2. TINJAUAN PUSTAKA

2.1 Goal Programming

Goal Programming merupakan pengembangan dari *Linear Programming*. Diperkenalkan oleh Charnes dan Cooper pada awal tahun 1960. Kemudian teknik ini disempurnakan oleh Ijiri pada pertengahan tahun 1960. Setelah itu terdapat penjelasan yang lengkap dengan beberapa aplikasi pengembangan oleh Ignizio dan Lee padan tahun 1970.

Perbedaan yang mencolok antara *Linear Programming* dengan *Goal Programming* adalah pada struktur dan penggunaan fungsi tujuan. Dalam *linear programming* tujuan yang ingin dicapai hanya satu, sedangkan pada *goal programming* tujuan yang ingin dicapai tidak hanya satu. Hal ini bisa dilakukan dengan cara mengekspresikan tujuan itu dalam bentuk kendala (*goal constraint*), memasukkan variabel simpangan (*deviation variable*) dalam kendala tersebut. Variabel simpangan tersebut digunakan menggambarkan seberapa jauh tujuan itu dicapai, dan menggabungkan variabel simpangan dalam fungsi tujuan.

Dalam *Linear Programming* tujuan yang dimiliki bisa berupa maksimasi atau minimasi. Sedangkan dalam *Goal Programming* tujuannya adalah meminimumkan penyimpangan–penyimpangan dari tujuan-tujuan tertentu. Hal ini berarti Goal Programming merupakan masalah minimasi.

Dikarenakan penyimpangan dari tujuan-tujuan yang diminimumkan, sebuah model *Goal Programming* dapat digunakan untuk menyelesaikan permasalahan yang memiliki tujuan yang berbeda, meskipun tujuan-tujuan tersebut memiliki dimensi atau satuan ukuran yang berbeda.

Dalam suatu perusahaan, pihak manajemen memiliki beberapa tujuan yang ingin dicapai, dan diantara target atau tujuan tersebut terkadang terjadi konflik. Permasalahan seperti ini dapat diselesaikan dengan menggunakan *Goal Programming*.

Jika perusahaan memiliki banyak tujuan yang ingin dicapai, maka prioritas atau urutan ordinalnya dapat ditentukan, dan proses penyelesaian *Goal Programming* berjalan sedemikian rupa sehingga tujuan-tujuan dengan prioritas tertinggi dipenuhi sedekat mungkin sebelum menyelesaikan tujuan-tujuan lain yang memiliki prioritas yang lebih rendah.

Pada *Linear Programming*, hasil akhir yang diperoleh adalah identifikasi solusi optimum dari suatu himpunan solusi yang layak. Sedangkan pada *Goal Programming*, hasil akhir yang diperoleh adalah identifikasi titik yang paling memuaskan dari persoalan dengan beberapa tujuan (*Romero,1991*).

2.2 Terminologi Goal Programming

Terdapat beberapa istilah dan lambang yang perlu dipahami dalam menyelesaikan permasalahan dengan *Goal Programming*. Berikut ini adalah istilah dan lambang yang digunakan dalam *Goal Programming*.

1. *Decision Variable*

   Decision Variable atau variabel keputusan merupakan sekelompok variabel yang tidak diketahui yang akan dicari nilainya. Dilambangkan dengan x_j, dimana j = 1,2,…,n.
2. *Right Hand Side*

   Right Hand Side atau nilai sisi kanan merupakan nilai-nilai yang menunjukkan ketersediaan sumber daya yang akan ditentukan kekurangan atau kelebihan penggunaannya. Dilambangkan dengan b_i.
3. *Goal*

   Goal atau tujuan merupakan keinginan yang ingin dicapai
4. *Goal Constraint*

   Goal Constraint merupakan sinonim dari *goal equation* atau kendala tujuan merupakan tujuan yang diekspresikan dalam persamaan matematik dengan memasukkan variabel simpangan.
5. *Preemptive Priority Factor*

   Preemptive Priority Factor merupakan suatu sistem urutan yang memungkinkan tujuan-tujuan yang disusun secara ordinal dalam model *Linear Goal Programming*. Dilambangkan dengan P_k, dimana k = 1,2,…,k . Sistem urutan ini menempatkan tujuan-tujuan dalam susunan dengan hubungan sebagai berikut:

$P_1 > P_2 >>> P_k$, dengan P_1 merupakan tujuan yang paling penting, P_2 merupakan tujuan yang kurang penting dan seterusnya.

