

I. Pendahuluan

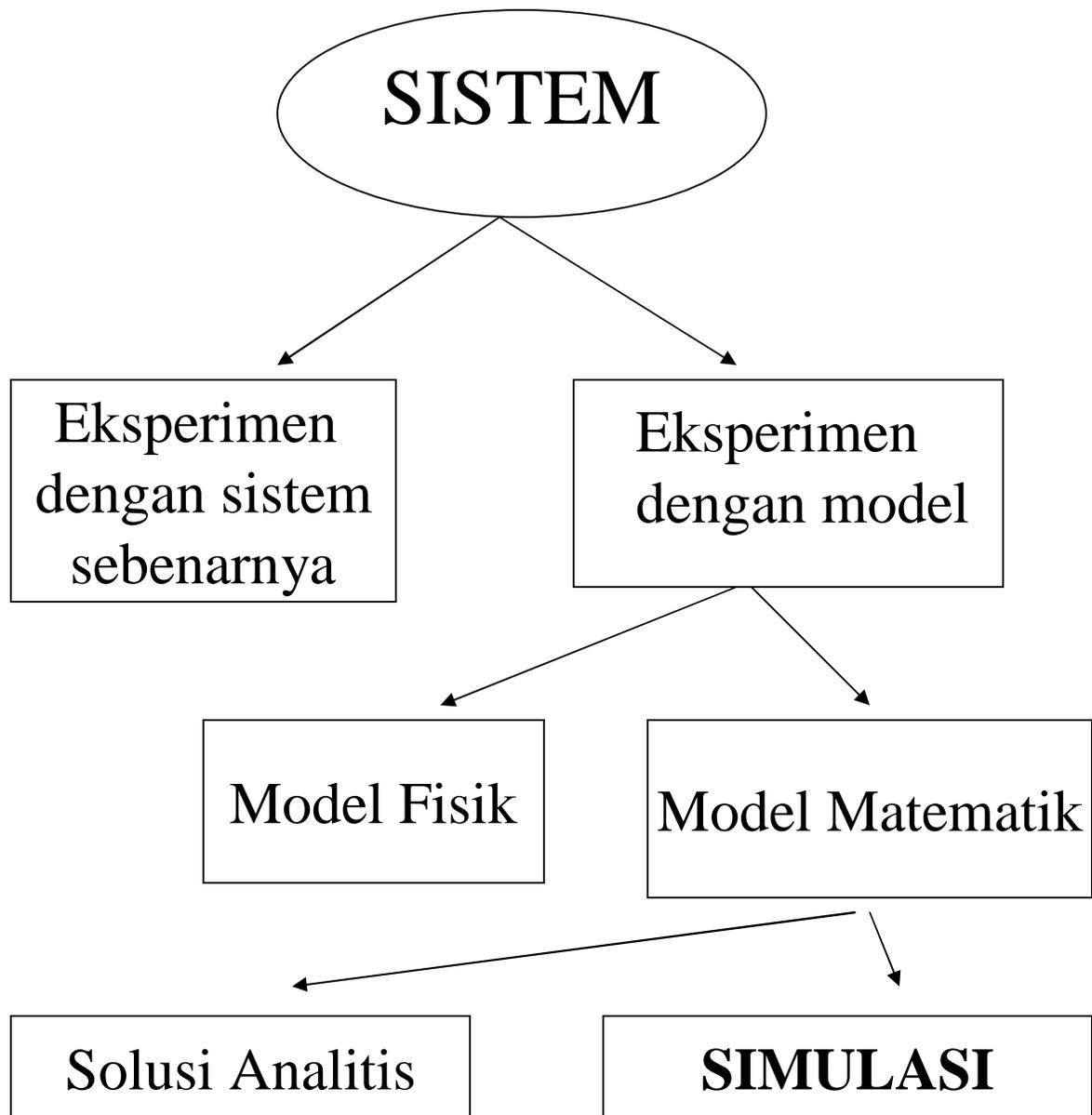
I.1 TUJUAN MEMPELAJARI SIMULASI

- Melalui kuliah ini diharapkan kita dapat mempelajari suatu sistem dengan memanfaatkan komputer untuk meniru (to simulate) perilaku sistem tersebut.

I.2. CARA MEMPELAJARI SISTEM

- Sistem dapat dipelajari dengan pengamatan langsung atau pengamatan pada model dari sistem tersebut.
- Model dapat diklasifikasikan menjadi model fisik dan model matematik
- Model matematik ada yang dapat diselesaikan dengan solusi analitis, ada yang tidak. Bila solusi analitis sulit didapatkan maka digunakan SIMULASI

Cara mempelajari Sistem



I.3 Definisi-definisi

- **SISTEM**

sekumpulan obyek yang tergabung dalam suatu interaksi dan inter-dependensi yang teratur. Sistem dibedakan menjadi dua tipe yaitu **sistem diskrit** dan **sistem kontinu**.

