

Simulasi Industri

by Nelly Budiharti

Submission date: 18-Nov-2022 09:24AM (UTC+0700)

Submission ID: 1957345701

File name: isiBuku_Simulasi_Sistem_Industri.pdf (1.29M)

Word count: 16746

Character count: 98454

Bagian I

SISTEM, MODEL DAN SIMULASI

Sistem simulasi industri merupakan materi mata kuliah yang meliputi penggabungan dari unsur sistem dan simulasi yang satu dengan lainnya saling berinteraksi untuk menyelesaikan permasalahan-permasalahan yang ada di dalam dunia industri, baik yang dinyatakan dalam bentuk model-model industri manufaktur maupun jasa.

Suatu sistem diidentifikasi sebagai kumpulan dari atribut-atribut, misalnya: orang atau mesin yang saling berinteraksi melakukan suatu kegiatan bersama-sama untuk mencapai suatu tujuan. Dalam prakteknya, apa yang dimaksud dengan sistem tergantung pada tujuan yang akan dicapai. Kumpulan atribut yang tersusun dalam sistem untuk suatu tujuan mungkin hanya menjadi bagian dari sistem yang lain secara keseluruhan. Sebagai contoh : jika seseorang ingin mempelajari sistem, baik untuk menentukan jumlah teller (fasilitas pelayanan) yang cukup untuk melayani nasabah dalam bertransaksi untuk membayar, mengecek atau menabung, maka sistem dapat

didefinisikan menjadi bagian fasilitas dari unit pelayanan bank dan nasabah menunggu dalam antrian untuk dilayani. Jadi hanya bagian peminjaman dan penabungan yang dimaksud dalam sistem. Definisi sistem disini harus dijelaskan secukupnya. Kita definisikan status sistem yang akan menjadi kumpulan beberapa variabel yang penting untuk menggambarkan suatu sistem pada waktu tertentu yang relatif terhadap tujuan yang satu dengan yang lainnya.

Dalam mempelajari sebuah bank, misalnya sebagai berikut: peluang suatu variabel adalah jumlah teller yang sibuk, jumlah pelayanan dalam bank dan waktu kedatangan masing-masing pelanggan/ nasabah dalam bank. Kita mengelompokkan sistem menjadi 2 tipe, yaitu diskrit dan kontinyu:

1. Suatu sistem diskrit adalah hanya satu (terbatas) status perubahan variabel yang dapat ditentukan dari sejumlah waktu yang tersedia. Sebuah bank adalah satu contoh dari status variabel dari sistem diskrit, yaitu jumlah nasabah dalam bank, perubahannya ketika kedatangan pelanggan atau ketika nasabah selesai dilayani dan pergi.
2. Suatu sistem kontinyu adalah suatu perubahan status variabel secara terus menerus sepanjang waktu. Contohnya: pergerakan pesawat diudara, dimana status variabel dari perubahan posisi atau kecepatan terhadap waktu.

Beberapa sistem dalam kenyataannya ada yang diskrit atau kontinyu tetapi hanya perubahan yang paling menonjol yang terjadi pada sistem tersebut sehingga cenderung diskrit atau kontinyu.

Kadang-kadang perlu mempelajari sebuah sistem untuk mengerti hubungan antara macam-macam komponen atau untuk memperkirakan hasilnya pada sebuah kebijakan operasi baru, karena percobaan secara langsung terhadap sistem tidak memungkinkan karena tidak efektif atau akan mengganggu operasi sistem yang ada. Dalam kenyataannya sering terjadi bahwa sistem yang akan dibahas belum ada atau tersedia. Sebagai contoh: pada saat ingin mempelajari (mungkin besarnya tabungan) mempengaruhi jumlah pengurangan teller pada suatu bank. Jika jumlah teller di bank dikurangi secara berarti, akan menunda nasabah dan menyita mereka untuk melakukan pekerjaan bisnis berikutnya. Karena ketidakmungkinan mempraktekkan sistem secara langsung, maka untuk menganalisa sistem tersebut sering menggunakan sebuah model dari sistem untuk menggambarkan kesimpulan tentang operasi dari sistem yang nyata tersebut.

Model adalah suatu penggambaran kembali dari suatu sistem untuk tujuan mempelajari sistem tersebut. Model seharusnya cukup detail atau valid untuk dapat menganalisa atau membuat keputusan untuk menggunakannya untuk membuat keputusan yang sama tentang sistem yang akan dibuat, jika memungkinkan untuk melakukan dengan sistem tersebut.

Pada buku ini kita membatasi perhatian kita untuk sebuah tipe khusus dari model matematika dari sistem yang kita sebut model simulasi (ada beberapa model fisika selain matematika). Misalnya: model skala dari suatu test pesawat dalam menerobos awan. Namun demikian secara umum, kita tidak akan secara jelas mendefinisikan suatu model simulasi. Kita membeda-

kan antara model simulasi yang statis atau dinamis, deterministik atau stolistik dan diskrit atau kontinyu. Sebuah model simulasi statis adalah sebuah penggambaran kembali sebuah sistem pada waktu tertentu, misalnya model simulasi montekarlo.

Sebuah model simulasi dinamik adalah penggambaran kembali dari sebuah sistem terhadap waktu yang mengikutinya. Misalnya: model simulasi dari sebuah kegiatan pada sebuah bank pada 8 jam kerja/ hari. Sebuah model simulasi dikatakan deterministik jika tidak berisi random variabel. Untuk model deterministik ada satu set keluaran data yang unik untuk pemberian satu set input. Dengan kata lain model simulasi stolistik adalah jika berisi satu atau beberapa random variabel. Data output untuk model stolistik adalah random mereka sendiri dan hanya estimasi dari model karakteristik yang benar. Sebuah model simulasi bank akan secara normal memperlakukan waktu antara kedatangan dan waktu pelayanan dari pelanggan sebagai random variabel dengan masing-masing distribusi peluangnya. Kita definisikan model simulasi diskrit dan kontinyu analog dengan sistem diskrit dan kontinyu yang kita definisikan diatas.. Keputusan apakah untuk menggunakan model diskrit atau kontinyu untuk sistem tertentu tergantung pada tujuan pembelajaran tertentu. Sebagai contoh: sebuah model dari lajunya aliran *traffic* pada jalan bebas hambatan bisa digambarkan jika sifat-sifat dan gerakan masing-masing mobil penting. Alternatifnya, jika mobil-mobil dapat dilakukan dalam gabungan/ kelompok, laju *traffic* dapat digambarkan melalui persamaan model kontinyu.

Bagian II

SIMULASI KEJADIAN DISKRIT

Simulasi kejadian diskrit khususnya dari permodelan dari sistem sebagai penyelesaian menyeluruh dari sebuah representasi yang mana status variabel berubah hanya pada sejumlah perhitungan perwaktu. Waktu disini adalah terjadinya suatu kejadian dimana 'kejadian' ini didefinisikan sebagai suatu kejadian yang saat itu juga merubah status dari sebuah sistem. Walaupun simulasi kejadian diskrit secara konsep dapat dilakukan dengan cara manual, jumlah data yang harus dicatat dan dianalisa untuk sebagian besar sistem nyata memerlukan computer digital.

Contoh 1.

Sebuah fasilitas pelayanan dengan satu service. Misalnya: satu operator/ satu tukang cukur/ satu meja informasi di *airport*. Misalnya: kita akan memperkirakan rata-rata penundaan dalam antrian dari kedatangan konsumen, dimana penundaaan dalam antrian pelanggan adalah lamanya waktu antara kedatangannya, pelanggan secara langsung/ cepat pada fasilitas itu. Secara langsung dia mulai dilayani untuk perkiraan yang objektif rata-

rata penundaan pelanggan. Status variabel untuk kejadian (model kejadian) diskrit dari fasilitas menjadi status pelayanan. Misalnya nganggur atau sibuk. Sejumlah konsumen menunggu (dalam antrian) untuk dilayani (beberapa) dan waktu kedatangan dari masing-masing orang yang sedang menunggu dalam antrian.

Status pelayanan diperlukan untuk menentukan pada seorang pelanggan yang datang. Apakah langsung dapat dilayani atau harus bergabung dalam antrian yang terakhir. Ketika server menyelesaikan service ke pelanggan, jumlah pelanggan dalam antrian digunakan untuk menentukan apakah server akan menjadi nganggur atau melayani pelanggan pertama dalam antrian. Waktu kedatangan pelanggan diperlukan untuk menghitung penundaannya dalam antrian yaitu waktu dia mulai dilayani (yang akan diketahui) dikurangi waktu kedatangannya. Ada 2 tipe kejadian untuk sistem ini, yaitu kedatangan pelanggan sampai berakhirnya. Dan untuk pelanggan yang dalam keberangkatan pelanggan. Sebuah kedatangan adalah sebuah kejadian karena (penentuan variabel) pembahasan status dari nganggur ke sibuk atau (penentuan variabel) pelanggan dalam antrian akan bertambah satu dengan demikian selah keberangkatan adalah sebuah kejadian sebab status akan berubah dari sibuk menjadi menganggur atau jumlah pelanggan dalam antrian akan berkurang satu. Kita dapat melihat secara detail bagaimana membuat model simulasi kejadian diskrit dengan satu pelayanan pada sistem Antrian.

Pada kedua contoh diatas, tipe kejadian terkecuali perubahan status sistem, tetapi pada beberapa model simulasi

kejadian diskrit, kejadian-kejadian digunakan untuk tujuan yang tidak membuat suatu perubahan seperti itu, sebagai contoh : sebuah kejadian mungkin digunakan lebih/ untuk berakhirnya simulasi pada waktu tertentu atau waktu untuk memutuskan tentang system operasi pada waktu tertentu yang mungkin tidak memberikan suatu perubahan dalam status system, ini kita katakan bahwa mungkin sebuah kejadian sistem.yang berubah status.

Bagian III

MEKANISME – WAKTU – KEJADIAN

Penyebab dari dinamika alam dari model-model simulasi kejadian diskrit, perlu kita perhatikan jejak dari nilai akhir dari waktu simulasi sebagai hasil simulasi. Kita juga perlu sebuah mekanisme untuk kelanjutan waktu simulasi dari satu nilai ke nilai yang lain yang kita sebut variabel dalam sebuah model simulasi yang memberikan nilai akhir dari waktu simulasi terhadap simulasi *clock*. Unit waktu untuk simulasi *clock* dinyatakan tidak pernah jelas/ nyata ketika sebuah model ditulis pada sebuah bahasa seperti FORTRAN, BASIC, dan bahasa pemrograman yang lainnya dan ini diasumsikan menjadi waktu yang sama sebagai parameter input karena tidak ada hubungan antara waktu simulasi dan keperluan waktu untuk menjalankan simulasi pada computer.

Secara histories ada 2 prinsip pendekatan untuk kelanjutan memperbincangkan simulasi *CLOCK* yaitu: '*Time Advance*' dan '*Fixed Increment*'. *Time Advance*, pendekatan pertama digunakan oleh sebagian besar bahasa simulasi dan oleh sebagian besar orang yang meng-kode model mereka dalam bahasa yang

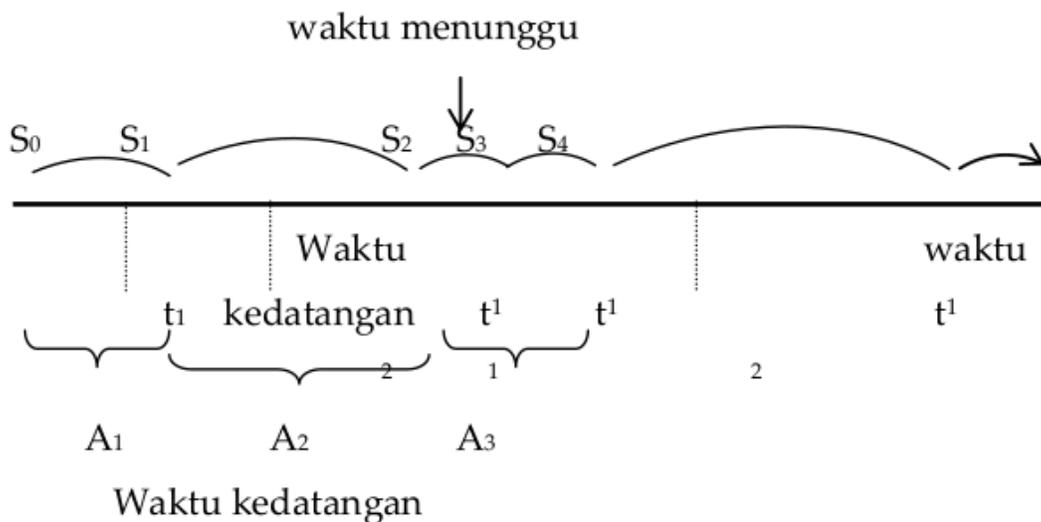
umum dan *Fixed Increment* pendekatan yang kedua adalah kasus khusus dan yang pertama kita akan mendiskusikan *next event time advance* untuk semua pendekatan model simulasi kejadian diskrit dalam buku ini. Dengan pendekatan *the next event time*, simulasi *clock* diawal ke nol dan waktu kejadian untuk kejadian mendatang ditentukan. Simulasi *clock* kemudian dilanjutkan ke waktu kejadian dari kejadian mendatang yang paling dekat, yang mana status sistem diubah untuk menghitung kenyataan bahwa sebuah kejadian telah terjadi dan pengetahuan kita dari waktu yang terjadi dari kejadian mendatang juga dirubah. Kemudian simulasi *clock* dilanjutkan ke waktu (bisa) kejadian yang paling dekat. Status sistem dirubah dan waktu kejadian mendatang ditetapkan dan seterusnya. Proses ini dilanjutkan sampai berakhirnya.

Contoh 2

Sekarang kita gambarkan detail kejadian berikut dengan *Time Advance* untuk system antrian *Single Service* dari contoh 1, kita butuh notasi berikut:

- t_t = waktu kedatangan pelanggan
- A_t = $t_t - t_{t-1}$ = waktu antar kedatangan
- S_t = waktu yang dihabiskan untuk melayani ke i (kadang-kadang pelanggan tertunda dalam antrian)
- D_t = waktu tunggu dalam antrian pelanggan ke i
- C_t = $t_t + D_t + S_t$ = waktu pelanggan ke i selesai dilayani dan meninggalkan system
- S_i = waktu kejadian i dalam beberapa tipe (nilai simulasi *clock* i ditetapkan dengan nilai $S_i = 0$)

Masing-masing definisi kuantitatif akan menjadi variabel random. Asumsi bahwa probabilitas distribusi dari waktu antar kedatangan A_1, A_2, \dots dan waktu service S_1, S_2, \dots lihat dan dicatat seketika oleh F_A, F_S, \dots (umumnya F_A dan F_S akan ditentukan oleh sekumpulan dari data sistem yang diteliti dan kemudian menggunakan distribusi yang sesuai). Jadi waktu $S_0 = 0$ status server adalah nganggur dan waktu kedatangan pertama t_1 ditentukan dengan membangkitkan A_t dari F_A (teknik untuk membangkitkan variabel bilangan random dengan distribusi tertentu didiskusikan tersendiri) dan menambahkannya/memasukkannya ke nol. Kemudian simulasi *clock* dilanjutkan dari S_0 ke waktu dari kejadian berikutnya (pertama). $S_1 = t_1$ (lihat gambar 1. dimana arah kurva penampilan kelanjutan simulasi *clock*)



Gambar 1.
Gambaran pendekatan waktu kejadian berikutnya sistem
antrian server tunggal

Karena waktu kedatangan konsumen pada waktu t mendapatkan server nganggur dan segera masuk atau dilayani dan mempunyai penundaan dalam antrian $D_t = 0$ kemudian status server diubah dari nganggur ke sibuk. Waktu C_1 waktu kedatangan pelanggan kemudian menyelesaikan servis dihitung dengan membangkitkan S_1 dan F_s dan ditambahkan ke t_1 . Akhirnya waktu kedatangan ke 2 t_2 dihitung sebagai $t_2 = t_1 + A_2$ dimana A_2 dibangkitkan dari F_A jika $t_2 < C_1$ seperti digambarkan pada gambar 1. Simulasi *clock* dilanjutkan dari S_1 ke C_1 ketika kedatangan pelanggan pada waktu t_2 menemukan server telah sibuk jumlah pelanggan dalam antrian ditambah dari 0 ke 1 dan waktu kedatangan dari pelanggan telah dicatat, namun waktu pelayanannya S_2 tidak dibangkitkan pada waktu ini juga waktu kedatangan ke 3, t_3 terhitung sebagai $t_3 = t_2 + A_3$. jika $C_1 < t_3$ seperti dijabarkan pada gambar, simulasi *clock* dilanjutkan dari S_2 ke waktu kejadian berikut sebagai $S_3 = C_1$ dimana pelanggan meninggalkan pelayanan yang telah selesai, pelanggan dalam antrian (yakni: salah satu yang datang, pada waktu t_2 memulai pelayanan dan menundanya dalam antrian dan waktu penyelesaian pelayanan dihitung sebagai $D_2 = C_1 - t_2$ dan $C_2 = C_1 + S_1$ sedang dibangkitkan dari F_s dan jumlah pelanggan-pelanggan dalam antrian ditambah adalah berkurang dari 1 ke 0. Jika $t_2 < C_2$ simulasi *clock* dilanjutkan ke S_3 ke waktu kejadian berikut $S_4 = t_3$ dan seterusnya. Simulasi diakhiri ketika katakanlah jumlah pelanggan-pelanggan tertunda diteliti mencapai nilai tertentu.

Bagain IV

KOMPONEN DAN ORGANISASI DARI MODEL SIMULASI KEJADIAN DISKRIT

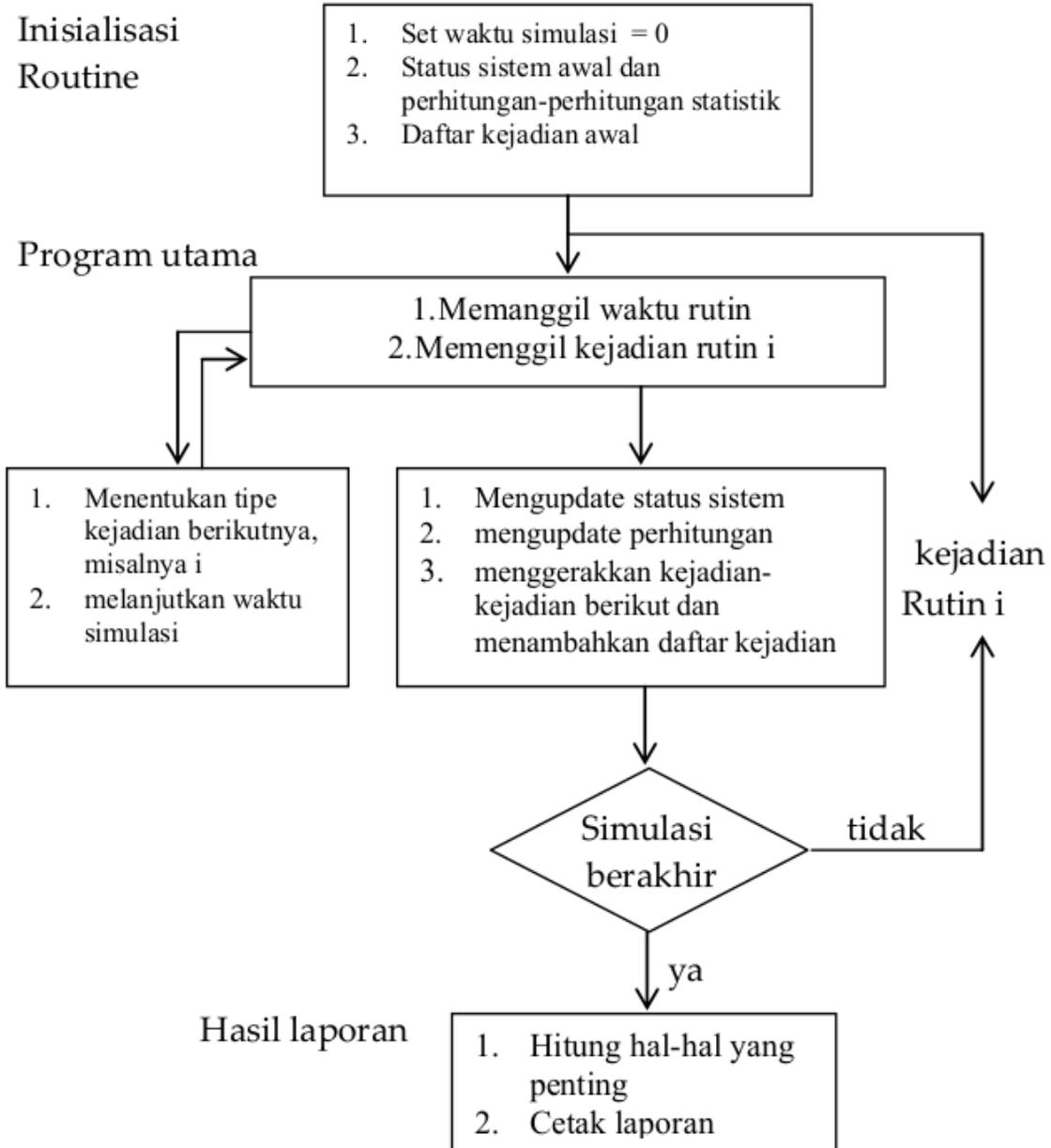
Walaupun simulasi telah digunakan untuk membedakan sistem dunia nyata, model-model simulasi kejadian diskrit mempunyai sejumlah komponen-komponen umum dan ada sebuah organisasi logis untuk komponen-komponen ini dengan mengembangkan peng-kode-an dan perubahan yang akan datang juga program computer model simulasinya, khususnya komponen-komponen yang akan ditemukan pada sebagian besar model simulasi kejadian diskrit yang menggunakan pendekatan *next event time advance*.

Status sistem	Pentingnya mengumpulkan status variabel untuk menggambarkan sistem pada waktu tertentu
Simulasi <i>clock</i>	Sebuah variabel yang memberikan nilai akhir dari unit simulasi
Event list	Daftar yang berisi waktu, selanjutnya ko-

	tak masing-masing tipe dari kejadian akan terjadi
Perhitungan-perhitungan statistik	Penggunaan variabel untuk menyimpan informasi secara statistik tentang kondisi/peformansi sistem
Rutin inisialisasi	Penggunaan sebuah sub-routine untuk inisial model simulasi pada waktu nol
Rutin timing	Sebuah sub-routine yang menentukan kejadian berikutnya dari daftar kejadian dan selanjutnya simulasi <i>clock</i> ke waktu dimana kejadian akan terjadi.
Event routine	Sebuah sub-routine yang mengup-date status sistem ketika sebuah tipe kejadian khusus terjadi (ada sub-routine kejadian untuk masing-masing tipe kejadian)
Report generator	Sebuah sub-routine yang memperkirakan perhitungan-perhitungan dari ukuran-ukuran yang diinginkan dari hasil dan print/ cetakan sebuah laporan ketika simulasi berakhir.
Main program	Sebuah sub-program yang memanggil waktu rutin untuk menentukan kejadian berikut dan untuk merubah/ memindah control untuk menghubungkan <i>event routine</i> ke sistem yang di update pada status yang sesuai.

