

BAB II

LANDASAN TEORI

2.1 Pengertian Simulasi.

Dalam kamus definisi simulasi adalah mengasumsikan, menyerupakan atau mencirikan sesuatu. Dari pengertian tersebut jelas belum memberikan gambaran yang lengkap dalam ilmu komputer. Berawal dari pengertian tersebut, banyak orang yang memberikan definisi lengkap tentang simulasi. Definisi yang cukup tepat tentang pengertian simulasi adalah sebagai berikut :

“Proses desain matematika atau model logika dari sistem riil dan selanjutnya dengan perantaraan dasar-dasar komputer dan eksperimen-eksperimen model dilakukan untuk menggambarkan, menjelaskan dan meramalkan perilaku sistem yang sesungguhnya”.

Dari definisi tersebut nampak bahwa dalam simulasi diperlukan model yang jelas dan bersifat matematik atau logika. Kemampuan untuk merancang model yang bagus adalah landasan suksesnya analisa simulasi. Keahlian lain yang mendukung analisa simulasi, diperlukan untuk dapat menggambarkan model dalam bentuk yang dapat dipakai, juga menghasilkan desain sistem yang relevan dengan masalah yang sedang dipecahkan dan cara-cara pengoperasiannya. Program komputer menjadikan model kedalam bentuk yang dapat dipakai. Sehingga usulan pokok dari analisa simulasi adalah memprediksi kelakuan sistem.

2.2 Klasifikasi Model Simulasi.

Untuk menghasilkan model yang baik kita harus memahami tipe model. Klasifikasi model-model simulasi antara lain model preskriptif atau deskriptif, model diskret atau kontinyu, model probabilistik atau deterministik, model statik atau dinamik.

2.2.1 Deskriptif atau preskriptif.

Model termasuk tipe model preskriptif jika digunakan untuk merumuskan dan mengoptimalkan penyelesaian dari masalah yang diberikan. Dengan demikian akan memberikan penyelesaian terbaik. Sedangkan tipe model deskriptif adalah jika model hanya menggambarkan perilaku sistem tanpa mempertimbangkan optimalisasi total proses dalam penanganan analisis.

2.2.2 Diskrit atau kontinyu.

Klasifikasi diskret atau kontinyu ditunjukkan pada variabel modelnya. Variabel kontinyu harus bernilai bilangan riil, sedangkan variabel diskrit merupakan sebuah pendekatan, nilainya berupa bilangan yang tertentu. Contoh untuk membedakan model diskrit atau kontinyu adalah variabel waktu dalam antrian, orang yang datang mulai dilayani dalam suatu titik waktu diskret sehingga model diskret lebih tepat.

2.2.3 Deterministik atau probabilistik.

Untuk membedakan deterministik atau probabilistik dilihat dari variabel modelnya. Jika digunakan variabel random harus didefinisikan dengan fungsi probabilitas yang tepat. Sedang jika variabel model yang digunakan merupakan variabel matematik yang nilainya bisa ditentukan dengan tepat dan

cukup untuk menggambarkan perilaku sistem, maka model ini digolongkan model deterministik

2.2.4 Statik atau dinamik.

Model digolongkan sebagai model statik atau dinamis tergantung apakah variabel model berubah setiap saat atau tidak. Contohnya dalam antrian termasuk model dinamik, karena variabel modelnya bisa berubah setiap saat. Antrian berisi kedatangan yang terbaru dan proses selalu dilakukan dengan lengkap yang meliputi saat masih lama dilayani dan lama pengantri berada dalam antrian. Semua model antrian biasanya termasuk dalam model dinamik.

2.3 Langkah Dasar Analisis Simulasi.

Dalam analisa simulasi harus dilakukan spesifikasi prosedur secara lengkap dalam menggunakan dan merancang model simulasi, sehingga diperoleh aplikasi yang berhasil. Langkah-langkah dalam analisa simulasi antara lain perumusan masalah, pengumpulan dan analisa data, pengembangan model, validasi dan verifikasi, percobaan dan optimasi model, dan yang terakhir adalah implementasi.

2.3.1 Perumusan Masalah .

Perumusan masalah adalah langkah terpenting dalam analisa simulasi. Tetapi sebelum masalah dapat dirumuskan harus diidentifikasi lebih dahulu. Masalah yang muncul adalah menentukan beberapa masalah yang sama pentingnya pada satu permasalahan, menentukan kriteria masalah apakah termasuk masalah teknik, ekonomi atau politik dan sebagainya. Pemilihan kriteria

masalah tergantung pada ketepatan analisa dan hasil implementasi. Pembahasan perumusan masalah meliputi beberapa hal, antara lain menentukan variabel-variabel dan batasan-batasannya, menentukan ukuran unjuk kerja sistem (*performance*) dan tugas-tugas obyektif, serta menyusun pendekatan model.