6. *Deviational Variables*

   Deviational Variables atau variabel simpangan adalah variabel-variabel yang menunjukkan kemungkinan penyimpangan negatif dari suatu nilai RHS kendala tujuan. Dimana d^+ adalah penyimpangan positif dari suatu nilai RHS dan d^- adalah penyimpangan negatif dari RHS.

7. *Differential Weight*

   Differential weight atau bobot adalah timbangan matematik yang diekspresikan dengan angka kardinal. Dilambangkan dengan w_{ki}, dimana k = 1,2,…,k ; i = 1,2,…,m dan digunakan untuk membedakan variabel simpangan i dalam suatu tingkat prioritas k.

8. *Technological Coefficient*

   Technological coefficient atau koefisien teknologi merupakan nilai-nilai numerik yang menunjukkan penggunaan nilai b_I per unit untuk menciptakan x_j. Dilambangkan dengan a_{ij}.

Setiap model *Goal Programming* minimal memiliki 3 komponen, yaitu fungsi tujuan, kendala tujuan dan kendala non negatif.

2.2.1 Fungsi Tujuan

Terdapat 3 jenis fungsi tujuan *Linier Goal Programming*, yaitu:

$$Minimumkan\ Z = \sum_{i=1}^{m} d_i^- + d_i^+ \qquad (2.4)$$

Fungsi tujuan ini digunakan jika variabel simpangan dalam suatu masalah tidak dibedakan menurut prioritas atau bobot

$$Minimumkan\ Z = \sum_{i=1}^{m} P_k (d_i^- + d_i^+) \quad untuk\ k = 1,2,...,k \qquad (2.5)$$

Fungsi tujuan ini digunakan jika urutan tujuan diperlukan, tetapi variabel simpangan di dalam setiap tingkat prioritas memiliki kepentingan yang sama.

$$Minimumkan\ Z = \sum_{i=1}^{m} W_{ki} P_k (d_i^- + d_i^+) \quad untuk\ k = 1,2,...,k \qquad (2.6)$$

Fungsi tujuan ini digunakan jika tujuan-tujuan diurutkan dan variabel simpangan pada setiap tingkat prioritas dibedakan dengan menggunakan bobot yang berlainan w_{ki}.

2.2.2 Kendala Tujuan

Terdapat 6 jenis kendala tujuan yang berbeda satu sama lain. Kendala-kendala tersebut ditentukan oleh hubungannya dengan fungsi tujuan.

Tabel 2.1.Tabel Jenis Kendala Tujuan

Kendala Tujuan	Variabel Simpangan Dalam Fungsi Tujuan	Kemungkinan Simpangan	Penggunaan Nilai RHS yang Diinginkan
$a_{ij}x_j + d_i^- = b_i$	d_i^-	Negatif	$= b_i$
$a_{ij}x_j - d_i^+ = b_i$	d_i^+	Positif	$= b_i$
$a_{ij}x_j + d_i^- - d_i^+ = b_i$	d_i^-	Negatif dan Positif	b_i atau lebih
$a_{ij}x_j + d_i^- - d_i^+ = b_i$	d_i^-	Negatif dan Positif	b_i atau kurang
$a_{ij}x_j + d_i^- - d_i^+ = b_i$	d_i^- *dan* d_i^+	Negatif dan Positif	$= b_i$
$a_{ij}x_j - d_i^+ = b_i$	d_i^+	Tidak ada	$= b_i$

2.2.3 Kendala Non-Negatif

Untuk kendala non-negatif pada *Goal Programming* sama dengan kendala non-negatif pada *Linear Programming*. Variabel-variabel pada model *Linear Goal Programming* biasanya bernilai lebih besar atau sama dengan nol. Semua model *Linear Goal Programming* terdiri dari simpangan dan variabel keputusan, sehingga pernyataan non negatif dilambangkan sebagai berikut: $x_j, d^-_i, d^+_i \geq 0$.

