

BAB I

PENDAHULUAN

1.1 Latar Belakang

Di era globalisasi yang semakin kompetitif ini mengharuskan kita sebagai pelaku utama kehidupan untuk senantiasa bekerja keras dan pandai dalam mengambil peluang. Berdasarkan uraian diatas maka dibutuhkan kondisi badan yang sehat dan prima guna menjawab tantangan kehidupan yang semakin beragam tersebut. Dengan tubuh yang sehat akan mempermudah kita untuk mendapat apa yang kita ingin dan cita-citakan. Maka dari itu kesehatan adalah salah satu kebutuhan primer yang mutlak untuk kita jaga. Ketika manusia sakit, segala cara mereka lakukan untuk memperoleh kesembuhan. Salah satu cara yang dilakukan adalah dengan berobat ke rumah sakit. Berdasarkan undang-undang No. 44 Tahun 2009 tentang rumah sakit, yang dimaksudkan dengan rumah sakit adalah institusi pelayanan kesehatan yang menyelenggarakan pelayanan kesehatan perorangan yang menyediakan pelayanan rawat inap, rawat jalan, dan gawat darurat.

RSUD Dr. Iskak adalah salah satu rumah sakit umum yang ada di Kabupaten Tulungagung. Rumah sakit ini beralamatkan di Jalan Dr. Wahidin Sudirohusodo, Kedungwaru, Tulungagung. Dengan fasilitas yang cukup lengkap dan daya dukung SDM yang memenuhi standar memungkinkan untuk rumah sakit ini dapat melayani pasien dengan semaksimal mungkin. Untuk memperlancar pelayanan pasien, dibutuhkan pengelolaan yang baik mengenai penyedia layanan darurat pada rumah sakit. Dalam hal ini penyedia layanan darurat yang dimaksud adalah sarana transportasi ambulans. Ketersediaan ambulans pada rumah sakit sangatlah penting. Sebagai salah satu contoh, armada ini dibutuhkan ketika pasien diharuskan untuk dirujuk ke rumah sakit lain ataupun ketika pasien yang telah meninggal harus dipulangkan ke rumah duka.

Terdapat beberapa metode yang dapat digunakan untuk menghasilkan keputusan maksimum. Salah satunya adalah dengan metode program dinamik. Program dinamik adalah suatu teknik matematis yang digunakan dalam pengambilan keputusan yang terdiri dari banyak tahap (*multistage*). Suatu masalah yang *multistage* dipisahkan menjadi masalah yang lebih kecil (subbagian) yang berurutan dan saling berhubungan (Hillier dan Lieberman, 2005).

Dalam program dinamik, prosedur perhitungannya adalah sistematis dan memaksimalkan seluruh sumber daya yang ada untuk menentukan kombinasi pengambilan keputusan sehingga pada akhirnya diperoleh keputusan yang paling optimal. Program dinamik telah banyak diterapkan dalam berbagai masalah bisnis dan perindustrian, misalnya analisa *network*, penelitian dan pengembangan.

Pada skripsi sebelumnya, Widiastuti (2006) telah membahas aplikasi dari program dinamik ini pada pengendalian persediaan bahan baku, sedangkan skripsi ini membahas tentang aplikasi dari program dinamik dalam masalah pembagian armada ambulan. Berdasarkan uraian diatas, dalam skripsi ini akan dibahas salah satu masalah yang dapat diselesaikan menggunakan program dinamik yaitu model dari masalah pembagian armada ambulan, serta solusi optimal yang dapat diperoleh dari masalah tersebut.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, pokok permasalahan dalam skripsi ini adalah sebagai berikut.

Bagaimanakah solusi optimal yang diperoleh dari masalah pembagian armada ambulan menggunakan metode program dinamik.

1.3 Tujuan

Tujuan dari skripsi ini adalah sebagai berikut.

Mengetahui solusi optimal yang diperoleh dari masalah pembagian armada ambulan menggunakan metode program dinamik.

1.4 Manfaat

Manfaat dari skripsi ini adalah sebagai berikut.

1. Mengetahui dan memahami ilmu yang didapat dari perkuliahan yang kemudian menerapkannya, sehingga dapat dibandingkan antara teori dan penerapannya.
2. Memberikan pengetahuan tentang penyelesaian masalah dengan menggunakan program dinamik.

1.5 Asumsi-asumsi

Adapun asumsi-asumsi dasar dari skripsi ini adalah:

1. Seluruh armada tidak mengalami kerusakan sehingga tidak mengganggu pengiriman armada ambulans.
2. Tidak ada penolakan dan sistem layanan bersifat FCFS (*First Come first Served*).
3. Permintaan layanan bersifat acak.
4. Kondisi dan tingkat kepadatan jalan diabaikan.

1.6 Batasan Masalah

Adapun batasan masalah dari skripsi ini adalah:

1. Penelitian bertempat di RSUD Dr. Iskak Tulungagung dan menggunakan data historis selama bulan Januari – Desember 2012.
2. Ambulan yang diamati adalah semua jenis ambulan yang tersedia di rumah sakit (sebanyak 8 armada).
3. Ruang lingkup dari penelitian ini adalah sebatas wilayah Karesidenan Kediri.

UNIVERSITAS BRAWIJAYA



BAB II TINJAUAN PUSTAKA

2.1 Definisi Program Dinamik

Program dinamik (*dynamic programming*) adalah suatu kumpulan teknik-teknik programasi matematis yang digunakan untuk pengambilan keputusan yang terdiri dari banyak tahap (*multistage*). Suatu masalah pengambilan keputusan yang *multistage* dipisahkan menjadi suatu seri masalah (atau submasalah) yang berurutan dan saling berhubungan (Subagyo, P., dkk., 2000). Sementara itu, menurut Taha, program dinamik adalah program matematis yang terutama dirancang untuk memperbaiki efisiensi perhitungan masalah program matematis tertentu dengan menguraikan menjadi bagian-bagian masalah yang lebih kecil, sehingga lebih sederhana dalam perhitungan.

Pada aplikasi yang sering dijumpai, masalah program dinamik diselesaikan dengan menggunakan perhitungan mundur, dari akhir masalah menuju awal, dengan memecah masalah yang luas menjadi bagian-bagian kecil sehingga masalah tersebut lebih mudah dikerjakan. Program dinamik pada umumnya menjawab masalah dalam *stage*, dengan setiap *stage* meliputi tepat satu variabel optimasi. Perhitungan di *stage* yang berbeda-beda dihubungkan melalui perhitungan rekursif dengan cara menghasilkan pemecahan optimal yang memungkinkan bagi seluruh masalah.

2.2 Proses Keputusan dengan Banyak Tahap/Stage

Setelah mengetahui definisi program dinamik dan sebelum mempelajari lebih jauh harus dipahami mengenai *multistage decision processes* atau proses keputusan dengan banyak tahap, karena proses inilah yang digunakan dalam penyelesaian masalah pengambilan keputusan dengan program dinamik.

Multistage decision processes adalah suatu proses yang dapat dipisahkan ke dalam banyak urutan langkah atau *stage*, yang mungkin dapat dilengkapi dalam satu atau banyak cara. Pilihan untuk melengkapi tahap/*stage* ini disebut keputusan. Kebijakan adalah urutan dari keputusan, satu untuk semua *stage* dari proses tersebut.

Kondisi suatu proses pada tiap *stage* yang diberikan disebut sebagai keadaan/*state* dari tiap *stage*, setiap pengaruh dari keputusan beralih dari *state* sekarang ke *state* berikutnya dengan *stage* berikutnya. Proses keputusan dengan banyak *stage* ini memiliki keuntungan dengan setiap keputusan dan keuntungan itu mungkin berbeda baik *stage* dan *state* dari proses tersebut. Tujuan dari proses analisis adalah menentukan kebijakan optimal, dari satu hasil adalah hasil total yang terbaik. (Bronson, 1983)

Sehingga, inti dari teknik ini adalah membagi satu persoalan atas beberapa bagian persoalan yang dalam program dinamik disebut *stage*, kemudian memecahkan tiap *stage* dengan mengoptimalkan keputusan atas tiap *stage* sampai seluruh persoalan terpecahkan. Keputusan optimal atas seluruh persoalan adalah kumpulan dari sejumlah keputusan optimal atas seluruh *stage* yang kemudian disebut sebagai kebijakan optimal. (Siagian, 1987)

2.3 Ciri-ciri Pokok Program Dinamik

Program dinamik adalah suatu pendekatan matematis tentang optimasi proses banyak tahap/*stage*. Konsep dasarnya diungkapkan dalam *Principle of Optimality* oleh Bellman dalam Mulyono (2002). Ciri-ciri dasar dari suatu masalah program dinamik adalah.

- a. Dalam masalah program dinamik, keputusan tentang suatu masalah ditandai dengan optimasi pada *stage* berikutnya, bukan keserantakan. Hal ini berarti jika suatu masalah akan diselesaikan dengan program dinamik, maka harus dipisahkan menjadi n submasalah.
- b. Program dinamik berkaitan dengan masalah-masalah dimana pilihan atau keputusan dibuat pada masing-masing *stage*. Seluruh kemungkinan diatur oleh sistem status atau *state* pada setiap *stage*.
- c. Berkaitan dengan setiap keputusan pada setiap *stage* adalah *return function* yang mengevaluasi pilihan yang dibuat dalam arti sumbangan yang diberikan kepada tujuan keseluruhan (maksimasi atau minimasi).
- d. Pada setiap *stage* proses keputusan dihubungkan dengan *stage* yang berdekatan melalui fungsi transisi (*transition function*).

Fungsi ini dapat berupa kuantitas yang diskrit maupun kontinu tergantung pada sifat masalah.

- e. Suatu hubungan rekursif digunakan untuk menghubungkan kebijaksanaan optimum pada *stage n* dengan *n-1*. Ada dua macam prosedur rekursif yaitu *forward* dan *backward*.

Hubungan tersebut adalah:

Forward recursive equation (perhitungan dari depan ke belakang).

$$f_0(X_0) = 0 \quad (2.1)$$

$$f_j^*(X_j) = \text{maks} \{R_j(k_j) @ f_{j-1}^*(x_j @ k_j)\} \quad (2.2)$$

Backward recursive equation (perhitungan dari belakang ke depan).

$$f_n(Y_n) = 0 \quad (2.3)$$

$$f_j^*(Y_j) = \text{maks} \{R_j(k_j) @ f_{j+1}^*(Y_j @ k_j)\} \quad (2.4)$$

keterangan:

- $f_j^*(X_j)$ atau $f_j^*(Y_j)$ = *optimum return function*
- X atau Y = *state*
- $X @ k$ atau $Y @ k$ = fungsi transisi
- j = *stage* ke n
- k = variabel keputusan

Simbol @ menyatakan hubungan matematis antara X_j dan Y_j dengan k_j , misalnya tambah, kurang, kali, atau akar, dan lain-lain.

