

OPTIMALISASI PENEMPATAN KONTAINER PRIORITAS DI PELINDO REGIONAL 2 JAMBI MENGGUNAKAN MODEL KNAPSACK 0/1

Farhan Syifa Prayoga^{1,*}, Corry Sormin²

^{1,2}Program Studi Matematika, FST, Universitas Jambi

*Corresponding author: farhansyifa678@gmail.com

ABSTRACT

Pelindo Regional 2 Jambi, as a port operator, plays a crucial role in maintaining smooth logistic flows through container handling and yard storage activities. Increasing cargo volumes combined with limited yard capacity often create an imbalance between incoming containers and available storage space. This situation may lead to operational bottlenecks, longer waiting times, and reduced service efficiency. Although a Terminal Operating System (TOS) is utilized to support operational management, the system still faces limitations in handling emergency conditions such as overcapacity. To address this issue, an optimization approach using the 0/1 Knapsack method is applied to determine which containers should be allocated to the main yard block based on priority levels and capacity constraints. The model is solved using Dynamic Programming to obtain optimal solutions systematically and efficiently. The results indicate that this approach can generate more optimal container placement recommendations, reduce the risk of capacity overload, and enhance the effectiveness of container yard planning at Pelindo Regional 2 Jambi.

Keywords: Container Yard Management, Port, 0/1 Knapsack, Dynamic Programming, Optimization.

ABSTRAK

Pelindo Regional 2 Jambi sebagai operator pelabuhan memiliki peran penting dalam memastikan kelancaran arus logistik melalui aktivitas bongkar muat dan penumpukan kontainer. Peningkatan volume barang dan keterbatasan kapasitas blok penumpukan sering menimbulkan ketidakseimbangan antara jumlah kontainer yang masuk dan ruang yang tersedia. Kondisi ini dapat menyebabkan *bottleneck* operasional, bertambahnya waktu tunggu, serta penurunan efisiensi layanan. Meskipun *Terminal Operating System* (TOS) digunakan untuk mendukung manajemen operasional, sistem ini masih memiliki keterbatasan dalam menangani kondisi darurat seperti *overcapacity*. Untuk mengatasi hal tersebut, diterapkan pendekatan optimasi menggunakan metode Knapsack 0/1 guna menentukan kontainer yang sebaiknya ditempatkan pada blok utama berdasarkan prioritas dan kapasitas yang tersedia. Penyelesaian model dilakukan dengan *Dynamic Programming* untuk memperoleh solusi optimal secara sistematis dan efisien. Hasil penerapan model menunjukkan bahwa pendekatan ini mampu memberikan rekomendasi penataan kontainer yang lebih optimal, membantu mengurangi potensi kelebihan kapasitas, serta meningkatkan efektivitas perencanaan penumpukan kontainer di Pelindo Regional 2 Jambi.

Kata Kunci: Penumpukan Kontainer, Pelabuhan, Knapsack 0/1, *Dynamic Programming*, Optimasi.

ARTICLE INFO

Submission received: 19 November 2025

Accepted: 30 December 2025

Revised: 15 December 2025

Published: 31 December 2025

Available on: <https://doi.org/10.32493/sm.v7i3.54859>

StatMat: Jurnal Statistika dan Matematika is licenced under a Creative Commons Attribution-ShareAlike 4.0 International License.

1. PENDAHULUAN

Sebagai gerbang utama distribusi barang, pelabuhan memegang peran penting dalam menjaga kelancaran arus logistik antar wilayah maupun antar negara, sehingga aktivitas bongkar muat dan penumpukan kontainer harus dikelola dengan efisien untuk menghindari kemacetan dan gangguan operasional. Pengelolaan stok peti kemas yang kurang teratur dapat menimbulkan biaya tambahan, potensi kerusakan barang, serta menurunkan reputasi perusahaan, sementara berbagai kendala seperti keterlambatan truk, kekurangan kontainer, dan kerusakan peralatan bongkar muat dapat memperpanjang waktu tunggu dan menghambat distribusi. Tantangan tersebut semakin meningkat seiring bertambahnya volume barang yang masuk, terutama ketika jumlah barang tidak seimbang dengan kapasitas kontainer kapal yang digunakan untuk mengangkutnya. Ketidaksesuaian kapasitas ini dapat menyebabkan penumpukan kontainer, peningkatan biaya operasional, waktu tunggu yang lebih panjang, serta penurunan tingkat pelayanan terhadap pengguna jasa pelabuhan (Havidzi *et al.*, 2024).

Begitu pula di Pelindo Regional 2 Jambi, sebagai operator pelabuhan di wilayah Jambi, berkewajiban menerima semua kontainer yang masuk tanpa terkecuali. Namun demikian, kapasitas yard penumpukan memiliki batas tertentu yang harus dioptimalkan penggunaannya. Penempatan kontainer yang tidak terencana dapat menimbulkan *bottleneck* (keterbatasan) di jalur bongkar muat kapal ini mengakibatkan terlambatnya proses *handling*, dan memengaruhi kepuasan pelanggan (Battum *et al.*, 2022). Dalam satu minggu, rata-rata hanya terdapat satu hingga dua kapal yang bersandar di Pelabuhan Regional 2 Jambi, dengan jumlah muatan sekitar 130 kontainer atau lebih untuk setiap kapal. Namun, pada minggu ke-4 bulan Juli, tercatat sebuah kapal yang membawa 132 kontainer dalam satu kali kedatangan. Kondisi ini menjadi perhatian karena kapasitas blok utama hanya 178 TEUs, sementara blok buffer yang berfungsi sebagai area cadangan hanya mampu menampung 81 TEUs (Pelindo Jambi, 2025). Meskipun Pelindo Regional 2 Jambi telah menggunakan *Terminal Operating System* (TOS) untuk manajemen operasional terminal, sistem ini tetap memiliki keterbatasan. Selain sangat bergantung pada listrik dan jaringan, TOS membutuhkan pelatihan intensif karena kompleksitas operasionalnya, kurang fleksibel saat menghadapi situasi darurat seperti *overcapacity* atau kontainer yang harus diproses lebih cepat karena jadwal atau prioritas tertentu, dan memerlukan biaya implementasi serta pemeliharaan yang tinggi (*Business Research Insights*, 2023). Oleh karena itu, pendekatan optimasi berbasis perhitungan

matematika diperlukan sebagai alternatif untuk mendukung pengambilan keputusan manual atau semi-otomatis saat TOS tidak berfungsi optimal.

Sebagai salah satu solusi yang dapat diterapkan, digunakan pendekatan metode Knapsack 0/1. Menurut Taha (2017), metode Knapsack 0/1 merupakan model optimasi kombinatorial yang bertujuan memilih kombinasi objek dengan nilai keuntungan tertinggi tanpa melebihi kapasitas yang tersedia. Dalam konteks penempatan kontainer di Pelindo Regional 2 Jambi, metode ini dapat digunakan untuk menentukan kontainer mana yang ditempatkan di blok utama sesuai dengan kategori kontainernya, seperti kontainer domestik di blok domestik dan kontainer ekspor-impor di blok ekspor-impor. Apabila blok utama mengalami kelebihan kapasitas, maka kontainer dengan kategori prioritas lebih rendah dapat dialihkan ke blok buffer, sehingga seluruh kontainer tetap tertampung dengan efisien tanpa mengganggu kelancaran operasional pelabuhan. Menurut Devita dan Wibawa (2020), Untuk menyelesaikan permasalahan Knapsack 0/1 secara efisien, digunakan pendekatan *Dynamic Programming* karena mampu memecah masalah besar menjadi submasalah yang lebih kecil dengan solusi yang saling berkaitan. Metode ini menerapkan persamaan rekursif dalam bentuk tabel (*bottom-up*) untuk memperoleh solusi optimal secara sistematis.