2.2.4 Kendala Struktural

Kendala struktural merupakan kendala-kendala lingkungan yang tidak berhubungan langsung dengan fungsi tujuan masalah yang dipelajari. Variabel simpangan tidak dimasukkan dalam kendala ini. Oleh karena itu kendala ini tidak dimasukkan ke dalam fungsi tujuan.

2.3 Asumsi Model Goal Programming

Untuk menyelesaikan permasalahan dengan menggunakan *Goal Programming* terdapat asumsi-asumsi dasar yang perlu diperhatikan. Asumsi-asumsi tersebut adalah sebagai berikut:

a. Additivitas dan Linearitas

Diasumsikan bahwa proporsi penggunaan b_i yang ditentukan oleh a_{ij} harus tetap benar tanpa memperhatikan nilai solusi x_j yang dihasilkan. Hal ini berarti bahwa *Left Hand Side* (LHS) dari kendala tujuan harus sama dengan dengan nilai sisi kanan.

b. Divisibilitas

Diasumsikan bahwa nilai-nilai x_j, d^+_i, dan d^-_i yang dihasilkan dapat dipecah. Artinya kita dapat menyelesaikan jumlah pecahan nilai x_j dan menggunakan jumlah pecah sumber daya dalam solusi tersebut. Asumsi ini tidak membatasi penggunaan model Goal Programming.

c. Terbatas

Diasumsikan bahwa nilai-nilai x_j, d^+_i, dan d^-_i yang dihasilkan harus terbatas. Artinya kita tidak memiliki nilai variabel keputusan, sumber daya, atau penyimpangan tujuan yang tak terbatas.

d. Kepastian dan Periode Waktu Statis

Diasumsikan bahwa parameter model Goal Programming seperti a_{ij}, b_j, P_k dan w_{ki} diketahui dengan pasti dan mereka akan tetap statis selama periode perencanaan dimana hasil model digunakan.

2.4 Perumusan Masalah *Goal Programming*

Perumusan model *Goal Programming* terdiri dari beberapa langkah. Langkah-langkah tersebut adalah sebagai berikut:

1. Menentukan variabel keputusan.

   Semakin tepat mendefinisikan variabel keputusan maka akan semakin mudah memodelkannya.

2. Menentukan sistem kendala.

   Faktor yang paling menentukan adalah penentuan nilai-nilai sisi kanan dan kemudian menentukan koefisien teknologi yang cocok dan variabel keputusan yang dikutsertakan dalam kendala. Selain itu juga perlu diperhatikan jenis penyimpangan yang diperbolehkan dari nilai right hand side atau nilai sisi kanan. Jika penyimpangan diperbolehkan dalam dua arah, maka kedua variabel simpangan tersebut ditempatkan pada kendala tersebut. Tetapi jika penyimpangan yang diperbolehkan hanya arah, maka hanya terdapat satu variabel simpangan pada kendala tersebut.

3. Menentukan prioritas utama.

   Penentuan urutan merupakan preferensi yang berbeda-beda dari setiap individu yang ada dalam perusahaan. Selain itu juga dipengaruhi oleh keterbatasan sumber daya yang dimiliki dan batasan-batasan lain yang secara eksplisit ataupun implisit menentukan dalam pemilihan variabel keputusan.

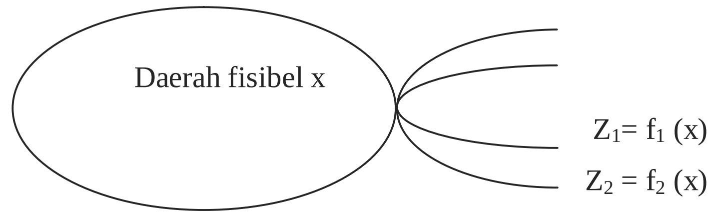
4. Menentukan bobot.
5. Menentukan fungsi tujuan.

   Setiap fungsi tujuan harus dinyatakan sebagai fungsi dari variabel keputusan yang dilambangkan dengan $f_i(x_j)$, yaitu fungsi dari variabel keputusan yang berhubungan dengan tujuan ke-i, sedangkan x adalah vektor variabel keputusan yang dilambangkan dengan $a_{ij}x_j$, dimana a_{ij} merupakan batasan koefisien teknologi.

