

II. TINJAUAN PUSTAKA

2.1 Tinjauan Penelitian Terdahulu

Usaha perkebunan kelapa sawit (*Elaeis guineensis* Jacq.) merupakan salah satu usaha agribisnis yang memuat serangkaian perencanaan yang sangat kompleks seperti perencanaan sumber daya manusia (tenaga kerja), sumber daya lahan (teknik pengolahan dan pemanfaatan lahan), sumber daya modal (pengelolaan keuangan perusahaan), hingga sistem manajemennya. Perencanaan sumber daya manusia dan modal, dalam penelitian ini untuk aplikasi tandan kosong kelapa sawit tidaklah sederhana. Pengambil keputusan perlu mempertimbangkan berbagai aspek seperti jam kerja karyawan, penjadwalan aplikasi, hingga perhitungan biaya yang harus dikeluarkan. Poin yang paling penting adalah cara untuk menentukan kombinasi yang tepat dari ketiga aspek tersebut di atas.

Menurut Luthfiyanti (2003) dalam penelitiannya menyebutkan bahwa solusi untuk permasalahan dengan tujuan ganda tidak dapat diselesaikan dengan *linear programming*, tetapi dengan *goal programming* karena tujuan-tujuan yang ditetapkan saling bertentangan satu sama lain serta saling berbeda tingkat kepentingannya. Penelitian tersebut menggunakan 40 kendala tujuan, 12 kendala fungsional dan 16 variabel keputusan serta penetapan deviasi atas dan deviasi bawah untuk setiap nilai *Right Hand Side* (RHS) yang diinginkan untuk dicapai.

Menurut Ardiana (2011) dalam penelitiannya menyebutkan bahwa masalah manajemen hara dalam penelitian tersebut dicari gabungan kombinasi pupuk sehingga dicapai hasil panen maksimal dengan biaya minimal. GP (*Goal Programming*) adalah teknik penting untuk memecahkan masalah tersebut. Dalam GP konvensional, parameter masalah perlu didefinisikan secara tepat. Dalam masalah perencanaan pertanian, nilai dari beberapa parameter mungkin tidak diketahui persis atau didefinisikan dalam arti samar. Kesamaran dapat bersumber dari faktor keacakan statistik atau sifat *fuzzy* dari informasi. Konsep *fuzzy* dianggap dapat menggambarkan situasi nyata yang sebenarnya terjadi.

Ardiana menggunakan pendekatan Hannan dalam konsep *fuzzynya*. Alasannya adalah karena pendekatan Hannan lebih sederhana dibandingkan dengan pendekatan yang lain (pendekatan Narasimhan atau Chebyshev model, dan pendekatan Tiwari atau *weighted distance matrices model*)

Winarti (2009) dalam penelitiannya menyebutkan bahwa dalam pengambilan keputusan yang samar dan subyektif diperlukan suatu model keputusan yang mempertimbangkan *fuzzyness* (ketidakpastian) untuk memungkinkan menampung faktor *imprecise* (tidak jelas) dalam proses penetapan proses kriteria. Penggunaan Goal Programming sendiri untuk menyelesaikan masalah ini memerlukan penetapan tingkat aspirasi subyektif untuk mencapai suatu goal. Karenanya model dalam penelitiannya menggunakan konsep *fuzzy* hanya pada penetapan tingkat pencapaian suatu goal yang menjadi aspirasi pengambil keputusan. Penetapan preferensi berdasarkan *judgement* manusia untuk memberikan pembobotan level aspirasi dan prioritas goal pada suatu elemen keputusan sangat berkaitan dengan konsep *fuzzy*. Konsep *fuzzy* merupakan perluasan dari konsep *crisp* yang dikembangkan oleh Lutfi A. Zadeh pada tahun 1965.

Ada perbedaan mendasar antara himpunan *crisp* dan *fuzzy*. Batas keanggotaan pada himpunan *crisp* adalah jelas yaitu $[0, 1]$ yang berarti bahwa nilai nol menunjukkan suatu item tidak menjadi anggota dalam suatu himpunan, sedangkan nilai satu menunjukkan bahwa suatu item menjadi anggota suatu himpunan. Batas keanggotaan pada himpunan *fuzzy* adalah tidak jelas/samar yaitu $(0, 1)$. Teori *fuzzy* digunakan untuk menyelesaikan masalah penentuan goal yang tidak pasti dan sebagai konsep dasar dalam suatu *framework* untuk menampung adanya informasi yang samar. Penggunaan pendekatan Fuzzy Goal Programming (FGP) digunakan karena lebih fleksibel daripada GP dan tujuan tidak perlu didefinisikan secara akurat (Cox, 1994).

