# Bab 2
# Landasan Teori

## 2.1. *Linear Programming*

Model pemrograman linear tidak mampu menyelesaikan kasus-kasus manajemen yang menghendaki sasaran-sasaran tertentu dicapai secara simultan. Kelemahan ini dilihat oleh A. *Charnes* dan W.M. *Cooper*. Mereka berdua kemudian mengembangkan model pemrograman linear agar mampu menyelesaikan kasus-kasus tersebut. Dalam hal ini, konsep dasar model yang mereka temukan itu sudah mulai diperkenalkan pada tahun 1955. Selanjutnya, pada tahun 1961, mereka mulai mempopulerkan model tersebut dengan nama *Goal programming*. Model ini mampu menyelesaikan kasus-kasus pemrograman linear yang memiliki lebih dari satu sasaran yang hendak dicapai.

Model *Goal programming* merupakan perluasan dari model pemrograman linear, sehingga seluruh asumsi, notasi, formulasi model matematis, prosedur perumusan model dan penyelesaiannya tidak berbeda. Perbedaan hanya terletak pada kehadiran sepasang variabel deviasional yang akan muncul di fungsi tujuan dan di fungsi-fungsi kendala. Oleh karena itu, konsep dasar pemrograman linear akan selalu melandasi pembahasan model *goal programming*.

## 2.2. *Goal programming*

Dasar dari pendekatan *goal programming* adalah untuk menentukan / menetapkan hasil perhitungan angka yang spesifik untuk setiap objek, formulasi dan fungsi objek untuk setiap objek lalu menentukan solusi untuk meminimasi jumlah deviasi fungsi objek dari perkiraan hasil yang ingin dicapai.

*Goal programming* adalah salah satu model matemetis yang dipakai sebagai dasar dalam mengambil keputusan untuk menganalisis dan membuat solusi persoalan yang melibatkan banyak tujuan sehingga diperoleh *alternative* pemecahan masalah yang optimal.

Beberapa asumsi dasar yang diperlukan dalam *goal programming* adalah:

1. Linieritas asumsi ini menunjukkan perbandingan antara input yang satu dengan input yang lain atau untuk suatu input dengan output besarnya tetap dan terlepas pada tingkat produksi. Hubungannya bersifat linier.
2. Proporsionalitas asumsi ini menyatakan bahwa jika peubah pengambilan keputusan berubah, maka dampak perubahannya akan menyebar dalam proporsi yang sebanding dengan fungsi tujuan dan juga fungsi kendalanya. Jadi tidak berlaku hukum kenaikan hasil yang semakin berkurang.
3. Aditivitas asumsi ini menyatakan nilai parameter suatu kriteria optimasi merupakan jumlah dari nilai individu-individu. Dampak total terhadap kendala ke-I merupakan jumlah dampak individu terhadap peubah pengambilan keputusan
4. Divisibilitas asumsi ini menyatakan bahwa peubah pengambilan keputusan, jika diperlukan dapat dibagi ke dalam pecahan-pecahan.
5. Deterministik asumsi ini menghendaki agar semua parameter tetap dan diketahui atau ditentukan secara pasti.

### 2.2.1. Istilah-istilah *Goal programming*

Ada beberapa istilah yang digunakan pada *goal programming*, antara lain sebagai berikut:

1. Varabel keputusan (*decision variables*), merupakan nilai-nilai yang tidak diketahui yang berada di bawah control pengambilan keputusan, yang berpengaruh terhadap solusi permasalahan dan keputusan yang diambil. Biasanya dilambangkan dengan $X_j$ (j = 1,2,3,….n) .
2. Nilai sisi kanan (*right hand sides value*), merupakan nilai-nilai yang biasanya menunjukkan ketersediaan sumber daya (dilambangkan dengan $b_i$) yang akan ditentukan kekurangan atau kelebihan penggunaannya.
3. Koefisien teknologi (*technology coefficient*), merupakan nilai-nilai numerik yang dilambangkan dengan *aij* yang akan dikombinasikan dengan variable keputusan, dimana akan menunjukan penggunaan terhadap pemenuhan nilai kanan.

4. Fungsi tujuan adalah fungsi matematis dari variable-variable keputusan yang menunjukan hubungan dengan nilai sisi kanan.
5. Fungsi pencapaian adalah fungsi matematis dari variable-variable simpang yang menyatakan kombinasi sebuah objektif.
6. Fungsi tujuan mutlak merupakan tujuan yang tidak boleh dilanggar dengan pengertian mempunyai penyimpangan positif dan negatif bernilai 0. Proritas pencapaian dari fungsi tujuan ini berada pada urutan pertama, solusi yang dapat dihasilkan adalah terpenuhi atau tidak terpenuhi.
7. Variable simpanganadalah variable yang menunjukan kemungkinan penyimpanan-penyimpanan negatifdan positif dari nila sisi kanan fungsi tujuan. Dalam model *goal programming* sistem urutan tersebut menempatkan tujuan-tujuan tersebut dalam hubungan susunan seri. Hubungan tersebut dapat dilambangkan sebagai berikut: P1>P2>…..>>>Pk.
8. Pembobotan merupakan timbangan matematis yang dinyatakan dengan angka ordinal, dilambangkan dengan $W_{ki}$, dimana k = 1,2,…, n serta i = 1,2,…, n dan yang digunakan untuk membedakan variable simpangan i dalam suatu tingkat priositas k.

### 2.3. Konsep Dasar *Goal programming*

Di dalam model pemrograman linear kita mengenal variabel *slack* yang terdapat pada fungsi kendala yang berupa pembatas, dan variabel *surplus* pada fungsi kendala yang berupa syarat. Kehadiran kedua variabel itu dalam penyelesaian suatu kasus pemrograman linear adalah untuk menampung kelebihan atau kekurangan nilai ruas kiri suatu fungsi kendala agar sama dengan nilai ruas kanannya.

Nilai variabel *slack* atau *surplus* sangat tergantung kepada hasil penyelesaian optimal. Dalam penyelesaian Sukra Rasmi yang ditayangkan pada gambar 1.1, variabel *slack* $S_1$= 0, $S_2$= 0, $S_3$= 8, dan $S_4$= 4 karena $X_1 = 7$ dan $X_2$ = 6 agar nilai fungsi tujuan maksimum. Dalam kasus tersebut kita tidak mungkin mengendalikan $S_i$ agar minimum karena nilai tersebut diturunkan oleh penemuan variabel keputusan $X_i$. yang akan membuat nilai fungsi tujuan maksimum. Dengan

demikian, jelas bahwa nilai variabel *slack* atau *surplus* sama sekali tidak bisa dikendalikan di dalam penyelesaian sebuah kasus pemrograman linear.

```
MAX     40 X1 + 30 X2
SUBJECT TO
  2)  2 X1 + X2 <=     20
  3)  2 X1 + 3 X2 <=     32
  4)  2 X1 - X2 >=     0
  5)  X2 >=     2

END

  OBJECTIVE FUNCTION VALUE

  1)  460.000000

 VARIABLE                VALUE      REDUCED COST
       X1              7.000000          .000000
       X2              6.000000          .000000

      ROW    SLACK OR SURPLUS       DUAL PRICES
        2)              .000000        15.000000
        3)              .000000         5.000000
        4)             8.000000          .000000
        5)             4.000000          .000000

NO. ITERATIONS=  4
```

Gambar 2.1. Penyelesaian kasus Sukra Rasmi.

