

BAB III

ANALISIS DAN PERANCANGAN

3.1 Deskripsi Masalah

Diberikan n buah objek $O = \{o_1, o_2, \dots, o_n\}$ dengan masing-masing berat (*weight*) $B = \{w_1, w_2, \dots, w_n\}$, masing-masing nilai (*profit/value*) $V = \{v_1, v_2, \dots, v_n\}$ dan sebuah *knapsack* dengan kapasitas maksimum W . Permasalahannya bagaimana cara memilih objek-objek yang akan dimuat ke dalam *knapsack* sehingga memaksimalkan keuntungan tetapi total berat objek yang dipilih tidak melebihi kapasitas maksimum *knapsack*. Dalam hal ini diperbolehkan untuk mengambil objek secara fraksional sebesar x_i dengan $0 \leq x_i \leq 1$. Dengan demikian, komponen-komponen dari masalah *fractional knapsack* ini terdiri dari :

1. Banyaknya objek yang tersedia.
2. Profit / nilai (*value*) dari masing-masing objek.
3. Berat / bobot (*weight*) masing-masing objek.
4. Kapasitas maksimum media penyimpanan.
5. Profit / nilai maksimum yang didapat dari pemilihan objek.

3.2 Formulasi Masalah *Fractional Knapsack*

Formulasi secara matematis masalah *fractional knapsack* dapat ditulis:

Maksimumkan
$$Z = \sum_{i=1}^n v_i x_i$$

Dengan kendala
$$\sum_{i=1}^n w_i x_i \leq W$$

$$0 \leq x_i \leq 1, \quad i = 1, 2, 3 \dots n$$

Dimana, Z = Fungsi tujuan yang akan dicapai (memaksimumkan).

x_i = Besarnya fraksional objek yang akan dipilih.

v_i = Nilai (*value*) dari masing-masing objek.

w_i = Berat/bobot (*weight*) masing-masing objek.

W = Kapasitas media penyimpanan.

n = Banyaknya objek yang tersedia.

3.3 Inisialisasi Masalah

Salah satu permasalahan *fractional knapsack* adalah masalah optimisasi dalam hal investasi modal. Berikut ini adalah contoh kasusnya.

Misalkan tersedia modal sebesar \$14.000 diinvestasikan ke dalam 4 jenis kesempatan usaha. Informasi mengenai permodalan dan perkiraan profit dalam usaha tersebut ditunjukkan dalam bentuk **Tabel 3.1**, yaitu:

Tabel 3.1 Data investasi

	Usaha 1	Usaha 2	Usaha 3	Usaha 4
Perkiraan Profit	\$2.000	\$3.000	\$1.000	\$1.000
Besar investasi	\$5.000	\$7.000	\$4.000	\$3.000

Bagaimanakah cara menentukan besarnya investasi yang akan dialokasikan ke dalam masing-masing usaha tersebut sehingga diperoleh keuntungan (profit) maksimum?

Formulasi secara matematis masalah diatas dapat ditulis sebagai berikut:

$$\text{Maksimumkan } Z = 2x_1 + 3x_2 + x_3 + x_4$$

$$\text{Dengan kendala } 5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14$$

$$0 \leq x_i \leq 1, \quad i = 1, 2, 3, 4$$

3.4 Metode Penyelesaian

Tujuan utama dari masalah *fractional knapsack* ini adalah menentukan besarnya investasi yang akan dialokasikan ke dalam masing-masing usaha tersebut sehingga diperoleh keuntungan (profit) maksimum tanpa melebihi jumlah modal yang tersedia.

Adapun algoritma yang digunakan untuk menyelesaikan masalah *fractional knapsack* adalah :

1. Algoritma *greedy*.
2. Algoritma simpleks direvisi (primal).

3.4.1 Algoritma *Greedy*

Notasi-notasi yang digunakan dalam algoritma *greedy* dalam penyelesaian *fractional knapsack* :

n = Banyaknya objek.

w_i = Berat dari tiap objek ke- i , dengan $i = 1, 2, \dots, n$.

v_i = Profit dari tiap objek ke- i , dengan $i = 1, 2, \dots, n$.

x_i = Nilai fraksional pemilihan objek, dengan $0 \leq x_i \leq 1, \quad i = 1, 2, \dots, n$

X = Array berukuran n dengan $x_i \in X$

W = Kapasitas maksimum *knapsack*.

R_w = Kapasitas kosong *knapsack* yang masih tersedia.

Berikut ini adalah langkah-langkah penyelesaian masalah *fractional knapsack* dengan menggunakan algoritma *greedy* :

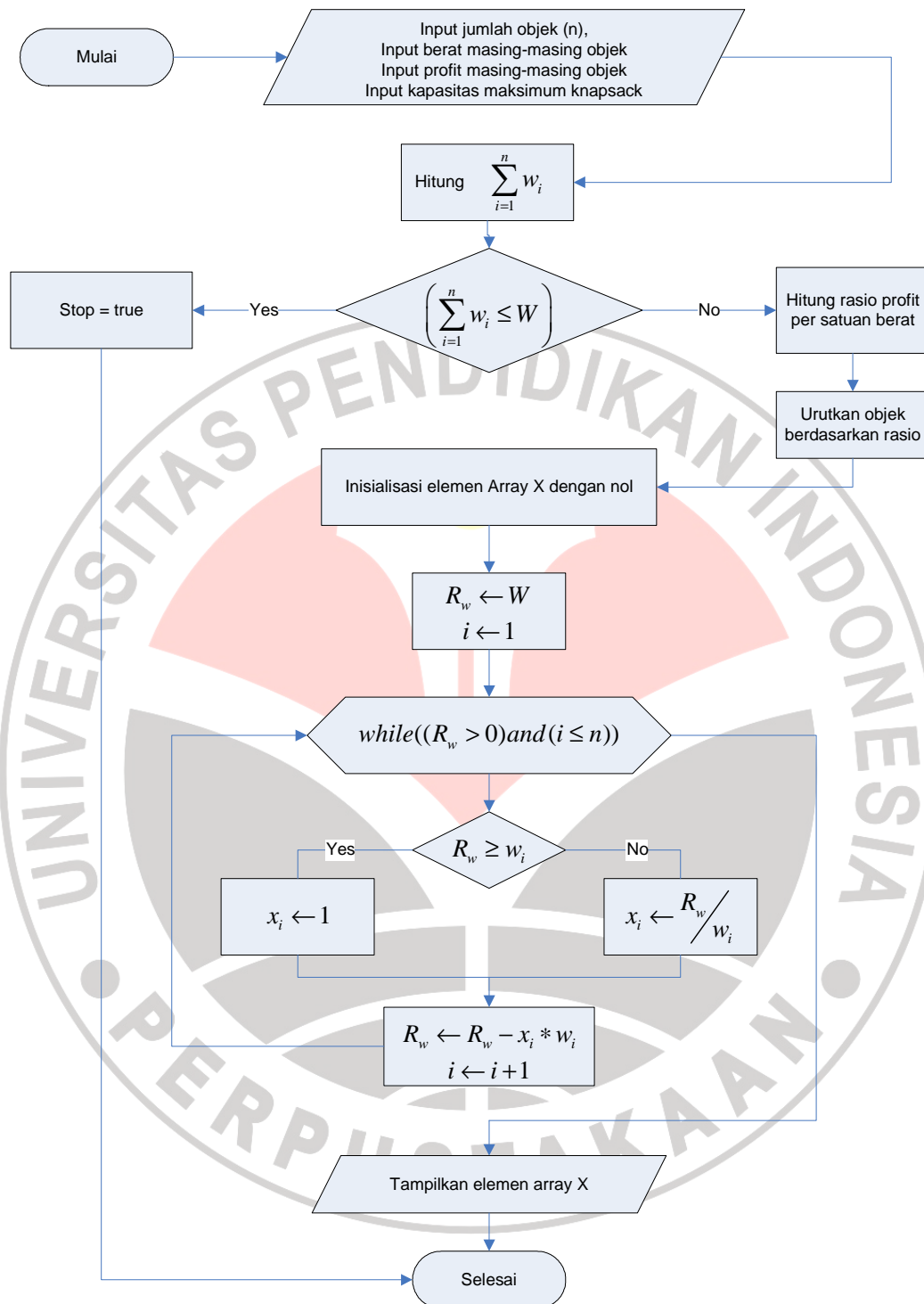
1. Hitung jumlah berat dari semua objek, yakni $\sum_{i=1}^n w_i$
2. Jika jumlah berat dari semua objek lebih kecil dari kapasitas *knapsack* $\left(\sum_{i=1}^n w_i \leq W\right)$, maka semua objek dapat diambil dan proses selesai.