Hubungan logis (aliran control) antara komponen ini ditunjukkan pada gambar 2. Simulasi mulai pada waktu nol dengan program utama memanggil *routine* inisialisasi, dimana simulasi *clock* di set nol, status sistem dan perhitungan-perhitungan statistik dimulai dan daftar kejadian dimulai setelah mengecek lagi ke main program, disebut *routine timing* untuk tipe kejadian yang telah terjadi semua. Apabila sebuah tipe kejadian *i* adalah kejadian berikutnya simulasi *clock* dilanjutkan ke waktu, yang mana tipe kejadian tipe *i* akan terjadi dan control dikembalikan ke program utama memanggil *event routine i*, dimana 3 tipe khusus dari aktifitas terjadi:

1. Mengupdate status sistem untuk mengaitkan kenyataan bahwa sebuah kejadian tipe telah terjadi.
2. Mengumpulkan informasi tentang peformansi sistem dengan meng-*update* perhitungan-perhitungan statistik
3. Menggerakkan waktu dari kejadian-kejadian berikutnya dan menambahkan daftar informasi kejadian. Setelah semua selesai diproses, juga dalam *event routine i* atau dalam *main* program.



Gambar 2.

Diagram alir untuk pendekatan *next – event – time - advance*

Suatu pengontrol dibuat untuk menentukan (relatif untuk beberapa kondisi penyetop) apakah simulasi sekarang harus diakhiri jika simulasi sekarang akan berakhir, pembangkit laporan dipanggil dari program utama untuk memperkirakan perhitungan (dari perhitungan-perhitungan statik) dari pengu-

kuran-pengukuran yang diinginkan dari performasi dan selanjutnya laporan dicetak. Jika belum berakhir, control dilewatkan kembali ke program utama dan program utama – routine timing – program utama – *event routine* – berakhir siklus cek dilanjutkan kembali sampai kondisi betul-betul memuaskan untuk distop.

Sebelum menyimpulkan bagian ini, beberapa kata tambahan tentang status sistem, mungkin dalam hal ini sebagaimana keterangan dalam bagian 2. Sebuah sistem adalah kumpulan entitis. *Entiti* adalah disifatkan dengan nilai-nilai data yang disebut atribut, dan atribut adalah bagian dari status sistem. Untuk model simulasi kejadian diskrit. Lebih lanjut lagi, entiti dengan beberapa sifat umum yang dikelompokkan bersama dalam daftar-daftar (file atau set) untuk masing-masing entiti, ada sebuah catatan dalam daftar yang berisi entitinya atribut dan dalam catatannya masing-masing ditempatkan dalam daftar tergantung pada beberapa rumus tertentu.

Untuk fasilitas antrian pelayanan tunggal dari contoh 1 dan 2 entiti adalah pelayanan-pelayanan dan pelanggan-pelanggan. Pelayan memperoleh atribut 'status' dan pelanggan-pelanggan menunggu dalam antrian memperoleh atribut 'waktu kedatangan' (jumlah pelanggan-pelanggan dalam antrian mungkin juga dipertimbangkan sebuah atribut pelayanan).

Bagian V

KEUNTUNGAN DAN KERUGIAN DARI SIMULASI

Beberapa kemungkinan alasan penggunaan simulasi. Keuntungannya antara lain :

1. Begitu kompleks, sistem dunia nyata dengan elemen-elemen stokastik tidak dapat digambarkan secara akurat dengan sebuah model matematika yang dapat dievaluasi secara analitik. Jadi sebuah simulasi sering menjadi kemungkinan penyelidikan.
2. Simulasi memberikan suatu performance untuk memperkirakan keberadaan sistem pada beberapa proyek dari kondisi-kondisi operasi.
3. Rancangan usulan alternatif sistem (atau alternatif kebijakan-kebijakan operasi untuk sistem tunggal) dapat dibandingkan melalui simulasi untuk melihat penemuan-penemuan yang terbaik untuk suatu keinginan tertentu.
4. Pada sebuah simulasi kita dapat memelihara/ menjaga banyak control lebih baik melebihi kondisi percobaan

secara umum daripada kemungkinan ketika percobaan dengan sistem itu sendiri.

5. Simulasi mengizinkan kita untuk mempelajari sebuah sistem sepanjang waktu, misalnya: sebuah sistem ekonomi, dalam waktu singkat atau alternatif yang lain dengan mempelajari secara detail dari sebuah sistem kerja dalam waktu yang lama.

Simulasi tidak mungkin tanpa kekurangan, ada beberapa kekurangan atau kerugian antara lain:

1. Model -model simulasi sering mahal dan menghabiskan waktu untuk mengembangkannya.
2. Untuk setiap *run* (hasil), sebuah model simulasi stokastik hanya menghasilkan perkiraan dari model-model sebenarnya untuk sebuah input parameter tertentu. Jadi menghasilkan beberapa model bebas yang mungkin diperlukan untuk masing-masing sekumpulan input parameter untuk dipelajari untuk alasan ini, model-model simulasi umumnya tidak sebaik pada teori optimasi. Dengan kata lain sebuah model analitik, jika cocok sering dapat dengan mudah menghasilkan sifat-sifat nyata yang sebenarnya dari model itu untuk bermacam-macam input parameter. Jadi jika sebuah model analitik 'tepat' tersedia atau dapat dengan mudah dikembangkan, ini akan tersedia secara umum untuk sebuah model simulasi.
3. Sejumlah besar jumlah produksi melalui mempelajari sebuah simulasi sering cenderung menghasilkan kepercayaan yang lebih besar dari sistem yang sedang dipelajari, hasil simulasi, tidak seberapa dibandingkan kesan mereka

yang mungkin tampak/ terlihat akan memberikan kegunaan kecil informasi dari sistem yang nyata. Ketika memutuskan apakah mempelajari simulasi atau tidak adalah sesuaikan pada situasinya. Kita hanya dapat menasehati bahwa keuntungan dan kerugiannya menjadi perhatian/ pemikiran dan semua yang berhubungan dengan situasi yang dibawa untuk menghasilkan yang terbaik. Akhirnya, yang perlu dicatat bahwa dalam beberapa studi untuk model-model simulasi dan analitik mungkin kedua-duanya digunakan, khususnya simulasi dapat digunakan untuk mengecek ketepatan dari asumsi-asumsi yang diperlukan dalam model analitik. Dengan kata lain, sebuah model analitik dapat memberikan alternatif-alternatif alasan untuk penyelidikan dalam mempelajari simulasi.

Bagian VI

SIMULASI DARI SEBUAH SISTEM ANTRIAN PELAYANAN TUNGGAL

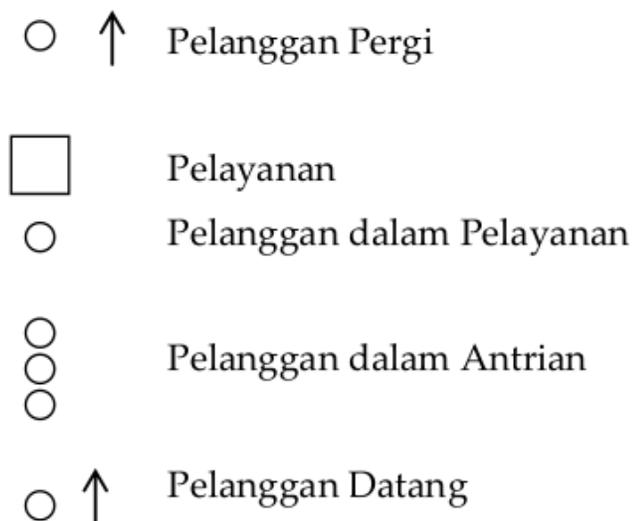
Bagian ini menunjukkan secara detail bagaimana sistem simulasi antrian pelayanan tunggal seperti satu operator tempat pemangkas rambut. Walaupun sistem ini sepertinya sangat sederhana dibandingkan kejadian-kejadian penting yang lain, yang biasanya terjadi pada dunia nyata, namun demikian, mensimulasikan kejadian ini masih cukup mewakili proses simulasi-simulasi yang begitu kompleks. Diskusi dibagi menjadi 3 bagian penting yaitu:

Kondisi tujuan secara terpadu.

Secara intuisi kita akan mempelajari bagaimana mensimulasikan sistem, dengan menunjukkan suatu 'potret' dari sistem simulasi setelah beberapa kejadian baru saja terjadi. Kita mendiskusikan sebuah model simulasi dengan bahasa pemrograman FORTRAN.

Menyatakan Masalah

Anggap sistem antrian pelayanan tunggal (lihat gambar 3) yang mana waktu antar kedatangan A_1, A_2, \dots adalah mengikuti distribusi bebas (IID) variabel random ('mengikuti distribusi' artinya bahwa waktu kedatangan mempunyai distribusi peluang bebas). Seorang pelanggan yang datang dan menemukan pelayan nganggur segera masuk kedalam antrian dan waktu pelayanan S_1, S_2, \dots dari pelanggan berturut-turut adalah variabel bebas dari waktu antar kedatangan. Seorang pelanggan yang datang dan menemukan pelayan sibuk pada waktu akhir antrian tunggal. Untuk menyelesaikan pelayanan satu pelanggan, pelayan memilih seorang pelanggan dari antrian (jika ada) pada antrian yang ke 1 (fifo).



Gambar 3.
Suatu Sistem Antrian Pelayanan Tunggal

Simulasi sistem ini sampai n pelanggan yang telah tertunda dan untuk memperkirakan rata-rata pelanggan tertunda, $d(n)$ dan jumlah rata-rata pelanggan dalam antrian, $Q(n)$ mengukur hasil $d(n)$ diperkirakan dengan n :

$$d(n) = \frac{\sum_{i=1}^n D_i}{n}$$

Juga disebut rata-rata sampel dari D_i s dan dicatat dengan $D(n)$. pada buku ini symbol (n) diatas adalah sebuah ukuran estimator hasil $Q(n)$, dengan kata lain, diestimasi dengan:

$$d(n) = \frac{\int_0^T Q(t) dt}{T} = \frac{\sum_{i=1}^m R_i}{n}$$

Dimana:

T = Jumlah waktu yang diperlukan untuk n pelanggan untuk menyelesaikan ketertundaan mereka (suatu variabel random)

$Q(t)$ = Jumlah pelanggan dalam antrian pada waktu t ($0 \leq t \leq T$)

R_i = Daerah pada bujur sangkar $Q(t)$ antara S_{i-1} (S_i adalah waktu kejadian i dari beberapa tipe, dan $S_0 = 0$)

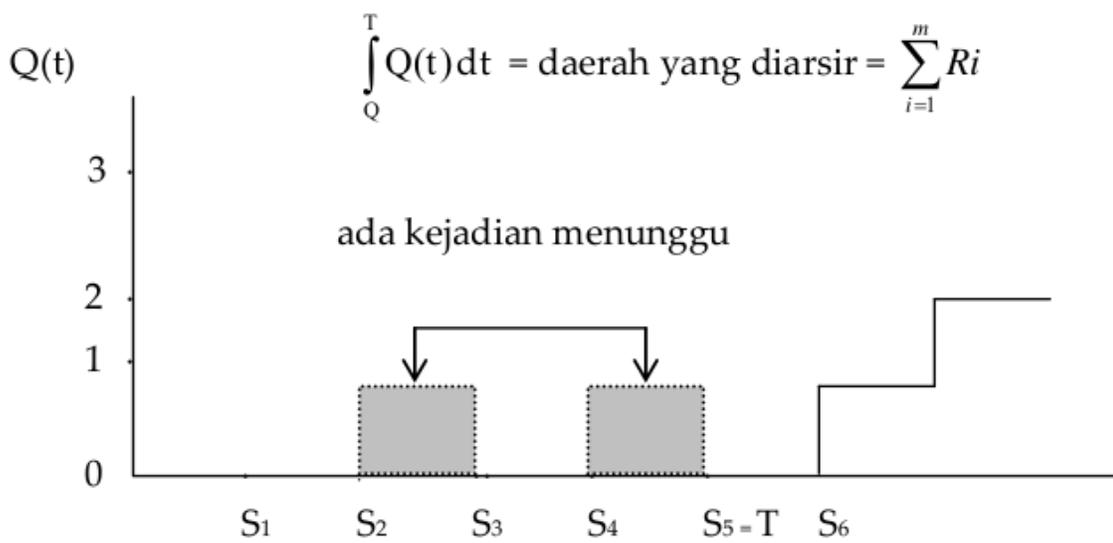
m = Jumlah kejadian yang terjadi pada interval $(0, T)$ (sebuah random variabel) mengulang dari kalkulus

bahwa $\int_0^T Q(t) dt$ adalah parameter pada $Q(t)$ antara 0

dan T kemudian.

$Q^i(n)$ adalah rata-rata nilai fungsi dari $Q(t)$ pada interval $(0, T)$ persamaan dari $\int_Q^T Q(t) dt$ dan $\sum_{i=1}^m Ri$ digambarkan untuk kasus $n = 3$ pada gambar 4.

Kejadian-kejadian untuk sistem ini adalah kedatangan dari seorang pelanggan setelah pelayanan selesai; status atau kondisi variabel penting untuk $d(n)$ dan $Q(n)$ yaitu kondisi pelayanan jumlah pelanggan dalam antrian, waktu kedatangan masing-masing pelanggan dalam antrian dan waktu kejadian antrian (hampir berakhir).



Gambar 4.

Persamaan dari $\int_0^T Q(t) dt$ dan $\sum_{i=0}^m Ri$

**Menggambarkan : Terjadi Kedatangan pada S_1, S_2, S_4, S_6
Keberangkatan terjadi S_3 , dan S_5**

Waktu kejadian terakhir didefinisikan menjadi S_{i-1} . Jika $S_{i-1} \leq t < S_i$ (t adalah waktu) diperlukan untuk menghitung area R_i pada waktu S_i dari kejadian i . Sebuah estimator seperti $d(n)$ yang mana rata-rata dari D_i nya adalah/ disebut sampel statistik. Ketika estimator seperti $Q(n)$ yang mana rata-rata nilai fungsi $Q(t)$ pada interval $(0, T)$ disebut waktu rata-rata statistik. Kedua tipe estimasi umumnya ada pada model-model simulasi kejadian diskrit.

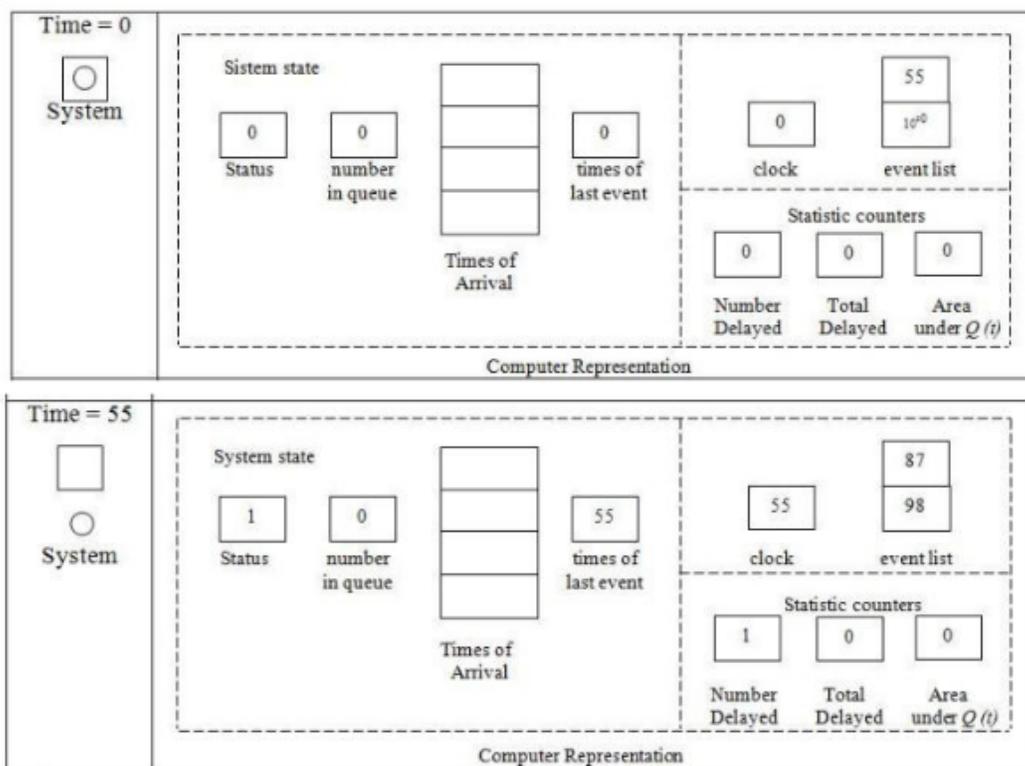
Penjelasan berdasarkan Intuisi

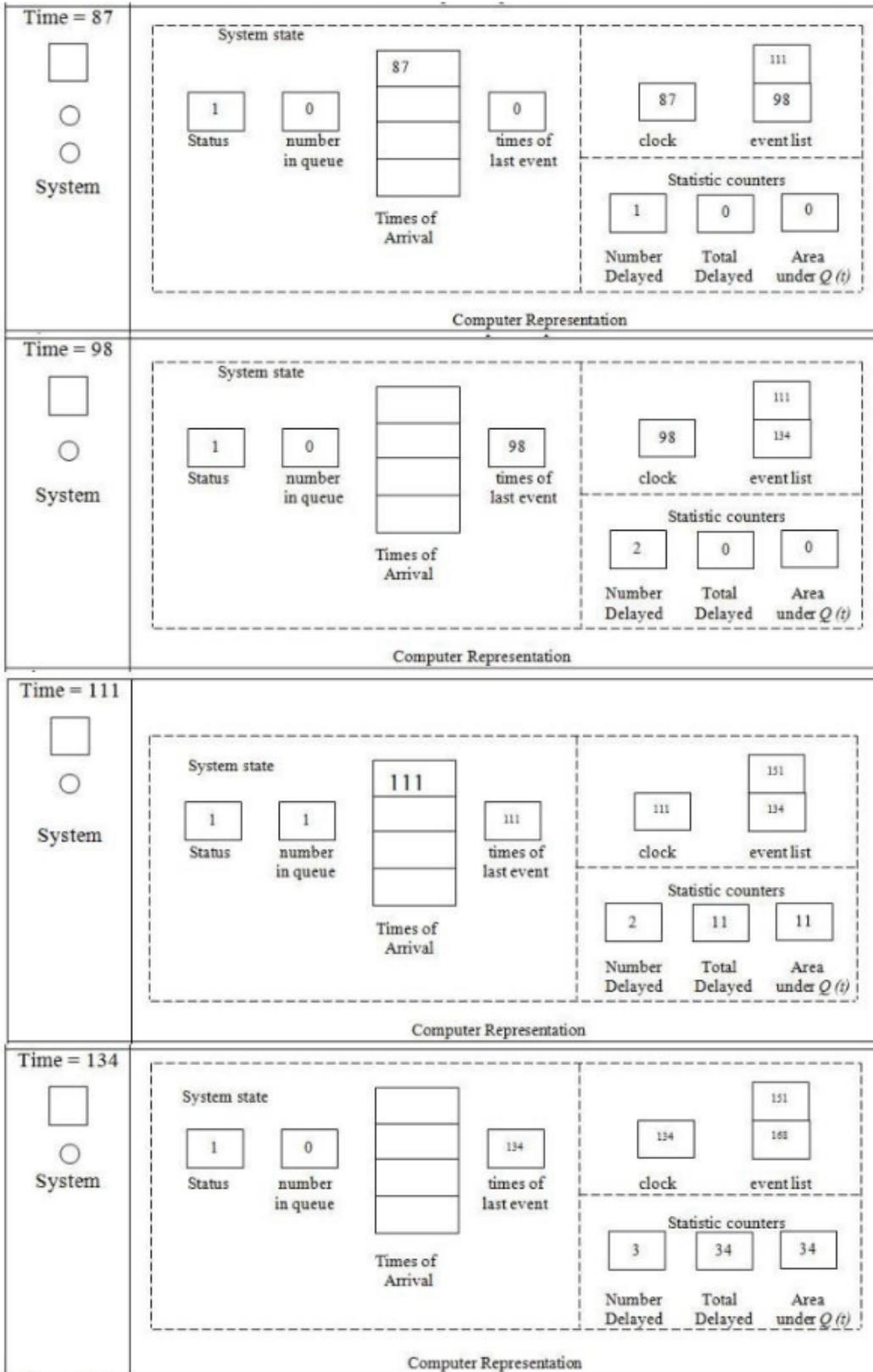
Kita mulai keterangan kita bagaimana mensimulasikan sistem antrian dengan pelayanan tunggal dengan menunjukkan bagaimana model simulasinya akan ditampilkan dalam komputer pada saat $S_0=0$ dan pada saat kejadian terjadi berturut-turut S_1, S_2, \dots untuk hubungan yang tepat, kita asumsikan bahwa waktu antar kedatangan dan pelayanan pelanggan diketahui dan mempunyai nilai-nilai :

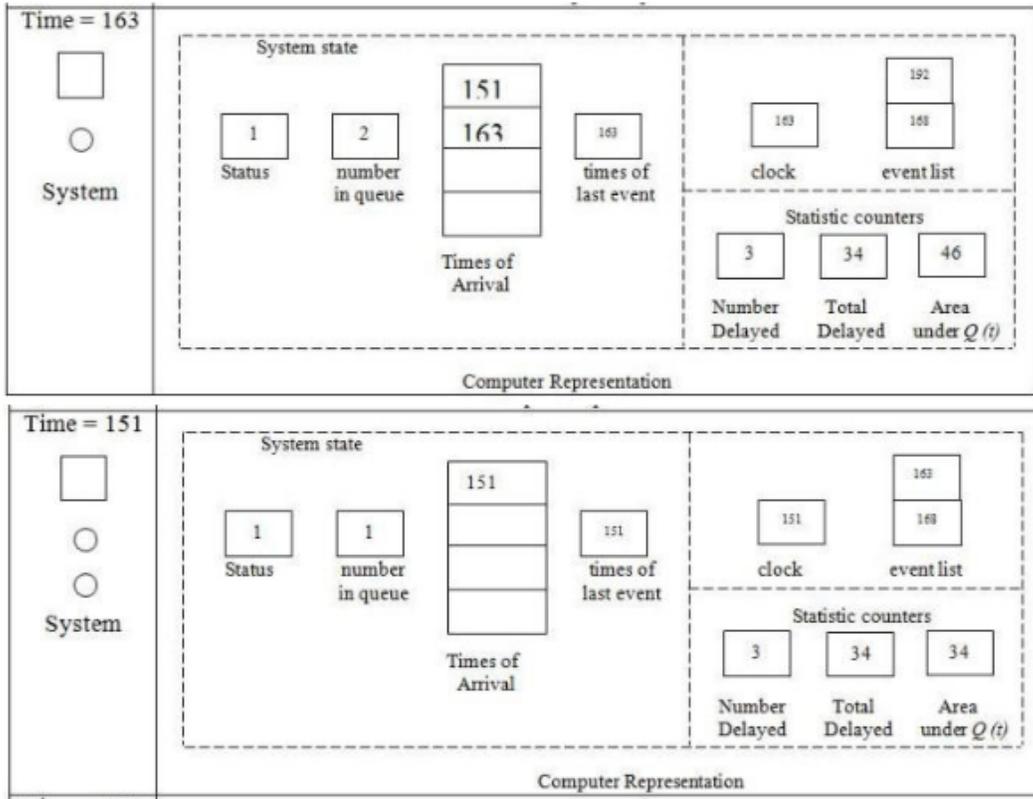
$$\begin{array}{cccccc} A_1 = 55 & A_2 = 32 & A_3 = 24 & A_4 = 40 & A_5 = 12 & \\ A_6 = 29 & S_1 = 43 & S_2 = 36 & S_3 = 34 & & \end{array}$$

Jadi antara 0 dan antara kedatangan pelanggan yang pertama adalah 55 unit waktu, antara kedatangan pelanggan yang pertama dengan yang kedua adalah 32 unit waktu, dan seterusnya, waktu pelayanan pelanggan pertama adalah 43 unit dan seterusnya. Dalam sebuah simulasi yang nyata (lihat no 3) A_1^s dan S_1^s akan dibangkitkan dari probabilitas distribusi mereka yang relevan sebagaimana perlunya selama adanya simulasi. Gambar 5 memberikan gambaran sistem dan penampilan computer dari sistem pada masing-masing waktu $S_0 = 0$,

$S_1 = 55$, $S_2 = 87$, $S_3 = 98$, $S_4 = 111$, $S_5 = 134$, $S_6 = 151$ dan $S_7 = 163$. Hal ini terfokus pada bagaimana perubahan tampilan computer pada masing-masing waktu kejadian. Pada waktu 0 program utama memanggil inialisasi routine ke model inisial. Penampilan komputer setelah semua inialisasi dilakukan dan ditunjukkan dalam gambar ke-1. Mencatat status, bahwa jumlah pelanggan dalam antrian, waktu akhir kejadian simulasi clock dan menetapkan semua perhitungan-perhitungan statistic ke nol. Gunakan 0 untuk menyatakan pelayanan nganggur dan 1 untuk sibuk. Ada sederetan untuk menyimpulkan waktu kedatangan pelanggan dalam antrian yang semula kosong. Lihat Gambar 5.