Dengan adanya model optimasi seperti Knapsack 0/1, diharapkan pengelolaan penumpukan kontainer di Pelindo Regional 2 Jambi dapat berjalan lebih efisien, waktu bongkar muat dapat diminimalkan, serta pelayanan kepelabuhanan menjadi semakin optimal. Model ini dirancang untuk digunakan dalam perencanaan mingguan, yaitu dengan mempertimbangkan kapasitas blok penumpukan yang tersedia satu minggu sebelum kedatangan kapal. Dengan demikian, proses perencanaan penempatan kontainer dapat dilakukan secara terukur, sehingga potensi kelebihan kapasitas dapat diantisipasi lebih awal.

2. METODE PENELITIAN

2.1 Pengumpulan Data

Metode pengumpulan data dilakukan melalui observasi lapangan dan dokumentasi dari Terminal Peti Kemas Pelindo Regional 2 Jambi. Data yang diperoleh mencakup data kontainer kapal PT Pelayaran Sukses Sindo Damai serta data luas dan kapasitas blok penyimpanan kontainer. Data tersebut digunakan sebagai dasar penerapan metode Knapsack 0/1 dengan *Dynamic Programming* dalam proses optimasi penempatan kontainer. Adapun data yang digunakan pada penelitian ini didefinisikan pada tabel 1 dan tabel 2.

Tabel 1. Data Kontainer Kapal PT Pelayaran Sukses Sindo Damai

Shipper	Bobot Prioritas	Total Kontainer	w_k (Volume)	R_k (Bobot Prioritas * w_k)
CV. Sinar Buana	2	3	3 TEUs	6
PT. Indofood	2	2	2 TEUs	4
PT. Indofood	2	8	8 TEUs	16
PT. Bogasari	1	4	4 TEUs	4
CV Sinar Buana	3	2	2 TEUs	6
PT Sayap Mas	3	4	8 TEUs	24
Bintang Mas	3	1	1 TEUs	3
PT Sayap Mas Utama	3	4	8 TEUs	24
PT Sayap Mas Utama	3	3	6 TEUs	18
REPO MTY OOCL	0	4	8 TEUs	0

Pada tabel 1, prioritas kontainer ditetapkan berdasarkan *stuffing date* dan *delivery order*, yang mencerminkan urutan serta waktu pengiriman. Nilai prioritas kemudian dikonversi ke bentuk numerik dengan pengelompokan sebagai berikut: kontainer kosong (0), prioritas rendah (1), prioritas sedang (2), dan prioritas tinggi (3). Setiap nilai prioritas kemudian dikalikan dengan kapasitas kontainer dalam TEUs untuk menghasilkan nilai total prioritas (R_k), sesuai prinsip *weighted priority* yang umum digunakan dalam studi optimasi terminal.

Tabel 2. Data Blok Kontainer Lini 1 Pelindo Regional 2 Jambi

Daftar Blok	Kapasitas Blok (TEUs)	Kapasitas Tersedia (TEUs)
Blok 2B (Domestik)	506	178
Blok Buffer	81	81

Tabel tersebut menunjukkan kapasitas dan ketersediaan ruang pada dua blok penumpukan. Blok 2B memiliki kapasitas 506 TEUs dengan ketersediaan 178 TEUs, sedangkan Blok Buffer berkapasitas 81 TEUs dan seluruhnya masih tersedia.

2.2 Analisis Data

Untuk mendukung proses analisis dalam penelitian ini, pendekatan optimasi yang digunakan merujuk pada model 0/1 Knapsack. Model ini relevan karena struktur permasalahannya sesuai dengan kebutuhan penentuan pemilihan kontainer berdasarkan prioritas dan kapasitas penumpukan. Menurut Devita dan Wibawa (2025), Knapsack 0/1

termasuk *Integer Knapsack*, yaitu objek yang dimasukkan ke dalam media penyimpanan, dimana dimensinya harus dimasukkan semua atau tidak sama sekali. Jika diberikan satu set n barang dan satu knapsack dengan:

r_k : Keuntungan (profit) dari barang ke- k

w_k : Berat (weight) dari barang ke- k

c : Kapasitas Knapsack

Untuk mencari keuntungan maksimum dalam masalah Knapsack 0/1, fungsi objektif dan kendala dapat dirumuskan sebagai berikut:

Fungsi Objektif (*Maximization*):

$$\sum_{k=1}^n r_k x_k$$

Kendala Kapasitas:

$$\sum_{k=1}^n w_k x_k \leq c$$

Di mana nilai $x_k = 0$ atau $1; k = 1, 2, 3, \dots, n$. Jika x_k bernilai 1 maka barang ke- k terpilih. Sebaliknya, jika x_k bernilai 0 maka barang ke- k tidak terpilih. Dalam konteks penelitian ini, notasi pada model knapsack disesuaikan dengan kasus penempatan kontainer di lini 1, yaitu:

- r_k : Jumlah prioritas maksimal kontainer ke- k . Semakin tinggi nilainya, semakin besar peluang kontainer tersebut untuk ditempatkan di blok utama.
- w_k : Volume dalam TEUs kontainer ke- k , dimana kontainer 20 Feet memiliki volume 1 TEUs dan kontainer 40 Feet memiliki volume 2 TEUs.
- c : Kapasitas knapsack, yang dianalogikan sebagai kapasitas blok utama, sehingga hanya sejumlah kontainer tertentu yang dapat ditampung sesuai batas ini, jika blok utama melebihi kapasitas, akan ditempatkan di blok buffer.
- x_k : variabel keputusan untuk kontainer ke- k .

Dengan merujuk pada model 0/1 Knapsack, proses analisis selanjutnya memanfaatkan prinsip pemrograman dinamik untuk menentukan kombinasi kontainer dengan nilai prioritas maksimum pada kapasitas blok yang tersedia. Dalam konteks ini, fungsi prioritas blok didefinisikan sebagai $f_k(y) = \max\{f_{k-1}(y), r_k + f_{k-1}(y - w_k)\}$. Persamaan ini digunakan untuk menentukan nilai maksimum prioritas yang dapat dicapai pada kapasitas blok tertentu. Model tersebut sejalan dengan pendekatan optimasi berbasis *weighted priority* yang

dijelaskan oleh Weerasinghe *et al.*, (2024) serta konsep pemrograman dinamik untuk masalah kapasitas terbatas sebagaimana dijelaskan oleh Hillier dan Lieberman (2021).

Dalam penelitian ini, pendekatan pemrograman dinamik diterapkan pada permasalahan penempatan kontainer di lapangan penumpukan. Permasalahan tersebut dapat dimodelkan sebagai kasus *0-1 Knapsack*, di mana setiap kontainer dipandang sebagai sebuah item dengan bobot berupa kapasitas *Twenty-foot Equivalent Unit* (TEUs) dan nilai berupa tingkat prioritas (tinggi, sedang, atau rendah). Tujuan optimasi adalah memaksimalkan total nilai prioritas kontainer yang ditempatkan, dengan tetap memperhatikan keterbatasan kapasitas blok penumpukan. Melalui proses *forward dynamic programming* dilakukan pengisian tabel nilai maksimum berdasarkan kombinasi kontainer yang mungkin, sedangkan proses *backtracking* digunakan untuk menentukan kontainer mana saja yang dipilih secara optimal. Dengan pendekatan ini, penempatan kontainer diharapkan lebih efisien dalam penggunaan kapasitas serta sesuai dengan tingkat prioritas, sehingga mendukung kelancaran operasional logistik di terminal.