Tetapi jika $\sum_{i=1}^n w_i > W$, maka lakukan langkah selanjutnya.

3. Hitung nilai profit per satuan berat $\left(\frac{v_i}{w_i}\right)$ untuk setiap objek ke- i .
4. Urutkan objek secara menurun berdasarkan rasio profit per satuan berat $\left(\frac{v_1}{w_1} \geq \frac{v_2}{w_2} \geq \dots \geq \frac{v_n}{w_n} \geq 0\right)$.
5. Inialisasi *array* X dengan nol ($X = (0, 0, \dots, 0)$).
6. Inialisasi R_w dengan kapasitas muat ($R_w \leftarrow W$), dan inialisasi i dengan 1 ($i \leftarrow 1$).
7. Selama *knapsack* belum penuh ($R_w > 0$) dan kandidat objek yang akan dimuat masih tersedia ($i \leq n$), lakukan langkah 8. Jika kondisi tersebut tidak terpenuhi, lakukan langkah 11.

8. Jika $R_w \geq w_i$ maka nilai x_i diisi dengan 1 ($x_i \leftarrow 1$), jika $R_w < w_i$, maka nilai x_i diisi dengan $\frac{R_w}{w_i}$.
9. Periksa kapasitas muat yang masih tersedia pada *knapsack* ($R_w \leftarrow R_w - x_i * w_i$)
10. $i \leftarrow i + 1$, kembali ke langkah 7.
11. Tampilkan komponen X.

Agar langkah-langkah penyelesaian masalah *fractional knapsack* dengan menggunakan algoritma *greedy* dapat dipahami dengan lebih jelas, karenanya diperlukan diagram alir (*flow chart*) yang menggambarkan uraian proses algoritma tersebut. Adapun diagram alir algoritma *greedy* untuk masalah *fractional knapsack* disajikan pada **Gambar 3.1**.



Gambar 3.1 Diagram alir algoritma *greedy* untuk masalah *fractional knapsack*

Penyelesaian masalah *fractional knapsack* untuk persoalan pada sub bab 3.3 dengan menggunakan algoritma *greedy* adalah sebagai berikut:

Langkah 1

$$\sum_{i=1}^4 w_i = 5 + 7 + 4 + 3 = 19$$

Langkah 2

Karena $\sum_{i=1}^4 w_i = 19 \leq W = 14$, lakukan langkah 3

Langkah 3

Nilai rasio $\left(\frac{v_i}{w_i}\right)$ dari setiap objek ditunjukkan pada tabel berikut

Tabel 3.2 Data investasi

	Usaha 1	Usaha 2	Usaha 3	Usaha 4
Perkiraan Profit	2	3	1	1
Besar investasi	5	7	4	3
Rasio	0.4	0.42857143	0.25	0.33333334

Formulasi secara matematis masalah tersebut masih tetap seperti semula:

$$\text{Maksimumkan } Z = 2x_1 + 3x_2 + x_3 + x_4$$

$$\text{dengan kendala } 5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14$$

$$0 \leq x_i \leq 1, \quad i = 1, 2, 3, 4$$

Langkah 4

Nilai rasio $\left(\frac{v_i}{w_i}\right)$ terurut secara menurun dari setiap objek ditunjukkan

pada tabel berikut:

Tabel 3.3 Data investasi terurut berdasarkan rasio

	Usaha 2	Usaha 1	Usaha 4	Usaha 3
Urutan Objek	Objek 1	Objek 2	Objek 3	Objek 4
Perkiraan Profit	3	2	1	1
Besar investasi	7	5	3	4
Rasio	0.42857143	0.4	0.33333334	0.25

Berdasarkan **Tabel 3.3**, formulasi secara matematis masalah tersebut disusun ulang menjadi:

$$\text{Maksimumkan } Z = 3x_1 + 2x_2 + x_3 + x_4$$

$$\text{dengan kendala } 7x_1 + 5x_2 + 3x_3 + 4x_4 \leq 14$$

$$0 \leq x_i \leq 1, \quad i = 1, 2, 3, 4$$

Dalam hal ini,

Langkah 5

Inisialisasi *array* X dengan nol. Dengan demikian nilai $x_i = 0; \quad i = 1, 2, 3, 4$

Langkah 6

$R_w \leftarrow W$ artinya knapsack masih kosong, $i \leftarrow 1$ menandakan bahwa objek 1 akan diseleksi untuk diambil.

Langkah 7

Karena knapsack belum penuh ($R_w = 14 > 0$) dan kandidat objek yang akan dimuat ($i = 1 \leq n = 4$) masih tersedia, lakukan langkah 8.

Langkah 8

Karena $R_w = 14 \geq w_1 = 7$ maka nilai $x_1 = 1$

Langkah 9

Saat ini nilai $R_w = 14 - 1 \times 7 = 7$

Langkah 10

Saat ini nilai $i = 1 + 1 = 2$, kembali ke langkah 7.

Langkah 7

Karena *knapsack* belum penuh ($R_w = 7 > 0$) dan kandidat objek yang akan dimuat ($i = 2 \leq n = 4$) masih tersedia, lakukan langkah 8.

Langkah 8

Karena $R_w = 7 \geq w_2 = 5$ maka nilai $x_2 = 1$

Langkah 9

Saat ini nilai $R_w = 7 - 1 \times 5 = 2$

Langkah 10

Saat ini nilai $i = 2 + 1 = 3$, kembali ke langkah 7.

Langkah 7

Karena *knapsack* belum penuh ($R_w = 2 > 0$) dan kandidat objek yang akan dimuat ($i = 3 \leq n = 4$) masih tersedia, lakukan langkah 8.

Langkah 8

Karena $R_w = 2 < w_3 = 3$ maka nilai $x_3 = \frac{R_w}{w_3} = \frac{2}{3} \approx 0.6666667$

Langkah 9

Saat ini nilai $R_w = 2 - \frac{2}{3} \times 3 = 0$

Langkah 10

Saat ini nilai $i = 3 + 1 = 4$, kembali ke langkah 7.

Langkah 7

Karena *knapsack* sudah penuh ($R_w = 0$) tapi kandidat objek yang akan dimuat ($i = 4 \leq n = 4$) masih tersedia, lakukan langkah 11.