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

2.4 Prinsip Program Dinamik

Teori utama dalam program dinamik adalah prinsip optimalitas (Taha, 1996). Prinsip ini pada dasarnya menentukan bagaimana suatu masalah yang diuraikan dengan benar dapat dijawab dalam *stage* (bukan sebagai satu kesatuan) melalui pemakaian perhitungan rekursif.

Pendekatan program dinamik didasarkan pada prinsip optimalitas Bellman dalam Siagian (1987) yang mengatakan:

“Suatu kebijakan optimal mempunyai sifat bahwa apapun keadaan/state dan keputusan awal, keputusan berikutnya harus membentuk suatu kebijakan optimal dengan memperhatikan state dari hasil keputusan pertama.”

2.5 Pernyataan Matematik Program Dinamik

Masalah program dinamik dapat dinyatakan dalam bentuk umum (Subagyo, P., dkk, 2000):

Maksimumkan:
$$f_n(X) = \sum_{j=1}^n r_j(X_j), \quad (2.5)$$

Dengan batasan:
$$X = \sum_{j=1}^n X_j, \quad (2.6)$$

dan
$$X_j \geq 0 \quad (j= 1, 2, 3, \dots, n),$$

dimana:

$f_n(X)$ = perolehan total dari seluruh kegiatan (*stage*),

X_j = kuantitas sumber daya yang dialokasikan ke kegiatan (*stage*) ke j ,

$r_j(X_j)$ = perolehan dari kegiatan ke j ,

n = jumlah kegiatan (*stage*) bebas (*independent*),

X = sumber daya total yang tersedia untuk n kegiatan.

Dalam masalah umum diatas, perolehan maksimal dari seluruh kegiatan ditentukan oleh sumber daya total X yang tersedia dan

perolehan dari kegiatan-kegiatan individual $r_j(X_j)$. Oleh sebab itu, perolehan keseluruhan dari n kegiatan-kegiatan dapat dinyatakan oleh suatu urutan fungsi-fungsi sebagai berikut:

$$f_n(X) = \max_{X_j} f(X_1, X_2, \dots, X_{n-1}, X_n) \quad (2.7)$$

Sumberdaya total yang tersedia X harus dialokasikan secara berurutan ke semua kegiatan-kegiatan pada *stage* yang berbeda, untuk mencapai hasil yang maksimal. Bila dialokasikan sejumlah X_n dari sumberdaya ke kegiatan ke- n , dimana $0 \leq X_n \leq X$, akan didapatkan penghasilan (f_n, X_n) dari kegiatan tersebut. Masalah yang dipunyai sejumlah (X, X_n) sumberdaya yang tersedia untuk $(n - 1)$ kegiatan. Bila penghasilan total dari $(n - 1)$ kegiatan ditunjukkan oleh:

$$f_{n-1}(X - X_n) = \sum_{j=1}^{n-1} r_j(X_j), X_j \geq 0. \quad (2.8)$$

Penghasilan total dari n kegiatan dapat dinyatakan sebagai berikut:

$$f_n(X) = r_n(X_n) + f_{n-1}(X - X_n). \quad (2.9)$$

Kuantitas sumberdaya optimal yang dialokasikan ke n kegiatan, X_n menentukan nilai $(X - X_n)$, dan hal ini sebaliknya akan menentukan nilai maksimal persamaan penghasilan total. Oleh sebab itu, masalah program dinamik dapat dinyatakan dalam bentuk fungsi umum sebagai berikut:

$$f_n(X) = \max \{r_n(X_n) + f_{n-1}(X - X_n)\} n = 2, 3, \dots \quad (2.10)$$

Untuk menentukan keputusan optimal dari seluruh persoalan dimulai dari keputusan optimal dari *stage* n yaitu *stage* terakhir dan kemudian menelusuri keputusan optimal pada *stage* sebelumnya. Cara yang dilakukan adalah sebagai berikut:

Diketahui $x_n = B$ pada *stage* n sehingga keputusan optimal untuk *stage* n dapat ditentukan, misalnya pada alternatif k . tentu b_{nk} , yaitu biaya yang diperlukan untuk alternatif k sudah dapat diketahui. Karena itu:

$$\begin{aligned}x_{n-1} &= X_n - b_{nk} \\ &= B - b_{nk}\end{aligned}\quad (2.11)$$

Setelah melakukan perhitungan pada *stage* $(n-1)$, keputusan optimal pada *stage* ini dapat ditentukan sesuai jumlah x_{n-1} . Misalkan keputusan jatuh pada alternatif j dan karena itu $b_{n-1,j}$ pun dapat diketahui, sehingga:

$$\begin{aligned}x_{n-2} &= X_{n-1} - b_{n-1,j} \\ &= (B - b_{nk}) - b_{n-1,j}\end{aligned}\quad (2.12)$$

Proses ini dilanjutkan terus, hingga pada akhirnya diperoleh harga x_1 dimana dapat ditentukan keputusan optimal pada *stage* kesatu. Keputusan optimal dari masing-masing *stage*. Jadi jelaslah bahwa untuk menyelesaikan persoalan dengan program dinamik harus dilakukan dengan format yang seragam untuk semua tahap/*stage*, artinya tiap perolehan f_j ditentukan dari $j-1$. Pendekatan inilah yang disebut pendekatan rekursif.

2.6 Langkah-langkah Program Dinamik

Adapun langkah-langkah dalam penyelesaian masalah program dinamik adalah:

1. membagi masalah awal menjadi submasalah-submasalah yang dinamakan *stage*,
2. memecahkan *stage* akhir untuk seluruh kemungkinan keadaan atau *state*,
3. perhitungan ke belakang dari *stage* akhir, memecahkan tiap bagian *stage*. Perhitungan selesai dengan menentukan keputusan optimal dari *stage* tersebut menuju *stage* akhir, dan
4. didapatkan solusi optimal dari masalah awal dengan menyelesaikan setiap *stage* secara berurutan.

2.7 Karakteristik Dasar Program Dinamik

1. Tahap/*stage*, merupakan suatu bagian atau submasalah.
2. *Variable stage*, dikatakan juga sebagai *variable input*.
3. Variabel keputusan merupakan alternatif atau keputusan yang paling mungkin di tiap *state*.
4. Ukuran keputusan, merupakan pernyataan mengenai tujuan suatu masalah.
5. Kebijakan optimal, merupakan himpunan keputusan yang merupakan hasil dari kriteria keputusan yang memberikan keputusan optimal dari setiap *state* di tiap *stage*.

2.8 Penguraian Masalah Program Dinamik

Konsep dasar pertama dari pendekatan program dinamik dalam perumusan masalah dan analisis adalah penguraian masalah ke dalam serangkaian submasalah yang lebih kecil. Setiap submasalah dianggap sebagai *stage* atau titik keputusan. Berdasarkan prinsip tersebut, masalah seringkali dapat dievaluasi dengan lebih mudah dan efisien melalui *stage* tersebut. (Lee,1990)

Pada umumnya persoalan yang membutuhkan penerapan program dinamik dapat dibagi ke dalam N *stage*. Untuk perhitungan dengan proses mundur, *stage* tersebut dapat disusun dari *stage* yang terbesar (N) sampai yang terkecil ($ke-1$) yang secara sistematis seperti dibawah ini:



Gambar 2.1 Perhitungan dengan Proses Mundur

Tanda panah diatas menunjukkan hubungan dari satu *stage* (submasalah) ke *stage* selanjutnya dan umumnya menggambarkan aliran informasi tentang status keadaan *stage* ke- N , *stage* $N-1$, dan seterusnya.

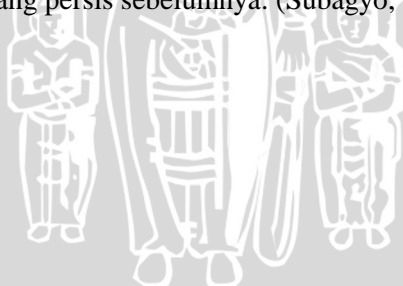
Jika kita menghitung dengan memulai dari $f_1(x_1)$ terus ke $f_2(x_2)$ dan berakhir pada $f_n(x_n)$ dimana $j= 1,2,..n$ maka dapat dikatakan perhitungan perolehan dilakukan dengan proses maju yaitu:



Gambar 2.2 Perhitungan dengan Proses Maju

Jadi perhitungan maju (*forward recursive*) adalah mendahulukan perhitungan mulai dari *stage* pertama sampai *stage* terakhir. Apabila perhitungan dimulai dari *stage* terakhir dan beralih mundur ke *stage* kesatu maka metode ini disebut perhitungan mundur (*backward recursive*). Perbedaan yang utama antara perhitungan maju dan mundur terdapat pada cara menetapkan keadaan/*state* dari sistem tersebut. (Taha, 1996)

Persamaan rekursif dapat digunakan baik untuk perhitungan ke depan maupun ke belakang dalam pemecahan masalah-masalah yang *multistage*. Kedua metode ini mengarahkan ke penyelesaian optimal yang sama dari suatu masalah program dinamik. Hal paling penting untuk diperhatikan, bahwa setiap penyelesaian dari submasalah digunakan sebagai I masukan (*input*) untuk penyelesaian submasalah berikutnya. Jadi, prosedur perhitungan meliputi hanya dua aspek yaitu submasalah sekarang yang sedang dalam perhitungan dan hasil perhitungan submasalah yang persis sebelumnya. (Subagyo, 1985)



2.9 Dekomposisi Pergandaan

Prinsip dari dekomposisi pergandaan, pada hakikatnya tidak berbeda dengan prinsip dekomposisi penjumlahan. Tergantung pada permasalahan yang dihadapi. Pada dekomposisi pergandaan, proses penyelesaian persoalan program dinamik dilakukan seperti berikut.

