BAB 2

LANDASAN TEORI

2.1. Operasi Riset *(Operation Research)*

Menurut *Operation Research Society of Great Britain*, *operation research* adalah penerapan metode-metode ilmiah dalam masalah yang kompleks dan suatu pengelolaan system manajemen yang besar, baik yang menyangkut manusia, mesin, bahan dan uang dalam industry, bisnis, pemerintahan dan pertahanan. Pendekatan ini menggabungkan dan menerapkan metode ilmiah yang sangat kompleks dalam suatu pengelolaan manajemen dengan menggunakan faktor-faktor produksi yang ada dan digunakan secara efektif dan efesien untuk membantu pengambilan keputusan dalam kebijakan suatu perusahaan. Definisi lain menurut *Operation Research Society of America* (ORSA), *operation research* berkaitan dengan pengambilan keputusan secara ilmiah dan bagaimana membuat suatu model yang baik dalam merancang dan menjalankan sistem yang melalui alokasi sumber daya yang terbatas. Inti dari beberapa kesimpulan di atas adalah bagaimana proses pengambilan keputusan yang optimal dengan menggunakan alat analisis yang ada dan adanya keterbatasan sumber daya.

2.2.Pengertian Program Dinamik

Program Dinamik adalah suatu teknik matematika yang digunakan untuk mengoptimalkan proses pengambilan keputusan secara bertahap-ganda. Dalam teknik ini, keputusan yang menyangkut suatu persoalan dioptimalkan secara bertahap dan bukan secara sekaligus. Jadi inti dari teknik ini adalah membagi satu persoalan atas beberapa bagian persoalan yang dalam program dinamik disebut tahap. Kemudian memecahkan tiap tahap dengan mengoptimalkan keputusan atas tiap tahap sampai seluruh persoalan telah terpecahkan. Keputusan yang optimal atas seluruh persoalan ialah kumpulan dari sejumlah keputusan optimal atas seluruh tahap yang kemudian disebut sebagai kebijakan optimal. (P.Siagian, 1987)

Programasi dinamik memberikan prosedur yang sistematis untuk penentuan kombinasi pengambilan keputusan yang memaksimumkan keseluruhan efektivitas. Berbeda dengan Linier Programming, dalam program dinamik tidak ada rumusan (formulasi) matematis standard. Program dinamik lebih merupakan suatu tipe pendekatan umum untuk pemecahan masalah dan persamaan-persamaan khusus yang akan digunakan harus dikembangkan sesuai dengan setiap situasi individual (Aidawayati R. 2013)

Pendekatan program dinamik didasarkan pada prinsip optimasi Bellman (1950) yang mengatakan :

Suatu kebijakan optimal mempunyai sifat bahwa apa pun keadaan dan keputusan awal, keputusan berikutnya harus membentuk suatu kebijakan optimal dengan memperhatikan keadaan dari hasil keputusan pertama.

