BAB II

KAJIAN PUSTAKA

A. Perawat

Perawat merupakan salah satu tenaga medis penting yang memberikan pelayanan untuk membantu kesembuhan pasien di rumah sakit, oleh karena itu peran perawat sangatlah dibutuhkan dan memiliki nilai kepentingan tersendiri. Berdasarkan Peraturan Menteri Kesehatan Republik Indonesia No. HK. 02.02/MENKES/148/1/2010 tentang Izin dan Penyelenggaraan Praktik Perawat, definisi perawat adalah seseorang yang telah lulus pendidikan perawat baik di dalam maupun di luar negeri sesuai dengan peraturan perundang-undangan. Seorang perawat dituntut untuk meningkatkan kualitas pelayanan kesehatan kepada masyarakat (Selvia, 2013).

Menurut Wijaya (2005) bahwa perawat bertanggung jawab meningkatkan kesehatan, pencegahan penyakit, penyembuhan, pemulihan serta pemeliharaan kesehatan dengan penekanan kepada upaya pelayanan kesehatan utama sesuai wewenang, tanggung jawab dan etika profesi keperawatan. Dalam memberikan pelayanan kesehatan perawat dituntut untuk lebih profesional agar kualitas pelayanan kesehatan yang diberikan semakin meningkat. Dalam pernyataan Alimul (2002) dalam Selvia (2013:17) bahwa di dalam etika keperawatan terdapat beberapa unsur yang terkandung di dalamnya antara lain pengorbanan, dedikasi, pengabdian dan hubungan antara perawat dengan pasien, dokter, sejawat maupun diri sendiri.

B. *Shift* Kerja

Waktu kerja berlebih dan *shift* kerja sering diterapkan oleh suatu instansi atau perusahaan untuk meningkatkan produktivitas kerja dan memaksimalkan sumber daya yang ada. *Shift* kerja memiliki perbedaan dengan hari kerja biasa, yaitu pada hari kerja biasa terdapat suatu waktu yang rutin dan berlaku untuk semua pekerja, sementara *shift* kerja memiliki dua atau lebih waktu kerja yang terbagi di dalam 24 jam kerja dan setiap pekerja dialokasikan pada waktu kerja tertentu. Biasanya perusahaan yang berjalan secara kontinu yaitu misalnya menerapkan jam kerja 24 jam per hari adalah perusahaan yang menerapkan aturan *shift* kerja ini.

1. Definisi *Shift* Kerja

Shift kerja mempunyai berbagai definisi tetapi biasanya *shift* kerja disamakan dengan pekerjaan yang dibentuk di luar jam kerja biasa (08.00-17.00). Ciri khas tersebut adalah kontinuitas, pergantian dan jadwal kerja khusus. Secara umum yang dimaksud dengan *shift* kerja adalah semua pengaturan jam kerja, sebagai pengganti atau tambahan kerja siang hari sebagaimana yang biasa dilakukan. Namun demikian ada pula definisi yang lebih operasional dengan menyebutkan jenis *shift* kerja tersebut. *Shift* kerja disebutkan sebagai pekerjaan yang secara permanen atau sering pada jam kerja yang tidak teratur (Kuswadji,1997).

2. Karakteristik dan Sistem *Shift* Kerja

Menurut Knauth (1988) terdapat 5 faktor *shift* kerja yaitu jenis *shift* (pagi, siang, malam), panjang waktu tiap *shift*, waktu di mulai dan berakhir satu *shift*,

distribusi waktu istirahat, dan arah transisi *shift*. Coleman (1995) membagi *shift* kerja menjadi 6 bentuk dasar :

a. *Fixed shifts* (*straight shift*)

   Setiap karyawan sudah mempunyai jam kerja tetap dan tidak dapat diubah.

b. *Rotating shifts*

   Karyawan secara bergiliran bekerja pada *shift* yang telah diatur.

c. *Oscilatting shifts*

   Satu kelompok karyawan mempunyai *shift* tetap dan kelompok sisa rotasi.

d. *Primary shifts*

   Setiap karyawan mempunyai *shift* tetap tetapi dapat dipindah sementara.

e. *Staggeret Shifts*

   Shift tetap dengan nomor waktu mulai dan nomor karyawan.

f. *Mixed Shifts*

   Gabungan beberapa *shift* untuk pekerja dalam bagian yang sama.

Sistem *shift* kerja dapat berbeda antar instansi atau perusahaan, ada yang menggunakan tiga *shift* setiap hari dengan masing-masing delapan jam kerja ada pula yang menggunakan dua *shift* dengan masing-masing dua belas jam kerja. Ada beberapa macam sistem *shift* kerja yang terdiri dari :

a. *Shift* permanen

   Tenaga kerja yang bekerja pada *shift* yang tetap setiap harinya.

b. Sistem Rotasi

Tenaga kerja tidak selamanya ditempatkan pada *shift* yang tetap. *Shift* rotasi adalah sistem yang paling mengganggu terhadap irama sirkadian

dibandingkan dengan *shift* permanen bila berlangsung dalam jangka waktu yang panjang (Ramayuli, 2004).

Menurut Kroemer (2005) *shift* kerja terdiri dari pagi, siang, malam dan setiap bagian mempunyai kelebihan dan kekurangan. Contohnya pada *shift* pagi, terdapat seorang ibu yang harus mengantarkan anak-anaknya bersekolah atau mempunyai jarak rumah yang jauh dari tempat kerja. *Shift* malam tidak cocok bagi pekerja yang tidak terbiasa bekerja pada malam hari sehingga dapat menurunkan kualitas pelayanan. Oleh karena itu, diperlukan jadwal yang baik dan cocok bagi para pekerja untuk memaksimalkan kinerja dan kualitas pelayanan.

Pembagian berikutnya adalah sistem *shift* terputus yakni berlangsung antara hari senin sampai dengan hari sabtu. Kemudian sistem *shift* kerja yang terus-menerus berlangsung selama 7 hari seminggu termasuk hari-hari libur. Pembagian sistem *shift* kerja lainnya adalah: jumlah hari kerja malam yang berturut-turut, awal dan akhir kerja *shift*, jangka waktu masing-masing *shift*, urutan rotasi *shift*, jangka daur *shift* dan keteraturan sistem *shift* (Kuswadji, 1997).

C. Pembagian Waktu Sistem *Shift* Kerja

Menurut jumlah hari kerja malam yang berturut-turut paling sedikit ada tiga jenis:

1. *Continental Rota*

Sistem *continental rota* di negara Eropa sering dipakai dan dijadikan rekomendasi untuk *shift* kerja. Pada sistem ini pekerja bekerja menurut giliran

2-2-3 (pagi, pagi, siang, siang, malam, malam, malam, libur, libur). Pada sistem ini hari libur sabtu dan minggu akan terjadi setiap 4 minggu (Kroemer, 2005).

2. *Metropolitan Rota*

Pada sistem ini pekerja bekerja menurut giliran 2-2-2 (pagi, pagi,siang, siang, malam, malam, libur). Sistem ini hari libur sabtu dan minggu hanya terjadi sekali dalam 8 minggu. Itu saja terjadinya pada minggu ke-8 (Kroemer, 2005).

Menurut awal dan akhir jam kerja *shift*, lama satu *shift*, dan keteraturannya sistem *shift* dapat di bagi menjadi tiga (Kuswadji, 1997) yaitu:

a. Sistem 3 *shift* biasa

Pada sistem ini masing-masing pekerja akan mengalami 8 jam kerja yang sama selama 24 jam: dinas pagi antara pukul 6-14, dinas sore antara pukul 14-22 dan dinas malam antara pukul 22-6. Dinas pagi memungkinkan keluarga dapat berkumpul bersama pada malam harinya. Bila dinas pagi dimulai terlalu pagi misalnya pukul 4, akan sangat melelahkan dan tidur malam menjadi lebih singkat. Dinas sore sangat tidak baik untuk kehidupan sosial, namun sebaliknya untuk tidur sangat menguntungkan. Dinas malam lebih berdampak buruk dibandingkan dinas pagi dan sore, karena dinas malam dapat mengganggu tidur akibat berbagai sebab: bising di siang hari, tidur terputus karena harus makan siang, tidur terus sampai sore. Akibatnya meraka mengalami kelelahan karena tidur yang tidak pulas.

b. Sistem Amerika

Menurut sistem ini dinas pagi mulai pukul 8-16, dinas sore antara pukul 16-24 dan dinas malam antara pukul 24-8. Sistem ini memberikan keuntungan fisiologik dan sosial, kesempatan tidur akan banyak terutama pada pagi dan sore. Setiap *shift* akan mengalami makan bersama keluarga paling sedikit sekali dalam seminggu.

c. Sistem 12 -12

Sistem ini dipakai untuk penambangan minyak lepas pantai. Selama 12 jam dinas pagi dan selama 12 jam dinas malam. Jadwal antara 7-19 dan 19-7. Satu minggu kerja siang dan satu minggu kerja malam pisah dengan keluarga. Setelah dinas 2 minggu, biasanya pulang kerumah dan tinggal bersama keluarga dipandang dari sudut kesehatan atau ergonomi bekerja menurut cara demikian tidak baik. Namun beberapa perkecualian dapat dilakukan, misalnya bila pekerjaan ini tidak terlalu berat. Bila pekerjaan *shift* dilakukan selama ini, masing-masing *shift* baik siang atau malam, harus diikuti dengan istirahat dua hari.