Langkah 11

Komponen *array* X saat ini adalah $X = (x_1, x_2, x_3, x_4) = \left(1, 1, \frac{2}{3}, 0\right)$

Dengan demikian, keuntungan maksimum yang diperoleh adalah

$$Z = 3(1) + 2(1) + 1\left(\frac{2}{3}\right) + 1(0) = 5\frac{2}{3} \approx 5,6666667$$

Dalam hal ini keuntungan yang sebenarnya adalah $5\frac{2}{3} \times (\$1.000)$

3.4.2 Algoritma Simpleks Direvisi (Primal)

Notasi-notasi yang digunakan dalam algoritma simpleks direvisi (primal) dalam penyelesaian *fractional knapsack* :

n = Banyaknya objek.

w_i = Berat (*weight*) dari tiap objek ke- i , dengan $i = 1, 2, \dots, n$

v_i = Profit/nilai (*value*) dari tiap objek ke- i , dengan $i = 1, 2, \dots, n$

x_i = Nilai fraksional pemilihan objek, dengan $0 \leq x_i \leq 1$, $i = 1, 2, \dots, n$

s_k = Variabel *slack*, dengan $s_k > 0$, $k = 1, 2, \dots, n + 1$

W = Kapasitas maksimum *knapsack*

$\mathbf{X}_{\text{basis}}$ = Matriks variabel basis

$\mathbf{X}_{\text{non_basis}}$ = Matriks variabel non basis

$\mathbf{B}_{\text{old}}^{-1}$ = Matriks invers basis lama

$\mathbf{B}_{\text{new}}^{-1}$ = Matriks invers basis baru

\mathbf{I} = Matriks identitas

\mathbf{P} = Matriks non basis

\mathbf{P}_i = Vektor kolom pada matriks non basis

\mathbf{Q} = Matriks basis

\mathbf{Q}_k = Vektor kolom pada matriks basis

$\mathbf{C}_{\text{basis}}$ = Matriks konstanta basis

$\mathbf{C}_{\text{non_basis}}$ = Matriks konstanta non basis

\mathbf{X}_B = Matriks konstanta basis

\mathbf{R}_i = Vektor kolom yang meninggalkan basis

α_i = Elemen matriks ξ

α_r = Elemen pivot

\mathbf{e}_k = Vektor kolom matriks identitas

Adapun langkah-langkah penyelesaian masalah *fractional knapsack* dengan menggunakan algoritma simpleks direvisi (primal) adalah sebagai berikut :

1. Hitung jumlah berat dari semua objek, yakni $\sum_{i=1}^n w_i$

2. Jika jumlah berat dari semua objek lebih kecil dari kapasitas *knapsack*

$$\left(\sum_{i=1}^n w_i \leq W \right), \text{ maka semua objek dapat diambil dan proses selesai.}$$

Tetapi jika $\sum_{i=1}^n w_i > W$, maka lakukan langkah selanjutnya.

3. Tambahkan variabel *slack* pada semua kendala (*constraint*) sehingga bentuk LP menjadi:

Maksimumkan $Z = \sum_{i=1}^n v_i x_i$

Dengan kendala $\left(\sum_{i=1}^n w_i x_i \right) + s_1 = W$

$$x_i + s_{i+1} = b_i$$

$$x_i \geq 0; \quad i = 1, 2, \dots, n$$

$$s_k \geq 0; \quad k = 1, 2, \dots, n+1$$

Jika $\left(\frac{W}{w_i} \geq 1 \right)$ maka nilai $b_i = 1$, tapi jika $\left(\frac{W}{w_i} < 1 \right)$ maka nilai $b_i = \frac{W}{w_i}$

Bentuk di atas dapat diuraikan menjadi bentuk array berikut:

Array dengan ukuran $1 \times n$	Array dengan ukuran $1 \times (n+1)$	
$v_1 \quad v_2 \quad \dots \quad v_{n-1} \quad v_n$	$0 \quad 0 \quad 0 \quad \dots \quad 0 \quad 0$	
$w_1 \quad w_2 \quad \dots \quad w_{n-1} \quad w_n$	$1 \quad 0 \quad 0 \quad \dots \quad 0 \quad 0$	W b_1 b_2 \vdots b_{n-1} b_n
$1 \quad 0 \quad \dots \quad 0 \quad 0$	$0 \quad 1 \quad 0 \quad \dots \quad 0 \quad 0$	
$0 \quad 1 \quad \dots \quad 0 \quad 0$	$0 \quad 0 \quad 1 \quad \dots \quad 0 \quad 0$	
$\vdots \quad \vdots$	$\vdots \quad \vdots$	
$0 \quad 0 \quad \dots \quad 1 \quad 0$	$0 \quad 0 \quad 0 \quad \dots \quad 1 \quad 0$	
$0 \quad 0 \quad \dots \quad 0 \quad 1$	$0 \quad 0 \quad 0 \quad \dots \quad 0 \quad 1$	
Array dengan ukuran $(n+1) \times n$	Array dengan ukuran $(n+1) \times (n+1)$	

4. Inisialisasi matriks

$$\mathbf{X}_{\text{basis}} = \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \\ s_{n+1} \end{pmatrix} \quad \mathbf{X}_{\text{non_basis}} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \quad \mathbf{b} = \begin{pmatrix} W \\ b_1 \\ b_2 \\ \vdots \\ b_{n-1} \\ b_n \end{pmatrix}$$

$$\mathbf{B}_{\text{old}}^{-1} = \mathbf{B}_{\text{new}}^{-1} = \mathbf{I} = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & & & & \vdots & \\ 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \end{pmatrix}$$

$$\mathbf{P} = (\mathbf{P}_1 \quad \mathbf{P}_2 \quad \dots \quad \mathbf{P}_{n-1} \quad \mathbf{P}_n) = \begin{pmatrix} w_1 & w_2 & \dots & w_{n-1} & w_n \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & & & \vdots & \\ 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix}$$

$$\mathbf{Q} = (\mathbf{Q}_1 \quad \mathbf{Q}_2 \quad \dots \quad \mathbf{Q}_n \quad \mathbf{Q}_{n+1}) = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & & & & \vdots & \\ 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \end{pmatrix}$$

$$\mathbf{C}_{\text{basis}} = \underbrace{(0 \quad 0 \quad 0 \quad \dots \quad 0 \quad 0)}_{\text{Matriks dengan ukuran } l \times (n+1)}$$

$$\mathbf{C}_{\text{non_basis}} = (v_1 \quad v_2 \quad \dots \quad v_{n-1} \quad v_n)$$

Iterasi saat ini bernilai nol.

5. Hitung $z_j - c_j$ untuk non basis, dalam hal ini adalah matriks \mathbf{P} , dengan hubungan $z_j - c_j = \mathbf{C}_{\text{basis}} \cdot \mathbf{B}_{\text{new}}^{-1} \cdot \mathbf{P} - \mathbf{C}_{\text{non_basis}}$. Iterasi bertambah 1.
6. Jika seluruh nilai $z_j - c_j \geq 0$ proses selesai, akhiri kalkulasi dengan menghitung nilai $\mathbf{X}_B = \mathbf{B}_{\text{new}}^{-1} \cdot \mathbf{b}$ dan $Z = \mathbf{C}_{\text{basis}} \cdot \mathbf{X}_{\text{basis}}$. Jika tidak demikian, lanjutkan ke langkah 7.
7. Pilih nilai negatif terkecil pada $z_j - c_j$. Kemudian pilih vektor kolom pada matriks \mathbf{P} yang bersesuaian dengan indeks nilai tersebut pada $z_j - c_j$. Vektor kolom ini nantinya akan masuk ke matriks \mathbf{Q} (basis) dan diganti oleh vektor kolom yang dipilih untuk meninggalkan matriks \mathbf{Q} .
8. Hitung vektor \mathbf{X}_B dengan hubungan $\mathbf{X}_B = \mathbf{B}_{\text{new}}^{-1} \cdot \mathbf{b}$
9. Hitung vektor $\boldsymbol{\alpha}$ dengan hubungan $\boldsymbol{\alpha} = \mathbf{B}_{\text{new}}^{-1} \cdot \mathbf{R}_i$, dengan \mathbf{R}_i adalah vektor kolom yang dipilih pada langkah 7.
10. Tentukan $\theta = \min_{>0} \left\{ \frac{\mathbf{X}_B}{\boldsymbol{\alpha}}; \boldsymbol{\alpha} > \mathbf{0} \right\}$, indeks elemen matriks $\boldsymbol{\alpha}$ dipilih sebagai elemen pivot kemudian pilih vektor kolom pada matriks \mathbf{Q} yang bersesuaian dengan indeks elemen pivot tersebut. Vektor kolom ini bertukar tempat dengan vektor kolom yang telah dipilih pada langkah 7.