- **MODEL**

penyederhanaan dari sistem yang akan dipelajari.

- **SIMULASI**

suatu prosedur kuantitatif, yang menggambarkan sebuah sistem, dengan mengembangkan sebuah model dari sistem tersebut dan melakukan sederetan uji coba untuk memperkirakan perilaku sistem pada kurun waktu tertentu.

I.4. Penggunaan Simulasi

- Alternatif terakhir, bila cara lain tak dapat digunakan.
- Pada kenyataannya, berdasarkan hasil riset di US tahun 1971, dari 1000 perusahaan :
 - 20 % (paling banyak) menggunakan teknik Simulasi,
 - 21% menggunakan Linier Programming,
 - 2% menggunakan Inventori dan sisanya menggunakan berbagai teknik-teknik lain.

Mengapa Perlu Simulasi

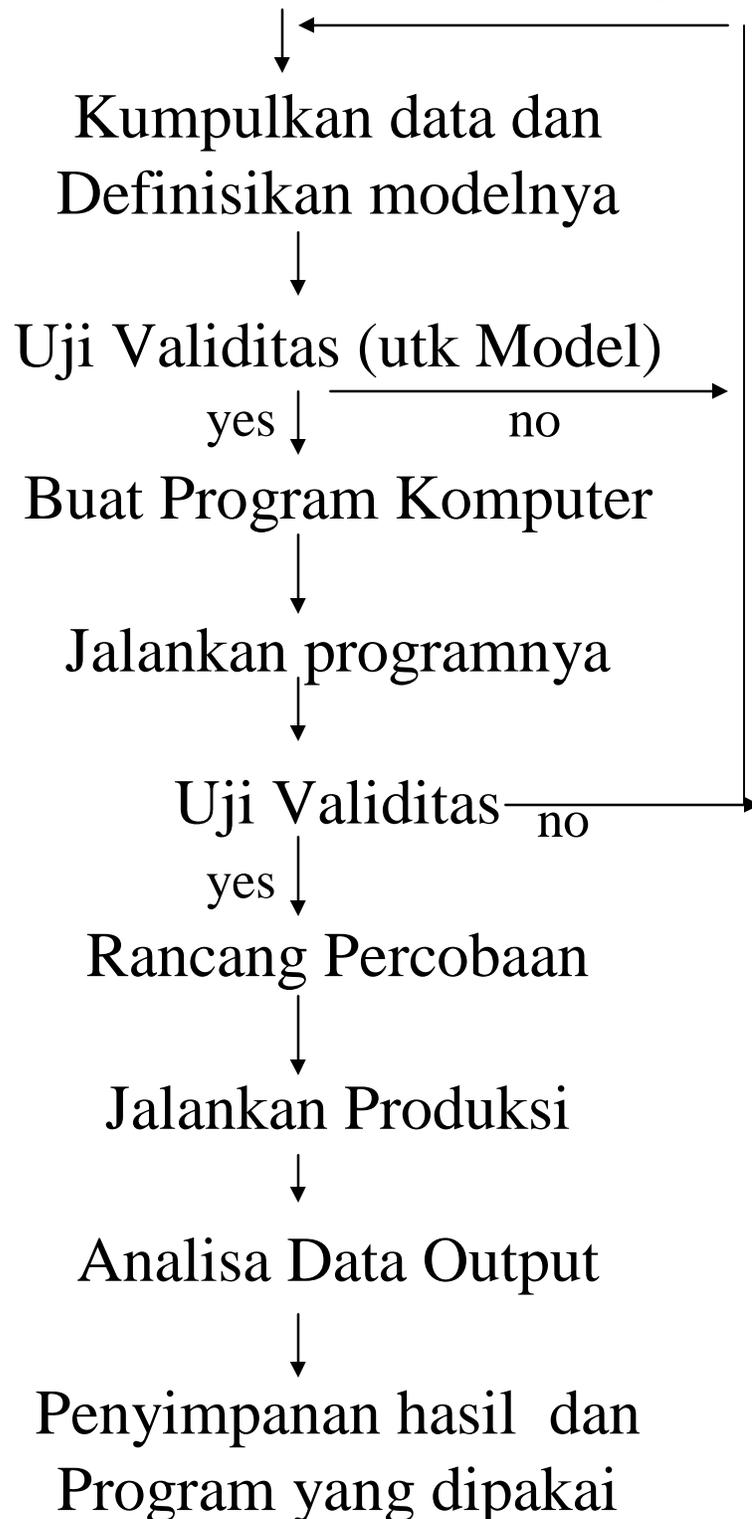
1. Simulasi adalah satu-satunya cara yang dapat digunakan untuk mengatasi masalah, jika sistem nyata sulit diamati secara langsung

Contoh : Jalur penerbangan pesawat ruang angkasa atau satelit.