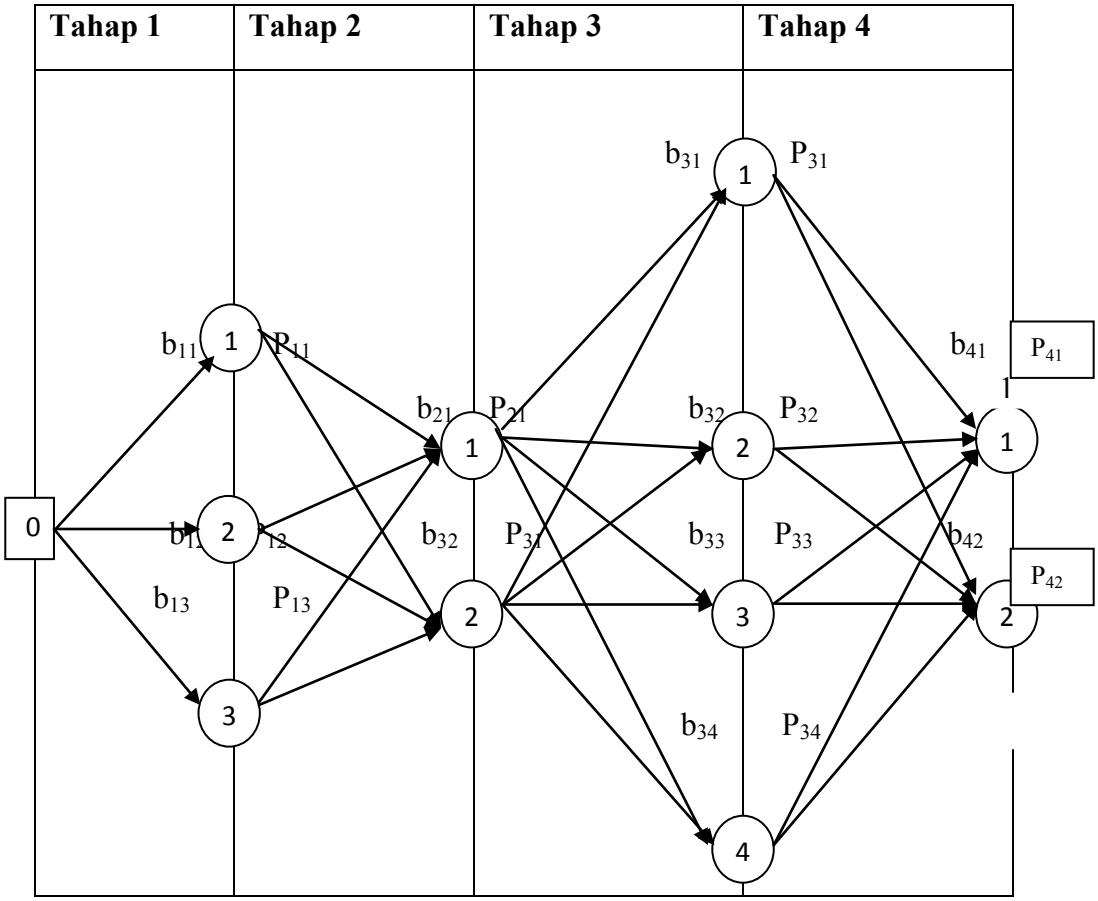
Prinsip ini mengandung arti bahwa :

1. Diperkenankan untuk mengambil keputusan yang layak bagi tahap persoalan yang masih tersisa tanpa melihat kembali keputusan-keputusan masa lalu atau tahap-tahap terdahulu
2. Dalam rangkaian keputusan yang telah diambil, hasil dari masing tergantung pada hasil keputusan sebelumnya dalam rangkaian.

Program Dinamik merupakan rangkaian prosedur yang mengoptimalkan dimana diberi fungsi objektif yang disebut hasil dan fungai biaya yang tergantung apakah fungsi akan dimaksimumkan atau diminimumkan. Berkenaan dengan variabel yang mana akan dioptimalkan disebut keputusan. Masalah dibuat adalah dugaan dalam program dinamik atau tahap. Dengan cara ini masalah dipecah menjadi beberapa tahap atau point waktu dan tujuan setiap tahap memilih keputusan optimal sehingga fungsi objektif mendapat nilai optimal dari tahap-tahap tersebut. Dengan memilih keputusan khusus yang diberikan suatu tahap dapat mempengaruhi jalannya proses yang disusun untuk interval selanjutnya. Proses ini dilakukan sepanjang interval yang berhubungan.

Sehingga dasar program dinamik adalah teknik yang memilih cara yang paling optimal dari antara semua cara yang mungkin sehingga fungsi objektif yang diberikan (dimana umumnya tergantung pada cara yang diikuti atau dipakai dan keputusan yang diambil) adalah optimal. Dalam rangkaian akan selalu berfikir optimis yang akan menjadi prosedur meminimumkan jika diketahui meminimumkan adalah fungsi yang dituliskan fungsi negative dari memaksimumkan.

P.Siagian (1987) mengemukakan bahwa prosedur pemecahan persoalan dalam program dinamik dilakukan secara rekursif. Ini berarti bahwa setiap kali mengambil keputusan, harus memperhatikan keadaan yang dihasilkan oleh keputusan sebelumnya. Karena itu, keadaan yang diakibatkan oleh suatu keputusan sebelumnya dan merupakan landasan bagi keputusan berikutnya, sehingga konsep tentang keadaan adalah sangat penting.