Gagasan dasar Charnes dan Cooper, pada dasarnya, berpijak pada konsep variabel *slack* dan *surplus.* Bila ada variabel-variabel di dalam model pemrograman linear yang mempunyai karakteristik mirip dengan kedua variabel tersebut, dan berada di dalam suatu persamaan kendala, maka pengendalian terhadap variabel tersebut di dalam fungsi tujuan berarti pengendalian terhadap nilai ruas kiri persamaan kendala tersebut. Jadi, kita bisa mengendalikan nilai ruas kiri suatu kendala agar sama dengan nilai ruas kanannya dengan cara mengendalikan variabel tersebut. Inilah pijaran awal bagi Charnes dan Cooper untuk mengembangkan model *Goal programming.*

*Goal programming* biasanya diterapkan pada masalah-masalah linear dengan memasukkan berbagai tujuan dalam formulasi modelnya. Setiap tujuan dinyatakan sebagai suatu *goal* dan direpresentasikan secara numerik. *Goal* yang dinyatakan secara numerik inilah yang dicoba untuk dicapai. Akan tetapi, berbagai *goal* tidak selalu dapat dicapai secara bersamaan karena dapat terjadi penyimpangan (deviasi) dari *goal*. Oleh karena itu, dalam formulasi *goal programming*, *goal*

dalam numerik untuk setiap tujuan harus ditetapkan lebih dahulu. Kemudian, solusi yang ingin dicari adalah meminimalkan jumlah penyimpangan tujuan-tujuan ini terhadap masing-masing *goal*nya. Dengan kata lain, fungsi tujuan dalam *goal programming* dinyatakan sebagai minimisasi penyimpangan dari fungsi preferensi atau fungsi pencapaian *goal*.

Untuk menyatakan penyimpangan (deviasi) dalam formulasi modelnya diperlukan suatu variabel yang disebut variabel deviasi. Variabel deviasi ini menyatakan tingkat pencapaian *goal* dalam pengertian kurang tercapai (*underachievement of goal*) dan melebihi *goal* (*overachievement of goal*). Tingkat *underachievement* dan *overachievement of goal* ini tidak mungkin terjadi bersamaan. Oleh karena itu, salah satu atau kedua variabel deviasi ini akan sama dengan nol.

Karena dalam formulasi modelnya setiap *goal* dimasukkan dalam kendala, maka kendala *goal programming* disebut *goal constraints*. *Goal constraint* ini pun ditulis dalam variabel deviasi. Selain variabel deviasi, diperlukan pula variabel keputusan. Variabel keputusan tidak ditampakkan dalam fungsi tujuan.

Dalam suatu situasi tertentu, penyimpangan dari suatu *goal* mungkin dianggap lebih penting dari penyimpangan *goal* lainnya. Atau mungkin juga timbul suatu situasi di mana penyimpangan *overachievement* dianggap lebih penting dari penyimpangan *underachievement*, dan sebaliknya. Dalam situasi-situasi seperti ini, diperlukan suatu timbangan di mana timbangan-timbangan ini mencerminkan relatif pentingnya berbagai penyimpangan dari *goal* tersebut.

Penentuan timbangan yang mencerminkan relatif pentingnya berbagai penyimpangan ini, mungkin tampak bersifat subjektif atau sembarangan. Untuk mengatasi masalah ini dalam kasus-kasus tertentu, relatif pentingnya *goal* tersebut dinyatakan dalam prioritas *absolut*. Dalam kerangka ini, pencapaian suatu set *goal* pada suatu tingkat prioritas tertentu lebih dunginkan dari pada pencapaian suatu set *goal* lain dengan prioritas yang lebih rendah.

Konsep tentang berbagai pendekatan dalam formulasi *goal programming* di atas dapat diringkas sebagai berikut:

1. Setiap *goal constraint* selain mempunyai variabel keputusan, yaitu $x_i$ ($x1$,$x2$, ... $x.$) juga mempunyai variabel deviasi yang nonnegatif, yaitu $u_i$ dan $e_i$ Variabel $u_i$ menyatakan penyimpangan yang *underachievement* dan variabel $e_i$, menyatakan penyimpangan yang *overachievement.*
2. Fungsi tujuan dalam *goal programming* adalah minimisasi penyimpangan atau minimisasi variabel deviasi $u_i$ atau $e_i$. Variabel keputusan $x_i$ tidak dinyatakan dalam fungsi tujuan.
3. Untuk meminimalkan penyimpangan *underachievement* $u_i$ formulasi *goal constraint* adalah:

   $$\text{gi}(x1, x2, \ldots x\,n + u_i \leq b_i;\ u_i \geq 0$$

   *dengan fungsi tujuan meminimalkan* $u_i$.
4. Untuk meminimalkan penyimpangan *overachievement* $e_i$, formulasi *goal constraintnya* adalah:

   $$g_i(x1, x2, \ldots x\,n) - e, \geq b_i;\ e_i \geq 0$$

   $$\leq$$

   dengan fungsi tujuan minimisasi *ei*.
5. Untuk meminimalkan penyimpangan *underachievement* dan *overachievement*, formulasi *goal constraintnya* adalah:

   $$\text{g}_\text{i}(\text{x}1, \text{x}2, \ldots \text{x n}) + \text{u}_\text{i} - e_i = b_i.$$

   dengan fungsi tujuan minirnisasi $u_i$ + *ei*.
6. Untuk menyatakan preferensi atas suatu penyimpangan, dipergunakan faktor timbangan dalam formulasi fungsi tujuannya. Fungsi tujuan tertimbang dalam model *goal programming* dituliskan sebagai:

   $$\text{Minimisasi } \sum_{i=1}^{n} w_i, u_i + w_i + e_i$$

7. Karena salah satu atau kedua variabel deviasi $\text{u}_\text{i}$ dan $e_i$ sama dengan 0 (nol), maka *goal constraint* yang mempunyai variabel deviasi bermlai positif merupakan kendala aktif.

Dengan konsep *goal programming* ini, dibahas berbagai tipe model *goal programming* yang paling berguna.

**2.4. Kendala-Kendala Sasaran**

Di dalam *goal programming*, *Charnes* dan *Cooper* menghadirkan sepasang variable yang dinamakan "*variable deviasional*" dan berfungsi untuk menampung penyimpangan atau deviasi yang akan terjadi pada nilai ruas kiri suatu persamaan kendala terhadap nilai ruas kanannya. Agar deviasi itu minimum, artinya nilai ruas kiri suatu persamaan kendala "sebisa mungkin" mendekati nilai ruas kanannya maka variable deviasional itu harus diminimumkan di dalam fungsi tujuan.

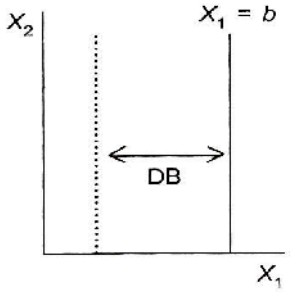
Pemanipulasian model pemrograman linier yang dilakukan oleh *Charner* dan *Cooper* telah mengubah makna kendala fungsional. Bila pada model pemrograman linier, kendala-kendala fungsional menjadi pembatas bagi usaha pemaksimuman atau peminimuman fungsi tujuan, maka pada model *Goal programming* kendala-kendala itu merupakan sara untuk mewujudkan sasaran yang hendak dicapai. Sasaran-sasaran, dalam hal ini dinyatakan sebagai nilai konstan pada ruas kanan kendala.

Sebagai contohnya sasaran laba, anggaran yang tersedia, resiko investasi, ketersediaan bahan baku, ketersediaan jam kerja, kapasitas produksi dan lain-lain. Mewujudkan suatu sasaran, dengan demikian berarti mengusahakan agar nilai ruas kiri suatu persamaan kendala sama dengan nilai ruas kanannya. Itulah sebabnya kendala-kendala di dalam model *goal programming* selalu berupa persamaan dan dinamakan "kendala sasaran". Disamping itu, keberadaan sebuah kendala ditandai dengan kehadiran *variable deviasional* sehingga setiap kendala sasaran pasti memiliki *variable deviasional*.