11. Tentukan matriks $\boldsymbol{\xi} = \begin{pmatrix} -\alpha_1/\alpha_r \\ -\alpha_2/\alpha_r \\ \vdots \\ 1/\alpha_r \\ \vdots \\ -\alpha_{n+1}/\alpha_r \end{pmatrix}$ dengan α_i adalah elemen matriks $\boldsymbol{\alpha}$,

dan α_r adalah elemen pivot.

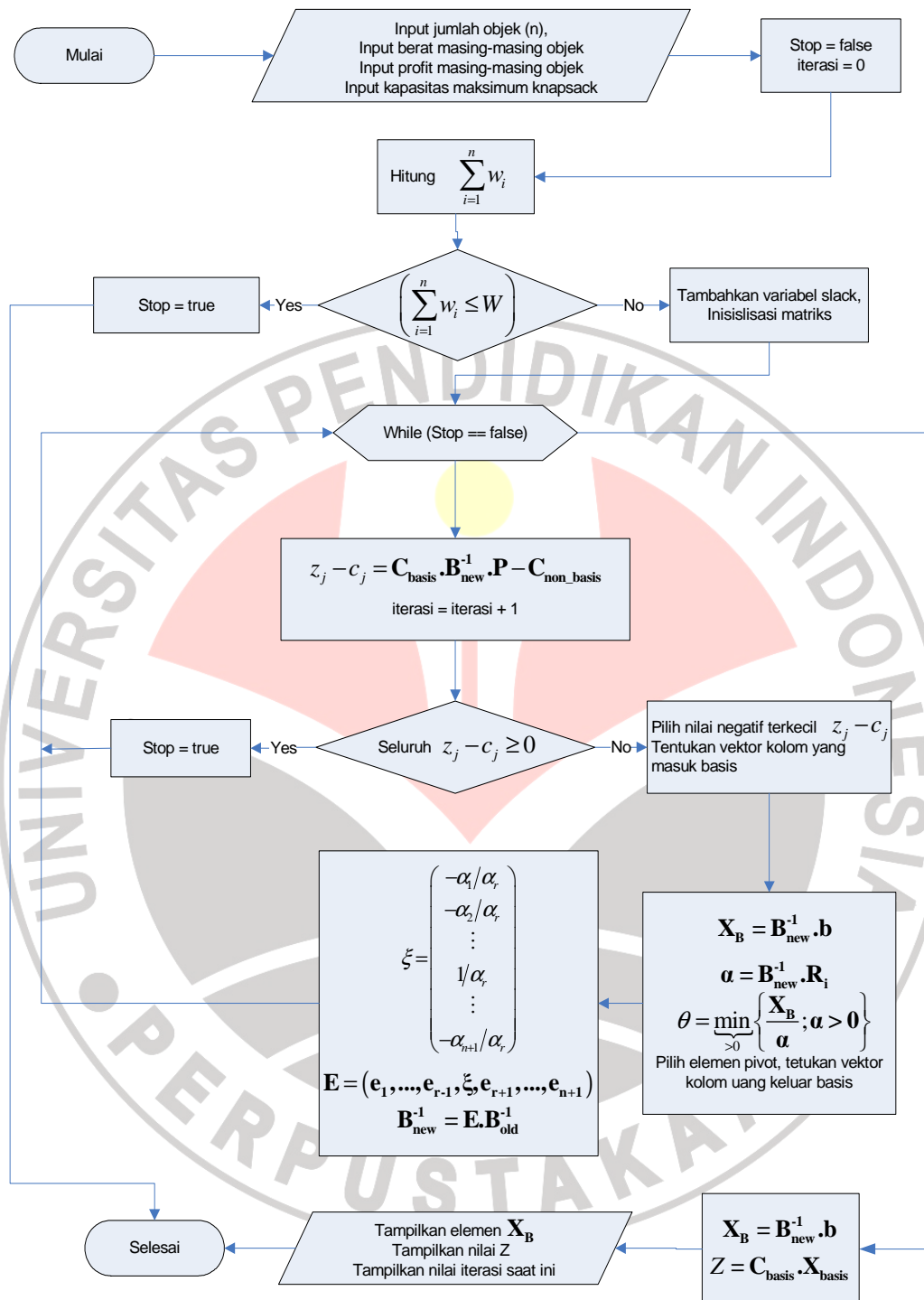
12. Tentukan matriks $\mathbf{E} = (\mathbf{e}_1, \dots, \mathbf{e}_{r-1}, \xi, \mathbf{e}_{r+1}, \dots, \mathbf{e}_{n+1})$, dengan \mathbf{e}_i adalah vektor kolom pada matriks identitas \mathbf{I} .

13. Tentukan matriks $\mathbf{B}_{\text{new}}^{-1} = \mathbf{E} \cdot \mathbf{B}_{\text{old}}^{-1}$

14. Tentukan elemen matriks $\mathbf{X}_{\text{basis}}$, $\mathbf{X}_{\text{non_basis}}$, $\mathbf{C}_{\text{basis}}$, \mathbf{Q} , $\mathbf{C}_{\text{non_basis}}$, dan \mathbf{P} , kemudian ganti seluruh elemen matriks $\mathbf{B}_{\text{old}}^{-1}$ dengan matriks $\mathbf{B}_{\text{new}}^{-1}$.

Kembali ke langkah 5.

Agar langkah-langkah penyelesaian masalah *fractional knapsack* dengan menggunakan algoritma simpleks direvisi (primal) dapat dipahami dengan lebih jelas, karenanya diperlukan diagram alir (*flow chart*) yang menggambarkan uraian proses algoritma tersebut. Adapun diagram alir algoritma simpleks direvisi (primal) untuk masalah *fractional knapsack* disajikan pada **Gambar 3.2**.



Gambar 3.2 Diagram alir algoritma simpleks direvisi (primal) untuk masalah *fractional knapsack*

Penyelesaian masalah *fractional knapsack* untuk kasus pada sub bab 3.3 dengan menggunakan metode algoritma simpleks direvisi (primal) adalah sebagai berikut:

Langkah 1

$$\sum_{i=1}^4 w_i = 5 + 7 + 4 + 3 = 19$$

Langkah 2

Karena $\sum_{i=1}^4 w_i = 19 \leq W = 14$, lakukan langkah 3

Langkah 3

Tabel 3.4 Data investasi

	Usaha 1	Usaha 2	Usaha 3	Usaha 4
Perkiraan	2	3	1	1
Besar investasi	5	7	4	3

Formulasi secara matematis masalah diatas masih tetap seperti semula:

$$\text{Maksimumkan } Z = 2x_1 + 3x_2 + x_3 + x_4$$

$$\text{dengan kendala } 5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14$$

$$0 \leq x_i \leq 1, \quad i = 1, 2, 3, 4$$

Setelah ditambahkan variabel *slack*, bentuk LP menjadi

$$\text{Maksimumkan } 2x_1 + 3x_2 + x_3 + x_4 \qquad = Z$$

Dengan kendala :

$$5x_1 + 7x_2 + 4x_3 + 3x_4 + s_1 \qquad = 14$$

$$x_1 \qquad \qquad \qquad + s_2 \qquad = 1$$

$$x_2 \qquad \qquad \qquad + s_3 \qquad = 1$$

$$x_3 \qquad \qquad \qquad + s_4 \qquad = 1$$

$$x_4 \qquad \qquad \qquad + s_5 \qquad = 1$$

Karena $\left(\frac{W}{w_i} \geq 1\right)$ maka nilai $b_i = 1$.