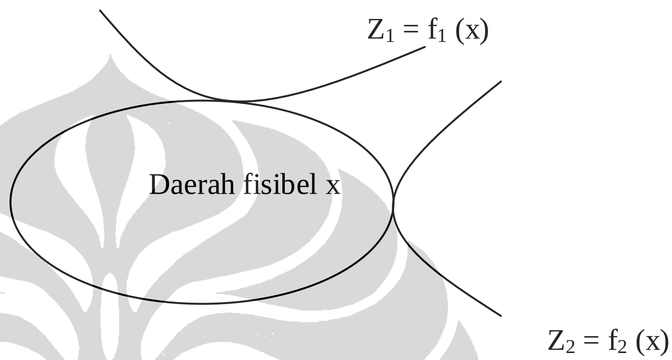
   Setiap fungsi tujuan memiliki nilai yang berhubungan dengan nilai sisi kanan (b_i) yang merupakan target atau tujuan yang ingin dicapai dari fungsi tujuan tersebut.

2.4.1 Kriteria Pertentangan

Beberapa kriteria tujuan dikatakan konflik jika salah satu tujuan dipenuhi, maka akan mengakibatkan tingkat kepuasan tujuan lainnya akan berkurang, atau pada setiap kenaikan nilai salah satu kriteria diikuti oleh penurunan nilai pada kriteria lainnya (*Tabucanon, 1988*). Pada gambar dapat dilihat fungsi tujuan yang tidak konflik satu sama lain (gambar 2.1) dan fungsi tujuan yang saling konflik (gambar 2.2) di bawah ini:



Gambar 2.1 Fungsi Tujuan yang Tidak Konflik



Gambar 2.2 Fungsi Tujuan Saling Konflik

2.4.2 Pengambilan Keputusan Multi Tujuan

Pengambilan keputusan multi tujuan merupakan suatu pengambilan keputusan, dimana tujuan-tujuan yang ditetapkan bisa bertolak belakang (*conflicting goal*), tujuan yang sejalan (*unconflicting goal*). Model multi tujuan ditandai dengan vektor fungsi tujuan yang memiliki banyak dimensi. Jika terdapat n tujuan, maka bisa dibuat suatu model multi tujuan sebagai berikut:

Minimasi $Z = (Z_{1(x)}, Z_{2(x)}, ..., Z_{n(x)}) \| x \in y$ (2.7)

Dimana ; Z = fungsi tujuan yang ingin dicapai

x = n variabel keputusan

y = himpunan solusi yang memenuhi batasan yang ada

Terdapat beberapa pendekatan yang bisa digunakan dalam menyelesaikan problem multi tujuan, yaitu:

1. Metode Utilitas

   Metode ini mengekspresikan semua tujuan yang ingin dicapai dalam ukuran mata uang. Dengan cara demikian problem multi tujuan dapat ditransformasikan ke dalam satu fungsi tujuan. Kelemahan dalam metode ini adalah ketidakakuratan transformasi tujuan yang berbeda dalam suatu ukuran yang sama.

2. Metode Ranking/Prioritas

   Metode ini menggunakan ranking atau prioritas untuk menunjukkan tingkat kepentingan masing–masing tujuan yang ditetapkan.

3. Metode Solusi Efisien

   Dalam metode ini dibangkitkan suatu himpunan solusi efisien dari himpunan solusi yang memenuhi fungsi batasan yang ada. Himpunan solusi efisien dari himpuan lain dapat digunakan sebagai perbaikan untuk satu tujuan atau lebih.

Programa tujuan ganda (*Multiple Goal Programming*) merupakan metode yang menggunakan struktur prioritas dari tujuan-tujuan yang dipertimbangkan dengan alasan sebagai berikut:

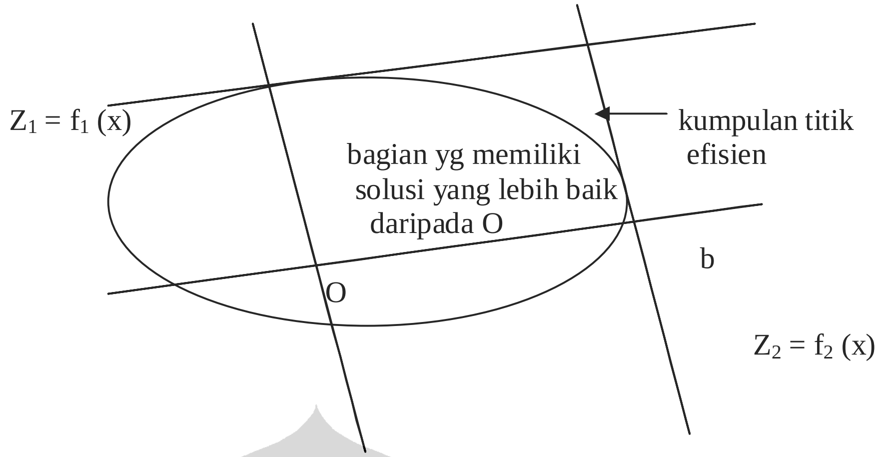
1. Pengembangan model relatif sederhana.
2. Metode pemecahan masalah sangat sederhana dan hanya merupakan perbaikan atau modifikasi dari metode simplek dua fase.
3. Model dan asumsinya konsisten dengan problem dunia nyata.

2.4.3 Optimasi Multi Tujuan

Solusi optimal dalam pengertian klasik adalah mencapai solusi optimal dari semua tujuan secara serentak. Solusi X^* adalah solusi optimal yang didefinisikan jika $x^* \in S$ dan $f_l(x^*) \geq f_l(x)$ untuk semua i dan untuk semua $x \in S$, dimana S adalah daerah fisibel (*feasible region*).

Pada kriteria tujuan yang terjadi konflik, tidak akan terdapat solusi yang benar-benar optimal. Oleh karena itu digunakan suatu istilah solusi efisien atau solusi kompromi. Pada konsep solusi efisien, tidak terdapat kenaikan yang dihasilkan pada suatu tujuan tanpa menyebabkan penurunan secara simultan pada tujuan yang lain.

Solusi x^* dikatakan efisien jika tidak ada $x \in S$. Sehingga $f_l(x) \geq f_l(x^*)$ untuk semua i dan $f_l(x) > f_l(x^*)$ untuk paling sedikit satu (l).



Gambar 2.3 Konsep Solusi Efisien

2.5 Model EOQ pada Multi Produk Inventori

Setelah penemuan formula EOQ (*Economic Order Quantity*) tahun 1918 oleh Wilson, berbagai EOQ model telah diselesaikan oleh banyak peneliti didasarkan dengan berbagai asumsi dan literatur yang ditinjau oleh banyak pengarang (Raymond, Hadley dan Whitin, Clark, Silver, dll), Di kebanyakan kasus-kasus dihubungkan langsung dengan pelajaran dan teori dari satu produk dan satu system instalasi. Sebagai contoh satu produk disimpan pada satu lokasi, dengan memanfaatkan keberhasilan beberapa teknik optimasi yang menghasilkan solusi optimal. Penekanan dalam riset yang menerapkan teori multi produk di berbagai lokasi relatif lebih sedikit. Hal Ini berkaitan dengan kurang tersedianya dan pemanfaatan yang kurang sesuai dengan teknik optimasi *multi-objective* pada area pengambilan keputusan.

Situasi permasalahan di dunia nyata, banyak timbul konflik dari masalah inventori multi produk seperti fungsi-fungsi biaya, anggaran dll. Beberapa kasus, fungsi biaya adalah tidak linier secara alami yang berkaitan dengan biaya yang tersedia di perusahaan. Saat ini, Teknik *Goal Programming* yang dikembangkan oleh Charnes dan Cooper, dan metodelogi pengembangan teknik GP oleh Lee, Ignizio, dll, telah digunakan pada sistem control untuk inventori multi produk. Pada Goal Programming, tingkat target yang diharapkan disatukan untuk setiap tujuan-tujuan. Kemudian tujuan-tujuan seperti halnya batasan-batasan structural dipertimbangkan sebagai goal dengan dibawah dan lebih deviational variable-variabel ke masing-masing diantara mereka untuk keputusan jalan positif atau negative jika mereka tidak tercapai. Goal kemudian menjadi rank-order menurut pentingnya mereka dalam konteks membuat keputusan. Sekali diantara prioritas-prioritas dipilih, bobot diberikan menurut kepentingan relative dari goal pada tingkat prioritas yang sama. Secara keseluruhan dalam literatur biasanya ada dua jenis GP yang ditemukan: (1) penyimpangan yang dihargai untuk semua gol mencapai nilai minimal menggunakan sasaran tunggal dan (2) penyimpangan ditempatkan menurut kepentingan mereka dalam konteks pengambilan keputusan pada order fashion lexicographical dan mengoptimalkan dari tingkat prioritas tertinggi yang didapatkan pertama kemudian pencapaian optimal dari tingkat prioritas berikutnya. Penggunaan teknik GP, Golanly yang telah mengembangkan model inventory yang diaplikasikan pada pabrik kimia besar untuk kepentingan memenuhi beberapa sasaran konflik secara kwantitatif.