2. Solusi Analitik tidak bisa dikembangkan, karena sistem sangat kompleks.
3. Pengamatan sistem secara langsung tidak dimungkinkan, karena :
 - sangat mahal
 - memakan waktu yang terlalu lama
 - akan merusak sistem yang sedang berjalan.

Langkah-langkah Studi Simulasi

Formulasikan Masalah &
Buat Rencana Pemecahannya



I.8. Kekurangan Simulasi

1. Simulasi tidak akurat.

Teknik ini bukan proses optimisasi dan tidak menghasilkan sebuah jawaban tetapi hanya menghasilkan sekumpulan output dari sistem pada berbagai kondisi yang berbeda. Dalam banyak kasus, ketelitiannya sulit diukur.

2. Model simulasi yang baik bisa jadi sangat mahal, bahkan sering dibutuhkan waktu bertahun-tahun untuk mengembangkan model yang sesuai.

3. Tidak semua situasi dapat dievaluasi dengan simulasi,

Hanya situasi yang mengandung ketidakpastian yang dapat dievaluasi dengan simulasi. Karena tanpa komponen acak semua eksperimen simulasi akan menghasilkan jawaban yang sama.

4. Simulasi menghasilkan cara untuk mengevaluasi solusi, bukan menghasilkan cara untuk memecahkan masalah.

Jadi sebelumnya perlu diketahui dulu solusi atau pendekatan solusi yang akan diuji.

I.7 Aplikasi Studi Simulasi

- Design dan analisa sistem manufaktur
- Mengetahui kebutuhan software dan hardware untuk sebuah sistem komputer.
- Mengevaluasi sistem persenjataan baru, dalam bidaang militer
- Menentukan pengaturan dalam sistem inventory/persediaan.
- Mendesign sistem transportasi
- Mendesign sistem komunikasi
- Mengevaluasi sistem pelayanan dalam bidang perbankan.
- Mengevaluasi sistem ekonomi dan finansial.

CONTOH KASUS: SIMULASI MONTE CARLO

Berikut merupakan tabel data umur server untuk komputer.

Umur (bln)	Prob (%)
1	10
2	15
3	25
4	20
5	15
6	10
7	5

Buat simulasi yang menggambarkan pergantian server komputer selama 2 tahun

Solusi

Buat tabel interval bil.random

Umur(bln)	Prob	Total Prob
1	0,10	0,10
2	0,15	0,25
3	0,25	0,50
4	0,20	0,70
5	0,15	0,85
6	0,10	0,95
7	0,05	1,00

Buat simulasi :

i	X_i	$U_i = X_i/m$	Umur(bln)	Total
1	6	0,375	3	3
2	1	0,063	1	4
3	8	0,500	3	7
4	11	0,688	4	11
5	10	0,625	4	15
6	5	0,313	3	18
7	12	0,750	5	23
8	15	0,938	7	30
9	14	0,875	6	36

Sampai dengan akhir thn ke 3 , terjadi pergantian server sebanyak 9 kali

III. Bilangan Acak & Var. Acak

- **CARA MEMPEROLEH :**
 - ZAMAN DAHULU, dgn cara :
 - Melempar dadu
 - Mengocok kartu
 - ZAMAN MODERN (>1940), dgn cara :
membentuk bilangan acak secara numerik/
aritmatik(menggunakan komputer) , disebut
“Pseudo Random Number” (bilangan
pseudo acak).
- **PEMBANGKIT BILANGAN ACAK,
HARUS :**
 - Berdistribusi uniform(0,1) dan tidak berkorelasi antar bilangan.
 - Membangkitkan cepat, storage tidak besar
 - Dapat di “reproduce”
 - Periode besar, karena mungkin bil.acak dibangkitkan berulang

Pseudo Random Number Generator

- **METODE KONGRUEN MULTIPLIKATIF**

$$X_n = (aX_{n-1}) \text{ modulo } m$$

Dimana :

- Bil. Pseudo dimulai dgn nilai awal X_0 yang disebut benih.
 - a & m : bilangan bulat positif tertentu
 - aX_{n-1} dibagi dgn m dan sisanya diambil sebagai nilai X_n
- Agar X_n berperilaku acak yang dapat dipertanggungjawabkan :
 - Modulo m dipilih sebesar mungkin untuk memperbesar periode
 - a dipilih agar korelasi antar X_n minimum
 - Benih X_0 : bil. Bulat positif ganjil, $X_0 < m$
 - Bil acak : $U_i = X_n/m$

Metode Pembangkit Kongruen Campuran

$$X_n = (aX_{n-1} + C) \text{ mod. } m$$

Pemilihan a, c, m dan x_0 :