2.3.2 Pengumpulan dan Analisis Data.

Pengumpulan data biasanya dalam bentuk tabel. Sedang dalam simulasi data dikumpulkan dan selanjutnya untuk dibuat modelnya.

a. Pengumpulan data.

Metode pengumpulan data sangat bervariasi tergantung dari permasalahan yang akan dipecahkan. Ada beberapa hal yang harus dipertimbangkan antara lain: kemampuan untuk mengumpulkan data secara cepat dan tepat, pengaruh proses pengumpulan data pada perilaku sistem diukur, data hasil pengamatan harus dapat diubah untuk diproses dengan komputer, biaya pengamatan.

b. Analisis data.

Salah satu tujuan analisis data adalah menentukan model data. Apabila data yang dianalisa telah diketahui nilainya dengan tepat maka model data tersebut adalah deterministik. Sedang apabila data diperoleh dari bilangan random maka model data tersebut adalah probabilistik.

2.3.3. Pengembangan Model.

Pada langkah pengembangan model kita membuat diskripsi sistem menjadi model yang eksplisit dengan menilai hubungan antara semua variabel dan

ukuran-ukuran unjuk kerja sistem. Kemudian dibuat program komputer yang tepat sebagai implementasi model.

Ada dua teknik obyektif yang biasanya digunakan yang cukup praktis dan efektif, yaitu pendekatan diagram alir (*Physical Flow Approach*) dan pendekatan perubahan keadaan (*State Change Approach*).

Pada pendekatan diagram alir mengidentifikasi masukan-masukan yang akan diproses atau mentransformasikan fungsi pokok sistem ke dalam suatu diagram alir. Masukan-masukan disini adalah langkah-langkah yang ada dalam sistem. Sebuah diagram masukan dan bagian-bagian proses sistem harus cukup merepresentasikan model sistem yang selanjutnya dapat disusun ke dalam program komputer.

Sedangkan pada pendekatan perubahan keadaan, kita harus mendefinisikan tambahan variabel endogenous yaitu variabel yang berhubungan dengan output, variabel keadaan dan pengantar konsep model kejadian yang baru. Variabel keadaan mencakup status berbagai hal dalam sistem.

2.3.4. Validasi dan Verifikasi Model.

Verifikasi memfokuskan pada konsistensi internal model. Validasi memfokuskan hubungan antara model dan keadaan sesungguhnya.

2.3.5. Percobaan dan Optimasi Model.

Dengan mengevaluasi alternatif desain sistem yang mengacu pada output unjuk kerja model simulasi, kita dapat memilih yang terbaik sebagai desain sistem yang bagus. Kemudian untuk memberikan alternatif pembanding yang tepat maka keluaran model simulasi memerlukan analisis statistik.

a. Analisis Output.

Ada beberapa aspek dalam menginterpretasikan bentuk keluaran yang khusus dalam analisis simulasi. Pada simulasi dengan sampel data numerik, statistik inferensi diperlukan untuk menjelaskan output model simulasi. Masalah pokok pada estimasi sampel adalah bagaimana menentukan ukuran sampel yang cukup menggambarkan perilaku sistem dan ukuran besar kecilnya sampel sehingga cukup tepat untuk mengestimasi ukuran unjuk kerja.

b. Percobaan Model.

Masalah utama dalam percobaan model adalah mengumpulkan informasi-informasi perilaku sistem yang membantu dalam pengambilan keputusan. Dalam memahami unjuk kerja sistem harus diperkirakan bagaimana perilaku sistem yang benar-benar diharapkan dan harus diperhatikan juga satu atau lebih pilihan konfigurasi sistem. Kita bisa memfokuskan pada salah satu ukuran unjuk kerja sistem, atau mempertimbangkan beberapa hal yang saling berhubungan.

2.3.6. Implementasi.

Pada tahap implementasi kadang-kadang mengalami kegagalan. Disebabkan oleh kurangnya pemahaman tentang sistem, keterlambatan implementasi dan kemampuan pengguna yang kurang, atau manajer yang kurang memahami analisa simulasi. Yang dimaksud dengan sistem adalah suatu bentuk desain matematika atau model logika. Untuk mengatasi hal yang bisa menghambat tersebut terdapat pendekatan yang biasanya digunakan yaitu dengan

memahami teknik-teknik yang digunakan dalam pengembangan dan implementasi sistem informasi.