Bentuk di atas dapat diuraikan menjadi bentuk *array* berikut:

$$\begin{array}{|cccc|cccc|} \hline 2 & 3 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ \hline 5 & 7 & 4 & 3 & 1 & 0 & 0 & 0 & 0 & 14 \\ \hline 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ \hline 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ \hline 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ \hline 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ \hline \end{array}$$

Langkah 4

Inisialisasi matriks

$$\mathbf{X}_{\text{basis}} = \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{pmatrix} \quad \mathbf{X}_{\text{non_basis}} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \quad \mathbf{b} = \begin{pmatrix} 14 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

$$\mathbf{B}_{\text{old}}^{-1} = \mathbf{B}_{\text{new}}^{-1} = \mathbf{I} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix};$$

$$\mathbf{P} = (\mathbf{P}_1 \quad \mathbf{P}_2 \quad \mathbf{P}_3 \quad \mathbf{P}_4) = \begin{pmatrix} 5 & 7 & 4 & 3 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{Q} = (\mathbf{Q}_1 \quad \mathbf{Q}_2 \quad \mathbf{Q}_3 \quad \mathbf{Q}_4 \quad \mathbf{Q}_5) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{C}_{\text{basis}} = (0 \quad 0 \quad 0 \quad 0 \quad 0)$$

$$\mathbf{C}_{\text{non_basis}} = (2 \quad 3 \quad 1 \quad 1)$$

Iterasi saat ini bernilai nol.

Langkah 5

$$z_j - c_j = \mathbf{C}_{\text{basis}} \cdot \mathbf{B}_{\text{new}}^{-1} \cdot \mathbf{P} - \mathbf{C}_{\text{non_basis}} = \begin{pmatrix} -2 & \underbrace{-3}_{\text{min}} & -1 & -1 \end{pmatrix}$$

Iterasi saat ini adalah 1

Langkah 6

Karena tidak semua nilai $z_j - c_j \geq 0$, proses berlanjut

Langkah 7

Nilai negatif terkecil pada matriks $z_j - c_j = -3$ pada indeks 2. Dengan

demikian $\mathbf{P}_2 = \begin{pmatrix} 7 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$ dipilih untuk masuk ke matriks \mathbf{Q} (basis).

Langkah 8

$$\mathbf{X}_B = \mathbf{B}_{\text{new}}^{-1} \cdot \mathbf{b} = \begin{pmatrix} 14 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Langkah 9

$$\boldsymbol{\alpha} = \mathbf{B}_{\text{new}}^{-1} \cdot \mathbf{R}_i = \begin{pmatrix} 7 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \text{ dengan } \mathbf{R}_i = \mathbf{P}_2 \text{ yakni vektor kolom yang dipilih untuk}$$

masuk ke basis pada langkah 7.

Langkah 10

$$\begin{aligned} \theta &= \min_{>0} \left\{ \frac{\mathbf{X}_B}{\boldsymbol{\alpha}} ; \boldsymbol{\alpha} > \mathbf{0} \right\} \\ &= \min_{>0} \{ 14/7 \quad - \quad 1/1 \quad - \quad - \} \\ &= 1 \end{aligned}$$

Dengan demikian elemen pivot $\alpha_r = 1$ bersesuaian dengan indeks ke-3 pada matriks $\boldsymbol{\alpha}$, dengan demikian \mathbf{Q}_3 meninggalkan basis diganti oleh \mathbf{P}_2

Langkah 11

$$\boldsymbol{\xi} = \begin{pmatrix} -\alpha_1/\alpha_r \\ -\alpha_2/\alpha_r \\ 1/\alpha_r \\ -\alpha_4/\alpha_r \\ -\alpha_5/\alpha_r \end{pmatrix} = \begin{pmatrix} -7/1 \\ -0/1 \\ 1/1 \\ -0/1 \\ -0/1 \end{pmatrix} = \begin{pmatrix} -7 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

Langkah 12

$$\mathbf{E} = (\mathbf{e}_1 \quad \mathbf{e}_2 \quad \xi \quad \mathbf{e}_4 \quad \mathbf{e}_5) = \begin{pmatrix} 1 & 0 & -7 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Langkah 13

$$\mathbf{B}_{\text{new}}^{-1} = \mathbf{E} \cdot \mathbf{B}_{\text{old}}^{-1} = \begin{pmatrix} 1 & 0 & -7 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Langkah 14

Diperoleh:

$$\mathbf{X}_{\text{basis}} = \begin{pmatrix} s_1 \\ s_2 \\ x_2 \\ s_4 \\ s_5 \end{pmatrix}; \quad \mathbf{X}_{\text{non_basis}} = \begin{pmatrix} x_1 \\ s_3 \\ x_3 \\ x_4 \end{pmatrix};$$

$$\mathbf{P} = (\mathbf{P}_1 \quad \mathbf{Q}_3 \quad \mathbf{P}_3 \quad \mathbf{P}_4) = \begin{pmatrix} 5 & 0 & 3 & 4 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{Q} = (\mathbf{Q}_1 \quad \mathbf{Q}_2 \quad \mathbf{P}_2 \quad \mathbf{Q}_4 \quad \mathbf{Q}_5) = \begin{pmatrix} 1 & 0 & 7 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix};$$

$$\mathbf{B}_{\text{old}}^{-1} = \begin{pmatrix} 0 & 0 & -7 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{C}_{\text{basis}} = (0 \quad 0 \quad 3 \quad 0 \quad 0)$$

$$\mathbf{C}_{\text{non_basis}} = (2 \quad 0 \quad 1 \quad 1)$$

Kembali ke langkah 5

Langkah 5

$$z_j - c_j = \mathbf{C}_{\text{basis}} \cdot \mathbf{B}_{\text{new}}^{-1} \cdot \mathbf{P} - \mathbf{C}_{\text{non_basis}} = \left(\underbrace{-2}_{\text{min}} \quad 3 \quad -1 \quad -1 \right)$$

Iterasi saat ini adalah 2

Langkah 6

Karena tidak semua nilai $z_j - c_j \geq 0$ proses berlanjut

Langkah 7

Nilai negatif terkecil pada $z_j - c_j = -2$ pada indeks 1. Dengan demikian

$$\mathbf{P}_1 = \begin{pmatrix} 5 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \text{ dipilih untuk masuk ke matriks } \mathbf{Q} \text{ (basis).}$$

Langkah 8

$$\mathbf{X}_B = \mathbf{B}_{\text{new}}^{-1} \cdot \mathbf{b} = \begin{pmatrix} 7 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Langkah 9

$$\boldsymbol{\alpha} = \mathbf{B}_{\text{new}}^{-1} \cdot \mathbf{R}_i = \begin{pmatrix} 5 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \text{ dengan } \mathbf{R}_i = \mathbf{P}_1 \text{ yakni vektor kolom yang dipilih untuk}$$

masuk ke basis pada langkah 7.