- Variabel Deviasional

  Variabel deviasional, sesuai dengan fungsinya, yaitu menampung deviasi hasil terhadap sasaran-sasaran yang dikehendaki, dibedakan menjadi dua yaitu:

1. Variabel deviasional untuk menampung deviasi yang berada di bawah sasaran yang dikehendaki. Sasaran itu tercermin pada nilai ruas kanan suatu kendala sasaran. Dengan kata lain, variabel deviasional ini berfungsi untuk menampung deviasi negatif. Kita menggunakan notasi DB untuk menandai jenis variabel deviasional ini. Karena variabel deviasional DB berfungsi untuk menampung deviasi negatif maka,




$$\sum_{j=1}^{n} a_{ij} \,.\, \mathrm{X}_{ij} = b_i - \mathrm{DB}_i$$

atau

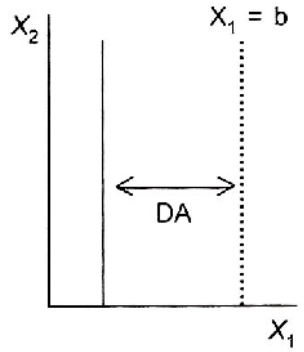
$$\sum_{j=1}^{n} a_{ij} \,.\, \mathrm{X}_{ij} + \mathrm{DB}_i = b_i$$

Di mana, $i = 1, 2, \ldots, m$

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

sehingga DB akan selalu mempunyai koefisien +1 pada setiap kendala sasaran.

2. Variabel deviasional untuk menampung deviasi yang berada di atas sasaran. Dengan kata lain, variabel deviasional ini berfungsi untuk menampung deviasi positif. Notasi DA digunakan untuk menandai jenis variabel deviasional ini. Karena variabel deviasional DA berfungsi untuk menampung deviasi positif maka,




$$\sum_{j=1}^{n} a_{ij} \,.\, \mathrm{X}_{ij} = b_i + \mathrm{DA}_i$$

atau

$$\sum_{j=1}^{n} a_{ij} \,.\, \mathrm{X}_{ij} - \mathrm{DA}_i = b_i$$

Di mana, $i = 1, 2, \ldots, m$

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

sehingga DA akan selalu mempunyai koefisien –1 pada setiap kendala sasaran.

Dengan demikian, jelas bahwa kedua variabel deviasional tersebut mempunyai fungsi yang berbeda. Bila variabel deviasional DB menampung penyimpangan nillai di bawah sasaran maka variabel deviasional DA menampung penyimpangan mlai di atas sasaran. Sehingga sebenarnya cukup mudah untuk dimengerti bahwa nilai penyimpangan minimum di bawah maupun di atas sasaran adalah nol dan tidak mungkin negatif atau,

$$DB_I \geq 0 \text{ untuk } i = 1, 2, \ldots\ldots\ldots\ldots m.$$
$$DA_i \geq 0 \text{ untuk } i = 1, 2, \ldots\ldots\ldots\ldots m.$$

Untuk membuktikan kebenaran kedua rumus di atas, marilah kita membayangkan kendala sasaran di mana penyimpangan di bawah dan di atas sasaran tidak diperkenankan. Dengan kata lain, *sasaran* itu harus tercapai. Secara matematis, bentuk umum kendala sasaran itu adalah:

$$\sum_{j=1}^{n} a_{ij} \,.\, X_{ij} = b_i + DA_i + DB_i$$

atau

$$\sum_{j=1}^{n} a_{ij} \,.\, X_{ij} - DA_i + DB_i = b_i$$

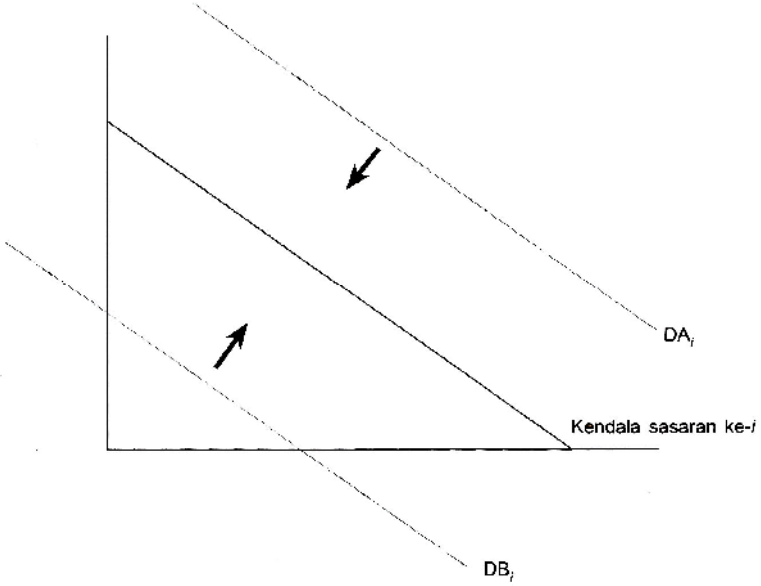
❖ **Fungsi Tujuan**

Ciri khan lain yang menandai model *Goal programming* adalah kehadiran variabel deviasional di dalam fungsi tujuan yang harus diminimumkan. Hal ini merupakan konsekuensi logis dari tujuan kehadiran variabel deviasional di dalam fungsi kendala sasaran.

Dari [9-8] kita mengetahui bahwa sasaran yang telah ditetapkan *(b)* akan tercapai bila variabel deviasional $DA_i$ dan $DB_i$ bernilai nol. Oleh karena itu, $DA_i$ dan $DB_i$ harus & minimumkan di dalam fungsi tujuan sehingga fungsi tujuan model *Goal*

$$\text{Minimumkan} = \sum_{i=1}^{m} \text{DB}_i + \text{DA}_i$$

*programming* adalah




Gambar 2.2. Variabet deviasional dan kendala sasaran.

### 2.4.1. Empat Macam Kendala Sasaran

Beberapa cara penggunaan variabel deviasional untuk mewujudkan sasaran-sasaran manajerial. Pada dasarnya, penggunaan tersebut dapat clikelompokkan ke dalam empat macam cara, yaitu:

1. Untuk mewujudkan suatu sasaran dengan nilai tertentu.

   Sasaran yang dikehendaki dituangkan ke dalam paramater $b_i$ atau lebih populer dengan istilah *nilai ruas kanan kendala.* Agar sasaran ini tercapai, maka penyimpangan di bawah dan di atas nilai $b_i$ harus diminimumkan.

   $$\sum_{j=1}^{n} a_{ij} \,.\, \text{X}_{ij} + \text{DB}_i - \text{DA}_i = b_i$$

   Maka fungsi tujuan menjadi :

   $$\text{Minimumkan} = \sum_{i=1}^{m} \text{DB}_i + \text{DA}_i$$

Di dalam penyelesaian optimal, bila $DA_i > 0$ maka $DB_i = 0$; dan bila $DA_i = 0$ maka $DB_i > 0$. Bila $DA_i > 0$ maka terjadi penyimpangan di atas nilai $b_i$ dan ini berarti sasaran terlampaui dan kebalikannya bila $DB_i > 0$, maka terjadi penyimpangan di bawah nilai $b_i$ dan dikatakan bahwa sasaran tidak tercapai.