- $m = 2^w - 1$
- $a \cong 2^{w/2}$ dan $a \equiv 1 \pmod{4}$
- c & X_0 bil. Bulat positif ganjil $< m$
($c < m$, $X_0 < m$)

Catatan:

- Periode pembangkit multiplikatif $m/4$
- Pembangkit campuran dgn periode penuh
($=m$) jika :
 - m dan c pembagi bersamanya adalah 1
 - Jika m habis dibagi oleh bil. q yang prima, maka $(a-1)$ juga habis dibagi oleh q
 - Jika m habis dibagi 4 maka begitu pula $(a-1)$

CONTOH:

- METODE KONGRUEN MULTIPLIKATIF

misal komputer berkapasitas 12 bit word

➤ $W = 12$

$$m = 2^{w-1} = 2^{11} = 2048$$

$$a = 67 \Rightarrow a \approx 2^6 \text{ \& } a \equiv 3 \pmod{8}$$

misal : $X_0 = 129$

➤ $X_1 = (67)(129) \pmod{2048} = 451$

$$X_2 = (67)(451) \pmod{2048} = 1545$$

$$X_3 = (67)(1545) \pmod{2048} = 1115$$

$$X_4 = (67)(1115) \pmod{2048} = 977$$

➤ $U_1 = 451/2048 = 0,22015$

$$U_2 = 1545/2048 = 0,754395$$

$$U_3 = 1115/2048 = 0,544434$$

$$U_4 = 977/2048 = 0,477051$$

➤ Periode : $m/4 = 2048/4 = 512$

$$U_1 = U_{513}$$

$$U_2 = U_{514}$$

CONTOH:

- METODE KONGRUEN CAMPURAN
misal komputer berkapasitas 12 bit word

$$\blacktriangleright a = 65 (\approx 2^6 \ \& \equiv 1 \pmod{4})$$

$$m = 2^{12-1} = 2048$$

$$\text{misal } c = 1, X_0 = 129$$

$$X_1 = \{(65).(129)+1\} \pmod{2048} = 194$$

$$X_2 = \{(65).(194)+1\} \pmod{2048} = 323$$

$$X_3 = \{(65).(323)+1\} \pmod{2048} = 516$$

$$X_4 = \{(65).(516)+1\} \pmod{2048} = 773$$

$$U_1 = 194/2048 = 0,094727$$

$$U_2 = 323/2048 = 0,157715$$

$$U_3 = 516/2048 = 0,251953$$

$$U_4 = 773/2048 = 0,377441$$

Variabel Acak (Random Number)

- Definisi: suatu fungsi atau aturan yang menunjukkan sebuah bilangan riil untuk suatu titik sampel pada ruang sampel S
- Biasanya variabel acak dinyatakan dengan huruf besar X, Y, Z dan nilai variabel acaknya dimisalkan dengan huruf kecil x, y, z .
- Variabel Acak terdiri dari :
 - * Variabel Acak Diskrit
 - * Variabel Acak Kontiniu

Variabel Acak Diskrit

- jika X banyak nilainya dapat dihitung (berkorelasi 1-1 dengan bilangan bulat positif)

Utk var.acak diskret X : $p(x)=P(X=x)$

Variabel Acak Kontinu

- jika banyaknya nilai x_i tak dapat dihitung, bila ada fungsi non-negatif $f(x)$ sedemikian shg utk sekumpulan bil. Riil B (misal $1 < B < 2$)

$$P(X \in B) = \int f(x) dx \text{ atau } \int f(x) dx$$

$$\text{dan } \int f(x) dx = 1$$

Semua nilai probabilitas X dapat dihitung melalui $f(x)$ yang disebut : fungsi densitas probabilitas var. acak kontiniu X

Utk var.acak kontiniu X :

$$P(X=x) = P(X \in [x,x]) = \int f(y) dy = 0$$

$$\text{tetapi } P(X \in [x,x+\Delta x]) = \int f(y) dy$$

Variabel Acak Diskrit

Distribusi Binomial

- Ciri: * Percobaan terdiri dari n ulangan independen, yang dapat diklasifikasikan menjadi berhasil atau gagal
* Probabilitas berhasil (p) dari satu ulangan ke ulangan lainnya konstan.
- Fungsi Probabilitas: $C_{n,k} p^k (1-p)^{n-k}$
- Nilai Ekspektasi: np
- Varians: $np(1-p)$

Algoritma Binomial:

- Bangkitkan U
- $C=P/(1-P)$, $I=0$, $pr=(1-P)^n$, $F=pr$
- if $U < F$, then $x=I$, stop
- $Pr=\{C(n-i)/(i+1)\}pr$, $F=F+pr$, $i=i+1$
- Go to 3

Distribusi Poisson

- Ciri: Dalam selang waktu T jumlah peristiwa terjadi independen terhadap jumlah kejadian yang terjadi pada waktu yang lain, dengan peluang kejadian tunggal selama periode waktu sangat singkat proporsional terhadap panjang interval waktu. Peluang lebih dari satu kejadian dlm waktu yang sangat singkat *neglibible*.