Langkah 10

$$\begin{aligned} \theta &= \min_{>0} \left\{ \frac{\mathbf{X}_B}{\boldsymbol{\alpha}} ; \boldsymbol{\alpha} > \mathbf{0} \right\} \\ &= \min_{>0} \{ 7/5 \quad 1/1 \quad - \quad - \quad - \} \\ &= 1 \end{aligned}$$

Dengan demikian elemen pivot $\alpha_r = 1$ bersesuaian dengan indeks ke-2 pada matriks $\boldsymbol{\alpha}$, dengan demikian \mathbf{Q}_2 meninggalkan basis diganti oleh \mathbf{P}_1

Langkah 11

$$\boldsymbol{\xi} = \begin{pmatrix} -\alpha_1/\alpha_r \\ 1/\alpha_r \\ -\alpha_3/\alpha_r \\ -\alpha_4/\alpha_r \\ -\alpha_5/\alpha_r \end{pmatrix} = \begin{pmatrix} -5/1 \\ 1/1 \\ -0/1 \\ -0/1 \\ -0/1 \end{pmatrix} = \begin{pmatrix} -5 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Langkah 12

$$\mathbf{E} = (\mathbf{e}_1 \quad \xi \quad \mathbf{e}_3 \quad \mathbf{e}_4 \quad \mathbf{e}_5) = \begin{pmatrix} 1 & -5 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Langkah 13

$$\mathbf{B}_{\text{new}}^{-1} = \mathbf{E} \cdot \mathbf{B}_{\text{old}}^{-1} = \begin{pmatrix} 1 & -5 & -7 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Langkah 14

Diperoleh:

$$\mathbf{X}_{\text{basis}} = \begin{pmatrix} s_1 \\ x_1 \\ x_2 \\ s_4 \\ s_5 \end{pmatrix}; \quad \mathbf{X}_{\text{non_basis}} = \begin{pmatrix} s_2 \\ s_3 \\ x_3 \\ x_4 \end{pmatrix};$$

$$\mathbf{P} = (\mathbf{Q}_2 \quad \mathbf{Q}_3 \quad \mathbf{P}_3 \quad \mathbf{P}_4) = \begin{pmatrix} 0 & 0 & 3 & 4 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{Q} = (\mathbf{Q}_1 \quad \mathbf{P}_1 \quad \mathbf{P}_2 \quad \mathbf{Q}_4 \quad \mathbf{Q}_5) = \begin{pmatrix} 1 & 7 & 5 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix};$$

$$\mathbf{B}_{\text{old}}^{-1} = \begin{pmatrix} 1 & -7 & -5 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad \mathbf{C}_{\text{basis}} = (0 \quad 2 \quad 3 \quad 0 \quad 0)$$

$$\mathbf{C}_{\text{non_basis}} = (0 \quad 0 \quad 1 \quad 1)$$

Kembali ke langkah 5

Langkah 5

$$z_j - c_j = \mathbf{C}_{\text{basis}} \cdot \mathbf{B}_{\text{new}}^{-1} \cdot \mathbf{P} - \mathbf{C}_{\text{non_basis}} = \begin{pmatrix} 2 & 3 & \underbrace{-1}_{\text{min}} & -1 \end{pmatrix}$$

Iterasi saat ini adalah 3

Langkah 6

Karena tidak nilai $z_j - c_j \geq 0$ proses berlanjut

Langkah 7

Nilai negatif terkecil pada matriks $z_j - c_j = -1$ pada indeks 3. Dengan

demikian $\mathbf{P}_3 = \begin{pmatrix} 4 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$ dipilih untuk masuk ke matriks \mathbf{Q} (basis).

Langkah 8

$$\mathbf{X}_B = \mathbf{B}_{\text{new}}^{-1} \cdot \mathbf{b} = \begin{pmatrix} 2 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Langkah 9

$$\boldsymbol{\alpha} = \mathbf{B}_{\text{new}}^{-1} \cdot \mathbf{R}_i = \begin{pmatrix} 4 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \text{ dengan } \mathbf{R}_i = \mathbf{P}_3 \text{ yakni vektor kolom yang dipilih untuk}$$

masuk ke basis pada langkah 7.

Langkah 10

$$\begin{aligned} \theta &= \min_{>0} \left\{ \frac{\mathbf{X}_B}{\boldsymbol{\alpha}} ; \boldsymbol{\alpha} > \mathbf{0} \right\} \\ &= \min_{>0} \{ 2/4 \quad - \quad - \quad 1/1 \quad - \} \\ &= 2/4 \end{aligned}$$

Dengan demikian elemen pivot $\alpha_r = 2/4$ bersesuaian dengan indeks ke-1 pada matriks $\boldsymbol{\alpha}$, dengan demikian \mathbf{Q}_1 meninggalkan basis diganti oleh \mathbf{P}_3

Langkah 11

$$\boldsymbol{\xi} = \begin{pmatrix} 1/\alpha_r \\ -\alpha_2/\alpha_r \\ -\alpha_3/\alpha_r \\ -\alpha_4/\alpha_r \\ -\alpha_5/\alpha_r \end{pmatrix} = \begin{pmatrix} 1/4 \\ -0/4 \\ -0/4 \\ -1/4 \\ -0/4 \end{pmatrix} = \begin{pmatrix} 1/4 \\ 0 \\ 0 \\ -1/4 \\ 0 \end{pmatrix}$$

Langkah 12

$$\mathbf{E} = (\xi \quad \mathbf{e}_2 \quad \mathbf{e}_3 \quad \mathbf{e}_4 \quad \mathbf{e}_5) = \begin{pmatrix} 1/4 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ -1/4 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Langkah 13

$$\mathbf{B}_{\text{new}}^{-1} = \mathbf{E} \cdot \mathbf{B}_{\text{old}}^{-1} = \begin{pmatrix} 1/4 & -5/4 & -7/4 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ -1/4 & 5/4 & 7/4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Langkah 14

Diperoleh:

$$\mathbf{X}_{\text{basis}} = \begin{pmatrix} x_3 \\ x_1 \\ x_2 \\ s_4 \\ s_5 \end{pmatrix}; \quad \mathbf{X}_{\text{non_basis}} = \begin{pmatrix} s_2 \\ s_3 \\ s_1 \\ x_4 \end{pmatrix};$$

$$\mathbf{P} = (\mathbf{Q}_2 \quad \mathbf{Q}_3 \quad \mathbf{Q}_1 \quad \mathbf{P}_4) = \begin{pmatrix} 0 & 0 & 0 & 3 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{Q} = (\mathbf{P}_3 \quad \mathbf{P}_1 \quad \mathbf{P}_2 \quad \mathbf{Q}_4 \quad \mathbf{Q}_5) = \begin{pmatrix} 4 & 5 & 7 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{B}_{\text{old}}^{-1} = \begin{pmatrix} 1/4 & -5/4 & -7/4 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ -1/4 & 5/4 & 7/4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad \begin{array}{l} \mathbf{C}_{\text{basis}} = (1 \ 2 \ 3 \ 0 \ 0) \\ \mathbf{C}_{\text{non_basis}} = (0 \ 0 \ 0 \ 1) \end{array}$$

Kembali ke langkah 5

Langkah 5

$$z_j - c_j = \mathbf{C}_{\text{basis}} \cdot \mathbf{B}_{\text{new}}^{-1} \cdot \mathbf{P} - \mathbf{C}_{\text{non_basis}} = (3/4 \ 5/4 \ 1/4 \ -1/4)$$

Iterasi saat ini adalah 4

Langkah 6

Karena tidak semua nilai $z_j - c_j \geq 0$ proses berlanjut

Langkah 7

Nilai negatif terkecil pada matriks $z_j - c_j = -1/4$ pada indeks 4. Dengan

demikian $\mathbf{P}_4 = \begin{pmatrix} 3 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$ dipilih untuk masuk ke matriks \mathbf{Q} (basis).