Menurut *International Labour Organization* 1983 sistem *shift* kerja terbagi:

1. Sistem 3 *shift* 4 kelompok (4x8 *hours continous shift* work), yaitu 3 kelompok *shift* bekerja setiap 8 jam dan 1 kelompok istirahat. Sistem ini digunakan pada aktivitas terus menerus tanpa hari libur. Rotasi *shift* 2-3 hari.

2. Sistem 3 *shift* 3 kelompok (4x8 *hours semi continous shift* work), yaitu kelompok *shift* bekerja setiap 8 jam, pada akhir minggu libur. Rotasi *shift* 5 hari.

D. Efek *Shift* Kerja

Shift kerja mempunyai dampak bagi pekerja (Coleman, 1995) yaitu:

1. *Job Performance*

Perubahan jadwal *shift* kerja yang terus-menerus dilakukan dapat menyebabkan pekerja harus terus beradaptasi dengan perubahan tersebut.

2. *Job related Attitude*

Karyawan yang bekerja pada *shift* malam sering menunjukkan sikap dan emosi karena stres yang menumpuk atau keletihan fisik.

3. *Personal Health*

Pekerjaan yang menggunakan sistem *shift* dapat mengganggu kesehatan secara fisik dan mental, karena perbedaan situasi dan kondisi yang terjadi di lapangan dalam waktu kerja yang berbeda. Pekerja harus beradaptasi setiap kali bekerja di *shift* yang berbeda.

4. *Social and Domestic Factors*

Pembagian *shift* kerja dapat menyebabkan kehidupan di luar kerja pekerja seperti bersosialisasi, aktivitas bersama keluarga dan sanak saudara, hingga kegiatan religius akan terganggu dan sulit untuk diatur. Biasanya, pekerja akan mengorbankan sedikit atau beberapa kegiatan di luar kerja bila terpaksa.

Menurut Fish yang dikutip oleh (Firdaus, 2005) Efek *shift* kerja yang dapat dirasakan tenaga kerja yaitu :

1. Efek fisiologis, berpengaruh terhadap :
    a. Kualitas tidur perlu dijaga untuk menebus kurang tidur akibat kerja malam
    b. Kapasitas fisik kerja yang menurun akibatnya perasaan mengantuk dan lelah
    c. Menurunnya nafsu makan dan gangguan pencernaan
2. Efek psikososial

    Efek ini menunjukkan masalah lebih besar seperti gangguan kehidupan keluarga, hilangnya waktu luang, kecil kesempatan untuk berinteraksi dengan teman, menganggu aktivitas kelompok dalam masyarakat. Demikian pula adanya pandangan di suatau daerah membenarkan wanita bekerja pada malam hari mengakibatkan tersisih dari masyarakat.
3. Efek Kinerja

    Dalam melakukan *shift* kerja malam dapat mengakibatkan terjadinya penurunan kinerja dari pekerja. Hal ini karena dipengaruhi dari efek fisiologisdan psikososial.
4. Efek terhadap kesehatan

    Efek *shift* kerja dapat mengakibatkan gangguan sistem pencernaan seperti *dyspepsia* atau *ulcus ventriculi* di mana masalah ini kritis pada umur 40-45 tahun. Selain itu efek *shift* kerja terhadap kesehatan adalah keseimbangan kadar gula dalam darah dengan insulin pada penderita diabetes.
5. Efek terhadap keselamatan kerja

E. *Shift* Kerja dan Irama Tubuh (*Circadian Rhytm)*

Tubuh memiliki irama sirkadian, yaitu fungsi tubuh yang mengatur siklus tidur dan bangun dalam keseharian manusia. Siklus tersebut akan terganggu dan mengakibatkan jam tidur dan bangun pada setiap orang berbeda-beda apabila irama tersebut mengalami pergeseran waktu.

Menurut beberapa penelitian terjadi pergeseran irama sirkadian antara *onset* waktu tidur regular dengan waktu tidur *irregular* atau *bringing* irama sirkadian (Japardi, 2002).

Menurut Kuswadji (1997) masing-masing orang mempunyai jam biologis sendiri-sendiri, kehidupan mereka diatur menjadi sama dan seragam dalam daur hidup 24 jam sehari. Pengaturan itu dilakukan oleh penangguh waktu yang ada di luar tubuh seperti:

a. Perubahan antara gelap dan terang
b. Kontak sosial
c. Jadwal Kerja
d. Adanya Jam weker

Fungsi tubuh yang sangat dipengaruhi oleh *circadian rhythm* adalah pola tidur, kesiapan bekerja, beberapa fungsi otonom, proses metabolisme, suhu tubuh, denyut jantung dan tekanan darah yang setiap siang hari meningkat dan pada malam hari menurun. Aktivitas tubuh ini dikenal dengan *circadian rhythm*. Pola aktivitas tubuh akan terganggu bila bekerja malam dan maksimal terjadi pada *shift* malam.

Menurut Kodrat (2009: 30) fungsi fisiologis tubuh seperti denyut jantung, oksigen yang dikonsumsi, suhu tubuh, tekanan darah, produksi adrenalin, sekresi urin, kapasitas fisik dan mental secara nyata iramanya berubah dalam waktu 24 jam. Fungsi tubuh tidak dapat dicapai maksimum atau minimum pada waktu yang sama. Umumya semua fungsi tubuh meningkat pada siang hari, mulai melemah pada sore hari dan menurun pada malam hari untuk pemulihan dan pembaharuan. Fenomena ini disebut dengan irama kehidupan.

Menurut beberapa peneliti menyatakan bahwa *shift* kerja dapat mempengaruhi irama sirkadian tubuh. Hal ini dapat dilihat dari waktu pembagian *shift* kerja ada yang pagi, siang, malam, dan *shift* kerja malam yang paling berpengaruh terhadap irama sirkadian dan kesehatan tubuh.

Menurut Kuswadji (1997) menyatakan bahwa 60% - 80% pekerja *shift* akan mengalami gangguan tidur. Pekerja yang melakukan *shift* kerja satu kali saja maka secara bertahap circadian rhytms akan kembali seperti semula, namun bila *shift* kerja dilakukan menetap circadian rhytms tidak akan kembali ke irama semula. Akibatnya pola tidur terganggu (Wijaya, 2005).

F. Optimisasi

Optimisasi adalah suatu pendekatan untuk mengidentifikasikan penyelesaian terbaik dalam pengambilan keputusan dari suatu permasalahan. Penyelesaian permasalahan dalam optimisasi ditujukan untuk memperoleh titik maksimum atau titik minimum dari suatu fungsi yang dioptimalkan. Contohnya seperti permasalahan suatu pabrik rumah tangga dalam menentukan jumlah produksi agar keuntungan maksimum dengan biaya minimum dapat diperoleh.

Dalam optimisasi, suatu permasalahan akan diselesaikan untuk mendapatkan hasil yang optimal sesuai dengan batasan yang ada. Jika permasalahan diformulasikan secara tepat, maka dapat memberikan nilai peubah keputusan yang optimal. Setelah solusi optimal diperoleh, permasalahan sering dievaluasi kembali pada kondisi yang berbeda untuk memperoleh penyelesaian yang baru (Rio Armindo, 2006). Tujuan dari optimisasi adalah untuk meminimumkan usaha yang diperlukan untuk biaya operasional dan memaksimalkan hasil yang ingin diperoleh. Jika hasil yang diinginkan dapat dinyatakan sebagai fungsi dari peubah keputusan, maka optimisasi dapat diasumsikan sebagai proses pencapaian kondisi maksimum atau minimum dari fungsi tersebut. Komponen penting dari permasalahan optimum adalah fungsi tujuan, yang dalam beberapa hal sangat tergantung pada peubah. Dalam penelitian operasional, optimisasi sering dikaitkan sebagai maksimisasi atau minimisasi pemecahan suatu masalah.