Gambar 5. Model Simulasi dan Analisa

Untuk daftar kejadian, penelitian waktu kedatangan berikutnya (ke 1) diberi sampel A ditetapkan ke 55 karena $A_1 = 55$, karena tidak ada pelanggan maka segera langsung dilayani. Waktu berikutnya pelanggan meninggalkan (diberi simbol 1) ditetapkan dengan nilai positif terbesar 1030 untuk mengatasi bahwa kejadian berikutnya yang terjadi adalah tidak selesainya pelayanan. (Nilai terbesar ini akan menjadi lebih besar dari pada beberapa nilai simulasi clock yang akan mengasumsikan simulasi). Setelah semua inisial dilakukan, control dikembalikan ke program utama, yang kemudian memanggil waktu routine untuk menentukan kejadian berikutnya karena $55 < 1030$, waktu *routine* menyatakan simulasi *clock* ke waktu kedatangan berikutnya, yaitu 55. Kontrol dikembalikan ke

program utama dengan sebuah penunjuk bahwa kejadian berikutnya adalah sebuah kedatangan.

Pada waktu 55, program utama memanggil routine kejadian kedatangan seorang pelanggan. Tampilan computer dari sistem setelah semua perubahan telah dibuat pada waktu 55 ditunjukkan dalam bagian gambar ke 2 karena pelanggan yang datang mendapatkan pelayanan nganggur (status 0) dimulai dengan segera melayani dan mempunyai waktu tunggu dalam antrian 0. Sebagai catatan bahwa status telah di set 1 sejak pelanggan sekarang sibuk. Jumlah pelanggan-pelanggan yang telah diselesaikan mereka. Waktu tunggu sekarang adalah 1 dan total waktu menunggu pada area Q (t) adalah 0. Sebuah waktu kedatangan berikutnya 87 dan 98 ditunjukkan dengan penambahan $A_1 = 32$ dan $S_1 = 43$, sehingga untuk waktu kedatangan 55 waktu kejadian terakhir adalah 55 dan control dikembalikan ke program utama. Program utama memanggil waktu *routine* *dank arena* $87 < 98$ simulasi *clock* dilanjutkan ke waktu kedatangan berikutnya, 87. Kontrol dikembalikan ke program utama dengan menunjukkan kejadian berikutnya adalah sebuah kedatangan.

Pada waktu 87, program utama sekali lagi memanggil kejadian *routine* kedatangan untuk memproses kedatangan pelanggan. Tampilan computer setelah semua penambahan yang dibuat pada waktu 87 ditunjukkan pada bagian gambar ke 3. Kedatangan pelanggan mendapatkan pelayanan sibuk dia bergabung (masih dalam antrian, jumlah dalam antrian ditetapkan menjadi 1 dan waktu kedatangannya dicatat dalam lokasi ke 1 dari deretan/ barisan). Nilai perhitungan statistik

tidak berubah karena pelanggan baru saja mulai ditunda *dank arena* $Q(t)$ adalah 0 sebuah kedatangan ini waktu kedatangan berikut 111, ditentukan dengan menambahkan $A_3 = 24$ ke waktu kedatangan 87 dan waktu keberangkatan berikutnya diubah (pelanggan pertama ada dalam layanan pada waktu 87) waktu kejadian terakhir ditetapkan menjadi 87 dan kontrol dikembalikan ke program utama. Program utama memanggil waktu *routine* dan karena $98 < 111$, simulasi *clock* dilanjutkan ke waktu keberangkatan berikutnya (penyelesaian pelanggan) 98. Kontrol dilanjutkan ke program utama dengan menunjukkan kejadian berikutnya adalah keberangkatan. Pada waktu 98, program utama memanggil kejadian *routine* keberangkatan untuk memproses keberangkatan seorang pelanggan (orang yang datang pada waktu 5, (lihat gambar bagian 4 untuk tampilan computer setelah semua pembahasan telah dibuat). Karena jumlah dalam antrian sebelum keberangkatan adalah 1, area pada $Q(t)$ harus dirubah/ diperbaharui dengan menambahkan sebelumnya jumlah antrian, 1 dikalikan dengan panjangnya interval waktu antara waktu kejadian terakhir 87 dan nilai simulasi *clock* 98 yaitu satu kali $(98-87)$ untuk area $Q(t)$ yang sebelumnya 0, menghasilkan area baru pada 11, juga pelanggan pertama dalam antrian pada waktu 87) melanjutkan pelayanan. Pengurangan jumlah dalam antrian dari 1 ke penambahan jumlah pelanggan yang menunggu dari 1 ke 2 dan penambahan total waktu menunggu dari 0 ke 11 [$11 = 0 = 0 + (98 - 87)$] (catatan bahwa ini penting untuk area pada $Q(t)$ sebelum mengurangi jumlah dalam antrian). Waktu kedatangan berikutnya, 134, dihitung dengan menambahkan $S_2 = 36$ untuk waktu pelanggan yang sedang memulai pelayanan 98,

dan waktu kedatangan berikutnya diubah. Waktu kejadian terakhir ditetapkan menjadi 98 dan kontrol dikembalikan ke program utama. Program utama memanggil waktu *routine dank arena* $111 < 134$, simulasi dilanjutkan ke program utama dengan sebuah penunjukkan/indikasi kejadian berikutnya adalah kedatangan dan seterusnya.

Pembaca harus berhati-hati memperbaiki perbaikan tampilan komputer dari sistem setelah semua penambahan telah dibuat. Pada waktu kejadian 111, 134, 151 dan 163, khususnya catatan waktu 163 adalah waktu pertama dimana ketika total waktu tunggu tidak sama dengan area $Q(t)$. Kenapa? simulasi akan dilanjutkan sampai jumlah pelanggan yang tertunda adalah sama dengan pada saat ini, program utama memanggil pembangkit laporan dan rata-rata waktu tunggu dan diperkirakan dengan membagi total waktu tunggu $\sum D_{i-1}$ dengan jumlah waktu tunggu n , begitu juga waktu rata-rata jumlah dalam antrian diperkirakan dengan membagi pada area $Q(t) \int^T Q(t)$ dengan nilai akhir dari simulasi clock OT .

Bagian VII

PROGRAM FORTRAN

Pengertian alami dari kejadian simulasi diskrit, tetapi ada perbedaan besar antar pengertian intuisi dan kemampuan untuk menulis program komputer yang efisien. Oleh karena itu kita sekarang akan menulis sebuah program *Fortran* untuk simulasi system antrian pelayanan tunggal.

Alasan untuk memilih bahasa *Fortran* yaitu:

1. Kita melihat bahwa, meskipun tersedia beberapa bahasa, khususnya simulasi, banyak model simulasi kejadian diskrit masih ditulis dalam bahasa *Fortran*.
2. Kita merasa bahwa dengan mempelajari simulasi dalam bahasa *Fortran* (kita dapat mempelajari/ memperkaitkan setiap kejadian secara rinci), kesalahan secara konsep, kurang/ sedikit terjadi.

Untuk menulis sebuah program FORTRAN, diperlukan beberapa asumsi tambahan tentang sistem antrian tunggal, khususnya asumsi bahwa waktu kedatangan pelanggan untuk sistem IID eksponensial variabel random dengan sebuah rata-rata 1 (satu) menit dan waktu pelayanan pelanggan adalah

variabel random exponential dengan sebuah rata-rata 0,5 menit. Distribusi *exponential* terpilih karena mudah untuk dibangkitkan dari komputer, yaitu variabel random kontinu dengan fungsi kepadatan probabilitas.

$$f(x) = \frac{1}{B} e^{-x/b} \text{ untuk } x \geq 0$$

Dimana parameter $p > 0$ adalah juga rata-rata (perlu lebih jauh mendiskusikan distribusi exponential). Kita sekarang mempunyai sebutan umum untuk sistem antrian yaitu antrian $M | M | I$, kondisi berhenti untuk simulasi adalah sampai n tepat 1000 pelanggan telah diselesaikan antriannya. Penggunaan no 1 dan 2 untuk persiapan 2 tipe kejadian untuk model (1 = kedatangan dan 2 = keberangkatan)

Tabel 1. Kejadian

Deskripsi kejadian	Tipe kejadian
- Kedatangan pelanggan ke sistem	1
- Keberangkatan pelanggan dari sistem setelah selesai dilayani	2

Tabel 2.

Sub-routine, Fungsi -Fungsi dan Variabel-variabel untuk Model Antrian FORTRAN

Sub-program	Tujuan
Unit	Inisialisasi rutin
Timing Arrive	Waktu rutin kejadian rutin yang memproses tipe kejadian
Depart	Kejadian rutin yang memproses tipe kejadian 2 laporan umum
Report	Laporan umum (jika simulasi selesai)
Expon (R	Fungsi yang membangkitkan variabel random

mean)	<i>exponensial</i> dengan rata-rata R mean.
Parameter-Parameter Input	
MARRVT	Waktu antar kedatangan
M SERVT TOTCUS	Waktu rata-rata pelayanan jumlah total n pelanggan yang menunggu akan dilayani
Variabel-variabel Model	
ANIQ	Daerah pada no dalam fungsi antrian
DELAY	Pelanggan menunggu dalam antrian
NEVNTS	Jumlah tipe kejadian untuk model ini (menggunakan waktu rutin ; NEVNTS disini adalah 2
NEXT	Tipe kejadian (1 atau 2) dari kejadian yang akan terjadi berikut (penentuan dengan waktu rutin)
NIQ	Jumlah pelanggan dalam antrian
NUMCUS	Jumlah pelanggan yang telah menyelesaikan antrian
R MEAN	Jumlah variabel random <i>exponensial</i> untuk dibangkitkan
R MIN	Penggunaan variabel dengan <i>TIMING</i> untuk menentukan waktu kejadian dari sebagian besar kejadian yang terdekat
STATUS	Status pelayanan (status = 0 jika pelayan nganggur dan status = 1 jika pelayan sibuk)
TARRVL (I)	Waktu kedatangan pelanggan yang sedang menunggu dalam antrian (TARRVL) mempunyai dimensi 100
TIME	Simulasi <i>clock</i>
TLEVNT	Waktu kejadian berakhir
TNE (I)	Waktu kejadian berikutnya. Tipe kejadian (l = 1, 2) bagian dari daftar kejadian.
TOTDEL	Total penundaan/ menunggu dari semua pelanggan yang telah diselesaikan penundaannya.
V	Variabel random uniform distribusi antara 0 dan 1

Variabel-variabel Output	
AVGDEL	Rata-rata menunggu dalam antrian
AVGNIQ	Rata-rata waktu menunggu dalam antrian

Berkaitan dengan program utama, program dari sub routine dan fungsi-fungsi yang ditunjukkan pada tabel 2. Tabel juga menunjukkan penggunaan variable (variabel-variabel model termasuk variabel status, perhitungan statistik dan variabel yang digunakan untuk memfasilitasi penulisan program). Program utama, dimana tampilan *Fortran* diberikan pada gambar 1 terutama memulai dengan *Model Common Block*. Variabel -variabel dalam model adalah salah satu yang kita inginkan menjadi variabel global (jika sebuah variabel ada dalam model maka sebuah variabel yang tidak dalam model akan menjadi cabang ke sub-program) yang dilihatkan dua pernyataan sebelumnya, pernyataan umum khususnya modulus (*integer or real*) dari masing-masing penggunaan variabel dalam program utama. Kita telah memberi pernyataan model dari masing-masing variabel secara nyata, tanpa memperhatikan apakah yang akhir atau yang pertama dari variable, hal ini tidak berbeda atau berubah dengan model *Fortran* konvensional.

```
c
c *** MAIN PROGRAM.
c   INTEGER NEVNTS, NEXT, NIQ, NUMCUS, STATUS,
c   TOTCUS, REAL ANIQ, MARRVT, MSERVT, TARRVL
c   (100), TIME, TLEVNT, TNE (2), TOTDEL, COMMON/
c   MODEL/ ANIQ, MARRVT, MSERVT, NEVNTS, NEXT,
c   NIQ, NUMCUS, STATUS, TARRVL, TIME, TLEVNT, TNE,
c   TOTCUS, TOTDEL.
```

```
c *** 1 SPECIFY THE NUMBER OF EVENT TYPES FOR THE  
TIMING ROUTINE  
c  
    NEVNTS = 2  
c  
c *** READ INPUT PARAMETERS.  
c  
    READ 10, MARRVT, MSERVT  
10 FORMAT (2 F10. 0)  
    READ 20, TOT CUS  
20 FORMAT (110)  
c  
c .....  
c  
c  
c *** INITIALIZE THE SIMULATION  
c  
    30 CALL TIMING  
c  
c *** DETERMINE THE NEXT EVENT  
c  
    30 CALL TIMING  
c  
c *** 1 CALL THE APPROPRIATE EVENT ROUTINE.  
c  
    GO TO (40,50), NEXT  
40 CALL ARRIVE  
    GO TO 60  
50 ALL DEPART  
c  
c *** IF THE SIMULATION IS OVER, CALL THE REPORT  
GENERATOR AND END THE  
c *** SIMULATION. IF NOT, CONTINUE THE SIMULATION.  
c
```

```
60 IF (NUMCUS. LT. TOTCUS) GO TO 30  
CALL REPORT  
STOP  
END
```

Gambar 6

Tampilan FORTRAN untuk program utama, model antrian

Main Program

```
1  
INTEGER NEVNTS, NEXT, NIQ, NUMCUS, STATUS,  
TOTCUS, REAL ANIQ, MARRVT, MSERVT, TARRVL (100),  
TIME, TLEVNT, TNE (2), TOTDEL, COMMOND/ MODEL/  
ANIQ, MARRVT, MSERVT, NEVNTS, NEXT, MIQ, NUMCUS,  
STATUS, ITARRUL, TIME, TLEVNT, TNE, TOTCUS, TOTDEL.
```

Tipe Kejadian Khusus *Timing Routine*

Kita juga mengkhususkan model dari masing-masing variabel dalam model, namun demikian variabel tersebut tidak digunakan dalam program utama. Pernyataan umum diikuti dengan pengkhususan jumlah kejadian-kejadian untuk model, baca input parameter dan panggil/ sebut / NIT untuk permulaan model pada waktu = 0. Waktu rutin, TIMING kemudian dipanggil untuk menentukan tipe kejadian, NEXT dari kenyataan berikut yang akan terjadi dan untuk melanjutkan simulasi *clock*. TIME untuk waktu kejadian dari kejadian berikut, kemudian suatu 'perhitungan' melanjutkan pernyataan berdasarkan pada NEXT, digunakan untuk melewatkan control kepada kejadian yang sesuai. Jika NEXT=1, kejadian rutin ARRIVE dipanggil untuk proses kedatangan seorang pelanggan, jika

NEXT=2, kejadian rutin *depart* dipanggil untuk proses keberangkatan seorang pelanggan setelah penyelesaian pelayanan setelah control, yaitu kembali ke program utama dan ARRIVE or DEPART, sebuah control dibuat untuk melihat apakah jumlah pelanggan yang telah menyelesaikan antrian mereka. NUMCUS yang ditambah melalui masing-masing penyelesaian pelanggan dari antriannya adalah masih kurang dari jumlah pelanggan yang tertunda. Kita ingin meneliti, TOTCUS, selanjutnya TIMING dipanggil untuk melanjutkan simulasi jika sejumlah penundaan telah diteliti, laporan dibangkitkan, REPORT dipanggil untuk menghitung dan mencetak estimasi dari hasil yang diinginkan dan hasil simulasi diakhiri.

Sebuah daftar/ tampilan FORTRAN untuk subroutine/ NIT diberikan pada gambar 7, program sangat jelas, masing-masing pernyataan/ NIT berkaitan dengan pernyataan kembali dari komputer pada bagian pertama dari gambar 5. Tercatat bahwa waktu kedatangan pertama, TNE ditetapkan dengan penambahan variabel random *exponensial* dengan rata-rata MARRVET, yaitu EXPON (MARRVT), ke simulasi clock, TIME = 0.

```
SUBROUTINE INIT
INTEGER REVNTS, 1 NEXT, NIQ, NUMCUS, STATUS,
TOTCUS
REAL ANIQ, MARRVT, MSERVT, TARRVL, NEVNTS,
NEXT, TLEVNT, TNE (2), TOTDEL
COMMON/ MODEL/ ANIQ, MARRVT, MSERVT,
NEVNTS, NEXT, NIQ, NUMCUS, STATUS
TARRVL, TIME, TLEVNT, TNE, TOTCUS, TOTDEL
1
... INITIALIZE THE SIMULATION
TIME = 0
```

```
... INITIALIZE THE STATE VARIABLES
STATUS = 0
NIQ = 0
TLEVNT = 0
... INITIALIZE THE STATISTICAL COUNTERS
NUMCUS = 0
TOTDEL = 0
ANIQ = 0
... INITIALIZE THE EVENT LIST. SINCE NO
CUSTOMERS ARE PRESENT, THE
... TIME OF THE NEXT DEPARTURE (SERVICE
COMPLETION) IS SET TO
... ' INVITNY
TNE ( 1 ) = TIME + EXPON (MARRVT)
TNE ( 2 ) = 1 . E + 30
RETURN
END
```

Gambar 7

Tampilan FORTRAN untuk subrutin Inisialisasi rutin waktu kejadian (INIT), model antrian

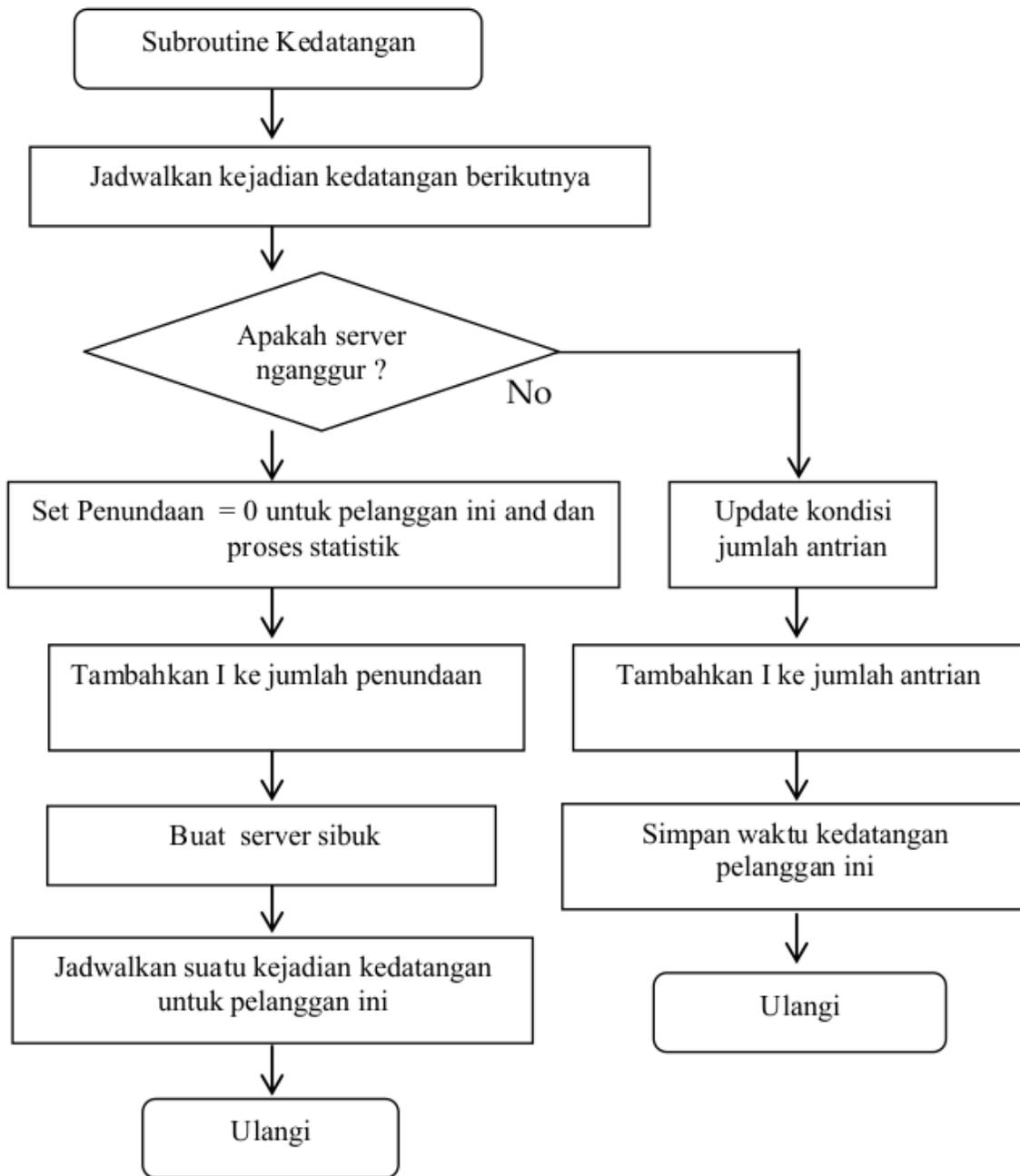
Kita secara jelas menggunakan TIME dalam pernyataan ini, walaupun mempunyai nilai 0, untuk menunjukkan formulir umum dari sebuah pernyataan untuk menentukan waktu sebuah kejadian mendatang. Karena tidak ada pelanggan yang datang pada TIME = 0, waktu berikutnya keberangkatan, TNE (2) adalah ditetapkan 1. E + 30 (notasi *Fortran* untuk 1030) yang akan menjamin bahwa kejadian pertama akan menjadi sebuah kedatangan.

```
SUBROUTINE TIMING
INTEGER NEVNTS, NEXT, NIQ, NUMCUS, STATUS,
TOTCUS
REAL ANIQ, MARRVT, MSERVT, TARRVL ( 100 ), TIME,
TLEVNT, TNE ( 2 ), TOTDEL
REAL RMIN
COMMON/ MODEL/ ANIQ, MARRVT, MSERVT,
NEVNTS, NEXT, NIQ, NUMCUS, STATUS
TARRVL, TIME, TLEVNT, TNE, TOTCUS, TOTDEL
RMIN = 1 . E + 29
NEXT = 0
c *** DETERMINE THE EVENT TYPE OF THE NEXT EVENT
      TO OCCUR
c
      DO 10 I = 1, NEVNTS
      IF ( TNE ( I ) . CE . RMIN ) GO TO 10
      RMIN = TNE ( I )
      NEXT = I
10 CONTINUE
c
c *** IF THE EVENT LIST IS EMPTY ( I . E . , NEXT = 0 )
      STOP YHE SIMULATION.
c *** OTHERWISE. ADVANCE THE SIMULATION CLOCK.
c
      IF ( NEXT . CT . O ) GO TO 30
      PRINT 20
20 FORMAT ( 1 H 1 . 5X . 'EVENT LIST EMPTY' )
      STOP
30 TIME = TNE ( NEXT )
      RETURN
      END
```

Gambar 8
Tampilan FORTRAN untuk subrutin Waktu simulasi,
model antrian

Sebuah tampilan FORTRAN untuk sub routine TIMING diberikan pada gambar 8, program membandingkan TNE (1), TNE (2), ... TNE (NEVNTS) dan menetapkan NEXT sama dengan tipe kejadian yang mana waktu kejadiannya paling kecil (catatan bahwa NEVNTS di set pada program utama). Pada kasus seri, jumlah kejadian paling kecil dipilih, kemudian waktu simulasi dilanjutkan untuk memilih tipe waktu kejadian yang terjadi, yakni TNE (NEXT). Program dibuat lebih menyeluruh, lebih lengkap dengan sebuah kontrol untuk kejadian yang tampilannya kosong, yang kita definisikan untuk rata-rata semua kejadian yang dijadwalkan untuk terjadi pada TIME 10^{30} . Jika kasus ini pernah seperti ditunjukkan oleh NEXT = 0, pesan yang salah 'daftar kejadian kosong' dicetak dan simulasi berakhir.