2.4 Langkah Penyusunan Model Simulasi.

Dalam penyusunan model simulasi ada beberapa langkah, antara lain: membuat diagram alir program dari modelnya, pemilihan bahasa pemrograman, bilangan acak, variasi acak dan statistika, pemrograman dan debugging.

2.4.1 Membuat diagram alir program komputer dari modelnya.

Langkah pertama dalam penulisan program komputer adalah membuat diagram alir. Jika pendekatan diagram alir digunakan untuk memahami sistem maka dalam hal ini dapat membantu dalam memperbaiki ulang seluruh diagram alir program.

2.4.2 Pemilihan bahasa pemrograman.

Peningkatan jumlah yang besar bahasa pemrograman yang muncul akhir-akhir ini mempermudah dalam pemilihan bahasa pemrograman yang digunakan untuk mengimplementasikan model simulasi. Bahasa simulasi yang terkenal diantaranya GPSS (*General Purpose Simulation Sistem*), SLAM (*Simulation Language for Alternatif Modeling*), SIMSCRIPT dan SIMULA untuk komputer IBM PC. Yang cukup memadai untuk analisa simulasi yaitu PASCAL, FORTRAN dan BASIC. Pemilihan bahasa khusus untuk analisa simulasi adalah efisien dalam pembuatan program dan debugging.

2.4.3 Bilangan acak, variasi acak dan statistika.

Dalam simulasi ini, data pengantri diperoleh dengan bilangan random. Tetapi bilangan random yang dihasilkan tersebut tidak perlu diperhatikan pola atau distribusinya (fungsi probabilitas), karena dalam hal ini data pengantri bisa diasumsikan hanya sebagai data contoh yang digunakan untuk menguji jalannya algoritma.

Hubungan dari bilangan-bilangan yang diberikan pada distribusi probabilistik merupakan bagian nilai variabel random. Pada contoh sistem antrian untuk menentukan waktu kedatangan juga dengan menggunakan bilangan random. Pada model simulasi yang menggunakan bilangan random dan fungsi probabilistik untuk menghasilkan bilangan random, statistik unjuk kerjanya bisa diperoleh diantaranya dengan menghubungkan fungsi probabilitas yang sama/mirip. Dalam simulasi pada sistem operasi untuk masalah *Dining Philosophers* ini, statistik output tolok ukur unjuk kerja yang dipakai adalah berdasar pengamatan cara kerja algoritma tersebut dan membandingkan hasil akhir (output) dari variasi input yang diberikan .

2.4.4 Pemrograman dan debugging.

Penyelesaian masalah akan lebih baik atau jelek tergantung dari pemilihan bahasa pemrogramannya. Penyusunan logika program yang tepat membantu dalam menjamin model. Bahasa simulasi menyediakan fasilitas *coding* program komputer dan memiliki penjelasan-penjelasan yang mempermudah *debugging* program, walau demikian penggunaan bahasa umum seperti bahasa

PASCAL bisa lebih mempermudah terutama pada simulasi model deskriptif yang memerlukan tampilan grafis.

2.5 Dasar-dasar Sistem Operasi.

Sebuah sistem operasi adalah sebuah program yang bekerja sebagai antar muka (*interface*) antara pengguna komputer dan perangkat keras komputer. Kegunaan sistem operasi adalah menyediakan sebuah lingkungan dimana pengguna bisa menjalankan program-programnya. Tujuan utama sistem operasi adalah menjadikan komputer mudah digunakan, sedang tujuan lainnya adalah agar perangkat keras komputer bisa digunakan secara efisien.

2.5.1 Pengertian Proses.

Konsep yang sangat penting dalam sistem operasi adalah pengertian proses-proses yang merupakan istilah abstraksi dari pengertian jalannya program. Semua komputer modern saat ini mampu mengerjakan beberapa hal pada saat bersamaan. Saat komputer sedang menjalankan program, saat itu juga komputer sambil membaca data di disk/mengirim data ke printer. Pada sistem pemrograman majemuk (*multiprogramming*), CPU juga mengunci dari satu program ke program lainnya, dan proses penguncian tersebut akan berlangsung dalam sepuluh atau ratusan milisekon. Sesungguhnya yang dilakukan oleh CPU pada satu saat hanya mengerjakan satu program, tetapi dalam selang waktu satu detik CPU secara bergiliran bisa mengerjakan beberapa program. Sehingga hal ini memberikan kesan seolah-olah proses berjalan secara paralel pada setiap program yang aktif. Proses penguncian antar proses yang sangat cepat oleh CPU tersebut disebut juga

pseudoparallelisme. Dalam hal ini terdapat perbedaan dengan pengertian *parallelisme hardware* yang sesungguhnya, dimana ada saat CPU sedang proses menghitung sementara ada satu atau lebih piranti I/O yang jalan. Menjaga langkah-langkah yang banyak secara paralel adalah sangat sulit untuk dikerjakan. Oleh karena itu sistem operasi selama bertahun-tahun telah dibuat dengan mempertimbangkan model *parallelisme* yang mudah.