Misalkan terdapat j *stage* dan tiap *stage* mempunyai alternatif x_j dimana $j=1,2,\dots,n$ dan $x_j=1,2,\dots,n$. f_j adalah perolehan dari *stage* j pada alternatif x_j dengan jumlah biaya sebesar x_j . Maka f_j adalah fungsi dari x_j dan dapat ditulis dengan,

$$f_j = r_j(x_j). \quad (2.13)$$

Jika f adalah jumlah seluruh perolehan untuk semua *stage*, maka persoalan program dinamik dalam bentuk dekomposisi pergandaan adalah,

$$\text{Maks } f = \prod r_n(x_n). \quad (2.14)$$

Agar pendekatan rekursif dapat dilakukan sesuai dengan model komposisi pergandaan, model persoalan diubah dalam persamaan rekursif seperti berikut:

Misalkan x_j sebagai alokasi sebanyak j *stage*, karena itu x_j adalah *state* dari j tahap/*stage*. Dengan demikian x_j adalah *state* dari *stage* kesatu sekaligus diartikan sebagai alokasi untuk *stage* kesatu dan x_2 adalah *state* dari *stage* kesatu dan kedua. Karena perolehan optimal pada *stage* kesatu adalah fungsi dari alokasi x_j maka:

$$f_j(x_j) = \text{maks} \{r_j(X_j)\}. \quad (2.15)$$

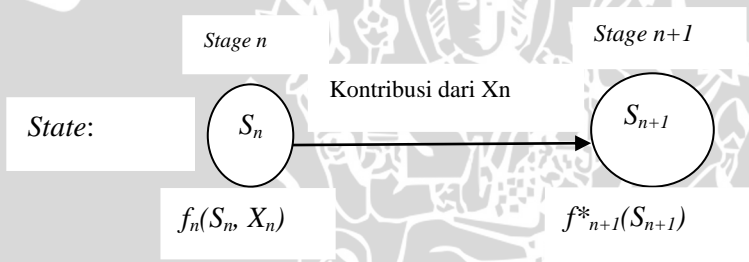
Secara umum dapat ditulis bahwa perolehan optimal untuk n *stage* adalah:

$$f_j(x_j) = \text{maks} \{r_j(X_j) \cdot f_{j-1}(X - X_j)\}. \quad (2.16)$$

(Siagian, 1987)

2.10 Masalah Deterministik Dan Stokastik

Program dinamik deterministik dapat diuraikan dengan diagram yang ditunjukkan pada gambar 2.1 (Hillier dan Lieberman, 2005). Pada *stage n* proses akan berada pada suatu *state* S_n , pembuatan keputusan kebijakan x_n selanjutnya menggerakkan proses ke *state* S_{n+1} pada *stage* $(n + 1)$. Kontribusi sesudahnya terhadap fungsi tujuan dibawah kebijakan optimal telah dihitung sebelumnya menjadi $f_{n+1}^*(S_{n+1})$. Keputusan kebijakan terhadap x_n juga memberi beberapa kontribusi kepada fungsi tujuan. Kombinasi kedua nilai ini dengan benar akan memberikan $f_n(S_n, x_n)$, yaitu kontribusi *n stage* kedepan kepada fungsi tujuan. Pengoptimalan terhadap x_n mengakibatkan $f^*(S_n) = f_n(S_n, x_n^*)$. Setelah ditemukan x_n^* dan $f_n^*(S_n)$ untuk setiap nilai S_n , prosedur penyelesaian sekarang siap bergerak mundur satu *stage*.

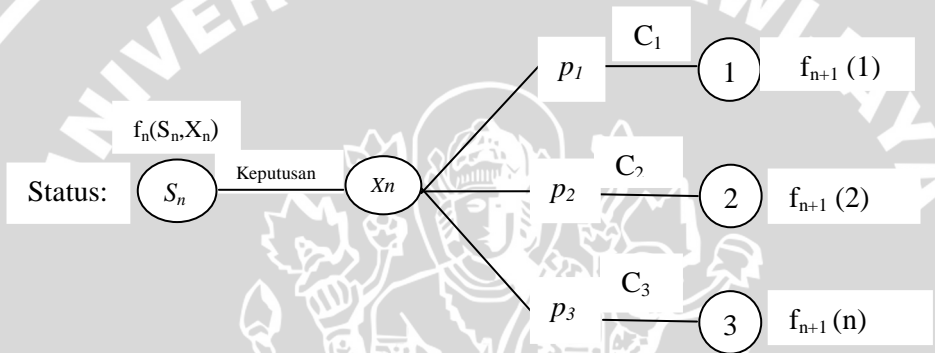


Gambar 2.3 Ilustrasi Umum *n Stage* Persoalan Program

Salah satu cara mengategorikan masalah program dinamik deterministik adalah dengan bentuk fungsi tujuan. Sebagai contoh, tujuannya adalah meminimumkan jumlah kontribusi dari *stage* individu, atau untuk memaksimumkan jumlah tersebut, atau untuk meminimumkan suatu produk, dan seterusnya. Pengkategorian lain adalah dalam bentuk sifat alamiah dari kumpulan *state* untuk *stage* yang bersesuaian. Khususnya *state* S_n dapat dinyatakan oleh peubah *state* diskrit (seperti pada masalah perjalanan armada), atau oleh peubah *state* kontinu, atau mungkin diperlukan suatu vektor *state* (lebih dari satu peubah).

Program dinamik stokastik yaitu program dinamik dengan suatu distribusi probabilitas untuk ketetapan dalam *stage* keputusan yang berurutan (Subagyo, P., dkk., 2000). *Stage* berikutnya tidak dapat seluruhnya ditentukan oleh status dan keputusan pada *stage* saat ini, tetapi ada suatu distribusi kemungkinan mengenai apa yang akan terjadi.

Struktur dasar program dinamik stokastik ditunjukkan pada gambar 2.4. Untuk gambar berikut indeks p_1, p_2, p_3 merupakan suatu distribusi kemungkinan (peluang) mengenai apa yang akan terjadi pada *stage* berikutnya.



Gambar 2.4 Struktur Dasar Program Dinamik Stokastik

UNIVERSITAS BRAWIJAYA



BAB III METODOLOGI

3.1 Obyek Penelitian

Berkaitan dengan judul penelitian yang diambil, maka penulis mengambil obyek penelitian di RSUD Dr. Iskak yang beralamatkan di Jalan Dr. Wahidin Sudiro Husodo, Kedungwaru, Tulungagung. Dipilihnya RSUD Dr. Iskak sebagai obyek penelitian karena rumah sakit ini memiliki fasilitas dan pelayanan yang cukup lengkap terhadap pasien. Rumah sakit ini memiliki 8 armada ambulan yang meliputi 4 armada ambulan untuk kereta jenazah dan 4 armada lain untuk ambulan pasien. Dalam penelitian ini ruang lingkup yang diteliti adalah wilayah Karesidenan Kediri meliputi:

1. Kota Kediri
2. Kabupaten Kediri
3. Kota Blitar
4. Kabupaten Blitar
5. Kabupaten Tulungagung
6. Kabupaten Trenggalek
7. Kabupaten Nganjuk

Pengumpulan data sekunder dan data primer yang dilakukan di RSUD Dr. Iskak ini dilaksanakan pada tanggal 26 Februari – 28 Februari 2013 pada pukul 08.00 WIB sampai pukul 12.00 WIB.

3.2 Sumber Data

Data yang digunakan pada skripsi ini adalah data historis mulai dari bulan Januari sampai dengan Desember 2012. Sumber data yang digunakan pada penelitian ini adalah:

1. Data primer.
Yaitu sumber data yang langsung memberikan data kepada peneliti. Sumber data ini berasal dari wawancara pada bagian instalasi ambulan dan bagian humas dari RSUD Dr. Iskak. Adapun data yang di dapat berupa sistem pelayanan ambulan mulai dari permintaan datang, persiapan armada, hingga armada siap diberangkatkan ke tempat tujuan.
2. Data sekunder.
Yaitu sumber yang tidak langsung memberikan data kepada peneliti, misalnya melalui data yang diperoleh dari mencatat dan

mempelajari arsip-arsip atau dokumen-dokumen instansi yang berkaitan dengan obyek penelitian.

Adapun data sekunder yang dibutuhkan adalah sebagai berikut.

- a. Jumlah permintaan layanan ambulans di rumah sakit selama satu tahun dalam lingkup wilayah Karesidenan Kediri.
- b. Jumlah permintaan layanan pengiriman pada setiap ambulans yang ada selama satu tahun.

3.3 Prosedur Pengumpulan atau Perekaman Data

Pengumpulan data yang dilakukan pada penelitian ini terdiri dari beberapa tahap, yaitu sebagai berikut.

1. Penelitian Langsung ke Lapangan atau Instansi (field research)
Tujuan dari penelitian secara langsung ke instansi adalah untuk memperoleh data-data yang mendukung proses penelitian dan dapat memahami permasalahan yang ada di perusahaan secara langsung. Stage pengumpulan data dengan penelitian secara langsung ke perusahaan dapat dilakukan dengan berbagai cara, yaitu.
 - a. Wawancara
Pengumpulan data dengan cara wawancara dapat dilakukan dengan melakukan komunikasi secara langsung dengan pihak perusahaan mengenai obyek penelitian.
 - b. Dokumentasi
Data-data yang diperoleh dari dokumentasi merupakan data sekunder. Hal tersebut dikarenakan data didapat dari data-data penelitian. Pengumpulan data dengan dokumentasi dilakukan dengan mempelajari data yang berhubungan dengan obyek penelitian yang terdapat di instansi.
2. Studi Literatur
Tahapan literatur dapat membantu menyelesaikan permasalahan di instansi dengan menggunakan teori-teori yang ada.

3.4 Analisis Data

Setelah semua data yang diperlukan terkumpul, maka langkah selanjutnya adalah melakukan analisis terhadap data-data untuk memecahkan masalah yang sedang diteliti. Metode analisis

merupakan langkah-langkah yang dilakukan untuk mengolah semua data.

Dalam penelitian ini, data yang ada di analisis dengan menggunakan program dinamik untuk menentukan jumlah optimal pembagian armada ambulan yang dapat diberikan RSUD Dr. Iskak selama satu tahun.

Untuk masalah pembagian armada ini merupakan masalah deterministik. *Stage* berikutnya dapat seluruhnya ditentukan oleh status dan keputusan pada *stage* saat ini.

Adapun langkah-langkah analisis yang digunakan dalam penelitian ini adalah sebagai berikut.

1. Menentukan peubah keputusan.
Pada kasus RSUD Dr. Iskak, peubah keputusannya adalah banyaknya permintaan layanan pengiriman yang dapat dilayani di daerah tertentu dalam satu bulan yaitu b_{nm_n} (n dan $m= 1, 2, 3, 4, 5, 6, 7$)
2. Menentukan *State*.
State I_{nm} adalah jumlah armada ambulan yang digunakan dalam satu tahun.
3. Menentukan tujuan.
Misal $r_{nm_n}(b_{nm_n})$ adalah pelayanan ambulan pada *stage* tertentu, maka tujuan pada kasus ini adalah mencari solusi dalam pembagian armada ambulan dalam 7 daerah yang berbeda.
4. Menentukan hubungan rekursif yang sesuai.
Pada masalah optimasi pembagian armada ambulan persamaan rekursif yang digunakan adalah rekursif maju, sehingga perhitungan dimulai dari *stage* ke-1 bergerak ke belakang hingga *stage* ke-7.
5. Melakukan perhitungan terhadap data berdasarkan hubungan rekursif untuk memperoleh hasil yang optimal.

Contoh perhitungan sebagai berikut.