Penggunaan teknik NLGP, yang dikembangkan oleh Basu merupakan prosedur solusi dalam memecahkan masalah inventori multi produk dimana permasalahannya mempunyai karakteristik dynamic programming. Charnes dan Collomb, memperkenalkan the goal interval programming dimana pengambil keputusan bebas untuk memilih sebuah interval dan menghukum penyimpangan dari kedua batas interval.

Fungsi pinalti-U yang popular yang diperkenalkan oleh Charnes menyatakan penyimpangan-penyimpangan dari goal dikemukakan sebagai penghargaan untuk hukuman. Metode ini per hukuman terjadi antar setiap pasangan dari tujuan-tujuan dengan adanya batas atas ke variable-variabel deviational. Hanyalah untuk memperkenalkan batas-batas variable deviational tidak terbatas pada variable deviational yang akan menjadi dasar mulainya.

Baru-baru ini Jones dan Tamiz mengajukan teknik model pilihan yang menarik dimana per unit hukuman terjadi pada tujuan yang dipilih dengan menghilangkan batas dari variable deviational melalui pembukaan tujuan ekstra untuk target deviational berikutnya. Dan di proses ini finalisasi akan dilaksanakan menurut pilihan pembuat keputusan. Bagaimanapun semua metode memerlukan penentuan struktur prioritas yang sesuai tetapi seleksi prioritas merupakan usaha yang sulit untuk mengambil keputusan untuk lingkungan yang kompleks. Untuk mengatasi kesukaran ini maka perlu membahas metoda yang dapat digunakan sebagai cara menentukan memperoleh struktur prioritas solusi kompromi kemungkinan terbaik.

Struktur prioritas yang baik dari GP dipilih untuk mendapatkan solusi yang optimal. Pada prosedur jawaban sensitivtas analisis dengan variasi dari struktur prioritas yang diharapkan dari permasalahan telah dikerjakan. Menggunakan struktur prioritas yang berbeda, akan memperoleh kemungkinan solusi yang berbeda. Solusi yang ideal dari model kemudian di identifikasi dari solusi yang dihubungkan dengan struktur prioritas yang berbeda. Akhirnya diperoleh hasil yang menunjukkan nilai minimum dari D_i^-, jarak solusi yang berbeda ke solusi ideal yang diidentifikasi merupakan solusi kompromi terbaik.

2.5.1 Formulasi Model

Model umum dari masalah inventori multi produk adalah :

$$minimize \sum_{i=1}^{n} \left(C_i q_i + \frac{C_{si}}{q_i} \right) \tag{2.8}$$

$$\sum_{i=1}^{n} a_{ij} q_i \leq = \geq b_j, \quad 1 \leq j \leq m \tag{2.9}$$

$$0 < q_i \leq Q_i \quad 1 \leq i \leq n$$

Dimana C_1, C_2, … C_n adalah jumlah biaya-biaya pemesanan dan C_{s1}, C_{s2}, … C_{sn} adalah biaya set-up dan Q_1, Q_2, … Qn merupakan tingkat inventori dari q_1, q_2, … q_n berturut-turut.

