

## **Pemecahan Masalah Integer Programming Biner Dengan Metode Penambahan**

Wawan Laksito YS <sup>6)</sup>

### **Abstrak**

*Program Linier adalah perencanaan aktifitas-aktifitas untuk memperoleh suatu hasil yang optimal. Tidak semua variabel keputusan dalam Program Linier dapat berupa bilangan pecahan, Beberapa variabel Integer Programming dapat dinyatakan dalam bentuk nol-satu (biner). Penyelesaian masalah Integer Programming dimulai dengan menyelesaikan masalah Program Linier dengan metoda dual dan dilanjutkan dengan Metoda Penambahan dengan Perhitungan Implisit Nol-Satu (Biner)*

### **I. Pendahuluan**

Pengambilan Keputusan merupakan unsur utama dalam masalah perencanaan. Dalam setiap pengambilan keputusan, akan selalu berhadapan dengan lingkungan dimana salah satu karakteristik yang kurang menguntungkan adalah kenyataan bahwa lingkungan memiliki keterbatasan berkenaan dengan sumber-sumber yang tersedia. Untuk mengoptimalkan keterbatasan dari sumber-sumber yang tersedia diperlukan suatu cara untuk mengalokasikan sumber-sumber yang terbatas diantara beberapa aktifitas yang bersaing. Contoh situasi dari uraian diatas antara lain adalah persoalan pengalokasian fasilitas produksi, persoalan pengalokasian sumber daya nasional untuk kebutuhan domestik, penjadualan produksi, penyelesaian permainan, dan pemilihan pola pengiriman.

Salah satu cara untuk menyelesaikan persoalan diatas adalah Program Linier. Program Linier adalah perencanaan aktifitas-aktifitas untuk memperoleh suatu hasil yang optimal, yaitu suatu hasil yang mencapai tujuan terbaik diantara seluruh aktifitas yang fisibel. Dalam Program Linier semua aktifitas dinyatakan sebagai variabel keputusan yang berharga pecahan. Akan tetapi dalam kenyataannya tidak semua variabel keputusan dapat berupa bilangan pecahan. Misalnya variabel keputusan yang dihadapi berkaitan dengan jumlah menis yang diperlukan, maka

---

<sup>6)</sup> Staf Pengajar STMIK Sinar Nusantara Surakarta

jumlah mesin 10,5 sangat tidak realistis dalam konteks keputusan yang nyata. Variabel keputusan dapat juga menunjukkan suatu simbol dari pernyataan. Sebagai contoh kasus dimana persoalan yang dihadapi merupakan persoalan keputusan “Ya” atau “Tidak”. Misalnya keputusan untuk membuka pabrik baru, mengalokasikan modal, dan lain sebagainya. Dalam hal ini dapat dinyatakan dengan variabel biner  $x=0$  jika pernyataan “Tidak” dan  $x=1$  jika pernyataan “Ya”.

### **1.1 Perumusan Masalah.**

Dua masalah yang akan dibahas dalam makalah ini yaitu :

- a. Bagaimana Algoritma metoda Penambahan.
- b. Bagaimana penerapan Algorithma metoda Penambahan pada masalah Integer Programing Biner.

Pembatasan Analisa :

- Fungsi kendala dan fungsi tujuan sudah tersusun sebelumnya.

### **1.2 Tujuan**

Mencari penyelesaian optimal dari model Integer Programming yang berorientasi pemilihan salah satu aktifitas dari dua kondisi aktifitas.

### **1.3 Landasan Teori**

#### **1.4.1 Formulasi Persoalan Program Linier.**

Dalam membangun model matematika dari formulasi persoalan Program Linier digunakan karakteristik-karakteristik sebagai berikut :

- a. Variabel keputusan.

Variabel keputusan adalah variabel yang menguraikan secara lengkap keputusan-keputusan yang akan dibuat. Yang merupakan formulasi dari apa yang dicari dalam persoalan tersebut. Variabel keputusan ini ditulis dengan  $x_i$ ,  $i= 1,2,\dots,n$

- b. Fungsi Tujuan

Fungsi Tujuan merupakan fungsi-fungsi dari variabel keputusan yang harus dicapai agar penyelesaian optimal dapat ditentukan dari semua nilai-nilai fisibel.

- c. Fungsi Kendala.