Formulasi program dinamik untuk permasalahan pembagian armada ambulan ini adalah sebagai berikut:

$Stage\ n$ = Daerah tujuan pengiriman armada ambulan
($n = 1, 2, \dots, 7$),

b_n = Banyaknya armada ambulan yang dialokasikan pada tiap daerah,

$State\ S_n$ = Banyaknya armada ambulan yang masih tersedia untuk dialokasikan pada $stage\ n$.

Masalah ini adalah tujuh $stage$ masalah program dinamik. Penyelesaian optimal tiap $stage$ dapat diperoleh dengan menggunakan persamaan rekursif:

$$f_{nm_n}(X_{nm_n}) = \min f_{nm_n}(X_{nm_n}, b_{nm_n}), \quad (3.1)$$

untuk $stage\ 1$,
dan

$$f_{nm_n}(X_{nm_n}, b_{nm_n}) = \min \{R_{nm_n}(b_{nm_n}) \cdot f_{n-1}^*(X_n - b_{nm_n})\}, \quad (3.2)$$

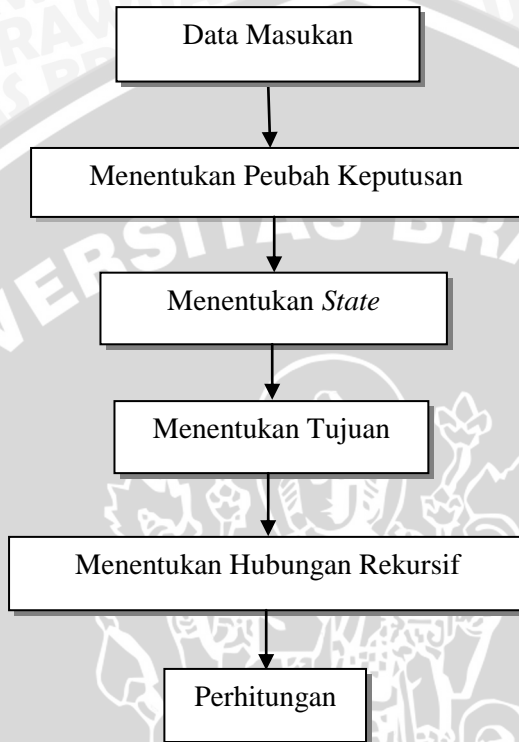
untuk $stage\ 2, 3, \dots, 7$.

Persamaan rekursif untuk masalah ini adalah:

$$f_{nm_n}^*(X_{nm_n}) = \min_{x=0,1,2,\dots,S_n} \{R_{nm_n}(b_{nm_n}) \cdot f_{n-1}(X_n - b_{nm_n})\}. \quad (3.3)$$

Setelah memperoleh persamaan rekursif diatas barulah menentukan nilai dari $f_1(X_1)$ terlebih dahulu, karena akan mempengaruhi nilai $f_2(X_2)$. Kemudian berurutan menentukan nilai $f_3(X_3), f_4(X_4), f_5(X_5), f_6(X_6)$, hingga $f_7(X_7)$.

Secara ringkas pengolahan data dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram Alir Pengolahan Data

3.5 Rancangan Penelitian

Penelitian secara operasional yang dilakukan untuk mencari solusi optimal dari layanan pengiriman ambulans secara efektif dan efisien, terdiri dari langkah-langkah berikut.

1. Identifikasi Masalah

Identifikasi masalah dilakukan untuk mengetahui masalah yang ada sehingga dapat diselesaikan menggunakan metode yang ada.

2. Studi Literatur

Studi kasus dilakukan dengan mempelajari ilmu pengetahuan yang berhubungan dengan permasalahan yang terjadi untuk mencari solusi yang tepat dalam menyelesaikannya.

3. Perumusan Masalah

Perumusan masalah dilakukan dengan mencari poin-poin dari permasalahan yang akan diselesaikan. Perumusan masalah ini dilakukan setelah melakukan studi literatur yang berkaitan dengan permasalahan yang ada.

4. Pengumpulan Data

Setelah merumuskan masalah, langkah selanjutnya adalah mengumpulkan data dan informasi yang diperlukan untuk menyelesaikan masalah pembagian armada ambulans menggunakan metode program dinamik. Dalam *stage* ini dilakukan pengumpulan data yang meliputi.

a. Jumlah permintaan layanan ambulans di rumah sakit selama satu tahun dalam lingkup wilayah Karesidenan Kediri.

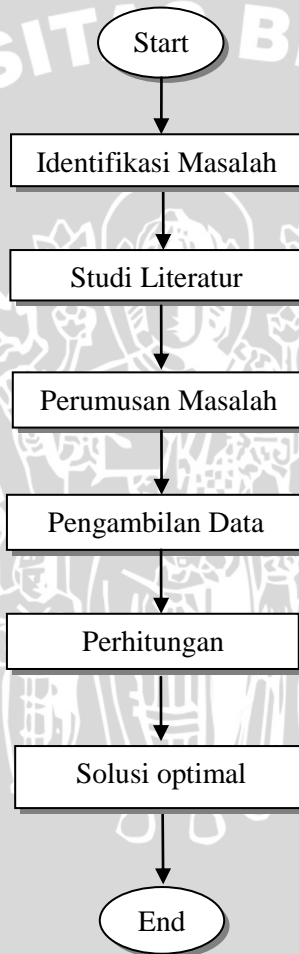
b. Jumlah permintaan layanan pengiriman pada setiap ambulans yang ada selama satu tahun.

Data-data tersebut digunakan untuk menghitung optimalisasi pembagian armada ambulans ambulans.

5. Pengolahan Data

Terdapat tujuh *stage* perhitungan dalam model pembagian armada ambulans. Dikarenakan model ini menggunakan metode program dinamik dengan rekursif maju, maka perhitungan berjalan dari *stage* ke-1 hingga *stage* ke-7.

6. Mensimulasikan Model Pembagian Armada Ambulan
Simulasi model pembagian armada ambulans di RSUD Dr. Iskak dengan metode program dinamik menggunakan *software Borland Delphi*.
7. Kesimpulan
Secara ringkas rancangan penelitian dan langkah pengerjaan dapat dilihat pada Gambar 3.2.



Gambar 3.2 Diagram Alir Rancangan Penelitian Metode Program Dinamik

UNIVERSITAS BRAWIJAYA



BAB IV HASIL DAN PEMBAHASAN

4.1 Hasil Penelitian

Hasil yang diperoleh dalam penelitian ini adalah data jumlah permintaan layanan ambulan di lingkup wilayah Karesidenan Kediri pada tahun 2012.

Tabel 4.1 Jumlah permintaan layanan ambulan di lingkup wilayah Karesidenan Kediri pada tahun 2012.

Ambulan	Daerah						
	1	2	3	4	5	6	7
1	29	21	21	18	27	18	17
2	28	17	27	13	25	19	19
3	14	18	20	21	30	14	15
4	19	24	18	19	21	21	15
5	27	12	17	16	19	20	19
6	30	17	19	14	25	12	14
7	31	19	25	15	28	10	16
8	18	14	29	17	23	17	18

Sumber data: Arsip data RSUD DR. Iskak, Tulungagung.

Keterangan:

Daerah 1: Kota Kediri

Daerah 2: Kabupaten Kediri

Daerah 3: Kota Blitar

Daerah 4: Kabupaten Blitar

Daerah 5: Kabupaten Tulungagung

Daerah 6: Kabupaten Trenggalek

Daerah 7: Kabupaten Nganjuk

4.2 Formulasi Model Secara Matematis

Fungsi tujuan yang diinginkan adalah untuk mencari optimalisasi pembagian 8 armada yang dimiliki RSUD Dr. Iskak untuk 7 daerah yang berbeda. Dikarenakan masalah pembagian armada ini menggunakan prinsip dekomposisi pergandaan maka bentuk fungsi tujuannya adalah sebagai berikut.

Adapun rumus rekursif untuk masalah ini adalah,

$$f_n(x_n) = \min \{R_{nm_n}(b_{nm_n})\}, \text{ dan} \quad (4.1)$$

$$f_n(x_n) = \min \{R_{nm_n}(b_{nm_n}) \cdot f_{n-1}(x_n - b_{nm_n})\} \quad (4.2)$$

dengan,

Fungsi tujuan,

minimumkan:

$$Z = \prod_{n=1}^7 r_{nm_n}(b_{nm_n}), \quad (4.3)$$

Fungsi pembatas,

$$B = \prod_{n=1}^7 b_{nm_n}, \quad (4.4)$$

dan $B_{nm_n} \geq 0$ dengan n dan $m_n = 1, 2, \dots, 7$

Keterangan:

- Z = Perolehan total yang di dapat dari ketujuh *stage*
- b_{nm_n} = Jumlah armada ambulan yang dialokasikan ketujuh *stage*
- $r_{nm_n}(b_{nm_n})$ = Penghasilan yang didapat dari kegiatan/*stage* ke n
- B = Jumlah total armada ambulan yang tersedia untuk tujuh kegiatan
- m_n = Jumlah armada ambulan yang dialokasikan di daerah n
- n = Daerah pengiriman

Persamaan Rekursif

1. Stage 1

$$\text{Rumus: } f_1(X_1) = \min \{R_1(b_1)\}$$

2. Stage 2

$$\text{Rumus: } f_2(X_2) = \min \{R_2(b_2) \cdot f_1(X_2 - b_2)\}$$

3. Stage 3

$$\text{Rumus: } f_3(X_3) = \min \{R_3(b_3) \cdot f_2(X_3 - b_3)\}$$

4. Stage 4

$$\text{Rumus: } f_4(X_4) = \min \{R_4(b_4) \cdot f_3(X_4 - b_4)\}$$

5. Stage 5

$$\text{Rumus: } f_5(X_5) = \min \{R_5(b_5) \cdot f_4(X_5 - b_5)\}$$

6. Stage 6

$$\text{Rumus: } f_6(X_6) = \min \{R_6(b_6) \cdot f_5(X_6 - b_6)\}$$

7. Stage 7

$$\text{Rumus: } f_7(X_7) = \min \{R_7(b_7) \cdot f_6(X_7 - b_7)\}$$

4.3 Hasil Perhitungan

Stage 1 (i=1)

Dengan rumus:

$$f_1(x_1) = \min \{R_{11}(b_{11}), R_{12}(b_{12}), R_{13}(b_{13}), R_{14}(b_{14}), R_{15}(b_{15}), \\ R_{16}(b_{16}), R_{17}(b_{17}), R_{18}(b_{18})\}$$

untuk $x_1=1$

$$f_1(1) = \min \{R_{11}(1), R_{12}(2), R_{13}(3), R_{14}(4), R_{15}(5), R_{16}(6), \\ R_{17}(7), R_{18}(8)\} \\ = \min \{29, -, -, -, -, -, -, -\} \\ = 29$$

Diperoleh $m_1^* = 1$

untuk $x_1=2$

$$f_1(2) = \min \{R_{11}(1), R_{12}(2), R_{13}(3), R_{14}(4), R_{15}(5), R_{16}(6), \\ R_{17}(7), R_{18}(8)\} \\ = \min \{29, 28, -, -, -, -, -, -\} \\ = 28$$

Diperoleh $m_1^* = 2$

untuk $x_1=3$

$$f_1(3) = \min \{R_{11}(1), R_{12}(2), R_{13}(3), R_{14}(4), R_{15}(5), R_{16}(6),$$

$$\begin{aligned}
 & R_{17}(7), R_{18}(8) \\
 & = \min \{29, 28, 14, -, -, -, -\} \\
 & = 14
 \end{aligned}$$

Diperoleh $m_1^* = 3$

untuk $x_1 = 4$	$f_1(4) = 14$	diperoleh $m_1^* = 3$
untuk $x_1 = 5$	$f_1(5) = 14$	diperoleh $m_1^* = 3$
untuk $x_1 = 6$	$f_1(6) = 14$	diperoleh $m_1^* = 3$
untuk $x_1 = 7$	$f_1(7) = 14$	diperoleh $m_1^* = 3$
untuk $x_1 = 8$	$f_1(8) = 14$	diperoleh $m_1^* = 3$

Nilai optimal pada *stage* 1 adalah sebesar 29. Hasil akhir perhitungan dapat dilihat lebih lengkap pada tabel *stage* 1 pada lampiran 1.