Sebuah *flowchart* dan tampilan program untuk kejadian subroutine ARRIVE diberikan pada gambar 9 dan 10 subroutine mulai dengan jadwal kejadian kedatangan berikutnya pada waktu TIME EXPON (MARRVT). Catatan bahwa TIXAE adalah waktu kedatangan pelanggan yang baru saja datang, kemudian sebuah control dibuat untuk menentukan apakah pelayan mengganggu, misalnya, jika STATUS = 0, jika pelayan mengganggu, tertundanya kedatangan pelanggan ditetapkan 0, penundaan ini ditambah ke penghitung TOTDEL, jumlah pelanggan-pelanggan tertunda yaitu: siapa yang telah menyelesaikan antrian mereka. NUMCUS ditambah dengan 1 menjadi 1 pelayan dibuat sibuk dan sebuah kejadian keberangkatan dijadwal untuk pelanggan ini pada waktu TIME + EXPON (MSERVT).



Gambar 9
Flowchart untuk subroutine Kedatangan

```
SUBROUTINE ARRIVE, INTEGER REVNTS, 1NEXT, NIQ,
NUMCUS, STATUS, TOTCUS, REAL ANIQ, MARRVT,
MSERVT, TARRVL ( 100 ), TLEVNT, TNE ( 2 ), TOTDEL,
REAL DELAY, COMMON/ MODEL/ ANIQ, MARRVT,
MSERVT, NEVNTS, NEXT, NIQ, NUMCUS, STATUS
1 TARRVL, TIME, TLEVNT, TNE, TOTCUS, TOTDEL
C
C *** SCHEDULE THE NEXT ARRIVAL
C
      TNE ( 1 ) = TIME + EXPON ( MARRVT )
C
C
.....
C
C
C *** IF THE SERVER IS IDLE, START SERVICE ON THE
ARRIVING CUSTOMER.
C
      IF ( STATUS . EQ . 1 ) GO TO 10
C
1C *** CUSTOMER HAS A DELAY OF ZERO. ( THE
FOLLOWING TWO STATEMENT ARE
C *** FOR PROGRAM CLARITY AND DO NOT AFFECT
THE RESULTS OF THE
C *** SIMULATION. )
      DELAY = 0
      TOTDEL = TOTDEL + DELAY
C
C *** ADD ONE TO THE NUMBER OF CUSTOMER
DELAYED.
C
      NUMCUS = NUMCUS + 1
C
C *** MAKE SERVER BUSY
```

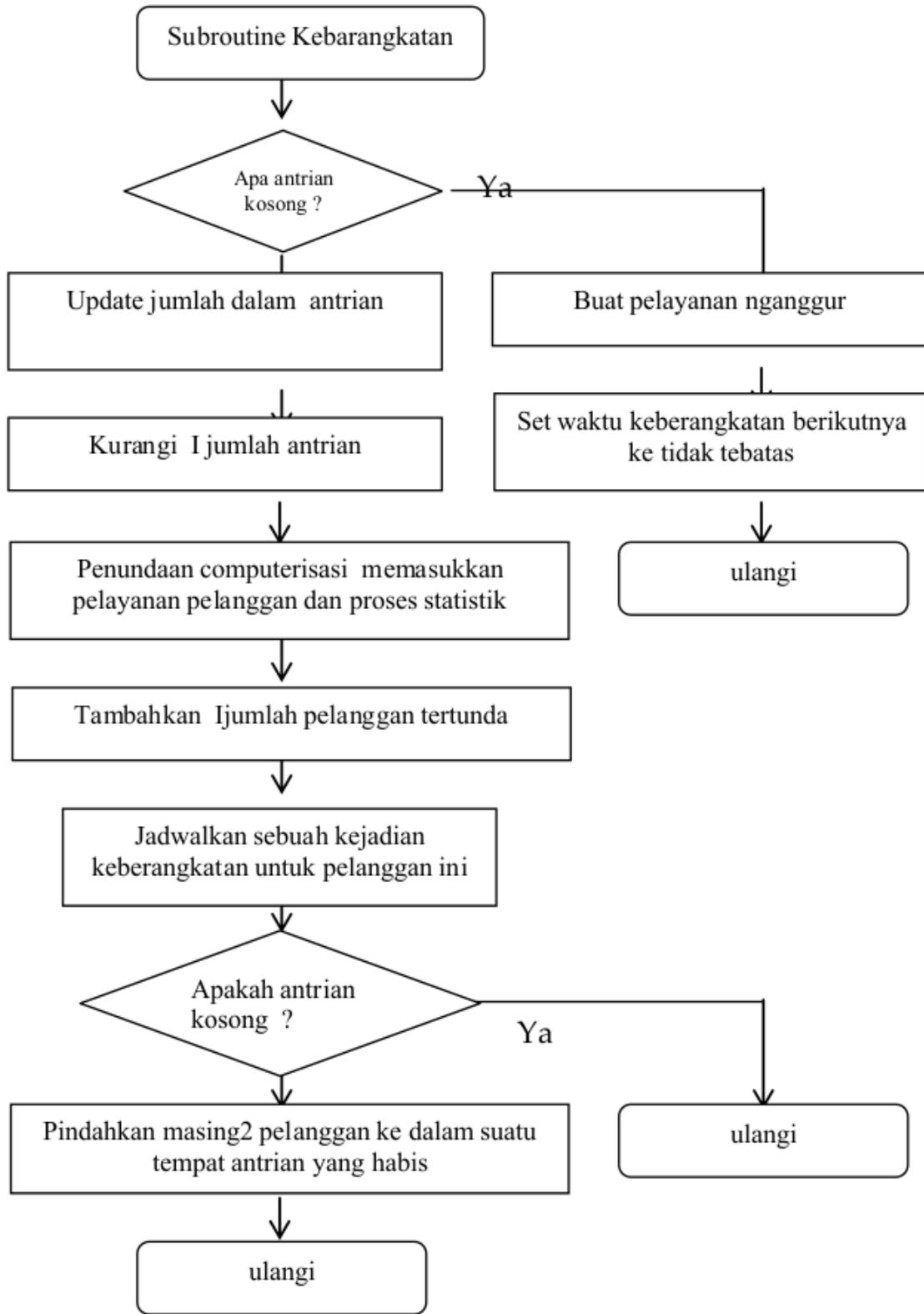
```
STATUS = 1
c
c *** SCHEDULE A DEPARTURE (SERVICE COMPLETION)
c
      TNE ( 2 ) = TIME + EXPON ( MSERVT )
      RETURN
c
.....
c
c *** SERVER IS BUSY. UPDATE THE AREA UNDER THE
      NUMBER IN QUEUE
c *** FUNCTION.
      10 ANIQ = ANIQ + NIQ * ( TIME - TLEVNT )
         TLEVNT = TIME
c
c *** ADD ONE TO NUMBER OF CUSTOMER IN THE
      QUEUE. IF AN OVERFLOW
c *** CONDITION EXISTS, STOP THE SIMULATION.
      OTHERWISE, STORE THE TIME
c *** OF ARRIVAL OF THE ARRIVING CUSTOMER IN THE
      ARRAY ' TARRVL .'
c
      NIQ = NIQ + 1
      IF ( NIQ . LE . 100 ) GO TO 30
      PRINT 20
20 FORMAT ( 1 H 1 , 5 X , 'OVERFLOW OF THE ARRAY
      TARRVL' )
      STOP
30 TARRVL ( NIQ ) = TIME
      RETURN
      END
```

Gambar 10
Tampilan FORTRAN untuk subrutin kedatangan,
model antrian

Dengan kata lain, jika pelanggan sibuk, area pada jumlah dalam antrian antara waktu kejadian terakhir. TLEVANT dan waktu sekarang, TIME ditambahkan ke perhitungan ANIQ dan waktu kejadian lalu di set ke waktu sekarang. Kemudian jumlah pelanggan dalam antrian ditambahkan dengan 1 kedatangan pelanggan masih dalam antrian. Jika jumlah (baru) dalam antrian berkurang atau sama dengan (\leq) 100 (jumlah susunan TARRVL), waktu kedatangan dari kedatangan pelanggan disimpan dalam tempat yang tepat dalam TARRVL mungkin ditentukan selama waktu berjalan. Jika tidak ada ruang dalam susunan TARRVL akan tersimpan waktu kedatangan baru, pesan kesalahan-kesalahan “kelebihan susunan TARRVL” dicetak dan simulasi berakhir.

Kejadian subroutine (rutin) DEPART yang mana flowchart dan tampilan diberikan dalam gambar 11 dan 12 dipanggil dari program utama ketika sebuah pelayanan selesai (dan berurutan keberangkatan-keberangkatan) terjadi jika keberangkatan pelanggan meninggalkan dibelakang tanpa/ tidak ada pelanggan (ditunjukkan dengan NIQ = 0) pelayan dibuat nganggur dengan menset STATUS=0 dan waktu berikutnya kedatangan, TNE (2) diset menjadi 10^{10} karena kejadian berikutnya pasti sebuah kedatangan (jika pernyataan ini nantinya diabaikan, program akan mendapatkan sebuah putaran yang tidak terbatas). Jika satu atau lebih pelanggan tertinggal dibelakang dengan memberangkatkan pelanggan, daerah pada jumlah fungsi dalam antrian diperbaharui dan waktu kejadian akhir diteliti, karena kedatangan pelanggan yang ke 1 dalam antrian akan mulai dilayani. Jumlah dalam antrian berkurang dengan 1, penundaan dari pelanggan ini diketahui sebagai TIME –

TARRVL (1) dan ditambah ke penghitung TOTDEL. Jumlah penundaan atau antrian ditambah dengan 1 dari kejadian keberangkatan dijadwalkan untuk pelanggan ini. Jika pelanggan dalam antrian lebih banyak, masing-masing waktu kedatangan dalam susunan TARRVL dipindahkan satu lokasi. Ini terjamin bahwa waktu kedatangan dari pelanggan memenuhi pelayanan setelah tertunda dalam antrian akan sering disimpan dalam TARRVL (1)



Gambar 11
Flowchart untuk subrutin keberangkatan, model antrian

```
SUBROUTINE DEPART
INTEGER NEVNTS, NEXT, NIQ, NUMCUS, STATUS,
1 TOTCUS, INTEGER I . I 1
REAL ANIQ, MARRVT, MSERVT, TARRVL ( 100 ),
TLEVNT, TNE ( 2 ), TOTDEL
REAL DELAY
COMMON/ MODEL/ ANIQ, MARRVT, MSERVT,
NEVNTS, NEXT, NIQ, NUMCUS, STATUS
1 TARRVL, TIME, TLEVNT, TNE, TOTCUS, TOTDEL
C
C *** 1 IF THE QUEUE IS EMPTY. MAKE THE SERVER IDLE
AND SET THE TIME OF
C *** THE NEXT DEPARTURE ( SERVICE COMPLETION )
TO ' INFINITY '
C
IF ( NIQ . CT . 0 ) GO TO 10
STATUS = 0
TNE ( 2 ) = 1 . E + 30
RETURN
C
.....
C
C *** QUEUE IS NOT EMPTY. UPDATE THE AREA UNDER
THE NUMBER IN QUEUE
C *** FUNCTION.
C
10 ANIQ = ANIQ + NIQ * ( TIME - TLEVNT )
TLEVNT = TIME
C
C *** SUBSTRACT ONE FROM THE NUMBER OF
CUSTOMER IN THE QUEUE.
C
1 NIQ = NIQ - 1
C
```

```
c *** COMPUTE DELAY OF THE CUSTOMER WHO IS
      BEGINNING SERVICE.
c
      DELAY = TIME - TARRVL ( 1 )
      TOTDEL = TOTDEL + DELAY
c
c *** ADD ONE TO THE NUMBER OF CUSTOMER
      DELAYED.
c
      NUMCUS = NUMCUS + 1
c
c *** SCHEDULE A DEPARTURE
c
      TNE ( 2 ) = TIME + EXPON ( MSERVT )
c
c *** IF THE QUEUE IS NOT EMPTY. MOVE IT CUSTOMER
      IN THE QUEUE UP ONE
c *** PLACE
c
      IF ( NIQ . EQ . 0 ) GO TO 30
      DO 20 I = 1 , NIQ
      I 1 = I + 1
      TARRVL ( I ) = TARRVL ( I 1 )
20 CONTINUE
30 RETURN
      END
```

Gambar 12
Tampilan FORTRAN untuk subrutin keberangkatan,
model antrian

Tampilan FORTRAN untuk subrutin diberikan pada gambar 13. Subrutin memulai dengan mencetak input parameter. Ini sering dinyatakan/disetujui untuk meyakinkan bahwa

input parameter dibaca kedalam komputer secara benar dan untuk mengidentifikasi input parameter sehubungan dengan sekumpulan data output. Rata-rata pelanggan menunggu, AVGDEL, dihitung sebagai TOTDEL/ NUMCUS dan rata-rata jumlah pelanggan dalam antrian, AVGNIQ, sebagai ANIQ/ TIME (waktu disini adalah waktu dimana pelanggan TOTCUS masuk dalam pelayanan) dan kemudian kedua data kuantitas dicetak. Pernyataan cetakan/ hasil print pada akhir subrutin REPORT membuat laporan simulasi dengan halaman yang bebas dari beberapa print yang asing.

Fungsi EXPON, yang membangkitkan variabel random exponential dengan rata-rata RMEAN ditampilkan dalam gambar 14. Program awal dengan mengisi sebuah variabel random U dengan distribusi normal pada interval 0 sampai dengan 1 (seperti sebuah variabel random selanjutnya akan dicatat dengan U (0,1). U (0,1) variabel random mempunyai fungsi kepadatan peluang sebagai berikut :

$$f(x) = \begin{cases} 1 - x & \text{if } 0 \leq x \leq 1 \\ 0 & \text{sebaliknya} \end{cases}$$

```
SUBROUTINE REPORT
INTEGER NEVNTS, NEXT, NIQ, NUMCUS, STATUS,
TOTCUS
REAL ANIQ, MARRVT, MSERVT, TARRVL ( 100 ), TIME,
TLEVNT, TNE ( 2 ), TOTDEL
REAL AVGDEL, AVGNIQ
COMMON/ MODEL/ ANIQ, MARRVT, MSERVT, NEVNTS,
NEXT, NIQ, NUMCUS, STATUS
TARRVL, TIME, TLEVNT, TNE, TOTCUS, TOTDEL

PRINT HEADING AND INPUT PARAMETER
```

```
PRINT 10
FORMAT ( 1 H 1 , 5X, ' SINGLE – SERVER QUEUEING
SYSTEM ' )
PRINT 20, MARRVT
FORMAT ( 1 H 0 , 5X, ' MEAN INTERARRIVAL TIME ' , 5X, F6
. 3 . ' MINUTES ' )
PRINT 30, MSERVT
FORMAT ( 1 H 0 , 5X, ' MEAN SERVICE TIME ' , 5X, F6 . 3 . '
MINUTES ' )
PRINT 40, TOTCUS
FORMAT ( 1 H 0 , 5 X , ' NUMBER OF CUSTOMERS ' , 9X , 15 )
```

1 COMPUTE AND PRINT ESTIMATES OF THE DESIRED MEASURES OF PERFORMANCE

```
VGDEL = TOTDEL/NUMCUS
VGNIQ = ANIQ/TIMES
PRINT 50
FORMAT ( // )
PRINT 60 , AVGDEL
FORMAT ( 1 H 0 , 5X, ' DELAY IN QUEUE ' , 5X, F6 . 3 . '
MINUTES ' )
PRINT 70, AVGNIQ
FORMAT ( 1 H 0 , 5X, ' NUMBER IN QUEUE ' , 5X, F6 . 3 . '
MINUTES ' )
PRINT 80
FORMAT ( 1 H 1 )
RETURN
END
```

Gambar 13
Tampilan FORTRAN untuk subrutin LAPORAN,
model antrian

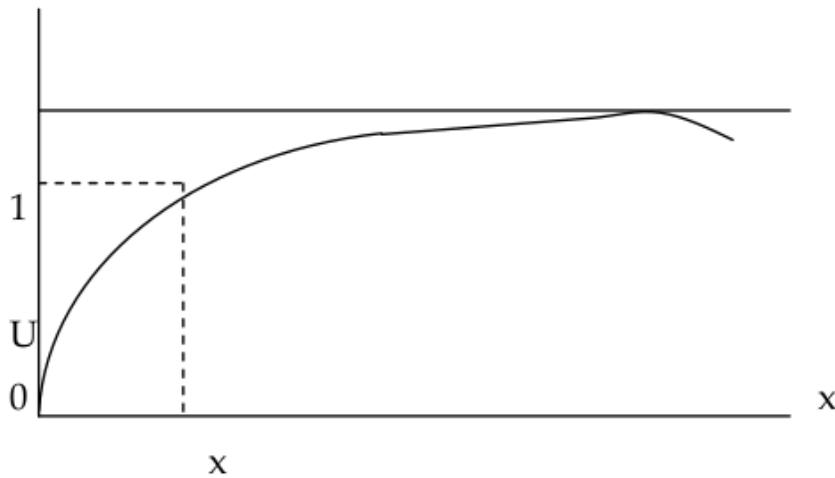
Ini untuk menunjukkan bahwa sebuah $U(0,1)$ variabel random jatuh dalam beberapa sub interval $[x, x + \Delta x]$ berisi dalam interval $(0,1)$ yaitu (uniform) Δx , sebagai $U(0,1)$ variabel random disebut “uniform” $U(0,1)$ variabel random adalah penyebab mendasar untuk model simulasi, akan kita lihat pada bab berikutnya. Beberapa random variabel penting dapat dibangkitkan dengan terlebih dahulu membangkitkan satu atau lebih $U(0,1)$ variabel random.

$U(0,1)$ variabel random U dibangkitkan pada komputer khusus dengan memanggil fungsi perpustakaan RANUN (Z). Mekanisme untuk membangkitkan $U(0,1)$ variabel random, seperti RANUN disebut pembangkit jumlah random dan akan kita didiskusikan pada bab berikutnya, harus menempatkan RANUN dengan nama dari jumlah pembangkit random mereka.

```
FUNCTION EXPON ( RMEAN )  
REAL MEAN , 0  
  
GENERATE A U ( 0 , 1 ) RANDOM VARIABEL  
THE FORM OF THIS STATEMENT  
DEPENDS ON THE COMPUTER USED  
  
U = RANUN ( Z )  
GENERATE AN EXPONENTIAL RANDOM VARIABEL WITH  
MEAN RMEAN  
EXPON = RMEAN * ALOG ( U )  
RETURN  
END
```

Gambar 14
Tampilan FORTRAN untuk fungsi EXPON, model antrian

$$F(x) = 1 - e^{-x/\beta}$$



Gambar 15

Pendekatan transformasi *inverse* untuk sebuah variabel random *exponensial*

Untuk membangkitkan sebuah variabel random *exponensial* dengan rata-rata RMEAN, logaritma netral (ln) dari U (0,1) variabel random U diambil dan dikalikan dengan nilai negatif dari RMEAN seperti terlihat pada gambar 14. Percobaan untuk beberapa pengertian mengapa metode ini menjadi sebuah kebenaran untuk membangkitkan sebuah variabel random. Jika x adalah sebuah variabel random *exponensial* dengan rata-rata β kemudian untuk beberapa $x \geq 0$ dimana peluang x adalah kurang dari atau sama dengan x adalah

$$P(X \leq x) = F(x) = \int_0^x f(t) dt = 1 - e^{-x/\beta} \quad (1.1)$$

Dimana $f(t)$ adalah fungsi densitas diberikan dengan persamaan (1,1) dan $f(x)$ adalah fungsi distribusi dari x. Jika U adalah sebuah U (0,1) variabel random, kemudian untuk beberapa $x \geq 0$.

$$P\{-\beta \ln U \leq x\} = P\{U \geq -e^{-x/\beta}\} = 1 - e^{-x/\beta}$$

sehingga variabel random $X = -\beta \ln U$ mempunyai sebuah distribusi *exponensial* dengan rata-rata β . Cara lain untuk memperoleh rumus yang tepat untuk X adalah menetapkan $U = F(x)$ dan menyelesaikan X , prosedur ini digambarkan pada gambar 15, disebut perubahan *inverse* untuk membangkitkan perubahan variabel-variabel random.

Ini harus diterangkan bahwa program *Fortran* diatas diberikan untuk M/M/I yang tentu saja tidak sesederhana, satu kemungkinan tetapi dirancang untuk menggambarkan bagaimana mungkin suatu organisasi sebuah program untuk sebuah simulasi yang lebih kompleks.

Output Simulasi dan Diskusi

Laporan simulasi diberikan pada gambar 16. Tercatat bahwa rata-rata waktu menunggu dalam antrian diperkirakan menjadi 0,497 menit dan waktu rata-rata jumlah pelanggan dalam antrian diperkirakan 0,500 (untuk 3 desimal), ini juga bukan kebetulan bahwa perkiraan adalah hampir persis/ dekat.

