v_i - \bar{v})^2}$	26	61,90	2,19
Harmonic Mean	$\frac{1}{\sum_{v_i \neq 0} 1/v_i}$	25	59,52	4,16
Selisih Dua Cost Terkecil yang Berbeda	$s_2^* - s_1$	25	59,52	3,83
Selisih Dua Cost Terbesar yang Berbeda	$l_1 - l_2^*$	21	50,00	3,96
Selisih Dua Cost Terbesar dengan Nilai Sama Tetap Dihitung	$l_1 - l_2$	20	47,62	4,17
Inverse Harmonic Mean	$\frac{1}{k^*} \sum_{v_i \neq 0} \frac{1}{v_i}$	8	19,05	9,83
Inverse Contra-Harmonic Mean	$\frac{\sum_i v_i}{\sum_i v_i^2}$	7	16,67	10,97

Tabel 4.10 memberikan tiga kesimpulan penting. Pertama, *penalty* selisih dua nilai terkecil dengan duplikat dihitung, yang dipakai RVM sendiri, menghasilkan akurasi tertinggi (38/42; 90,48%), mengonfirmasi secara empiris bahwa pilihan *penalty* pada Subbab 2.2.6 memang yang terbaik di antara 15 varian yang diuji, bukan sekadar pilihan teoretis yang kebetulan masuk akal.

Kedua, perbandingan varian duplikat dihitung lawan duplikat dilewati mengungkap asimetri yang relevan secara teoretis. Pada pasangan dua nilai terkecil, varian duplikat-dihitung (90,48%) jauh mengungguli varian duplikat-dilewati (59,52%), yaitu selisih 13 kasus. Hal ini sejalan dengan *complementary slackness* (Subbab 2.2.3): ketika beberapa sel pada satu lini sama-sama bernilai *reduced cost* terkecil, lini tersebut sesungguhnya fleksibel (banyak pilihan sama-sama *tight*) sehingga sepatutnya berpenalti rendah; memaksa s_2 menjadi nilai yang berbeda dari s_1 justru menaikkan penalti secara artifisial pada lini yang sebenarnya tidak mendesak, mengarahkan alokasi ke lini yang salah. Pada pasangan dua nilai terbesar, yang tidak punya landasan *complementary slackness* serupa karena RVM tidak pernah mengalokasikan ke sel bernilai besar, kedua varian sama-sama berperforma rendah dan bahkan sedikit berbalik arah (47,62% berbanding 50,00%). Hal ini menegaskan bahwa keputusan duplikat dihitung pada RVM bukan kebetulan teknis, melainkan konsekuensi langsung dari teori yang mendasarinya.

Ketiga, bahkan fungsi *penalty* sederhana seperti rata-rata *cost* (32/42; 76,19%) tetap jauh mengungguli VAM standar (22/42; 52,38%) yang bekerja pada biaya asli. Artinya, sumber

utama keunggulan RVM adalah **reduksi zero-point**, bukan bentuk *penalty* tertentu; reduksi mengangkat performa seluruh varian, sedangkan bentuk *penalty* menentukan seberapa tinggi performa tersebut. *Penalty* berbasis rata-rata harmonik terbalik (*Inverse Harmonic Mean* dan *Inverse Contra-Harmonic Mean*, 7–8/42) menghasilkan performa terburuk karena transformasi kebalikannya membesar-besarkan kontribusi nilai tereduksi kecil yang justru seharusnya menandakan sel *tight* berbiaya rendah, sehingga bertentangan langsung dengan prinsip *regret* yang melandasi Subbab 2.2.6.

4.1.5 Analisis Degenerasi

Sebagai metode konstruksi, RVM dapat menghasilkan solusi degenerate, yaitu solusi dengan sel basis positif kurang dari $m+n-1$, ketika sisa *supply* dan *demand* mencapai nol bersamaan dalam satu alokasi. Degenerasi teramati terutama pada kasus berukuran besar dan kasus dengan banyak biaya bernilai sama. Sebagaimana dijelaskan pada Subbab 3.1, degenerasi merupakan fenomena standar pada metode IBFS dan tidak memengaruhi kelayakan solusi maupun nilai biaya total. Apabila diperlukan oleh prosedur MODI, basis dapat dilengkapi dengan alokasi bernilai nol tanpa mengubah biaya.

4.2 Pembahasan

Hasil keseluruhan menunjukkan bahwa RVM mencapai kualitas solusi terbaik di antara metode IBFS terpublikasi pada *benchmark* 42 kasus. Keunggulan ini dapat dijelaskan secara teoretis, yakni reduksi *zero-point* mengubah matriks biaya menjadi matriks *reduced cost* berlandaskan dualitas, sehingga *penalty* Vogel yang dihitung di atasnya benar-benar mengukur *regret* relatif terhadap batas bawah *dual*, bukan sekadar selisih biaya nominal. Alokasi pun terarah ke sel *tight* yang menurut *complementary slackness* dipilih oleh solusi optimal.

Meskipun demikian, RVM tetaplah heuristik dan tidak menjamin optimalitas. Pada empat kasus kegagalan (n05, n10, n33, S4), struktur biaya memuat banyak nilai sama atau berdekatan sehingga keputusan *tie-breaking* pada matriks tereduksi tidak selalu menuju struktur optimal; kasus n10 dengan deviasi 8,79% merupakan yang terjauh. Hal ini sejalan dengan *No Free Lunch Theorem* bahwa tidak ada metode tunggal yang optimal untuk seluruh kasus. Keterbatasan lain penelitian ini adalah cakupan yang dibatasi pada *balanced transportation problem* dengan biaya deterministik, serta sifat satu-langkah tanpa penyempurnaan. Kasus tak seimbang berada di luar cakupan, sedangkan kasus dengan ketidakpastian biaya (*fuzzy*) maupun integrasi dengan langkah optimisasi lanjutan belum dieksplorasi dan menjadi peluang penelitian berikutnya.

4.2.1 Analisis Kasus Keunggulan RVM atas RBSM

Bagian ini menelusuri secara spesifik kasus-kasus tempat RVM mengungguli RBSM, metode konstruksi terbaik kedua pada *benchmark* ini (Tabel 4.3). Dari 42 kasus, terdapat empat kasus tempat RVM tepat mencapai solusi optimal sedangkan RBSM tidak, yaitu n03, n16, n17, dan n25. Keempatnya ditelusuri per iterasi di bawah ini untuk memperlihatkan *pada tahap mana* RVM mengambil keputusan alokasi yang lebih tepat dibanding RBSM.

Kasus n03 (5×5 , $Z^* = 1102$) memuat matriks biaya pada Tabel 4.11. Reduksi *zero-point* pada iterasi pertama menghasilkan matriks tereduksi pada Tabel 4.12, dengan penalti maksimum bernilai 17 pada kolom D_3 – jauh lebih besar dari lini lain, sehingga tidak terjadi seri. Sel *reduced cost* nol pada kolom itu adalah (S_1, D_3) , dan RVM mengalokasikan $q = \min(8, 10) = 8$ unit berbiaya 9 per unit, memenuhi seluruh pasokan S_1 sekaligus.

Tabel 4.11 Kasus n03 (5×5), $Z^* = 1102$

	D_1	D_2	D_3	D_4	D_5	Supply
S_1	73	40	9	79	20	8
S_2	62	93	96	8	13	7
S_3	96	65	80	50	65	9
S_4	57	58	29	12	87	3
S_5	56	23	87	18	12	5
Demand	6	8	10	4	4	

Tabel 4.12 Matriks Tereduksi Kasus n03 pada Iterasi 1

	D_1	D_2	D_3	D_4	D_5	Penalti
S_1	20	20	0	70	11	11
S_2	10	74	88	0	5	5
S_3	2	4	30	0	15	2
S_4	1	35	17	0	75	1
S_5	0	0	75	6	0	0
Penalti	1	4	17	0	5	

Pada iterasi kedua, penalti maksimum bernilai 13 kembali jatuh pada kolom D_3 , dan RVM mengalokasikan $q = \min(3, 2) = 2$ unit pada sel (S_4, D_3) berbiaya 29 per unit, menuntaskan seluruh permintaan D_3 . Tiga iterasi berikutnya mengalokasikan sel-sel sisa secara berurutan: $(S_5, D_5) = 4$ unit berbiaya 12, $(S_5, D_2) = 1$ unit berbiaya 23, $(S_2, D_4) = 4$ unit berbiaya 8, $(S_2, D_5) = 3$ unit berbiaya 13, $(S_4, D_1) = 1$ unit berbiaya 57, $(S_3, D_2) = 7$ unit berbiaya 65, dan $(S_3, D_1) = 2$ unit berbiaya 96, menghasilkan total biaya optimal 1.102.

Perbedaannya dengan RBSM terletak pada perlakuan pasokan S_1 , dan dapat ditelusuri persis pada titik mana kedua metode berpisah jalur. Tabel 4.13 membandingkan langkah-langkah kunci keduanya secara berdampingan.

Tabel 4.13 Komparasi Langkah Kunci RVM dan RBSM pada Kasus n03

Tahap	RVM (berbasis penalti <i>reduced cost</i>)	RBSM (berbasis rasio biaya terkecil pertama/kedua)
Awal	Reduksi <i>zero-point</i> membentuk matriks tereduksi (Tabel 4.12); penalti kolom D_3 tertinggi (17)	Alokasi <i>minimum-cell</i> awal langsung mengisi kolom termurah tiap baris tanpa reduksi: $(S_1, D_3) = 10$, $(S_2, D_4) = 4$, $(S_5, D_1) = 6$, $(S_5, D_2) = 8$, $(S_5, D_5) = 4$
Keputusan kunci pada S_1	Penalti tertinggi (17) jatuh pada kolom D_3 ; seluruh pasokan S_1 (8 unit) dialokasikan sekaligus ke (S_1, D_3) berbiaya 9 – S_1 langsung habis di iterasi pertama	Mekanisme donor-penerima memaksa S_1 ber-status <i>excess row</i> (alokasi awal 10 melebihi pasokan 8), sehingga kelebihan 2 unit didonasikan; pada Iterasi 1–2 pasokan S_1 akhirnya terpecah menjadi $(S_1, D_3) = 7$ dan $(S_1, D_5) = 1$ berbiaya 20
Dampak pada D_1	Karena S_1 sudah habis di sel murah D_3 , permintaan D_1 dipenuhi campuran $(S_3, D_1) = 2$ berbiaya 96 dan $(S_4, D_1) = 1$ berbiaya 57	Karena S_1 terpecah dan tidak lagi tersedia untuk D_1 , seluruh 6 unit permintaan D_1 jatuh ke (S_3, D_1) berbiaya 96 tanpa alternatif lebih murah
Total biaya	1.102 (optimal)	1.127 (selisih +25)

Akar perbedaannya terletak pada *kapan* keputusan dievaluasi. RVM mengevaluasi seluruh submatriks aktif ulang setiap iterasi melalui reduksi *zero-point*, sehingga penalti kolom D_3 yang tinggi (17) langsung terlihat dan pasokan S_1 dialokasikan penuh ke sana sejak awal, tanpa pernah tersentuh oleh D_1 . RBSM, sebaliknya, mengalokasikan berdasarkan biaya termurah tiap baris pada matriks asli lebih dahulu (fase *minimum-cell*) dan baru mengoreksi ketidakseimbangan lewat mekanisme donor-penerima; pada fase koreksi inilah S_1 "kebetulan" dipecah menjadi dua sel karena rasio biaya donor yang dihitung ($x_1 = 57$, $x_2 = 13$ pada Iterasi 1) menguntungkan pemecahan tersebut secara lokal, padahal akibatnya D_1 kehilangan akses ke S_1 sama sekali. RVM, karena mengisi (S_1, D_3) penuh 8 unit sejak iterasi pertama, membatasi alokasi pada (S_3, D_1) hanya 2 unit dan mengalihkan sisanya ke $(S_4, D_1) = 1$ unit berbiaya 57 – jauh lebih murah daripada seluruhnya jatuh ke (S_3, D_1) berbiaya 96.

Kasus n16 (3×4 , $Z^* = 1510$) memuat matriks biaya pada Tabel 4.14. Reduksi iterasi pertama menghasilkan matriks tereduksi pada Tabel 4.15, dengan penalti maksimum bernilai 4 pada kolom D_1 , tidak berseri dengan lini lain. RVM mengalokasikan $q = \min(50, 30) = 30$ unit pada sel (S_2, D_1) berbiaya 13 per unit, menghabiskan seluruh permintaan D_1 .

Tabel 4.14 Kasus n16 (3×4), $Z^* = 1510$

	D_1	D_2	D_3	D_4	Supply
S_1	22	17	25	15	30
S_2	13	10	16	20	50
S_3	16	19	9	5	50
Demand	30	30	50	20	

Tabel 4.15 Matriks Tereduksi Kasus n16 pada Iterasi 1

	D_1	D_2	D_3	D_4	Penalti
S_1	4	2	6	0	2
S_2	0	0	2	10	0
S_3	8	14	0	0	0
Penalti	4	2	2	0	

Titik keputusan berikutnya muncul pada iterasi kedua, tempat penalti maksimum bernilai 2 berseri di 4 lini sekaligus (S_1, S_2, D_2, D_3), dan aturan baris-dulu memilih S_1 , mengisi (S_1, D_4) = 20 unit berbiaya 15. Iterasi ketiga mengalokasikan (S_3, D_3) = 50 unit berbiaya 9 pada penalti maksimum 10, menghabiskan pasokan S_3 sekaligus permintaan D_3 . Dua iterasi terakhir menyelesaikan sisa alokasi (S_1, D_2) = 10 unit berbiaya 17 dan (S_2, D_2) = 20 unit berbiaya 10, menghasilkan total biaya optimal 1.510.