Stage 2 (i=2)

Dengan rumus:

$$\begin{aligned}
 f_2(x_2) = \min \{ & R_{21}(b_{21}) \cdot f_1(x_2 - b_{21}), R_{22}(b_2) \cdot f_1(x_2 - b_2), R_2(b_2) \cdot f_1(x_2 - \\ & b_{23}), R_{24}(b_{24}) \cdot f_1(x_2 - b_{24}), R_{25}(b_{25}) \cdot f_1(x_2 - b_{25}), R_{26}(b_{26}) \cdot \\ & f_1(x_2 - b_{26}), R_{27}(b_{27}) \cdot f_1(x_2 - b_{27}), R_{28}(b_{28}) \cdot f_1(x_2 - b_{28}) \}
 \end{aligned}$$

untuk $x_2 = 1$

$$\begin{aligned}
 f_2(1) = \min \{ & R_{21}(1) \cdot f_1(1-1), R_{22}(2) \cdot f_1(1-2), R_{23}(3) \cdot f_1(1-3), \\ & R_{24}(4) \cdot f_1(1-4), R_{25}(5) \cdot f_1(1-5), R_{26}(6) \cdot f_1(1-6), R_{27}(7) \cdot \\ & f_1(1-7), R_{28}(8) \cdot f_1(1-8) \} \\
 & = \min \{609, -, -, -, -, -, -, -\} \\
 & = 609
 \end{aligned}$$

Diperoleh $m_2^* = 1$

untuk $x_2 = 2$

$$\begin{aligned}
 f_2(2) = \min \{ & R_{21}(1) \cdot f_1(2-1), R_{22}(2) \cdot f_1(2-2), R_{23}(3) \cdot f_1(2-3), \\ & R_{24}(4) \cdot f_1(2-4), R_{25}(5) \cdot f_1(2-5), R_{26}(6) \cdot f_1(2-6), R_{27}(7) \cdot \\ & f_1(2-7), R_{28}(8) \cdot f_1(2-8) \} \\
 & = \min \{588, 493, -, -, -, -, -, -\} \\
 & = 493
 \end{aligned}$$

Diperoleh $m_2^* = 2$

untuk $x_2=3$	$f_2(3) = 294$	diperoleh $m_2^* = 1$
untuk $x_2=4$	$f_2(4) = 238$	diperoleh $m_2^* = 2$
untuk $x_2=5$	$f_2(5) = 238$	diperoleh $m_2^* = 2$
untuk $x_2=6$	$f_2(6) = 238$	diperoleh $m_2^* = 2$
untuk $x_2=7$	$f_2(7) = 168$	diperoleh $m_2^* = 5$
untuk $x_2=8$	$f_2(8) = 168$	diperoleh $m_2^* = 5$

Nilai optimal untuk *stage* 2 sebesar 609. Hasil akhir perhitungan dapat dilihat lebih lengkap pada tabel *stage* 2 pada lampiran 2.

Stage 3 (i=3)

Dengan rumus:

$$f_3(x_3) = \min \{ R_{31}(b_{31}) \cdot f_2(x_3-b_{31}), R_{32}(b_{32}) \cdot f_2(x_3-b_{32}), R_{33}(b_{33}) \cdot f_2(x_3-b_{33}), R_{34}(b_{34}) \cdot f_2(x_3-b_{34}), R_{35}(b_{35}) \cdot f_2(x_3-b_{35}), R_{36}(b_{36}) \cdot f_2(x_3-b_{36}), R_{37}(b_{37}) \cdot f_2(x_3-b_{37}), R_{38}(b_{38}) \cdot f_2(x_3-b_{38}) \}$$

untuk $x_3=1$

$$\begin{aligned} f_3(1) &= \min \{ R_{31}(1) \cdot f_2(1-1), R_{32}(2) \cdot f_2(1-2), R_{33}(3) \cdot f_2(1-3), \\ &R_{34}(4) \cdot f_2(1-4), R_{35}(5) \cdot f_2(1-5), R_{36}(6) \cdot f_2(1-6), R_{37}(7) \cdot \\ &f_2(1-7), R_{38}(8) \cdot f_2(1-8) \} \\ &= \min \{ 12789, -, -, -, -, -, -, - \} \\ &= 12789 \end{aligned}$$

Diperoleh $m_3^* = 1$

untuk $x_3=2$

$$\begin{aligned} f_3(2) &= \min \{ R_{31}(1) \cdot f_2(2-1), R_{32}(2) \cdot f_2(2-2), R_{33}(3) \cdot f_2(2-3), \\ &R_{34}(4) \cdot f_2(2-4), R_{35}(5) \cdot f_2(2-5), R_{36}(6) \cdot f_2(2-6), R_{37}(7) \cdot \\ &f_2(2-7), R_{38}(8) \cdot f_2(2-8) \} \\ &= \min \{ 10353, 16443, -, -, -, -, - \} \\ &= 10353 \end{aligned}$$

Diperoleh $m_3^* = 1$

untuk $x_3=3$	$f_3(3) = 6174$	diperoleh $m_3^* = 1$
untuk $x_3=4$	$f_3(4) = 4998$	diperoleh $m_3^* = 1$
untuk $x_3=5$	$f_3(5) = 4998$	diperoleh $m_3^* = 1$
untuk $x_3=6$	$f_3(6) = 4760$	diperoleh $m_3^* = 3$
untuk $x_3=7$	$f_3(7) = 3528$	diperoleh $m_3^* = 1$
untuk $x_3=8$	$f_3(8) = 3528$	diperoleh $m_3^* = 1$

Nilai optimal untuk *stage* 3 sebesar 3528. Hasil akhir perhitungan dapat dilihat lebih lengkap pada tabel *stage* 3 pada lampiran 3.

Stage 4 (i=4)

Dengan rumus:

$$f_4(x_4) = \min \{R_{41}(b_{41}) \cdot f_3(x_4-b_{41}), R_{42}(b_{42}) \cdot f_3(x_4-b_{42}), R_{43}(b_{43}) \cdot f_3(x_4-b_{43}), R_{44}(b_{44}) \cdot f_3(x_4-b_{44}), R_{45}(b_{45}) \cdot f_3(x_4-b_{45}), R_{46}(b_{46}) \cdot f_3(x_4-b_{46}), R_{47}(b_{47}) \cdot f_3(x_4-b_{47}), R_{48}(b_{48}) \cdot f_3(x_4-b_{48})\}$$

untuk $x_4=1$

$$\begin{aligned} f_4(1) &= \min \{R_{41}(1) \cdot f_3(1-1), R_{42}(2) \cdot f_3(1-2), R_{43}(3) \cdot f_3(1-3), \\ &R_{44}(4) \cdot f_3(1-4), R_{45}(5) \cdot f_3(1-5), R_{46}(6) \cdot f_3(1-6), R_{47}(7) \cdot f_3(1-7), \\ &R_{48}(8) \cdot f_3(1-8)\} \\ &= \min \{230202, -, -, -, -, -, -, -\} \\ &= 230202 \end{aligned}$$

Diperoleh $m_4^* = 1$

untuk $x_4=2$

$$\begin{aligned} f_4(2) &= \min \{R_{41}(1) \cdot f_3(2-1), R_{42}(2) \cdot f_3(2-2), R_{43}(3) \cdot f_3(2-3), \\ &R_{44}(4) \cdot f_3(2-4), R_{45}(5) \cdot f_3(2-5), R_{46}(6) \cdot f_3(2-6), R_{47}(7) \cdot f_3(2-7), \\ &R_{48}(8) \cdot f_3(2-8)\} \\ &= \min \{186354, 166257, -, -, -, -, -, -\} \\ &= 166257 \end{aligned}$$

Diperoleh $m_4^* = 2$

untuk $x_4=3$	$f_4(3) = 111132$	diperoleh $m_4^* = 1$
untuk $x_4=4$	$f_4(4) = 80262$	diperoleh $m_4^* = 2$
untuk $x_4=5$	$f_4(5) = 64974$	diperoleh $m_4^* = 2$
untuk $x_4=6$	$f_4(6) = 64974$	diperoleh $m_4^* = 2$
untuk $x_4=7$	$f_4(7) = 61880$	diperoleh $m_4^* = 2$
untuk $x_4=8$	$f_4(8) = 45864$	diperoleh $m_4^* = 2$

Nilai optimal untuk *stage* empat sebesar 45864. Hasil akhir perhitungan dapat dilihat lebih lengkap pada tabel *stage* 4 pada lampiran 4.