Gambar 2.1. Diagram Jaringan Keputusan

Secara umum dapat dinyatakan bahwa :

1. $b_{11} = b_{ij}$ yang menyatakan b di mana $i = 1$ dan $j = 1$, hal yang sama berlaku untuk P_{11} yakni $P_{11} = P_{ij}$.
2. Tahap *i* diperluas dengan alternatif rencana perluasan *j*.
3. Besaran-besaran b_{ij} menyatakan jumlah biaya yang diperlukan untuk perluasan dan P_{ij} menyatakan jumlah perolehan dari tahap *i* untuk rencana perluasan *j*.
4. $b_{ij} = P_{ij} = 0$ untuk alternatif awal yaitu tanpa perluasan sama sekali.

Keistimewaan dasar yang mencirikan masalah program dinamik adalah :

1. Permasalahan dapat dibagi-bagi dalam tahap-tahap, dengan suatu keputusan kebijakan (*policy decision*) diperlukan di setiap tahap, masalah program dinamik memerlukan pembuatan suatu urutan keputusan yang saling berhubungan, di mana setiap keputusan berhubungan dengan suatu tahap permasalahan.
2. Setiap tahap memiliki sejumlah keadan (*state*) yang bersesuaian. Secara umum, keadaan adalah berbagai kondisi yang mungkin, dimana system berada pada tahap tertentu dari keseluruhan permasalahan.
3. Pengaruh keputusan kebijakan pada setiap tahap adalah untuk merubah keadaan sekarang menjadi keadan yang saling berkaitan dengan tahap berikutnya.
4. Prosedur penyelesaian dirancang untuk menemukan suatu kebijakan optimal untuk keseluruhan masalah, yaitu pemberian keputusan kebijakan optimal pada setiap tahap untuk setiap kemungkinan keadan.
5. Bila diketahui keadaan sekarang, kebijakan optimal untuk tahap-tahap yang tersisa adalah bebas terhadap kebijakan yang dipakai pada tahap-tahap sebelumnya. Ini adalah prinsip keoptimalan program dinamik
6. Prosedur penyelesaian dimulai dengan menemukan kebijakan optimal untuk tahap terakhir. Kebijakan optimal untuk tahap terakhir memberikan keputusan kebijakan optimal untuk setiap kemungkinan keadaan pada tahap tersebut.

7. Tersedia hubungan rekursif yang mengidentifikasi kebijakan optimal pada tahap n, bila diketahui kebijakan optimal untuk tahap (n+1).

   Dengan demikian untuk menemukan keputusan kebijakan optimal, bila dimulai pada keadaan s pada tahap n, memerlukan penemuan nilai yang mengoptimalkan. Dengan menggunakan nilai x_n dan mengikuti kebijakan optimal bila dimulai dari keadaan pada tahap (n+1).

   Bentuk pasti dari hubungan rekursif berbeda-beda diantara masalah-masalah program dinamik. Akan tetapi notasi yang serupa ini dapat terus digunakan seperti yang di ringkas sebagai berikut :

   N = banyaknya tahap

   n = label untuk tahap sekarang (n = 1, 2, 3,..., N)

   s_n = keadaan sekarang untuk tahap n

   x_n = peubah keputusan untuk tahap n

   x_n^* = nilai optimal x_n (diketahui S_n)

   $f_n(s_n, x_n)$ = kontribusi tahap n, n+1, ..., N kepada fungsi tujuan bila sistem dimulai dari keadaan s_n pada tahap n, keputusan sekarang adalah x_n dan keputusan optimal dibuat sesudahnya.

   $f_n^*(s_n) = f_n(s_n, x_n^*)$

   Hubungan rekursif akan selalu memiliki bentuk :

   $f_n^*(s_n) = \max \{f_n(s_n, x_n)\}$ atau $f_n^*(s_n) = \min \{f_n(s_n, x_n)\}$

   dimana $f_n(s_n, x_n)$ akan dinyatakan dalam s_n, x_n, $f^*_{n+1}(s_{n+1})$ dan mungkin beberapa ukuran tentang keefektifan (atau ketidakefektifan) tahap pertama dari x_n.