Perbedaannya dengan RBSM terletak pada perlakuan pasokan S_1 : RBSM langsung mengalokasikan seluruh pasokan S_1 ke (S_1, D_2) sebesar 30 unit sejak awal, sehingga permintaan D_3 terpecah ke (S_2, D_3) = 20 unit berbiaya 16 dan (S_3, D_3) = 30 unit berbiaya 9. Pengalokasian 20 unit D_3 pada sel berbiaya lebih tinggi ini menaikkan total biaya RBSM menjadi 1.590. RVM menghindari pola ini karena penalti pada iterasi pertama mengarahkan alokasi ke D_1 lebih dulu, bukan langsung menghabiskan S_1 ke D_2 .

Kasus n17 dan n25 (3×4) berbagi struktur biaya yang hampir identik, dengan $Z^* = 743$ untuk n17 dan $Z^* = 809$ untuk n25 (Tabel 4.17). Pada n17, reduksi iterasi pertama menghasilkan penalti maksimum bernilai 30 pada kolom D_3 , tidak berseri, dan RVM mengalokasikan $q = \min(9, 7) = 7$ unit pada sel (S_2, D_3) berbiaya 40 per unit, menghabiskan seluruh permintaan D_3 . Tabel 4.16 menunjukkan matriks tereduksi pada iterasi ini; n25 mengikuti pola serupa dengan hasil akhir yang sepadan.

Tabel 4.16 Matriks Tereduksi Kasus n17 pada Iterasi 1

	D_1	D_2	D_3	D_4	Penalti
S_1	0	20	30	0	0
S_2	31	0	0	30	30
S_3	23	0	52	12	12
Penalti	23	0	30	12	

Empat iterasi berikutnya melengkapi alokasi: pasokan S_1 terbagi ke (S_1, D_1) = 5 unit berbiaya 19 dan (S_1, D_4) = 2 unit berbiaya 10, sisa pasokan S_2 mengisi (S_2, D_2) = 2 unit

berbiaya 30, dan S_3 menuntaskan permintaan D_2 dan D_4 melalui $(S_3, D_2) = 6$ unit berbiaya 8 dan $(S_3, D_4) = 12$ unit berbiaya 20, menghasilkan total biaya optimal 743. Pada n25, alokasi akhir berpola sama: $(S_1, D_1) = 5$, $(S_1, D_4) = 2$, $(S_2, D_2) = 2$, $(S_2, D_3) = 8$, $(S_3, D_2) = 5$, dan $(S_3, D_4) = 13$, menghasilkan total biaya optimal 809.

Sel termahal pada kedua matriks ini adalah (S_2, D_4) berbiaya 60 (n17) dan 60 (n25), dan RVM tidak pernah mengalokasikan satu unit pun ke sel tersebut karena reduksi *zero-point* memperbarui biaya peluang di setiap iterasi, mengarahkan pasokan S_2 sepenuhnya ke D_2 dan D_3 yang lebih murah. RBSM, sebaliknya, menyisakan 2 unit pasokan S_2 yang terpaksa dialokasikan ke (S_2, D_4) , meningkatkan total biaya menjadi 779 pada n17 dan 849 pada n25.

Tabel 4.17 Kasus n17 dan n25 (3×4)

n17 ($Z^* = 743$)						n25 ($Z^* = 809$)					
	D_1	D_2	D_3	D_4	Sup.		D_1	D_2	D_3	D_4	Sup.
S_1	19	30	50	10	7	S_1	19	30	50	12	7
S_2	70	30	40	60	9	S_2	70	30	40	60	10
S_3	40	8	70	20	18	S_3	40	10	60	20	18
Dem.	5	8	7	14		Dem.	5	7	8	15	

4.2.2 Analisis Faktor Kasus Tidak Optimal

Bagian ini menelaah secara rinci empat kasus tempat RVM gagal mencapai optimal (n05, n10, n33, S4). Tabel 4.18 merangkum karakteristik keempat kasus beserta indikator kepadatan *twin cost*. Berikut ditelusuri eksekusi per iterasi pada masing-masing kasus dengan menyoroti titik keputusan pemicu ketakoptimalan.

Tabel 4.18 Karakteristik Kasus yang Tidak Mencapai Optimal

Kasus	Ukuran	Optimal	RVM	Deviasi (%)	<i>twin cost</i>	Repetisi maks
n05	4×4	410	415	1,22	11 dari 16	$3 \times$ (nilai 3)
n10	3×4	910	990	8,79	3 dari 12	$3 \times$ (nilai 20)
n33	3×3	1250	1310	4,80	6 dari 9	$2 \times$ (nilai 3, 5, 6)
S4	11×12	92212	94060	2,00	121 dari 132	$7 \times$ (nilai 81)

Kasus n05 (4×4 , $Z^* = 410$, RVM=415) memuat matriks biaya pada Tabel 4.19. Titik keputusan yang menentukan hasil akhir terjadi pada iterasi pertama: reduksi menghasilkan penalti bernilai 1 yang berseri di 4 lini sekaligus (S_2, S_3, D_3, D_4), masing-masing dengan tepat satu sel bernilai nol. Aturan pemecahan seri baris-dulu memilih baris S_2 dan mengalokasikan 25 unit pada sel (S_2, D_2) berbiaya 3 per unit, menghabiskan seluruh pasokan S_2 . Keputusan ini mengunci permintaan D_2 pada S_2 , padahal basis optimal memerlukan S_2 melayani D_1 sebesar 5 unit agar S_1 tidak terpaksa mengisi sel mahal pada iterasi berikutnya.

Iterasi berikutnya berjalan lurus mengikuti konsekuensi keputusan itu: penalti 4 pada kolom D_2 mengisi sel (S_1, D_2) sebesar 5 unit sisa permintaan (biaya 5 per unit), lalu penalti 3 pada kolom D_3 mengisi (S_1, D_3) sebesar 20 unit (biaya 9 per unit). Alokasi yang tersisa pada iterasi keempat terpaksa mengisi sel-sel mahal, yaitu $(S_1, D_1) = 5$ unit berbiaya 7, $(S_3, D_1) = 20$ unit berbiaya 3, $(S_4, D_1) = 5$ unit berbiaya 2, dan $(S_4, D_4) = 10$ unit berbiaya 3, sehingga total biaya RVM menjadi 415. Faktor kegagalannya persis pada pemecahan seri di iterasi pertama: solusi

optimal menempuh siklus pertukaran tertutup $(S_1, D_1) \rightarrow (S_1, D_2) \rightarrow (S_2, D_2) \rightarrow (S_2, D_1)$ bernilai 5 unit, selisih yang sama dengan deviasi 415 terhadap 410.

Tabel 4.19 Kasus n05 (4×4), $Z^* = 410$

	D_1	D_2	D_3	D_4	Supply
S_1	<u>7</u>	<u>5</u>	9	11	30
S_2	4	3	<u>8</u>	<u>6</u>	25
S_3	3	<u>8</u>	10	<u>5</u>	20
S_4	2	<u>6</u>	<u>7</u>	3	15
Demand	30	30	20	10	

Seluruh sel bergaris bawah pada Tabel 4.19 adalah 11 dari 16 sel yang memiliki pasangan bernilai sama (Tabel 4.18), terbagi dalam lima kelompok *twin cost*: nilai 3 pada (S_2, D_2) , (S_3, D_1) , (S_4, D_4) ; nilai 5 pada (S_1, D_2) , (S_3, D_4) ; nilai 6 pada (S_2, D_4) , (S_4, D_2) ; nilai 7 pada (S_1, D_1) , (S_4, D_3) ; dan nilai 8 pada (S_2, D_3) , (S_3, D_2) . Sel bercetak tebal, yaitu (S_2, D_2) , (S_3, D_1) , dan (S_4, D_4) yang berbagi nilai biaya 3, adalah repetisi maksimum (satu-satunya nilai yang muncul tiga kali) dan berkontribusi langsung pada seri penalti di iterasi pertama; sisanya (nilai 5, 6, 7, 8) masing-masing hanya berulang dua kali sehingga tidak memicu seri secara langsung namun tetap menyumbang pada kepadatan *twin cost* keseluruhan kasus ini.

Tabel 4.20 Perbandingan Alokasi RVM dan Optimal pada n05

Sel (biaya)	Alokasi RVM	Alokasi Optimal
(S_1, D_1) (7)	5	0
(S_1, D_2) (5)	5	10
(S_1, D_3) (9)	20	20
(S_2, D_1) (4)	0	5
(S_2, D_2) (3)	25	20
(S_3, D_1) (3)	20	20
(S_4, D_1) (2)	5	5
(S_4, D_4) (3)	10	10
Total biaya	415	410

Untuk menunjukkan persis pada iterasi mana RVM dan RBSM mulai mengambil keputusan berbeda, Tabel 4.21 menyandingkan status alokasi kedua metode pada n05 setelah iterasi pertama masing-masing.

Tabel 4.21 Status Alokasi Setelah Iterasi 1 RVM vs RBSM pada Kasus n05

RVM – Iterasi 1 (penalti maks. 1, seri 4 lini, baris-dulu → S_2)					
	D_1	D_2	D_3	D_4	Supply
S_1	7	5	9	11	30
S_2	4	3 (25)	8	6	25 → 0
S_3	3	8	10	5	20
S_4	2	6	7	3	15
Demand	30	30 → 5	20	10	

RBSM – Iterasi 1 (donor S_4 , FLC=2 pada D_1 , SLC=3 pada D_4)					
	D_1	D_2	D_3	D_4	Supply
S_1	0	0	0	0	30
S_2	0	30 (3)	0	0	25 (<i>excess</i>)
S_3	20 (3)	0	0	0	20 → 0
S_4	30 (2)	0	20 (7)	10 (3)	15 (<i>excess</i>)
Demand	30	30	20	10	

Sel yang diarsir menandai alokasi yang sudah terisi pada tahap tersebut, dengan angka dalam kurung sebagai biaya per unit. Perbedaannya sudah tampak sejak iterasi pertama: RVM, lewat reduksi *zero-point* dan penalti Vogel, langsung mengunci S_2 sepenuhnya ke $(S_2, D_2) = 25$ berbiaya 3 karena baris ini paling terbatas dalam seri penalti 4-lini. RBSM, yang bekerja dari matriks biaya asli tanpa reduksi, mengalokasikan berdasar rasio FLC/SLC per baris terlebih dahulu sehingga S_4 langsung tersebar ke tiga kolom sekaligus (D_1, D_3, D_4) sebagai baris *excess* pertama, sementara S_2 baru terisi $(S_2, D_2) = 30$ – lebih besar dari kebutuhan sebenarnya (permintaan D_2 total hanya 30, dan S_1 belum mendapat bagian sama sekali). Kelebihan alokasi S_2 dan S_4 inilah yang pada iterasi-iterasi berikutnya perlu direlokasi lewat mekanisme donor-penerima RBSM (Tabel 4.18), dan proses realokasi itulah yang akhirnya mendorong RBSM menutup sisa D_1 dengan sel $(S_3, D_1) = 20$ berbiaya 3 sekaligus menyisakan alokasi tersebar tak efisien pada S_4 , sehingga total biaya RBSM menjadi 430 – lebih mahal 15 unit dari RVM (415) dan 20 unit dari optimal (410). RVM justru lebih dekat ke optimal di kasus ini meski sama-sama gagal, karena reduksi *zero-point*-nya menunda komitmen penuh sampai penalti benar-benar dihitung, sedangkan RBSM sudah berkomitmen pada alokasi *minimum-cell* sebelum tahu apakah komitmen itu akan menyisakan permintaan yang sulit dipenuhi.

Kasus n10 (3×4 , $Z^* = 910$, RVM=990) menunjukkan pola kegagalan berbeda: komitmen berlebih, bukan salah pilih akibat seri. Matriks biayanya disajikan pada Tabel 4.23. Iterasi pertama berjalan wajar, penalti maksimum 17 pada kolom D_1 mengisi sel (S_3, D_1) sebesar 10 unit berbiaya 8. Titik kritisnya ada di iterasi kedua: penalti 5 berseri antara baris S_1 dan kolom D_4 , dan aturan baris-dulu memilih S_1 , lalu mengalokasikan seluruh pasokan S_1 sebesar 30 unit sekaligus ke sel (S_1, D_2) berbiaya 2 per unit. Tabel 4.22 memperlihatkan matriks tereduksi pada iterasi ini, tempat ketiadaan *look-ahead* memancing RVM langsung mengurus pasokan ke sel termurah secara seketika.