- Fungsi Probabilitas
$$\frac{\lambda^k e^{-\lambda}}{k!}$$

- Nilai Ekspektasi : λ

- Varians : λ

- Algoritma:

- Bangkitkan $U \sim U(0,1)$

- $i=0, p=e^{-\lambda}, F=P$

- if $U < F$ then $x=i$ stop

- $p=\lambda p/(i+1), F=F+P, i=i+1$

- Go to 3

Distribusi Hipergeometri

- Ciri: Sampel acak dengan ukuran n dipilih dari populasi ukuran N , dimana sejumlah k dapat diklasifikasikan sukses dan $N-k$ gagal.

- Fungsi Probabilitas:

$$\frac{C_{M, x} C_{N - M, n - x}}{C_{N, n}}$$

- Nilai Ekspektasi:

$$n \left(\frac{M}{N} \right)$$

- Varians:

$$\frac{nM (N - M)(N - n)}{N^2 (N - 1)}$$

Distribusi Acak Kontinu

Algoritma:

- bangkitkan bilangan acak U_1 dan U_2
- Set $t = -\log(U_1 U_2)$
- Bangkitkan bilangan acak U_3
- $X = tU_3$, $Y = t - X$

Distribusi Eksponensial

• Fungsi Probabilitas: $f(x) = ae^{-ax}$

• Nilai Ekspektasi $\frac{1}{a}$

• Varians: $\frac{1}{a^2}$

Distribusi Normal

Fungsi Probabilitas:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

• Nilai Ekspektasi: 0

• Varians: 1

Algoritma:

• Bangkitkan bilangan acak U_1 dan U_2

• Set $V_1=2U_1-1$; $V_2=2U_2-1$; $W=V_1^2+V_2^2$

• if $W > 1$ go to 1

else set $Y =$

$$X_1 = V_1 Y; \quad X_2 = V_2 Y$$

Validasi dan Verifikasi Model

- Salah satu masalah tersulit yang dihadapi oleh seorang analis simulasi adalah mencoba untuk menentukan apakah sebuah model simulasi merupakan representasi yang akurat dari sistem aktualnya,