   Hubungan rekursif dinamakan demikian karena hubungan tersebut selalu berulang setiap bergerak ke belakang tahap demi tahap. Bila tahap sekarang bernomor n diturunkan satu tahap, maka fungsi $f^*(s_n)$

   baru akan diturunkan menggunakan $f^*_{n+1}(s_{n+1})$ yang baru saja diturunkan dalam iterasi sebelumnya, proses ini berulang terus.

8. Bila menggunakan hubungan rekursif ini, prosedur penyelesaian bergerak mundur tahap demi tahap setiap kali menemukan kebijakan optimal untuk tahap tersebut sampai ditemukan kebijakan optimal yang dimulai dari tahap awal.

2.2.1. Program Dinamik Deterministik

Pendekatan program dinamik ke masalah deterministik, dimana keadaan pada tahap berikut ditentukan sepenuhnya oleh keadaan dan keputusan kebijakan pada tahap sekarang. Pada tahap n proses akan berada pada suatu keadaan s_n. Pembuatan keputusan kebijakan x_n selanjutnya menggerakkan proses ke keadaan s_{n+1} pada tahap (n+1). Kontribusi sesudahnya terhadap fungsi tujuan di bawah kebijakan yang optimal telah dihitung sebelumnya sebagai $f^*_{n+1}(s_{n+1})$.

Keputusan kebijakan x_n juga meberikan kontribusi kepada fungsi tujuan. Kombinasi kedua nilai ini dengan benar akan memberikan $f_n(S_n, x_n)$ yaitu kontribusi n tahap ke depan kepada fungsi tujuan. Pengoptimalan terhadap $x_n f_n^*(s_n) = f_n(s_n, x_n^*)$. Setelah ditemukan x_n^* dan $f_n^*(s_n)$ untuk setiap nilai s_n , prosedur penyelesaian sekarang bergerak mundur satu tahap. Program dinamik deterministik dapat diuraikan dengan diagram yang ditunjukkan dibawah ini :

Gambar 2.2. Program Dinamik Deterministik.

Keterangan

Satu cara dari kategori masalah program dinamik deterministik adalah dengan fungsi objektif. Misalnya, memperkecil jumlah kontribusi dari masing-masing tahap (seperti masalah perhentian), atau untuk memaksimalkan. Kategori lain dalam himpunan asli

dari tahap untuk *respective* tahap. Secara khusus, status s_n mungkin dapat digantikan dengan variabel status yang diskrit (seperti masalah perhentian) atau dengan variabel tahap yang kontinu, atau mungkin tahap vektor (lebih dari satu variabel) diperlukan.