3. HASIL DAN PEMBAHASAN

3.1 Perhitungan *Forward*

Proses *forward* digunakan untuk menentukan nilai maksimum prioritas total yang dapat dicapai berdasarkan kapasitas blok penumpukan yang tersedia. Tahap ini berfungsi membangun tabel keputusan (*state table*) yang merekam nilai optimal untuk setiap kombinasi kapasitas dan jumlah kontainer.

Diketahui:

- Banyaknya jenis kontainer sejenis (k) = 10. Menunjukkan bahwa terdapat 10 jenis atau kelompok kontainer yang dianalisis.
- Kapasitas blok kontainer 0/ 1 (c) =
 - Blok Utama / Domestik (1) : 178 TEUs dari 506 TEUs
 - Blok Buffer (0) : 81 TEUs
- Volume Kontainer (w_k) = $\{w_1, w_2, w_3, \dots, w_{10}\} = \{3, 2, 8, 4, 2, 8, 1, 8, 6, 8 \text{ TEUs}\}$
- Nilai Prioritas Kontainer (R_k) = $\{R_1, R_2, R_3, \dots, R_{10}\} = \{6, 4, 16, 4, 6, 24, 3, 24, 18, 0\}$
- Prioritas awal blok saat tidak ada kontainer yang dimasukkan ($f_0(y)$) = 0

Dalam hal ini y menyatakan jumlah kontainer yang terdapat di dalam suatu blok penumpukan pada saat tertentu (*state variable*). Ketika belum ada kontainer dimasukkan

($y = 0$), maka nilai prioritas blok ($f_0(y)$) juga nol, karena blok belum memiliki beban atau tingkat kepentingan apa pun.

Tahap 1 :

k = Kontainer ke-1

$w_1 = 3$ TEUs

$R_1 = 6$

Fungsi awal: ($f_0(y)$) = 0

Model fungsi prioritas blok mengikuti bentuk rekursif:

$$f_k(y) = \max\{f_{k-1}(y), R_k + f_{k-1}(y - w_k)\}$$

Untuk kontainer pertama ($k = 1$):

$$f_1(y) = \max\{f_{1-1}(y), R_1 + f_{1-1}(y - w_1)\}$$

$$f_1(y) = \max\{f_0(y), R_1 + f_0(y - w_1)\}$$

$$f_1(y) = \max\{0, 6 + f_0(y - 3)\}$$

Tabel 3. Perhitungan *forward* tahap 1

y	$f_0(y)$	$6 + f_0(y - 3)$	$f_1(y)$	$x_1^*, x_2^*, x_3^*, x_4^*, \dots, x_{10}^*$	Keterangan
0	0	$6 + (-\infty) = -\infty$	0	(0,0,0,0,0,0,0,0,0,0)	Kontainer tidak dipilih
1	0	$6 + (-\infty) = -\infty$	0	(0,0,0,0,0,0,0,0,0,0)	Kontainer tidak dipilih
2	0	$6 + (-\infty) = -\infty$	0	(0,0,0,0,0,0,0,0,0,0)	Kontainer tidak dipilih
3	0	$6 + 0 = 6$	6	(1,0,0,0,0,0,0,0,0,0)	Diambil Kategori Kontainer 1
4	0	$6 + 0 = 6$	6	(1,0,0,0,0,0,0,0,0,0)	Diambil Kategori Kontainer 1
5	0	$6 + 0 = 6$	6	(1,0,0,0,0,0,0,0,0,0)	Diambil Kategori Kontainer 1
6	0	$6 + 0 = 6$	6	(1,0,0,0,0,0,0,0,0,0)	Diambil Kategori Kontainer 1
7	0	$6 + 0 = 6$	6	(1,0,0,0,0,0,0,0,0,0)	Diambil Kategori Kontainer 1
8	0	$6 + 0 = 6$	6	(1,0,0,0,0,0,0,0,0,0)	Diambil Kategori Kontainer 1
9	0	$6 + 0 = 6$	6	(1,0,0,0,0,0,0,0,0,0)	Diambil Kategori Kontainer 1
10	0	$6 + 0 = 6$	6	(1,0,0,0,0,0,0,0,0,0)	Diambil Kategori Kontainer 1

Tahap 2:

k = Kontainer ke-2

$w_2 = 2$ TEUs

$R_2 = 4$

Model fungsi prioritas blok mengikuti bentuk rekursif:

$$f_k(y) = \max\{f_{k-1}(y), R_k + f_{k-1}(y - w_k)\}$$

Untuk kontainer kedua ($k = 2$):

$$f_2(y) = \max\{f_{2-1}(y), R_2 + f_{2-1}(y - w_2)\}$$

$$f_2(y) = \max\{f_1(y), R_2 + f_1(y - w_2)\}$$

$$f_2(y) = \max\{f_1(y), 4 + f_1(y - 2)\}$$

Tabel 4. Perhitungan *forward* tahap 2

y	$f_1(y)$	$4 + f_1(y - 2)$	$f_2(y)$	$x_1^*, x_2^*, x_3^*, x_4^*, \dots, x_{10}^*$	Keterangan
0	0	$4 + (-\infty) = -\infty$	0	(0,0,0,0,0,0,0,0,0,0)	Kontainer tidak dipilih
1	0	$4 + (-\infty) = -\infty$	0	(0,0,0,0,0,0,0,0,0,0)	Kontainer tidak dipilih
2	0	$4 + 0 = 4$	4	(0,1,0,0,0,0,0,0,0,0)	Diambil Kategori Kontainer 2
3	6	$4 + 6 = 10$	10	(1,1,0,0,0,0,0,0,0,0)	Diambil Kombinasi Kontainer 1 dan 2
4	6	$4 + 6 = 10$	10	(1,1,0,0,0,0,0,0,0,0)	Diambil Kombinasi Kontainer 1 dan 2
5	6	$4 + 6 = 10$	10	(1,1,0,0,0,0,0,0,0,0)	Diambil Kombinasi Kontainer 1 dan 2
6	6	$4 + 6 = 10$	10	(1,1,0,0,0,0,0,0,0,0)	Diambil Kombinasi Kontainer 1 dan 2
7	6	$4 + 6 = 10$	10	(1,1,0,0,0,0,0,0,0,0)	Diambil Kombinasi Kontainer 1 dan 2
8	6	$4 + 6 = 10$	10	(1,1,0,0,0,0,0,0,0,0)	Diambil Kombinasi Kontainer 1 dan 2
9	6	$4 + 6 = 10$	10	(1,1,0,0,0,0,0,0,0,0)	Diambil Kombinasi Kontainer 1 dan 2
10	6	$4 + 6 = 10$	10	(1,1,0,0,0,0,0,0,0,0)	Diambil Kombinasi Kontainer 1 dan 2

Tahap 3:

$k =$ Kontainer ke-3

$w_3 = 8$ TEUs

$R_3 = 16$

Model fungsi prioritas blok mengikuti bentuk rekursif:

$$f_k(y) = \max\{f_{k-1}(y), R_k + f_{k-1}(y - w_k)\}$$

Untuk kontainer ketiga ($k = 3$):

$$f_3(y) = \max\{f_{3-1}(y), R_3 + f_{3-1}(y - w_3)\}$$

$$f_3(y) = \max\{f_2(y), R_3 + f_2(y - w_3)\}$$

$$f_3(y) = \max\{f_1(y), 16 + f_2(y - 8)\}$$

Tabel 5. Perhitungan *forward* tahap 3

y	$f_2(y)$	$16 + f_2(y - 8)$	$f_3(y)$	$x_1^*, x_2^*, x_3^*, x_4^*, \dots, x_{10}^*$	Keterangan
0	0	$16 + (-\infty) = -\infty$	0	(0,0,0,0,0,0,0,0,0,0)	Kontainer tidak dipilih
1	0	$16 + (-\infty) = -\infty$	0	(0,0,0,0,0,0,0,0,0,0)	Kontainer tidak dipilih
2	4	$16 + (-\infty) = -\infty$	4	(0,1,0,0,0,0,0,0,0,0)	Dipilih kategori kontainer 2
3	10	$16 + (-\infty) = -\infty$	10	(1,1,0,0,0,0,0,0,0,0)	Dipilih kombinasi kontainer 1 dan 2
4	10	$16 + (-\infty) = -\infty$	10	(1,1,0,0,0,0,0,0,0,0)	Dipilih kombinasi kontainer 1 dan 2
5	10	$16 + (-\infty) = -\infty$	10	(1,1,0,0,0,0,0,0,0,0)	Dipilih kombinasi kontainer 1 dan 2
6	10	$16 + (-\infty) = -\infty$	10	(1,1,0,0,0,0,0,0,0,0)	Dipilih kombinasi kontainer 1 dan 2
7	10	$16 + (-\infty) = -\infty$	10	(1,1,0,0,0,0,0,0,0,0)	Dipilih kombinasi kontainer 1 dan 2
8	10	$16 + 10 = 26$	26	(1,1,1,0,0,0,0,0,0,0)	Dipilih kombinasi kontainer 1, 2, dan 3
9	10	$16 + 10 = 26$	26	(1,1,1,0,0,0,0,0,0,0)	Dipilih kombinasi kontainer 1, 2, dan 3
10	10	$16 + 10 = 26$	26	(1,1,1,0,0,0,0,0,0,0)	Dipilih kombinasi kontainer 1, 2, dan 3

Tahap 4:

k = Kontainer ke-4

$w_4 = 4$ TEUs

$R_4 = 4$

Model fungsi prioritas blok mengikuti bentuk rekursif:

$$f_k(y) = \max\{f_{k-1}(y), R_k + f_{k-1}(y - w_k)\}$$

Untuk kontainer keempat ($k = 4$):

$$f_4(y) = \max\{f_{4-1}(y), R_4 + f_{4-1}(y - w_4)\}$$

$$f_4(y) = \max\{f_3(y), R_4 + f_3(y - w_4)\}$$

$$f_4(y) = \max\{f_3(y), 4 + f_3(y - 4)\}$$

Tabel 6. Perhitungan *forward* tahap 4

y	$f_3(y)$	$4 + f_3(y - 4)$	$f_4(y)$	$x_1^*, x_2^*, x_3^*, x_4^*, \dots, x_{10}^*$	Keterangan
0	0	$4 + (-\infty)$ $= -\infty$	0	(0,0,0,0,0,0,0,0,0,0)	Kontainer tidak dipilih
1	0	$4 + (-\infty)$ $= -\infty$	0	(0,0,0,0,0,0,0,0,0,0)	Kontainer tidak dipilih
2	4	$4 + (-\infty)$ $= -\infty$	4	(0,1,0,0,0,0,0,0,0,0)	Dipilih kategori kontainer 2
3	10	$4 + (-\infty)$ $= -\infty$	10	(1,1,0,0,0,0,0,0,0,0)	Dipilih kombinasi kontainer 1 dan 2
4	10	$4 + 10 = 14$	14	(1,1,0,1,0,0,0,0,0,0)	Dipilih kombinasi kontainer 1, 2 dan 4
5	10	$4 + 10 = 14$	14	(1,1,0,1,0,0,0,0,0,0)	Dipilih kombinasi kontainer 1, 2 dan 4
6	10	$4 + 10 = 14$	14	(1,1,0,1,0,0,0,0,0,0)	Dipilih kombinasi kontainer 1, 2 dan 4
7	10	$4 + 10 = 14$	14	(1,1,0,1,0,0,0,0,0,0)	Dipilih kombinasi kontainer 1, 2 dan 4
8	26	$4 + 26 = 30$	30	(1,1,1,1,0,0,0,0,0,0)	Dipilih kombinasi kontainer 1, 2, 3 dan 4
9	26	$4 + 26 = 30$	30	(1,1,1,1,0,0,0,0,0,0)	Dipilih kombinasi kontainer 1, 2, 3 dan 4
10	26	$4 + 26 = 30$	30	(1,1,1,1,0,0,0,0,0,0)	Dipilih kombinasi kontainer 1, 2, 3 dan 4

Tahap 5:

$k =$ Kontainer ke-5

$w_5 = 2$ TEUs

$R_5 = 6$

Model fungsi prioritas blok mengikuti bentuk rekursif:

$$f_k(y) = \max\{f_{k-1}(y), R_k + f_{k-1}(y - w_k)\}$$

Untuk kontainer kelima ($k = 5$):

$$f_5(y) = \max\{f_{5-1}(y), R_5 + f_{5-1}(y - w_5)\}$$

$$f_5(y) = \max\{f_4(y), R_5 + f_4(y - w_5)\}$$

$$f_5(y) = \max\{f_4(y), 6 + f_4(y - 2)\}$$

Tabel 7. Perhitungan *forward* tahap 5

y	$f_4(y)$	$6 + f_4(y - 2)$	$f_5(y)$	$x_1^*, x_2^*, x_3^*, x_4^*, \dots, x_{10}^*$	Keterangan
0	0	$6 + (-\infty)$ $= -\infty$	0	(0,0,0,0,0,0,0,0,0,0)	Kontainer tidak dipilih
1	0	$6 + (-\infty)$ $= -\infty$	0	(0,0,0,0,0,0,0,0,0,0)	Kontainer tidak dipilih
2	4	$6 + 4 = 10$	10	(0,1,0,0,1,0,0,0,0,0)	Dipilih kategori kontainer 2 dan 5
3	10	$6 + 10 = 16$	16	(1,1,0,0,1,0,0,0,0,0)	Dipilih kombinasi kontainer 1, 2, dan 5
4	14	$6 + 14 = 20$	20	(1,1,0,1,1,0,0,0,0,0)	Dipilih kombinasi kontainer 1, 2, 4, dan 5
5	14	$6 + 14 = 20$	20	(1,1,0,1,1,0,0,0,0,0)	Dipilih kombinasi kontainer 1, 2, 4, dan 5
6	14	$6 + 14 = 20$	20	(1,1,0,1,1,0,0,0,0,0)	Dipilih kombinasi kontainer 1, 2, 4, dan 5
7	14	$6 + 14 = 20$	20	(1,1,0,1,1,0,0,0,0,0)	Dipilih kombinasi kontainer 1, 2, 4, dan 5
8	30	$6 + 30 = 36$	36	(1,1,1,1,1,0,0,0,0,0)	Dipilih kombinasi kontainer 1, 2, 3, 4, dan 5
9	30	$6 + 30 = 36$	36	(1,1,1,1,1,0,0,0,0,0)	Dipilih kombinasi kontainer 1, 2, 3, 4, dan 5
10	30	$6 + 30 = 36$	36	(1,1,1,1,1,0,0,0,0,0)	Dipilih kombinasi kontainer 1, 2, 3, 4, dan 5