Optimalisasi penggunaan sumber daya dalam usaha perkebunan kelapa sawit dihadapkan pada ketersediaan sumber daya yang terbatas jumlahnya. Lee dan Weng (1997) dalam penelitiannya menyebutkan bahwa untuk melakukan manajemen kualitas air pada lembah sungai, para pengambil keputusan (*decision maker*) pemerintah harus memiliki level aspirasi yang pasti terhadap serangkaian

goal yang ingin dicapai. Pada kondisi *existing* pemerintahan, pengambil keputusan (*decision maker*) tidak memiliki level aspirasi yang pasti agar serangkaian *goal* yang ada dianggap tercapai. Hal ini menyebabkan kesamaran (*vagueness*) dalam penentuan level aspirasi dari serangkaian *goal* yang ingin dicapai, sehingga menyulitkan dalam mencari solusi kompromi agar serangkaian *goal* yang ada terpenuhi.

2.2 Manajemen Tenaga Kerja

Manusia merupakan sumber daya yang sangat kompleks dan sulit diprediksi sehingga dalam mengelola sumber daya manusia diperlukan usaha yang besar. Beberapa faktor yang perlu diperhatikan dalam mengelola sumber daya manusia/tenaga kerja adalah hubungan manusia (*human relations*), pengelolaan pribadi tenaga kerja (*personal management of labor*), pengelolaan tenaga kerja secara umum (*impersonal management of labor*), serta hubungan *industry (industrial relations)* (Anderson dan Stuart,1981).

Dengan melihat faktor dalam manajemen tenaga kerja, maka dapat dikatakan bahwa manajemen tenaga kerja adalah suatu pengelolaan tenaga kerja dengan melakukan proses pengambilan keputusan yang berhubungan dengan:

1. Penentuan ukuran dan jumlah tenaga kerja serta personil proyek
2. Rekrutmen tenaga kerja dan pengendalian jumlah tenaga kerja yang dibutuhkan selama proyek berlangsung
3. Struktur dan pembagian tenaga kerja menjadi kelompok kerja
4. Perencanaan, penjadwalan, pengarahan dan pengawasan kegiatan kerja
5. Komposisi tenaga kerja untuk kegiatan tertentu dan jenis pekerjaan untuk setiap tenaga kerja

Selain itu, salah satu masalah yang dihadapi dalam penjadwalan dan perkiraan jumlah tenaga kerja adalah menentukan jumlah dan komposisi pekerja yang wajar dan dapat bekerja dengan efektif (Hassanein dan Melin, 1997). Pernyataan tersebut diperlukan untuk dua tujuan, yaitu:

1. Menentukan durasi pekerjaan, karena ukuran dan bentuk komposisi kelompok kerja mempengaruhi lamanya pekerjaan berlangsung

2. Menentukan estimasi biaya pekerjaan, karena ukuran dan bentuk komposisi kelompok kerja berpengaruh dalam estimasi biaya pekerja dalam suatu pekerjaan

Menurut Russel dan Fayek (1994), masalah kesulitan mencari tenaga kerja (*undermanning*) dan kelebihan tenaga kerja (*overmanning*) kadang terjadi pada suatu proyek. Pihak manajemen sudah seharusnya mengantisipasi masalah ini dengan membuat perencanaan yang lebih baik, karena *undermanning* dan *overmanning* dapat membawa dampak pada biaya tenaga kerja yang akan dikeluarkan.

Menurut Leni, et al (2005), terdapat 9 (sembilan) sumber resiko dalam rekrutmen tenaga kerja yang dapat mempengaruhi biaya tenaga kerja, antara lain:

1. Kurang tepat dalam penempatan tenaga kerja
2. Kesulitan dalam mencari tenaga kerja
3. Tenaga kerja yang tersedia lebih mahal dari yang dianggarkan
4. Kelebihan perekrutan tenaga kerja
5. Keterlambatan dalam penyediaan tenaga kerja
6. Kualitas mandor yang kurang baik
7. Kurang atau tidak adanya pelatihan untuk tenaga kerja
8. Pembayaran tenaga kerja yang lebih ahli dan mahal untuk pekerjaan tertentu
9. Pengetahuan dan pengalaman pekerja kurang dalam menjalankan spesifikasi kerja