Harjiyanto (2014) menyebutkan bahwa penyelesaian masalah optimisasi dengan program matematika dapat dilakukan melalui program linear, proram non-linear, program integer, dan program dinamik. Fungsi tujuan secara umum merupakan langkah minimisasi biaya atau penggunaan bahan-bahan baku, dan sebagainya. Penentuan fungsi tujuan dikaitkan dengan permasalahan yang dihadapi. Penentuan kondisi optimum dikenal sebagai pemrograman teknik matematik. Rio Armindo (2006), menyebutkan bahwa tujuan dan kendala-kendala dalam pemrograman matematik diberikan dalam bentuk fungsi-fungsi matematika dan hubungan fungsional (hubungan keterkaitan). Hubungan

keterkaitan tersebut dapat diartikan sebagai hubungan yang saling mempengaruhi, hubungan interaksi, timbal-balik, dan saling menunjang.

Teknik optimisasi dalam penelitian operasional merupakan pendekatan ilmiah dalam memecahkan masalah-masalah operasi pengolahan (Harjiyanto, 2014). Penerapan teknik ini menyangkut pembentukan deskripsi matematis atau pembentukan model keputusan. Analisa kepekaan teknik ini dapat menganalisa hubungan yang menyatakan akibat-akibat yang mungkin terjadi di masa mendatang sebagai keputusan yang telah diambil. Rio Armindo (2006), menjelaskan bahwa penyelesaian optimisasi dengan program matematika dapat dilakukan melalui *linear programming*, *non-linear programming*, dan *dynamic programming*.

G. Pemrograman Linear

1. Pengertian Pemrograman Linear

Pemrograman linear merupakan salah satu teknik penelitian operasional yang digunakan untuk menyelesaikan masalah optimisasi suatu model linear dengan keterbatasan-keterbatasan sumber daya yang tersedia. Pemrograman linear mengalami perkembangan pesat setelah masa perang dunia karena banyak industri yang menggunakannya (Harjiyanto, 2014). George B. Dantzig menemukan metode simpleks pada tahun 1947, sedangkan John Von Neumann menemukan teori dualitasnya di tahun yang sama. Selanjutnya, berbagai alat dan metode dikembangkan untuk menyelesaikan masalah program linear bahkan sampai pada masalah penelitian operasional hingga tahun 1950-an

seperti pemrograman dinamik, teori antrian, teori persandian, dan *goal programming*. Pemrograman linear merupakan dasar dari *goal programming*.

Pemrograman linear sering kali digunakan untuk menyelesaikan masalah optimisasi di dalam bidang industri, perbankan, pendidikan, dan masalah lain yang dapat dinyatakan dalam bentuk linear. Secara umum, fungsi pada pemrograman linear ada dua macam yaitu fungsi tujuan dan fungsi kendala. Fungsi tujuan digunakan untuk menentukan nilai optimal dari fungsi tersebut, yaitu nilai maksimal untuk masalah keuntungan dan nilai minimal untuk masalah biaya. Fungsi kendala diperlukan dikarenakan adanya keterbatasan sumber daya yang tersedia. Tujuan utama dari pemrograman linear adalah menentukan nilai optimal (maksimal atau minimal) dari suatu fungsi yang telah ditetapkan, untuk menyelesaikan permasalahannya, yaitu dengan metode grafik, dan metode simpleks.

2. Model Pemrograman Linear

Model pemrograman linear adalah model matematis perumusan masalah umum untuk mengalokasian sumber daya pada berbagai kegiatan. Model ini merupakan bentuk dan susunan dalam penyajian masalah-masalah yang akan dipecahkan dengan teknik pemrograman linear. Bentuk umum dari model pemrograman linear adalah sebagai berikut:

Memaksimumkan atau meminimumkan

$$Z = \sum_{j=1}^{n} C_j x_j \qquad (2.1)$$

dengan kendala:

$$\sum_{i=1}^{m} a_{ij}\, x_j \{<,=,>\} b_i \text{ untuk } i = 1,2,3,\ldots,m \qquad (2.2)$$

$$x_j \geq 0 \text{ untuk } j = 1,2,3,\ldots,n \qquad (2.3)$$

Keterangan:

Z	: nilai fungsi tujuan
C_j	: sumbangan per unit kegiatan, untuk masalah maksimisasi C_j menunjukkan keuntungan atau penerimaan per unit, sementara dalam kasus minimisasi menunjukkan biaya per unit.
x_j	: banyaknya kegiatan j, dengan j = 1, 2, 3, …, n
a_{ij}	: banyaknya sumberdaya i dengan konsumsi kegiatan j
b_i	: jumlah sumberdaya i (i = 1, 2, …, m).

Persamaan (2.1) dinamakan fungsi tujuan, yaitu fungsi matematis dari variabel-variabel keputusan yang menunjukkan hubungan dengan nilai sisi kanannya. Persamaan (2.2) dinamakan kendala utama, yaitu fungsi matematis dari variabel-variabel deviasi yang menyatakan kombinasi sebuah tujuan. Persamaan (2.3) dinamakan kendala non negatif, yaitu tujuan yang tidak boleh dilanggar dengan pengertian mempunyai nilai nol untuk penyimpangan positif dan atau negatif.

Bentuk umum persamaan pemrograman linear di atas belum tentu cocok untuk menjabarkan semua permasalahan yang ada. Oleh karena itu, Juanawati Marpaung (2009) juga mengatakan bahwa bentuk-bentuk berikut juga termasuk dalam bentuk umum pemrograman linear. Masalah tersebut antara lain:

a. Masalah meminimalkan, permasalahan untuk menentukan kombinasi (*output*) yang dapat meminimalkan fungsi tujuan (misal: biaya). Dalam hal ini, fungsi tujuan dinyatakan sebagai berikut:

Meminimalkan : $Z = C_1x_1 + C_2x_2 + \ldots + C_nx_n$

b. Masalah fungsi kendala fungsional yang memiliki tanda matematis ≥ ; sehingga apabila dirumuskan terlihat sebagai berikut:

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \geq b_i$$

c. Masalah fungsi kendala fungsional yang memiliki tanda matematis = ; sehingga apabila dirumuskan sebagai berikut:

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n = b_i$$

d. Masalah tertentu, yaitu fungsi kendala non-negatif tidak diperlukan atau dengan kata lain x_j tidak terbatas.

3. Penyelesaian Masalah Pemrograman Linear

Berdasarkan uraian tentang pemrograman linear sebelumnya, penyelesaian pemrograman linear dapat diselesaikan dengan metode simpleks.

a. Pengertian Metode Simpleks

Metode simpleks pertama kali diperkenalkan oleh George B. Dantzig pada tahun 1947 dan telah diperbaiki oleh para ahli lain. Metode ini menyelesaikan masalah program linear melalui perhitungan ulang atau iterasi di mana langkah-langkah perhitungan yang sama diulang berkali-kali sampai solusi optimal dicapai (Harjiyanto, 2014). Metode simpleks adalah suatu metode untuk menyelesaikan masalah pemrograman linear dengan prosedur berulang

yang bergerak dari satu jawaban layak basis ke jawaban layak basis berikutnya sedemikian rupa sehingga harga fungsi tujuan terus naik (dalam permasalahan memaksimalkan). Proses ini akan berkelanjutan sampai dicapainya jawaban optimal, jika ada, yang memberi harga maksimal atau biaya minimal.

Metode simpleks digunakan untuk memecahkan masalah pada pemrograman linear yang terdiri dari tiga variabel atau lebih, sehingga tidak dapat diselesaikan dengan menggunakan metode grafik karena terlalu rumit untuk diselesaikan. Pengertian metode simpleks menurut Heizer Jay dan Rander Barry (2005: 674) mengemukakan bahwa: "Metode simpleks sebenarnya adalah algoritma (atau serangkaian perintah) yang digunakan untuk menguji titik sudutnya secara metodik sehingga sampai pada solusi terbaik dengan keuntungan tertinggi atau biaya terendah."

Ada tiga ciri dari solusi simpleks dari suatu bentuk umum pemrograman linear, di antaranya adalah sebagai berikut (Harjiyanto, 2014):

1) Semua kendala harus berada dalam bentuk persamaan dengan nilai ruas kanan tidak negatif.
2) Semua variabel yang tidak terlibat bernilai negatif.
3) Fungsitujuan dapat berupa memaksimalkan maupun meminimalkan.

Dalam masalah pemrograman linear dengan kendala terlebih dahulu diubah menjadi bentuk kanonik. Bentuk kanonik adalah bentuk sistem persamaan linear dan memuat variabel basis (variabel yang memiliki koefisien 1). Untuk membentuk kendala menjadi bentuk kanonik diperlukan

penambahan variabel basis baru. Variabel basis baru tersebut adalah sebagai berikut (Harjiyanto, 2014):

1) Variabel *slack*, yaitu variabel yang dibutuhkan pada fungsi kendala yang memuat hubungan kurang dari atau sama dengan.