Tahap 6:

k = Kontainer ke-6

$w_6 = 8$ TEUs

$R_6 = 24$

Model fungsi prioritas blok mengikuti bentuk rekursif:

$$f_k(y) = \max\{f_{k-1}(y), R_k + f_{k-1}(y - w_k)\}$$

Untuk kontainer keenam ($k = 6$):

$$f_6(y) = \max\{f_{6-1}(y), R_6 + f_{6-1}(y - w_6)\}$$

$$f_6(y) = \max\{f_5(y), R_6 + f_5(y - w_6)\}$$

$$f_6(y) = \max\{f_5(y), 24 + f_5(y - 8)\}$$

Tabel 8. Perhitungan *forward* tahap 6

y	$f_5(y)$	$24 + f_5(y - 8)$	$f_6(y)$	$x_1^*, x_2^*, x_3^*, x_4^*, \dots, x_{10}^*$	Keterangan
0	0	$24 + (-\infty)$ $= -\infty$	0	(0,0,0,0,0,0,0,0,0,0)	Kontainer tidak dipilih
1	0	$24 + (-\infty)$ $= -\infty$	0	(0,0,0,0,0,0,0,0,0,0)	Kontainer tidak dipilih
2	10	$24 + (-\infty)$ $= -\infty$	10	(0,1,0,0,1,0,0,0,0,0)	Dipilih kategori kontainer 2 dan 5
3	16	$24 + (-\infty)$ $= -\infty$	16	(1,1,0,0,1,0,0,0,0,0)	Dipilih kombinasi kontainer 1, 2, dan 5
4	20	$24 + (-\infty)$ $= -\infty$	20	(1,1,0,1,1,0,0,0,0,0)	Dipilih kombinasi kontainer 1, 2, 4, dan 5
5	20	$24 + (-\infty)$ $= -\infty$	20	(1,1,0,1,1,0,0,0,0,0)	Dipilih kombinasi kontainer 1, 2, 4, dan 5
6	20	$24 + (-\infty)$ $= -\infty$	20	(1,1,0,1,1,0,0,0,0,0)	Dipilih kombinasi kontainer 1, 2, 4, dan 5
7	20	$24 + (-\infty)$ $= -\infty$	20	(1,1,0,1,1,0,0,0,0,0)	Dipilih kombinasi kontainer 1, 2, 4, dan 5
8	36	$24 + 36 = 60$	60	(1,1,1,1,1,1,0,0,0,0)	Dipilih kombinasi kontainer 1, 2, 3, 4, 5, dan 6
9	36	$24 + 36 = 60$	60	(1,1,1,1,1,1,0,0,0,0)	Dipilih kombinasi kontainer 1, 2, 3, 4, 5, dan 6
10	36	$24 + 36 = 60$	60	(1,1,1,1,1,1,0,0,0,0)	Dipilih kombinasi kontainer 1, 2, 3, 4, 5, dan 6

Tahap 7:

k = Kontainer ke-7

$w_7 = 1$ TEUs

$R_7 = 3$

Model fungsi prioritas blok mengikuti bentuk rekursif:

$$f_k(y) = \max\{f_{k-1}(y), R_k + f_{k-1}(y - w_k)\}$$

Untuk kontainer ketujuh ($k = 7$):



$$f_7(y) = \max\{f_{7-1}(y), R_7 + f_{7-1}(y - w_7)\}$$

$$f_7(y) = \max\{f_6(y), R_7 + f_6(y - w_7)\}$$

$$f_7(y) = \max\{f_6(y), 3 + f_6(y - 1)\}$$

Tabel 9. Perhitungan *forward* tahap 7

y	$f_6(y)$	$3 + f_6(y - 1)$	$f_7(y)$	$x_1^*, x_2^*, x_3^*, x_4^*, \dots, x_{10}^*$	Keterangan
0	0	$3 + (-\infty) = -\infty$	0	(0,0,0,0,0,0,0,0,0,0)	Kontainer tidak dipilih
1	0	$3 + 0 = 3$	3	(0,0,0,0,0,0,1,0,0,0)	Dipilih kategori kontainer 7
2	10	$3 + 10 = 13$	13	(0,1,0,0,1,0,1,0,0,0)	Dipilih kategori kontainer 2, 5, dan 7
3	16	$3 + 16 = 19$	19	(1,1,0,0,1,0,1,0,0,0)	Dipilih kombinasi kontainer 1, 2, 5, dan 7
4	20	$3 + 20 = 23$	23	(1,1,0,1,1,0,1,0,0,0)	Dipilih kombinasi kontainer 1, 2, 4, 5, dan 7
5	20	$3 + 20 = 23$	23	(1,1,0,1,1,0,1,0,0,0)	Dipilih kombinasi kontainer 1, 2, 4, 5, dan 7
6	20	$3 + 20 = 23$	23	(1,1,0,1,1,0,1,0,0,0)	Dipilih kombinasi kontainer 1, 2, 4, 5, dan 7
7	20	$3 + 20 = 23$	23	(1,1,0,1,1,0,1,0,0,0)	Dipilih kombinasi kontainer 1, 2, 4, 5, dan 7
8	60	$3 + 60 = 63$	63	(1,1,1,1,1,1,1,0,0,0)	Dipilih kombinasi kontainer 1, 2, 3, 4, 5, 6, dan 7
9	60	$3 + 60 = 63$	63	(1,1,1,1,1,1,1,0,0,0)	Dipilih kombinasi kontainer 1, 2, 3, 4, 5, 6, dan 7
10	60	$3 + 60 = 63$	63	(1,1,1,1,1,1,1,0,0,0)	Dipilih kombinasi kontainer 1, 2, 3, 4, 5, 6, dan 7

Tahap 8:

k = Kontainer ke-8

$w_8 = 8$ TEUs

$R_8 = 24$

Model fungsi prioritas blok mengikuti bentuk rekursif:

$$f_k(y) = \max\{f_{k-1}(y), R_k + f_{k-1}(y - w_k)\}$$

Untuk kontainer kedelapan ($k = 8$):