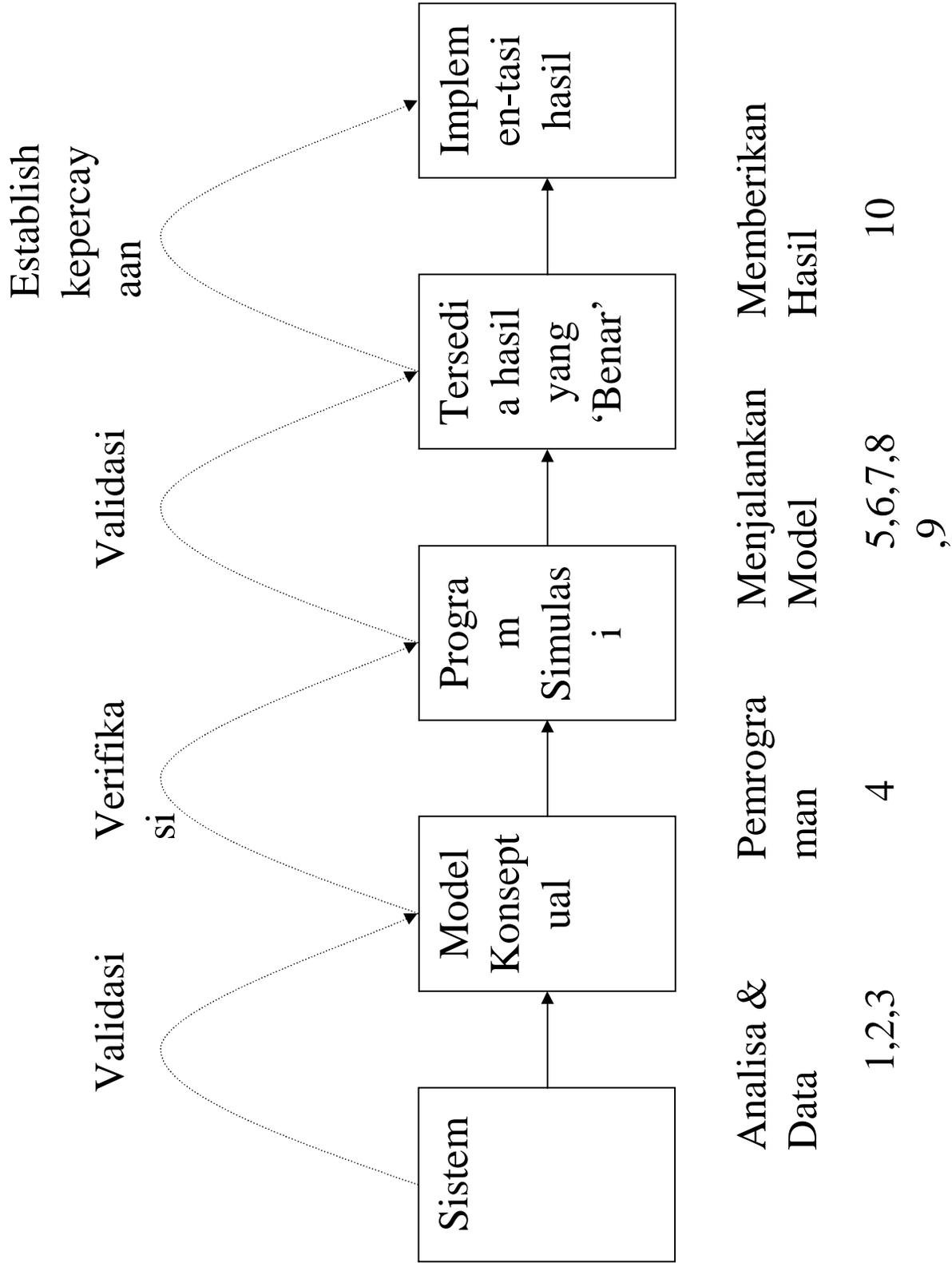
Validasi

menentukan apakah sebuah model simulasi merupakan representasi yang akurat dari sistem aktualnya.

Verifikasi

Menentukan apakah sebuah program komputer simulasi sesuai dengan tujuannya.

Waktu dan hubungan dari Validasi dan Verifikasi :



Prinsip pemodelan Simulasi Valid

Praktisi simulasi harus dapat menentukan aspek apa saja, dari sistem yang kompleks, yang perlu disertakan dalam model simulasi

Petunjuk umum dalam menentukan tingkat kedetailan yang diperlukan dalam model simulasi :

- ❖ Hati-hati dalam mendefinisikan Model-model tidak valid secara universal

- ❖ Memanfaatkan 'pakar' dan analisis sensitivitas untuk membantu menentukan tingkat kedetailan model

Teknik Verifikasi Program

Teknik 1.

Buatlah dan debug program komputer dalam modul-modul atau subprogram-subprogram

Teknik 2.

Buatlah program komputer secara bersama-sama (lebih dari satu orang)

Teknik 3.

Menjalankan simulasi dengan berbagai variasi parameter input dan memeriksa apakah outputnya reasonable

Teknik 4.

Melakukan “trace”. Teknik ini merupakan salah satu teknik yang powerful yang dapat digunakan untuk mendebug program simulasi event diskrit.

Teknik 5.

Model sebaiknya dapat dijalankan (jika memungkinkan) dengan asumsi sederhana.

Teknik 6.

Untuk beberapa model simulasi, akan lebih bermanfaat untuk melakukan observasi sebuah animasi dari output simulasi.

Teknik 7.

Tuliskan mean sampel dan variansi sampel untuk setiap probabilitas distribusi input simulasi, dan bandingkan dengan mean dan variansi yang diinginkan (misalnya secara historis)

Teknik 8.

Gunakan paket simulasi

Perspektif Umum Simulasi

- Eksperimen dengan model simulasi untuk eksperimen sistem aktual
- Kemudahan atau kesulitan dari proses validasi tergantung pada kompleksitas sistem yang dimodelkan
- Sebuah model simulasi dari sebuah sistem yang kompleks hanya dapat menjadi pendekatan terhadap aktual sistem
- Sebuah model simulasi sebaiknya selalu dibangun untuk sekumpulan tujuan tertentu
- Sebuah buku catatan dari asumsi-asumsi model simulasi sebaiknya diupdate berkala

- Sebuah model simulasi sebaiknya divalidasi relatif terhadap ukuran kinerja yang akan digunakan untuk pengambilan keputusan
- Pembentukan model dan validasi sebaiknya dilakukan sepanjang pensimulasian
- Pada umumnya tidak mungkin untuk membentuk validasi statistik secara formal diantara data output model dengan data output sistem aktual

Pendekatan Tiga Langkah untuk membangun Model Simulasi yang valid dan dapat dipercaya

Langkah 1.

Membangun sebuah model dengan usaha melibatkan informasi semaksimal mungkin.