   Contoh:

   $2X_1+3X_2 \leq 10$ diubah menjadi $2X_1+3X_2+S_1 = 10$

   Sehingga S_1 menjadi variabel basis baru.

2) Variabel surplus, yaitu variabel yang ditambahkan pada fungsi kendala yang memuat hubungan lebih dari atau sama dengan.

   Contoh:

   $2X_1+3X_2 \geq 10$ diubah menjadi $2X_1+3X_2 = 10+t_1$

   atau $2X_1+3X_2 - t_1 = 10$

   Variabel t_1 bukan variabel basis (ketika ruas kiri koefisiennya bukan +1)

3) Variabel artifisial, yaitu variabel yang ditambahkan pada fungsi kendala yang belum memuat variabel basis pada poin kedua. Contoh:

   $2X_1+3X_2 - t_1 = 10$ perlu ditambahkan variabel artifisial $u \geq 0$ sehingga menjadi $2X_1+3X_2 - t_1+u_1 = 10$

b. Tabel Simpleks

Misalkan saja sebuah permasalahan pemrogram linear mempunyai tujuan memaksimalkan dengan beberapa kendala. Bentuk umum formulasinya adalah sebagai berikut:

Memaksimumkan: $Z = C_1x_1 + C_2x_2 + \cdots + C_nx_n$

dengan kendala,

$$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n + s_1 \leq b_1$$

$$a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n + s_2 \leq b_2$$

. . .

. . .

. . .

$$a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n + s_m \leq b_m$$

$$x_1, x_2, \ldots, x_n \geq 0$$

Apabila bentuk umum tersebut dimasukkan ke dalam tabel, akan diperoleh bentuk baku tabel simpleks.

Tabel 2.1. Tabel Simpleks.

	C_j	C_1	C_2	…	C_n	0	0	…	0	b_i	R_i
$\bar{C}_i$	$\bar{x}_i/x$	x_1	x_2	…	x_n	S_1	S_2	…	S_m		
0	S_1	a_{11}	a_{12}	…	a_{1n}	1	0	…	0	b_1	R_1
0	S_2	a_{21}	a_{22}	…	a_{2n}	0	1	…	0	b_2	R_2
0	S_m	a_{m1}	a_{m2}	…	a_{mn}	0	0	…	1	b_m	R_m
	Z_j	Z_1	Z_2	…	Z_n	C_1	C_2	…	C_n	Z	
	$Z_j - C_j$	$Z_1 - C_1$	$Z_2 - C_2$	…	$Z_n - C_n$	0	0	…	0	Z	

Keterangan:

x_i = variabel-variabel keputusan
a_{ij} = koefisien teknis
b_i = suku tetap (tak negatif)
C_j = koefisien ongkos
$\bar{x}_i$ = variabel yang menjadi basis dalam tabel yang ditinjau

$\bar{C}_i$ = koefisien ongkos milik variabel basis $\bar{x}_i$
Z_j = jumlah hasilkali dari $\bar{C}_i$ dengan kolom a_{ij}
Z = jumlah hasilkali dari $\bar{C}_i$ dengan kolom b_i
$Z_j - C_j$ = selisih dari Z_j dengan C_j
R_i = rasio antara b_i dengan C_j

Tahapan dalam penyelesaian pemrogram linear dengan menggunakan metode simpleks meliputi beberapa langkah berikut (Subagyo, dkk., 1995:34) dalam Harjiyanto (2014:22):

1) Mengubah fungsi tujuan dan kendala. Dalam hal ini fungsi tujuan diubah menjadi fungsi implisit, artinya semua $C_j x_{ij}$ digeser ke kiri.
2) Tabulasikan persamaan-persamaan yang diperoleh pada langkah pengubahan kendala. Setelah mengubah formulasi kemudian memasukkan variabel atau bilangan ke dalam sebuah tabel dan nilai variabel *slack* = , seperti pada Tabel 2.1.
3) Menentukan variabel masuk (*entering variable*).

   Memilih kolom kunci yang mempunyai nilai pada garis fungsi tujuan yang bernilai negatif dengan angka terbesar.
4) Menentukan variabel keluar (*leaving variable*).

   Memilih baris kunci, yaitu dengan mencari indeks tiap-tiap baris dengan cara membagi nilai-nilai pada kolom ruas kanan (RHS) dengan nilai sebaris pada kolom kunci.
5) Menentukan persamaan pivot baru.

   Mengubah nilai-nilai baris kunci, yaitu dengan cara membaginya dengan angka kunci.
6) Menentukan persamaan-persamaan baru selain persamaan pivot baru.

Baris baru = (Lawan koefisien pada kolom kunci x Nilai baru pada garis pivot) + baris lama.

7) Lanjutkan perbaikan.

Ulangi langkah-langkah perbaikan mulai langkah 3 sampai langkah 6 untuk memperbaiki tabel-tabel yang telah diubah. Perubahan berhenti bila pada baris pertama (fungsi tujuan) tidak ada yang bernilai negatif.

Selanjutnya, akan diberikan sebuah contoh kasus penggunaan model pemrogram linear.

Contoh 2.1:

Sebuah pabrik kue memproduksi 2 jenis produk yang berbeda, yaitu kue bolu dan kue kering. Kue bolu membutuhkan 4kg tepung dan 1kg cokelat, sementara kue kering membutuhkan 5kg tepung dan 2kg cokelat. Masing-masing produk membutuhkan masa kerja selama 2 jam. Produksi sebisa mungkin menggunakan bahan yaitu cokelat kurang dari 25kg, tepung kurang dari 120kg, dengan masa kerja kurang dari 40 jam. Berapakah kue yang harus diproduksi pabrik tersebut untuk mendapatkan laba maksimal?

Tabel 2.2. Tabel Contoh 2.1 Produksi Kue

Produk	Tepung (kg)	Cokelat (kg)	Masa Kerja (jam)	Laba
Kue Kering	5	2	2	30
Kue Bolu	4	1	2	40
	120	25	40	

Fungsi Tujuan:

$$Z = 30x_1 + 40x_2$$

Fungsi Kendala:

$$3x_1 + 4x_2 \leq 120$$

$$x_2 \leq 25$$

$$2x_1 + 2x_2 \leq 40$$

$$x_1, x_2 \geq 0$$

Masalah pemrograman linear ini diubah menjadi bentuk kanonik dengan menambahkan variabel *slack* $S_1 \geq 0$ pada kendala 1, $S_2 \geq 0$ pada kendala 2, dan $S_3 \geq 0$ pada kendala 3 sehingga bentuk umumnya menjadi:

Fungsi Tujuan:

$$Z = 30x_1 + 40x_2 + 0S_1 + 0S_2 + 0S_3$$

Fungsi Kendala:

$$5x_1 + 4x_2 + S_1 = 120$$

$$x_2 + S_2 = 25$$

$$2x_1 + 2x_2 + S_3 = 40$$

$$x_1, x_2, S_1, S_2, S_3 \geq 0$$

Tabel 2.3. Tabel Simpleks dalam Contoh Kasus 2.1

	C_j	30	40	0	0	0	b_i	R_i
$\bar{C}_i$	$\bar{x}_i/x$	x_1	x_2	S_1	S_2	S_3		
0	S_1	3	4	1	0	0	120	30
0	S_2	0	1	0	1	0	25	25

0	S_3	2	2	0	0	1	40	20
	Z_j	0	0	0	0	0	0	
	$Z_j - C_j$	-30	-40	0	0	0	0	

Karena masih terdapat *Zj – Cj ≤ 0*, maka tabel di atas belum optimal. Nilai *Zj – Cj ≤ 0* terkecil ada pada kolom variabel x_2 sehingga x_2 merupakan variabel baru yang masuk. Nilai R_i terkecil adalah 15, yaitu pada variabel S_3 sehingga S_3 keluar dan digantikan oleh x_2. Elemen pivotnya adalah 2 yang terletak pada perpotongan kolom x_2 dan baris S_3.

Pada tabel 2.3, untuk mengubah 2 menjadi 1 akan dilakukan operasi baris elementer (OBE) mengalikan 2 dengan seperdua. Elemen pada kolom x_2 lainnya, yaitu 4 diubah menjadi 0 dengan melakukan OBE menambah baris 1 dengan -4 dikali baris ketiga, dan untuk baris kedua yaitu 1 dengan menambah baris kedua dengan -1 kali baris ketiga. Diperoleh tabel simpleks baru seperti pada Tabel 2.4.

