

## **BAB 5**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Berdasarkan hasil dan pembahasan yang telah dijelaskan pada bab sebelumnya, maka dapat disimpulkan bahwa :

1. Model optimasi yang diperoleh untuk *integer knapsack problem* 0/1 menggunakan algoritma *dynamic programming* ini adalah sebagai berikut :

$$f_i(y) = \begin{cases} f_{i-1}(y) & \text{jika } y < w_k \\ \max\{f_{i-1}(y), p_k + f_{i-1}(y - w_k)\} & \text{jika } y \geq w_k \end{cases}$$

Dengan  $i$  menyatakan banyaknya barang dan  $y$  menyatakan kapasitas muatan, dapat disimpulkan :

Tanggal 1 Juli 2025 terdapat 34 barang yang masuk dengan kapasitas muatan 700 kg

Tanggal 2 Juli 2025 terdapat 44 barang yang masuk, dan kapasitas muatan 673 kg

Tanggal 3 Juli 2025 terdapat 73 barang yang masuk, dan kapasitas muatan 650 kg

2. Pada data studi kasus *J&T Cargo* Purwokerto tanggal 1-3 Juli, diperoleh hasil penyelesaian *integer knapsack problem* 0/1 menggunakan *dynamic programming* sebagai berikut :

- a. Tanggal 1 Juli 2025 setelah dilakukan perhitungan menggunakan algoritma *dynamic programming* diperoleh 32 barang yang akan diangkut pada hari itu dengan berat maksimum sebesar 700 kg dan keuntungan maksimum yang diperoleh sebesar Rp3.038.850.
- b. Tanggal 2 Juli 2025 setelah dilakukan perhitungan menggunakan algoritma *dynamic programming* diperoleh total 43 barang yang akan diangkut pada hari itu dengan berat maksimum 700 kg dan keuntungan maksimum yang diperoleh sebesar Rp4.884.985.
- c. Tanggal 3 Juli 2025 setelah dilakukan perhitungan menggunakan algoritma *dynamic programming* diperoleh total 22 barang yang akan

diangkut pada hari itu dengan berat maksimum 699 kg dan keuntungan maksimum yang diperoleh sebesar Rp7.732.155.

## 5.2 Saran

Untuk penelitian selanjutnya dapat dikembangkan dengan menggunakan variabel lain seperti variabel volume barang sehingga tidak hanya menggunakan variabel berat barang dan keuntungan barang saja dan dapat menggunakan algoritma *dynamic programming* rekursif mundur atau menggunakan algoritma lain yang dapat menyelesaikan permasalahan *integer knapsack*. Selanjutnya, juga bisa dikembangkan dengan menggunakan permasalahan knapsack yang berbeda seperti *bounded knapsack* atau *fractional knapsack* untuk menyesuaikan permasalahan pada realita yang ada.