- ❖ Berdiskusi dengan para 'pakar' sistem
- ❖ Melakukan observasi terhadap sistem
- ❖ Memanfaatkan Teori yang ada
- ❖ Memanfaatkan hasil dari Model simulasi yang sama dan relevan
- ❖ Menggunakan pengalaman atau intuisi

- ❖ Memanfaatkan Teori yang ada
- ❖ Memanfaatkan hasil dari Model simulasi yang sama dan relevan
- ❖ Menggunakan pengalaman atau intuisi

Langkah 2.

Menguji asumsi-asumsi model secara empiris

Jika distribusi probabilitas secara teoritis cocok dengan observasi dan digunakan sebagai input untuk model simulasi, dapat diuji dengan pembuatan grafik dan uji goodness-of-fit

Jika beberapa himpunan data diobservasi untuk fenomena random yang sama, maka perbaikan dari penggabungan data tersebut dapat ditentukan dengan uji Kruskal-Wallis

Salah satu utiliti yang sangat berguna adalah analisis sensitivitas

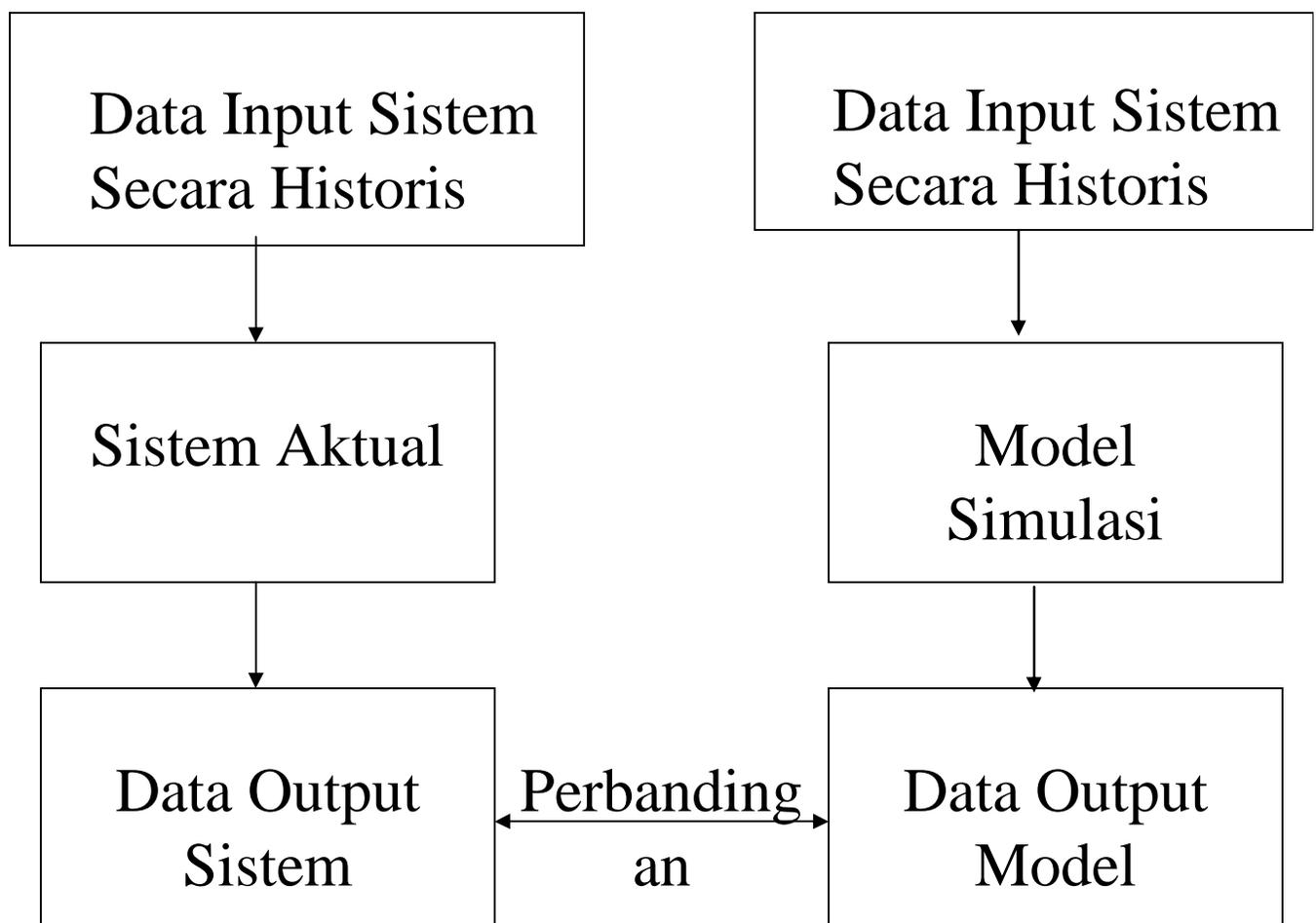
Langkah 3.