$$f_8(y) = \max\{f_{8-1}(y), R_8 + f_{8-1}(y - w_8)\}$$

$$f_8(y) = \max\{f_7(y), R_8 + f_7(y - w_8)\}$$

$$f_8(y) = \max\{f_7(y), 24 + f_7(y - 8)\}$$

Tabel 10. Perhitungan *forward* tahap 8

y	$f_7(y)$	$24 + f_7(y - 8)$	$f_8(y)$	$x_1^*, x_2^*, x_3^*, x_4^*, \dots, x_{10}^*$	Keterangan
0	0	$24 + (-\infty)$ $= -\infty$	0	(0,0,0,0,0,0,0,0,0,0)	Kontainer tidak dipilih
1	3	$24 + (-\infty)$ $= -\infty$	3	(0,0,0,0,0,0,1,0,0,0)	Dipilih kategori kontainer 7
2	13	$24 + (-\infty)$ $= -\infty$	13	(0,1,0,0,1,0,1,0,0,0)	Dipilih kategori kontainer 2, 5, dan 7
3	19	$24 + (-\infty)$ $= -\infty$	19	(1,1,0,0,1,0,1,0,0,0)	Dipilih kombinasi kontainer 1, 2, 5, dan 7
4	23	$24 + (-\infty)$ $= -\infty$	23	(1,1,0,1,1,0,1,0,0,0)	Dipilih kombinasi kontainer 1, 2, 4, 5, dan 7
5	23	$24 + (-\infty)$ $= -\infty$	23	(1,1,0,1,1,0,1,0,0,0)	Dipilih kombinasi kontainer 1, 2, 4, 5, dan 7
6	23	$24 + (-\infty)$ $= -\infty$	23	(1,1,0,1,1,0,1,0,0,0)	Dipilih kombinasi kontainer 1, 2, 4, 5, dan 7
7	23	$24 + (-\infty)$ $= -\infty$	23	(1,1,0,1,1,0,1,0,0,0)	Dipilih kombinasi kontainer 1, 2, 4, 5, dan 7
8	63	$24 + 63 = 87$	87	(1,1,1,1,1,1,1,0,0,0)	Dipilih kombinasi kontainer 1, 2, 3, 4, 5, 6, 7, dan 8
9	63	$24 + 63 = 87$	87	(1,1,1,1,1,1,1,0,0,0)	Dipilih kombinasi kontainer 1, 2, 3, 4, 5, 6, 7, dan 8
10	63	$24 + 63 = 87$	87	(1,1,1,1,1,1,1,0,0,0)	Dipilih kombinasi kontainer 1, 2, 3, 4, 5, 6, 7, dan 8

Tahap 9:

k = Kontainer ke-9

$w_9 = 6$ TEUs

$R_9 = 18$

Model fungsi prioritas blok mengikuti bentuk rekursif:

$$f_k(y) = \max\{f_{k-1}(y), R_k + f_{k-1}(y - w_k)\}$$

Untuk kontainer kesembilan ($k = 9$):

$$f_9(y) = \max\{f_{9-1}(y), R_9 + f_{9-1}(y - w_9)\}$$

$$f_9(y) = \max\{f_8(y), R_9 + f_8(y - w_9)\}$$

$$f_9(y) = \max\{f_8(y), 18 + f_8(y - 6)\}$$

Tabel 11. Perhitungan *forward* tahap 9

y	$f_8(y)$	$18 + f_8(y - 6)$	$f_9(y)$	$x_1^*, x_2^*, x_3^*, x_4^*, \dots, x_{10}^*$	Keterangan
0	0	$18 + (-\infty)$ $= -\infty$	0	(0,0,0,0,0,0,0,0,0,0)	Kontainer tidak dipilih
1	3	$18 + (-\infty)$ $= -\infty$	3	(0,0,0,0,0,0,1,0,0,0)	Dipilih kategori kontainer 7
2	13	$18 + (-\infty)$ $= -\infty$	13	(0,1,0,0,1,0,1,0,0,0)	Dipilih kategori kontainer 2, 5, dan 7
3	19	$18 + (-\infty)$ $= -\infty$	19	(1,1,0,0,1,0,1,0,0,0)	Dipilih kombinasi kontainer 1, 2, 5, dan 7
4	23	$18 + (-\infty)$ $= -\infty$	23	(1,1,0,1,1,0,1,0,0,0)	Dipilih kombinasi kontainer 1, 2, 4, 5, dan 7
5	23	$18 + (-\infty)$ $= -\infty$	23	(1,1,0,1,1,0,1,0,0,0)	Dipilih kombinasi kontainer 1, 2, 4, 5, dan 7
6	23	$18 + 23 = 41$	41	(1,1,0,1,1,0,1,0,1,0)	Dipilih kombinasi kontainer 1, 2, 4, 5, 7, dan 9
7	23	$18 + 23 = 41$	41	(1,1,0,1,1,0,1,0,1,0)	Dipilih kombinasi kontainer 1, 2, 4, 5, 7, dan 9
8	87	$18 + 87 = 105$	105	(1,1,1,1,1,1,1,1,1,0)	Dipilih kombinasi kontainer 1, 2, 3, 4, 5, 6, 7, 8, dan 9
9	87	$18 + 87 = 105$	105	(1,1,1,1,1,1,1,1,1,0)	Dipilih kombinasi kontainer 1, 2, 3, 4, 5, 6, 7, 8, dan 9
10	87	$18 + 87 = 105$	105	(1,1,1,1,1,1,1,1,1,0)	Dipilih kombinasi kontainer 1, 2, 3, 4, 5, 6, 7, 8, dan 9

Tahap 10:

$k =$ Kontainer ke-10

$$w_{10} = 8 \text{ TEUs}$$

$$R_{10} = 0$$

Model fungsi prioritas blok mengikuti bentuk rekursif:

$$f_k(y) = \max\{f_{k-1}(y), R_k + f_{k-1}(y - w_k)\}$$

Untuk kontainer kesembilan ($k = 10$):

$$f_{10}(y) = \max\{f_{10-1}(y), R_{10} + f_{10-1}(y - w_{10})\}$$

$$f_{10}(y) = \max\{f_9(y), R_{10} + f_9(y - w_{10})\}$$

$$f_{10}(y) = \max\{f_9(y), 0 + f_9(y - 8)\}$$

Tabel 12. Perhitungan *forward* tahap 10

y	$f_9(y)$	$0 + f_9(y - 8)$	$f_{10}(y)$	$x_1^*, x_2^*, x_3^*, x_4^*, \dots, x_{10}^*$	Keterangan
0	0	$0 + (-\infty) = -\infty$	0	(0,0,0,0,0,0,0,0,0,0)	Kontainer tidak dipilih
1	3	$0 + (-\infty) = -\infty$	3	(0,0,0,0,0,0,1,0,0,0)	Dipilih kategori kontainer 7
2	13	$0 + (-\infty) = -\infty$	13	(0,1,0,0,1,0,1,0,0,0)	Dipilih kategori kontainer 2, 5, dan 7
3	19	$0 + (-\infty) = -\infty$	19	(1,1,0,0,1,0,1,0,0,0)	Dipilih kombinasi kontainer 1, 2, 5, dan 7
4	23	$0 + (-\infty) = -\infty$	23	(1,1,0,1,1,0,1,0,0,0)	Dipilih kombinasi kontainer 1,2,4,5, dan 7
5	23	$0 + (-\infty) = -\infty$	23	(1,1,0,1,1,0,1,0,0,0)	Dipilih kombinasi kontainer 1,2, 4, 5, dan 7
6	41	$0 + (-\infty) = -\infty$	41	(1,1,0,1,1,0,1,0,1,0)	Dipilih kombinasi kontainer 1, 2, 4, 5, 7, dan 9
7	41	$0 + (-\infty) = -\infty$	41	(1,1,0,1,1,0,1,0,1,0)	Dipilih kombinasi kontainer 1, 2, 4, 5, 7, dan 9
8	105	$0 + 105 = 105$	105	(1,1,1,1,1,1,1,1,1,0)	Dipilih kombinasi kontainer 1, 2, 3, 4, 5, 6, 7, 8, dan 9
9	105	$0 + 105 = 105$	105	(1,1,1,1,1,1,1,1,1,0)	Dipilih kombinasi kontainer 1, 2, 3, 4, 5, 6, 7, 8, dan 9
10	105	$0 + 105 = 105$	105	(1,1,1,1,1,1,1,1,1,0)	Dipilih kombinasi kontainer 1, 2, 3, 4, 5, 6, 7, 8, dan 9