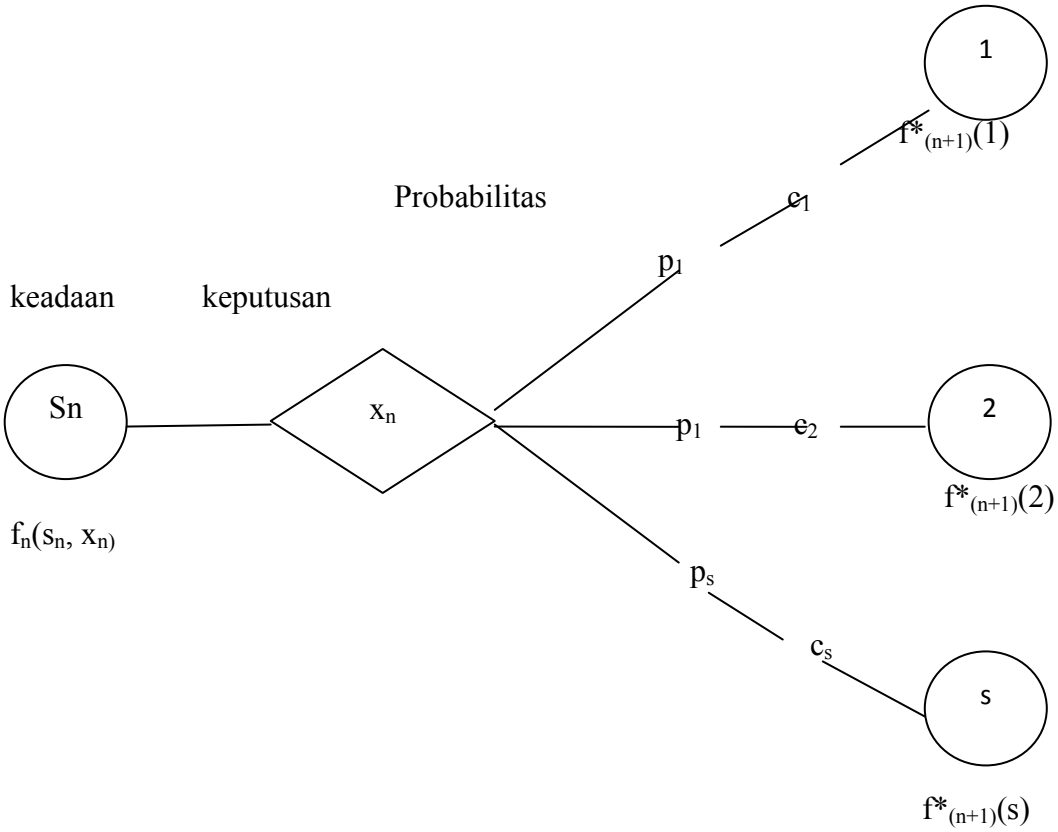
2.2.2. Program Dinamik Probabilistik

Program Dinamik Probabilistik berbeda dengan Program Dinamik Deterministik. Di mana pemrograman dinamik Deterministik, pada tahap berikutnya sepenuhnya ditentukan oleh keadaan dan keputusan kebijakan pada tahap sebelumnya, sedangkan pemrograman dinamik probabilistik terdapat suatu probabilitas keadaan mendatang yang distribusi peluang ini tetap ditentukan oleh keadaan dan keputusan kebijakan pada keadaan sebelumnya (Aidawayati R. 2013)

Terdapat dua hal dalam pemrograman dinamik probabilistik yaitu :

1. Stage berikutnya tidak seluruhnya ditentukan oleh stage dan keputusan pada stage saat ini, tetapi ada suatu distribusi kemungkinan mengenai apa yang akan terjadi.
2. Distribusi kemungkinan ini masih seluruhnya ditentukan oleh state dan keputusan pada stage saat ini.

Struktur dasar dalam pemrograman dinamik probabilistik diuraikan pada gambar berikut :



Gambar 2.3. Dinamik Probabilistik

Dimana :

a. s melambangkan banyaknya keadaan yang mungkin pada tahap (stage) n + 1 dan keadaan ini didambarkan pada sisisebelah kanan sebagai 1, 2, …,s
   (p_1, p_1, ..., p_s) adalah distribusi kemungkinan dari terjadinya suatu state berdasarkan state s_n dan keputusan x_n pada stage n
b. c_i adalah kontribusi dari stage n terhadap fungsi tujuan jika state berubah menjadi state i
c. $f_n(s_n, x_n)$ menunjukkan jumlah ekspektasi minimal dari tahap n ke depan, dengan diberikan status dan keputusan pada tahap n masing-masing s_n dan x_n

Karena adanya struktur probabilistik, hubungan antara $f_n(s_n, x_n)$ dan $f_{n+1}(s_n, x_n)$ agak lebih rumit dari pada untuk pemrograman dinamik deterministik. Bentuk yang tepat dari hubungan tersebut tergantung pada bentuk fungsi tujuan secara umum. Dalam pemrograman dinamik probabilistik juga terdapat hubungan

rekursif yang mengidentifikasi kebijakan optimal. Ada dua prosedur rekursif dalam pemrograman probabilistik yaitu :

a. *Forward Rescursive equation* (perhitungan dari depan ke belakang). Program dinamik bergerak dari tahap 1 sampai tahap n. Peubah keputusan adalah x_1, x_2, … x_n,
b. *Backward Recursive equation* (perhitungan dari belakang ke depan). Program dinamik bergerak mulai dari n, terus mundur ke tahap n-1, n-2, dan seterusnya sampai tahap 1. Peubah keputusan adalah x_n, x_{n-1}, …,x_1