Habisnya pasokan S_1 menutup akses sel termurah D_4 , yaitu (S_1, D_4) berbiaya 11, sehingga permintaan D_4 terpaksa dilayani sel mahal (S_2, D_4) sebesar 20 unit berbiaya 20 dan (S_3, D_4) sebesar 10 unit berbiaya 18. Total biaya RVM menjadi 990, deviasi 8,79% terhadap optimal,

deviasi terbesar di antara keempat kasus. Faktor kegagalannya adalah komitmen berlebih pasokan S_1 ke D_2 pada iterasi kedua: solusi optimal justru memecah S_1 menjadi $(S_1, D_2) = 10$ dan $(S_1, D_4) = 20$, lalu menutup sisa D_2 lewat $(S_2, D_2) = 20$ berbiaya 7 – keputusan yang tidak mungkin diambil RVM karena konstruksinya satu-lintasan tanpa peninjauan ulang.

Tabel 4.22 Snapshot Matriks Tereduksi Kasus n10 pada Iterasi 2

	D_2	D_3	D_4	Supply	Penalti
S_1	0 (2)	16 (20)	0 (11)	30	5
S_2	0 (7)	0 (9)	4 (20)	50	0
S_3	5 (14)	5 (16)	0 (18)	10	5
Demand	30	30	30		
Penalti	0	0	5		

Tabel 4.23 Kasus n10 (3×4), $Z^* = 910$

	D_1	D_2	D_3	D_4	Supply
S_1	<u>20</u>	2	<u>20</u>	11	30
S_2	24	7	9	<u>20</u>	50
S_3	8	14	16	18	20
Demand	10	30	30	30	

Sel bergaris bawah dan tercetak tebal pada Tabel 4.23, yaitu (S_1, D_1) , (S_1, D_3) , dan (S_2, D_4) , adalah tiga sel yang berbagi nilai biaya 20 – repetisi maksimum sekaligus satu-satunya kelompok *twin cost* pada kasus ini (3 dari 12 sel, Tabel 4.18). Kepadatan yang jauh lebih rendah dibanding n05 atau S4 ini menegaskan bahwa deviasi 8,79% pada n10 bukan berasal dari banyaknya seri penalti, melainkan dari pola komitmen berlebih yang dijelaskan berikut.

Tabel 4.24 Perbandingan Alokasi RVM dan Optimal pada n10

Sel (biaya)	Alokasi RVM	Alokasi Optimal
(S_1, D_2) (2)	30	10
(S_1, D_4) (11)	0	20
(S_2, D_2) (7)	0	20
(S_2, D_3) (9)	30	30
(S_2, D_4) (20)	20	0
(S_3, D_1) (8)	10	10
(S_3, D_4) (18)	10	0
Total biaya	990	910

Kasus n33 (3×3 , $Z^* = 1250$, RVM=1310) memuat matriks biaya pada Tabel 4.26. Reduksi pada iterasi pertama menghasilkan matriks tereduksi pada Tabel 4.25, dengan penalti 1 yang berseri di 4 lini sekaligus (S_2, S_3, D_2, D_3) . Aturan baris-dulu memilih S_2 , mengalokasikan seluruh pasokannya (80 unit) ke sel (S_2, D_3) berbiaya 4 per unit. Iterasi kedua kembali berseri, kali ini antara S_3 dan D_2 pada penalti 4; baris-dulu memilih S_3 dan mengisi (S_3, D_1) sebesar 70 unit berbiaya 6. Dua iterasi terakhir hanya menyisakan satu pilihan: (S_1, D_2) terisi 90 unit

berbiaya 3, dan sisa 30 unit pasokan S_3 terdorong ke sel mahal (S_3, D_2) berbiaya 10, sehingga total biaya RVM menjadi 1.310.

Tabel 4.25 Matriks Tereduksi Kasus n33 pada Iterasi 1

	D_1	D_2	D_3	Penalti
S_1	0	0	2	0
S_2	2	1	0	1
S_3	0	4	1	1
Penalti	0	1	1	

Tabel 4.27 membandingkan alokasi ini dengan solusi optimal. Bedanya terletak pada bagaimana permintaan D_2 dan D_3 dibagi antara S_2 dan S_3 : solusi optimal memecah pasokan S_2 menjadi $(S_2, D_2) = 30$ dan $(S_2, D_3) = 50$, lalu menutup sisa D_3 dengan $(S_3, D_3) = 30$ berbiaya 7 – jauh lebih murah daripada sel $(S_3, D_2) = 10$ yang terpaksa dipakai RVM. Faktor kegagalannya sama dengan pola n05: pemecahan seri di dua iterasi berturut-turut (baris S_2 pada iterasi pertama, baris S_3 pada iterasi kedua) menghabiskan opsi murah lebih cepat daripada yang dibutuhkan basis optimal, sehingga 30 unit sisa pasokan S_3 tidak punya pilihan lain selain sel termahal di matriks tersebut.

Tabel 4.26 Kasus n33 (3×3), $Z^* = 1250$

	D_1	D_2	D_3	Supply
S_1	<u>3</u>	<u>3</u>	<u>5</u>	90
S_2	<u>6</u>	<u>5</u>	4	80
S_3	<u>6</u>	10	7	100
Demand	70	120	80	

Sel bergaris bawah pada Tabel 4.26 menandai tiga pasangan nilai yang identik: biaya 3 pada (S_1, D_1) dan (S_1, D_2) ; biaya 5 pada (S_1, D_3) dan (S_2, D_2) ; serta biaya 6 pada (S_2, D_1) dan (S_3, D_1) . Ketiganya berulang tepat dua kali (repetisi maksimum pada kasus ini, Tabel 4.18), berkontribusi pada 6 dari 9 sel matriks yang memiliki *twin cost*, kepadatan tertinggi kedua setelah S4 di antara keempat kasus kegagalan.

Tabel 4.27 Perbandingan Alokasi RVM dan Optimal pada n33

Sel (biaya)	Alokasi RVM	Alokasi Optimal
(S_1, D_2) (3)	90	90
(S_2, D_2) (5)	0	30
(S_2, D_3) (4)	80	50
(S_3, D_1) (6)	70	70
(S_3, D_2) (10)	30	0
(S_3, D_3) (7)	0	30
Total biaya	1.310	1.250

Kasus S4 (11×12 , $Z^* = 92.212$, RVM=94.060) berbeda dari ketiga kasus sebelumnya karena ukurannya yang jauh lebih besar (132 sel) membuat matriks penuh tidak praktis dita-

mpilkan dan ditandai sel per sel; pola kegagalannya diringkas dari trace eksekusi. Kepadatan *twin cost* pada kasus ini adalah yang tertinggi di antara keempat kasus kegagalan, yakni 121 dari 132 entri biaya memiliki pasangan bernilai sama, dengan satu nilai (81) muncul hingga tujuh kali (Tabel 4.18). Kepadatan ini memicu kesamaan penalti berulang kali pada lima iterasi pertama, dan aturan baris-dulu secara konsisten mengunci alokasi pada baris-baris awal sub-matriks. Iterasi keenam hingga kesebelas mengakumulasi sisa alokasi pada dua sel berbiaya tinggi, menghasilkan total biaya RVM 94.060, deviasi 2,00% terhadap optimal. Pola ini konsisten dengan n05 dan n33: kepadatan *twin cost* yang tinggi memaksa keputusan pemecah seri bekerja berulang kali, dan sifat konstruksi satu-lintasan RVM tanpa peninjauan ulang membuat akumulasi kecil dari tiap keputusan itu tidak dapat dikoreksi selama fase pembentukan solusi awal.

4.2.3 Sintesis Komparasi Karakteristik Matriks

Analisis per-iterasi dan pola kegagalan pada Subbab 4.2.1 dan 4.2.2 menunjukkan dua kelompok karakteristik matriks biaya yang menentukan keberhasilan RVM, dirangkum pada Tabel 4.28.

Tabel 4.28 Komparasi Karakteristik Matriks pada Kondisi Optimal vs Tidak Optimal

Faktor		RVM Cenderung Optimal	RVM Berisiko Tidak Optimal
Kepadatan <i>twin cost</i>		Rendah hingga sedang; gradien biaya bervariasi jelas antarbaris dan antarkolom	Tinggi; banyak sel bernilai $\bar{c}_{ij} = 0$ serentak setelah reduksi
Sinyal penalti Vogel		Selisih s_1 dan s_2 lebar, sehingga <i>regret</i> yang diukur kuat dan tegas mengarahkan ke sel <i>tight</i> krusial	Selisih s_1 dan s_2 sempit atau nol pada banyak lini, memicu ambiguitas seri penalti berulang kali
Peran <i>tie-breaking</i>		Jarang aktif; keputusan alokasi murni dari perbedaan biaya	Sering aktif; aturan <i>greedy</i> seperti baris-dulu mengambil keputusan lokal yang belum tentu selaras dengan basis optimal global
Struktur khusus biaya		Tidak ada kesenjangan ekstrem antarsel	Satu sel jauh lebih murah daripada sekitarnya (seperti n10), memancing komitmen berlebihan pada satu sel tanpa mempertimbangkan alternatif pembagian pasokan
Dampak sifat satu-lintasan		Tidak relevan, karena tidak ada keputusan yang perlu dikoreksi	Keputusan <i>tie-breaking</i> yang meleset di awal tidak dapat dikoreksi selama konstruksi, meski deviasi akhirnya tetap kecil

RVM unggul dan dominan pada mayoritas persoalan transportasi umum, dan hanya mengalami deviasi biaya kecil pada matriks yang memuat kepadatan *twin cost* tinggi atau kesen-

jangan biaya ekstrem. Hasil ini sejalan dengan *No Free Lunch Theorem*: keunggulan pada mayoritas kasus dibayar dengan ketakoptimalan pada sebagian kecil kasus berstruktur khusus, dan tidak ada satu metode konstruksi yang optimal untuk semua jenis persoalan.

4.3 Hasil pada Dataset Uji Tambahan (Dataset Uji 2 dan Dataset Uji 3)

Bagian ini menyajikan hasil pengujian pada dua himpunan data tambahan, Dataset Uji 2 (27 kasus, Subbab 3.2.2) dan Dataset Uji 3 (640 kasus sintetis, Subbab 3.2.2), yang dilaporkan **terpisah** dari hasil 42 kasus Dataset Uji 1 pada Subbab 4.1–4.2 di atas. Pengujian ini juga menyertakan IHW (Subbab 3.2.5) sebagai metode pembandingan kedelapan, sehingga total sembilan metode diuji: RVM dan delapan metode pembandingan (VAM, TOCM-MT, BCE, JHM-M, SSM, CSM, RBSM, IHW).

4.3.1 Akurasi dan Deviasi pada Dataset Uji 2

Tabel 4.29 merangkum akurasi (AP), deviasi rata-rata (DP), dan *runtime* rata-rata seluruh sembilan metode pada 27 kasus Dataset Uji 2.

Tabel 4.29 Hasil Sembilan Metode pada Dataset Uji 2 (27 Kasus)

Metode	Optimal	AP (%)	DP (%)	Runtime (ms)
VAM	14	51,85	3,51	31,60
TOCM-MT	16	59,26	3,92	23,91
BCE	13	48,15	5,11	24,20
JHM-M	14	51,85	3,65	31,56
SSM	14	51,85	3,88	28,88
CSM	10	37,04	8,33	33,93
RBSM	11	40,74	4,96	28,64
IHW	24	88,89	0,76	26,64
RVM	18	66,67	2,05	33,31

Berbeda dari hasil pada Dataset Uji 1 (Tabel 4.3) yang menempatkan RVM di posisi teratas, pada Dataset Uji 2 IHW justru unggul telak dengan AP 88,89% dan DP 0,76%, sementara RVM berada di posisi kedua dengan AP 66,67%. RVM tetap mengungguli ketujuh metode pembandingan lain dari garis VAM, TOCM-MT, BCE, JHM-M, SSM, CSM, dan RBSM, tetapi tidak lagi menjadi metode terbaik secara keseluruhan begitu IHW turut diuji. Pembahasan penyebab pergeseran posisi ini disajikan pada Subbab 4.3.4.

4.3.2 Akurasi dan Deviasi pada Dataset Uji 3

Tabel 4.30 merangkum hasil pada 640 kasus Dataset Uji 3.

Tabel 4.30 Hasil Sembilan Metode pada Dataset Uji 3 (640 Kasus)

Metode	Optimal	AP (%)	DP (%)	Runtime (ms)
VAM	397	62,03	1,14	37,59
TOCM-MT	397	62,03	1,66	15,84
BCE	397	62,03	0,84	14,35
JHM-M	399	62,34	0,47	15,01
SSM	399	62,34	0,57	14,90
CSM	397	62,03	0,64	14,83
RBSM	397	62,03	0,65	14,92
IHW	569	88,91	0,88	14,06
RVM	399	62,34	0,75	13,61

Pada skala 640 kasus, pola yang sama terlihat lebih tegas: IHW mencapai AP 88,91%, sedangkan RVM (62,34%) kini sejajar dengan JHM-M dan SSM, bahkan tidak lagi unggul telak seperti pada Dataset Uji 1. Meskipun demikian, DP RVM (0,75%) tetap kompetitif dan lebih baik dari VAM, TOCM-MT, dan RBSM, menunjukkan bahwa ketika RVM tidak mencapai optimal pada Dataset Uji 3, penyimpangannya tetap relatif kecil.