2. Untuk mewujudkan suatu sasaran di bawah nilai tertentu.

   Sasaran yang hendak dicapai dituangkan ke dalam $b_i$ dan tidak boleh dilampaui. Oleh karena itu, penyimpangan di atas nilai $b_i$ harus diminimumkan agar hasil penyelesaian tidak melebihi nilai $b_i$ atau paling banyak sebesar $b_i$.

$$\sum_{j=1}^{n} a_{ij} \,.\, \mathrm{X}_{ij} - \mathrm{DA}_i = b_i$$

   Maka fungsi tujuan menjadi :

$$\text{Minimumkan} = \sum_{i=1}^{m} \mathrm{DA}_i$$

   Di dalam penyelesaian optimal, bila $DA_i = 0$ maka dikatakan bahwa sasaran tercapai, akan tetapi bila $DA_i > 0$ maka terjadi penyimpangan di atas $b_i$ dan hal ini menunjukkan bahwa sasaran yang dikehendaki telah terlampaui.

3. Untuk *mewujudkan* suatu sasaran di atas nilai tertentu.

   merupakan kebalikan dari butir 2. Di sini, penyimpangan di bawah nilai $b_i$ harus diminimumkan agar hasil penyelesaian paling sedikit sama dengan $b_i$.

$$\sum_{j=1}^{n} a_{ij} \,.\, \mathrm{X}_{ij} + \mathrm{DB}_i = b_i$$

   Maka fungsi tujuan menjadi :

$$\text{Minimumkan} = \sum_{i=1}^{m} \mathrm{DA}_i$$

   Di dalam penyelesaian optimal $DB_i$ mungkin bernilai nol, artinya sasaran tercapai namun mungkin juga bernilai positif, artinya sasaran yang dikehendaki tidak tercapai.

4. Untuk mewujudkan suatu sasaran yang pada interval nilai tertentu.

   Bila interval itu clibatasi oleh $a_i$ dan $\boldsymbol{b}_i$ maka hasil penyelesaian yang diharapkan akan berada di antara interval tersebut atau,

$$a_i \le \sum_{j=1}^{n} a_{ij} . X_{ij} \le b_i$$

.

   Hasil penyelesaian akan menyimpang di bawah nilai $a_i$ atau juga tidak di atas nilai $b_i$. Kemungkinan penyimpangan-penyimpangan itu harus diminimumkan. Oleh karena itu, kita perlu menghadirkan $DB_i$ guna membatasi penyimpangan di bawah $a_i$ dan juga $DA_i$ guna membatasi penyimpangan di atas *bi*.

$$a_i - \mathrm{DB}_i \le \sum_{j=1}^{n} a_{ij} . X_{ij} \le b_i + \mathrm{DA}_i$$

   Dalam hal ini setara dengan :

$$\sum_{j=1}^{n} a_{ij} . \mathrm{X}_{ij} + \mathrm{DB}_i \ge a_i$$

dan

$$\sum_{j=1}^{n} a_{ij} . \mathrm{X}_{ij} - \mathrm{DA}_i \le b_i$$

   Maka fungsi tujuan menjadi :

$$\text{Minimumkan} = \sum_{i=1}^{m} \mathrm{DB}_i + \mathrm{DA}_i$$

Pertidaksamaan $\sum_{j=1}^{n} a_{ij} . \mathrm{X}_{ij} + \mathrm{DB}_i \ge a_i$ dan $\sum_{j=1}^{n} a_{ij} . \mathrm{X}_{ij} - \mathrm{DA}_i \le b_i$ adalah fungsi kendala sasaran di mana sasaran itu berada pada interval antara $a_i$ dan $b_i$. Agar peranan kendala sasaran dan variable deviasional itu menjadi semakin jelas, kita bisa saja mengubah kedua bentuk fungsi pertidaksamaan tersebut menjadi fungsi-fungsi persamaan dengan cara menambahkan variabel baru yaitu $SDA_j$ dan $SDB_i$ yang berfungsi sebagai variabel *slack* dan *surplus,* yaitu:

$$\sum_{j=1}^{n} a_{ij} . \mathrm{X}_{ij} + \mathrm{DB}_i - \mathrm{SDB}_i \ge a_i$$

$$\sum_{j=1}^{n} a_{ij} \,.\, \mathrm{X}_{ij} - \mathrm{DA}_i - \mathrm{SDA}_i \geq b_i$$

Variabel $SDA_i$ dan $SDB_i$ di atas bukan variabel deviasional dan kehadirannya tidak diperhitungkan di dalam fungsi tujuan. Oleh karena itu, fungsinya benar-benar seperti variabel *slack* dan *surplus* di mana nilainya sangat tergantung kepada hasil penyelesaian optimal. Dengan demikian, peminimuman $DA_i$ dan $DB_i$ akan menggiring penyelesaian optimal berada di antara interval $a_i$ dan $b_i$.

### 2.4.1.1. Bentuk Umum Model *Goal programming*

Bentuk umum model matematis *Goal programming* dapat dirumuskan sebagai berikut :

$$\mathrm{Min} = \sum_{i=1}^{m} \mathrm{DB}_i + \mathrm{DA}_i$$

ST

$$\begin{aligned}
&a_{11}X_1 + a_{12}X_2 + \ldots\ldots\ldots\ldots + a_{1n}X_n + \mathrm{DB}_1 - \mathrm{DA}_1 = b_1 \\
&a_{21}X_1 + a_{22}X_2 + \ldots\ldots\ldots\ldots + a_{mn}X_n + \mathrm{DB}_m - \mathrm{DA}_m = b_m \\
&\quad\vdots \qquad\quad \vdots \qquad\qquad\qquad\quad \vdots \qquad \vdots \qquad \vdots \qquad \vdots \\
&a_{m1}X_1 + a_{m2}X_2 + \ldots\ldots\ldots\ldots + a_{mn}X_n + \mathrm{DB}_m - \mathrm{DA}_m = b_m
\end{aligned}$$

dan

$X_j$, $DA_i$, dan $DB_i \geq 0$, untuk $i = 1,2, \ldots\ldots, m$

## 2.5. Perumusan Masalah *Goal programming*

Perumusan permasalahan *goal programming* hampir sama dengan perumusan *linear programming.* Perbedaannya adalah dalam penentuan fungsi tujuan, yang digunakan pada *linear programming* ada variabel simpangannya, sementara pada *goal programming* adalah variabel keputusannya. Berikut ini beberapa langkah dalam perumusan masalah *goal programming.*

1. Penentuan variabel keputusan, merupakan dasar dalam pembuatan model keputusan untuk mendapatkan solusi yang dicari. Makin tepat penentuan variabel keputusan akan mempermudah pengambilan keputusan yang dicari.