Tabel 2.4. Tabel Simpleks Iterasi 1 Contoh Kasus 2.1

	C_j	30	40	0	0	0	b_i	R_i
C_i	$\bar{x}_i/x$	x_1	x_2	S_1	S_2	S_3		
0	S_1	-1	0	1	0	-2	40	60
0	S_2	-1	0	0	1	-1/2	5	5
40	x_2	1	1	0	0	1/2	20	15
	Z_j	40	40	0	0	0	**800**	
	$Z_j - C_j$	10	0	0	0	0	0	

Dari Tabel 2.3 tidak ada lagi $Z_j - C_j \leq 0$. Dengan demikian telah dicapai penyelesaian optimal:

$$\{x_1, x_2, S_1, S_2, S_3\} = \{0,20,0,0,0\}$$

dengan nilai maksimum: Z = 30(0) + 40(20) + 0.0 + 0.0 + 0.0 = 800

H. Goal Programming

Goal programming merupakan salah satu model matematis yang dapat digunakan sebagai dasar dalam pengambilan keputusan untuk menganalisis dan membuat solusi suatu persoalan yang melibatkan banyak tujuan sehingga diperoleh solusi yang optimal. Pendekatan dasar *goal programming* adalah untuk menentukan suatu tujuan, merumuskan suatu fungsi tujuan untuk setiap tujuan, dan kemudian mencari penyelesaian dengan meminimalkan jumlah penyimpangan (deviasi) pada fungsi tujuan tersebut. Model *goal programming* berusaha untuk meminimalkan deviasi di antara berbagai tujuan atau sasaran yang telah ditentukan sebagai targetnya, maksudnya nilai ruas kiri persamaan kendala sebisa mungkin mendekati nilai ruas kanannya.

Model *goal programming* merupakan perluasan dari model pemrograman linear yang dikembangkan oleh A. Charnes dan W. M. Cooper pada tahun 1956 sehingga seluruh asumsi, notasi, formulasi matematika, prosedur perumusan model, dan penyelesaian tidak berbeda. Perbedaannya hanya terletak pada kehadiran sepasang variabel deviasi yang akan muncul pada fungsi tujuan dan fungsi kendala. Pemrograman linear sendiri adalah sebuah model matematis yang digunakan untuk menemukan suatu penyelesaian optimal dengan cara memaksimalkan atau meminimalkan fungsi tujuan

terhadap suatu kendala. Model *goal programming* mempunyai tiga unsur utama, yaitu variabel keputusan, fungsi tujuan, dan fungsi kendala.

Ada beberapa istilah yang digunakan dalam *goal programming* (Harjiyanto, 2014), yaitu:

1. Variabel keputusan, adalah seperangkat variabel yang tidak diketahui yang berada di bawah kontrol pengambilan keputusan, yang berpengaruh terhadap solusi permasalahan dan keputusan yang akan diambil. Biasanya dilambangkan dengan X_j (j = 1, 2, …, n)
2. Nilai sisi kanan, merupakan nilai-nilai yang biasanya menunjukkan ketersediaan sumber daya (dilambangkan dengan b_i) yang akan ditentukan kekurangan atau penggunaannya.
3. Koefisien teknologi, merupakan nilai-nilai numerik yang dilambangkan dengan a_{ij} yang akan dikombinasikan dengan variabel keputusan, di mana akan menunjukkan penggunaan terhadap pemenuhan nilai kanan.
4. Variabel deviasional, adalah variabel yang menunjukkan kemungkinan penyimpangan-penyimpangan negatif dan positif dari nilai sisi kanan fungsi tujuan. Variabel penyimpangan negatif berfungsi untuk menampung penyimpangan yang berada di bawah sasaran yang dikehendaki, sedangkan variabel penyimpangan positif berfungsi untuk menampung penyimpangan yang berada di atas sasaran. Dalam *goal programming* dilambangkan dengan d_i^- untuk penyimpangan negatif dan d_i^+ untuk penyimpangan positif dari nilai sisi kanan tujuan.

5. Fungsi pencapaian, adalah fungsi matematis dari variabel-variabel simpangan yang menyatakan kombinasi sebuah objektif.

6. Variabel non-negatif, adalah semua variabel bernilai positif atau sama dengan nol. Jadi, variabel keputusan dan variabel deviasi dalam masalah *goal programming* bernilai positif atau sama dengan nol. Contoh pernyataan non negatif dilambangkan $x_i,\ d_i^-, d_i^+ \geq 0$.

7. Prioritas, adalah suatu sistem urutan dari banyaknya tujuan pada model yang memungkinkan tujuan-tujuan tersebut disusun berdasarkan kepentingan masing-masing.

8. Pembobotan, merupakan pemberian bobot yang dinyatakan dalam angka yang digunakan untuk membedakan variabel simpangan *i* dalam suatu tingkat prioritas *k*.

1. Bentuk Umum Goal Programming

Bentuk umum dari *goal programming* dengan faktor prioritas di dalam strukturnya adalah sebagai berikut (Nasendi, B.D & Anwar Affendi, 1985):

Meminimumkan

$$\sum_{i=1}^{m} (P_y\ W_i^+ d_i^+ + P_s\ W_i^- d_i^-) \qquad (2.4)$$

dengan kendala,

$$\sum_{i=1}^{m} a_{ij} X_j\ + d_i^- - d_i^+ = b_i \qquad (2.5)$$

kendala tujuan

untuk $i = 1, 2, \ldots, m$

$$\sum_{i=1}^{n} g_{kj} X_j \ \leq atau \geq C_k \qquad (2.6)$$

Kendala fungsional

Untuk $k = 1, 2, \ldots, p$

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

dan $X_i \ , d_i^-, d_i^+ \geq 0$

Keterangan:

X_j	= variabel keputusan.
b_i	= target atau tujuan
a_{ij}	= koefisien fungsi kendala tujuan
d_i^-	= variabel deviasi yang mempresentasikan tingkatpencapaian di bawah target (*under achievement*).
d_i^+	= variabel deviasi yang mempresentasikan tingkat pencapaian di atas target (*over achievement*)
$W_{i,y}^+$ dan $W_{i,y}^-$	= bobot untuk masing-masing penyimpangan d_i^+ dan d_i^- dalam urutan (*ranking*) ke-y dan ke-s.
g_{kj}	= koefisien fungsi kendala biasa
C_k	= jumlah sumber daya *k* yang tersedia
$P_y \ , P_s$	= faktor-faktor prioritas $W_{i,y}^+ W_{i,y}^-$

2. Fungsi Tujuan

Berbeda dengan program linear yang fungsi tujuannya dapat memaksimalkan atau meminimalkan, untuk *goal programming* fungsi tujuannya hanya untuk meminimalkan jarak antara atau deviasi. Deviasi atau jarak antara merupakan ciri khas yang menandai model *goal programming*.

3. Memaksimalkan

Memaksimalkan fungsi tujuan *f(x)* berarti, jika dimisalkan *f(x)*= b_i merupakan suatu tujuan, maka b_i merupakan batas bawahnya sehingga hasil dari memaksimalkan fungsi tujuan *f(x)* haruslah $f(x) \geq b_i$

4. Meminimalkan

Untuk meminimalkan fungsi tujuan *f(x)* merupakan kebalikan dari memaksimalkan. Jika dimisalkan *f(x)* = b_i merupakan tujuan, maka b_i merupakan batas atasnya sehingga hasil dari meminimalkan fungsi tujuan *f(x)* haruslah $f(x) \leq b_i$

5. Variabel Deviasi

Variabel deviasi atau jarak antara merupakan perbedaan yang khusus membedakan antara pemrograman linear dengan *goal programming*.