4.3.3 Analisis Runtime dan Kompleksitas Komputasi

Kolom *runtime* pada Tabel 4.29 dan 4.30 adalah rata-rata waktu eksekusi *wall-clock* tiap metode per kasus, diukur langsung dari proses kompilasi biner masing-masing metode (bukan estimasi teoretis semata), untuk melengkapi analisis kompleksitas $O(mn(m+n))$ RVM (Subbab 3.1.7) dengan bukti empiris pada skala matriks yang jauh lebih besar dari Dataset Uji 1.

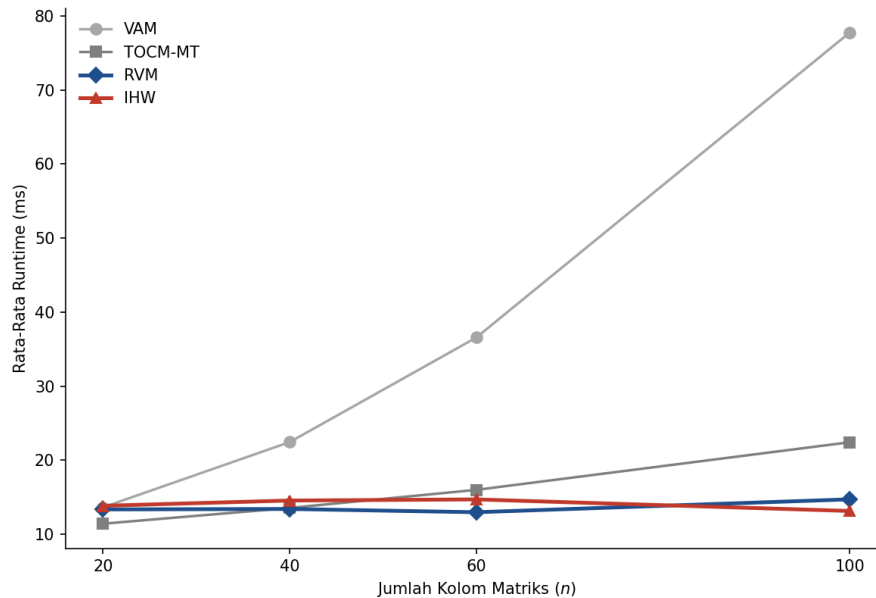
Tabel 4.31 menyajikan rata-rata *runtime* tiap metode pada empat kelompok ukuran Dataset Uji 3 (10×20 hingga 10×100), yang seluruhnya memiliki jumlah baris sama ($m = 10$) sehingga variabel yang berubah murni jumlah kolom n .

Tabel 4.31 Rata-Rata Runtime (ms) Menurut Jumlah Kolom pada Dataset Uji 3

Metode	$n = 20$	$n = 40$	$n = 60$	$n = 100$	Rasio ($n=100/n=20$)
VAM	13,63	22,44	36,56	77,74	5,71×
TOCM-MT	11,42	13,55	15,99	22,42	1,96×
BCE	12,25	13,79	14,98	16,38	1,34×
JHM-M	12,78	15,20	15,34	16,74	1,31×
SSM	13,33	14,11	15,59	16,57	1,24×
CSM	12,97	16,35	14,81	15,18	1,17×
RBSM	14,41	15,25	13,95	16,06	1,11×
IHW	13,84	14,55	14,70	13,14	0,95×
RVM	13,35	13,41	12,97	14,71	1,10×

Ketika jumlah kolom membesar 5 kali lipat (dari $n = 20$ ke $n = 100$), *runtime* VAM membesar 5,71 kali lipat, hampir sebanding lurus dengan pertambahan n itu sendiri, konsisten dengan implementasi VAM yang menghitung ulang penalti dari matriks biaya asli tanpa struktur data

pendukung untuk mempercepat pencarian dua nilai terkecil pada tiap iterasi. Sebaliknya, *runtime* RVM hanya membesar 1,10 kali lipat dan IHW bahkan sedikit menurun (0,95 kali, dalam batas wajar variasi pengukuran *wall-clock*), menunjukkan bahwa kompleksitas praktis keduanya jauh lebih landai terhadap penambahan kolom dibanding VAM. Gambar 4.3 memvisualisasikan pola pertumbuhan ini untuk empat metode yang paling representatif (VAM, TOCM-MT, RVM, IHW).



Gambar 4.3 Pertumbuhan Runtime Menurut Jumlah Kolom pada Dataset Uji 3

Pola serupa tampak pada Dataset Uji 2 yang rentang ukurannya lebih ekstrem (dari 3×2 hingga 30×32 sel): pada kasus terbesar (30×32 , 960 sel), VAM membutuhkan 41,45 ms berbanding RVM yang hanya 10,65 ms, sekitar 4 kali lebih cepat. Temuan ini memperkuat klaim kompleksitas $O(mn(m + n))$ pada Subbab 3.1.7: RVM tidak menambahkan biaya komputasi asimtotik dibanding metode klasik lain, dan pada praktiknya skala dengan lebih baik dibanding VAM ketika ukuran matriks membesar.

4.3.4 Karakteristik Struktur Biaya dan Performa RVM

Posisi RVM bergeser dari yang terbaik pada Dataset Uji 1 menjadi sejajar sebagian besar metode pembanding pada Dataset Uji 2 dan Dataset Uji 3. Tabel 4.32 merangkum *twin-cost ratio* dan AP RVM pada ketiga himpunan data untuk memperlihatkan polanya.

Tabel 4.32 Twin-Cost Ratio dan AP RVM pada Ketiga Himpunan Data

Himpunan	<i>Twin-Cost Ratio</i>	AP RVM (%)
Dataset Uji 1	52,3%	90,48
Dataset Uji 2	62,4%	66,67
Dataset Uji 3	98,6%	62,34

Urutan *twin-cost ratio* (Dataset Uji 1 < Dataset Uji 2 < Dataset Uji 3) berbanding terbalik dengan urutan AP RVM pada ketiga himpunan, sehingga kepadatan *twin cost*, bukan ukuran matriks atau rentang biaya semata, menjadi determinan utama performa RVM pada data tambahan

ini.

Mekanismenya sejalan dengan analisis kegagalan RVM pada Dataset Uji 1 (Subbab 4.2.2): makin banyak sel bernilai *reduced cost* sama dalam satu lini, makin sering keputusan alokasi bergantung pada aturan *tie-breaking* MRV dan baris-dulu (Subbab 3.1.5), bukan pada perbedaan biaya itu sendiri. Bedanya, pada Dataset Uji 3 kondisi ini terjadi *sistematis* di hampir seluruh sel matriks (98,6% biaya berulang), bukan pada segelintir sel seperti kasus kegagalan Dataset Uji 1 (n05, n10, n33, S4). Pemeriksaan pada 64 kelompok parameter Dataset Uji 3 (Subbab 3.2.3) juga tidak menunjukkan korelasi kuat antara jumlah replikasi optimal RVM per kelompok dengan rentang biaya R , tingkat variasi *supply-demand* V , maupun lebar matriks n ; kepadatan *twin cost* berasal dari cara pembangkitan seluruh himpunan Dataset Uji 3 secara acak seragam pada rentang nilai yang sempit relatif terhadap ukuran matriksnya (Subbab 3.2.2), bukan dari kombinasi parameter tertentu.

IHW tidak terpengaruh sebesar RVM oleh kepadatan *twin cost*: AP-nya 88,89% pada Dataset Uji 2 dan 88,91% pada Dataset Uji 3, hampir tidak berubah dari satu himpunan ke himpunan lain. Perbedaan ini berasal dari komponen $Weight = MC \times FA \times ICHM$ (Subbab 3.2.5), yang mempertimbangkan sebaran biaya satu lini penuh lewat ICHM, bukan hanya dua nilai terkecil seperti penalti Vogel pada RVM (Subbab 2.2.6). Ketika banyak biaya pada satu lini bernilai sama, penalti Vogel ($s_2 - s_1$) cenderung kecil atau nol untuk banyak lini sekaligus, sehingga prioritas alokasi RVM lebih sering jatuh ke aturan *tie-breaking*. ICHM tetap membedakan lini-lini tersebut dari sebaran biaya keseluruhannya, sehingga IHW mempertahankan daya beda antarlini meski *twin cost* padat.

Dengan demikian, keunggulan RVM pada Bab IV bagian utama (Subbab 4.1–4.2) berlaku kuat pada *benchmark* berasal dari kasus literatur, kasus sintesis dengan struktur biaya ekstrem yang disengaja, dan *real-world*, tetapi tidak serta-merta berlaku pada data sintesis acak seragam berkepadatan *twin cost* sangat tinggi seperti Dataset Uji 3. Ini sejalan dengan *No Free Lunch Theorem* (Subbab 4.2): keunggulan pada satu kelas persoalan dibayar dengan performa yang lebih biasa pada kelas persoalan lain, dan IHW pada Dataset Uji 2 dan Dataset Uji 3 menjadi bukti empirisnya. Salah satu arah pengembangan lanjutan yang relevan (Subbab 5.2) adalah mempertimbangkan sebaran biaya satu lini penuh, bukan hanya dua nilai terkecilnya, pada kondisi *twin cost* padat.

4.3.5 Verifikasi Silang Performa IHW pada Dataset Uji 1

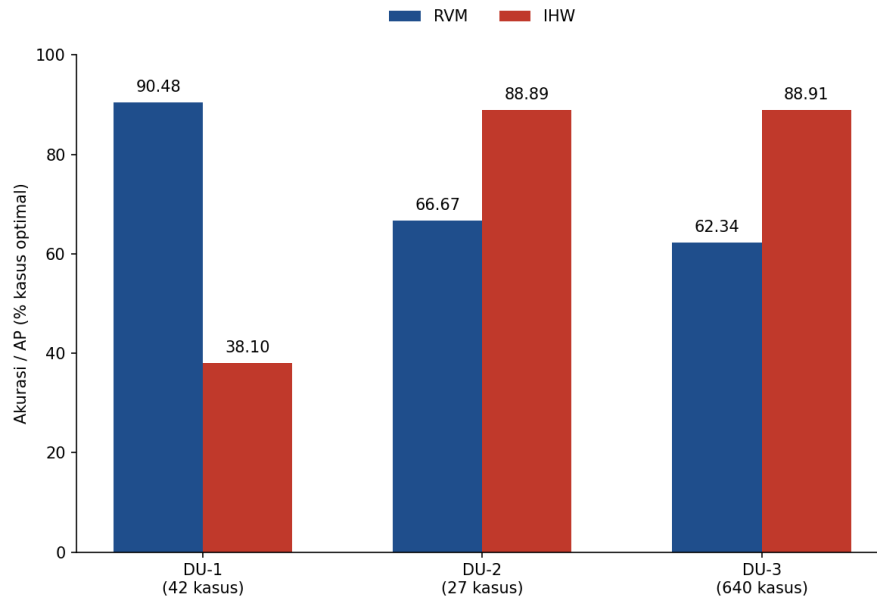
Agar perbandingan RVM dan IHW benar-benar setara (*apple-to-apple*) dan tidak hanya bertumpu pada Dataset Uji 2 dan Dataset Uji 3, IHW turut dijalankan pada 42 kasus Dataset Uji 1 dengan kondisi identik terhadap kedelapan metode pembandingan lain pada Tabel 4.3. Hasil ini dilaporkan terpisah sebagai verifikasi silang dan tidak menggantikan Tabel 4.3, karena IHW tidak termasuk metode pembandingan yang dipilih pada Subbab 2.1.

Tabel 4.33 Perbandingan RVM dan IHW pada Ketiga Himpunan Data

Himpunan	Jumlah Kasus	AP RVM (%)	AP IHW (%)	<i>twin-cost ratio</i>
Dataset Uji 1	42	90,48	38,10	52,3%
Dataset Uji 2	27	66,67	88,89	62,4%
Dataset Uji 3	640	62,34	88,91	98,6%

Tabel 4.33 menunjukkan pola yang sepenuhnya terbalik antara RVM dan IHW pada Dataset

Uji 1: RVM unggul telak (AP 90,48%) sementara IHW justru menjadi metode terlemah pada himpunan ini (AP 38,10%, DP rata-rata 6,81%), jauh di bawah bahkan VAM klasik pada Tabel 4.3 (52,38%). Pola ini adalah kebalikan sempurna dari Dataset Uji 2 dan Dataset Uji 3, tempat IHW unggul telak dan RVM sejajar mayoritas metode pembandingan. Gambar 4.4 memvisualisasikan pola silang ini pada ketiga himpunan data sekaligus.



Gambar 4.4 Perbandingan Akurasi RVM dan IHW pada Dataset Uji 1, Dataset Uji 2, dan Dataset Uji 3

Pola ini menguatkan kesimpulan pada Subbab 4.3.4: keunggulan suatu metode konstruksi IBFS bersifat spesifik terhadap karakteristik struktur biaya *benchmark*, bukan sifat universal metode itu sendiri. RVM, yang berbasis reduksi *zero-point* dan penalti dua-nilai, unggul pada Dataset Uji 1 yang biayanya berasal dari *real-world* dan literatur dengan variasi nilai lebih alami (*twin-cost ratio* 52,3%); IHW, yang berbasis sebaran biaya penuh lewat ICHM, unggul pada Dataset Uji 2 dan Dataset Uji 3 yang kepadatan *twin cost*nya jauh lebih tinggi (62,4% dan 98,6%). Tidak ada satu metode yang mendominasi pada ketiga himpunan sekaligus. Ini demonstrasi *No Free Lunch Theorem* yang lebih kuat daripada yang bisa ditunjukkan RVM sendirian terhadap ketujuh metode pembandingan pada Tabel 4.3, karena di sini kedua metode yang dibandingkan silih berganti menjadi yang terbaik dan yang terlemah, tergantung himpunan data yang diuji.