Pada (2.8) *cost* yang dihubungankan dengan produk q_1, q_2, …q_n adalah tujuan yang hendak diminimumkan oleh beberapa batasan-batasan. Sejak biaya berhubungan dengan masing-masing produk adalah mandiri dan mereka adalah saling berhubungan oleh beberapa batasan sehingga meminimalkan keseluruhan sasaran fungsi tujuan mengikuti batasan-batasan yang dapat diperlakukan sebagai n fungsi sasaran tujuan kepada batasan-batasan yang sama dalam kaitan dengan disaggregation ini solusi yang terakhir tidak akan efektif. Tujuan (2.8) dapat dipertimbangkan sebagai fungsi tujuan. Pada GP setiap tujuan $C_i q_i$ + (C_{si}/q_i) merupakan tujuan dengan variable deviasional untuk dibawah adalah d_{ik}^{-} dan diatas d_{ik}^{+} dan level target G_i, ($1 \le i \le n$).

Prioritas didasarkan pada model NLGP untu permasalahan inventori multi produk dengan bentuk : mendapatkan $\bar{q}$ (q_1, q_2, … q_n) menjadi :

$$minimize \quad P_1 \left[\sum_i \left(w_{i1}^{-} d_{i1}^{-} + w_{i1}^{+} d_{i1}^{+} \right) + \sum_j \left(w_{n+j,1}^{-} d_{n+j,1}^{-} + w_{n+j,1}^{+} d_{n+j,1}^{+} \right) \right] \tag{2.10}$$

. . . .

$$minimize \quad P_k \left[\sum_i \left(w_{ik}^{-} d_{ik}^{-} + w_{ik}^{+} d_{ik}^{+} \right) + \sum_j \left(w_{n+j,k}^{-} d_{n+j,k}^{-} + w_{n+j,k}^{+} d_{n+j,k}^{+} \right) \right]$$

. . . .

$$minimize \quad P_K \left[\sum_i \left(w_{ik}^{-} d_{ik}^{-} + w_{ik}^{+} d_{ik}^{+} \right) + \sum_j \left(w_{n+j,k}^{-} d_{n+j,k}^{-} + w_{n+j,k}^{+} d_{n+j,k}^{+} \right) \right]$$

$$C_i q_i + \frac{Csi}{q_i} + d_{ik}^{-} - d_{ik}^{+} = G_i, \qquad 1 \le i \le n \tag{2.11}$$

$$\sum_{i=1}^{n} a_{ij} q_i + d_{n+j,k}^{-} - d_{n+j,k}^{+} = b_j, \qquad 1 \le j \le m \tag{2.12}$$

$$0 < q_i \le Q_i$$

$$d_{ik}^{-} \; d_{ik}^{+} = d_{n+j,k}^{-} \; d_{n+j,k}^{+} = 0$$

$$d_{ik}^{-}, \;\; d_{n+j,k}^{-}, \;\; d_{ik}^{+}, \;\; d_{n+j,k}^{+} \ge 0, \qquad 1 \le i \le n, \qquad 1 \le j \le m, \qquad 1 \le k \le K$$

Dimana Pk ($1 \le k \le K : K \le n + m$), k adalah factor prioritas yang ditugaskan untuk menyatukan tujuan dari group bersama dalam formulasi masalah. P_1, P_2, … P_K mewakili berat/beban prioritas atau prioritas pembelian yang lebih dahulu menentukan hirarki tujuan. Tujuan tingkatan pada prioritas yang lebih tinggi dicukupi pertama dan baru setelah itu boleh tujuan prioritas yang lebih rendah dipertimbangkan. Menurunkan prioritas tujuan tidak bisa mengubah pencapaian tujuan tersebut dari tingkatan prioritas yang lebih tinggi. Sistem Prioritas pembelian lebih dahulu didasarkan pada lexicographical yang memesan menurut pentingnya komponen mereka, Lee[12]. Jika seperti prioritas tujuan ditingkatan prioritas P1 tidak pernah dapat dicukupi oleh tujuan pada prioritas pengukur P2 dan seterusnya. w^{-}_{ik}, $w^{-}_{n+j,k}$ (≥ 0) dan w^{+}_{ik}, $w^{+}_{n+j,k}$ (≥ 0) adalah anak timbangan yang kuantitatif berhubungan dengan deviational variabel d^{-}_{ik}, $d^{-}_{n+j,k}$ dan d^{+}_{ik}, $d^{+}_{n+j,k}$ ($1 \leq i \leq n$; $1 \leq k \leq K$) berturut-turut.