Sebagai ilustrasi, misalkan tujuannya adalah minimalkan jumlah yang diharapkan dan kontribusi tahap-tahap secara terpisah. Pada kasus ini $f_n(s_n, x_n)$ menggambarkan jumlah minimal yang diharapkan dari tahap n dan seterusnya, bila diketahui bahwa keadaan dan keputusan kebijakan pada tahap n adalah s_n dan x_n akibatnya,

$$f_n(s_n, x_n) = \sum_{i=1}^{n} p_i \left[c_i + f_{n+1}(i)\right]$$

Dengan $f^*_{n+1}(i)$ = minimal f_{n+1} = minimal $\{ f_{n+1}(i, x_{n+1})\}$

Dimana minimal ini di buat di atas nilai kelayakan bagi x_{n+1}

Penggunaan program dinamik probabilistik selalu di pakai dalam game (permainan), penambahan penolakan (reject allowence) atau jumlah tambahan produk yang diproduksi, dan lain-lain.

2.3. Perumusan Masalah dalam program Dinamik

Perlu diterangkan bahwa yang dimaksud dengan perawat dalam penulisan skripsi ini adalah perawat-perawat yang bekerja disetiap ruangan di rumah sakit. Perawat-perawat dialokasikan untuk meningkatkan pelayanan kesehatan dirumah sakit tersebut. Oleh sebab itu manajemen harus mempunyai penempatan yang baik dalam menempatkan tenaga kerja di setiap shift. Pengalokasian tenaga kerja tersebut harus dapat meningkatkan pelayanan medis yang diberikan kepada pasien yang datang.

Bila tenaga perawat tersebut dibagi menjadi beberapa kelompok maka manajemen rumah sakit perlu menentukan berapa banyak tenaga perawat untuk dialokasikan disetiap shift dimana shift tersebut ada tiga bagian yaitu pagi, siang dan malam, guna memaksimumkan keefektifan total dari tenaga kerja. Banyaknya tenaga kerja yang akan dialokasikan adalah bilangan bulat.

Perumusan masalah ini memerlukan pembuatan tiga keputusan yang saling berkaitan, yaitu berapa banyak tenaga kerja untuk dialokasikan ke setiap shift. Dengan demikian walaupun tidak ada urutan tetap, ketiga shift ini dapat di anggap sebagai tahap dalam rumusan program dinamik . Peubah keputusan x_n (n = 1, 2, 3) adalah banyaknya kelompok yang dialokasikan ke tahap (shift) n.

2.4.Program Dinamik dalam Perhitungan Mundur dan Perhitungan Maju.

Dalam program dinamik perhitungan dalam tahap-tahap dengan merinci masalah menjadi beberapa bagian masalah. Setiap bagian masalah kemudian dipertimbangkan secara terpisah dengan tujuan untuk mengurangi jumlah dan kerumitan perhitungan. Tetapi, karena semua bagian masalah saling bergantung maka harus dipikirkan sebuah prosedur untuk menghubungkan perhitungan dengan cara yang menjamin bahwa pemecahan yang layak untuk tiap-tiap tahap juga layak untuk keseluruhan masalah.

Gagasan program dinamik secara praktis menghilangkan pengaruh saling ketergantungan antara tahap-tahap dengan menghubungkan definisi suatu keadaan dengan setiap tahap. Suatu keadaan bisa didefinisikan untuk menunjukkan status batasan yang mengikat semua tahap secara bersama-sama.

1. Untuk program dinamik yang akan diselesaikan dengan perhitungan mundur (*backward recursive*), maka perhitungan n tahap untuk mendapatkan keputusan optimal dimulai dari keadaan dari masalah yang akhir ke masalah pertama.

   $f_0(x_0) = 0$

   $f_n^*(s_n) = \text{opt}\ \{ p_n(x_n) \otimes f^*_{n+1}(s_n \otimes x_n)\}$, n = N, …, 1

   dimana :

$f_n^*(s_n)$: fungsi optimum

s : state (status)