Halaman ini sengaja dikosongkan

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian dan pembahasan, dapat ditarik kesimpulan berikut sesuai dengan rumusan masalah.

1. Penelitian ini berhasil merancang *Reduced-cost Vogel Method* (RVM), yaitu metode konstruksi IBFS satu-langkah yang menghitung *penalty* Vogel di atas matriks hasil reduksi *zero-point*. Setiap tahapan metode memiliki landasan matematis yang jelas, yakni reduksi baris dan kolom membentuk pasangan *dual* layak sehingga matriks tereduksi tepat berisi *reduced cost* (Proposisi 2.2 dan 2.3), *penalty* selisih dua terkecil merupakan ukuran *regret* yang setara dengan *reduced cost* terkecil kedua, dan alokasi ke sel bernilai nol konsisten dengan *complementary slackness*. Metode bersifat deterministik, berjalan dalam satu lintasan, dan berhenti setelah paling banyak $m + n - 1$ alokasi.
2. Pada *benchmark* 42 kasus, RVM mencapai solusi optimal pada 38 kasus (90,48%) dengan deviasi rata-rata 0,40%, mengungguli ketujuh metode terpublikasi yang dibandingkan pada kondisi pengujian yang identik, yaitu VAM (52,38%), TOCM-MT (61,90%), BCE (69,05%), JHM-M (73,81%), SSM (73,81%), CSM (78,57%), dan RBSM (80,95%).
3. Studi ablasi terhadap 15 fungsi *penalty* menunjukkan bahwa *penalty* selisih dua nilai terkecil dengan duplikat dihitung, yang dipakai RVM, menghasilkan akurasi tertinggi (90,48%), dan bahkan fungsi *penalty* sederhana tetap jauh mengungguli VAM standar. Hal ini membuktikan bahwa sumber utama keunggulan RVM adalah reduksi *zero-point*, bukan bentuk *penalty* tertentu, sekaligus mengonfirmasi bahwa pilihan *penalty* berbasis *regret*, termasuk keputusan menghitung *twin cell* sebagai s_1 dan s_2 , adalah yang terbaik secara teoretis maupun empiris.
4. Pengujian tambahan pada 667 kasus di luar *benchmark* utama (Dataset Uji 2 dan Dataset Uji 3) beserta metode pembanding kedelapan (IHW) menunjukkan bahwa keunggulan RVM tidak bersifat universal. RVM tetap unggul pada *benchmark* 42 kasus utama (90,48%), tetapi sejajar mayoritas metode pembanding pada Dataset Uji 2 (66,67%) dan Dataset Uji 3 (62,34%); IHW menunjukkan pola sebaliknya (38,10% pada 42 kasus utama berbanding 88,89–88,91% pada Dataset Uji 2 dan Dataset Uji 3). Pergeseran ini berkorelasi kuat dengan *twin-cost ratio* tiap himpunan data (52,3% pada 42 kasus utama berbanding 98,6% pada Dataset Uji 3): keunggulan suatu metode konstruksi IBFS bergantung pada karakteristik struktur biaya *benchmark* yang diujikan, sejalan dengan *No Free Lunch Theorem*.

5.2 Saran

Beberapa saran untuk penelitian selanjutnya adalah sebagai berikut.

1. Mengevaluasi RVM sebagai inisialisasi bagi prosedur optimisasi lanjutan seperti MODI untuk mengukur pengurangan jumlah iterasi menuju solusi optimal.
2. Memperluas pengujian ke himpunan kasus acak berskala lebih besar dan beragam guna memperkuat bukti generalisasi, sebagaimana studi 640 *instance* oleh Mathirajan et al. (2022) – saran ini telah ditindaklanjuti sebagian pada penelitian ini melalui pengujian tambahan 667 kasus (Dataset Uji 2 dan Dataset Uji 3), namun hasil pada Subbab 4.3.4 menunjukkan bahwa perluasan pengujian tetap perlu dilanjutkan pada himpunan data dengan karakteristik *twin cost* rendah agar generalisasi keunggulan RVM lebih menyeluruh teruji.
3. Mengembangkan aturan *tie-breaking* yang lebih cerdas pada matriks tereduksi untuk mengurangi kegagalan pada kasus dengan banyak biaya bernilai sama.
4. Mengkaji perluasan RVM ke *transportation problem* dengan ketidakpastian biaya (*fuzzy*) atau *solid transportation problem*, sejalan dengan arah penelitian terkini (Shivani dan Rani, 2024).

DAFTAR PUSTAKA

- Ahmed, M. M., Islam, M. A., Katun, M., Yesmin, S., & Uddin, M. S. (2015). New procedure of finding an initial basic feasible solution of the time minimizing transportation problems. *Open Journal of Applied Sciences*, 5(10), 634–640. <https://doi.org/10.4236/ojapps.2015.510062>
- Ahmed, M. M., Khan, A. R., Uddin, M. S., & Ahmed, F. (2016). A new approach to solve transportation problems. *Open Journal of Optimization*, 5(1), 22–30.
- Amaliah, B., Fatichah, C., & Suryani, E. (2019). Total opportunity cost matrix – Minimal total: A new approach to determine initial basic feasible solution of a transportation problem. *Egyptian Informatics Journal*, 20(2), 131–141. <https://doi.org/10.1016/j.eij.2019.01.002>
- Amaliah, B., Fatichah, C., & Suryani, E. (2022a). A new heuristic method of finding the initial basic feasible solution to solve the transportation problem. *Journal of King Saud University – Computer and Information Sciences*, 34(5), 2298–2307. <https://doi.org/10.1016/j.jksuci.2020.07.007>
- Amaliah, B., Fatichah, C., & Suryani, E. (2022b). A supply selection method for better feasible solution of balanced transportation problem. *Expert Systems with Applications*, 203, 117399. <https://doi.org/10.1016/j.eswa.2022.117399>
- Amreen, M. (2024). A new way for solving transportation issues. *Journal of Applied Mathematics & Informatics*, 42(3), 647–661.
- Amreen, M., & Venkateswarlu, B. (2024). An innovative method to develop the initial basic feasible solution to a transportation issue. *African Journal of Biological Sciences*, 6(14), 348–363.
- Babu, M. A., Hoque, M. A., & Uddin, M. S. (2020). A heuristic for obtaining better initial feasible solution to the transportation problem. *OPSEARCH*, 57(1), 221–245. <https://doi.org/10.1007/s12597-019-00429-5>
- Bazaraa, M. S., Jarvis, J. J., & Sherali, H. D. (2010). *Linear programming and network flows* (4th ed.). John Wiley & Sons.
- Boah, D. K., Fiele, S. A., & Etwire, C. J. (2026). A new algorithm for finding initial basic feasible solutions of transportation problems. *AppliedMath*, 6(4), 58. <https://doi.org/10.3390/appliedmath6040058>
- Bunyamin, R. F., Amaliah, B., & Saikhu, A. (2026a). A cost supply method for finding an initial basic feasible solution of the transportation problem. *Engineering, Technology & Applied Science Research*, 16(2), 33024–33030. <https://doi.org/10.48084/etasr.16468>
- Bunyamin, R. F., Amaliah, B., & Saikhu, A. (2026b). Enhanced supply selection for better initial basic feasible solutions in transportation problems. *MethodsX*, 16, 103825. <https://doi.org/10.1016/j.mex.2026.103825>
- Charnes, A., & Cooper, W. W. (1954). The stepping stone method of explaining linear pro-

gramming calculations in transportation problems. *Management Science*, 1(1), 49–69. <https://doi.org/10.1287/mnsc.1.1.49>

- Dantzig, G. B. (1963). *Linear programming and extensions*. Princeton University Press.
- Das, U. K., Babu, R. A., Khan, A., & Uddin, M. S. (2014). Logical development of Vogel's approximation method (LD-VAM): An approach to find basic feasible solution of transportation problem. *International Journal of Scientific & Technology Research*, 3(2).
- Goyal, S. K. (1984). Improving VAM for unbalanced transportation problems. *Journal of the Operational Research Society*, 35(12), 1113–1114. <https://doi.org/10.1057/jors.1984.217>
- Gupta, P. (2020). A new method to find basic feasible solution of transportation problem. *Journal of Xidian University*, 14(10). <https://doi.org/10.37896/jxu14.10/083>
- Hitchcock, F. L. (1941). The distribution of a product from several sources to numerous localities. *Journal of Mathematics and Physics*, 20(1–4), 224–230. <https://doi.org/10.1002/sapm1941201224>
- Haralick, R. M., & Elliott, G. L. (1980). Increasing tree search efficiency for constraint satisfaction problems. *Artificial Intelligence*, 14(3), 263–313. [https://doi.org/10.1016/0004-3702\(80\)90051-X](https://doi.org/10.1016/0004-3702(80)90051-X)
- Hussein, H. A., & Shiker, M. A. K. (2020). A modification to Vogel's approximation method to solve transportation problems. *Journal of Physics: Conference Series*, 1591, 012029. <https://doi.org/10.1088/1742-6596/1591/1/012029>
- Juman, Z. A. M. S., & Hoque, M. A. (2015). An efficient heuristic to obtain a better initial feasible solution to the transportation problem. *Applied Soft Computing*, 34, 813–826. <https://doi.org/10.1016/j.asoc.2015.05.009>
- Kacher, Y., & Singh, P. (2021). A comprehensive literature review on transportation problems. *International Journal of Applied and Computational Mathematics*, 7(5), 206. <https://doi.org/10.1007/s40819-021-01134-y>
- Karagül, K., & Şahin, Y. (2020). A novel approximation method to obtain initial basic feasible solution of transportation problem. *Journal of King Saud University – Engineering Sciences*, 32(3), 211–218. <https://doi.org/10.1016/j.jksues.2019.03.003>
- Kaur, L., Rakshit, M., & Singh, S. (2018). An improvement in maximum difference method to find initial basic feasible solution for transportation problem. *International Journal of Computer Sciences and Engineering*, 6(9), 214–220. <https://doi.org/10.26438/ijcse/v6i9.533535>
- Kirca, Ö., & Şatir, A. (1990). A heuristic for obtaining an initial solution for the transportation problem. *Journal of the Operational Research Society*, 41(9), 865–871. <https://doi.org/10.1057/jors.1990.124>
- Koopmans, T. C. (1949). Optimum utilization of the transportation system. *Econometrica*, 17, 136–146. <https://doi.org/10.2307/1907301>
- Korukoğlu, S., & Balli, S. (2011). An improved Vogel's approximation method for the transportation problem. *Mathematical and Computational Applications*, 16(2), 370–381. <https://doi.org/10.3390/mca16020370>
- Kousalya, P., & Malarvizhi, P. (2016). A new technique for finding initial basic feasible solu-

- tion to transportation problem. *International Journal of Engineering and Management Research*, 6(3).
- Kuhn, H. W. (1955). The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1–2), 83–97. <https://doi.org/10.1002/nav.3800020109>
- Mathirajan, M., & Meenakshi, B. (2004). Experimental analysis of some variants of Vogel's approximation method. *Asia-Pacific Journal of Operational Research*, 21(4), 447–462. <https://doi.org/10.1142/S0217595904000333>
- Mathirajan, M., Reddy, S., & Vimala Rani, M. (2022). An experimental study of newly proposed initial basic feasible solution methods for a transportation problem. *OPSEARCH*, 59(1), 102–145. <https://doi.org/10.1007/s12597-021-00533-5>
- Murugesan, R., & Esakkiammal, T. (2020). TOCM-VAM method versus ASM method in transportation problems. *Advances in Mathematics: Scientific Journal*, 9(6), 3549–3566. <https://doi.org/10.37418/amsj.9.6.34>
- Pandian, P., & Natarajan, G. (2010). A new method for finding an optimal solution for transportation problems. *International Journal of Mathematical Sciences and Engineering Applications*, 4(2), 59–65.
- Patel, R. G., Bhathawala, P. H., & Patel, B. S. (2017). An alternate approach to find an optimal solution of a transportation problem. *IOSR Journal of Mathematics*, 13(1), 1–5. <https://doi.org/10.9790/5728-1301030105>
- Rafi, F. S., & Islam, S. (2020). A comparative study of solving methods of transportation problem in linear programming problem. *Journal of Advances in Mathematics and Computer Science*, 35(5), 45–67. <https://doi.org/10.9734/jamcs/2020/v35i530281>
- Ramadan, S. Z., & Ramadan, I. Z. (2012). Hybrid two-stage algorithm for solving transportation problem. *Modern Applied Science*, 6(4), 12–22. <https://doi.org/10.5539/mas.v6n4p12>
- Reinfeld, N. V., & Vogel, W. R. (1958). *Mathematical programming* (pp. 59–70). Prentice-Hall.
- Reza, A. K. M. S., Jamali, M. J. U., & Biswas, B. (2019). A modified algorithm for solving unbalanced transportation problems. *Journal of Engineering Science*, 10(1), 93–101.
- Russell, E. J. (1969). Extension of Dantzig's algorithm to finding an initial near-optimal basis for the transportation problem. *Operations Research*, 17(1), 187–191. <https://doi.org/10.1287/opre.17.1.187>
- Russell, S. J., & Norvig, P. (2021). *Artificial intelligence: A modern approach* (4th ed.). Pearson.
- Sahito, S. (2021). Modification of Vogel's approximation method for optimality of transportation problem by statistical technique. *Quaid-e-Awam University Research Journal of Engineering, Science & Technology*, 19(2), 42–48. <https://doi.org/10.52584/qrj.1902.07>
- Samuel, A. E. (2012). Improved zero point method (IZPM) for the transportation problems. *Applied Mathematical Sciences*, 6(109), 5421–5426.
- Shivani, & Rani, D. (2024). An extended Vogel's approximation algorithm for efficiently solving Fermatean fuzzy solid transportation problems. *Soft Computing*, 28(17–18), 9711–9734. <https://doi.org/10.1007/s00500-024-09812-x>