SYSTEM ANTRIAN – PELAYANAN TUNGGAL

Rata-rata Waktu Antar Kedatangan	= 1.000 Menit
Rata-rata Waktu Pelayanan	= 0.500 Menit
Jumlah Pelanggan	= 1.000
Rata-rata Menunggu dalam Antrian	= 0.497 Menit
Rata-rata Jumlah dalam Antrian	= 0.500

Gambar 16
Laporan Simulasi untuk Model Antrian

Dalam mempelajari beberapa simulasi, kita mungkin ingin memperkirakan keadaan/ kondisi model yang tetap, misal; sifat-sifat model yang telah dijalankan simulasi sepanjang (tidak terbatas) waktu untuk antrian sederhana M/M/I telah kita pertimbangkan, ini memungkinkan untuk menghitung secara analitik kondisi rata-rata waktu menunggu dalam antrian dan status kondisi tetap, jumlah waktu rata-rata dalam antrian. Kedua ukuran ini menghasilkan 0,5. Jadi kita ingin menentukan ukuran-ukuran status tetap ini, perkiraan kita berdasarkan pada rumus stop $n = 1000$ antrian-antrian agak berdekatan, namun demikian kita beruntung jika $n = 1000$ dipilih bergantian. Dalam praktek pemilihan rumus stop yang akan memberikan perkiraan bagus ukuran-ukuran kondisi tetap adalah cukup sulit untuk menggambarkan point ini, misalnya: untuk antrian M/M/I yang mana ukuran kedatangan pelanggan ditambah dengan 80% dari 1 per menit sampai 1,8 per menit. (Rata-rata waktu antar kedatangan sekarang adalah 0,556 menit) yang mana rata-rata waktu pelayanan tidak dirubah dan kita berharap dapat memperkirakan ukuran yang semu pada status I kondisi tetap dari hasil sepanjang $n = 1000$ antrian. Kita menghasilkan hasil simulasi ini dan dapat memperkirakan untuk rata-rata antri pada kondisi tetap dan jumlah rata-rata pada kondisi tetap dalam antrian 3,078 menit dan 5,501 berturut-turut. Karena nilai-nilai yang benar dari ukuran ini adalah 4,5 menit dan 8,1, ini jelas bahwa rumus pemberhentian tidak dapat dipilih bergantian. Kita diskusikan bagaimana memfokuskan hasil sepanjang simulasi untuk kondisi yang tetap dalam bahasan berikutnya.

Pembaca mungkin kaget mengapa kita tidak dapat memperkirakan waktu tunggu rata-rata dalam system dari seorang pelanggan, $d(n)$, dimana waktu menunggu seorang pelanggan didefinisikan sebagai interval waktu dari kedatangan pelanggan yang cepat / mendadak sampai waktu pelayanan pelanggan yang cepat pergi/ berangkat. Ada 2 alasan, pertama, banyak system antrian yang kita percayai bahwa pelanggan-pelanggan tertunda dalam antrian ketika menunggu pelayanan-pelayanan lain untuk diteliti, yaitu sebagian besar masalah pelanggan yang menunggu dalam system. Alasan kedua adalah efisiensi. Kebiasaan perkiraan $w(n)$ dan menjadi :

$$w(n) = \frac{\sum_{i=1}^n w_i}{n} = \frac{\sum_{i=1}^n D_i}{n} + \frac{\sum_{i=1}^n S_i}{n} = d(n) + \hat{S}(n) \quad (1.2)$$

dimana $W_i = D_i + S_i$ adalah waktu menunggu dalam sistem dari pelanggan ke i . karena distribusi waktu pelayanan akan diketahui untuk menghasilkan sebuah simulasi dalam tempat perkiraan atau rata-rata waktu pelayanan $E(S)$ akan diketahui juga dan alternatif estimator dari $w(n)$ adalah :

$$\hat{w}(n) = d(n) + E(S)$$

(catatan bahwa $s(n)$ adalah estimator $E(S)$ dari persamaan (1.2.) untuk semua simulasi antrian, $w(n)$ akan lebih efisien (variabel sedikit) perkiraan $w(n)$ daripada $\hat{w}(n)$ dan ini lebih baik (kedua estimator tidak bias). Oleh karena itu, jika seseorang menginginkan sebuah estimator $w(n)$, perkiraan $d(n)$ dengan $d(n)$ dan bertambah dikenal sebagai waktu perkiraan pelayanan, $E(S)$. Umumnya, moral adalah memperkirakan kembali melalui nilai-nilai perkiraan mereka, ketika memungkinkan estimator melakukan perubahan perkiraan

Bagian VIII

RUMUS-RUMUS ALTERNATIF PENYETOPAN

Pada contoh antrian diatas, simulasi diakhiri ketika NUMCUS, jumlah pelanggan tertunda sama dengan TOTCUS. Pada kasus ini, nilai akhir dari TIME adalah variabel random. Bagaimanapun untuk beberapa simulasi system antrian dunia nyata, diinginkan menghentikan simulasi setelah beberapa waktu yang tepat, katakan 480 menit (8 jam). Namun demikian, jika A_{1S} dan S_{1S} sebagai contoh kita, adalah variabel random kontinyu. Kemungkinan simulasi berakhir setelah tepat 480 menit adalah 0 (ketepatan yang sia-sia dari computer). Oleh karena itu, untuk menghentikan simulasi pada waktu-waktu tertentu, kita perkenalkan sebuah *dummy* (contoh "simulasi berakhir" (panggil) sebuah kejadian tipe 3) yang dijadwalkan untuk terjadi pada $TIME = 480$. ketika waktu kejadian ini terjadi, TNE (3) berkurang dari TNE (1) dan TNE (2), pembangkit laporan dipanggil dan simulasi diakhiri. Jumlah pelanggan menunggu sekarang adalah variabel random. Ide-ide ini dapat diimplementasikan pada program komputer dengan membuat perubahan dalam program utama. INIT dan REPORT, seperti

ditunjukkan pada gambar 17 sampai dengan 19. Pada gambar 17 tercatat bahwa NEVNTS sekarang adalah 3, dimana simulasi berjalan sepanjang yang diinginkan, TEND, adalah sekarang. Sebuah input parameter dan sebuah anggota umum MODEL blok (TOTCUS setelah diubah) dan pernyataan bahwa setelah “computer pergi ke pertanyaan” telah diubah. Perubahan hanya terjadi untuk INIT (kemudian ke model blok umum) adalah pernyataan pada TNE (3) = TEND, yang mana penjadwalan simulasi berakhir,. Pada gambar 19, bahwa pernyataan penelitian telah ditambah untuk memulai RAPORT untuk memperbaharui area pada Q (t) dari waktu akhir kejadian kedatangan atau kejadian keberangkatan sampai simulasi benar-benar berakhir, TEND (perlu dicatat bahwa ketika REPORT dipanggil, TIME =TEND). Kita harus benar-benar melihat kembali perubahan untuk ketiga program ini.

Jika sistem antrian telah benar-benar dipertimbangkan satu operator (pada tukang cukur) buka dari jam 9 pagi sampai dengan jam 5 sore, simulasi berhenti tepat setelah 8 jam yang mungkin meninggalkan pelanggan dengan rambut yang baru terpotong (tercukur) separuh. Pada kasus ini diinginkan untuk menutup salon setelah 8 jam, tetapi akan melanjutkan simulasi sampai semua pelanggan (jika ada) telah dilayani ketika salon akan ditutup. Pembaca ditanya pada problem ini, untuk memberi perubahan program yang penting untuk menerapkan rumus penyetopan ini

```
c
c *** MAIN PROGRAM.
c   INTEGER NEVNTS, NEXT, NIQ, NUMCUS, STATUS
c   REAL ANIQ, MARRVT, MSERVT, TARRVL (100),
c   TIME, TLEVNT, TNE (3), TOTDEL
c   COMMON/ MODEL/ ANIQ, MARRVT, MSERVT,
c   NEVNTS, NEXT, NIQ, NUMCUS, STATUS,
c   TARRVL, TEND, TIME, TLEVNT, TNE, TOTDEL.

c
c *** 1 SPECIFY THE NUMBER OF EVENT TYPES FOR THE
c   TIMING ROUTINE

c
c   NEVNTS = 3

c
c *** READ INPUT PARAMETERS.
c
c   READ 10, MARRVT, MSERVT
c   10 FORMAT ( 2 F 10. 0 )
c   READ 20, TENT
c   20 FORMAT ( F 10 . 0 )

c
c   .....

c
c *** INITIALIZE THE SIMULATION
c
c   CALL INIT

c
c *** DETERMINE THE NEXT EVENT
c
c   CALL TIMING

c
c *** CALL THE APPROPRIATE EVENT ROUTINE.
c
c   GO TO ( 40, 50, 60 ), NEXT
```

```
40 CALL ARRIVE
   GO TO 30
50 CALL DEPART
   GO TO 30
C
C *** 1 THE SIMULATION IS OVER. CALL THE
      REPORT GENERATOR AND END
C *** THE SIMULATION
C
60 CALL REPORT
   STOP
   END
```

Gambar 17

**TAMPILAN Program Utama FORTRAN, Model Antrian
dengan Hasil Tetap Sepanjang Simulasi**

```
SUBROUTINE INIT
1 INTEGER NEVNTS,NEXT,NIQ,NUMCUS,STATUS, REAL
  ANIQ,MSERT,TARRVL(100),TEND,TIME,TLEVNT,TNE(3),T
  OTDEL, COMMON /MODEL/
  ANIQ,MARRVT,MSERVT,NEVNTS,NEXT,NIQ,NUMCUS,ST
  ATUS,1TARRVL,TEND,TIME,TLEVNT,TNE,TOTDEL
C
C ### INITIALIZE THE SIMULATION CLCK
C
      TIME
C
C ### INITIALIZE THE STATE VARIABELS.
      STATUS=0
      NIQ=0
      TLEVNT=0
C
C 1 INITIALIZE THE STATISTICAL COUNTERS.
```

```
C
  NUMCUS=0
  TOTDEL=0
  ANIQ=0

C
C ### INITIALIZE THE EVENT LIST. SINCE NO CUSTOMERS
ARE PRESENT, THE
C ### TIME OF THE NEXT DEPARTURE (SERVICE
COMPLETION) IS SET TO
C ### 'INFINITY'
C
  TNE(1)=TIME+EXPON(MARRVT)
  TNE(2)=1.E+30
  TNE(3)=TEND
  RETURN
  END
```

Gambar 18
Tampilan FORTRAN untuk subrutin INIT, model antrian
dengan panjang run yang tetap

```
SUBROUTINE REPORT
INTEGER NEVNTS,NEXT,NIQ,NUMCUS,STATUS
REAL, ANIQ,MSERT,TARRVL(100),TEND,TIME,TLEVNT,
TNE(3),TOTDEL REAL AVGDEL,AVGNIQ
COMMON /MODEL/
ANIQ,MARRVT,MSERVT,NEVNTS,NEXT,NIQ,NUMCUS,ST
ATUS,1TARRVL,TEND,TIME,TLEVNT,TNE,TOTDEL
C ### UPDATE THE AREA UNDER THE NUMBER IN QUEUE
FUNCTION TO THE C ### END OF THE SIMULATION.
  ANIQ=ANIQ+HIQ*(TIME-TLEVNT)
C ### PRINT HEADING AND INPUT PARAMETERS.
```

```
PRINT 10
10 FORMAT (1H1,5X,'SINGLE-SERVER QUEUEING
SYSTEM')
PRINT 20,MARRVT
20 FORMAT 91H0,5X,'MEAN ANTERARRIVAL
TIME',5X,F6.3,'MINUTES')
PRINT 30,MSERT
30 FORMAT(1H0,5X,'MEAN SERVICE
TIME',10X,F6.3,'MINUTES')
PRINT 40,TEND
40 FORMAT(1H0,5X,'LENGTH OF THE
SIMULATION',2X,F7.3,'MINUTES')
C
C ### 1 COMPUTE AND PRINT ESTIMATES OF THE DESIRED
MEASURES OF PERFORMANCE
C
AVGDEL=TOTDEL/NUMCUS
AVGNIQ=ANIQ/TIME
PRINT 50
50 FORMAT(//)
PRINT 60,AVGDEL
60 FORMAT (1H0,5X,'AVERAGE DELAY IN
QUEUE',5X,F6.3,' MINUTES')
PRINT 70,AVGNIQ
70 FORMAT (1H0,5X,'AVERAGE NUMBER IN
QUEUE',4X,F6.3)
PRINT 80
80 FORMAT (1H1)
RETURN
END
```

Gambar 19
Tampilan FORTRAN untuk subrutin REPORT,
model antrian dengan panjang run yang tetap

Bagain IX

SIMULASI SISTEM PERSEDIAAN

Kita dapat melihat bagaimana simulasi dapat digunakan untuk membandingkan alternatif kebijakan-kebijakan pemesanan untuk sebuah sistem persediaan. Beberapa elemen dari model yang representatif telah ditemukan dalam sistem persediaan yang nyata/ sebenarnya.

Pernyataan Problem

Sebuah perusahaan yang menjual sebuah produk tunggal dan akan diputuskan berapa banyak item-item yang akan menjadi persediaan untuk masing-masing bulan (n) berikutnya. Waktu antar permintaan (*lead time*) mengikuti distribusi independen (independen distribusi) variabel random *ekspensial* dengan rata-rata 0, 1 bulan. Jumlah permintaan (D) adalah mengikuti variabel random IID, sebagai berikut:

$$D = \begin{cases} 1 \text{ D.P } \frac{1}{6} \\ 2 \text{ D.P } \frac{1}{3} \\ 3 \text{ D.P } \frac{1}{3} \\ 3 \text{ D.P } \frac{1}{6} \end{cases}$$

Dimana $D.P$ dibaca "dengan probabilitas". Pada awal bulan, perusahaan mengecek tingkat persediaan dan memutuskan berapa banyak item-item akan dipesan dari *supplier* jika perusahaan memesan Z item-item biaya tambahan $K+iZ$, dimana $K = \$32$ adalah biaya set up dan $n=\$3$ biaya tambahan peritem. (Jika $Z= 0$, tidak ada biaya tambahan), jika sebuah pesanan sudah diletakkan, waktu yang diperlukan untuk meletakkan pesanan ini sampai datang (disebut keterlambatan pengiriman) adalah sebuah variabel random yang mengikuti distribusi uniform dengan interval 0,5 sampai 1 bulan.

Perusahaan menggunakan ketepatan (S,s) kebijakan untuk memutuskan berapa banyak akan dipesan, missal

$$Z= S-I$$

$$0$$

Jika $I < s$

Jika $I \geq s$

Dimana I adalah level persediaan awal bulan (sebelum memesan), ketika sebuah permintaan terjadi, dapat dipenuhi jika level persediaan \geq jumlah permintaan. Jika jumlah permintaan $>$ persediaan, kelebihan permintaan dari *supply* maka tidak bisa dipenuhi dan bisa dipenuhi pada pengiriman mendatang. Dalam kasus ini persediaan baru= persediaan lama dikurangi jumlah permintaan. Sebuah bilangan negatif bulat, ketika sebuah pesanan datang, ini dipergunakan pertama kali untuk mengurangi sejumlah pesanan yang mungkin tidak bisa dipenuhi (jika ada) dan untuk mengingat pesanan (jika ada) yang akan ditambah kedalam persediaan.

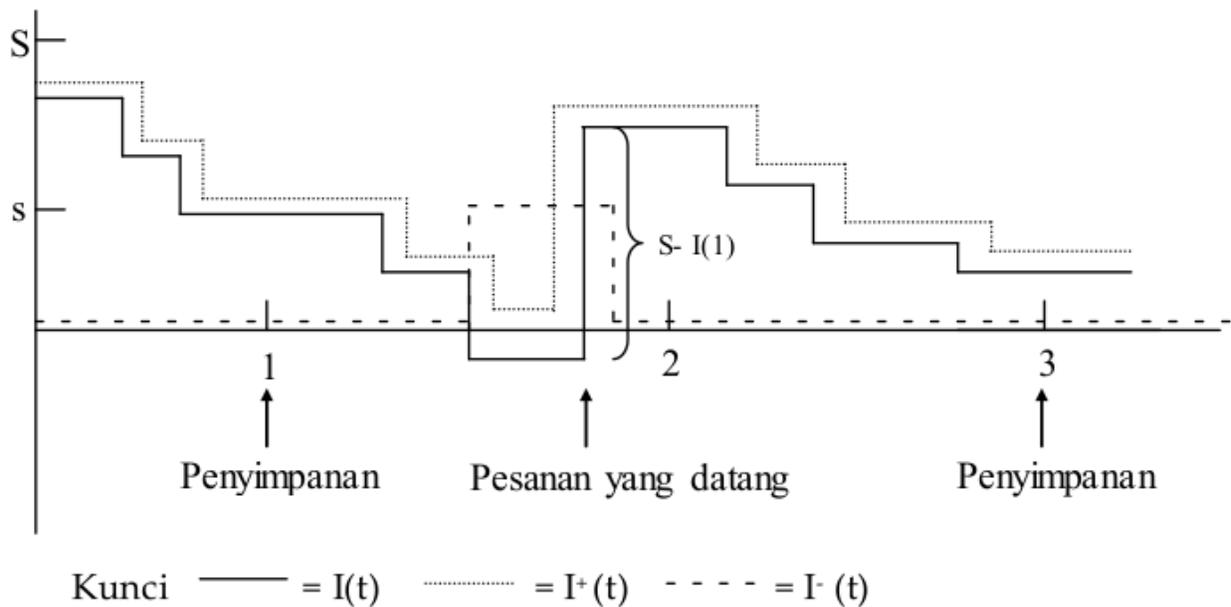
Sejauh yang telah kita diskusikan hanya satu tipe yang terjadi dalam sistem persediaan, yaitu biaya pemesanan, padahal sebagian besar sistem persediaan dalam dunia nyata mempunyai 2 biaya tambahan yaitu biaya penyimpanan dan biaya resiko, yang kita bicarakan setelah memperkenalkan beberapa simbol tambahan sebagai berikut;

$I(t)$ = level persediaan pada waktu t , mungkin positif, nol atau negatif

$I^+(t) = \max \{I(t), 0\}$
=Jumlah item-item kecuali pada waktu persediaan t $\{I^+(t) \geq 0\}$

$I^-(t) = \max \{-I(t), 0\}$
= permintaan tidak bisa dipenuhi pada waktu t $\{I^-(t) \geq 0\}$
kemungkinan terjadi $I(t)$, $I^+(t)$ dan $I^-(t)$ untuk $t \geq 0$

ditunjukkan pada gambar 20



Gambar 20
Sebuah Realisasi dari $I(t)$, $I^+(t)$ dan $I^-(t)$

Pada saat titik-titik pada $I(t)$ berkurang waktu lembur permintaan untuk model ini, kita asumsikan bahwa pesanan menambah rata-rata biaya simpan untuk n bulan

$$h \frac{\int_0^n I^+(t) dt}{n}$$

dimana $n = \$1$ per item per bulan dan $\frac{\int_0^n I^+(t) dt}{n}$ adalah jumlah rata-rata item terkecuali dalam persediaan, biaya simpan termasuk sewa gudang, asuransi, pajak dan pemeliharaan, termasuk biaya kesempatan mendapatkan keuntungan/ model hubungan dengan adanya persediaan dibanding investasi ke yang lain. Kita telah mengenal dalam rumus kita bahwa kenyataannya biaya penyimpanan beberapa biaya penyimpanan diabaikan jika $I^+(t) = 0$, namun demikian, jika tujuan kita untuk membandingkan kebijakan-kebijakan pemesanan faktor pengabaian ini (yang bebas dari kebijakan yang dipakai) dan tidak akan mempengaruhi keputusan kita dari kebijakan yang terbaik.

Kita juga mengasumsikan bahwa perusahaan menambah sebuah biaya resiko untuk n bulan.

$$\mu = \frac{\int_0^n I^-(t) dt}{n}$$

dimana $\pi = \$5$ per item perbulan dan $\frac{\int_0^n I^-(t) dt}{n}$ adalah (back log) rata-rata jumlah item yang tidak bisa dipenuhi. Biaya resiko termasuk biaya tambahan pencatatan ketika back log terjadi dan juga suatu biaya akibat terjadinya kehilangan pelanggan.

Asumsikan bahwa I awal = 60 % dan tidak ada pemesanan, kita mensimulasikan sistem persediaan untuk $n = 120$ bulan (10 tahun) dan rata-rata total penggunaan biaya perbulan yaitu jumlah rata-rata biaya pemesanan perbulan, rata-rata biaya penyimpanan perbulan dan rata-rata biaya resiko perbulan akan dibandingkan. Berikut kebijakan-kebijakan pemesan persediaan.

S	20	20	20	20	40	40	40	60	60
S	40	60	80	100	60	80	100	80	100

Kita bekerja tanpa petunjuk bagaimana kebijakan tertentu dipertimbangkan untuk dipilih. Teknik –teknik statistik untuk membuat sesuatu ketetapan akan didiskusikan pada bab berikutnya. Ini akan dicatat bahwa variabel yang dinyatakan untuk sebuah model simulasi dari sistem persediaan adalah Tingkat Persediaan $I(t)$, jumlah pemesanan dari perusahaan ke supplier dan waktu kejadian terakhir yang merubah tingkat persediaan (yang diperlukan untuk menghitung fungsi luas area pada $I^+(t)$ dan $I^-(t)$)

Program FORTRAN

Tabel 3. Model system persediaan dengan tipe kejadian sebagai berikut :

Diskripsi kejadian	Tipe kejadian
Kejadian pemesanan ke perusahaan dari supplier	1
Permintaan produk dari pelanggan	2
Berakhirnya simulasi setelah n bulan	3
Evaluasi inventory (dari kemungkinan pesan) pada setiap awal bulan	4

Kita telah memilih akan mengakhiri simulasi pada kejadian tipe 3 dari pada tipe 4 karena pada WAKTU =120 diantara kejadian "simulasi berakhir" dan " evaluasi persediaan akan segera dijadwalkan dan kita lebih menyukai untuk mengerjakan kejadian pada saat ini. (Jika simulasi berakhir pada WAKTU= 120, tidak bermaksud mengevaluasi persediaan atau kemungkinan pesan), eksekusi/ pengerjaan tipe kejadian 3 dan 4 ditunda/ diperpanjang sebab waktu rutin, WAKTU yang akan memberi pilihan ke jumlah yang paling rendah walaupun 2 atau lebih kejadian dijadwalkan akan terjadi bersamaan (ini disebut jam karet). Umumnya sebuah simulasi akan dirancang untuk memproses kejadian-kejadian dalam sebuah periode/ urutan yang tepat jika jam karet terjadi.

Tabel 4. Subprograman Variabel Fortran untuk model persediaan

Sub program	Maksud
INIT	Rutin awal/ permulaan
TIMING	Waktu rutin/ rutin kejadian yang memproses tipe kejadian
ORDARV	Rutin kejadian yang memproses tipe kejadian 1
DEMAND	Rutin kejadian yang memproses tipe kejadian 2
REPORT	Rutin kejadian yang memproses tipe kejadian 3 (laporan pembangkit)
EVALU 8	Rutin kejadian yang memproses tipe kajian 4
UPDATE	Subrutin untuk mengupdate fungsi area $I^+(t)$ dan $I^-(t)$ jika tingkat persediaan berubah dan jika simulasi berakhir (tidak ada kejadian rutin)
EXPON (R. MEAN)	Fungsi yang membangkitkan variabel random eksponensial dengan rata-rata R. Mean

RANDI (Z)	Fungsi yang membangkitkan sebuah integer random antara I dan Fungsi distribusi PROBD (I-1,2,...n nilai dimana Z adalah jika variabel semu, jika X adalah peluang integer random di mana X bernilai $\leq I$ yang diberikan melalui PROBD dst
UNIFRM (A,B)	Fungsi yang membangkitkan sebuah variabel distribusi uniform pada interval [A, B] dimana A dan B adalah nilai real dan A harus $\leq B$

Tabel 5 Parameter Input

Parameter Input	Definisi
BIGS	S1 nomor ke 2 dalam spesifikasi dari (S1 S) Kebijakan persediaan
H	H, Penjumlahan untuk rata-rata biaya penyimpanan
INCRMC	i, Pertambahan biaya pesabn peritem

Tabel 6 Variabel Model

Varibel-Variabel Model	Definisi
Aminus	Fungsi area pada I ⁻ (t)
AMOUNT	Jumlah , Z pemesanan
APLUS	Fungsi area under I ⁺ (t)
DSIZE	Jumlah permintaan tertentu
INVLEV	I ⁺ (t), level persediaan
NEVNTS	Nomer tipe kejadian(missal 1,...,4)
NEXT	Tipe kejadian (1,2,3 atau 4) yang akan terjadi berikutnya
TIME	Waktu Simulasi
TLEVNT	Waktu kejadian terakhir yang merubah level persediaan

TNE (1)	Waktu kejadian terjadinya tipe kejadian I (I=1,2,...,4)
TORDC	Total biaya pemesanan
TSLE	Waktu kejadian akhir

Variabel-variabel output	Definisi
ACOST	Rata-rata total biaya perbulan
AHLDC	Rata-rata biaya simpan perbulan
AORDC	Rata-rata pesan perbulan
ASHRC	Rata-rata biaya kekurangan perbulan

Catatan $AORDC = \frac{TDRDS}{NMNTHS}$

$$AHLDC = H * \left[\frac{APLUS}{NMNTHS} \right]$$

$$ACOST = AORDC + AHLDC + ASHRC$$

Program komputer untuk model ini terdiri dari sebuah program utama dan sub programs dan tampilan variabel-variabel FORTRAN pada bab berikutnya.