2. Penentuan fungsi tujuan. Langkah-langkah yang dilakukan dalam memformulasikan fungsi tujuan adalah sebagai berikut.
   a. Setiap fungis tujuan hares dinyatakan sebagai fungsi darl variabel keputusan yang disimbolkan dengan fi (xi), yaitu fungsi darl variabel keputusan yang berhubungan dengan tujuan ke i, sedangkan x adalah vektor variabel keputusan yang disimbolkan dengan $a_{ij}x_j$, di mana $a_{ij}$ merupakan konstanta koefisien teknologi.
   b. Setiap fungsi tujuan memiliki nilai yang berhubungan dengan nilai sisi kanan ($b_i$) yang merupakan target atau tujuan darl fungsi tujuan tersebut. Ada 3 macam kemungkinan hubungan tersebut, yaitu $f_i(x_i) = b_i$, $f_i(x_i) \geq b_i$ dan atau $f_i(x_i) \leq b_i$.
3. Perumusan fungsi sasaran. Pada langkah ini tiap tujuan pada sisi kirinya ditambahkan dengan variabel simpangan, baik simpangan positif maupun simpangan negatif. Dengan ditambahkannya variabel simpangan, maka bentuk darl fungsi sasaran menjadi $f_i(x,) + d_i^- - d_i^+ = b_i$.
4. Penentuan prioritas utama. Pada langkah ini dibuat urutan darl tujuan–tujuan. Penentuan tujuan ini tergantung pada hal-hal berikut.
   - Keinginan darl pengambil keputusan.
   - Keterbatasan sumber-sumber yang ada.
   - Batasan-batasan yang lain yang secara eksplisit atau pun implisit menentukan dalam pemilihan variabel keputusan.
5. Penentuan pembobotan. Pada tahap ini merupakan kunci dalam menentukan urutan dalam suatu tujuan dibandingkan dengan tujuan yang lain.
6. Penentuan fungsi pencapaian *(achievement function).* Di sini kuncinya adalah memilih variabel simpangan yang benar untuk dimasukkan dalam fungsi pencapaian dan kemudian ditambahkan prioritas dan bobot yang diperlukan. Langkah pertama yang dilakukan adalah fungsi linear variabel simpangan. Selanjutnya dalam memformulasikan fungsi pencapaian adalah menggabungkan Setiap tujuan yang berbentuk minimasi variabel simpangan sesuai dengan prioritasnya. Dengan demikian. persamaan maternatis dapat ditulis sebagai berikut.

7. Minimasi a = { $P_1$ ($g_i$, $d_i$-, $d_i$+), $P_2$ ($d_2^-$, $d_2^+$), …, $P_k$ ($g_k$ ($d_k^-$ , $d_k^+$))}

   Minimasi yang dilakukan tergantung pada pertimbangan nilai sisi kanannya terhadap nilai variabel keputusan yang diinginkan, terlihat pada tabel berikut ini.

   Tabel 2.1. Prosedur Fungsi Pencapaian

| Tujuan | Kemungkinan Simpangan | Prosedur |
|---|---|---|
| Xi ≥ bi | di | Minimasi di |
| Xi ≤ bi | di- | Minimasi $di^-$ |
| Xi = bi | di, $di^+$ | Minimasi di-, $di^+$ |

8. Tentukan nilai nonegatif. Langkah ini merupakan bagian resmi untuk perumusan masalah *goal programming* karena semua variabel yang digunakan pada model *goal programming* tidak boleh bemilai negatif.
9. Penyelesaian model *goal programming* dengan metodologi solusi seperti metode simpleks yang dimodifikasi.

### 2.5.1. Masalah Khusus Dalam *Goal programming*

Pada penyelesaian *goal programming* dengan metode simplex ini, juga akan dijumpai masalah-masalah yang sama seperti pada penyelesaian *linear programming* metode *simplex*.

Masalah-masalah tersebut meliputi hal-hal sebagai berikut.

1. *Alternative Optima*

   Dalam penyelesaian soal *goal programming* mungkin timbul solusi yang bersifat multipel.
2. Soal yang *Unbounded*

   Soal yang *unbounded* tidak terjadi dalam *goal programming*, karena setiap *goal constraint* mempunyai koefisien fungsi tujuan yang dikaitkan dengan koefisien sisi sebelah kanan kendala. Karenanya setiap solusi akan memenuhi atau tidak memenuhi koefisien sisi sebelah kanan *goal constraint.*
3. Solusi yang *Inconsistent* atau *Infeasible*

   Solusi yang inconsistent dapat terjadi dalam*goal programming.* Meskipun

demikian, *inconsistency* dalam *goal programming* bukanlah suatu masalah karena variabel-variabel deviasi yang ingin dipenuhi dinyatakan sebagai kendala.

### 2.5.2. Metode Pemecahan Masalah

Ada dua macam metode yang digunakan untuk menyelesaikan model *Goal programming*, yaitu metode grafis dan metode algoritma simpleks.

1. Metode Grafis

   Metode grafis digunakan untuk menyelesaikan masalah *Goal programming* dengan dua variable.Langkah-langkah penyelesaian dengan metode grafis adalah :

   a. Menggambar fungsi kendala pada bidang kerja sehingga diperoleh daerah yang memenuhi kendala.
   b. Meminimumkan variabel deviasional agar sasaran-sasaran yang diinginkan tercapai dengan cara menggeser fungsi atau garis yang dibentuk oleh variabel deviasional terhadap daerah yang memenuhi kendala.

2. Metode algoritma simpleks

   Algoritma simpleks digunakan untuk menyelesaikan masalah *Goal programming* dengan menggunakan variabel keputusan lebih dari dua. Langkah-langkah penyelesaian *Goal programming* dengan metode algoritma simpleks adalah :

   a. Membentuk tabel simpleks awal
   b. Pilih kolom kunci dimana Cj-Zj memiliki nilai negative terbesar. Kolom kunci ini disebut kolom pivot.
   c. Pilih baris kunci yang berpedoman pada bi/aij dengan rasio terkecil dimana bi adalah nilai sisi kanan dari setiap persamaan. Baris kunci ini disebut baris pivot.
   d. Mencari sistem kanonikal yaitu system dimana nilai elemen pivot bernilai 1 dan elemen lain bernilai nol dengan cara mengalikan baris pivot dengan -1 lalu menambahkannya dengan semua elemen dibaris pertama. Dengan demikian, diperoleh tabel simpleks iterasi *i*.

e. Pemeriksaaan optimalitas, yaitu melihat apakah solusi sudah layak atau tidak. Solusi dikatakan layak bila variabel adalah positif atau nol.

### 2.5.3. Masalah Bobot dan Prioritas Sasaran

Di dalam praktek organisasi, manajemen sering menghendaki suatu sasaran memperoleh prioritas untuk dicapai lebih dahulu dibanding prioritas-prioritas yang lain. Keinginan ini dapat dituangkan ke dalam model *Goal programming* dengan cara mengatur urutan peminimuman variabel deviasional. Urutan peminimuman variabel deviasional di dalam analisis geometrik akan menentukan urutan sasaran yang tercapai. Oleh karena itu, pengaturan prioritas sasaran yang hendak dicapai dapat dilakukan dengan mengendalikan urutan pemilihan variabel deviasional yang harus diminimumkan. Ada tiga macam sasaran di dalam model *Goal programming* yaitu,

1. Sasaran-sasaran dengan prioritas yang sama
2. Sasaran-sasaran dengan prioritas yang berbeda
3. Sasaran-sasaran dengan prioritas dan bobot yang berbeda

#### 2.5.3.1. Sasaran Dengan Prioritas Yang Sama

Model ini menganggap bahwa semua sasaran sama pentingnya sehingga apabila terpaksa harus ada sasaran yang dikorbankan agar sasaran yang lain tercapai. Dalam hal ini, penentuan sasaran mana yang harus dikorbankan atau sasaran mana yang harus tercapai tidak begitu penting karena semua sasaran dianggap mempunyai harga yang sama atau setiap sasaran yang dikorbankan mempunyai *opportunity cost* yang sama dengan sasaran yang terpilih. Karena setiap sasaran mempunyai *opportunity cost* yang sama, maka setiap variabel deviasional bisa dipilih untuk diminimumkan terlebih dahulu.

Sasaran dengan prioritas sama lebih menunjukkan keacuhan terhadap sasaran-sasaran yang akan dicapai dan bukan merupakan kondisi khusus yang harus diperhatikan di dalam penyelesaian atau proses penghitungan. Pembuktian di atas telah menunjukkan bahwa kasus *goal programming* dengan prioritas sasaran yang tidak berbeda memberikan keleluasaan di dalam penyelesaiannya.