- Sultan, A. (1988). Heuristic for finding an initial basic feasible solution in transportation problems. *OPSEARCH*, 25, 197–199.
- Taha, H. A. (2017). *Operations research: An introduction* (10th ed.). Pearson.
- Wireko, F. A., Mensah, I. D. K., Aborhey, E. N. A., Appiah, S. A., Sebil, C., & Ackora-Prah, J. (2025). The maximum range method for finding initial basic feasible solution for transportation problems. *Results in Control and Optimization*, 19, 100551. <https://doi.org/10.1016/j.rico.2025.100551>
- Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1), 67–82. <https://doi.org/10.1109/4235.585893>

LAMPIRAN

Lampiran A: Kode Sumber Inti RVM (Bahasa C)

Berikut rutin inti implementasi RVM: reduksi *zero-point* sekuensial (*build_reduced*), fungsi *penalty* $s_2 - s_1$ (*line_penalty*), dan langkah konstruksi beserta aturan *tie-breaking*. Kode lengkap beserta program penelusuran (*tracer*) per iterasi disertakan sebagai berkas pen-damping.

```
/* Reduksi zero-point SEKUENSIAL pada submatriks aktif:
   kurangi minimum baris, lalu kurangi minimum kolom (dari hasil
   reduksi baris). Setelah satu putaran, tiap baris & kolom punya 0. */
static void build_reduced(const int rm[], const int cm[]) {
    for (int i=0; i<M; i++) for (int j=0; j<N; j++)
        R[i][j] = (rm[i]&&cm[j]) ? COST[i][j] : INF;
    for (int pass=0; pass<M+N; ++pass) {
        int changed=0;
        for (int i=0; i<M; i++){ if(!rm[i]) continue; /* min baris */
            long long mn=INF;
            for (int j=0; j<N; j++) if(cm[j]&&R[i][j]<mn) mn=R[i][j];
            if(mn>0&&mn<INF){ for(int j=0; j<N; j++) if(cm[j]) R[i][j]-=mn; changed=1; } }
        for (int j=0; j<N; j++){ if(!cm[j]) continue; /* min kolom */
            long long mn=INF;
            for (int i=0; i<M; i++) if(rm[i]&&R[i][j]<mn) mn=R[i][j];
            if(mn>0&&mn<INF){ for(int i=0; i<M; i++) if(rm[i]) R[i][j]-=mn; changed=1; } }
        if(!changed) break;
    }
}

/* Penalty = selisih dua nilai tereduksi terkecil (s2 - s1). */
static double line_penalty(double *v, int k){
    if(k<=0) return -1.0;
    if(k==1) return v[0];
    qsort(v, k, sizeof(double), cmp_d);
    return v[1]-v[0];
}

/* Inti konstruksi: tiap langkah pilih lini penalty maksimum
   (baris didahulukan), lalu sel reduced-cost minimum dengan
   \textit{tie-breaking}: alokasi terbesar -> biaya asli terkecil -> indeks. */
while (ada_baris_aktif && ada_kolom_aktif) {
    build_reduced(rm, cm);
    best_pen = -INF; kind = -1; idx = -1;
    for (i : baris aktif) { p = line_penalty(R[i][*], k);
        if (p > best_pen) { best_pen=p; kind=0; idx=i; } }
    for (j : kolom aktif) { p = line_penalty(R[*][j], k);
        if (p > best_pen) { best_pen=p; kind=1; idx=j; } } /* > : baris menang seri */
    /* pada lini terpilih: min R; seri -> max min(sup,dem); -> min COST; -> min indeks */
    pilih_sel(idx, kind, &ci, &cj);
    x = min(sup[ci], dem[cj]);
    alloc[ci][cj] = x; sup[ci]-=x; dem[cj]-=x;
    if(sup[ci]==0) rm[ci]=0;
```

```

    if (dem[cj]==0) cm[cj]=0;
}

```

Lampiran B: Hasil Lengkap Studi Ablasi Fungsi Penalti

Tabel B.1 menyajikan hasil pengujian seluruh 15 fungsi *penalty* dengan kerangka RVM yang sama (reduksi *zero-point* sekuensial dan aturan *tie-breaking* yang identik), diurutkan menurut akurasi. *Penalty* selisih dua nilai terkecil dengan duplikat dihitung ($s_2 - s_1$) – yang dipakai RVM sendiri – memberi hasil terbaik. Notasi yang dipakai pada tabel: s_1 dan s_2 adalah dua nilai tereduksi terkecil pada lini (varian berbeda melewati duplikat), l_1 dan l_2 adalah dua nilai tereduksi terbesar (sepadan dengan s_1, s_2), v_i adalah nilai tereduksi tiap sel aktif, $\bar{v}$ rata-ratanya, dan k jumlah sel aktif; jumlah sel aktif taknol dipakai pada formula rata-rata geometris dan harmonik.

Tabel B.1 Hasil Studi Ablasi 15 Fungsi Penalti pada 42 Kasus

No.	Nama Penalti	Optimal	Akurasi (%)	Deviasi (%)
1	Selisih Dua Cost Terkecil dengan Nilai Sama Tetap Dihitung	38	90,48	0,40
2	Median Cost - Cost Terendah	35	83,33	0,57
3	Rata-rata Selisih Cost terhadap Cost Terendah	32	76,19	1,00
4	Rata-rata Cost	32	76,19	1,00
5	Cost Tertinggi - Cost Terendah	30	71,43	1,73
6	Total Selisih Cost terhadap Cost Terendah	29	69,05	1,86
7	Contra Harmonic Mean	27	64,29	2,38
8	Geometric Mean	27	64,29	3,17
9	Standar Deviasi Cost	26	61,90	2,19
10	Harmonic Mean	25	59,52	4,16
11	Selisih Dua Cost Terkecil yang Berbeda	25	59,52	3,83
12	Selisih Dua Cost Terbesar yang Berbeda	21	50,00	3,96
13	Selisih Dua Cost Terbesar dengan Nilai Sama Tetap Dihitung	20	47,62	4,17
14	Inverse Harmonic Mean	8	19,05	9,83
15	Inverse Contra-Harmonic Mean	7	16,67	10,97

Hasil ini menegaskan bahwa keunggulan RVM bersumber terutama pada reduksi *zero-point*, bukan semata bentuk *penalty*: bahkan *penalty* sederhana seperti rata-rata *cost* (32 optimal) masih jauh mengungguli VAM klasik (22 optimal), sementara *penalty* Vogel ($s_2 - s_1$, duplikat dihitung) di atas matriks tereduksi memberi hasil terbaik. Skrip yang mereproduksi tabel ini (RVM-benchmark/ablas_i_penalty.py dan gen_tabel_ablas_i.py) disertakan sebagai berkas pendamping untuk menjamin keterulangan hasil.

Lampiran C: Matriks Lengkap Seluruh Kasus Uji Benchmark

Lampiran ini menyajikan matriks biaya, *supply*, dan *demand* lengkap untuk 39 dari 42 kasus uji *benchmark* (kasus n01, n02, dan n11 sudah ditampilkan sebagai contoh pada Bab III sehingga tidak diulang di sini), agar seluruh data masukan dapat ditelusuri dan direproduksi tanpa merujuk berkas eksternal.

Kasus literatur lainnya (selain n01, n02, n11 yang sudah ditampilkan pada Bab III).

Tabel C.1 Instance n03 (5×5), $Z^* = 1102$

	D_1	D_2	D_3	D_4	D_5	Supply
S_1	73	40	9	79	20	8
S_2	62	93	96	8	13	7
S_3	96	65	80	50	65	9
S_4	57	58	29	12	87	3
S_5	56	23	87	18	12	5
Demand	6	8	10	4	4	

Tabel C.2 Instance n04 (3×5), $Z^* = 1580$

	D_1	D_2	D_3	D_4	D_5	Supply
S_1	6	4	4	7	5	100
S_2	5	6	7	4	8	125
S_3	3	4	6	3	4	175
Demand	60	80	85	105	70	

Tabel C.3 Instance n05 (4×4), $Z^* = 410$

	D_1	D_2	D_3	D_4	Supply
S_1	7	5	9	11	30
S_2	4	3	8	6	25
S_3	3	8	10	5	20
S_4	2	6	7	3	15
Demand	30	30	20	10	

Tabel C.4 Instance n06 (4×4), $Z^* = 7750$

	D_1	D_2	D_3	D_4	Supply
S_1	10	8	4	3	500
S_2	12	14	20	2	400
S_3	6	9	23	25	300
S_4	0	0	0	0	150
Demand	250	350	600	150	

Tabel C.5 Instance n07 (3×4), $Z^* = 1160$

	D_1	D_2	D_3	D_4	Supply
S_1	6	1	9	3	70
S_2	11	5	2	8	55
S_3	10	12	4	7	90
Demand	85	35	50	45	

Tabel C.6 Instance n09 (4×4), $Z^* = 111$

	D_1	D_2	D_3	D_4	Supply
S_1	4	6	5	2	6
S_2	6	4	1	4	10
S_3	5	2	3	1	12
S_4	4	6	7	8	14
Demand	9	16	10	7	

Tabel C.7 Instance n10 (3×4), $Z^* = 910$

	D_1	D_2	D_3	D_4	Supply
S_1	20	2	20	11	30
S_2	24	7	9	20	50
S_3	8	14	16	18	20
Demand	10	30	30	30	

Tabel C.8 Instance n12 (3×3), $Z^* = 425$

	D_1	D_2	D_3	Supply
S_1	15	7	25	12
S_2	8	12	14	17
S_3	17	19	21	7
Demand	12	10	14	

Tabel C.9 Instance n13 (3×3), $Z^* = 4525$

	D_1	D_2	D_3	Supply
S_1	6	8	10	150
S_2	7	11	11	175
S_3	4	5	12	275
Demand	200	100	300	

Tabel C.10 Instance n14 (3×4), $Z^* = 920$

	D_1	D_2	D_3	D_4	Supply
S_1	4	6	8	8	40
S_2	6	8	6	7	60
S_3	5	7	6	8	50
Demand	20	30	50	50	

Tabel C.11 Instance n15 (3×4), $Z^* = 156$

	D_1	D_2	D_3	D_4	Supply
S_1	4	6	9	5	16
S_2	2	6	4	1	12
S_3	5	7	2	9	15
Demand	12	14	9	8	

Tabel C.12 Instance n16 (3×4), $Z^* = 1510$

	D_1	D_2	D_3	D_4	Supply
S_1	22	17	25	15	30
S_2	13	10	16	20	50
S_3	16	19	9	5	50
Demand	30	30	50	20	

Tabel C.13 Instance n17 (3×4), $Z^* = 743$

	D_1	D_2	D_3	D_4	Supply
S_1	19	30	50	10	7
S_2	70	30	40	60	9
S_3	40	8	70	20	18
Demand	5	8	7	14	

Tabel C.14 Instance n18 (3×4), $Z^* = 1650$

	D_1	D_2	D_3	D_4	Supply
S_1	6	10	14	0	50
S_2	12	19	21	0	50
S_3	15	14	17	0	50
Demand	30	40	55	25	

Tabel C.15 Instance n21 (3×3), $Z^* = 4205$

	D_1	D_2	D_3	Supply
S_1	135	71	260	12
S_2	81	126	144	17
S_3	170	189	201	7
Demand	12	10	14	

Tabel C.16 Instance n22 (3×4), $Z^* = 12075$

	D_1	D_2	D_3	D_4	Supply
S_1	11	13	17	14	250
S_2	16	18	14	10	300
S_3	21	24	13	10	400
Demand	200	225	275	250	

Tabel C.17 Instance n23 (3×4), $Z^* = 23$

	D_1	D_2	D_3	D_4	Supply
S_1	7	2	3	7	2
S_2	1	5	4	5	1
S_3	7	2	4	5	3
Demand	2	1	1	2	

Tabel C.18 Instance n24 (3×4), $Z^* = 3460$

	D_1	D_2	D_3	D_4	Supply
S_1	20	22	17	4	120
S_2	24	37	9	7	70
S_3	32	37	20	15	50
Demand	60	40	30	110	

Tabel C.19 Instance n25 (3×4), $Z^* = 809$

	D_1	D_2	D_3	D_4	Supply
S_1	19	30	50	12	7
S_2	70	30	40	60	10
S_3	40	10	60	20	18
Demand	5	7	8	15	