Fungsi kendal merupakan formulasi dari kendala-kendala yang dihadapi dalam menentukan nilai-nilai variabel keputusan.

d. Pembatas Tanda

Pembatas tanda adalah pembatas yang menjelaskan apakah variabel keputusan hanya berharga nonnegatif atau boleh positif, nol, negative. (Dimiyati, 1992,2)

**1.4.2 Bentuk Standar Program Linier**

Dalam menyelesaikan masalah Program Linier, maka model matematika PL harus disajikan dalam bentuk Standar. Bentuk Standar Program Linier mempunyai ciri-ciri :

- Semua fungsi kendala berbentuk persamaan dengan ruas kanan nonnegatif.
- Semua variabel adalah nonnegatif.
- Mengoptimalkan (memaksimalkan atau meminimalkan fungsi tujuan).

Bentuk Standar Program Linier dapat dituliskan sebagai berikut :

$$\text{Mengoptimalkan. } Z = \sum_{j=1}^n c_j x_j$$

$$\begin{aligned} \text{Dengan kendala } \sum_{j=1}^n a_{ij} x_j &= b_i \quad ; i = 1, 2, \dots, m \\ x_j &\geq 0 \quad ; j = 1, 2, \dots, n \end{aligned}$$

**II. Pembahasan**

**2.1 Perhitungan Implisit Nol-Satu (Biner)**

Beberapa variabel Integer Programming dapat dinyatakan dalam bentuk nol-satu (biner). Misalnya  $0 \leq x \leq n$  adalah variabel bilangan bulat, dimana  $n$  adalah bilangan bulat batas atas.  $Y_1, Y_2, \dots, Y_n$  adalah variabel nol-satu, dan  $x = Y_1 + Y_2 + \dots + Y_n$  adalah sebuah biner yang mewakili semua harga  $x$  yang mungkin (fisibel). Pernyataan lain untuk variabel biner yang banyaknya kurang dari  $n$  adalah :

$x = Y_0 + 2Y_1 + 2^2 Y_2 + \dots + 2^k Y_k$  dengan  $k$  bilangan bulat terkecil yang memenuhi  $2^{k+1} - 1 \geq n$ . Hal inilah yang akan dipakai untuk mengembangkan algoritma yang efisien.

### Algoritma Penambahan

Pada algoritma ini dimulai dengan menggunakan fisibel dual, yaitu penyelesaian optimal, tetapi tidak fisibel. Selain itu semua fungsi kendala harus bertipe " $\leq$ ". Bentuk ini dapat selalu dicapai, sebagai contoh adalah sebagai berikut :

$$\text{Meminimalkan } Z = \sum_{j=1}^n c_j x_j$$

$$\text{Dengan kendala } \sum_{j=1}^n a_{ij} x_j + s_i = b_i$$

$$x_j = 0 \text{ atau } 1 \text{ dan } s_i \geq 0$$

$$\text{dengan } i = 1, 2, \dots, m$$

$$j = 1, 2, \dots, n$$

$s_i$  merupakan variabel Slack pada fungsi kendala ke- $i$ .

b Masalah diatas menjadi fisibel dual, bila setiap  $c_j \geq 0$ . Bila  $c_j < 0$ , maka dapat diubah dengan mensubstitusikan  $x_j = 1 - x_j$  ke dalam variabel  $x_j$ , dimana  $x_j$  adalah variabel biner, pada fungsi tujuan dan kendalanya.

Jika pada penambahan ini masalah primalnya fisibel, maka tidak ada lagi yang dikerjakan. Karena harganya akan menjadi minimum dengan memberikan harga nol untuk semua variabelnya. Sebaliknya bila primalnya tidak fisibel maka dipakai Algoritma Penambahan untuk mencari penyelesaian yang optimal.

Ide dari Algoritma ini adalah menghitung semua penyelesaian yang mungkin, yaitu  $2^n$  kemungkinan. Tetapi beberapa penyelesaian dapat diabaikan tanpa harus diselidiki.

Pada masalah biner ini pertama-tama diasumsikan bahwa semua variabel berharga nol. Hal ini bisa diterima secara logis karena semua  $c_j \geq 0$ . Jika penyelesaiannya tidak fisibel (beberapa variabel slack  $s_i$  mungkin negatif), maka perlu untuk menaikkan beberapa harga variabel menjadi berharga satu. Proses ini jelas akan membawa penyelesaian ke arah fisibel, yaitu membuat  $s_i \geq 0$  untuk semua  $i$ .

Pencabangan biner dapat digunakan untuk manipulasi data dengan cara yang sederhana. Untuk melaksanakannya perlu diperhatikan definisi-definisi dibawah ini.

a. Variabel Bebas.  
 Pada beberapa titik cabang, sebuah variabel biner dikatakan bebas, jika tidak ditentukan oleh beberapa pencabangan ke titik cabang tersebut.

b. Penyelesaian Parsial  
 Penyelesaian Parsial akan menunjukkan variabel mana yang akan berharga satu atau nol. Untuk mempermudah penganalisaan penyelesaian parsial ini dibentuk sebagai order set. Misalnya  $j_t$  merupakan penyelesaian parsial pada titik cabang ke-t, dan notasi  $+j$  adalah notasi dari  $x_j = 1$ , notasi  $-j$  adalah notasi dari  $x_j=0$ . Maka elemen  $j_t$  adalah subscript dari variabel tertentu, dan tanda (-) atau (+) menunjukkan bahwa variabel tersebut berharga nol atau satu, dan set  $j_t$  harus diurutkan.