Misalkan *d* merupakan variabel sembarang, maka $d = d^+ - d^-$ inilah yang disebut variabel deviasi.

dengan : $d^+ = \begin{cases} +d, untuk\ d \geq 0 \\ 0, untuk\ d < 0 \end{cases}$

$$d^- = \begin{cases} 0, untuk\ d \geq 0 \\ -d, untuk\ d < 0 \end{cases}$$

dengan d^+ = komponen positif dari *d*

d^- = komponen negatif dari *d*

Variabel deviasi mempunyai fungsi sebagai penampung terhadap tujuan-tujuan yang dikehendaki yang dibedakan menjadi dua bagian Harjiyanto (2014), yaitu:

a. Deviasi positif (d^+) untuk menampung deviasi yang berada di atas tujuan yang dikehendaki, maka d^+ akan selalu berkoefisien -1 pada setiap kendala tujuan, sehingga bentuk kendalanya adalah:

$$\sum_{i=1}^{m} a_{ij} X_j \; - d_i^+ = b_i$$

$$\sum_{i=1}^{m} a_{ij} X_j \; = b_i \; + d_i^+$$

untuk : $i = 1, 2, \ldots, m$

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

b. Deviasi negatif (d^-) untuk menampung deviasi yang berada di bawah tujuan yang dikehendaki, maka d^- akan selalu berkoefisien+1 pada setiap kendala tujuan sehingga bentuk kendalanya adalah:

$$\sum_{i=1}^{m} a_{ij} X_j \; + d_i^+ = b_i$$

$$\sum_{i=1}^{m} a_{ij} X_j \; = b_i \; - d_i^+$$

untuk : $i = 1, 2, \ldots, m$

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

Jika kedua deviasi tersebut digabungkan maka terbentuk model umum dari kendala tujuan sebagai berikut:

$$\sum_{i=1}^{m} a_{ij} X_j \; + d_i^- - d_i^+ = b_i \;\; , \sum_{i=1}^{n} a_{ij} X_j \; > b_i$$

Dikarenakan nilai minimum dari d^+ dan d^- adalah nol, maka dari model umum dari kendala tujuan di atas dapat disimpulkan sebagai berikut:

a. $d_i^+ = d_i^- = 0$ sehingga $\sum_{i=1}^{n} a_{ij} X_j = b_i$ artinya tujuan tercapai.

b. $d_i^+ > 0$ dan $d_i^- = 0$ sehingga $\sum_{i=1}^{n} a_{ij} X_j = b_i + d_i^+$ artinya tujuan tidak tercapai karena $\sum_{i=1}^{n} a_{ij} X_j < b_i$

c. $d_i^+ = 0$ dan $d_i^- > 0$ sehingga $\sum_{i=1}^{n} a_{ij} X_j = b_i + d_i^+$ artinya akan terlampaui karena $\sum_{i=1}^{n} a_{ij} X_j > b_i$.

Jelas bahawa kondisi di mana $d_i^+ > 0$ dan $d_i^- > 0$ pada sebuah kendala tujuan tidak akan mungkin terjadi.

6. Kendala Tujuan

Kendala tujuan merupakan kendala-kendala yang dihadapi dalam mencapai tujuan. Pada *goal programming* kendala-kendala adalah alat untuk mewujudkan tujuan yang hendak dicapai.

7. Kendala fungsional

Kendal fungsional atau struktural adalah kendala-kendala lingkungan yang tidak berhubungan langsung dengan tujuan-tujuan masalah yang dihadapi. Variabel deviasi tidak dimasukkan ke dalam kendala struktural, karena hal ini bukanlah merupakan fungsi tujuan.

8. Perumusan Masalah *Goal Programming*

Juanawati Marpaung (2009) menyatakan langkah perumusan permasalahan *goal programming* adalah sebagai berikut:

a. Penentuan variabel keputusan, merupakan dasar dalam pembuatan model keputusan untuk mendapatkan solusi yang dicari. Semakin tepat penentuan variabel keputusan akan mempermudah pengambilan keputusan yang dicari.
b. Penentuan fungsi tujuan, yaitu tujuan-tujuan yang ingin dicapai oleh perusahaan.
c. Perumusan fungsi tujuan, di mana setiap sasaran pada sisi kirinya ditambahkan dengan variabel simpangan, baik simpangan positif maupun simpangan negatif. Dengan ditambahkannya variabel simpangan, maka bentuk dari fungsi sasaran menjadi $f_i(x_i) + d_i^- - d_i^+ = b_i$.
d. Penentuan prioritas utama. Pada langkah ini dibuat urutan dari tujuan-tujuan. Penentuan tujuan ini tergantung pada keinginan dari pengambil keputusan dan keterbatasan sumber-sumber yang ada.
e. Penentuan pembobotan. Pada tahap ini merupakan kunci dalam menentukan urutan dalam suatu tujuan dibandingkan dengan tujuan yang lain.
f. Penentuan fungsi pencapaian. Dalam hal ini, yang menjadi kuncinya adalah memilih variabel simpangan yang benar untuk dimasukkan dalam fungsi pencapaian. Dalam memformulasikan fungsi pencapaian adalah menggabungkan setiap tujuan yang berbentuk minimasi variabel penyimpangan sesuai prioritasnya.
g. Penyelesaian model *goal programming*.

Pendekatan *goal programming* menganalisa seberapa banyak deviasi dari solusi yang didapatkan dari setiap tujuan yang ditentukan. Oleh karena itu, untuk setiap tujuan ditentukan sepasang variabel deviasinya (dengan satu variabel yang sama dengan jumlah di mana solusi yang tercapai melebihi tujuan; dan satu variabel yang lain yang sama dengan jumlah di mana solusi yang tercapai gagal mencapai tujuan).

Contoh 2.2.

Misalkan sebuah perusahaan tengah mempertimbangkan tiga bentuk pengiklanan: jumlah banyaknya pengiklanan via televisi (X_1), radio (X_2), dan koran (X_3). Setiap segmen televisi memerlukan biaya tiga juta atau 3000 (dalam ribu rupiah) dan mampu mencapai 1000 pelanggan potensial baru; setiap segmen radio memerlukan biaya 800 (dalam ribu rupiah) dan mampu mencapai 500 pelanggan potensial baru; dan setiap iklan koran membutuhkan biaya 250 (dalam ribu rupiah) dan mampu mencapai 200 pelanggan baru. Misalkan saja perusahaan tersebut memiliki tiga tujuan sebagai berikut:

Tujuan 1: Menggunakan biaya pengiklanan tidak lebih dari 25000 (dalam ribu rupiah).

Tujuan 2: Mencapai paling tidak 30000 pelanggan potensial baru.

Tujuan 3: Menjalankan paling tidak 10 segmen televisi.

Jika ketiga hal tersebut merupakan kendala, maka kendala fungsionalnya akan dituliskan sebagai berikut:

$$3000X_1 + 800X_2 + 250X_3 \leq 25000$$

$$1000X_1 + 500X_2 + 200X_3 \geq 30000$$

$$X_1 \qquad \geq \quad 10$$

Apabila kendala ketiga dipenuhi, maka nilai minimal untuk sisi kiri dari kendala pertama akan menjadi (3000)(10) = 30000. Karena hal ini melebihi batas maksimum yaitu 25000, tidak ada solusi layak yang memenuhi ketiga kondisi secara bersamaan.

Jika ketiga kondisi merupakan tujuan atau target, maka untuk setiap tujuan ke-i, didefinisikan:

d_i^- = jumlah nilai sisi kiri yang gagal mendekati nilai sisi kanan.

d_i^+ = jumlah nilai sisi kiri yang melebihi nilai sisi kanan.

Tujuan-tujuan tersebut sekarang dapat dituliskan sebagai persamaan-persamaan berikut:

$$3000X_1 + 800X_2 + 250X_3 + d_1^- - d_1^+ = 25000$$

$$1000X_1 + 500X_2 + 200X_3 + d_2^- - d_2^+ = 30000$$

$$X_1 \qquad + d_3^- - d_3^+ = \quad 10$$

Untuk d_1^+ (jumlah biaya yang digunakan melebihi 25000), d_2^- (angka pelanggan potensial baru yang tercapai di bawah 30000), dan d_3^- (jumlah iklan yang ditayangkan di televisi kurang dari 10) merepresentasikan *deviasi yang merugikan* untuk permasalahan ini. Sementara itu, menggunakan biaya di bawah anggaran, mencapai lebih dari 30000 pelanggan baru, atau menjalankan lebih dari 10 iklan televisi merupakan indikasi bahwa tujuan-tujuan tersebut telah dipenuhi. Tujuan dari sebuah permasalahan *goal programing* berkaitan dengan meminimasi variabel yang merugikan, yang memiliki dua pendekatan

berbeda untuk menyelesaikan masalah *goal programming preemptive* dan *non-preemptive*.

9. Metode *Non-Preemptive* atau Pembobotan

Prioritas sebagai suatu ukuran dari variabel-variabel deviasi yang diminimalkan sering mempunyai ukuran yang berbeda-beda. Hal ini terdapat dalam peminimuman biaya (yang mempunyai satuan rupiah) dan pemaksimuman kuantitas barang (yang mempunyai satuan unit) berada dalam prioritas yang sama. Secara sepintas hasil dari peminimuman variabel-variabel deviasi yang bersangkutan terdengar bertentangan.

Untuk mengatasi hal itu, dalam fungsi tujuan masing-masing variabel deviasi yang ada dalam satu prioritas diberi bobot. Sedangkan dalam hal kepentingan dari tujuan-tujuan yang berada dalam suatu prioritas yang tidak sama, untuk mengatasi kejadian tersebut maka masing-masing variabel deviasi diberi bobot.