Langkah 8

$$\mathbf{X}_B = \mathbf{B}_{\text{new}}^{-1} \cdot \mathbf{b} = \begin{pmatrix} 1/2 \\ 1 \\ 1 \\ 1/2 \\ 1 \end{pmatrix}$$

Langkah 9

$$\boldsymbol{\alpha} = \mathbf{B}_{\text{new}}^{-1} \cdot \mathbf{R}_i = \begin{pmatrix} 3/4 \\ 0 \\ 0 \\ -3/4 \\ 1 \end{pmatrix} \text{ dengan } \mathbf{R}_i = \mathbf{P}_3 \text{ yakni vektor kolom yang dipilih}$$

untuk masuk ke basis pada langkah 7.

Langkah 10

$$\begin{aligned} \theta &= \min_{>0} \left\{ \frac{\mathbf{X}_B}{\boldsymbol{\alpha}}; \boldsymbol{\alpha} > \mathbf{0} \right\} \\ &= \min_{>0} \{ 2/3 \quad - \quad - \quad -2/3 \quad 1 \} \\ &= 2/3 \end{aligned}$$

Dengan demikian elemen pivot $\alpha_r = 2/3$ bersesuaian dengan indeks ke-1 pada matriks $\boldsymbol{\alpha}$, dengan demikian \mathbf{P}_3 meninggalkan basis diganti oleh \mathbf{P}_4

Langkah 11

$$\boldsymbol{\xi} = \begin{pmatrix} 1/\alpha_r \\ -\alpha_2/\alpha_r \\ -\alpha_3/\alpha_r \\ -\alpha_4/\alpha_r \\ -\alpha_5/\alpha_r \end{pmatrix} = \begin{pmatrix} 1/3 \\ -0/4 \\ -0/4 \\ 3/4 \\ -0/4 \end{pmatrix} = \begin{pmatrix} 4/3 \\ 0 \\ 0 \\ 1 \\ -4/3 \end{pmatrix}$$

Langkah 12

$$\mathbf{E} = (\boldsymbol{\xi} \quad \mathbf{e}_2 \quad \mathbf{e}_3 \quad \mathbf{e}_4 \quad \mathbf{e}_5) = \begin{pmatrix} 4/3 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ -4/3 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Langkah 13

$$\mathbf{B}_{\text{new}}^{-1} = \mathbf{E} \cdot \mathbf{B}_{\text{old}}^{-1} = \begin{pmatrix} 1/3 & -5/3 & -7/3 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -1/3 & 5/3 & 7/3 & 0 & 1 \end{pmatrix}$$

Langkah 14

$$\mathbf{X}_{\text{basis}} = \begin{pmatrix} x_4 \\ x_1 \\ x_2 \\ s_4 \\ s_5 \end{pmatrix}; \quad \mathbf{X}_{\text{non_basis}} = \begin{pmatrix} s_2 \\ s_3 \\ s_1 \\ x_3 \end{pmatrix};$$

$$\mathbf{P} = (\mathbf{Q}_2 \quad \mathbf{Q}_3 \quad \mathbf{Q}_1 \quad \mathbf{P}_3) = \begin{pmatrix} 0 & 0 & 0 & 4 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{Q} = (\mathbf{P}_4 \quad \mathbf{P}_1 \quad \mathbf{P}_2 \quad \mathbf{Q}_4 \quad \mathbf{Q}_5) = \begin{pmatrix} 3 & 5 & 7 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{B}_{\text{old}}^{-1} = \begin{pmatrix} 1/3 & -5/3 & -7/3 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -1/3 & 5/3 & 7/3 & 0 & 1 \end{pmatrix} \quad \begin{matrix} \mathbf{C}_{\text{basis}} = (1 \ 2 \ 3 \ 0 \ 0) \\ \mathbf{C}_{\text{non_basis}} = (0 \ 0 \ 0 \ 1) \end{matrix}$$

Kembali ke langkah 5

Langkah 5

$$z_j - c_j = \mathbf{C}_{\text{basis}} \cdot \mathbf{B}_{\text{new}}^{-1} \cdot \mathbf{P} - \mathbf{C}_{\text{non_basis}} = (1/3 \ 2/3 \ 1/3 \ 1/3)$$

Iterasi saat ini adalah 5

Langkah 6

Karena semua nilai $z_j - c_j \geq 0$ proses berhenti. Kalkulasi diakhiri dengan menghitung nilai

$$\mathbf{X}_B = \mathbf{B}_{\text{new}}^{-1} \cdot \mathbf{b} = \begin{pmatrix} 1/3 & -5/3 & -7/3 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -1/3 & 5/3 & 7/3 & 0 & 1 \end{pmatrix} \begin{pmatrix} 14 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 2/3 \\ 1 \\ 1 \\ 1 \\ 1/3 \end{pmatrix}$$

Dengan demikian:

$$\mathbf{X}_{\text{basis}} = \begin{pmatrix} x_4 \\ x_1 \\ x_2 \\ s_4 \\ s_5 \end{pmatrix} = \begin{pmatrix} 2/3 \\ 1 \\ 1 \\ 1 \\ 1/3 \end{pmatrix}$$

$$Z = \mathbf{C}_{\text{basis}} \cdot \mathbf{X}_{\text{basis}} = (1 \ 2 \ 3 \ 0 \ 0) \begin{pmatrix} 2/3 \\ 1 \\ 1 \\ 1 \\ 1/3 \end{pmatrix} = 5 \frac{2}{3} \approx 5,666667.$$

Dalam hal ini keuntungan yang sebenarnya adalah $5 \frac{2}{3} \times (\$1.000)$

3.5 Perancangan Sistem

Setelah melakukan analisis terhadap masalah *fractional knapsack* dengan menggunakan algoritma *greedy* dan algoritma simpleks direvisi (primal), langkah selanjutnya adalah melakukan perancangan sistem yang bertujuan untuk memberikan gambaran tentang program aplikasi yang akan dibuat.

3.5.1 Rancangan Antarmuka (*User Interface*)

Penyajian program aplikasi dengan menggunakan tampilan antarmuka (*user interface*) selain membuat program aplikasi menjadi lebih menarik, juga dapat memudahkan pengguna dalam mengoperasikan program tersebut. Pada tahap ini akan dijelaskan gambaran tentang tampilan program aplikasi yang akan dibuat. Berikut adalah tampilan rancangan-rancangan untuk program sistem penyelesaian masalah *fractional knapsack* dengan menggunakan algoritma *greedy* dan algoritma simpleks direvisi (*primal*).

Rancangan pertama adalah membuat tampilan input untuk pembuatan tabel. Dalam rancangan ini data inputnya terdiri dari jumlah objek, dan kapasitas maksimum *knapsack* (satuan berat). Khusus untuk input jumlah barang dimaksudkan dalam menentukan jumlah baris pada tabel yang akan dibuat, sehingga program hanya akan menyediakan ukuran jumlah baris yang dibutuhkan sesuai dengan jumlah data yang tersedia. Hal ini dimaksudkan untuk mengurangi kesalahan ketika menginput data dan pemrosesan pada tahap selanjutnya.

No	Nama Barang	Berat	Nilai

Gambar 3.3 Rancangan tampilan input awal

Rancangan selanjutnya adalah membuat tampilan input untuk input data. Rancangan ini hampir sama seperti pada rancangan input untuk pembuatan tabel, hanya saja tabel telah tersedia sesuai dengan jumlah objek yang diinput. Terdapat juga tiga tombol baru yaitu tombol reset, tombol algoritma *greedy*, dan tombol algoritma simpleks.

The screenshot shows a window titled 'Fsack' with a menu bar containing 'Berkas' and 'Bantuan'. The main title is 'PROGRAM FRACTIONAL KNAPSACK'. On the left, there are two input fields: 'Jumlah Objek' and 'Kapasitas Maksimum', each followed by a text box. Below these are three buttons: 'Reset', 'Algoritma Greedy', and 'Algoritma Simpleks Direvisi'. On the right, there is a table with four columns: 'No', 'Nama Barang', 'Berat', and 'Nilai'. The table has 10 rows, with the first row containing headers and the remaining 9 rows being empty.