2.5.2 Prosedur Penyelesaian

Di dalam prioritas konvensional berdasarkan NLGP, solusi menerangkan pembuat keputusan menekan bahwa struktur prioritas diperlakukan sebagai solusi yang optimal. Tetapi berbeda di dalam situasi pengambilan keputusan yang kompleks, solusi yang diinginkan tidak harus diterima di bawah struktur berat. Dengan demikian suatu solusi lebih baik selalu diharapkan sebagai prioritas alternatif di bawah berat/beban yang harus dipertimbangkan. Untuk memilih struktur prioritas yang sesuai di bawah struktur berat/beban, analisis kepekaan perlu dilakukan. Di dalam model, prioritas K dipertimbangkan. Maka keterlibatan prioritas K menunjukkan bahwa K! solusi dapat berbeda yang diperoleh dari K! permasalahan yang muncul untuk K! adalah jumlah struktur prioritas berbeda.
Misalkan { $q_1^{(r)}$, $q_2^{(r)}$,······ $q_n^{(r)}$}, $1 \leq r \leq K!$ maka K! jumlah solusi yang diperoleh dengan mengubah susunan tingkatan prioritas K. Misalkan untuk i = n .

$$\min_{1 \leq r \leq K!} \left(C_n q_n^{(r)} + \frac{C_{sn}}{q_n^{(r)}} \right) = \min_{1 \leq r \leq K!} \left(C_n q_n^{*} + \frac{C_{sn}}{q_n^{*}} \right) \qquad (2.13)$$

dimana q_n adalah manapun nilai $q_n^{(r)}$, $1 \leq r \leq K!$ kemudian solusi yang ideal q digambarkan oleh { q_1, q_2,······ q_n} Cohon[6].
Tetapi dalam praktek solusi ideal tidak pernah dapat dicapai. Solusi, adalah yang terdekat ke solusi yang ideal, diterima sebagai solusi kompromi yang terbaik, dan sesuai dengan struktur prioritas yang dikehendaki ketika struktur prioritas yang paling sesuai dalam kaitannya dengan perencanaan. Untuk memperoleh solusi kompromi yang terbaik, mengikuti permasalahan GP diharapkan mampu menjadi pemecahan.

$$\min_{1 \leq r \leq K!} \sum_{i=1}^{n} \left(d_{ir}^{+} + d_{ir}^{-} \right) \qquad (2.14)$$

pada subjek

$$q_i^* - q_i^{(r)} + d_{ir}^- - d_{ir}^+ = 0, \quad 1 \le r \le n \tag{2.15}$$

dan

$$d_{ir}^+ \ge 0 \;\; d_{ir}^- \ge 0, \;\; d_{ir}^+ . d_{ir}^- = 0, \quad 1 \le r \le K!, \quad 1 \le i \le n \tag{2.16}$$

dimana, d_{ir}^- dan d_{ir}^+ adalah di bawah- dan over- deviational variable yang berurutan. Sekarang,

$$(D_1)^r = \sum_{i=1}^{n} \mid q_i^* - q_i^{(r)} \mid \tag{2.17}$$

D_1-distance dari solusi yang ideal { q_1, q_2,…… q_n}, ke r-th solusi { $q_1^{(r)}$, $q_2^{(r)}$, … $q_n^{(r)}$}, $1 \le r \le K!$. Oleh karena itu,

$$\begin{aligned} (D_1)_{opt} &= \min_{1 \le r \le K!} (D_1)^r = \min_{1 \le r \le K!} \sum_{i=1}^{n} \mid q_i^* - q_i^{(r)} \mid \\ &= \min_{1 \le r \le K!} \sum_{i=1}^{n} (d_{ir}^+ + d_{ir}^-) = \sum_{i=1}^{n} (d_{ir}^+ + d_{ir}^-), \;\; say \\ &= (D_1)^p, \quad 1 \le p \le K! \end{aligned}$$

Oleh karena itu, {q_1, q_2, …..q_n}, adalah solusi kompromi yang terbaik menurut situasi masalah tersebut.