Bobot adalah besaran numerik yang diberikan pada variabel-variabel yang diminimalkan pada fungsi tujuan *goal programming*. Bobot yang diberikan pada fungsi tujuan *goal programming* terjadi apabila:

a. Variabel-variabel deviasi yang terdapat pada suatu prioritas mempunyai ukuran yang berbeda.
b. Tingkat kepentingan untuk mencapai nilai tujuan dari setiap tujuan dalam suatu prioritas berbeda.

Dalam pendekatan *goal programming non-preemptive*, bobot relatif ditetapkan pada setiap deviasi yang merugikan. Hal ini bertindak sebagai

penalti setiap unit untuk kegagalan pemenuhan tujuan yang disebutkan. Bobot ini bisa duganakan untuk mengubah model *goal programming* menjadi model pemrograman linear yang mempunyai tujuan miminimalkan total deviasi yang diberi bobot dari tujuan-tujuannya. Hasilnya, model pemrograman linear tersebut akan mempunyai bentuk sebagai berikut (Frederick S.H. dan Gerald J.L, 2001):

Minimalkan (total deviasi yang diberi bobot dari tujuan-tujuannya)

Dengan kendala (1) Persamaan tujuannya.

(2) Kendala fungsional, jika ada.

(3) Kendala non negatif untuk semua variabel deviasi dan keputusan.

Dari Contoh 2.2, misalkan manajemen menentukan bahwa perusahaan membutuhkan 1 (dalam ribu rupiah) untuk setiap biaya ekstra yang dihabiskan bila pengiklanan di atas 25000 (dalam ribu rupiah) dan perusahaan rugi sebanyak 5 (dalam ribu rupiah) untuk setiap pelanggan potensial yang tidak mencapai 30000. Oleh sebab itu, bobot pada variabel deviasi yang merugikan yaitu d_2^- adalah lima kali lipat lebih banyak dari bobot yang ditetapkan pada d_1^+. Bobot yang ditetapkan pada percabangan ‘mempunyai kurang dari sepuluh segmen iklan televisi’ bisa jadi merupakan perkiraan yang subjektif. Pembuat keputusan bisa saja mempunyai ‘prediksi’ bahwa setiap segmen iklan televisi (di bawah 10) tersebut senilai dengan 100 kali biaya di atas anggaran.

Tujuan dari pendekatan solusi *non-preemptive* adalah untuk meminimasi jumlah total deviasi merugikan yang diberi bobot dari tujuan-tujuannya.

Pendekatan ini mengasumsikan bahwa tidak ada keuntungan yang diperoleh dengan melebihi target tujuan. Maka, fungsi objektif *goal programming* metode *non-preemptive* untuk masalah ini adalah:

Minimalkan $d_1^+ + 5d_2^- + 100U_3$

Dengan kendala

$$3000X_1 + 800X_2 + 250X_3 + d_1^- - d_1^+ = 25000$$

$$1000X_1 + 500X_2 + 200X_3 + d_2^- - d_2^+ = 30000$$

$$X_1 + d_3^- - d_3^+ = 10$$

$$X_J, d_J^+, d_J^- \geq 0 \ \forall J$$

10. Metode *Preemptive* atau Prioritas

Pengambil keputusan menghadapi suatu persoalan dengan tujuan ganda, tapi satu tujuan dengan tujuan lainnya saling bertentangan(*multiple and conflicting goals*). Dalam memecahkan persoalan tersebut, maka pengambil keputusan harus menentukan mana dari antara berbagai tujuan tersebut yang diutamakan atau diprioritaskan (Firdaus, 2005).

Tujuan yang paling penting ditentukan sebagai prioritas ke-1. Tujuan yang kurang begitu penting ditentukan sebagai prioritas ke-2 demikian seterusnya.Pembagian prioritas tersebut dikatakan sebagai pengutamaaan (*preemptive*), yaitu mendahulukan tercapainya tujuan yang telah diberikan prioritas utama sebelum menuju kepada tujuan-tujuan atau prioritas-prioritas berikutnya. Jadi, harus disusun dalam suatu urutan (*ranking*) menurut prioritasnya.

Dalam perumusan *goal programming* dinyatakan faktor prioritas tersebut sebagai P_i (untuk i = 1,2,...,m). Faktor-faktor prioritas tersebut memiliki hubungan sebagai berikut: $P_1 > P_2 > P_3 > P_{i+1}$ di mana > berarti ”jauh lebih tinggi daripada”.

Hubungan prioritas tersebut di atas menunjukkan bahwa walaupun faktor prioritas W_i tersebut kita gandakan atau kalikan sebanyak n kali (dimana $n > 0$), namun faktor yang diprioritaskan tersebut akan tetap menjadi yang teratas. Dengan kata lain prioritas di bawahnya tidak dapat menjadi lebih tinggi daripada prioritas di atasnya, walaupun sudah dikalikan sebanyak n kali . Jadi hubungan $nP_{i+1} > P_i$ tidak mungkin terjadi dalam persoalan *goal programming* yang memakai ketentuan pengutamaan (urutan prioritas).

Metode *preemptive goal programming* mempunyai sedikit pendekatan yang berbeda dengan metode *non-preemptive*. Pendekatan ini memaksa pembuat keputusan untuk memprioritaskan tujuan-tujuannya ke dalam tingkatan prioritas-prioritas (1, 2, 3, dan seterusnya), yang pada setiap prioritas memiliki satu atau lebih tujuan. Jika sebuah tingkat prioritas memiliki dua atau lebih tujuan, tujuan-tujuan ini akan diberikan bobot sebagai mana sama halnya dengan yang ada di metode *non-preemptive*.

Pembuat keputusan menempatkan tujuan-tujuan pada tingkat prioritas yang berbeda-beda, yang berarti pembuat keputusan akan berusaha sekuat mungkin untuk mencapai tujuan pada prioritas 1, yang selanjutnya mencoba mencapai prioritas 2, kemudian dilanjutkan pada prioritas 3, dan seterusnya (Frederick S.H. & Gerald J.L, 2001). Pada contoh sebelumnya, perhatian

pertama manajemen bisa berupa memenuhi jumlah anggaran dan menarik pelanggan potensial baru; keduanya adalah tujuan-tujuan prioritas 1, yang diikuti menjalankan 10 iklan televisi sebagai prioritas 2.

Ide utama dari pendekatan *preemptive goal programming* adalah supaya prioritas dengan tingkat tujuan yang lebih rendah tidak seharusnya dicapai dengan mengorbankan tujuan dengan prioritas yang lebih tinggi, karena prioritas pertama sangat diutamakan. Oleh karena itu, apabila total bobot deviasi minimum dari tujuan-tujuan dalam prioritas 1 mempunyai suatu nilai, V_1, maka harus dipastikan bahwa nilai prioritas 1 tetap pada V_1. Jika deviasi yang dibobotkan pada tujuan dalam prioritas 2 dalam kondisi ini adalah V_2, maka ketika meminimasi total bobot deviasi pada prioritas 3, nilai fungsi tujuan prioritas 1 harus tetap V_1, dan nilai fungsi tujuan prioritas 2 harus tetap V_2, dan seterusnya.

Untuk mengilustrasikannya, misalkan sebuah masalah dengan tujuh tujuan: dua tujuan dalam prioritas tingkat 1, tiga dalam prioritas tingkat 2, dan dua dalam prioritas tingkat 3; deviasi yang merugikan dari tujuan-tujuan tersebut adalah d_1^+, d_2^-, d_3^+, d_4^-, d_5^+, d_6^-, dan d_7^+, secara berurutan. Diasumsikan bobot W_1, W_2 telah ditentukan di antara tujuan tingkat 1, bobot W_3, W_4, W_5 di antara tujuan tingkat 2, dan bobot W_6, W_7 di antara tujuan tingkat 3. Prioritas 1 kemudian akan diselesaikan dengan pendekatan *preemptive goal programming*:

Prioritas 1.

Minimalkan $W_1 d_1^+ + W_2 d_2^-$

Dengan kendala Persamaan tujuan

Kendala Fungsional

Kendala non negatif

Misalkan solusinya menghasilkan sebuah nilai minimum fungsi tujuan, V_1, maka pemrograman linear untuk prioritas 2 selanjutnya akan diselesaikan.

Prioritas 2.

Minimalkan $W_3 d_3^+ + W_4 d_4^- + W_5 d_5^+$

Dengan kendala Persamaan Tujuan

Kendala fungsional

$W_1 d_1^+ + W_2 d_2^- = V_1$ (kendala baru)

Kendala non negatif

Misalkan solusinya menghasilkan nilai minimum fungsi tujuan, V_2. Maka, pemrograman linear untuk prioritas 3 selanjutnya akan diselesaikan.

Prioritas 3.