Tabel C.20 Instance n26 (4×5), $Z^* = 490$

	D_1	D_2	D_3	D_4	D_5	Supply
S_1	20	4	6	8	9	20
S_2	4	10	8	18	8	30
S_3	9	11	20	40	6	15
S_4	7	6	9	14	16	13
Demand	40	6	8	18	6	

Tabel C.21 Instance n27 (5×5), $Z^* = 59356$

	D_1	D_2	D_3	D_4	D_5	Supply
S_1	46	74	9	28	99	461
S_2	12	75	6	36	48	277
S_3	35	199	4	5	71	356
S_4	61	81	44	88	9	488
S_5	85	60	14	25	79	393
Demand	278	60	461	116	1060	

Tabel C.22 Instance n28 (4×4), $Z^* = 2090$

	D_1	D_2	D_3	D_4	Supply
S_1	3	1	7	4	300
S_2	2	6	5	9	400
S_3	8	3	3	2	270
S_4	0	0	0	0	370
Demand	250	350	290	450	

Tabel C.23 Instance n29 (3×4), $Z^* = 3857$

	D_1	D_2	D_3	D_4	Supply
S_1	15	1	42	33	23
S_2	80	42	26	81	44
S_3	90	40	66	60	33
Demand	23	31	16	30	

Tabel C.24 Instance n30 (3×3), $Z^* = 465$

	D_1	D_2	D_3	Supply
S_1	13	21	14	13
S_2	8	12	21	20
S_3	15	17	19	5
Demand	12	15	11	

Tabel C.25 Instance n31 (3×4), $Z^* = 1480$

	D_1	D_2	D_3	D_4	Supply
S_1	15	27	13	19	40
S_2	18	21	24	14	40
S_3	21	15	16	17	20
Demand	30	20	30	20	

Tabel C.26 Instance n32 (4×5), $Z^* = 1780$

	D_1	D_2	D_3	D_4	D_5	Supply
S_1	4	4	9	10	13	100
S_2	7	9	8	10	4	90
S_3	9	3	7	10	6	80
S_4	11	4	10	6	9	70
Demand	60	40	90	70	80	

Tabel C.27 Instance n33 (3×3), $Z^* = 1250$

	D_1	D_2	D_3	Supply
S_1	3	3	5	90
S_2	6	5	4	80
S_3	6	10	7	100
Demand	70	120	80	

Tabel C.28 Instance n34 (4×5), $Z^* = 510$

	D_1	D_2	D_3	D_4	D_5	Supply
S_1	7	6	4	5	9	40
S_2	8	5	6	7	8	30
S_3	6	8	9	6	5	20
S_4	5	7	7	8	6	10
Demand	30	30	15	20	5	

Tabel C.29 Instance n35 (3×4), $Z^* = 280$

	D_1	D_2	D_3	D_4	Supply
S_1	1	6	5	9	20
S_2	3	2	4	15	25
S_3	8	9	3	10	15
Demand	15	15	10	20	

Kasus sintetik.

Tabel C.30 Instance S1 (5×5), $Z^* = 18855$

	D_1	D_2	D_3	D_4	D_5	Supply
S_1	27	27	9	33	9	150
S_2	36	15	33	30	15	240
S_3	18	21	27	36	33	135
S_4	33	36	21	24	9	300
S_5	27	9	9	21	15	285
Demand	270	195	210	270	165	

Tabel C.31 Instance S2 (5×5), $Z^* = 153500$

	D_1	D_2	D_3	D_4	D_5	Supply
S_1	80	80	20	100	20	400
S_2	110	40	100	90	40	700
S_3	50	60	80	110	100	350
S_4	100	110	60	70	20	900
S_5	80	20	20	60	40	850
Demand	800	550	600	800	450	

Tabel C.32 Instance S3 (11×12), $Z^* = 1044180$

	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	Supply
S_1	3	156	159	162	165	168	171	174	177	180	183	36	150
S_2	189	42	195	198	201	204	207	210	213	216	219	222	180
S_3	225	228	81	234	237	240	243	246	249	252	255	258	210
S_4	261	264	267	120	273	276	279	282	285	288	291	294	240
S_5	297	300	303	306	159	312	315	318	321	324	327	330	270
S_6	333	336	339	342	345	198	351	354	357	360	363	366	300
S_7	369	372	375	378	381	384	237	390	393	396	399	402	330
S_8	405	408	411	414	417	420	423	276	429	432	435	438	360
S_9	441	444	447	450	453	456	459	462	315	468	471	474	390
S_{10}	477	480	483	486	489	492	495	498	501	354	507	510	420
S_{11}	513	516	519	522	525	528	531	534	537	540	393	546	930
Demand	150	180	210	240	270	300	330	360	390	420	450	480	

Tabel C.33 Instance S4 (11×12), $Z^* = 92212$

	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	Supply
S_1	13	35	30	36	35	30	30	33	31	36	15	39	200
S_2	39	30	25	32	28	13	11	32	34	27	14	25	106
S_3	12	39	15	11	13	30	30	31	12	25	32	13	139
S_4	33	10	14	38	38	12	12	15	13	15	37	14	137
S_5	37	11	25	10	14	35	29	14	33	14	37	37	249
S_6	79	88	70	70	83	71	92	93	77	95	82	77	171
S_7	80	98	98	84	73	91	88	86	81	78	73	73	172
S_8	83	94	77	94	76	91	81	70	85	91	99	84	113
S_9	78	75	78	81	79	81	81	98	83	95	88	77	230
S_{10}	99	81	79	87	89	81	76	74	76	78	89	94	241
S_{11}	91	83	80	92	97	97	80	89	92	71	83	99	199
Demand	315	309	41	43	110	162	274	184	70	201	125	123	

Tabel C.34 Instance S5 (11×12), $Z^* = 116020$

	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	Supply
S_1	1	52	53	54	55	56	57	58	59	60	61	12	50
S_2	63	14	65	66	67	68	69	70	71	72	73	74	60
S_3	75	76	27	78	79	80	81	82	83	84	85	86	70
S_4	87	88	89	40	91	92	93	94	95	96	97	98	80
S_5	99	100	101	102	53	104	105	106	107	108	109	110	90
S_6	111	112	113	114	115	66	117	118	119	120	121	122	100
S_7	123	124	125	126	127	128	79	130	131	132	133	134	110
S_8	135	136	137	138	139	140	141	92	143	144	145	146	120
S_9	147	148	149	150	151	152	153	154	105	156	157	158	130
S_{10}	159	160	161	162	163	164	165	166	167	118	169	170	140
S_{11}	171	172	173	174	175	176	177	178	179	180	131	182	310
Demand	50	60	70	80	90	100	110	120	130	140	150	160	

Kasus *real-world*. Kelima *real-world* (R1–R5) berukuran 2×94 , yaitu dua sumber dan 94 tujuan. Karena matriks asli terlalu lebar untuk ditampilkan sebagai tabel biasa, data ditranspos per kolom tujuan D_j pada Tabel C.35–C.39; nilai c_{1j} dan c_{2j} adalah biaya dari sumber S_1 dan S_2 ke tujuan D_j .

Tabel C.35 Instance R1 (2×94), $Z^* = 12928429$, Supply $S_1=308252$, $S_2=138801$; ditranspos per kolom tujuan

D_j	c_{1j}	c_{2j}	Demand
1	22	26	3852
2	38	41	4759

Tabel C.35 – *lanjutan*

D_j	c_{1j}	c_{2j}	Demand
3	32	36	1312
4	17	21	3451
5	31	35	7537
6	15	20	2827
7	23	27	3878
8	25	30	15224
9	25	28	2530
10	20	25	5690
11	33	38	2494
12	30	34	2383
13	18	23	32901
14	12	15	7314
15	23	20	2486
16	18	15	0
17	26	23	249
18	20	18	2735
19	15	14	498
20	18	17	1989
21	15	13	4475
22	25	23	6091
23	28	25	373
24	23	25	5492
25	16	17	17978
26	18	23	4305
27	21	23	23799
28	28	33	1597
29	20	18	2773
30	20	19	2556
31	25	28	48509
32	18	23	3279
33	18	20	4183
34	18	18	1303
35	25	25	795
36	27	31	4024
37	21	20	1381
38	25	25	2597
39	23	25	1666
40	26	23	4848
41	33	39	125
42	38	36	0
43	29	28	0
44	17	12	10192
45	38	33	0
46	17	14	4475
47	39	38	3356

Tabel C.35 – *lanjutan*

D_j	c_{1j}	c_{2j}	Demand
48	39	38	0
49	22	20	2611
50	22	28	2984
51	35	33	1368
52	25	28	0
53	28	26	1119
54	31	28	249
55	33	31	0
56	20	17	9447
57	36	38	2984
58	31	30	2362
59	23	30	125
60	27	25	2984
61	25	22	26599
62	35	33	746
63	30	26	249
64	31	29	0
65	28	31	14055
66	27	34	1437
67	28	28	2283
68	30	33	556
69	27	31	812
70	54	49	5786
71	43	38	2034
72	52	49	5911
73	56	52	5026
74	46	43	524
75	59	56	1133
76	53	49	464
77	40	36	1792
78	58	54	1330
79	44	40	1457
80	51	46	1668
81	40	36	3827
82	43	39	7947
83	45	41	297
84	49	46	759
85	48	44	9074
86	57	54	1436
87	46	41	864
88	46	43	2217
89	60	57	1160
90	61	57	519
91	66	62	4166
92	59	55	614

Tabel C.35 – *lanjutan*

D_j	c_{1j}	c_{2j}	Demand
93	55	52	19405
94	56	51	30392

Tabel C.36 Instance R2 (2×94), $Z^* = 13657053$, Supply $S_1=341015$, $S_2=138738$; ditranspos per kolom tujuan

D_j	c_{1j}	c_{2j}	Demand
1	22	26	3332
2	38	41	3976
3	32	36	1252
4	17	21	3494
5	31	35	9871
6	15	20	3027
7	23	27	5214
8	25	30	20029
9	25	28	2437
10	20	25	5584
11	33	38	2615
12	30	34	2184
13	18	23	39154
14	12	15	6641
15	23	20	2735
16	18	15	0
17	26	23	249
18	20	18	2984
19	15	14	746
20	18	17	2238
21	15	13	4848
22	25	23	6837
23	28	25	373
24	23	25	4677
25	16	17	17447
26	18	23	4487
27	21	23	24968
28	28	33	1804
29	20	18	3137
30	20	19	2538
31	25	28	50087
32	18	23	3445
33	18	20	4375
34	18	18	1444
35	25	25	919
36	27	31	4024
37	21	20	1389
38	25	25	2960

Tabel C.36 – *lanjutan*

D_j	c_{1j}	c_{2j}	Demand
39	23	25	1660
40	26	23	5221
41	33	39	125
42	38	36	0
43	29	28	0
44	17	12	11063
45	38	33	0
46	17	14	4972
47	39	38	3729
48	39	38	125
49	22	20	2859
50	22	28	5096
51	35	33	1492
52	25	28	0
53	28	26	871
54	31	28	249
55	33	31	125
56	20	17	10129
57	36	38	3232
58	31	30	2735
59	23	30	249
60	27	25	12554
61	25	22	28961
62	35	33	746
63	30	26	125
64	31	29	0
65	28	31	12664
66	27	34	1382
67	28	28	2405
68	30	33	479
69	27	31	724
70	54	49	6258
71	43	38	1924
72	52	49	5195
73	56	52	5217
74	46	43	536
75	59	56	924
76	53	49	385
77	40	36	2053
78	58	54	1111
79	44	40	1340
80	51	46	1712
81	40	36	3630
82	43	39	8300
83	45	41	321

Tabel C.36 – *lanjutan*

D_j	c_{1j}	c_{2j}	Demand
84	49	46	679
85	48	44	8954
86	57	54	1385
87	46	41	862
88	46	43	2372
89	60	57	1225
90	61	57	480
91	66	62	4369
92	59	55	556
93	55	52	20814
94	56	51	27658

Tabel C.37 Instance R3 (2×94), $Z^* = 12296587$, Supply $S_1=280110$, $S_2=138801$; ditranspos per kolom tujuan

D_j	c_{1j}	c_{2j}	Demand
1	22	26	3300
2	38	41	4628
3	32	36	1250
4	17	21	2852
5	31	35	8202
6	15	20	2916
7	23	27	3912
8	25	30	16157
9	25	28	2695
10	20	25	5526
11	33	38	2699
12	30	34	2271
13	18	23	41855
14	12	15	7074
15	23	20	1616
16	18	15	0
17	26	23	125
18	20	18	1741
19	15	14	373
20	18	17	1243
21	15	13	2859
22	25	23	3854
23	28	25	249
24	23	25	5151
25	16	17	17117
26	18	23	4368
27	21	23	24291
28	28	33	1560
29	20	18	2679