2.3 Pengenalan Tandan Kosong Kelapa Sawit (TKKS)

2.3.1. Pengertian TKKS

TKKS merupakan hasil sampingan dari pengolahan minyak kelapa sawit yang pemanfaatannya masih terbatas sebagai pupuk, bahan baku pembuatan matras dan media untuk pertumbuhan jamur dan tanaman (Iriani, 2009). TKKS merupakan hasil samping dari pengolahan Tandan Buah Segar (TBS) menjadi CPO di PKS. Setiap ton TBS diolah dihasilkan TKKS antar 190 – 240 kg atau 19 – 24 % dari TBS diolah. Karena produksi harian TKKS dari suatu PKS cukup banyak maka perlu dikelola dengan baik dan tepat agar tidak menimbulkan masalah baik terhadap operasional PKS maupun terhadap lingkungan (Santobri, 2008).

Di pabrik minyak kelapa sawit, tandan kosong kelapa sawit hanya dibakar dan sekarang telah dilarang karena adanya kekhawatiran pencemaran lingkungan, atau dibuang sehingga menimbulkan keluhan/masalah karena dapat menurunkan kemampuan menyerap air. Di samping itu, tandan kosong kelapa sawit yang membusuk di tempat akan menarik kedatangan jenis kumbang tertentu yang berpotensi merusak pohon kelapa sawit hasil peremajaan di lahan sekitar tempat pembuangan (Roliadi dan Fitriasari, 2011).

2.3.2. Keuntungan Aplikasi TKKS

Biaya pemupukan anorganik sangat mahal yaitu 25 – 30% dari total biaya produksi CPO. Oleh karena itu, kebijakan mensubstitusi sebagian pupuk anorganik dengan pupuk organik yang berasal dari *by product* PKS seperti tandan kosong kelapa sawit dan kompos. Penggunaan *by product* PKS sebagai pupuk organik signifikan dalam mengurangi biaya pemupukan, mempertahankan produksi TBS, peremajaan tanah dan mengurangi polusi lingkungan (Caryeyre dan Demirayak, 2009).

Sebagai alternatif untuk aplikasi langsung dari limbah tandan buah kosong untuk dicerna kebun, kompos organik juga dapat dihasilkan dengan menyemprotkan limbah dicerna ke tandan buah diparut kosong. Setelah proses dekomposisi, tandan buah kosong akan kehilangan setengah beratnya dan 80% dari volume tandan tersebut, namun kandungan gizi akan tetap sama. Alternatif ini memungkinkan transpirasi efisien dan penerapan kompos tandan buah kosong untuk perkebunan yang terletak jauh dari pabrik (Caryeyre dan Demirayak, 2009).

Keunggulan TKKS meliputi : kandungan kalium yang tinggi, tanpa penambahan *starter* dan bahan kimia, memperkaya unsur hara yang ada di dalam tanah, dan mampu memperbaiki sifat fisik, kimia dan biologi. Selain itu menurut Pasaribu (2010), TKKS memiliki beberapa sifat yang menguntungkan antara lain :

1. Memperbaiki struktur tanah berlempung menjadi ringan;
2. Membantu kelarutan unsur-unsur hara yang diperlukan bagi pertumbuhan tanaman;
3. Bersifat homogen dan mengurangi risiko sebagai pembawa hama tanaman;

4. Merupakan pupuk yang tidak mudah tercuci oleh air yang meresap ke dalam tanah; dan
5. Dapat diaplikasikan pada sembarang musim.

Aplikasi TKKS sangat efektif sebagai mulsa. Cara ini dapat menurunkan temperatur tanah, mempertahankan kelembaban tanah dan membantu mengurangi dampak yang kurang baik terhadap pertumbuhan tanaman dan produksi pada saat kemarau. Untuk areal yang curah hujannya tinggi, TKKS secara signifikan dapat mengurangi losses nutrisi melalui proses pencucian dan aliran permukaan atau menjaga terjadinya erosi tanah (Tim Agronomi BGA, 2011).

2.3.3. Kandungan dalam TKKS

TKKS mengandung serat yang tinggi. Kandungan utama TKKS adalah selulosa dan lignin. Selulosa dalam TKKS mencapai 54-60 %, sedangkan kandungan lignin mencapai 22-27 % (Hambali, dkk, 2007).