### 2.5.3.2. Sasaran Dengan Prioritas Yang Berbeda

Urutan peminimuman variabel deviasional bisa dilakukan tanpa harus mengikuti suatu aturan tertentu. Cara ini, seperti telah dibahas sebelumnya, akan menghasilkan penyelesaian yang berbeda-beda. Oleh karena itu, kita bisa memilih sasaran mana yang akan memperoleh prioritas dengan cara memilih variabel deviasional yang berkaitan dengan sasaran itu untuk diminimumkan pertama kali.

Pemillhan variabel deviasional yang harus diminimumkan pertama kali adalah persoalan arbitrasi dan bukan berdasarkan pedoman atau formulasi matematis tertentu. Inilah salah satu keunikan model *goal programming.* Di dalam penyelesaian sebuah kasus *goal program ming*, kita hanya perlu memberi suatu notasi kepada setiap variabel deviasional di dalam fungsi tujuan agar kita dengan berpedoman notasi tersebut bisa mengurutkan peminimuman variabel deviasional sehingga sasaran-sasaran bisa dicapai sesuai dengan prioritas yang telah ditetapkan. Notasi yang digunakan untuk menandai prioritas sasaran tersebut adalah:

$$P_i(i = 1, 2, \ldots, m)$$

di mana $P_i$ bukan merupakan parameter atau variabel melainkan hanya sebuah notasi untuk menandai urutan prioritas sasaran yang hendak dicapai.

Dengan demikian, bentuk umum fungsi tujuan model *goal programming* dengan prioritas sasaran adalah :

$$\text{Min} \sum_{j=1}^{n} P_i \, (\text{DA}_i + \text{DB}_i)$$

Berikut akan diberikan sebuah contoh kasus penggunaan *Goal programming*. Perusahaan mebel ASRI yang memproduksi meja dan kursi. Setup minggu, perusahaan mendapat pasokan 100 lembar kayu mahoni. Untuk membuat sebuah kursi diperlukan 4 lembar kayu mahoni, dan untuk membuat meja diperlukan 6 lembar. Perusahaan memiliki 120 jam kerja-orang setiap minggunya (terdiri atas 3 orang karyawan yang bekerja 8 jam per hari dan bekerja 5 hari seminggu). Sebuah

kursi memerlukan waktu pengerjaan 4.5 jam kerja-orang. Sebuah meja memerlukan 5 jam-orang.

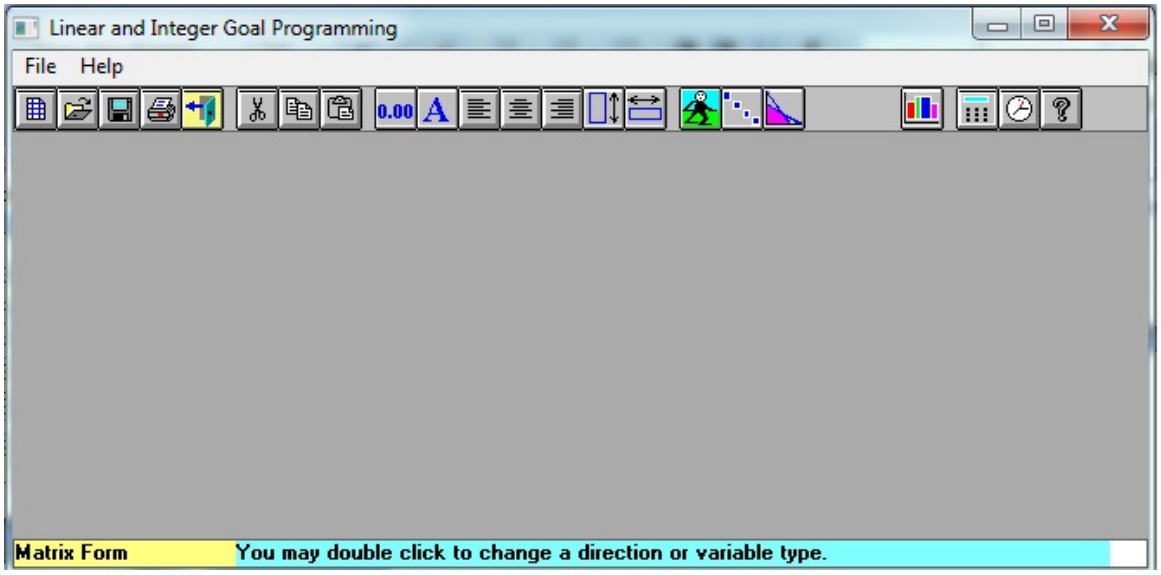
Perusahaan memperoleh laba sebesar Rp30,000 untuk setiap penjualan kursi dan Rp35.000 untuk setiap penjualan meja. Perusahaan dapat menjual semua meja dan kursi yang dibuatnya. Manajer perusahaan ingin memutuskan berapa banyak meja dan kursi yang harus dibuat agar diperoleh laba maksimum.

Sebagai informasi tambahan, manajer perusahaan juga ingin mencapai beberapa tujuan berikut:

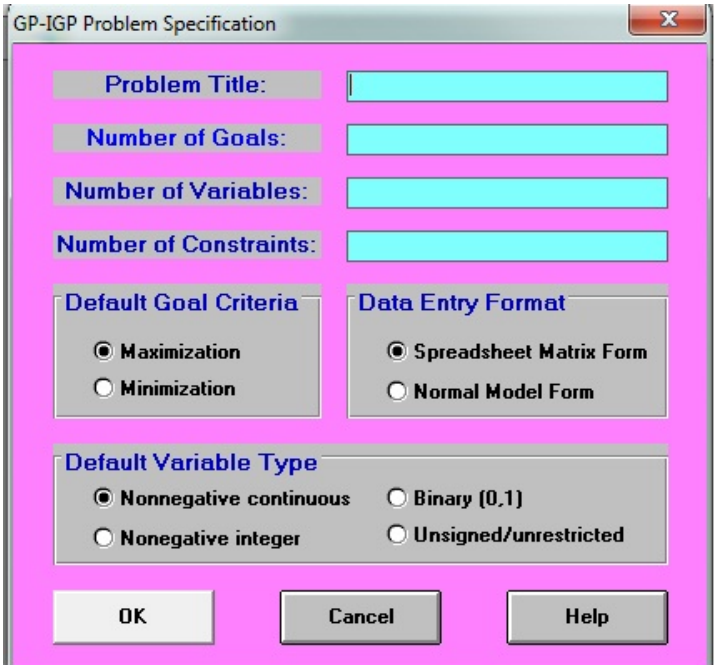
a. Laba yang diperoleh. setidak-tidaknya Rp700,000.
b. Meja diproduksi paling sedikit 10 buah.
c. Sebisa mungkin menggunakan jam kerja tidak lebih dari 100 jam kerja.