C

C ### MAIN PROGRAM

C

INTEGER, AMOUNT,BIGS
 ,INITIL,INVLEV,NEVNTS,NEXT,NMNTHS,SMALL
 INTEGER I, NPOLICY

```
REAL
AMINUS,AMOUNT,INCRMC,MDEMDT,PI,SETUPC,TIME,
TLEVNT,TNE(4,1 TORDC
COMMON /MODEL/
AMINUS,AMOUNT,APLUS,BIGS,N,INCRMC,INITIL,IN
LEV,1MDEMNT,NEVNTSNEXT,MNMTHS,PI,SETUPC,S
MALL,TIME,TLEVNT,TNE,TORDC
COMMON /RANDOM/ NVALUE,PROBD(25)
C
C ### 1 SPECIFY THE NUMBER OF EVENT TYPES FOR THE
TIMING ROUTINE.
C
    NEVNTS=4
C
C ### READ INPUT PARAMETER
C
    READ 10,INITIL,NMNTHS,NPOLICY,NVALUE
10 FORMAT (5F10)
    READ 20,MDEMNT,SETUPC,INCRMC,NVALUE
20 FORMAT (5F10.0)
    READ 30,(PROBD(1),I-1,NVALUE)
30 FORMAT (8F10,0)
C
#####
C
C #### PRINT REPORT HEADING
C
    PRINT 40
40 FORMAT (1H1,5X'SINGLE-PRODUCT INVENTORY
SYSTEM')
    PRINT 50,INITIL
50 FORMAT (1H0,5X,'INITIAL INVENTORY
LEVL',21X,13,'ITEMS')
    PRINT 60,NVALUE
```

```
60 FORMAT (1H0,5X,'NUMBER OF DEMAND
SIZES',22X,13)
    PRINT 70, (PROBD(I),I=1,NVALUE)
70 FORMAT (1H0,5X,'DISTRIBUTION FUNCTION OF
DEMAND SIZES' 5X,A(F5.3,3X1))
    PRINT 80, MDEMNT
80 FORMAT (1H0,5X,'MEAN INTERDEMAND
TIME',21X,F5.2,'MONTHS')
    PRINT 90, MNMTHS
90 FORMAT (1H0,5X,'LENGTH OF THE
SIMULATION',20X,13,'MONTH')
    PRINT 100,SETUPC,INCRMC,N,PI
100 FORMAT (1H0,5X,
'K=',F5.1,3X,'I=',F5.1,3X,'N=',F5.1,3X,'PI-',1,F5.1)
    PRINT 110
110 FORMAT (//)
    PRINT 120
120 FORMAT (1H0,7X,'POLICY',6X,'AVERAGE
COST',5X,'AVERAGE ORDERING
COST1',5X,'AVERAGE HOLDING COST',5X,'AVERAGE
SHORTAGE COST')
C #####
C
C ### RUNT THE SIMULATION VERYING 1THE
    INVENTORY POLICY.
C
    DO 190 I=1,NPOLICY
C
C ### READ INVENTORY POLICY.
C
    READ 130,SMALL,BIGS
130FORMAT (2110)
C
C ### INITIALIZE THE SIMULATION
```

```
C
  CALL INIT
C
C ### DETERMINE THE NEXT EVENT
C
  140 CALL TIMING
C
C ### CALL THE APROPRIATE EVENT ROUTINE
C
  1 GO TO (150,160,180,170),NEXT
150 CALL ORDARV
      GO TO 140
160 CALL DEMAND
      GO TO 140
170 CALL EVALU8
      GO TO 140
180 CALL REPORT
190 CONTINUE
      PRINT 200
200 FORMAT (1H1)
      STOP
      END
```

Gambar 21
FORTRAN, tampilan program, model persediaan

Tampilan FORTRAN untuk program utama diberikan pada gambar 21. Program dimulai dengan MODEL dan RANDOM blok umum (nanti berguna karena fungsi RANDOM di gunakan), Tetapkan NEVRTS = 4 dan baca input parameter-parameter. Selanjutnya jika subrutin REPORT suatu kejadian akan dipanggil untuk masing-masing kebijakan persediaan yang

akan dipertimbangkan, bagian depan / kepala laporan dicetak pada program utama (lebih disukai dari pada REPORT). Akhirnya, untuk masing-masing *run* simulasi yang diinginkan (sesuai dengan kebijakan masing-masing), nilai s dan s untuk *run* ini dibaca dari sebuah pencatat data, simulasi diawali dengan memanggil Subrutin INIT, tipe kejadian dari kejadian berikutnya akan terjadi, NEXT, ditetapkan dengan memanggil subrutin TIMING dan NEXT digunakan untuk melewati kontrol untuk mencocokkan kejadian rutin. Pada kondisi tertentu jika NEXT=3, REPORT dipanggil dan simulasi diakhiri untuk kebijakan yang telah ditetapkan. Pembaca akan mencatat pertimbangan yang hampir sama antara program utama untuk model persediaan dan antrian.

Subrutin INIT yang agak jelas ditampilkan pada gambar 22. Penelitian pertama pada evaluasi persediaan dijadwalkan pada WAKTU=0, karena umumnya persediaan awal dapat kurang dari s . Subrutin TIMING (pemilihan waktu) yang (kecuali untuk model blok umum) adalah identik dengan salah satu penggunaan untuk model antrian. ditampilkan pada gambar 23.

```
SUBROUTINE INIT
INTEGER AMOUNT, BIGS, INITIL, INVLEV, NEVNTS,
NEXT, NMNTHS, SMALLS
REAL
AMINUS, APLUS, H, INCRMC, MDEMMDT, PI, SETUPC, TI
ME, TLEVNT, TNE(4), 1TORDC
COMMON/MODEL/AMINUS, AMOUNT, APLUS, BIGS, H
, INCRMC, INITIL, INVLEV, IMDEMMDT, NEVNTS, NEXT, N
MNTHS, PI, SETUPC, SMALLS, TIME, TLEVNT, TNE, TOR
DC
```

C

```
1
C ### INITIALIZE THE SIMULATION CLOCK.
C
      TIME=0
C ### INITIALIZE THE STATE VARIABELS.
      INVLEV=INITIL
      TELVNT=0.

C
C ### INITIALIZE THE STATISTICAL COUNTERS.
      TORDC=0.
      APLUS=0.
      AMINUS=0.

C
C ### INITIALIZE THE EVENT LIST. SINCE NO ORDER IS
      OUTSANDING, THE TIME
C ### OF THE NEXT ORDER ARRIVAL IS SET TO 'INFINITY.'
C
      TNE (1)=1.E+30
      TNE (2)= EXPON (MDEMMDT)
      TNE (3)= NMNTHS
      TNE (4)= 0.
      RETURN
      END
```

Gambar 22

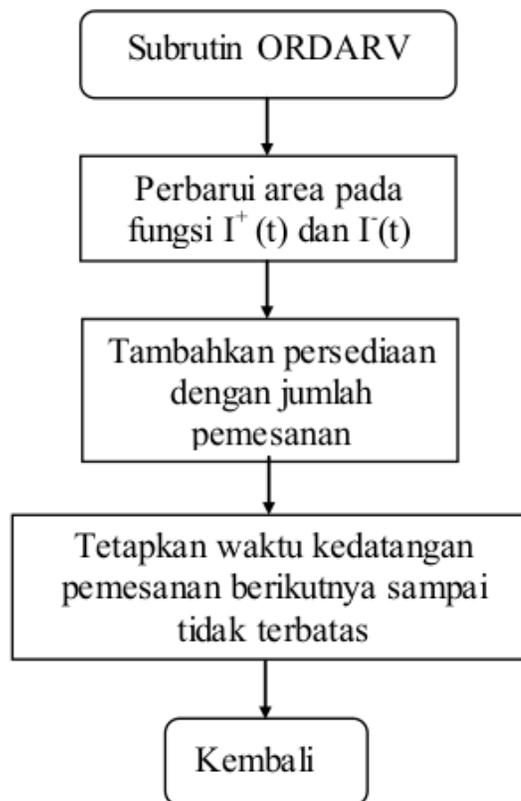
Tampilan FORTRAN untuk subrutin INIT, model persediaan

```
      SUBROUTINE TIMING
      INTEGER
      AMOUNT,BIGS,INITIL,INVLEV,NEVNTS,NEXT,NMNTHS,SM
      ALLS
```

```
REAL
AMINUS,APLUS,N,INCRMC,MDEMDT,PI,SETUPC,TIME,
TLEVNT,TNE(4),1 TORDC
REAL RMIN
COMMON /MODEL/
MINUS,AMOUNT,APLUS,BIGS,N,INCR,INITIL,INVLEV,
1
MDEMDT,NEVNTS,NEXT,NMNTHS,PI,SETUPC,SMALLS,
TIME,TLEVNT,TNE,TORDC
RMIN=1.E+29
NEXT=0
C
C ### DETERMINE THE EVENT TYPE OF THE NEXT EVENT
TO OCCUR
C
      DO 10I-1,NEVNTS
      IF(TNE(i).GE.RMIN) GO TO 30
      PRINT 20
10 CONTINUE
C
C ### IF THE EVENT LIST IS EMPTY (I.E., NEXT=0),STOP THE
SIMULATION.
C ### OTHERWISE, ADVANCE THE SIMULATION CLOCK.
C
      IF(NEXT.GT.0) GO TO 30
      PRINT 20
20 FORMAT (1h1,5X,'EVENT LIST EMPTY')
      STOP
30 TIME=TNE(NEXT)
      RETURN
      END
```

Gambar 23
Tampilan FORTRAN untuk subrutin pemilihan waktu,
model persediaan

Sebuah flowchart dan tampilan untuk kejadian subrutin ORDARV diberikan pada gambar 24 dan gambar 25 pertama program memanggil subrutin UPDATE untuk memperbaiki area pada $I^+(t)$ dan area pada $I(t)$ (lihat diskripsi UPDATE dibawah ini untuk lebih detail). Level persediaan akan berubah pada tipe kejadian ini. selanjutnya level persediaan, INVLEV, sebelumnya ditambah dengan jumlah pemesanan dan supplier, AMOUNT, dan waktu berikutnya pemesanan datang, TNE (I) diset sampai 10^{10} karena sebuah pemesanan tidak begitu panjang bila di selesaikan



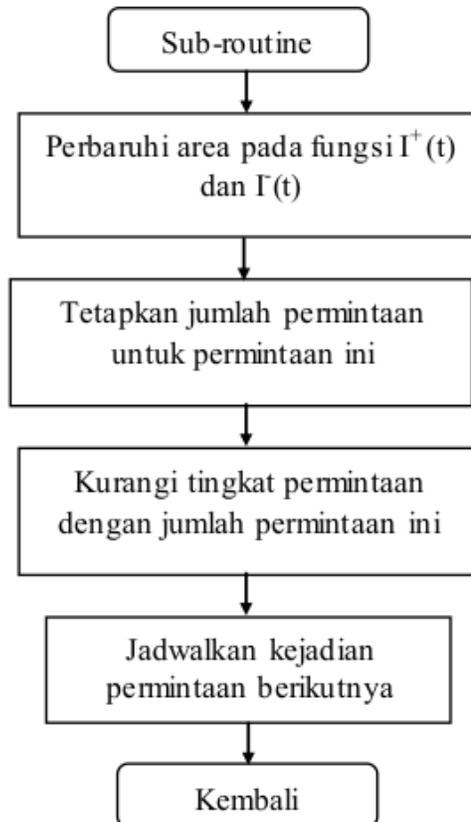
Gambar 24
Flow chart sub routine ORDARV, model persediaan

```
SUBROUTINE ORDARV
  1
  INTEGER
  AMOUNT,BIGS,INITIL,INVLEV,NEVNTS,NEXT,NMNTHS,SM
  ALLS
  REAL,AMINUS,APLUS,N,INCRMC,MDEMDT,PI,SETU
  PC,TIME,TLEVNT,TNE(4),1 TORDC
  COMMON /MODEL/
  MINUS,AMOUNT,APLUS,BIGS,N,INCR,INITIL,INVLEV,
  1
  MDEMDT,NEVNTS,NEXT,NMNTHS,PI,SETUPC,SMAL
  LS,TIME,TLEVNT,TNE,TORDC
C
C ### UPDATE 'APLUS AND 'AMINUS.'
C
  CALL UPDATE
C
C ### INCREMENT THE INVENTORY LEVEL BY THE
  AMOUNT ORDERED.
C
  INLEV=INLEV+AMOUNT
C
C ### SINCE NO ORDER IS NOW OUTSTANDING, SET THE
  TIME OF THE NEXT ORDER
C ### ARRIVAL TO 'INFINITY'
  TNE(1)=1.E+30
  RETURN
  END
```

Gambar 25
Tampilan FORTRAN untuk sub routine ORDARV

Gambar 25 tampilan FORTRAN untuk sub-routine ORDARV, model persediaan Kejadian sub-routine Demand/ Permintaan yang dipanggil ketika ada permintaan untuk satu item atau

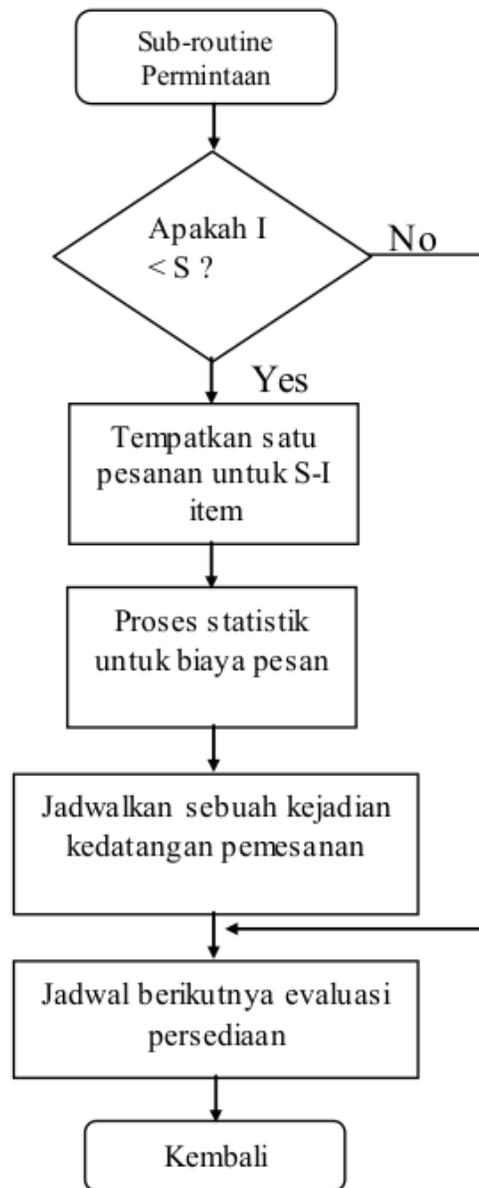
lebih dari perusahaan produk/ manufaktur. Flowchart dan tampilan pada gambar 26 dan 27 sub-routine PERBARUHI dipanggil untuk memperbarui area pada $I^+(t)$ dan $I(t)$ karena level persediaan akan berubah pada waktu kejadian permintaan terjadi ukuran permintaan yang akan terjadi pada saat ini, $DSIZE$, dituliskan/ dibangkitkan dengan fungsi RANDI dan berkurang dari level persediaan sebelumnya. Perlu dicatat bahwa ini dapat menghasilkan level persediaan menjadi negatif artinya bahwa permintaan satu item atau lebih dapat diberikan dari persediaan yang ada dan harus *Backlogged* (ditolak), waktu permintaan berikutnya $TNE(2)$ ditentukan dengan menambah sebuah variabel random *eksponensial* dengan rata-rata $MDEMDT$ ke nilai sekarang waktu simulasi.



Gambar 26
Flow chart sub-routine PERMINTAAN, model persediaan

```
SUBROUTINE ORDARV
INTEGER
AMOUNT,BIGS,INITIL,INVLEV,NEVNTS,NEXT,NMNTHS,SM
ALLS
REAL,AMINUS,APLUS,N,INCRMC,MDEMDT,PI,SETU
PC,TIME,TLEVNT,TNE(4),1 TORDC
COMMON /MODEL/
MINUS,AMOUNT,APLUS,BIGS,N,INCR,INITIL,INVLEV,
1
MDEMDT,NEVNTS,NEXT,NMNTHS,PI,SETUPC,SMAL
LS,TIME,TLEVNT,TNE,TORDC
C
C ### UPDATE 'APLUS' AND 'AMINUS'
C
C     CALL UPDATE
C
C ### GENERATE THE DEMAND SIZE
C     DSIZE=RANDI (Z)
C
C ### DECREMENT 1 THE INVENTORY LEVEL BY THE
DEMAND SIZE.
C
C     INVLEV=INVLEV-DSIZE
C
C ### SCHEDULE THE NEXT DEMAND
C
C     TNE(2)=TIME+EXPON(MDEMDT)
C     RETURN
C     END
```

Gambar 27
Daftar FORTRAN untuk sub-routine PERMINTAAN
model persediaan



Gambar 28
Flow chart sub-routine EVALU8, model persediaan

Subrutin kejadian EVALU8, yang dipanggil pada awal bulan untuk mengevaluasi tingkat persediaan dan kemungkinan memesan, diurutkan dan diperbarui pada gambar 28 dan 29, Gambar 29 Tampilan FORTRAN untuk subrutin EVALU8, model persediaan

```
SUBROUTINE EVALU8
INTEGER
AMOUNT,BIGS,INITIL,INVLEV,NEVNTS,NEXT,NMNT
HS,SMALLS
REAL,AMINUS,APLUS,N,INCRMC,MDEMDT,PI,SETU
PC,TIME,TLEVNT,TNE(4),1 TORDC
COMMON /MODEL/
MINUS,AMOUNT,APLUS,BIGS,N,INCR,INITIL,INVLEV,
1
MDEMDT,NEVNTS,NEXT,NMNTHS,PI,SETUPC,SMAL
LS,TIME,TLEVNT,TNE,TORDC
C
C ### IF THE INVENTORY LEVEL IS LESS THAN 'SMALL.'
PLACE AN ORDER FOR
C ### 'BIGS'-'INLEV' ITEMS.
      IF (INLEV.GE.SMALL) GO TO 10
      AMOUNT=BIGS-INLEV
      TORDC=TORDC+SETUPC+(INCRMC*AMOUNT)
C
C ### SCHEDULE THE ARRIVAL OF THE ORDER.
C
      TNE(1)=TIME+UNIFRM(.5,1.)
C
C ### SCHEDULE THE NEXT INVENTORY EVALUATION.
C
      10 TNE(4)=TIME+1.
      RETURN
      END
```

Gambar 29
Tampilan FORTRAN untuk subrutin EVALU8,
model persediaan

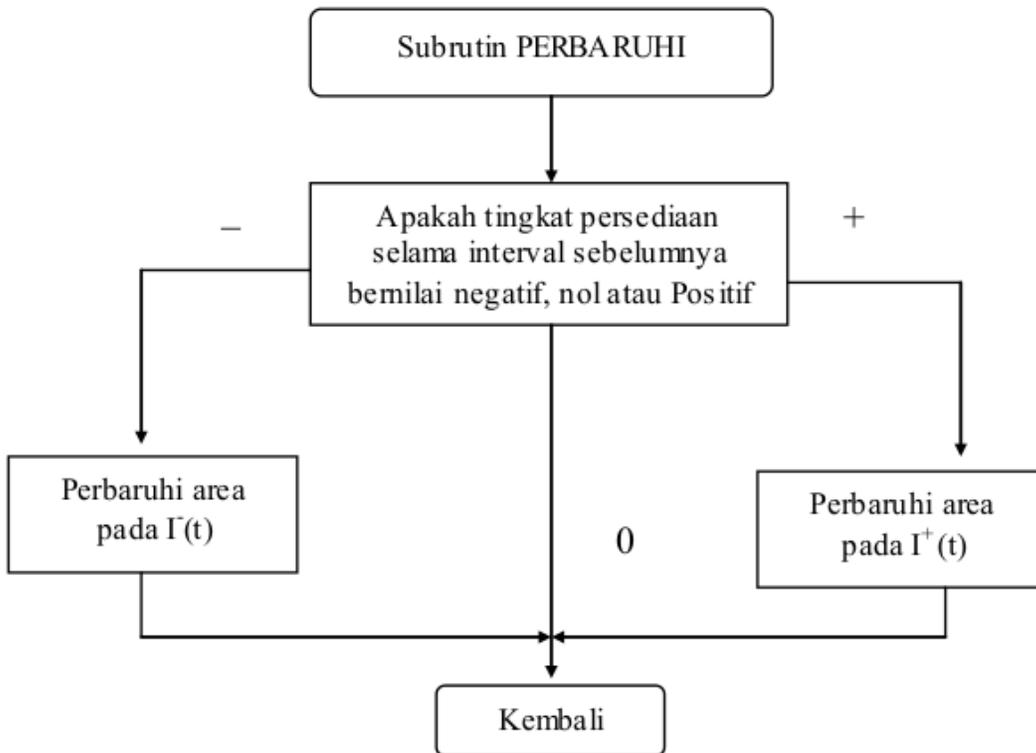
```
SUBROUTINE REPORT
INTEGER
AMOUNT,BIGS,INITIL,INVLEV,NEVNTS,NEXT,NMNTHS,SM
ALLS
REAL,AMINUS,APLUS,N,INCRMC,MDEMDT,PI,SETU
PC,TIME,TLEVNT,TNE(4),1 TORDC
REAL ACOST,AHLDC.AORDC,ASHRC
COMMON /MODEL/
MINUS,AMOUNT,APLUS,BIGS,N,INCR,INITIL,INVLEV,
1
MDEMDT,NEVNTS,NEXT,NMNTHS,PI,SETUPC,SMAL
LS,TIME,TLEVNT,TNE,TORDC
C
C ### UPDATE 'APLUS' AND 'AMINUS'
C
CALL UPDATE
C
C ### COMPUTE ESTIMATE OF THE DESIRED MEASURES
OF PERFORMANCE
C
AORDC=TORDC/NMNTHS
AHLDC=H*(APLUS/NMNTHS)
ASHRC=PI*(AMINUS/NMNTHS)
ACOST=AORDC+AHLDC+ASHRC
PRINT
10,SMALLS,BIGS,ACOST,AORDC,AHLDC,ASHRC
10 FORMAT (1H0,5X,('I3,',13,')',8X,F6.2,16X,F6.2,19X,F6.2,
19X,F6.21)
RETURN
END
```

Gambar 30
Tampilan FORTRAN untuk subrutin LAPORAN,
Model persediaan

Tampilan untuk subrutin LAPORAN diberikan pada gambar 30. Pertama Subrutin PERBARUHI dipanggil pertama untuk memperbarui daerah pada fungsi $I^+(t)$ dan $I^-(t)$ dari waktu kejadian terakhir yang merubah tingkat persediaan sampai waktu simulasi berakhir, WAKTU=NMNTHS. Kemudian perkiraan yang diinginkan untuk mengukur performance/ keberhasilan dihitung dengan tujuan yang jelas. Subrutin PERBARUHI yang bukan kejadian rutin dipanggil pada waktu tertentu/ khusus untuk memperbarui fungsi area pada $I^+(t)$ dan $I^-(t)$. Pemakaian PERBARUHI yang dipanggil dari ORDARV, PERMINTAAN dan LAPORAN, mungkin dari penulisan 3X kode blok yang sama.