Tabel 1. Komposisi kimiawi TKKS

Komponen	% Berat Kering
Abu	6,04
Lignin	15,70
Selulosa	36,81
Hemiselulosa	27,01

Sumber : Hambali, dkk, 2007

Unsur hara yang terkandung dalam TKKS sebagaimana terdapat pada tabel 2 berikut ini.

Tabel 2. Unsur hara utama pada TKKS

Unsur Hara	Kadar Hara Dalam TKKS (Kg/HA/Tahun)	Sebanding Dengan Pupuk Per Ton TKKS
N (Nitrogen)	5,4	8,0 kg Urea
P (Phosphorus)	0,4	2,9 kg RP
K (Kalium, Potassium)	35,3	18,3 kg MOP
Mg (Magnesium)	2,7	5,0 kg Kieserit

Sumber: Tim Agronomi BGA (2011)

Aplikasi TKKS sangat sesuai dalam memenuhi atau menggantikan sebagian pupuk anorganik, asalkan jumlah suplai haranya sebanding dengan pupuk anorganik tersebut (Tim Agronomi BGA, 2011). Perbedaan rasio C/N

sangat tinggi sehingga proses dekomposisi dan mineralisasi TKKS di lapang oleh mikroorganisme menjadi relatif lambat.

2.3.4. Dekomposisi dan Aplikasi Tandan Kosong Kelapa Sawit

Tabel 3 menjelaskan keperluan waktu proses penguraian hara TKKS.

Tabel 3. Waktu yang Diperlukan dalam Proses Penguraian Hara

Unsur Hara	Waktu Penguraian
N (Nitrogen)	205 hari
P (Phosphorus)	85 hari
K (Kalium, Potassium)	25 hari
Mg (Magnesium)	115 hari

Sumber: Tim Agronomi BGA (2011)

Dari keempat unsur hara tersebut unsur Kalium (K) yang paling cepat terurai. Oleh sebab itu menjadi pertimbangan begitu pentingnya sesegera mungkin mengaplikasikan TKKS dari PKS ke lapang, sehingga unsur hara yang terkandung di dalamnya dapat dimanfaatkan tanaman secara maksimal. Penguraian N, P, dan Mg yang lambat tidak seluruhnya merugikan. Kondisi tersebut dapat mengurangi resiko kehilangan unsur hara akibat proses penguapan, pencucian dan sebagainya (Tim Agronomi BGA, 2011).

Penempatan TKKS di lapangan dilakukan dengan dua cara yaitu pemberian sebagai mulsa dan aplikasi dalam rorak. Pemberian sebagai mulsa dilakukan dengan menebar TKKS pada gawangan mati dari jalur pokok sampai batas piringan. Aplikasi TKKS sangat efektif pada daerah-daerah dengan topografi bergelombang sampai berbukit (Simangunsong, 2011).

2.4 Model Goal Programming

Program tujuan ganda yang dalam bahasa asingnya dikenal sebagai Goal Programming atau *multiple objective programming* merupakan modifikasi atau variasi khusus dari program linier yang sudah kita kenal. Analisis Goal Programming bertujuan untuk meminimumkan jarak antara atau deviasi terhadap tujuan, target atau sasaran yang telah ditetapkan dengan usaha yang dapat ditempuh untuk mencapai target atau tujuan tersebut secara memuaskan sesuai dengan syarat ikatan yang ada, yang membatasinya berupa sumber daya yang tersedia, teknologi yang ada, kendala tujuan, dan sebagainya (Hartati, 2009).

Dalam keadaan dimana seorang pengambil keputusan dihadapkan pada suatu persoalan yang mengandung beberapa tujuan di dalamnya, maka program linier tidak dapat membantunya untuk memberikan pertimbangan yang rasional. Karena program linier hanya terbatas pada analisis tujuan tunggal (*single objective function*). Oleh karena itu, maka persoalan tersebut memerlukan bantuan program tujuan ganda. Dunia nyata yang kita hadapi ini adalah dunia yang penuh dengan berbagai tujuan sebagai target dan sasaran. Oleh karena itu maka Goal Programming merupakan alat analisis yang tepat untuk itu.

Goal Programming berusaha untuk meminimumkan deviasi berbagai tujuan, sasaran, atau target yang telah ditetapkan. Dengan analisis Goal Programming maka kita mencoba untuk memenuhi target paling tidak mendekati target yang telah kita tentukan menurut skala prioritasnya masing-masing. Goal Programming memuat dua tipe kendala yaitu kendala teknologi (*technological constraint*) yang merupakan permasalahan kapasitas sumber dan kendala lainnya yang bukan terhadap tujuan, kendala tujuan (*goal constraint*) yang mewakili atau menggambarkan target dari objek-objek dalam urutan prioritas (Jones dan Tamiz, 2010).