2.5.2 Penjadwalan Proses.

Sedangkan urutan kerja yang dilakukan komputer dalam sistem operasi memerlukan penjadwalan yang bertugas untuk memutuskan proses berjalan kapan dan selama berapa lama proses itu berjalan. Kriteria untuk mengukur dan optimasi kinerja penjadwalan adalah:

a. Adil (*fairness*).

Adil adalah proses-proses dilakukan sama yaitu mendapat jatah waktu pemroses yang sama dan tidak ada yang tidak kebagian layanan pemroses sehingga mengalami *starvation*. Sasaran penjadwalan seharusnya menjamin tiap proses mendapat pelayanan dari pemroses yang adil.

b. Efisiensi .

Efisiensi dihitung dengan membandingkan (*rasio*) waktu sibuk pemroses. Sasaran penjadwalan adalah menjaga agar pemroses tetap dalam keadaan sibuk sehingga efisiensi mencapai maksimum. Sibuk adalah pemroses tidak menganggur, termasuk waktu yang dihabiskan untuk mengeksekusi program pemakai dan sistem operasi.

c. *Respon Time*.

Waktu yang dibutuhkan oleh suatu proses dari minta dilayani hingga ada respon pertama yang menanggapi permintaan tersebut.

d. *Throughput*.

Adalah jumlah kerja yang dapat diselesaikan dalam satu unit waktu. Cara untuk mengekspresikan *throughput* adalah dengan jumlah kerja yang dapat dieksekusi dalam satu unit/interval waktu. Sasaran penjadwalan adalah memaksimalkan jumlah kerja yang diproses persatu interval waktu. Lebih tinggi angka *throughput* lebih banyak kerja yang dilakukan sistem.

e. *Turn Around Time*.

Adalah waktu yang dihabiskan dari saat program atau kerja mulai masuk ke sistem sampai proses diselesaikan sistem, diekspresikan sebagai penjumlahan waktu eksekusi (waktu pelayanan kerja) dan waktu menunggu .

Dari pengertian diatas nampak timbul beberapa pengertian yang saling kontradiksi untuk mendapatkan waktu respon yang kecil. Penjadwalan tidak boleh menjalankan keseluruhan kerja secara bersama-sama. Karena kontradiksi tersebut maka beberapa kerja mungkin tidak cocok untuk algoritma ini, karena bagaimanapun hal ini akan mengurangi kriteria *Turn Around Time*. Hal ini memperlihatkan bahwa algoritma penjadwalan memberikan kemurahan pada sekelompok kerja pasti akan merugikan sekelompok kerja yang lain (Kleinrock,1975). Total waktu CPU yang disediakan adalah sama dan tertentu untuk semua kerja. Jika sebuah kerja selesai dilayani dan masih terdapat sisa

waktu pelayanan maka sisa tersebut diberikan kepada kerja lain. Hal tersebut bisa terjadi apabila volume kerja kecil dari pada nilai kuantumnya.

Kesulitan akan muncul apabila penjadwalan harus menangani proses yang khusus dan sulit diperkirakan. Misal saat sedang ada proses file I/O, *semaphore* dan sebagainya. Untuk memastikan bahwa tidak ada proses berjalan terlalu lama, biasanya dalam setiap komputer terdapat rangkaian elektronika pencacah waktu atau timer clock (*electronic timer on clock*). Yang berfungsi melakukan interupsi (*interrupt*) pada periode tertentu, biasanya trik yang digunakan adalah 50-60 kali per detik. Tetapi pada beberapa komputer sistem operasi mengecek apakah proses yang sedang berjalan akan terus dilanjutkan atau apakah proses telah menghabiskan waktu yang diberikan oleh CPU dan harus menyerahkan kepada proses yang lain. Strategi proses yang sedang berjalan dan berhenti sesaat secara logika disebut penjadwalan *preemptive (preemptive scheduling)* dan kebalikannya adalah metode proses lengkap seperti pada sistem bertumpuk diatas. Tampak bahwa sebuah proses bisa dihentikan tanpa peringatan dan proses lain dapat langsung dijalankan. Hal ini yang akan bisa memunculkan kondisi bersaing sehingga mengharuskan pemakaian semapor, pencatat kejadian pengamat pesan, atau hal-hal lain yang bisa mencegah munculnya kondisi bersaing tersebut.