Pada proses *forward dynamic programming* tahap akhir, nilai rekursif prioritas maksimum yang diperoleh adalah sebesar 105. Hasil ini menunjukkan bahwa konfigurasi optimum pada tahap tersebut dicapai dengan memilih kategori kontainer ke-1, 2, 3, 4, 5, 6,

7, 8, dan 9, sedangkan kontainer ke-10 tidak termasuk dalam kombinasi optimum. Nilai ini menunjukkan hasil akumulasi optimal dari proses penelusuran ke depan yang mempertimbangkan setiap kemungkinan penempatan kontainer berdasarkan kapasitas dan kategori blok yang tersedia. Hasil dari proses *forward* ini kemudian digunakan sebagai acuan dalam tahap *backward*, di mana dilakukan pelacakan kembali untuk menentukan kombinasi kontainer yang menghasilkan nilai prioritas maksimum tersebut, sehingga susunan penempatan yang dihasilkan bersifat optimal dengan mempertimbangkan kapasitas blok yang tersedia.

3.2 Perhitungan *Backtracking* / *Backward*

Menurut Hillier dan Lieberman (2021), proses *backward* dalam pemrograman dinamis (*Dynamic Programming*) merupakan tahap perhitungan yang dilakukan secara mundur dari kondisi akhir menuju kondisi awal untuk menentukan keputusan optimal pada setiap tahap. Dalam konteks permasalahan *knapsack 0-1*, pendekatan ini digunakan untuk mencari kombinasi barang atau kontainer yang memberikan nilai maksimum tanpa melebihi kapasitas yang tersedia dengan rincian sebagai berikut:

- a. Banyaknya Kontainer yang sudah diklasifikasi (n) = 10, $k = 10, 9, 8, \dots, 1$.
- b. y : Kapasitas *sis*a (atau state) yang dibawa dari tahap sebelumnya.
- c. x_k : Keputusan untuk barang ke- k :
 - $x_k = 1 \rightarrow$ barang ke- k dipilih.
 - $x_k = 0 \rightarrow$ barang ke- k tidak dipilih.
- d. w_k : Bobot barang ke- k (TEUs).
- e. R_k : Nilai maksimum (total prioritas).
- f. q_k : Perhitungan nilai baru jika barang ke- k dipilih.
- g. Kapasitas blok domestik *knapsack 0 – 1* (c) = 178 TEUs

Tabel 13. Perhitungan *backward*

k	y	x_k	q_k	R_k	w_k	c
10	178	0	105	$q_{10} - R_{10}$	$y - w_{10}$	178
	TEUs			$105 - 0 = 105$	178 TEUs (Karena kontainer tidak dipilih)	TEUs
9	178	1	105	$q_9 - R_9$	$y - w_9$	172
	TEUs			$105 - 18 = 87$	$178 - 6 = 172$ TEUs	TEUs
8	172	1	87	$q_8 - R_8$	$y - w_8$	164
	TEUs			$87 - 24 = 63$	$172 - 8 = 164$ TEUs	TEUs
7	164	1	63	$q_7 - R_7$	$y - w_7$	163
	TEUs			$63 - 3 = 60$	$164 - 1 = 163$ TEUs	TEUs
6	163	1	60	$q_6 - R_6$	$y - w_6$	155
	TEUs			$60 - 24 = 36$	$163 - 8 = 155$ TEUs	TEUs
5	155	1	36	$q_5 - R_5$	$y - w_5$	153
	TEUs			$36 - 6 = 30$	$155 - 2 = 153$ TEUs	TEUs
4	153	1	30	$q_4 - R_4$	$y - w_4$	149
	TEUs			$30 - 4 = 26$	$153 - 4 = 149$ TEUs	TEUs
3	149	1	26	$q_3 - R_3$	$y - w_3$	141
	TEUs			$26 - 16 = 10$	$149 - 8 = 141$ TEUs	TEUs
2	141	1	10	$q_2 - R_2$	$y - w_2$	139
	TEUs			$10 - 4 = 6$	$141 - 2 = 139$ TEUs	TEUs
1	139	1	6	$q_1 - R_1$	$y - w_1$	136
	TEUs			$6 - 6 = 0$	$139 - 3 = 136$ TEUs	TEUs

Pada tahap ini diperoleh keputusan $x_1 = 1$, artinya kontainer ke-1 dipilih masuk ke dalam knapsack. Hal ini terjadi karena nilai prioritasnya lebih besar dari 0, sehingga memberikan kontribusi terhadap total nilai R_k . Dengan kondisi kapasitas blok masih banyak, pemilihan kontainer ke-1 ini tidak menimbulkan masalah kelebihan muatan. Dalam konteks penempatan lapangan, kontainer ke-1 diprioritaskan untuk ditempatkan pada blok utama yaitu blok domestik, karena nilai prioritasnya masih memberikan keuntungan optimal.

3.3 Hasil Akhir

Tabel 14. Hasil akhir perhitungan

No.	Shipper	Size	Total Kontainer	w_k (Volume)	Penempatan Blok
1	CV. Sinar Buana	20 Feet	3	3 TEUs	Blok Utama (Domestik)
2	PT. Indofood Sukses Makmur	20 Feet	2	2 TEUs	Blok Utama (Domestik)
3	PT. Indofood Sukses Makmur	20 Feet	8	8 TEUs	Blok Utama (Domestik)
4	PT. Bogasari Flour Mills – Gudang Bogasari	20 Feet	4	4 TEUs	Blok Utama (Domestik)
5	CV Sinar Buana Jaya Perkasa – Subang	20 Feet	2	2 TEUs	Blok Utama (Domestik)
6	PT Sayap Mas Utama / SMU2 / Jambi	40 Feet	4	8 TEUs	Blok Utama (Domestik)
7	Bintang Mas – Topso	20 Feet	1	1 TEUs	Blok Utama (Domestik)
8	PT Sayap Mas Utama / MIM / Bulian	40 Feet	4	8 TEUs	Blok Utama (Domestik)
9	PT Sayap Mas Utama / AMG / Jambi	40 Feet	3	6 TEUs	Blok Utama (Domestik)
10	REPO MTY OOCL	40 Feet	4	8 TEUs	Blok buffer
Jumlah Kontainer dan Volume			35 Kontainer	50 TEUs	

Berdasarkan perhitungan manual pada 10 kategori kontainer, diperoleh nilai prioritas maksimum sebesar 105. serta dapat disimpulkan hasil akhir dengan klasifikasi setiap blok sebagai berikut:

- Blok Utama (Domestik): 31 kontainer dengan memuat penempatan sebesar 42 TEUs dari 178 TEUs
- Buffer: 4 kontainer dengan memuat penempatan sebesar 8 TEUs dari 81 TEUs