Tabel C.37 – *lanjutan*

D_j	c_{1j}	c_{2j}	Demand
30	20	19	2952
31	25	28	42223
32	18	23	3585
33	18	20	4182
34	18	18	1423
35	25	25	838
36	27	31	3282
37	21	20	1491
38	25	25	2324
39	23	25	1553
40	26	23	3108
41	33	39	125
42	38	36	0
43	29	28	125
44	17	12	6464
45	38	33	0
46	17	14	2859
47	39	38	2238
48	39	38	0
49	22	20	1616
50	22	28	2921
51	35	33	871
52	25	28	0
53	28	26	498
54	31	28	125
55	33	31	0
56	20	17	6091
57	36	38	1865
58	31	30	1616
59	23	30	125
60	27	25	2921
61	25	22	17029
62	35	33	498
63	30	26	249
64	31	29	0
65	28	31	13885
66	27	34	1516
67	28	28	2480
68	30	33	410
69	27	31	799
70	54	49	6616
71	43	38	1980
72	52	49	5900
73	56	52	4721
74	46	43	584

Tabel C.37 – *lanjutan*

D_j	c_{1j}	c_{2j}	Demand
75	59	56	1032
76	53	49	419
77	40	36	1702
78	58	54	1290
79	44	40	1536
80	51	46	1725
81	40	36	3455
82	43	39	7792
83	45	41	341
84	49	46	632
85	48	44	9100
86	57	54	1269
87	46	41	897
88	46	43	2333
89	60	57	1296
90	61	57	452
91	66	62	3957
92	59	55	487
93	55	52	20317
94	56	51	32228

Tabel C.38 Instance R4 (2×94), $Z^* = 14117586$, Supply $S_1=350910$, $S_2=142165$; ditranspos per kolom tujuan

D_j	c_{1j}	c_{2j}	Demand
1	22	26	3512
2	38	41	4300
3	32	36	1291
4	17	21	3477
5	31	35	9394
6	15	20	3031
7	23	27	4597
8	25	30	22500
9	25	28	1776
10	20	25	5384
11	33	38	2824
12	30	34	2530
13	18	23	42638
14	12	15	7341
15	23	20	2735
16	18	15	0
17	26	23	249
18	20	18	2984
19	15	14	746
20	18	17	2238

Tabel C.38 – *lanjutan*

D_j	c_{1j}	c_{2j}	Demand
21	15	13	4848
22	25	23	6837
23	28	25	3737
24	23	25	5858
25	16	17	16806
26	18	23	4210
27	21	23	24717
28	28	33	1715
29	20	18	2549
30	20	19	2520
31	25	28	50440
32	18	23	3185
33	18	20	4359
34	18	18	1437
35	25	25	914
36	27	31	3120
37	21	20	1551
38	25	25	2109
39	23	25	1578
40	26	23	5345
41	33	39	125
42	38	36	0
43	29	28	0
44	17	12	11187
45	38	33	0
46	17	14	4972
47	39	38	3729
48	39	38	125
49	22	20	2859
50	22	28	8452
51	35	33	1492
52	25	28	0
53	28	26	871
54	31	28	249
55	33	31	125
56	20	17	10317
57	36	38	3232
58	31	30	2735
59	23	30	249
60	27	25	8452
61	25	22	29085
62	35	33	746
63	30	26	373
64	31	29	0
65	28	31	13607

Tabel C.38 – *lanjutan*

D_j	c_{1j}	c_{2j}	Demand
66	27	34	1421
67	28	28	2068
68	30	33	554
69	27	31	877
70	54	49	6207
71	43	38	1844
72	52	49	6259
73	56	52	4824
74	46	43	610
75	59	56	1175
76	53	49	403
77	40	36	1840
78	58	54	1110
79	44	40	1389
80	51	46	1739
81	40	36	3546
82	43	39	8113
83	45	41	348
84	49	46	753
85	48	44	8366
86	57	54	1329
87	46	41	970
88	46	43	2121
89	60	57	1181
90	61	57	530
91	66	62	3778
92	59	55	497
93	55	52	19912
94	56	51	34947

Tabel C.39 Instance R5 (2×94), $Z^* = 14017419$, Supply $S_1=359746$, $S_2=138801$; ditranspos per kolom tujuan

D_j	c_{1j}	c_{2j}	Demand
1	22	26	3269
2	38	41	4456
3	32	36	1277
4	17	21	2783
5	31	35	8888
6	15	20	3679
7	23	27	4621
8	25	30	18426
9	25	28	2745
10	20	25	5940
11	33	38	2408

Tabel C.39 – *lanjutan*

D_j	c_{1j}	c_{2j}	Demand
12	30	34	2437
13	18	23	48268
14	12	15	5585
15	23	20	3108
16	18	15	0
17	26	23	249
18	20	18	3356
19	15	14	746
20	18	17	2486
21	15	13	5594
22	25	23	7707
23	28	25	373
24	23	25	4944
25	16	17	19086
26	18	23	3758
27	21	23	27005
28	28	33	1884
29	20	18	3139
30	20	19	2275
31	25	28	48013
32	18	23	3411
33	18	20	4479
34	18	18	1386
35	25	25	911
36	27	31	3405
37	21	20	1402
38	25	25	3198
39	23	25	1657
40	26	23	5967
41	33	39	249
42	38	36	0
43	29	28	0
44	17	12	12678
45	38	33	0
46	17	14	5594
47	39	38	4226
48	39	38	125
49	22	20	3232
50	22	28	7955
51	35	33	1616
52	25	28	0
53	28	26	995
54	31	28	373
55	33	31	125
56	20	17	11560

Tabel C.39 – *lanjutan*

D_j	c_{1j}	c_{2j}	Demand
57	36	38	3605
58	31	30	3108
59	23	30	249
60	27	25	7955
61	25	22	32938
62	35	33	871
63	30	26	249
64	31	29	0
65	28	31	13500
66	27	34	1377
67	28	28	2189
68	30	33	589
69	27	31	752
70	54	49	5974
71	43	38	1588
72	52	49	6085
73	56	52	4578
74	46	43	512
75	59	56	1056
76	53	49	452
77	40	36	2240
78	58	54	1138
79	44	40	1268
80	51	46	1818
81	40	36	3634
82	43	39	8274
83	45	41	300
84	49	46	703
85	48	44	8162
86	57	54	1308
87	46	41	901
88	46	43	2438
89	60	57	1088
90	61	57	454
91	66	62	4199
92	59	55	613
93	55	52	19521
94	56	51	29812

Lampiran D: Ringkasan Dataset Uji Tambahan (Dataset Uji 2 dan Dataset Uji 3)

Lampiran ini menyajikan ringkasan dataset uji tambahan Dataset Uji 2 (27 kasus) dan Dataset Uji 3 (640 kasus sintetis) yang dipakai pada Subbab 3.2.2 dan 4.3. Berbeda dari Lampiran C yang menampilkan matriks lengkap tiap kasus, lampiran ini menyajikan ringkasan statistik

karena volume data yang jauh lebih besar (667 *instance* tambahan): Tabel D.1 merangkum seluruh 27 kasus Dataset Uji 2 secara individual (ukuran, rentang biaya, total *supply/demand*, dan nilai optimal), sedangkan Tabel D.2 merangkum 640 kasus Dataset Uji 3 per 64 kelompok parameter (masing-masing 10 replikasi dengan rentang biaya R dan variasi *demand* V yang sama), disertai nilai optimal salah satu replikasi sebagai sampel representatif tiap kelompok.

Tabel D.1 Ringkasan 27 Kasus Uji Dataset Uji 2

Kasus	Ukuran	Cost Min	Cost Maks	Total Supply	Total Demand	Optimal
G1_P07	3×3	7	18	35	35	291
G1_P11	3×2	2	10	170	170	600
G1_P17	3×3	1	8	165	165	475
G1_P18	3×3	1	8	150	150	555
G1_P19	3×3	3	10	270	270	1390
G2_P12	3×4	0	5	100	100	180
G3_P02	3×4	1	9	1200	1200	2850
G3_P03	3×4	2	9	45	45	149
G3_P04	3×4	1	40	100	100	450
G3_P09	3×3	1	14	170	170	845
G3_P13	3×4	2	12	600	600	3210
G3_P20	3×4	30	100	74	74	3320
G3_S07	3×4	2	20	45	45	305
G4_P01	3×4	1	12	125	125	605
G6_P05	3×5	1	5	135	135	290
G6_P10	4×6	4	14	30	30	212
G6_S04	5×7	6	17	390	390	3125
G7_P06	4×4	0	24	21	21	49
G7_P08	4×6	1	33	1200	1200	10625
G7_P14	3×5	3	15	25	25	175
G7_P15	3×6	4	20	1330	1330	10830
G7_P16	4×3	2	16	80	80	325
G8_S01	10×10	5	40	615	615	5067
G8_S02	15×15	3	12	315	315	1080
G8_S03	13×13	2	17	267	267	885
G8_S05	22×21	2	15	640	640	1586
G8_S06	30×32	5	50	649	649	4702

Tabel D.2 Ringkasan 64 Kelompok Parameter Dataset Uji 3 (640 Kasus, 10 Replikasi per Kelompok)

Kel.	Ukuran	R	V	Rep.	Optimal Sampel
G10	10×20	20	0.08	10	1005192
G11	10×20	20	0.46	10	757197
G12	10×20	20	0.5	10	981970
G13	10×20	20	0.7	10	907844
G14	10×20	99	0.08	10	957636
G15	10×20	100	0.25	10	909258
G16	10×20	100	0.44	10	976974
G17	10×20	98	0.34	10	1217383
G18	10×20	488	0.1	10	700931

Tabel D.2 – lanjutan

Kel.	Ukuran	R	V	Rep.	Optimal Sampel
G19	10×20	499	0.25	10	557560
G20	10×20	488	0.04	10	323788
G21	10×20	496	0.75	10	550448
G22	10×20	997	0.11	10	91513
G23	10×20	997	0.5	10	292206
G24	10×20	984	0.46	10	98758
G25	10×20	999	0.75	10	88701
G26	10×40	20	0.11	10	1963291
G27	10×40	20	0.25	10	1964153
G28	10×40	20	0.5	10	1963236
G29	10×40	20	0.75	10	1964915
G30	10×40	100	0.11	10	1814498
G31	10×40	100	0.25	10	1815638
G32	10×40	100	0.5	10	1823454
G33	10×40	100	0.75	10	1820226
G34	10×40	500	0.11	10	1107919
G35	10×40	486	0.11	10	1480679
G36	10×40	499	0.5	10	1117676
G37	10×40	499	0.75	10	1077221
G38	10×40	998	0.03	10	487173
G39	10×40	999	0.25	10	223071
G40	10×40	998	0.5	10	187309
G41	10×40	997	0.75	10	195867
G42	10×60	20	0.08	10	3024986
G43	10×60	20	0.25	10	2945592
G44	10×60	20	0.5	10	2946964
G45	10×60	20	0.28	10	4039456
G46	10×60	100	0.11	10	2729707
G47	10×60	100	0.25	10	2727350
G48	10×60	100	0.5	10	2730048
G49	10×60	100	0.75	10	2728017
G50	10×60	500	0.11	10	1644885
G51	10×60	500	0.25	10	1650690
G52	10×60	500	0.5	10	1640996
G53	10×60	496	0.71	10	2075339
G54	10×60	997	0.11	10	319306
G55	10×60	998	0.05	10	678926
G56	10×60	999	0.5	10	362644
G57	10×60	999	0.75	10	268785
G58	10×100	20	0.11	10	4864345
G59	10×100	20	0.25	10	4910242
G60	10×100	20	1.17	10	3403520
G61	10×100	20	0.72	10	4998642
G62	10×100	100	0.09	10	4678315
G63	10×100	100	0.25	10	4548852
G64	10×100	100	0.5	10	4551322
G65	10×100	100	0.75	10	4551368
G66	10×100	500	0.07	10	3025796

Tabel D.2 – lanjutan

Kel.	Ukuran	R	V	Rep.	Optimal Sampel
G67	10×100	500	0.27	10	2955481
G68	10×100	500	0.5	10	2751887
G69	10×100	500	0.75	10	2747848
G70	10×100	999	0.14	10	1159466
G71	10×100	999	0.09	10	1087260
G72	10×100	998	0.48	10	1127013
G73	10×100	999	0.75	10	505596

Seluruh 667 berkas data mentah Dataset Uji 2 dan Dataset Uji 3 (format identik dengan Subbab 3.2.2: matriks biaya, *supply*, *demand*, dan nilai optimal pada baris terakhir tiap berkas) disertakan sebagai berkas pendamping pada folder RVM-benchmark/testcase/Data-tambahan/DU-2/ dan .../DU-3/, beserta skrip `run_benchmark_du23.py` dan `gen_lampiran_d.py` yang mereproduksi seluruh hasil dan tabel pada lampiran ini, agar data dan hasil pengujian dapat ditelusuri dan direproduksi penuh tanpa merujuk sumber eksternal.

BIODATA PENULIS



Penulis dilahirkan di Malang, 7 Juli 2004, merupakan anak pertama dari 2 bersaudara. Penulis telah menempuh pendidikan formal yaitu di TK Dharma Wanita Prambon, MIN Hadiluwih dan MI Nurul Ulum, MTsN 1 Mojokerto dan MAN Insan Cendekia Pasuruan. Setelah lulus dari MAN pada tahun 2022, Penulis mengikuti SNMPTN dan diterima di Departemen Teknik Informatika FTEIC - ITS pada tahun 2022 dan terdaftar dengan NRP 5025221047.

Halaman ini sengaja dikosongkan