No	Nama Barang	Berat	Nilai

Gambar 3.4 Rancangan tampilan input data

Setelah proses input data, rancangan selanjutnya adalah tampilan hasil algoritma *greedy* dan algoritma simpleks direvisi (primal). Rancangan tampilan untuk masing-masing algoritma diperlihatkan pada **Gambar 3.5** dan **Gambar 3.6**.



Gambar 3.7 Rancangan tampilan data teks

3.5.2 Pembuatan Program

Sebagaimana yang telah disebutkan sebelumnya bahwa bahasa pemrograman yang dalam pembuatan aplikasi ini menggunakan Java. Dengan demikian, berkas kode program yang dibuat seluruhnya disimpan dengan ekstensi *.java. Setelah kode program dibuat, langkah selanjutnya adalah mengkompilasi berkas Java tersebut dan kemudian dihasilkan berkas dengan ekstensi *.class. Setelah itu seluruh berkas dengan ekstensi *.class dihimpun dalam sebuah berkas baru berekstensi *.jar. Dari berkas *.jar ini program aplikasi dapat langsung dijalankan asalkan pada komputer yang digunakan telah tersedia *Java Runtime Environment Update 6*.

Pada pembuatan program aplikasi ini terdapat 13 berkas kode Java. Sebelas diantaranya saling berhubungan satu sama lain. Untuk selanjutnya, berkas-berkas ini disebut sebagai kelas. Masing-masing kelas tersebut memiliki peran dan fungsi

tersendiri yang saling berhubungan antara satu kelas dan yang lainnya. Adapun penjelasan mengenai fungsi dari masing-masing kelas adalah sebagai berikut:

1. DATA_BARANG

Kelas ini adalah acuan ukuran tabel.

2. FRAC_KNAPSACK

Kelas ini adalah kelas utama untuk menjalankan program juga sebagai tampilan awal untuk menginput jumlah objek beserta kapasitas maksimum *knapsack* atau membuka berkas berisi data objek berekstensi *.fsack untuk dilakukan perhitungan ulang.

3. FRAC_KNAPSACK_ready

Kelas ini berfungsi menampilkan tabel kosong yang jumlah barisnya berdasarkan jumlah objek untuk menginput data objek secara manual ke dalam tabel.

4. FRAC_KNAPSACK_Random

Kelas ini berfungsi menampilkan data objek secara acak ke dalam tabel yang jumlah berdasarkan jumlah objek. Adapun data acak yang telah ditampilkan pada tabel dapat juga diedit atau dirubah sesuai kebutuhan.

5. FRAC_KNAPSACK_Load

Kelas ini berfungsi menampilkan kembali berkas berekstensi *.fsack. yang memuat data objek yang telah disimpan untuk dilakukan perhitungan ulang.

6. FORM_REPORT

Kelas ini berisi implementasi algoritma greedy dalam menyelesaikan masalah *fractional knapsack* sekaligus menampilkan hasilnya.

7. Simplex_FrackKNAPSACK

Kelas ini berisi implementasi algoritma simpleks direvisi (primal) dalam menyelesaikan masalah *fractional knapsack*.

8. Report_SIMPLEX

Kelas ini menampilkan hasil algoritma simpleks direvisi (primal) dalam menyelesaikan masalah *fractional knapsack*.

9. Laporan

Kelas ini berfungsi untuk menampilkan laporan berbasis teks dari data objek maupun hasil perhitungan kedua algoritma.

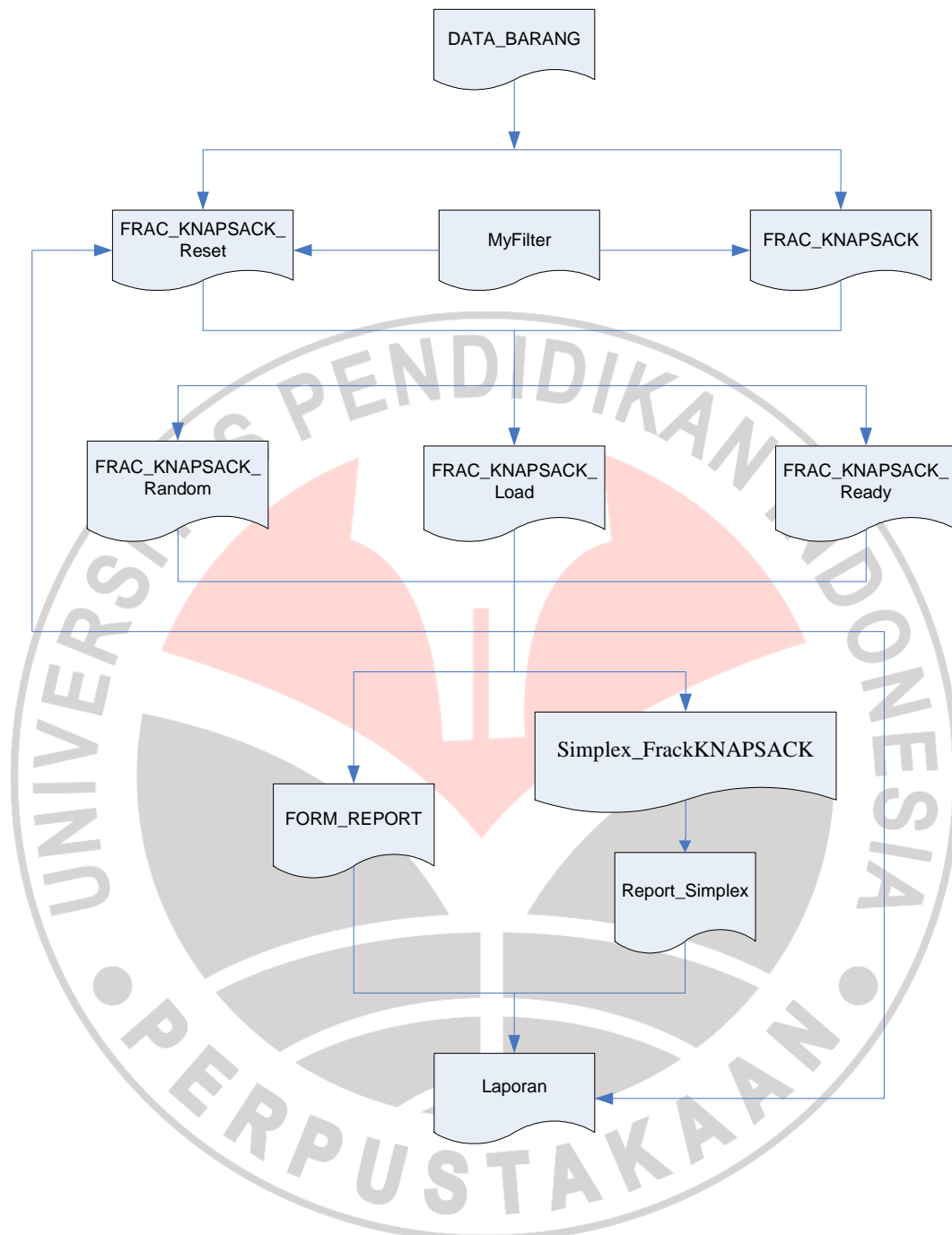
10. MyFilter

Kelas ini berfungsi menyaring tipe data pada saat membuka dan menyimpan data.

11. FRAC_KNAPSACK_Reset

Kelas ini berfungsi untuk kembali ke tampilan awal tanpa harus menutup program.

Adapun diagram alir hubungan antara satu kelas dengan kelas lainnya diperlihatkan pada **Gambar 3.8**.



Gambar 3.8 Diagram alir hubungan antar kelas