Penyelesaian Parsial tidak berlanjut bila :

- tidak dapat menghasilkan nilai fungsi tujuan yang lebih baik.
- Tidak dapat menghasilkan penyelesaian yang fisibel.

Pada penyelesaian parsial yang tidak berlanjut, tidak terdapat pencabangan.

Rincian algoritma penambahan adalah sebagai berikut :

Misalkan  $j_t$  adalah penyelesaian parsial pada titik cabang (t) (awal  $j_0=\emptyset$ ) dan diasumsikan  $Z^t$  adalah himpunan nilai-nilai  $Z$ , dan  $\underline{Z}$  adalah batas atas terbaik ( $\underline{Z}=\infty$ ).

Test 1 : Untuk variabel bebas  $x_k$ , jika  $a_{ik} \geq 0$  untuk semua  $i$ ,  $s_i^t < 0$  , maka  $x_k$  tidak dapat memperbaiki kefisibelan, sehingga harus dikeluarkan.

Test 2 : Untuk variabel bebas  $x_k$ , jika  $c_k + Z^t \geq \underline{Z}$ , maka  $x_k$  tidak menghasilkan penyelesaian yang lebih baik, sehingga harus dikeluarkan.

Test 3 : Ditinjau kendala ke-I

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n + s_i = b_i \text{ dimana } s_i^t < 0$$

Misalkan  $N_j$  adalah himpunan dari variabel bebas yang tidak dikeluarkan oleh test 2, maka tidak ada satupun variabel bebas dalam  $N_j$  yang menghasilkan penyelesaian yang fisibel, jika untuk paling tidak satu  $s_i^t$

$< 0$ , dipenuhi :  $\sum_{j \in N_t} \min(0, a_{ij}) < s_i^t$ . Dalam kasus ini  $j_t$  tidak berlanjut.

Test 4: Jika  $N_t \neq \emptyset$ , maka cabang variabel  $x_k$  dipilih yang memenuhi  $V_k^t = \max_{j \in N_t} (V_j^t)$  dengan

$$V_j = \sum_{i=1}^t \min\{0, s_i^t - a_{ij}\}$$

Jika  $V_k^t = 0$ ,  $x_k=1$  dan  $j_t$  menghasilkan penyelesaian fisibel yang sudah diperbaiki, dalam hal ini  $j_{t+1}$  dengan  $\{k\}$  ditambahkan di sebelah kanan, maka tidak berlanjut.

Test selanjutnya dilakukan pada  $j_{t+1}$  sampai perhitungannya selesai, yaitu semua elemen pada bagian penyelesaiannya adalah negatif.

### III. Kesimpulan

Pada masalah-masalah variabel keputusan yang berupa bilangan 0 dan 1, algoritma yang dikembangkan adalah Algoritma Penambahan. Algoritma Penambahan ini menyelidiki sebagian saja variabel keputusan. Pada Algoritma ini dimulai dengan menggunakan dual fisibel, yaitu penyelesaian optimal tetapi tidak fisibel dan pada keadaan awal semua variabel keputusan dianggap bernilai nol, kemudian secara iteratif diselidiki variabel-variabel keputusan yang dapat diberikan harga 1 yang akan memperbaiki kefisibelan.

### Daftar Pustaka

1. Bazaraa, Mohtar S, Jacob, John S, Sherali, Hanif D, **Linear Programming & Network Flows**, 2<sup>nd</sup> Ed, John Wiley & Sons, Singapore, 1990.
2. Dimiyati, Tjutju Tarlih, Dimiyati, Ahmad, **Operation Researc : Model-model pengambilan keputusan**, edisi 2, Sinar Baru, Bandung, 1987
3. Taha, Hamdy A., **Operatios Research : An Introduction**, 4<sup>rd</sup> ed, Macmillan Publishing Co. Inc, New York, 1992
4. Jean-Sébastien, Roy, "**Binarize and Project**" to generate cuts for **general mixed-integer programs**, Algorithmic Operations Research, Volume 2, Number 1 2007 <http://journals.hil.unb.ca>
5. Jensen ,Poul Q, **Operation research Model & Metode**, Mathematical techniques of operation research <http://www.londonexternal.ac.uk> (akses 2007)