Tabel 2.2. Model *goal programming*

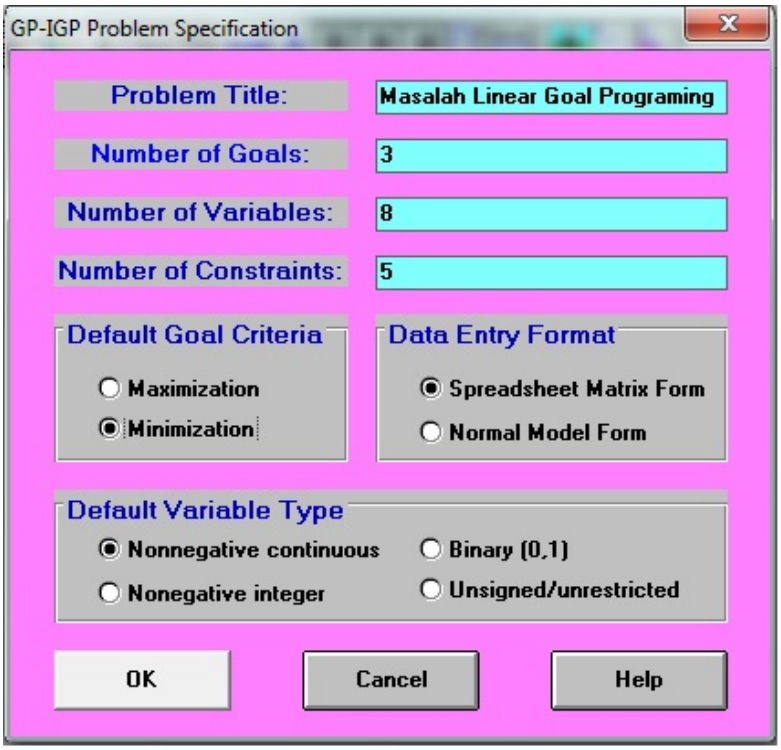
| | | |
|---|---|---|
| Fungsi tujuan | Maksimumkan laba: Rp30,000 K + Rp35,000 M | |
| Batasan | 4 K + 6 M <= 100 | Batasan jumlah kayu tersedia |
| | 4.5 K + 5 M <= 120 | Batasan jam kerja-orang tersedia |
| | K, M >= 0 | Batasan nonnegatif |
| Tujuan yang harus dicapai | 1. Rp 30,000 K + Rp 35.,000 M + U1 - E1 (tujuan laba)<br>2. Meja + U2 - E2 = 10 (tujuan produksi kursi)<br>3. 4,5 Kursi + 5 Meja + U3 - E3 = 100 (tujuan jam kerja) | |
| Prioritas untuk mencapai tujuan | Prioritas 1: U I (pencapaian di bawah Rp700,000)<br>Prioritas 2: U2 (prodtiksi kurang dari 10 kursi)<br>Prioritas 3: E3 (menghabiskan tenaga kerja lebih dari 100 jam) | |
| Tujuan prioritas | Prioritas 1: Minimisasi U1<br>Prioritas 2: Minimisasi U2<br>Prioritas 3: Minimisasi E3<br>U; jumlah kekurangan/sisa sisi kiri terhadap sisi kanan<br>$E_j$ jumlah kelebihan sisi kiri terhadap sisi kanan | |
| Variabel (ada 8) | K (kursi), M (meja), U1, U2, U3, E1, E2, E3 | |
| Batasan (ada 5) | (1) kayu yang tersedia; (2) jam kerja-orang tersedia; (3) tujuan laba; (4) tujuan produksi kursi; (5) pemakaian jam kerja | |



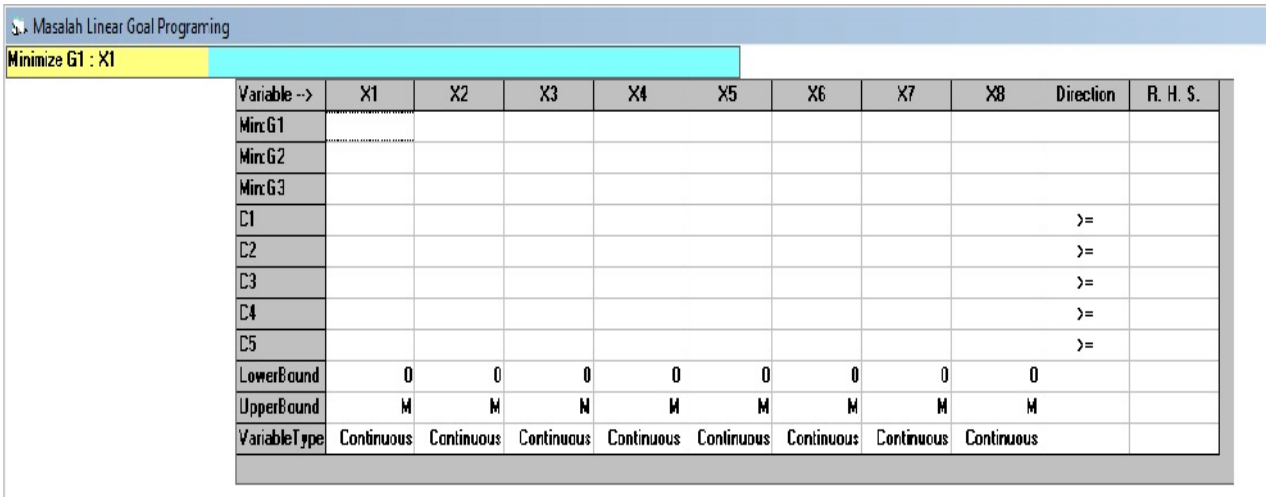

Gambar 2.3. Tampilan awal modul GP/IGP.



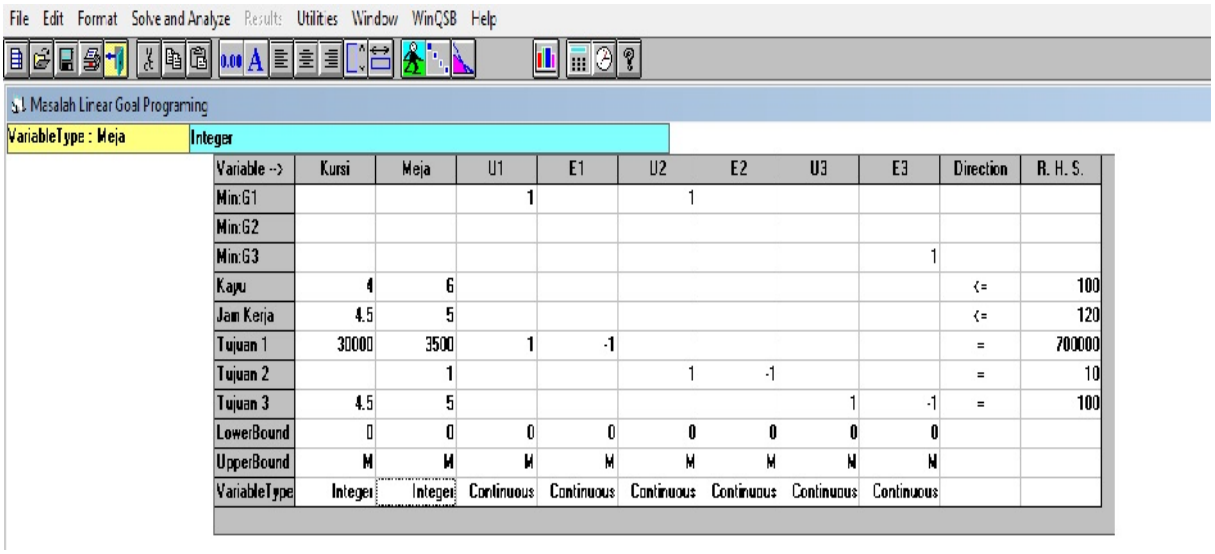

Gambar 2.4. Tampilan mengatur konfigurasi model GP.




Garnbar 2.5. Pengisian informasi dan aturan untuk GP/IGP.