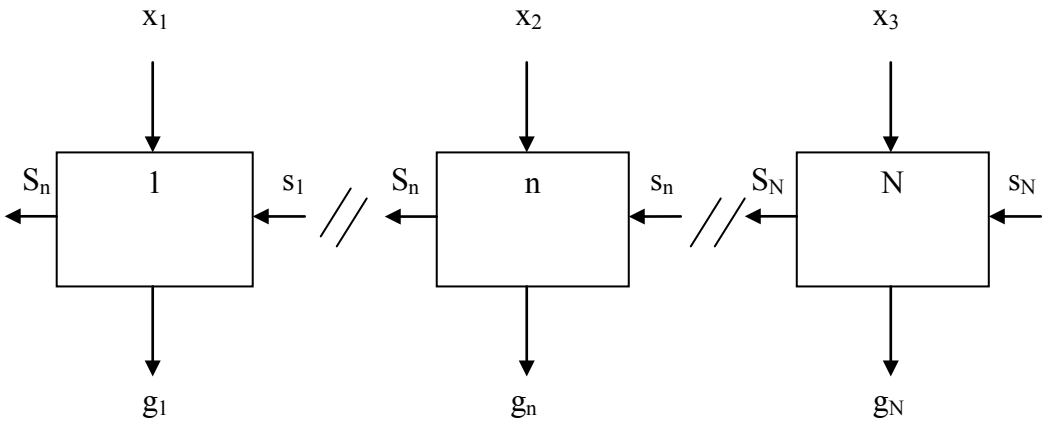
$s_n \otimes x_n$: fungsi transisi

n : tahap ke

x : variabel keputusan.

N : banyaknya tahap

Aidawayati Rangkuti (2013) menunjukkan konsep keadaan pada Rekursif mundur (*backward recursive)* pada gambar dibawah ini:



Gambar 2.4. konsep keadaan rekursif mundur

2. Untuk program dinamik yang akan diselesaikan dengan perhitungan maju (*fordward recursive*), maka perhitungan tahap untuk mendapatkan keputusan optimal di mulai dari keadaan masalah yang pertama ke masalah yang terakhir.

   $f_0(x_0) = 0$

   $f_n^*(s_n) = \text{opt } \{ p_n(x_n) \otimes f^*_{n-1}(s_n \otimes x_n)\}$ n = 1, …, N

   dimana :

   $f_n^*(s_n)$: fungsi optimum

   s : state (status)

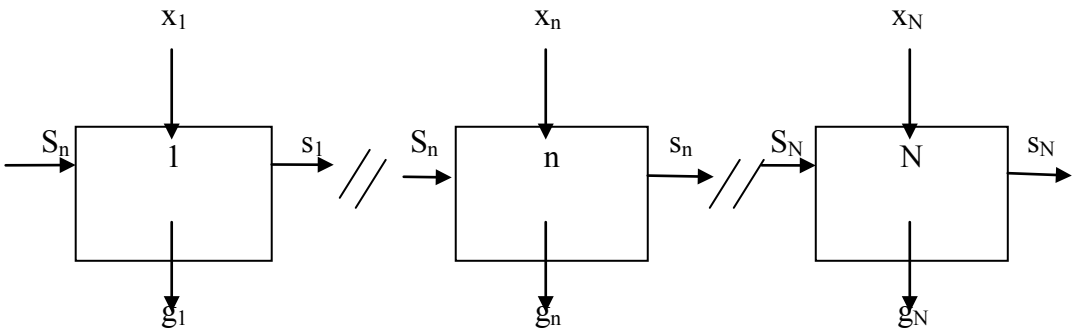
   $s_n \otimes x_n$: fungsi transisi

   n : tahap ke

x : variabel keputusan.

N : banyaknya tahap

Aidawayati Rangkuti (2013) menunjukkan konsep keadaan pada Rekursif mundur (*backward recursive)* pada gambar dibawah ini :



Gambar 2.5. konsep keadaan rekursif maju

perbedaan pokok antara metode *forward* dan *backward* terletak dalam cara mendefinisikan *state.* Simbol ⊗ menyatakan hubungan matematik antara s_n dengan x_n, misalnya tambah, kurang, kali, akar dan lain-lain.

Dengan menggunakan hubungan rekursif ini, prosedur penyelesaian bergerak dari tahap ke tahap sampai kebijaksanaan optimum tahap terakhir ditemukan. Sekali kebijaksanaan optimum tahap *n* telah ditemukan, *n* komponen keputusan dapat ditemukan kembali dengan melacak balik melalui fungsi transisi tahap *n*.