2.4.1 Konsep Dasar Goal Programming

Berikut ini adalah definisi dari beberapa istilah dan lambang yang biasa digunakan dalam Goal Programming (Jones dan Tamiz, 2010).

1. *Decision variables* (Variabel keputusan)

Merupakan seperangkat variabel yang tidak diketahui (dalam model Linier Goal Programming dilambangkan dengan x_j dimana $j = 1, 2, \dots, n$) yang akan dicari nilainya.

2. *Right hand side values* atau RHS (Nilai sisi kanan)

Merupakan nilai-nilai yang biasanya menunjukkan ketersediaan sumber daya (dilambangkan dengan b_i) yang akan ditentukan kekurangan atau kelebihan penggunaannya.

3. *Goal* (tujuan)

Merupakan keinginan untuk meminimumkan angka penyimpangan dari suatu nilai RHS pada suatu *goal constraint* tertentu.

4. *Goal Constraint* (kendala tujuan)

Merupakan sinonim dari istilah *goal equation*, yakni suatu tujuan yang diekspresikan dalam persamaan matematika yang memasukkan variabel simpangan.

5. *Preemptive priority factor*

Merupakan suatu sistem urutan yang dilambangkan dengan P_k , dimana $k = 1, 2, \dots, k$ dan k menunjukkan banyaknya tujuan dalam model yang memungkinkan tujuan-tujuan disusun secara ordinal dalam model Linier Goal Programming. Sistem urutan itu menempatkan tujuan-tujuan dalam susunan dengan hubungan sebagai berikut:

$$P_1 > P_2 \gg \gg P_k$$

P_1 merupakan tujuan yang paling penting

P_2 merupakan tujuan yang kurang penting dan seterusnya

6. *Deviation variables* (variabel simpangan)

Merupakan variabel-variabel yang menunjukkan kemungkinan penyimpangan negatif dari suatu nilai RHS kendala tujuan (dalam model Linier Goal Programming) dilambangkan dengan d_i^- dimana $i = 1, 2, \dots, m$ dan m adalah banyaknya kendala tujuan dalam model) atau penyimpangan positif dari suatu nilai RHS (dilambangkan dengan d_i^+). Variabel-variabel ini serupa dengan *slack variable* dalam Linier Programming.

7. *Differential weight* (bobot)

Merupakan timbangan matematika yang diekspresikan dengan angka kardinal (dilambangkan dengan w_{ki} dimana $k = 1, 2, \dots, k$ $i = 1, 2, \dots, m$) dan digunakan untuk membedakan variabel simpangan i di dalam suatu tingkat prioritas k .

8. *Technological coefficient* (koefisien teknologi)

Merupakan nilai-nilai numerik (dilambangkan dengan a_{ij}) yang menunjukkan penggunaan nilai b_i per unit untuk menciptakan x_j .

2.4.2 Unsur-Unsur Goal Programming

Goal Programming berusaha meminimumkan deviasi atau simpangan di antara berbagai tujuan atau sasaran yang telah ditetapkan sebagai targetnya, artinya nilai ruas kiri suatu persamaan kendala sebisa mungkin mendekati nilai

ruas kanannya. Jones dan Tamiz (2010) mengatakan bahwa setiap model Goal Programming paling sedikit terdiri dari tiga unsur, yaitu: fungsi tujuan, kendala-kendala tujuan, dan kendala non negatif.

1. Fungsi tujuan

Fungsi tujuan dalam Goal Programming berusaha meminimumkan deviasi atau penyimpangan terhadap nilai RHS pada suatu *goal constraint* tertentu. Ada tiga jenis fungsi tujuan dalam Goal Programming yaitu:

a. Minimize

$$Z = d_t^- + d_t^+ \quad m_i = 1 \quad i = 1, 2, 3, \dots, m \dots\dots\dots(2.1a)$$

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

b. Minimize

$$Z = P_k (d_t^- + d_t^+) \quad m_i = 1 \quad i = 1, 2, 3, \dots, m \dots\dots\dots(2.1b)$$

Fungsi tujuan tersebut digunakan dalam suatu masalah saat urutan tujuan diperlukan tetapi variabel simpangan di dalam setiap tingkat prioritas memiliki kepentingan yang sama.

c. Minimize

$$Z = w_{ki} P_k (d_t^- + d_t^+) \quad m_i = 1 \quad i = 1, 2, 3, \dots, m \dots\dots\dots(2.1c)$$

Fungsi tujuan tersebut akan tergantung pada situasi permasalahannya. Tujuan-tujuan diurutkan dan variabel simpangan pada setiap tingkat prioritas dibedakan dengan menggunakan bobot yang berlainan. Perlu diperhatikan bahwa dalam model Linier Goal Programming tidak ditemukan variabel keputusan pada fungsi tujuan.