Sebuah Flowchart dan tampilan untuk PERBARUHI diberikan pada gambar 31 dan 32. Jika tingkat persediaan sebelumnya bernilai negatif, area pada $I^-(t)$ dipengaruhi dan area pada $I^+(t)$ tidak diubah. Jika tingkat persediaan positif area pada $I^+(t)$ diubah dan area pada $I^-(t)$ tidak diubah. Penggunaan fungsi EKSPON disini identik dengan tampilan pada gambar 14. Sebuah tampilan untuk RAND I/ ACAK I diberikan pada gambar 33. Program bersifat umum yang akan membangkitkan sebagai bilangan bulat antara I dan NNILAI menurut fungsi distribusi PROB P, memberikan NNILAI dan PROBP (I) ($F=1,2,\dots, N \text{ NILAI}$) tertentu [pada kasus tertentu, nilai = 4 dan $\text{PROBP}(1) = \frac{1}{6}$, $\text{PROBP}(2) = \frac{1}{2}$, $\text{PROBP}(3) = \frac{5}{6}$ dan $\text{PROBP}(4) = 1$] jika $P(1) = \text{PROBP}(1)$ dan $P(1) = \text{PROBP}(1) - \text{PROBP}(I-1)$ untuk $I = 2,3,\dots, N \text{ NILAI}$. Dari tampilan dapat dilihat bahwa program akan dibangkitkan bilangan bulat, jika variabel random $U(0,1)$, U yaitu pada interval $[0, \text{PROBP}(1)]$, yang panjang-

nya = $P(1)$ dan akan membangkitkan I bilangan bulat 2 jika U yaitu pada interval [$PROBP(1)$, $PROBP(2)$ yang panjangnya adalah $P(2)$ dan seterusnya. Karena peluang variabel random pada $U(0,1)$ jatuh pada beberapa subinterval dari panjang ΔX (berisi interval $[0,1] = \Delta X$



Gambar 31

Flow chart untuk subrutin PERBARUHI, model persediaan

```
SUBROUTINE UPDATE
INTEGER AMOUNT, BIGS, INITIL, INVLEV, NEVNTS,
NEXT, NMNTHS, SMALLS
REAL AMINUS, APLUS, H, INCRMC, MDEMDT, PI,
SETUPC, TIME, TLEVNT, TNE(4)
1TORDC
REAL TSLE
COMMON / MODEL/ AMINUS, AMOUNT, APLUS,
BIGS, H, INITIL, INVLEV, IMDEMDT, NEVNTS, NEXT,
```

```
NMTHS, PI, SETUPC, SMALLS, TIME, TLEVNT, TNE,  
TORDC  
C ### COMPUTE THE TIME SINCE THE LAST EVENT WHICH  
CHANGED THE INVENTORY  
C ### LEVEL  
    TSLE=TIME-TLEVNT  
    TLEVNT=TIME  
C ### DETERMINE WHETHER THE INVENTORY LEVEL  
DURING THE PREVIOUS INTERVAL  
C ### WAS NEGATIF, ZERO OR POSITIVE.  
    IF(INVLEV) 10,20,30  
C  
C ### SINCE THE INVENTORY LEVEL, DURING THE  
PREVIOUS WAS  
C ### NEGATIF, UPDATE 'AMINUS.'  
  
    10 AMINUS=AMINUS+(-INVLEV*TSLE)  
  
C ### THE INVENTORY LEVEL DURING THE PREVIOUS  
INTERVAL WAS ZERO.  
    20 RETURN  
C  
C ### SINCE THE INVENTORY LEVEL DURING THE  
PREVIOUS INTERVAL WAS  
C ### POSITIVE, UPDATE 'APLUS.'  
  
    30 APLUS=APLUS+(INVLEV*TSLE)  
    RETURN  
    END
```

Gambar 32
Tampilan FORTRAN untuk subrutin PERBARUHI,
model persediaan

```
FUNCTION RANDI (Z)
INTEGER I, N1
REAL U
COMMON /RANDOM/ NVALUE, PROBD (25)
C
C ### GENERATE A U (0,1) RANDOM VARIABEL. THEFORM
      OF THIS STATEMENT
C ### DEPENDS ON THE COMPUTER USED.
      U=KANUN (Z)
C
C ### 1 GENERATE A RANDOM INTEGER BETWEEN 1 AND
      NVALUE IN ACCODANCE WITH
C ### DISTRIBUTION FUNCTION 'PROBD.'
C
      N1=NVALUE-1
      DO 10 I=1,N1
      IF (U.GE.PROBD (I)) GO TO 10
      RANDI=I
      RETURN
10 CONTINUE
      RANDI=NVALUE
      RETURN
      END
```

Gambar 33
Tampilan FORTRAN untuk fungsi RANDOM I,
model persediaan

```
1 FUNCTION UNIFORM (A,B)REAL A, B, U
C ### GENERATE A U (0,1) RANDOM VARIABEL. THE FORM
      OF THIS STATEMENT
C ### DEPENDS ON THE COMPUTER USED
      U=RANUN (Z)
C
```

```
C ### GENERATE A U (0,1) RANDOM VARIABLE.  
C  
    UNIFRM=A+(U*(B-A))  
    RETURN  
    END
```

Gambar 34
**Tampilan FORTRAN Untuk fungsi UNIFORM/ Seragam,
model persediaan**

Kita lihat bahwa program pembangkit bilangan bulat I1 dengan peluang $p(1) = \frac{1}{6}$ dan pembangkit bilangan bulat 2 dengan peluang $p(2) = \frac{1}{3}$, dan seterusnya. Distribusi ini diperlukan untuk variabel random p. seperti yang telah diberikan di awal, bagian 5. Sebuah tampilan untuk fungsi KESERAGAMAN diberikan pada gambar 34. Secara jelas bahwa sebuah pembangkit program variabel random kontinyu yang berdistribusi uniform pada interval [A, B].

Keluaran Simulasi dan Diskusi

Laporan simulasi untuk model persediaan diberikan pada gambar 35. rata-rata biaya total perbulan, akan tampak bahwa (20, 80) adalah kebijakan yang optimal.

SISTEM PERSEDIAAN PRODUK-TUNGGAL

Tingkat persediaan awal	= 60 item
Jumlah permintaan	= 4
Fungsi distribusi jumlah	= 0,167 0,500
Permintaan	= 0,833 1,000
Lead time	= 0, 10 bulan
Panjang simulasi	= 120 bulan
K = 32,0 I = 3,0 H= 1,0 PI= 5,0	

SINGLE-PRODUCT INVENTORY SYSTEM

INITIAL INVENTORY LEVEL : 60 ITEMS
NUMBER OF DEMAND SIZES : 4
DISTRIBUTION FUNCTION OF DEMAND SIZES : 167.500.833 1.000
MEAN INTERDEMAND TIME : 10 MONTHS
LENGTH OF THE SIMULATION : 120 MONTHS
K=32.1 I= 3.0 H= 1.0 PI =5.0

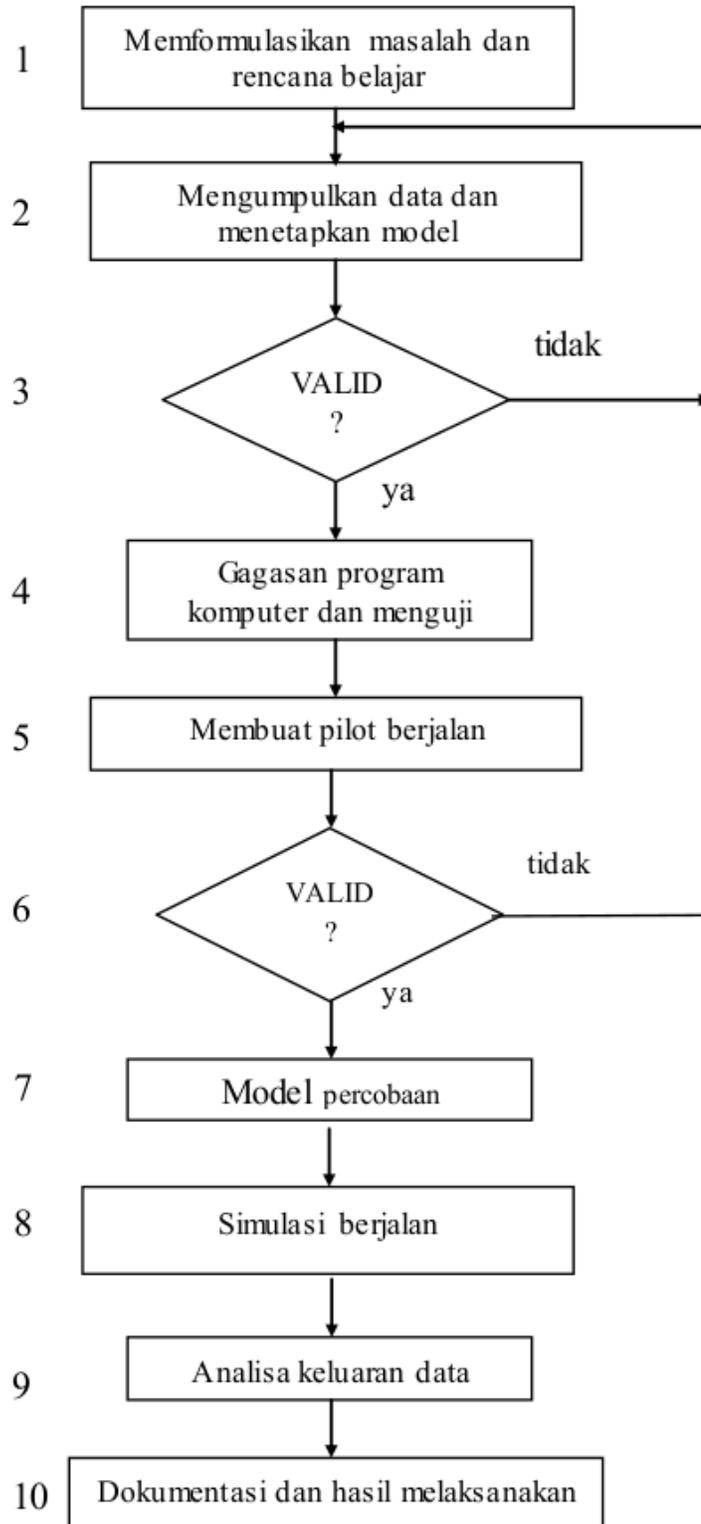
<i>Policy</i>	<i>Average cost</i>	<i>Average ordering cost</i>	<i>Average holding cost</i>	<i>Average shortage cost</i>
(20,40)	123.92	98.80	8.91	16.21
(20,60)	125.58	92.77	15.98	16.82
(20,80)	188.24	82.95	27.04	8.25
(20,100)	126.13	82.07	35.96	8.10
(40,60)	125.71	98.13	26.24	1.34
(40,80)	123.58	68.72	35.98	.87
(40,100)	134.35	87.68	45.15	1.52
(60,80)	145.69	101.20	44.29	.20
(60,100)	144.08	89.16	54.93	.00

Gambar 35
Laporan simulasi untuk model persediaan

Namun demikian, untuk konteks tersebut dimana panjang simulasi adalah tetap (perusahaan menginginkan horizon perencanaan 10 tahun), apa yang diinginkan untuk masing-masing kebijakan adalah nilai rata-rata perkiraan biaya total perbulan untuk bulan ke satu 120 bulan. Jumlah dalam gambar 35 diperkirakan dari nilai-nilai ekspektasi, masing-masing perkiraan berdasarkan pada sebuah ukuran sample (hasil simulasi atau pengulangan). Perkiraan ini mungkin mempunyai varian yang besar, perkiraan pemesanan mungkin berbeda dengan

pemesanan yang telah dipertimbangkan, yang formulasinya diperlukan kenyataannya, jika kita me-*run*-kan kembali simulasi menggunakan perbedaan $u(0,1)$ variabel random hasil estimasi mungkin berbeda besar sekali dari gambar 35. Lebih jauh lagi, selanjutnya perkiraan pemesanan baru mungkin juga menjadi berbeda.

Kita simpulkan dari diskusi diatas bahwa ketika kecepatan simulasi berjalan tetap sesuai kondisi masalah, pada umumnya ini tidak cukup untuk membuat *run* simulasi tunggal dari masing-masing kebijakan atau system yang dimaksud/ dibahas. Ada sepuluh (10) langkah- langkah dalam mempelajari simulasi kejadian diskrit. Gambar 36 menunjukkan langkah-langkah yang akan menyusun suatu kekhususan mempelajari simulasi, dan hubungan antara mereka (lihat juga shanom [16,p,23] dan Gordon [4,p,52]. Disamping menghadirkan jumlah simbol masing-masing langkah yang diinginkan untuk diskusi lebih rinci dari langkah yang tertera. Tidak semua studi penting memuat semua langkah-langkah ini dan sesuai pernyataannya. Beberapa studi mungkin berisi langkah-langkah yang tidak menggambarkan dalam diagram. Lebih jauh lagi, mempelajari sebuah simulasi tidak secara langsung urutan prosesnya. Satu laporan dengan satu studi dan memiliki pengertian/ pemahaman yang lebih baik tentang system yang diselidiki, ini sering diperlukan atau diinginkan mengulangi langkah-langkah sebelumnya sabagai untuk gambaran baru tentang system selama mempelajari mungkin penting. Merumuskan kembali masalah-masalah yang akan diselesaikan.



Gambar 36
Langkah-langkah dalam mempelajari suatu simulasi

Bagian X

MERUMUSKAN MASALAH DAN RENCANA MEMPELAJARINYA

Setiap studi harus mulai dengan sebuah pernyataan yang jelas dari objek-objek yang dipelajari. Tanpa suatu pernyataan maka harapan untuk berhasil kecil. Alternatif-alternatif sistem yang akan dipelajari harus digambarkan (jika memungkinkan) dan kriteria-kriteria untuk mengevaluasi kemanjuran alternatif-alternatif ini akan diberikan. Semua studi harus direncanakan peristilahannya, jumlah orang, biaya dan waktu yang diperlukan untuk masing-masing aspek studi.

Mengumpulkan data dan mendefinisikan sebuah model

Data harus dikumpulkan dalam sistem yang diamati dan digunakan untuk memperkirakan input parameter-parameter dan untuk membuat distribusi peluang variabel random yang cocok. Sebagai contoh pada model sebuah bank mungkin mencatat waktu antar kedatangan dan pelayanan dan menggunakan data-data ini untuk memenuhi secara teoritis distribusi

waktu antar kedatangan dan waktu pelayanan yang akan digunakan bila perlu, data pada performansi sistem, misalnya waktu tunggu dalam antrian akan dicatat untuk tujuan validasi. Susunan matematika dan logika model dari dunia sistem nyata untuk suatu tujuan adalah masih lebih bersifat seni dari pada iptek (pengetahuan).

Namun demikian ada beberapa rumus perusahaan yang akan menggunakan proses pemodelan yang menunjukkan kepada penulis suatu ide yang bagus untuk memulai dengan model yang sederhana yang kemudian dapat menjadi lebih menyenangkan. Sebuah model harus berisi rincian untuk menerangkan inti sari sistem untuk memahami model yang diharapkan. Tidak perlu memiliki satu per satu hubungan antar sub antara elemen-elemen model sistem. Sebuah model yang mempunyai banyak rincian mungkin akan menimbulkan biaya yang mahal untuk pembuatan program dan hasilnya sebuah diskusi yang baik dari seni memodelkan dapat dipelajari pada bagian tersendiri.

VALID?

Walaupun kita percaya bahwa kebenaran adalah sesuatu yang seharusnya dapat dilakukan seluruh simulasi dari pada setelah model dibuat dan hanya jika ada sisa waktu dan uang. Ada beberapa hal dalam mempelajari validasi dimana validasi adalah keutamaan salah satunya adalah langkah-langkah dalam pembuatan model, penting sekali untuk pemodel -pemodel memasukkan orang-orang yang akrab/ familiar dengan sistem yang sedang dipelajari. Ini juga nasehat untuk pemodel-pemo-

del untuk interaktif dengan pembuat keputusan (calon pengguna model) pada umumnya validasi modul ini akan bertambah aktual dan merasa validasi model untuk pengambil keputusan akan ditambah. Pada edisi ini, distribusi probabilitas secara teoritis sudah cukup mantap untuk meneliti data yang akan dites menggunakan *tes goodness of fit test*.

Membuat sebuah program komputer dan pengujiannya.

Pemodel-pemodel simulasi harus memutuskan apakah menggunakan sebuah model program dengan bahasa umum seperti FORTRAN atau menggunakan bahasa tertentu GAPS, GPSS atau SIMCRIPT dan yang lainnya. Sebuah bahasa umum akan mudah digunakan dan tersedia pada komputer pemodel-pemodel. Ini juga akan mempersingkat waktu eksekusi dengan kata lain dengan memberikan beberapa gambaran-gambaran yang diperlukan dalam sebuah model program, proses pemrograman dapat dipersingkat. Mendiskusikan teknik-teknik untuk membangkitkan variabel random pada sebuah komputer dengan distribusi peluang tertentu. Mendiskusikan hubungan pembangkit variabel random $u(0,1)$ (bilangan random) yang pada dasarnya membangkitkan semua tipe variabel-variabel random lainnya. Mendiskusikan teknik-teknik untuk verifikasi (pengujian) atau sebuah program komputer yang error.

Valid ? Jalannya program dapat digunakan untuk mengetes sensitifitasnya.

Perubahan kecil dalam output parameter jika output berubah besar, maka perkiraan input parameter akan diperoleh

lebih baik. Jika sistem hampir sama dengan salah satu kebenaran dan terakhir yang penting output data dan jalannya sebuah modul dari sistem yang ada dapat dibandingkan dengan output data dan sistem aktual yang ada (kumpulan langkah-langkah) jika pernyataan modul, "baik", "valid" dirubah maka system yang penting dihadirkan kembali kita berharap bahwa perubahan ini tidak begitu mahal

Desain eksperimen

Hubungan perancangan sistem akan disimulasikan seperti dalam kasus kenyataan, ada beberapa alat yang dapat memberikan laporan untuk disimulasikan. Kadang-kadang keputusan yang komplit tidak dapat dibuat pada saat ini/ sesaat. Untuk masing-masing perancangan sistem akan disimulasikan. Keputusan telah dibuat seperti isu-isu pada awal kondisi untuk hasil-hasil simulasi. Panjangnya hasil simulasi dan banyaknya keterkaitan hasil simulasi (pengulangan) akan dibuat isu-isu ini. Ketika merancang dan membuat hasil produksi, kadang-kadang memungkinkan penggunaan teknik-teknik pengurangan varian untuk mendapatkan hasil statistik yang lebih tepat (varian-varian dari ramalan berkurang) atau tidak bertambah. Teknik-teknik ini didiskusikan pada bagian tersendiri dan perlu mempelajari kembali dasar-dasar peluang dan statistik.

Pembuatan hasil/output

Hasil dibuat untuk memberikan hasil data pada perancangan sistem yang diamati.

Analisa data output

Teknik-teknik statistik digunakan untuk menganalisa data output dari hasil produksi. Tujuan utama adalah untuk membangun sebuah interfal kepercayaan untuk suatu ukuran pada suatu perancangan sistem tertentu atau untuk memutuskan sistem simulasi yang mana yang relatif paling baik ukuran keberhasilannya.

Dokumen dan penerapan hasil

Kumpulan model-model simulasi sering digunakan untuk lebih dari satu aplikasi, sekarang penting untuk mendokumentasikan/ menyimpan asumsi-asumsi yang telah ada pada model-model yang terbaik dalam program komputer. Akhirnya terakhir pelajaran simulasi yang berhasil pernah diterapkan adalah sebuah kegagalan yang paling besar. Jadi sebuah pelajaran tidak perlu diselesaikan sampai lengkap apabila hasilnya tidak diterapkan.

Bagian XI

TIPE-TIPE SIMULASI LAINNYA

Penekanan pada buku ini untuk simulasi kejadian diskrit, beberapa tipe simulasi yang lainnya telah dipertimbangkan kepentingannya pada dunia nyata. Tujuan kita disini adalah untuk menerangkan tipe-tipe simulasi lainnya dengan simulasi kejadian diskrit khususnya kita akan mendiskusikan simulasi kejadian kontinyu dan simulasi *Montecarlo*.

Simulasi Kontinyu

Simulasi kontinyu berhubungan dengan pemodelan sistem seluruh waktu dimana perubahan kondisi variabel secara kontinyu berkenaan dengan waktu khususnya. Model simulasi kontinyu memberikan satu atau lebih persamaan-persamaan diferensial yang memberikan hubungan-hubungan untuk ukuran perubahan status variable. Berkenaan dengan waktu jika persamaan differensialnya yang umum dan sederhana, mereka dapat diselesaikan secara analitik untuk memberikan nilai-nilai fungsi waktu dari nilai-nilai status variable. Pada waktu nol

untuk sebagian besar model-model kontinyu tidak mungkin bisa diselesaikan secara analitik, namun dapat diselesaikan menggunakan teknik analisa numerik, misalnya integrasi *range*, digunakan untuk integral persamaan differensial secara numerik, memberi nilai-nilai tertentu untuk status variabel pada waktu nol.

Sejumlah bahasa telah dirancang khusus untuk membuat model-model simulasi kontinyu. Misalnya CSMP III yaitu suatu bahasa yang berorientasi pada persamaan, dikembangkan oleh IBM dan Green [8]. Pada edisi bahasa simulasi kejadian diskrit SLAM (lihat Pritsker dan pegden [13] dan C-SIMSCRIPT (lihat Delfosse [3]) juga mempunyai kapasitas untuk pemodelan kontinyu. Disini ada tiga bahasa yang telah berhasil membahas kejadian diskrit dan kontinyu secara simultan pada satu model (kejadian gabungan).

Contoh 3, sekarang kita mempertimbangkan sebuah model kontinyu dari persaingan antara 2 populasi. Model-model dari biologi tipe-tipe ini yang disebut pencari mangsa atau serombongan parasit dipertimbangkan oleh beberapa penulis, termasuk Braun (2,p,583) dan Gordon (4,p.103). Sebuah lingkungan terdiri dari 2 populasi pencari mangsa yang berinteraksi satu dengan yang lainnya. Yang jadi mangsa adalah *passive*, sedangkan pencari mangsa yang kena mangsa sebagai sumber masukan mereka.