2.5. Formulasi Problema Program Dinamik

Sesuai dengan permasalahan yang telah dirumuskan maka yang akan menjadi tujuan utama adalah menetapkan tingkat pengadaan tenaga kerja berdasarkan penambahan/pengurangan tenaga kerja yang optimal untuk setiap shift sesuai dengan kebutuhan.

Selanjutnya notasi yang digunakan :

N = Banyaknya tahap

n = nomor tahap, n = 1, 2, 3,…, n

x_n = banyaknya tenaga kerja yang dialokasikan ke tahap n

s_n = banyaknya tenaga kerja yang masih tersedia untuk dialokasikan pada shift (tahap) yang tersisa ke tahap (1,…,n)

$p_i(x_i)$ = jumlah tenaga kerja untuk pengalokasian x_i orang ke shift i.

Dibawah ini perlu diterangkan beberapa terminologi antara lain :

Tahap (stage) adalah masa pengadaan tenaga kerja (dalam hal ini disebut shift). Maka dengan demikian masalah ini memiliki tiga tahap sehingga nomor tahap tersebut adalah n = 1, 2, 3

Keadaan (state) adalah alternatif-alternatif dalam setiap tahap. Dalam hal ini tingkat pengadaan tenaga kerja adalah variabel keadaan (s_i)

2.6. Pengambilan Keputusan

Prosedur pemecahan dalam program dinamik dilakukan secara rekursif. Ini berarti bahwa setiap kali mengambil keputusan harus memperhatikan keadaan yang dihasilkan oleh keputusan sebelumnya. Karena itu, keadaan yang diakibatkan oleh keputusan didasarkan pada keadaan dari keputusan sebelumnya dan merupakan landasan bagi keputusan berikutnya. Sehingga konsep tentang keadaan adalah sangat penting sekali.

Karena keadaan adalah berubah dari tahap ke tahap berikutnya maka nilai setiap tahap akan menggambarkan kondisi dari satu proses keputusan mengubah keadaan lama (awal) menjadi keadaan baru (akhir). Keadaan baru menjadi landasan bagi keputusan baru, dan keputusan baru mengubah keadaan baru (awal) menjadi lebih baru lagi (akhir), demikian seterusnya proses ini berlangsung. Karenanya hasil yang diharapkan dari satu keputusan tergantung dari awal dan akhir dari keadaan untuk keputusan tersebut dan kemudian menjumlahkan seluruhnya sebagai satu rangkaian keputusan yang maksimumkan hasil atau perolehan.

Pengambilan keputusan tidak lain dari penentuan x_i dan s_i sehingga f_i optimal.

Tujuan dari penulisan ini menemukan $x_1, x_2, x_3, \ldots, x_n$ sehingga :

mengoptimalkan jumlah tenaga perawat = opt $\sum_{i=1}^{n} p_i x_i$

dimana: $p_i(x_i)$ adalah jumlah tenaga kerja untuk pengalokasian x_i orang ke shift i.

Dengan kendala :

$$\sum_{1}^{n} x_i = s_n$$

x_i adalah bilangan bulat tidak negatif

Dengan menggunakan notasi, maka $f_n(s_n, x_n)$ adalah

$$f_n(s_n, x_n) = p_n(x_n) + \text{opt} \sum_{i=n+1}^{n} p_i(x_i)$$

Dimana : $f_n(s_n, x_n)$: menunjukkan kontribusi tahap n

$p_n(x_n)$: jumlah tenaga kerja untuk pengalokasian tahap n

dan

$$f_n^*(s_n) = \underset{x_{n=0,1,\ldots,s_n}}{\text{opt}} f_n(s_n, x_n)$$

jadi

$f_n(s_n, x_n) = p_n(x_n) + f_{n+1}^*(s_n - x_n)$

dimana : $f_{n+1}^*(s_n - x_n)$ menunjukkan kontribusi yang optimal.

Akibatnya, hubungan rekursif yang berhubungan dengan fungsi $f_1^*, f_2^*, \ldots, f_n^*$ untuk masalah ini adalah :

$$f_n^*(s_n) = \underset{x_{n=0,1,\ldots,s_n}}{\text{opt}} \{ p_n(x_n) + f_{n+1}^*(s_n - x_n) \} \text{ untuk } n = 1, 2, \ldots, n$$