Keterangan:

- P_k : *preemptive priority factor* (suatu sistem urutan dengan $k = 1, 2, \dots, k$ menunjukkan banyaknya tujuan dalam model)
- d_t^- dan d_t^+ : variabel deviasi negatif dan positif untuk tujuan dengan i menunjukkan banyaknya kendala tujuan dalam model
- w_{ki} : bobot yang diberikan
- m : jumlah kendala tujuan dalam model



2. Kendala Tujuan

Menurut Daihani (2004), ada enam jenis kendala tujuan yang berlainan. Maksud setiap jenis kendala itu ditentukan oleh hubungannya dengan fungsi tujuan.

Tabel 4. Macam Kendala Tujuan Goal Programming

Kendala Tujuan	Variabel Simpangan dalam Fungsi Tujuan	Kemungkinan Simpangan	Penggunaan Nilai RHS yang Diinginkan
$a_{ij} + d_i = b_i$	d_i	Negatif	$= b_i$
$a_{ij} - d_i^+ = b_i$	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	$= b_i$
$a_{ij} - d_i^+ = b_i$	d_i^+ (artf.)	Tidak ada	Pas $= b_i$

Terlihat bahwa setiap jenis kendala tujuan harus punya satu atau dua variabel simpangan yang ditempatkan pada fungsi tujuan. Dimungkinkan adanya kendala-kendala yang tidak memiliki variabel simpangan. Kendala-kendala ini sama seperti kendala-kendala persamaan linier. Persamaan pertama pada tabel maknanya serupa dengan kendala pertidaksamaan \leq dalam masalah program linier maksimisasi. Persamaan kedua maknanya serupa dengan kendala pertidaksamaan \geq pada masalah program linier minimisasi.

Persamaan ketiga, keempat dan kelima semuanya memperbolehkan penyimpangan dua arah, tetapi persamaan kelima mencari penggunaan sumber daya yang diinginkan sama dengan b_i . Ini serupa dengan kendala persamaan dalam Linier Programming, tetapi tidak menempel pada solusi karena dimungkinkan adanya penyimpangan negatif dan positif. Jika kendala persamaan dianggap perlu dalam perumusan model Linier Goal Programming, ia dapat dimasukkan dengan menempatkan sebuah artificial variabel d_i^+ seperti pada persamaan keenam. Persamaan ketiga dan keempat memperbolehkan adanya penyimpangan negatif dari nilai RHS nya. Dalam kendala Linier Programming tak ada pembandingan unuk persamaan ketiga dan keempat.

3. Kendala Non Negatif

Seperti pada Linier Programming, variabel-variabel model Linier Programming biasanya bernilai lebih besar atau sama dengan nol. Semua model Linier Goal Programming terdiri dari variabel simpangan dan variabel keputusan, sehingga pernyataan non negatif dilambangkan sebagai $x_j, d_i, d_i^+ \geq 0$.

2.5 Formulasi Model Keputusan Fuzzy Goal Programming

2.5.1 Model Fuzzy Goal Programming

Fuzzy Goal Programming bertujuan mencari nilai yang merupakan fungsi objektif yang akan dioptimalkan sedemikian hingga tunduk pada batasan-batasan yang dimodelkan dengan himpunan *fuzzy* (Gunes dan Umarosman, 2005). Model FGP secara umum dapat dituliskan sebagai berikut :

$$Z_k(x) \gtrsim b_k \quad k = 1, 2, 3, \dots, K_1 \dots\dots\dots(2.2a)$$

$$Z_k(x) \lesssim b_k \quad k = 1, 2, 3, \dots, K_2 \dots\dots\dots(2.2b)$$

dengan kendala

$$Ax \leq B \dots\dots\dots(2.2c)$$

$$Ax \geq B \dots\dots\dots(2.2d)$$

$$x \geq 0 \dots\dots\dots(2.2e)$$

Keterangan:

$Z_k(x)$: fungsi tujuan

x : variabel keputusan

b_k : level aspirasi tujuan ke- k

A : koefisien model kendala

B : nilai RHS (*Right Hand Side*)

Tanda \lesssim merupakan bentuk *fuzzy* dari \leq yang menginterpretasikan ‘pada dasarnya kurang dari atau sama dengan’. Demikian pula, tanda \gtrsim merupakan bentuk *fuzzy* dari \geq yang menginterpretasikan ‘pada dasarnya lebih dari atau sama dengan’ (Gunes dan Umarosman, 2005). Tujuan *fuzzy* tipe 1 mempunyai arti bahwa pengambil keputusan akan puas untuk nilai sedikit lebih kecil dari b_k sampai nilai penyimpangan yang ditandai oleh batas toleransi, sedangkan tujuan *fuzzy* tipe 2 mempunyai arti bahwa pengambil keputusan akan puas untuk nilai



sedikit lebih besar dari b_k sampai nilai penyimpangan yang ditandai oleh batas toleransi (Daihani, 2004).

2.5.2 Fungsi Keanggotaan Fuzzy

Menurut Supranto (1998), dengan mendefinisikan batas toleransi yang lebih rendah sesuai dengan tujuan *fuzzy* dan batas atas toleransi yang lebih tinggi sesuai dengan tujuan *fuzzy*, fungsi keanggotaan dinyatakan dalam proses pembuatan keputusan *fuzzy*. Batas toleransi ini tergantung pada pembuat keputusan dan konteks pengambilan keputusan. Fungsi keanggotaan tujuan *fuzzy* ditulis dalam proses pengambilan keputusan *fuzzy* yaitu:

Jika θ_k^+ menjadi deviasi positif untuk pencapaian kepuasan aspirasi b_k dari k tujuan *fuzzy* $Z_k(x) \gtrsim b$, maka fungsi keanggotaan menjadi:

$$\mu_{Z_k}(x) = \begin{cases} 1, & Z_k(x) \geq b_k \\ 1 - \frac{Z_k(x) - (b_k - \theta_k^+)}{\theta_k^+}, & b_k - \theta_k^+ \leq Z_k(x) < b_k \\ 0, & Z_k(x) < b_k - \theta_k^+ \end{cases} \dots\dots\dots(2.3a)$$

Jika θ_k^- menjadi deviasi negatif untuk pencapaian kepuasan aspirasi b_k dari k tujuan *fuzzy* $Z_k(x) \lesssim b$, maka fungsi keanggotaan menjadi:

$$\mu_{Z_k}(x) = \begin{cases} 1, & Z_k(x) \leq b_k \\ 1 - \frac{(b_k + \theta_k^-) - Z_k(x)}{\theta_k^-}, & b_k \leq Z_k(x) < b_k + \theta_k^- \\ 0, & Z_k(x) > b_k + \theta_k^- \end{cases} \dots\dots\dots(2.3b)$$

- $\mu_{Z_k}(x)$: fungsi keanggotaan dari himpunan *fuzzy*
- θ_k^+ dan θ_k^- : deviasi positif dan negatif dari nilai RHS masing-masing goal

Pada bagian ini akan disajikan model FGP berdasarkan pendekatan Hannan. Seperti disebutkan sebelumnya, model FGP sesuai dengan masalah pengambilan keputusan mungkin memiliki tujuan tipe $Z_k(x) \gtrsim b_k$ dan $Z_k(x) \lesssim b_k$.

Jika goal tipe pertama benar-benar tercapai, maka nilai θ_k^- akan nol serta nilai keanggotaan untuk tujuan sama dengan 1. Di sisi lain jika goal tidak benar-benar tercapai, maka θ_k^- akan bernilai tak nol dan nilai keanggotaan akan menjadi $[0, 1]$. Demikian juga jika goal tipe kedua benar-benar tercapai, maka nilai θ_k^+ akan bernilai nol, serta nilai keanggotaan untuk goal sama dengan 1. Di sisi lain



jika goal tidak tercapai, maka θ_k^+ akan bernilai tak nol dan nilai keanggotaan akan menjadi $[0, 1]$.

Menurut Hannan (1982), dalam pencapaian goal digunakan λ yang didefinisikan sebagai variabel tingkat pencapaian/kepuasan. Variabel λ berlaku untuk mewakili tingkat pencapaian goal secara keseluruhan. Adapun tingkat pencapaian masing-masing goal untuk dua goal dapat ditunjukkan dengan variabel λ_1 dan λ_2 .