Minimalkan $W_6 d_6^- + W_7 d_7^+$

Dengan kendala Persamaan tujuan

Kendala fungsional

$W_1 d_1^+ + W_2 d_2^- = V_1$

$W_3 d_3^+ + W_4 d_4^- + W_5 d_5^+ = V_2$ (kendala baru)

Kendala non negatif

Solusi optimal untuk prioritas 3 kemudian akan menjadi solusi optimal untuk permasalahan *goal programming* tersebut.

11. Metode Pemecahan Masalah

Masalah *goal programming* dapat diselesaikan dengan metode simpleks dengan menggunakan variabel keputusan lebih dari dua. Juanawati Marpaung (2009) menyatakan langkah-langkah penyelesaian *goal programming* dengan metode simpleks, sebagai berikut:

a. Membentuk tabel simpleks awal.

b. Pilih kolom kunci di mana *Cj - Zj* memiliki negatif terbesar. Kolom kunci ini disebut kolom pivot.

c. Pilih baris kunci yang berpedoman pada b_i/a_{ij} dengan rasio terkecil di mana b_i adalah nilai sisi kanan dari setiap persamaan. Baris kunci ini disebut baris pivot.

d. Mencari bentuk kanonik, yaitu sistem di mana nilai elemen pivot bernilai 1 dan elemen lain bernilai nol dengan cara mengalikan baris pivot dengan -1 lalu menambahkannya dengan semua elemen di baris pertama. Dengan demikian, diperoleh tabel simpleks iterasi 1.

e. Pemeriksaan optimisasi, yaitu melihat apakah solusi layak atau tidak. Solusi dikatakan layak bila variabel adalah positif atau nol.

Setelah model *goal programming* tersebut diselesaikan dengan metode simpleks, maka diperoleh nilai dari variabel $x_1, \ldots, x_n$ yang mengoptimalkan fungsi tujuan. Selain itu, diperoleh juga variabel-variabel simpangan yang diartikan sebagai besarnya penyimpangan dari tujuan, tetapi dijamin simpangan yang diperoleh tetap paling minimal. Tabel 2.5 sebagai tabel awal model *goal programming*.

Tabel 2.5. Tabel Awal Goal Programming.

	C_j	0	0	…	0	$\omega_1 P_1$	$\omega_1 P_1$	…	$\omega_m P_m$	$\omega_m P_m$		
$\bar{C}_i$	$\bar{x}_i$ /x	x_1	x_2	…	x_m	d_1^-	d_1^+	…	d_m^-	d_m^+	b_i	R_i
$\omega_1 P_1$	d_1^-	a_{11}	a_{12}	…	a_{1m}	1	-1	…	0	0	b_1	R_1
$\omega_1 P_1$	d_2^-	a_{21}	a_{22}	…	a_{2m}	0	0	…	0	0	b_2	R_2
$\omega_m P_m$	d_m^-	a_{m1}	a_{m2}	…	a_{mm}	0	0	…	1	-1	b_m	R_m
	Z_j	…	…	…	…	…	…	…	…	…	Z	
	C_j $- Z_j$	…	…	…	…	…	…	…	…	…	Z	

Keterangan:

$\bar{x}_i$: variabel basis
$\bar{C}_i$: koefisien dari $\bar{x}_i$
Z_j : $\sum_{i=1}^{m} \bar{C}_i \, a_{ij}$
Z : $\sum_{i=1}^{m} \bar{C}_i \, b_i$, nilai fungsi tujuan

R_i rasio antara b_i dan a_{ik} jika x_k terpilih menjadi variabel basis.

Contoh 2.3.

Sebuah pabrik memproduksi 2 macam produk yang berbeda, yaitu X_1 dan X_2. Kedua produk tersebut memiliki dua tahap pemrosesan. Proses pertama sanggup menghasilkan 5 unit produk X_1 dan 4 unit produk X_2 dengan kapasitas maksimum 50 unit. Proses kedua mampu menghasilkan 3 unit produk X_1 dan 2 unit produk X_2 dengan kapasitas maksimum sebanyak 36 unit. Dalam contoh kasus ini, pembuat keputusan di pabrik menetapkan 4 macam sasaran:

1. Kapasitas pada proses pertama dimanfaatkan secara maksimum.
2. Kapasitas pada proses kedua dimanfaatkan secara maksimum.
3. Produksi X_1 paling tidak 8 unit.
4. Produksi X_2 paling tidak 6 unit.

Berapakah jumlah produksi optimal yang dapat diproduksi oleh pabrik tersebut?

Penyelesaian:

Variabel keputusan dari contoh kasus di atas adalah:

X_1 = jumlah produk X_1 yang akan diproduksi

X_2 = jumlah produk X_2 yang akan diproduksi

Dengan kendala:

$$5X_1 + 4X_2 \leq 50 \qquad \text{(sasaran 1)}$$

$$3X_1 + 2X_2 \leq 36 \qquad \text{(sasaran 2)}$$

$$X_1 \geq 8 \qquad \text{(sasaran 3)}$$

$$X_2 \geq 6 \qquad \text{(sasaran 4)}$$

Menggunakan sasaran yang ingin dicapai sebagai acuan, maka model *goal programming* untuk contoh kasus di atas menjadi:

Minimalkan $Z = P_1(d_1^- + d_1^+) + P_2(d_2^- + d_2^+) + P_3(d_3^-) + P_4(d_4^-)$

Dengan kendala:

$$5X_1 + 4X_2 + d_1^- - d_1^+ = 50$$

$$3X_1 + 2X_2 + d_2^- - d_2^+ = 36$$

$$X_1 + d_3^- = 8$$

$$X_2 + d_4^- = 6$$

Model di atas kemudian akan diselesaikan menggunakan metode simpleks.

Tabel 2.6. Tabel Awal Contoh 2.3.

	C_j	0	0	1	1	1	1	1	1	b_i	R_i
C_i	$\bar{x}_i/x$	x_1	x_2	d_1^+	d_1^-	d_2^+	d_2^-	d_3^-	d_4^-		
1	d_1^-	5	4	-1	1	0	0	0	0	50	10
1	d_2^-	3	2	0	0	-1	1	0	0	36	12
1	d_3^-	1	0	0	0	0	0	1	0	8	8
1	d_4^-	0	1	0	0	0	0	0	1	6	∞
	Z_j	9	7	-1	1	-1	1	1	1	90	
	$C_j - Z_j$	-9	-7	2	0	2	0	0	0		

Sesuai dengan langkah-langkah penyelesaian masalah *goal programming* menggunakan metode simpleks, maka kolom ke-1 menjadi kolom kunci dan baris ke-3 menjadi baris kunci.

Tabel 2.7. Tabel Simpleks Iterasi I Contoh 2.3.

	C_j	0	0	1	1	1	1	1	1	b_i	R_i
C_i	$\bar{x}_i/x$	x_1	x_2	d_1^+	d_1^-	d_2^+	d_2^-	d_3^-	d_4^-		
1	d_1^-	0	4	-1	1	0	0	-5	0	10	5/2
1	d_2^-	0	2	0	0	-1	1	-3	0	12	6
0	x_1	1	0	0	0	0	0	1	0	8	∞
1	d_4^-	0	1	0	0	0	0	0	1	6	6
	Z_j	0	7	-1	1	-1	1	-8	0	90	
	$C_j - Z_j$	0	-7	2	0	2	0	9	0		

Dapat dilihat dari tabel di atas bahwa kolom dan baris kedua merupakan baris dan kolom kunci, maka akan dilakukan operasi baris elementer (OBE) untuk menentukan hasil yang optimal.

Tabel 2.8. Tabel Simpleks Iterasi II Contoh 2.3.

	C_j	0	0	1	1	1	1	1	1	b_i	R_i
C_i	$\bar{x}_i/x$	x_1	x_2	d_1^+	d_1^-	d_2^+	d_2^-	d_3^-	d_4^-		
0	x_2	0	1	-1/4	1/4	0	0	-5/4	0	10/4	
1	d_2^-	0	0	1/2	-1/2	-1	1	-1/2	0	7	
0	x_1	1	0	0	0	0	0	1	0	8	
1	d_4^-	0	0	1/4	-1/4	0	0	5/4	1	6	
	Z_j	0	0	3/4	-3/4	-1	1	3/4	1	90	
	$C_j - Z_j$	0	0	1/4	7/4	2	0	1/4	0		

Pada Tabel 2.8 diperoleh solusi optimal karena seluruh $C_j - Z_j \geq 0$. Dengan demikian, solusi yang optimal adalah pabrik yang memproduksi produk X_1sebanyak 8 unit dan produk X_2 sebanyak 10/4 unit, atau 2 unit jika produknya berupa satu benda utuh.