3.4 Output Python

```

=== Daftar Penempatan Kontainer ===
PT Sayap Mas Utama / AMG / Jambi      40 Feet Tinggi Blok 2
PT Sayap Mas Utama / AMG / Jambi      40 Feet Tinggi Blok 2
PT Sayap Mas Utama / AMG / Jambi      40 Feet Tinggi Blok 2
PT Sayap Mas Utama / MIM / Bulian      40 Feet Tinggi Blok 2
PT Sayap Mas Utama / MIM / Bulian      40 Feet Tinggi Blok 2
PT Sayap Mas Utama / MIM / Bulian      40 Feet Tinggi Blok 2
PT Sayap Mas Utama / MIM / Bulian      40 Feet Tinggi Blok 2
PT Sayap Mas Utama / MIM / Bulian      40 Feet Tinggi Blok 2
Bintang Mas - Topso                    20 Feet Tinggi Blok 2
PT Sayap Mas Utama / SMU2 / Jambi      40 Feet Tinggi Blok 2
PT Sayap Mas Utama / SMU2 / Jambi      40 Feet Tinggi Blok 2
PT Sayap Mas Utama / SMU2 / Jambi      40 Feet Tinggi Blok 2
PT Sayap Mas Utama / SMU2 / Jambi      40 Feet Tinggi Blok 2
CV Sinar Buana Jaya Perkasa - Subang    20 Feet Tinggi Blok 2
CV Sinar Buana Jaya Perkasa - Subang    20 Feet Tinggi Blok 2
PT. Indofood Sukses Makmur             20 Feet Sedang Blok 2
PT. Indofood Sukses Makmur             20 Feet Sedang Blok 2
PT. Indofood Sukses Makmur             20 Feet Sedang Blok 2
PT. Indofood Sukses Makmur             20 Feet Sedang Blok 2
PT. Indofood Sukses Makmur             20 Feet Sedang Blok 2
PT. Indofood Sukses Makmur             20 Feet Sedang Blok 2
PT. Indofood Sukses Makmur             20 Feet Sedang Blok 2
PT. Indofood Sukses Makmur             20 Feet Sedang Blok 2
PT. Indofood Sukses Makmur             20 Feet Sedang Blok 2
PT. Indofood Sukses Makmur             20 Feet Sedang Blok 2
PT. Indofood Sukses Makmur             20 Feet Sedang Blok 2
CV. Sinar Buana                        20 Feet Sedang Blok 2
CV. Sinar Buana                        20 Feet Sedang Blok 2
CV. Sinar Buana                        20 Feet Sedang Blok 2
PT. Bogasari Flour Mills - Gudang Bogasari 20 Feet Rendah Blok 2
PT. Bogasari Flour Mills - Gudang Bogasari 20 Feet Rendah Blok 2
PT. Bogasari Flour Mills - Gudang Bogasari 20 Feet Rendah Blok 2
PT. Bogasari Flour Mills - Gudang Bogasari 20 Feet Rendah Blok 2
REPO MTY OOCL                          40 Feet 0 Buffer
REPO MTY OOCL                          40 Feet 0 Buffer
REPO MTY OOCL                          40 Feet 0 Buffer
REPO MTY OOCL                          40 Feet 0 Buffer

=== Ringkasan Per Blok ===
Blok 1: 0 kontainer, 0 TEU dari 502 TEU
>> Nilai R Total = 0
Blok 2: 31 kontainer, 42 TEU dari 178 TEU
- Tinggi: 25 TEU → R = 75
- Sedang: 13 TEU → R = 26
- Rendah: 4 TEU → R = 4
>> Nilai R Total = 105
Buffer: 4 kontainer, 8 TEU dari 81 TEU
>> Nilai R Total = 0
PS C:\New folder>

```

4. KESIMPULAN

Berdasarkan hasil analisis penempatan kontainer di Pelindo Regional 2 Jambi, mayoritas kontainer 20 Feet maupun 40 Feet ditempatkan di blok utama (domestik) dengan total 31 kontainer atau 42 TEUs, sedangkan blok *buffer* menampung 4 kontainer atau 8 TEUs yang merupakan kontainer MTY dengan prioritas rendah. Penelitian ini menunjukkan bahwa dengan perencanaan semi-otomatis sebelum kedatangan kapal, Pelindo Regional 2 Jambi dapat memprediksi alokasi kontainer secara lebih efisien, memaksimalkan kapasitas blok utama untuk operasional rutin, dan menggunakan blok *buffer* sebagai area cadangan untuk

kontainer prioritas rendah. Dengan demikian, strategi ini mempermudah pengelolaan terminal, mempercepat proses penanganan kontainer, dan meningkatkan efisiensi operasional secara keseluruhan.

DAFTAR PUSTAKA

- Battum, C. H., Wiegman, B., Atasoy, B., Wingerden, E. V., Waal, D. A., & Tavasszy, L. A. (2022). Performance improvements in container terminals through the bottleneck mitigation cycle. *Maritime Economics and Logistics*, 25(1), 174 - 195.
- Christopher, M. (2016). *Logistics and Supply Chain Management: Logistics & Supply Chain Management 5th Edition*. New York: FT Publishing International.
- Devita, N. R., & Wibawa, A. P. (2020). Teknik-Teknik Optimasi Knapsack Problem. *Sains, Aplikasi, Komputasi, dan Teknologi Informasi*, 2(1), 35-40.
- Harmasendro, M. (2023). Penerapan Pemrograman Dinamis dalam Mengoptimalkan Pemilihan Destinasi Wisata dengan Batasan Biaya dan Rating. *Teknik Elektro dan Informatika*, 1(3), 1 - 9.
- Havidzi, M. R., Nugraha, B., & Gupron, A. K. (2024). Analisis Kendala Pelayanan Jasa Bongkar Muat Peti Kemas Terhadap . *Jurnal Riset Sosial Humaniora dan Ilmu Pendidikan*, Vol 3 (3), 248 - 275.
- Hillier, F. S., & Lieberman, G. J. (2021). *Introduction to Operations Research*. New York: McGraw-Hill Education.
- Insights, B. R. (2023). *Terminal Operating Systems (TOS) Market Size, Growth.....* Business Research Insights.
- Laura, S., & Aardal, K. (2022). *Integer Programming and Combinatorial Optimization*. Netherlands: Springer Cham.
- Ling, P., Tan, C., Lee, Y., & Mohammad, S. (2020). Technical Information on ISO Shipping Container. *IOP Conference Series: Materials Science and Engineering*, 2(1), 1 - 7.
- Nahry, Tjahjono, T., Iriantika, B., & Semedi, J. (2016). Location Analysis of Freight Distribution Terminal of Jakarta City, Indonesia. *Civil Engineering Dimension*, 18(1), 38 - 48.
- Pelindo. (2023). 1. Retrieved from Tentang Kami - Pelindo: pelindo.co.id
- Sabaruddin, R. (2016). Solusi Optimum Minmax 0/1 Knapsack Menggunakan Algoritma Greedy. *Jurnal Evolusi*, 4(2), 76 - 82.
- Song, D. W., & Panayides, P. (2021). *Maritime Logistics : A Guide to Contemporary Shipping and Port Management*. New York: KoganPage.
- Song, D. P. (2021). *Container Logistics and Maritime Transport*. London: Routledge.
- Surjawan, D. J., & Susanto, I. (2015). Aplikasi Optimalisasi Muat Barang Dengan Penerapan Algoritma Dynamic Programming Pada Persoalan Integer Knapsack. *Jurnal Teknik Informatika dan Sistem Informasi*, 1(2), 151-162.
- UNCTAD. (2021). *REVIEW OF MARITIME TRANSPORT*. New York: United Nations.
- Vaze, R. (2024). *Knapsack*. England: Cambridge University Press.
- Weerasinghe, B. A., Perera, H. N., & Bai, X. (2024). Optimizing container terminal operations: a systematic review of operations research applications. *Maritime Economics & Logistics*, 26(2), 307-341.