Stage 5 (i=5)

Dengan rumus:

$$f_5(x_5) = \min \{ R_{51}(b_{51}) \cdot f_4(x_5 - b_{51}), R_{52}(b_{52}) \cdot f_4(x_5 - b_{52}), R_{53}(b_{53}) \cdot f_4(x_5 - b_{53}), R_{54}(b_{54}) \cdot f_4(x_5 - b_{54}), R_{55}(b_{55}) \cdot f_4(x_5 - b_{55}), R_{56}(b_{56}) \cdot f_4(x_5 - b_{56}), R_{57}(b_{57}) \cdot f_4(x_5 - b_{57}), R_{58}(b_{58}) \cdot f_4(x_5 - b_{58}) \}$$

untuk $x_5=1$

$$\begin{aligned} f_5(1) &= \min \{ R_{51}(1) \cdot f_4(1-1), R_{52}(2) \cdot f_4(1-2), R_{53}(3) \cdot f_4(1-3), \\ &R_{54}(4) \cdot f_4(1-4), R_{55}(5) \cdot f_4(1-5), R_{56}(6) \cdot f_4(1-6), R_{57}(7) \\ &\cdot f_4(1-7), R_{58}(8) \cdot f_4(1-8) \} \\ &= \min \{ 6215454, -, -, -, -, -, -, - \} \\ &= 6215454 \end{aligned}$$

Diperoleh $m_5^* = 1$

untuk $x_5=2$

$$\begin{aligned} f_5(2) &= \min \{ R_{51}(1) \cdot f_4(2-1), R_{52}(2) \cdot f_4(2-2), R_{53}(3) \cdot f_4(2-3), \\ &R_{54}(4) \cdot f_4(2-4), R_{55}(5) \cdot f_4(2-5), R_{56}(6) \cdot f_4(2-6), R_{57}(7) \\ &\cdot f_4(2-7), R_{58}(8) \cdot f_4(2-8) \} \\ &= \min \{ 4488939, 575505, -, -, -, -, - \} \\ &= 4488939 \end{aligned}$$

Diperoleh $m_5^* = 1$

untuk $x_5 = 3$	$f_5(3) = 3000564$	diperoleh $m_5^* = 1$
untuk $x_5 = 4$	$f_5(4) = 2167074$	diperoleh $m_5^* = 1$
untuk $x_5 = 5$	$f_5(5) = 1754298$	diperoleh $m_5^* = 1$
untuk $x_5 = 6$	$f_5(6) = 1624350$	diperoleh $m_5^* = 2$
untuk $x_5 = 7$	$f_5(7) = 1625350$	diperoleh $m_5^* = 2$
untuk $x_5 = 8$	$f_5(8) = 1238328$	diperoleh $m_5^* = 1$

Nilai optimal untuk *stage* lima sebesar 1238328. Hasil akhir perhitungan dapat dilihat lebih lengkap pada tabel *stage* 5 pada lampiran 5.

Stage 6 (i=6)

Dengan rumus:

$$f_6(x_6) = \min \{ R_{61}(b_{61}) \cdot f_5(x_6 - b_{61}), R_{62}(b_{62}) \cdot f_5(x_6 - b_{62}), R_{63}(b_{63}) \cdot f_5(x_6 - b_{63}), R_{64}(b_{64}) \cdot f_5(x_6 - b_{64}), R_{65}(b_{65}) \cdot f_5(x_6 - b_{65}), R_{66}(b_{66}) \cdot f_5(x_6 - b_{66}), R_{67}(b_{67}) \cdot f_5(x_6 - b_{67}), R_{68}(b_{68}) \cdot f_5(x_6 - b_{68}) \}$$

untuk $x_6=1$

$$\begin{aligned} f_6(1) &= \min \{ R_{61}(1) \cdot f_5(1-1), R_{62}(2) \cdot f_5(1-2), R_{63}(3) \cdot f_5(1-3), \\ &R_{64}(4) \cdot f_5(1-4), R_{65}(5) \cdot f_5(1-5), R_{66}(6) \cdot f_5(1-6), R_{67}(7) \cdot f_5(1-7), R_{68}(8) \cdot f_5(1-8) \} \\ &= \min \{ 6215454, -, -, -, -, -, -, - \} \\ &= 6215454 \end{aligned}$$

Diperoleh $m_6^* = 1$

untuk $x_6=2$

$$\begin{aligned} f_6(2) &= \min \{ R_{61}(1) \cdot f_5(2-1), R_{62}(2) \cdot f_5(2-2), R_{63}(3) \cdot f_5(2-3), \\ &R_{64}(4) \cdot f_5(2-4), R_{65}(5) \cdot f_5(2-5), R_{66}(6) \cdot f_5(2-6), R_{67}(7) \cdot f_5(2-7), R_{68}(8) \cdot f_5(2-8) \} \\ &= \min \{ 80800902, 118093626, -, -, -, -, -, - \} \\ &= 80800902 \end{aligned}$$

Diperoleh $m_6^* = 1$

untuk $x_6=3$	$f_6(3)=54010152$	diperoleh $m_6^*=1$
untuk $x_6=4$	$f_6(4)=39007332$	diperoleh $m_6^*=1$
untuk $x_6=5$	$f_6(5)=31577364$	diperoleh $m_6^*=1$
untuk $x_6=6$	$f_6(6)=29238300$	diperoleh $m_6^*=1$
untuk $x_6=7$	$f_6(7)=24560172$	diperoleh $m_6^*=3$
untuk $x_6=8$	$f_6(8)=22289904$	diperoleh $m_6^*=1$

Nilai optimal untuk *stage* 6 sebesar 22289904. Hasil akhir perhitungan dapat dilihat lebih lengkap pada tabel *stage* 6 pada lampiran 6.

Stage 7 (i=7)

Dengan rumus:

$$f_7(x_7) = \min \{R_{71}(b_{71}) \cdot f_6(x_7-b_{71}), R_{72}(b_{72}) \cdot f_6(x_7-b_{72}), R_{73}(b_{73}) \cdot f_6(x_7-b_{73}), R_{74}(b_{74}) \cdot f_6(x_7-b_{74}), R_{75}(b_{75}) \cdot f_6(x_7-b_{75}), R_{76}(b_{76}) \cdot f_6(x_7-b_{76}), R_{77}(b_{77}) \cdot f_6(x_7-b_{77}), R_{78}(b_{78}) \cdot f_6(x_7-b_{78})\}$$

untuk $x_7=1$

$$\begin{aligned} f_7(1) &= \min \{R_{71}(1) \cdot f_6(1-1), R_{72}(2) \cdot f_6(1-2), R_{73}(3) \cdot f_6(1-3), \\ &R_{74}(4) \cdot f_6(1-4), R_{75}(5) \cdot f_6(1-5), R_{76}(6) \cdot f_6(1-6), R_{77}(7) \cdot f_6(1-7), \\ &R_{78}(8) \cdot f_6(1-8)\} \\ &= \min \{1901928924, -, -, -, -, -, -, -\} \\ &= 1901928924 \end{aligned}$$

Diperoleh $m_7^* = 1$

untuk $x_7=2$

$$\begin{aligned} f_7(2) &= \min \{R_{71}(1) \cdot f_6(2-1), R_{72}(2) \cdot f_6(2-2), R_{73}(3) \cdot f_6(2-3), \\ &R_{74}(4) \cdot f_6(2-4), R_{75}(5) \cdot f_6(2-5), R_{76}(6) \cdot f_6(2-6), R_{77}(7) \cdot f_6(2-7), \\ &R_{78}(8) \cdot f_6(2-8)\} \\ &= \min \{1373615334, 2125685268, -, -, -, -, -, -\} \\ &= 1373615334 \end{aligned}$$

Diperoleh $m_7^* = 1$

untuk $x_7 = 3$	$f_7(3) = 918172584$	diperoleh $m_7^* = 1$
untuk $x_7 = 4$	$f_7(4) = 663124644$	diperoleh $m_7^* = 1$
untuk $x_7 = 5$	$f_7(5) = 536815188$	diperoleh $m_7^* = 1$
untuk $x_7 = 6$	$f_7(6) = 487051100$	diperoleh $m_7^* = 1$
untuk $x_7 = 7$	$f_7(7) = 417522924$	diperoleh $m_7^* = 1$
untuk $x_7 = 8$	$f_7(8) = 378928368$	diperoleh $m_7^* = 1$

Nilai optimal untuk *stage* 7 sebesar 378928368. Hasil akhir perhitungan dapat dilihat lebih lengkap pada tabel *stage* 7 pada lampiran 7.

Setelah semua *stage* selesai dikerjakan, maka dapat ditentukan keputusan optimal tiap *stage* dengan perhitungan mundur mulai dari *stage* 7 dicari nilai $f_7(x_7)$ sehingga diperoleh nilai m_7^* dan b_{7m7} , kemudian melakukan perhitungan menggunakan persamaan (2.11) dicari nilai x_6 . Begitu seterusnya menggunakan persamaan (2.12) sampai diperoleh nilai m_1^* dari *stage* 1.

Stage 7: $f_7(x_7)$ minimum pada $x_7 = 8$ sehingga diperoleh $m_7^* = 1$ $b_{71} = 1$ Jadi $x_6 = x_7 - b_{71}$

$$x_6 = 8 - 1 = 7$$

Stage 6: $m_6^* = 1$ $b_{61} = 1$ Jadi $x_5 = x_6 - b_{61}$
 $x_5 = 7 - 1 = 6$

Stage 5: $m_5^* = 1$ $b_{51} = 1$ Jadi $x_4 = x_5 - b_{51}$
 $x_4 = 6 - 1 = 5$

Stage 4: $m_4^* = 2$ $b_{42} = 2$ Jadi $x_3 = x_4 - b_{42}$
 $x_3 = 5 - 2 = 3$

Stage 3: $m_3^* = 1$ $b_{31} = 1$ Jadi $x_2 = x_3 - b_{31}$
 $x_2 = 3 - 1 = 2$

Stage 2: $m_2^* = 1$ $b_{21} = 1$ Jadi $x_1 = 2 - 1 = 1$

Stage 1: $m_1^* = 1$ $b_{11} = 1$

Maka keputusan optimal $m^* = (m_1^*, m_2^*, m_3^*, m_4^*, m_5^*, m_6^*, m_7^*)$
 $= (1, 1, 1, 2, 1, 1, 1)$

Penyelesaian keputusan optimal permasalahan pembagian armada ambulan dengan menggunakan rumus rekursif pada program dinamik menghasilkan kebijakan optimal. Keputusan optimal dari masing-masing daerah adalah sebagai berikut, m_1^* merupakan keputusan untuk daerah Kota Kediri dimana setidaknya terdapat 1 ambulan yang dialokasikan, m_2^* merupakan keputusan untuk daerah Kabupaten Kediri dimana setidaknya terdapat 1 ambulan dialokasikan, m_3^* merupakan keputusan untuk daerah Kota Blitar dimana setidaknya terdapat 1 ambulan yang dialokasikan, m_4^* merupakan keputusan untuk daerah Kabupaten Blitar dimana setidaknya terdapat 2 ambulan yang dialokasikan, m_5^* merupakan keputusan untuk daerah Kabupaten Tulungagung dimana setidaknya terdapat 1 ambulan yang dialokasikan, m_6^* merupakan keputusan untuk daerah Kabupaten Trenggalek dimana setidaknya terdapat 1 ambulan yang dialokasikan, m_7^* merupakan keputusan untuk daerah Kabupaten Nganjuk dimana setidaknya terdapat 1 ambulan yang dialokasikan.