Sebagai contoh yang menjadi pemangsa ikan hiu dan yang jadi mangsa mungkin ikan-ikan kecil, lihat [2] jika $X(t)$ dan $Y(t)$, maka secara langsung jumlah individu-individu dalam pemangsa sehingga ada yang memberi mungkin secukupnya

kepada yang jadi mangsa yang lolos dari pemangsa, yang pertambahannya adalah $r x(t)$ untuk yang positif (kita dapat mengira r sehingga ukuran kelahiran yang alami tanpa ukuran kematian yang alami. Karena interaksi antara pemangsa dan yang dimangsa, ini beralasan untuk diasumsikan bahwa ukuran kematian yang dimangsa berhubungan secara proporsional dengan ukuran populasi ke 2 yaitu populasi $X(t)y(t)$, oleh karena itu semua ukuran populasi yang jadi mangsa berubah, dx/dt diberikan oleh

$$\frac{dx}{dt} = rx(t) - ax(t)y(t) \quad (3.1)$$

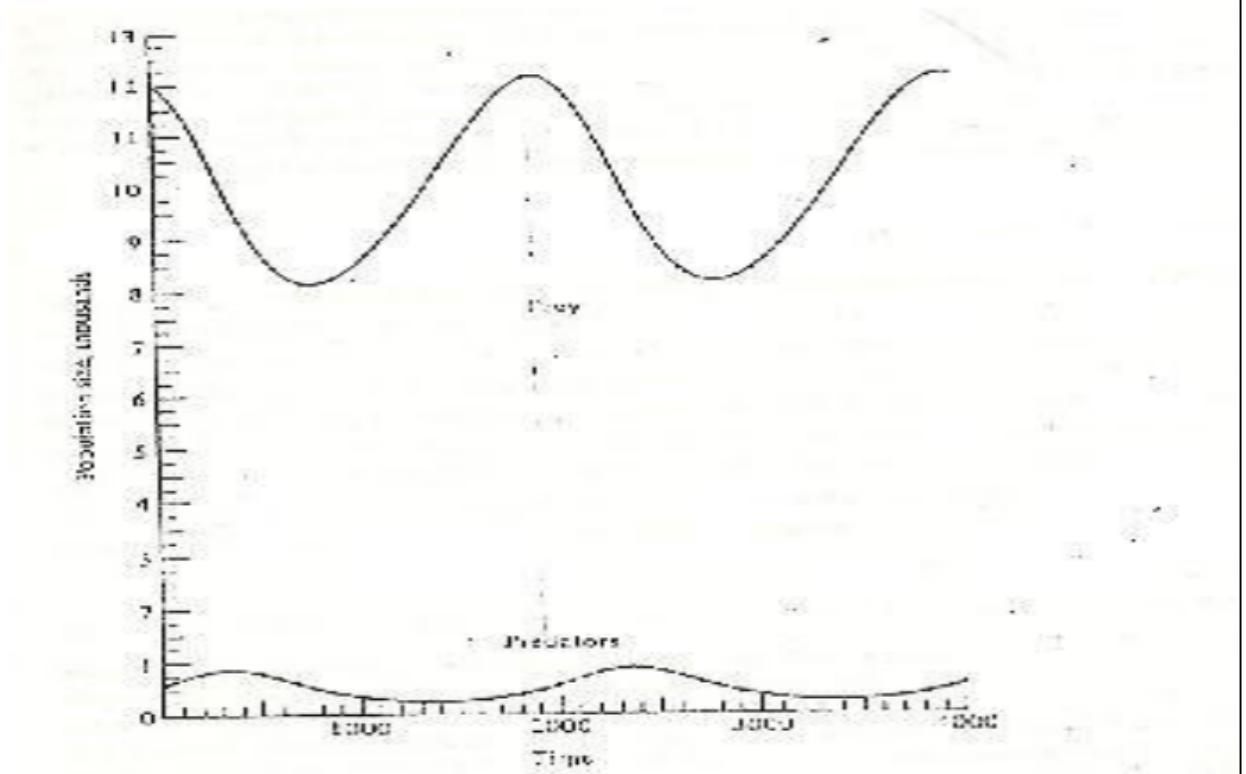
dimana a adalah sebuah ukuran konstanta positif kehadiran target pemangsa pada mangsanya. Perubahan ukuran pemangsa pada ketidak beradaannya mangsa adalah $-Sy(+)$ untuk beberapa S yang positif. Lebih lanjut lagi, hubungan antara kedua populasi ini menyebabkan populasi pemangsanya akan bertambah pada sebuah ukuran yang juga sebanding dengan $X(t)Y(t)$. Sehingga semua perubahan ukuran populasi pemangsa dy/dt adalah

$$\frac{dy}{dt} = -Sy(+) + bx(t)y(t) \quad (3.2)$$

dimana b adalah sebuah konstanta positif. Awal kondisi $X(0) > 0$ dan $Y(0) > 0$, penyesuaian model diberikan pada pers. (3.1) dan (3.2) yang mempunyai kepentingan $X(t) > 0$ dan $Y(t) > 0$ untuk semua $t \geq 0$ (lihat[1]). Jadi populasi mungkin tidak akan pernah dimatikan secara sempurna oleh pemangsa. Penyelesaian $(X(t), Y(t))$ adalah sebuah fungsi waktu dan periodik sehingga

positip yaitu $X(t + nT) = X(t)$ dan $Y(t + bT) = Y(t)$ untuk semua $n = 1, 2, \dots$ nilai ini tidak bisa diperkirakan.

Jika populasi pemangsa bertambah, populasi mangsa berkurang. Penyebab berkurangnya ukuran pemangsa, secara langsung akan menghasilkan berkurangnya jumlah pemangsa ini menyebabkan jumlah mangsa akan bertambah dan seterusnya. Jika nilai $r = 1,001$, $a = 2 \times 10^{-6}$, $S = 0,01$, $b = 10^{-6}$ dan awal ukuran populasi $X(0) = 12,000$ dan $Y(0) = 600$ gambar 37 adalah ukuran penyelesaian persamaan (3.1) dan (3.2) hasil dari pengukuran perancangan paket komputer untuk menyelesaikan persamaan-persamaan bilangan defferensial secara numerik (secara langsung sebuah bahasa simulasi kontinyu)



Gambar 37
Penyelesaian secara numerik
sebuah model pemangsa-mangsa

Simulasi Kombinasi Diskrit-Kontinyu

Kita definisikan simulasi *Montecarlo* akan menjadi sebuah skema kerja bilangan random, dimana variabel random, $U(0,1)$, yang digunakan untuk menyelesaikan masalah-masalah deterministik atau stokastik di mana penjelasan waktu permainan tidak memiliki rumus yang nyata.

Jadi, simulasi *Montecarlo* umumnya adalah statis dari pada npv semua penulis mendefinisikan simulasi *Montecarlo* akan menjadi beberapa penyelesaian simulasi yang menggunakan bilangan random, definisi kita lebih terbatas. Nama simulasi "*Monte Carlo*" atau metode asli selama perang dunia II digunakan untuk menyelesaikan masalah yang ada hubungan dengan bom atom. Untuk lebih rinci lagi diskusi simulasi *Montecarlo*, lihat Halton [6].

Seandainya kita akan mengevaluasi integral $I = \int_a^b g(x)dx$, dimana nilai fungsi nol yang secara analisa tidak dapat diintegrasikan (dalam prakteknya tidak akan mungkin menggunakan simulasi *Montecarlo* untuk mengevaluasi integral tunggal karena lebih efisien teknik analisa numerik untuk tujuan ini). Hal ini lebih suka untuk digunakan pada problem integral berganda dengan sebuah integral yang tidak berjalan. Untuk melihat bagaimana problem deterministik dapat dideteksi dengan simulasi *Montecarlo*, misal untuk variabel random $(b-a)g(x)$, dimana x adalah variabel random kontinyu dengan distribusi *uniform* pada interval $[a,b]$ seperti sebuah variabel random untuk selanjutnya disebut dengan $U(a,b)$ kemudian diketahui bahwa nilai harapan/ ekspektasi Y diberikan dengan :

$$\begin{aligned}
 E(Y) &= E[(b-a)g(X)] \\
 &= (b-a)E[g(X)] \\
 &= (b-a) \int_a^b g(x) f(x) dx \\
 &= (b-a) \frac{\int_a^b g(x) dx}{b-a} \\
 &= I
 \end{aligned}$$

Dimana $f(x) = \frac{1}{(b-a)}$ adalah peluang fungsi *density* variabel random dari a U $(b-a)$. Untuk menguji ke 3 persamaan diatas lihat sebagai contoh , Ross [14.p.35]. Jadi masalah evaluasi dari integral telah dikurangi dengan salah satu estimasi nilai ekspektasi $E(y) = I$ dengan rata-rata sampel

$$Y(n) = \frac{\sum_{i=1}^n Y_i}{n} = (b-a) \frac{\sum_{i=1}^n g(X_i)}{n}$$

Dimana X_1, X_2, \dots, X_N adalah variabel random IID (a,b) [ini gambar untuk memikirkan $Y(n)$ sebagai sebuah perkiraan dari daerah bujur sangkar yang mempunyai panjang $b-a$ dan tinggi $= (b-a) \int_a^b g(x) dx / (b-a)$. yang mempunyai nilai rata-rata $g(X)$ interval lebih dari (a,b)]. Lebih lanjut lagi dapat dilihat bahwa $E(Y(n))$ estimator yang tidak bias dari I dan variabel $[Y(n) = \text{variabel } (Y)/n$ (jika varibel bilangan tetap sekarang $Y(n)$ akan berubah-ubah mendekati I untuk n yang cukup besar untuk menggambarkan skema perhitungan. seandainya kita akan mengevaluasi integral $\int_0^\pi \sin x dx = z$

N	10	20	40	80	160
Y(n)	2.213	1.951	1.948	1.989	1.993

Simulasi *Montecarlo* banyak digunakan untuk menyelesaikan masalah-masalah yang pasti dari statistik yang tidak dapat diselesaikan secara analitik, sebagai contoh untuk meramalkan nilai-nilai kritis untuk tes hipotesa baru. Penentuan nilai-nilai kritis untuk tes kenormalan data Kologorov-Smirnov.

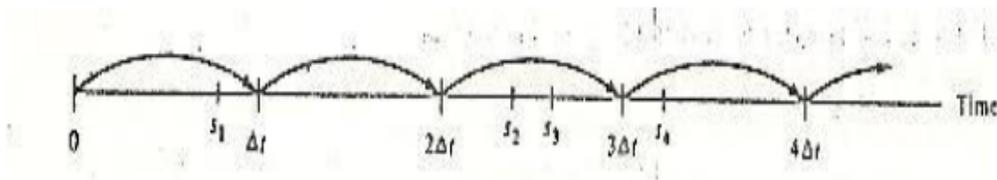
Bagian XII

LAMPIRAN-LAMPIRAN

LAMPIRAN A

Menentukan Tambahan Waktu Berikutnya

Sebagaimana telah di terangkan pada bagian 3. Prinsip kedua untuk menentukan waktu simulasi terdahulu/ sebelumnya pada model simulasi kejadian diskrit disebut "fixed-increment time advance. Dengan pendekatan ini, waktu simulasi dilanjutkan pada penambahan waktu (Δt) secara tepat untuk beberapa pilihan yang cocok dari Δt . Setelah masing-masing mengubah jam, suatu kontrol dibuat untuk menentukan apakah beberapa kejadian harus terjadi sepanjang interval Δt . Jika satu kejadian atau lebih dijadwalkan untuk terjadi sepanjang interval ini. Kejadian-kejadian ini dipertimbangkan untuk terjadi sampai akhir interval dari kondisi sistem perhitungan perhitungan statistik yang sesuai diperbaharui. Pendekatan penentuan tambahan waktu digambarkan pada gambar 38



Gambar 38.

Sebuah gambaran dari penentuan waktu tambahan

Kurva arah panah menggambarkan penambahan waktu simulasi dan S_i ($i=1, 2, \dots$, adalah waktu yang sebenarnya kejadian terjadi I tahun dari beberapa tipe kejadian (bukan waktu simulasi pada nilai I tahun). Pada interval waktu $(0, \Delta t)$ sebuah kejadian terjadi pada waktu S tetap dipertimbangkan akan terjadi Δt untuk model untuk menetapkan bahwa ini kasus. Kejadian terjadi pada waktu S_2 dan S_3 pada interval $(2\Delta t, 3\Delta t)$ tetapi kedua kejadian ini dipertimbangkan untuk terjadi pada waktu $3\Delta t$ dan seterusnya.

Sekumpulan rumus dapat dibuat kedalam model untuk memutuskan pada saat proses kejadian-kejadian apa waktu 2 atau 3. Kejadian dipertimbangkan mungkin untuk terjadi pada waktu yang sama dengan model dan kerugian dari penentuan waktu tambahan yaitu kesalahan memperkenalkan kejadian. Proses pada akhir interval di waktu mereka terjadi dan pentingnya memutuskan kejadian yang akan diproses pertama kali apabila kejadian -kejadian yang tidak serempak. Apabila seperti kenyataannya dilakukan oleh model. Masalah-masalah ini dapat dikurangi dengan membuat Δt lebih kecil, tapi akan menambah jumlah pengetikan kejadian yang harus terjadi dan akan menambah waktu eksekusi/proses karena pertimbangan ini, penentuan tambahan waktu umumnya tidak digunakan

untuk simulasi kejadian diskrit, karena waktu antara kejadian yang berturut-turut dapat bervariasi besar sekali.

Penggunaan utama pendekatan ini adalah untuk sistem dimana kejadian simulasi terjadi pada satu waktu $n \Delta t$ ($n=0, 1, 2, \dots$) untuk pilihan Δt yang sesuai untuk data pada sistem ekonomi sering tersedia pada tahun awal dan umumnya simulasi akan melanjutkan waktu simulasi untuk 1 tahun, lihat Naylor (10) untuk diskusi simulasi ekonomi. Untuk diskusi sistem persediaan yang dapat disimulasikan, tanpa kehilangan ketepatan dengan penentuan waktu tambahan.

Perlu dicatat bahwa penentuan waktu tambahan dapat direalisasikan ketika menggunakan pendekatan kejadian berikutnya dengan penjadwalan tiruan "kejadian-kejadian" yang akan terjadi setiap waktu Δt unit.

LAMPIRAN B

Keutamaan pada Sistem Antrian

Sebuah sistem antrian adalah sebuah sistem yang mempunyai satu pelayanan atau lebih yang memberikan berbagai macam pelayanan kepada pelanggan yang datang. Konsumen yang datang akan mendapatkan semua pelayanan sibuk (umumnya/ bergabung satu/ lebih antrian (atau berbaris) di depan pelayan-pelayaan, sehingga disebut sistem "antrian". Sejarahnya, sebagian besar pelayanan simulasi kejadian diskrit telah mengembangkan model-model sistem antrian dunia nyata atau sedikit dikitnya komponen sistem yang disimulasikan adalah sebuah sistem antrian.

Tabel 7 Contoh Sistem Antrian Dunia Nyata

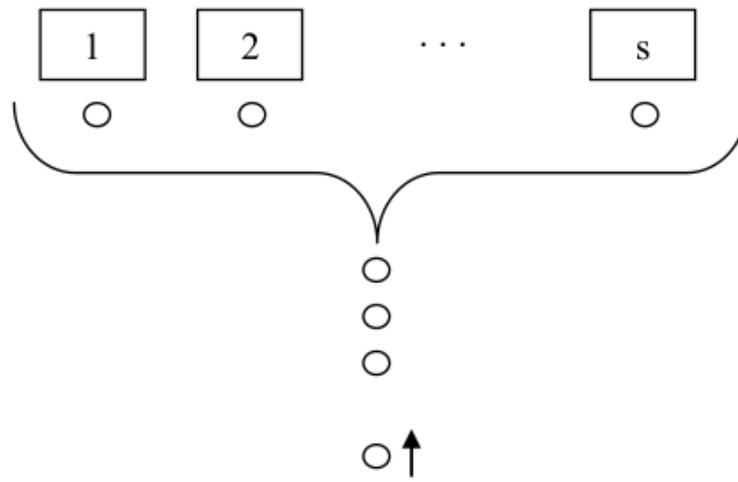
Sistem	Pelayanan	Pelanggan
Bank	Teller	Konsumen
Rumah sakit	Dokter, Perawat, Tempat tidur	Pasien
Komputer sistem	CPU, Input-output	Pekerjaan
perakitan	Pekerjaan-pekerjaan, mesin-mesin	Komponen pabrik
Airport	Landasan, Pemeriksaan Tempat, tempat cek in	Pesawat, perjalanan

Jadi kita percaya bahwa ini penting bagi mahasiswa yang belajar simulasi untuk mengerti penting tidak komponen? Sistem antrian, notasi standart sistem antrian dan ukuran – ukuran keberhasilan yang sering digunakan untuk menunjukkan kualitas pelayanan yang diberikan oleh sebuah sistem antrian. Beberapa untuk sistem antrian dunia nyata yang telah disimulasikan diberikan pada tabel 7 untuk tambahan informasi pada sistem antrian, pada umumnya lihat Gross dan Harris [5].

Komponen Sistem

Komponen–komponen sistem antrian sebuah sistem antrian dicirikan dengan 3 komponen. Proses kedatangan, mekanisme pelayanan dan disiplin antrian. Khususan proses kedatangan untuk sebuah sistem antrian terdiri dari gambaran bagaimana kedatangan konsumen ke sistem. Jika A , mempunyai interval waktu antara kedatangan $(i-1)$ dan I th pelanggan-pelanggan (lihat bagian 3). jika A_1, A_2, \dots diasumsikan dengan IID

variabel-variabel random, kita akan mencatat rata-rata (ekspektasi) interval waktu dengan $E(A)$ dan sebut $\lambda = \frac{1}{E(A)}$ ukuran kedatangan pelanggan-pelanggan



Gambar 39
Antrian A GI/G/s

Mekanisme pelayanan untuk sistem antrian dihubungkan dengan kekhususan jumlah pelayan-pelayan (disimbolkan dengan s) apakah masing-masing pelayan memiliki antrian-antriannya atau ada satu antrian dilayani semua pelayan dan memiliki distribusi peluang waktu pelayanan.

Jika s adalah waktu pelayanan dari 1 tahun kedatangan konsumen. sedangkan S_1, S_2, S_3, \dots adalah variabel random IID, kita akan mencatat rata-rata kedatangan konsumen dengan $E(S)$ dan sebut $w = \frac{1}{E(S)}$ ukuran pelayanan disiplin antrian dari sebuah sistem antrian menyediakan penggunaan rumus dimana seorang pelayan akan memilih konsumen berikutnya dari

antrian (jika ada beberapa) ketika pelayan menyelesaikan layanannya dari pelanggan yang terakhir.

Umumnya disiplin antrian menggunakan:

FIFO = First in – first out, pelanggan yang datang dilayani duluan

LIFO = Last in – First out, Yang datang terakhir dilayani duluan

Priority = Konsumen dilayani tergantung kepentingan atau kebutuhan

Simbol -Simbol Untuk Sistem Antrian

Sistem antrian sering terjadi dalam kenyataannya sehingga simbol -simbol standar diberikan khususnya pertimbangan dalam sistem antrian pada gambar 39 yang mempunyai sifat-sifat sebagai berikut :

1. S pelayan-pelayan dalam kondisi paralel dan satu antrian FIFO untuk semua pelayan.
2. A_1, A_2, \dots adalah variabel random IID
3. S_1, S_2, \dots adalah variabel random IID
4. A_i 's dan S_j 's independent

Kita sebut sistem antrian GI/G/s dimana GI (general Independent, mempunyai distribusi S_i 's dan G (general) mempunyai distribusi S_i 's. jika distribusi tertentu diberikan untuk A_i 's dan S_i 's (kasus yang sering terjadi pada simulasi , pencatatan simbol -simbol distribusi ini digunakan pada GI da G. Simbol M digunakan untuk distribusi eksponensial menurut *Markovian*, misalnya kehilangan memori, sifat-sifat distribusi eksponensial, simbol E_k untuk sebuah k distribusi peluang (jika X adalah

sebuah k-Erlang variabel random, kemudian $X = \sum_{i=1}^k Y_i$

dimana Y_i 's adalah variabel random eksponensial IID jadi sebuah sistem antrian pelayanan tunggal dengan waktu antrian dan pelayanan eksponensial dan disiplin antrian FIFO disebut sebuah antrian M/M/I

Untuk beberapa antrian GI/G/s, kita sebut kualitas $\rho = \lambda / (sw)$ factor kegunaan. Sistem antrian (sw adalah ukuran pelayanan dari sistem ketika semua pelayan sibuk) ini adalah ukuran yang menunjukkan betapa sibuknya sistem.

Ukuran Keberhasilan Untuk Sistem Antrian

Ada banyak kemungkinan ukuran keberhasilan untuk sistem antrian. Sehingga kita menggambarkan ukuran yang sering di gunakan dalam mempelajari matematik sistem antrian. Pembaca seharusnya tidak mengambil kesimpulan dari pilihan kita bahwa ukuran ini adalah paling penting atau relevan dalam prakteknya. Pada kenyataannya, beberapa sistem dunia nyata ukuran-ukuran ini tidak bagus/ sebaik yang didefinisikan, misalnya tidak tersedia / tidak ada.

Jika D_i = Penundaaan konsumen ke i

$W_i = D_i + S_i$ = Waktu menunggu konsumen ke i dan Sistem

$Q(t)$ = jumlah konsumen dan antrian pada waktu t

$L(t)$ = jumlah konsumen dalam sistem pada waktu t [$Q(t)$ + jumlah konsumen yang sedikit dilayani pada waktu t]

Kemudian ukurannya

$$d = \lim_{n \rightarrow \infty} \frac{\sum_{i=1}^n D_i}{n} \quad \text{w.p.I}$$

$$\text{dan } w = \lim_{n \rightarrow \infty} \frac{\sum_{i=1}^n W_i}{n} \quad \text{w.p.I}$$

(jika ada) disebut status tetap –rata-rata waktu dalam antrian dan status tetap rata-rata waktu dalam sistem.

Ini dan untuk semua buku ini, maksud “D.P.I” (Dengan peluang I) diberikan untuk membenarkan matematika dan dari prakteknya

kecil signifikannya. Sehingga untuk, jika $\sum_{i=1}^n D_i/n \rightarrow d$

sedangkan $n \rightarrow \infty$ (D.P.I) untuk beberapa sistem antrian disini artinya bahwa jika hasil eksperimen sangat besar (tidak terhingga),

padahal eksperimen sebenarnya $\sum_{i=1}^n D_i/n$ berubah

menjadi terhingga. Catatan bahwa $p < I$ adalah kondisi yang penting untuk $d, w, Q,$ dan L untuk antrian keberadaan GI/G/s

umumnya sebagian besar dan telah berhasil untuk sistem antrian yaitu perusahaan yang tetap yaitu :

$$Q = \lambda d \text{ dan } L = \lambda w$$

Perusahaan ini dipegang/ dimiliki setiap persamaan teori antrian untuk keberadaan d dan w (lihat Stidham [18]). dalam prakteknya persamaan yang lain dipertimbangkan yaitu $W = dt E(s)$, akan diterangkan bahwa diskusi ukuran keberhasilan

diatas dapat dihitung secara analitik untuk antrian $M/M/s$ ($s \geq I$), antrian $M/G/I$ untuk distribusi G dan untuk sistem antrian yang pasti lainnya. Pada umumnya ,distribusi antar kedatangan , distribusi pelayanan atau keduanya harus eksponensial (atau sebuah variabel yang eksponensial seperti k-Erlang) untuk penyelesaian analitik yang memungkinkan (lihat [5] atau [9,10])

Simulasi Industri

ORIGINALITY REPORT

5%

SIMILARITY INDEX

5%

INTERNET SOURCES

0%

PUBLICATIONS

0%

STUDENT PAPERS

PRIMARY SOURCES

1

fac.ksu.edu.sa

Internet Source

5%

Exclude quotes On

Exclude matches < 3%

Exclude bibliography On