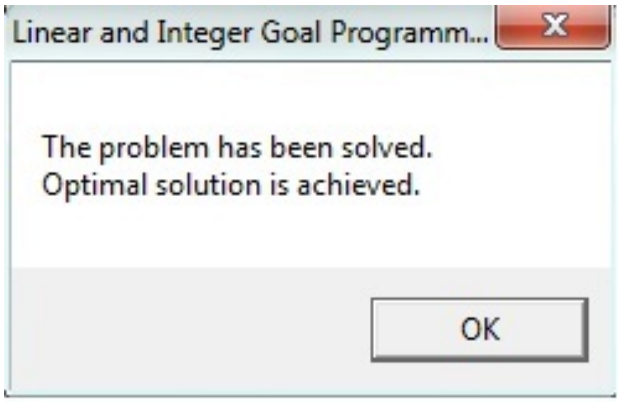



Gambar 2.6. Tampilan pengisian data untuk masalah GP/IGP.




| Variable --> | Kursi | Meja | U1 | E1 | U2 | E2 | U3 | E3 | Direction | R. H. S. |
|---|---|---|---|---|---|---|---|---|---|---|
| Min:G1 | | | 1 | | 1 | | | | | |
| Min:G2 | | | | | | | | | | |
| Min:G3 | | | | | | | | 1 | | |
| Kayu | 4 | 6 | | | | | | | <= | 100 |
| Jam Kerja | 4.5 | 5 | | | | | | | <= | 120 |
| Tujuan 1 | 30000 | 3500 | 1 | -1 | | | | | = | 700000 |
| Tujuan 2 | | 1 | | | 1 | -1 | | | = | 10 |
| Tujuan 3 | 4.5 | 5 | | | | | 1 | -1 | = | 100 |
| LowerBound | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| UpperBound | M | M | M | M | M | M | M | M | | |
| VariableType | Integer | Integer | Continuous | Continuous | Continuous | Continuous | Continuous | Continuous | | |

Gambar 2.7. Pengisian data untuk masalah GP/IGP telah selesai.




Gambar 2.8. Tampilan setelah ditemukan solusi.

| | 02:54:38 | | Saturday | July | 09 | 2011 | | |
|---|---|---|---|---|---|---|---|---|
| | Goal Level | Decision Variable | Solution Value | Unit Cost or Profit c(j) | Total Contribution | Reduced Cost | | |
| 1 | G1 | Kursi | 24.00 | 0 | 0 | 0 | | |
| 2 | G1 | Meja | 0 | 0 | 0 | -1.00 | | |
| 3 | G1 | U1 | 0 | 1.00 | 0 | 1.00 | | |
| 4 | G1 | E1 | 20,000.00 | 0 | 0 | 0 | | |
| 5 | G1 | U2 | 10.00 | 1.00 | 10.00 | 0 | | |
| 6 | G1 | E2 | 0 | 0 | 0 | 1.00 | | |
| 7 | G1 | U3 | 0 | 0 | 0 | 0 | | |
| 8 | G1 | E3 | 8.00 | 0 | 0 | 0 | | |
| 9 | G2 | Kursi | 24.00 | 0 | 0 | 0 | | |
| 10 | G2 | Meja | 0 | 0 | 0 | 0 | | |
| 11 | G2 | U1 | 0 | 0 | 0 | 0 | | |
| 12 | G2 | E1 | 20,000.00 | 0 | 0 | 0 | | |
| 13 | G2 | U2 | 10.00 | 0 | 0 | 0 | | |
| 14 | G2 | E2 | 0 | 0 | 0 | 0 | | |
| 15 | G2 | U3 | 0 | 0 | 0 | 0 | | |
| 16 | G2 | E3 | 8.00 | 0 | 0 | 0 | | |
| 17 | G3 | Kursi | 24.00 | 0 | 0 | 4.50 | | |
| 18 | G3 | Meja | 0 | 0 | 0 | 5.00 | | |
| 19 | G3 | U1 | 0 | 0 | 0 | 0 | | |
| 20 | G3 | E1 | 20,000.00 | 0 | 0 | 0 | | |
| 21 | G3 | U2 | 10.00 | 0 | 0 | 0 | | |
| 22 | G3 | E2 | 0 | 0 | 0 | 0 | | |
| 23 | G3 | U3 | 0 | 0 | 0 | 1.00 | | |
| 24 | G3 | E3 | 8.00 | 1.00 | 8.00 | 0 | | |
| | G1 | Goal | Value | (Min.) = | 10.00 | | | |
| | G2 | Goal | Value | (Min.) = | 0 | | | |
| | G3 | Goal | Value | (Min.) = | 8.00 | | | |
| | | | | | | | | |
| | Constraint | Left Hand Side | Direction | Right Hand Side | Slack or Surplus | ShadowPrice Goal 1 | ShadowPrice Goal 2 | ShadowPrice Goal 3 |
| 1 | Kayu | 96.00 | <= | 100.00 | 4.00 | 0 | 0 | 0 |
| 2 | Jam Kerja | 108.00 | <= | 120.00 | 12.00 | 0 | 0 | 0 |
| 3 | Tujuan 1 | 700,000.00 | = | 700,000.00 | 0 | 0 | 0 | 0 |

Gambar 2.9. Tampilan gabungan setelah ditemukan solusi.

| 07-09-2011 02:56:45 | Decision Variable | Solution Value | Basis Status | Reduced Cost Goal 1 | Reduced Cost Goal 2 | Reduced Cost Goal 3 |
|---|---|---|---|---|---|---|
| 1 | Kursi | 24.00 | at bound | 0 | 0 | 4.50 |
| 2 | Meja | 0 | at bound | -1.00 | 0 | 5.00 |
| 3 | U1 | 0 | at bound | 1.00 | 0 | 0 |
| 4 | E1 | 20,000.00 | basic | 0 | 0 | 0 |
| 5 | U2 | 10.00 | basic | 0 | 0 | 0 |
| 6 | E2 | 0 | at bound | 1.00 | 0 | 0 |
| 7 | U3 | 0 | at bound | 0 | 0 | 1.00 |
| 8 | E3 | 8.00 | basic | 0 | 0 | 0 |
| | Goal 1: | Minimize | G1 = | 10.00 | | |
| | Goal 2: | Minimize | G2 = | 0 | | |
| | Goal 3: | Minimize | G3 = | 8.00 | | |

Gambar 2.10. Hasil olahan model *goal programming*.

| 07-09-2011 02:57:14 | Constraint | Left Hand Side | Direction | Right Hand Side | Slack or Surplus | Shadow Price Goal 1 | Shadow Price Goal 2 | Shadow Price Goal 3 |
|---|---|---|---|---|---|---|---|---|
| 1 | Kayu | 96.00 | <= | 100.00 | 4.00 | 0 | 0 | 0 |
| 2 | Jam Kerja | 108.00 | <= | 120.00 | 12.00 | 0 | 0 | 0 |
| 3 | Tujuan 1 | 700,000.00 | = | 700,000.00 | 0 | 0 | 0 | 0 |
| 4 | Tujuan 2 | 10.00 | = | 10.00 | 0 | 1.00 | 0 | 0 |
| 5 | Tujuan 3 | 100.00 | = | 100.00 | 0 | 0 | 0 | 1.00 |

Gambar 2.11. Ringkasan batasan model *goal programming.*

Model *programming* yang mempunyai tujuan banyak. Karena tujuan yang banyak dan umumnya bersifat saling bertentangan, maka masalah *goal programming* selalu mempunyai tujuan minimisasi, yaitu meminimalkan penyimpangan terhadap semua tujuan. Dalam contoh pembahasan topik *goal programming*, tujuan-tujuan dapat bersifat *equal rank*, mempunyai ranking atau bahkan mempunyai prioritas.