UNIVERSITAS BRAWIJAYA



BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari perhitungan yang telah dilakukan dapat ditarik kesimpulan bahwa setidaknya terdapat 1 armada ambulance yang dialokasikan di Kota Kediri, terdapat 1 armada ambulance yang dialokasikan di Kabupaten Kediri, terdapat 1 armada ambulance yang dialokasikan di Kota Blitar, terdapat 2 armada ambulance yang dialokasikan di Kabupaten Blitar, terdapat 1 armada ambulance yang dialokasikan di Kabupaten Tulungagung, terdapat 1 armada ambulance yang dialokasikan di Kabupaten Trenggalek, dan terdapat 1 armada ambulance yang dialokasikan di Kabupaten Nganjuk. Terdapat 2 armada ambulance yang dialokasikan di Kabupaten Blitar dikarenakan untuk daerah ini terletak cukup jauh dari RSUD sehingga untuk dapat melayani permintaan tang ada maka dibutuhkan setidaknya 2 armada ambulance yang dialokasikan di daerah tersebut.

5.2 Saran

Untuk penelitian selanjutnya diharapkan tidak mengabaikan faktor- faktor yang dapat mempengaruhi hasil dari pembagian armada ambulance salah satunya adalah faktor jarak.

UNIVERSITAS BRAWIJAYA



DAFTAR PUSTAKA

- Bronson, R. 1983. *Schaum's Outline Series : Theory and Problem of Operations Research*. McGraw-Hill International Book. Singapore.
- Hiilier, F.S. dan G.J. Lieberman. 2005. *Introduction To Operation Research. Seventh Ed.* Mc Graw-Hill International Editions. Singapore.
- Lee, S.M., Moore, L.J., and Taylor, B.W. 1990. *Management Science, Third Ed.* Allyn and Bacon Inc. Boston.
- Mulyono, S. 2002. *Riset Operasi*. Lembaga Penerbit Fakultas Ekonomi Universitas Indonesia. Jakarta.
- Schmid, Verena. 2011. *Solving the dynamic Ambulance Relocation and Dispatching Problem Using Approximate Dynamic Programming*. Department of Bussines Administration, University of Vienna, Austria.
<http://www.sciencedirect.com/science/article/pii/S0377221711009830>. Diakses pada tanggal 13 Februari 2013.
- Siagian, P.1987. *Penelitian Operasional : Teori dan Praktek*. Penerbit Universitas Indonesia (UI-Press). Jakarta.
- Subagyo, P., Marwan Asri dan T. Hani Handoko. 1985. *Dasar-dasar Operations Research*. BPFE. Yogyakarta.
- Taha, H.A. 1996. *Operation Research an Introduction*. Macmillan Publishing Co, INC. New York.
- Taha, H.A. 1996. *Riset Operasi – Suatu Pengantar, Edisi Kelima, jilid 2, Terjemahan : Drs. Daniel Wirajaya*. Binarupa Aksara. Jakarta.

Widiastuti, Utami. 2006. *Aplikasi Program Dinamis Pada Pengendalian Persediaan Bahan Baku Di PT Miwon Indonesia*. Skripsi FMIPA Universitas Brawijaya.

UNIVERSITAS BRAWIJAYA



UNIVERSITAS BRAWIJAYA



UNIVERSITAS BRAWIJAYA



UNIVERSITAS BRAWIJAYA



UNIVERSITAS BRAWIJAYA



Lampiran 8. (Listing Program dan Tampilan)

<i>Listing Program</i>
<pre>unit Unit1; interface uses Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls, Grids, Math, ComCtrls, Menus; type TForm1 = class(TForm) MainMenu: TMainMenu; Menu1: TMenuItem; Save1: TMenuItem; Open1: TMenuItem; PageControl1: TPageControl; TabSheet1: TTabSheet; Label7: TLabel; Label8: TLabel; StringGrid2: TStringGrid; Button7: TButton; TabSheet2: TTabSheet; StringGrid3: TStringGrid; TabSheet3: TTabSheet; StringGrid4: TStringGrid; TabSheet4: TTabSheet; StringGrid5: TStringGrid; TabSheet5: TTabSheet; StringGrid6: TStringGrid; TabSheet6: TTabSheet; StringGrid7: TStringGrid; TabSheet7: TTabSheet; StringGrid8: TStringGrid; TabSheet8: TTabSheet; StringGrid9: TStringGrid;</pre>

```

Label1: TLabel;
StaticText1: TStaticText;
StaticText2: TStaticText;
StaticText3: TStaticText;
procedure FormCreate(Sender: TObject);
procedure Save1Click(Sender: TObject);
procedure Open1Click(Sender: TObject);
procedure Button7Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form1: TForm1;
  A, dataA : array [0..100,0..100] of
integer;
  hit1, hit2, hit3, hit4, hit5, min : array
[0..100] of integer;
  max1, max2, max3, max4, max5, total :
integer;

implementation

{$R *.dfm}

procedure TForm1.FormCreate(Sender:
TObject);
var
  i, j : integer;
begin
  PageControll1.ActivePage:=TabSheet1;
  for i:=1 to 7 do
    for j:=1 to 8 do
      begin
        StringGrid2.Cells[i,0]:=IntToStr(i);
        StringGrid2.Cells[0,j]:=IntToStr(j);
      end;
    end;
  end;

```

```

for i:=1 to 8 do
  for j:=1 to 8 do
    begin
      StringGrid3.Cells[i,0]:=IntToStr(i);
      StringGrid3.Cells[0,i]:=IntToStr(i);
      StringGrid3.Cells[i,j]:='-';
      StringGrid3.Cells[9,0]:='F1 (x1)';

      StringGrid4.Cells[i,0]:=IntToStr(i);
      StringGrid4.Cells[0,i]:=IntToStr(i);
      StringGrid4.Cells[i,j]:='-';
      StringGrid4.Cells[9,0]:='F2 (x2)';

      StringGrid5.Cells[i,0]:=IntToStr(i);
      StringGrid5.Cells[0,i]:=IntToStr(i);
      StringGrid5.Cells[i,j]:='-';
      StringGrid5.Cells[9,0]:='F3 (x3)';

      StringGrid6.Cells[i,0]:=IntToStr(i);
      StringGrid6.Cells[0,i]:=IntToStr(i);
      StringGrid6.Cells[i,j]:='-';
      StringGrid6.Cells[9,0]:='F4 (x4)';

      StringGrid7.Cells[i,0]:=IntToStr(i);
      StringGrid7.Cells[0,i]:=IntToStr(i);
      StringGrid7.Cells[i,j]:='-';
      StringGrid7.Cells[9,0]:='F5 (x5)';

      StringGrid8.Cells[i,0]:=IntToStr(i);
      StringGrid8.Cells[0,i]:=IntToStr(i);
      StringGrid8.Cells[i,j]:='-';
      StringGrid8.Cells[9,0]:='F6 (x6)';

      StringGrid9.Cells[i,0]:=IntToStr(i);
      StringGrid9.Cells[0,i]:=IntToStr(i);
      StringGrid9.Cells[i,j]:='-';
      StringGrid9.Cells[9,0]:='F7 (x7)';
    end;
end;

```

```
end;
```

```
procedure SaveStringGrid(StringGrid:
TStringGrid; const FileName: TFileName);
var
  f:      TextFile;
  i, k: Integer;
begin
  AssignFile(f, FileName);
  Rewrite(f);
  with StringGrid do
  begin
    Writeln(f, ColCount);
    Writeln(f, RowCount);
    for i := 0 to ColCount - 1 do
      for k := 0 to RowCount - 1 do
        Writeln(f, Cells[i, k]);
      end;
    CloseFile(f);
  end;
end;
```

```
procedure LoadStringGrid(StringGrid:
TStringGrid; const FileName: TFileName);
var
  f:      TextFile;
  iTmp, i, k: Integer;
  strTemp: String;
begin
  AssignFile(f, FileName);
  Reset(f);
  with StringGrid do
  begin
    Readln(f, iTmp);
    ColCount := iTmp;
    Readln(f, iTmp);
```



```

RowCount := iTmp;
for i := 0 to ColCount - 1 do
  for k := 0 to RowCount - 1 do
    begin
      Readln(f, strTemp);
      Cells[i, k] := strTemp;
    end;
  end;
CloseFile(f);
end;

procedure TForm1.Save1Click(Sender:
TObject);
begin
  SaveStringGrid(StringGrid2,
'+GetCurrentDir+'\data awal.txt');
  showmessage('Data sudah disimpan');
end;

procedure TForm1.Open1Click(Sender:
TObject);
begin
  LoadStringGrid(StringGrid2,
'+GetCurrentDir+'\data awal.txt');
end;

procedure TForm1.Button7Click(Sender:
TObject);
var
  i,j,k : integer;
  f1,f2,f3,f4,f5,f6 : array [0..100] of
integer;
begin
  for i:=1 to 8 do
    for j:=1 to 7 do

dataA[i,j]:=StrToInt(StringGrid2.Cells[j,i])
;

```

```

//perhitungan stage 1
for i:=1 to 8 do
  for j:=i to 8 do
    begin

StringGrid3.Cells[i,j]:=IntToStr(dataA[i,1]*
1);
    end;

  for i:=1 to 8 do
    for j:=i to 8 do
      begin

min[i]:=StrToInt(StringGrid3.Cells[1,i]);

        for k:=1 to i do
          if min[i]>
StrToInt(StringGrid3.Cells[k,i]) then

min[i]:=StrToInt(StringGrid3.Cells[k,i]);

StringGrid3.Cells[9,i]:=IntToStr(min[i]);
        end;

//perhitungan stage 2
      for i:=1 to 8 do
        f1[i]:=StrToInt(StringGrid3.Cells[9,i]);

      for i:=1 to 8 do
        for j:=i to 8 do
          begin
            if i=1 then

StringGrid4.Cells[i,j]:=IntToStr(dataA[i,2]*
f1[j])
            else

StringGrid4.Cells[i,j]:=IntToStr(dataA[i,2]*

```

```

f1[j-(i-1)];
    end;

    for i:=1 to 8 do
        for j:=i to 8 do
            begin

min[i]:=StrToInt(StringGrid4.Cells[1,i]);

                for k:=1 to i do
                    if min[i]>
StrToInt(StringGrid4.Cells[k,i]) then
min[i]:=StrToInt(StringGrid4.Cells[k,i]);

StringGrid4.Cells[9,i]:=IntToStr(min[i]);
                end;

//perhitungan stage 3
                for i:=1 to 8 do
                    f2[i]:=StrToInt(StringGrid4.Cells[9,i]);

                    for i:=1 to 8 do
                        for j:=i to 8 do
                            begin
                                if i=1 then

StringGrid5.Cells[i,j]:=IntToStr(dataA[i,3]*
f2[j])
                                else

StringGrid5.Cells[i,j]:=IntToStr(dataA[i,3]*
f2[j-(i-1)]);
                                end;

                            for i:=1 to 8 do
                                for j:=i to 8 do
                                    begin