Memaksimalkan nilai keanggotaan dirumuskan dengan menggunakan satu variabel tingkat pencapaian/kepuasan:

Maximize λ dengan kendala

$$\frac{Z_k(x)}{t_k} - \theta_k^+ + \theta_k^- = \frac{b_k}{t_k} \dots \dots \dots (2.4a)$$

$$\frac{(Ax)_k + (L_Q - \sigma_k)}{\sigma_k} \geq \lambda \dots \dots \dots (2.4b)$$

$$\frac{(B_k + \sigma_k) + (Ax)_k}{\sigma_k} \geq \lambda \dots \dots \dots (2.4c)$$

$$\lambda + \theta_1^+ + \theta_1^- \leq 1$$

$$\lambda + \theta_2^+ + \theta_2^- \leq 1$$

$$0 \leq \lambda \leq 1$$

$$\theta_1^-, \theta_2^-, \theta_1^+, \theta_2^+ \geq 0$$

A : koefisien dari kendala

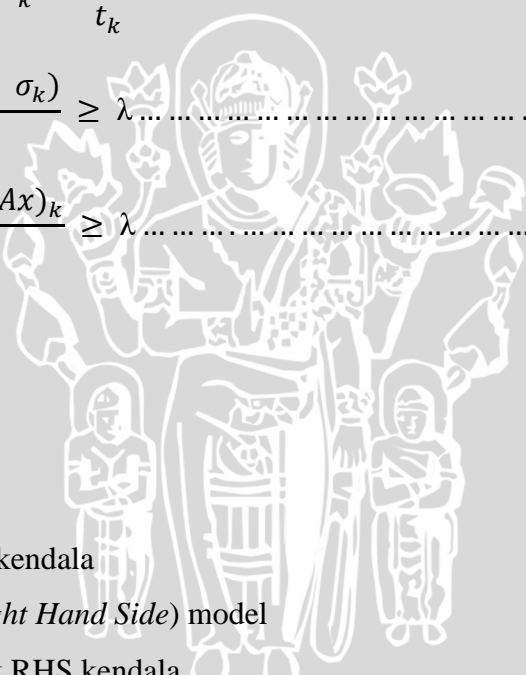
B_k : nilai RHS (*Right Hand Side*) model

σ_k : toleransi untuk RHS kendala

t_k : toleransi

θ_k^+ : variabel deviasi positif

θ_k^- : variabel deviasi negative



2.6 Pengenalan Program POM for Windows

Program POM for Windows merupakan paket program computer untuk menyelesaikan persoalan-persoalan metode kuantitatif, manajemen sains atau riset operasi. POM for Windows merupakan gabungan dari program terdahulu DS dan QM for Windows, jadi jika dibandingkan dengan program QM for Windows modul-modul yang tersedia di POM for Windows lebih banyak. Namun ada modul-modul yang hanya tersedia di program POM for Windows, atau hanya tersedia di program DS for Windows dan tidak tersedia di QM for Windows.

Program-program QM for Windows, DS dan POM for Windows, disediakan oleh penerbit Prentice Hall (www.prentice-hall.com), dan sebagian program merupakan bawaan dari beberapa buku terbitan Prentice Hall. Tampilan sementara (*splash*) setelah program POM for Windows dijalankan dapat dilihat pada lampiran 2.

Setelah tampilan sementara berakhir, akan muncul tampilan awal seperti pada lampiran 2 yang berarti program sudah siap untuk menjalankan modul-modul yang akan dipilih. Pilihan modul ada pada menu *Module* yang dapat diaktifkan dengan meng-klik tulisan *Module* di baris menu atau dengan menekan tombol Alt+M. modul-modul dari *Assignment* (metode penugasan) hingga *Waiting Lines* (antrian) dapat dipilih, disesuaikan dengan persoalan yang hendak diselesaikan.

Baris tool (*tool bar*) dan baris utilitas (*utility bar*) dapat diatur sesuai dengan kebutuhan dengan cara meng-klik kanan *mouse*, ketika kursor *mouse* berada pada *toolbar*. *Toolbar* dan *utility bar* dapat juga dipindahkan tempatnya dengan cara men-*drag & drop* bagian paling kiri dari *toolbar* atau *utility bar* tersebut atau dengan menu *View – Toolbar – Customize*.