```

```

min[i]:=StrToInt(StringGrid5.Cells[1,i]);

    for k:=1 to i do
        if min[i]>
StrToInt(StringGrid5.Cells[k,i]) then

min[i]:=StrToInt(StringGrid5.Cells[k,i]);

StringGrid5.Cells[9,i]:=IntToStr(min[i]);
    end;

//perhitungan stage 4
for i:=1 to 8 do
    f3[i]:=StrToInt(StringGrid5.Cells[9,i]);

for i:=1 to 8 do
    for j:=i to 8 do
        begin
            if i=1 then

StringGrid6.Cells[i,j]:=IntToStr(dataA[i,4]*
f3[j])
                else

StringGrid6.Cells[i,j]:=IntToStr(dataA[i,4]*
f3[j-(i-1)]);
            end;

for i:=1 to 8 do
    for j:=i to 8 do
        begin

min[i]:=StrToInt(StringGrid6.Cells[1,i]);

    for k:=1 to i do
        if min[i]>
StrToInt(StringGrid6.Cells[k,i]) then

```

```

min[i]:=StrToInt(StringGrid6.Cells[k,i]);

StringGrid6.Cells[9,i]:=IntToStr(min[i]);
end;

//perhitungan stage 5
for i:=1 to 8 do
  f4[i]:=StrToInt(StringGrid6.Cells[9,i]);

  for i:=1 to 8 do
    for j:=i to 8 do
      begin
        if i=1 then
StringGrid7.Cells[i,j]:=IntToStr(dataA[i,5]*
f4[j])
          else
StringGrid7.Cells[i,j]:=IntToStr(dataA[i,5]*
f4[j-(i-1)]);
          end;

        for i:=1 to 8 do
          for j:=i to 8 do
            begin

min[i]:=StrToInt(StringGrid7.Cells[1,i]);

          for k:=1 to i do
            if min[i]>
StrToInt(StringGrid7.Cells[k,i]) then

min[i]:=StrToInt(StringGrid7.Cells[k,i]);

StringGrid7.Cells[9,i]:=IntToStr(min[i]);
          end;

```

```

//perhitungan stage 6
for i:=1 to 8 do
    f5[i]:=StrToInt(StringGrid7.Cells[9,i]);

for i:=1 to 8 do
    for j:=i to 8 do
        begin
            if i=1 then
StringGrid8.Cells[i,j]:=IntToStr(dataA[i,6]*
f5[j])
                else
StringGrid8.Cells[i,j]:=IntToStr(dataA[i,6]*
f5[j-(i-1)]);
            end;

for i:=1 to 8 do
    for j:=i to 8 do
        begin
min[i]:=StrToInt(StringGrid8.Cells[1,i]);

            for k:=1 to i do
                if min[i]>
StrToInt(StringGrid8.Cells[k,i]) then
min[i]:=StrToInt(StringGrid8.Cells[k,i]);

StringGrid8.Cells[9,i]:=IntToStr(min[i]);
            end;

//perhitungan stage 7
for i:=1 to 8 do
    f6[i]:=StrToInt(StringGrid8.Cells[9,i]);

for i:=1 to 8 do

```

```

    for j:=i to 8 do
    begin
        if i=1 then

StringGrid9.Cells[i,j]:=IntToStr(dataA[i,7]*
f6[j])
            else

StringGrid9.Cells[i,j]:=IntToStr(dataA[i,7]*
f6[j-(i-1)]);
            end;

        for i:=1 to 8 do
            for j:=i to 8 do
                begin

min[i]:=StrToInt(StringGrid9.Cells[1,i]);

                    for k:=1 to i do
                        if min[i]>
StrToInt(StringGrid9.Cells[k,i]) then

min[i]:=StrToInt(StringGrid9.Cells[k,i]);

StringGrid9.Cells[9,i]:=IntToStr(min[i]);
                        end;

                    end;

                end.

```

Tampilan Program Penghitung Pembagian armada Ambulan Menggunakan Metode Program Dinamik

Form1
Menu File

Program Penghitung Pembagian Armada Ambulan Menggunakan Metode Program Dinamik

Nalsa Cintya Resti
0910940011

Data Awal | Stage 1 | Stage 2 | Stage 3 | Stage 4 | Stage 5 | Stage 6 | Stage 7

Daerah

	1	2	3	4	5	6	7
1	29	21	21	18	27	18	17
2	28	17	27	13	25	19	19
3	14	18	20	21	30	14	15
4	19	24	18	19	21	21	15
5	27	12	17	16	19	20	19
6	30	17	19	14	25	12	14
7	31	19	25	15	28	10	16
8	18	14	29	17	23	17	18

Ambulan

Proses

Form1
Menu File

Program Penghitung Pembagian Armada Ambulan Menggunakan Metode Program Dinamik

Nalsa Cintya Resti
0910940011

Data Awal | Stage 1 | Stage 2 | Stage 3 | Stage 4 | Stage 5 | Stage 6 | Stage 7

	1	2	3	4	5	6	7	8	F1(x1)
1	29	-	-	-	-	-	-	-	29
2	29	28	-	-	-	-	-	-	28
3	29	28	14	-	-	-	-	-	14
4	29	28	14	19	-	-	-	-	14
5	29	28	14	19	27	-	-	-	14
6	29	28	14	19	27	30	-	-	14
7	29	28	14	19	27	30	31	-	14
8	29	28	14	19	27	30	31	18	14

Form1
Menu File

**Program Penghitung Pembagian Armada Ambulan
Menggunakan Metode Program Dinamik**

Nalsa Cintya Resti
0910940011

Data Awal | Stage 1 | Stage 2 | Stage 3 | Stage 4 | Stage 5 | Stage 6 | Stage 7

	1	2	3	4	5	6	7	8	F2(x2)
1	609	-	-	-	-	-	-	-	609
2	588	493	-	-	-	-	-	-	493
3	294	476	522	-	-	-	-	-	294
4	294	238	504	696	-	-	-	-	238
5	294	238	252	672	348	-	-	-	238
6	294	238	252	336	336	493	-	-	238
7	294	238	252	336	168	476	551	-	168
8	294	238	252	336	168	238	532	406	168

Form1
Menu File

**Program Penghitung Pembagian Armada Ambulan
Menggunakan Metode Program Dinamik**

Nalsa Cintya Resti
0910940011

Data Awal | Stage 1 | Stage 2 | Stage 3 | Stage 4 | Stage 5 | Stage 6 | Stage 7

	1	2	3	4	5	6	7	8	F3(x3)
1	12789	-	-	-	-	-	-	-	12789
2	10353	16443	-	-	-	-	-	-	10353
3	6174	13311	12180	-	-	-	-	-	6174
4	4998	7938	9860	10962	-	-	-	-	4998
5	4998	6426	5880	8874	10353	-	-	-	4998
6	4998	6426	4760	5292	8381	11571	-	-	4760
7	3528	6426	4760	4284	4998	9367	15225	-	3528
8	3528	4536	4760	4284	4046	5586	12325	17661	3528

Form1
Menu File

**Program Penghitung Pembagian Armada Ambulan
Menggunakan Metode Program Dinamik**

Nalsa Cintya Resti
0910940011

Date Awal | Stage 1 | Stage 2 | Stage 3 | Stage 4 | Stage 5 | Stage 6 | Stage 7

	1	2	3	4	5	6	7	8	F4(x-4)
1	230202	-	-	-	-	-	-	-	230202
2	186354	166257	-	-	-	-	-	-	166257
3	111132	134589	266569	-	-	-	-	-	111132
4	89964	80262	217413	242991	-	-	-	-	80262
5	89964	64974	129654	196707	204624	-	-	-	64974
6	85680	64974	104958	117306	165648	179046	-	-	64974
7	63504	61880	104958	94962	98784	144942	191835	-	61880
8	63504	45864	99960	94962	79968	86436	155295	217413	45864

Form1
Menu File

**Program Penghitung Pembagian Armada Ambulan
Menggunakan Metode Program Dinamik**

Nalsa Cintya Resti
0910940011

Date Awal | Stage 1 | Stage 2 | Stage 3 | Stage 4 | Stage 5 | Stage 6 | Stage 7

	1	2	3	4	5	6	7	8	F5(x-5)
1	6215454	-	-	-	-	-	-	-	6215454
2	4488939	5755050	-	-	-	-	-	-	4488939
3	3000564	4156425	6906060	-	-	-	-	-	3000564
4	2167074	2778300	4987710	4834242	-	-	-	-	2167074
5	1754298	2006550	3333960	3491397	4373838	-	-	-	1754298
6	1754298	1624350	2407860	2333772	3158883	5755050	-	-	1624350
7	1670760	1624350	1949220	1685502	2111508	4156425	6445656	-	1624350
8	1238328	1547000	1949220	1364454	1524978	2778300	4655196	5294646	1238328

Form1
Menu File

Program Penghitung Pembagian Armada Ambulan
Menggunakan Metode Program Dinamik

Nalsa Cintya Resti
0910940011

Data Awal | Stage 1 | Stage 2 | Stage 3 | Stage 4 | Stage 5 | Stage 6 | Stage 7

	1	2	3	4	5	6	7	8	F6(x6)
1	111878172	-	-	-	-	-	-	-	111878172
2	80800902	118093626	-	-	-	-	-	-	80800902
3	54010152	85289841	87016356	-	-	-	-	-	54010152
4	39007332	57010716	62845146	130524534	-	-	-	-	39007332
5	31577364	41174406	42007896	94267719	124309080	-	-	-	31577364
6	29238300	33331662	30339036	63011844	89778780	74585448	-	-	29238300
7	29238300	30862650	24560172	45508554	60011280	53867268	62154540	-	24560172
8	22289904	30862650	22740900	36840258	43341480	36006768	44889390	105662718	22289904

Form1
Menu File

Program Penghitung Pembagian Armada Ambulan
Menggunakan Metode Program Dinamik

Nalsa Cintya Resti
0910940011

Data Awal | Stage 1 | Stage 2 | Stage 3 | Stage 4 | Stage 5 | Stage 6 | Stage 7

	1	2	3	4	5	6	7	8	F7(x7)
1	1901928924	-	-	-	-	-	-	-	1901928924
2	1373615334	2125685268	-	-	-	-	-	-	1373615334
3	918172584	1535217138	1678172580	-	-	-	-	-	918172584
4	663124644	1026192888	1212013530	1678172580	-	-	-	-	663124644
5	536815188	741139308	810152280	1212013530	2125685268	-	-	-	536815188
6	497051100	599689916	585109980	810152280	1535217138	1566294408	-	-	497051100
7	417522924	555527700	473660460	585109980	1026192888	1131212628	1790050752	-	417522924
8	378928368	466643268	438574500	473660460	741139308	756142128	1292814432	2013807096	378928368