Menentukan seberapa representatif data output Simulasi

Prosedur Statistik untuk membandingkan data output dari observasi dunia nyata dan simulasi

- Pendekatan Inspeksi

Korelasi pendekatan inspeksi :



- Pendekatan Interval Konfidensi berdasarkan data independen
- Pendekatan Time Series

CONTOH KASUS - 1

- **Kasus Komputer On-Line**
 - Komputer segera merespon perintah yang diterimanya
 - Perintah diterima melalui saluran komunikasi dengan kecepatan B perintah / detik
 - Rata-rata setiap perintah terdiri dari b karakter.
 - Sebagian perintah (k) membutuhkan jawaban rata-rata sebanyak r karakter.
 - Setiap perintah diterima oleh buffer (sekaligus tempat mengirim jawaban) yang berdaya tampung maksimum m karakter per detik.

CONTOH KASUS - 1 lanjutan

- Proses sebuah perintah yang diterima membutuhkan 2000 instruksi.
- Penyiapan jawaban membutuhkan program dengan 1000 instruksi.
- Proses interupsi dalam melakukan transfer data baik ke dalam / ke luar komputer membutuhkan eksekusi 1000 instruksi.
- **Asumsi :**
 1. Terdapat 3 jenis komputer berdasarkan tingkat kecepatan :
 - .kecepatan rendah ($P1 = 25.000$ instruksi/detik)
 - .kecepatan sedang ($P2 = 50.000$ instruksi/detik)
 - .kecepatan tinggi ($P3 = 100.000$ instruksi/detik)
 2. Terdapat 4 ukuran buffer ($m = 1, 2, 5,$ dan 10 karakter

CONTOH KASUS - 1 lanjutan

- **Permasalahan :**

- ✨ Bila data harga diketahui maka mana rancangan yang termurah ?

- ✨.Rancangan komputer yang mana yang mampu mempertahankan aliran data ?

- **Solusi :**

- Akan dihitung berapa karakter per detik yang ditransfer dan

- membandingkannya dengan jumlah instruksi yang harus dieksekusi setiap detiknya.

CONTOH KASUS - 1 lanjutan

Berdasarkan kondisi yang ada :

- Terdapat B perintah yang masuk dan kB jawaban yang ke luar per detik, sehingga akan membutuhkan $Bb + kBr$ karakter per detik untuk melewati Buffer.
- Kapasitas maksimum Buffer m karakter sehingga akan terdapat $(Bb + kBr)/m$ interupsi per detik.

Instruksi yang terjadi per detik:

- untuk perintah masuk = $2000 \times B$
- untuk jawaban = $10000 \times kB$
- untuk interupsi = $1000 \times B(b+kr)/m$

Jumlah Instruksi per detik (N) :

$$N = 2000 \times B + 10000 \times kB + 1000 \times B(b + kr)/m$$

CONTOH KASUS - 1 lanjutan

Jadi Model Matematikanya :

$$\mathbf{N = 2000 \times B + 10000 \times kB + 1000 \times B(b + kr)/m}$$

Simulasinya : Bila

B = 5 perintah

b = 15 karakter

k = 10 % dari perintah yang memerlukan jawaban

r = 50 karakter jawaban

CONTOH KASUS - 1 lanjutan

Maka rancangan komputer yang mampu mempertahankan aliran data yaitu :

$$N \leq P(i)$$

$$N = 2000 \times 5 + 10000 \times 5 \times 0.1 + 1000 \times 5 (15 + 5)/m$$

$$= 15000 + 100000/m \leq P(i) \quad \dots \quad (N \leq P(i))$$

$$3 + 20/m \leq P(i)/5000$$

$$20/m \leq P(i)/5000 - 3 \quad \dots\dots\dots(1)$$

CONTOH KASUS - 1 lanjutan

Berdasarkan pers (1), dengan kondisi minimal akan diperoleh rancangan dengan alternatif sebagai berikut :

1. Komputer **Kecepatan tinggi** dengan Buffer **2** karakter
2. Komputer **Kecepatan sedang** dengan Buffer **5** karakter
3. Komputer **Kecepatan rendah** dengan Buffer **10** karakter

Bila memperhatikan harga dari komputer berdasarkan tingkat kecepatannya maka solusi optimal yang diperoleh dengan tetap mempertahankan aliran data yang ada yaitu :

Rancangan Komputer Kecepatan rendah
dengan Buffer 10 karakter

CONTOH KASUS - 1 lanjutan

Algoritma Kasus 1 :

1. Input data sistem.
2. Hitung jumlah instruksi yang dibutuhkan per detik untuk setiap alternatif buffer yang ada
3. Bandingkan setiap hasil langkah-2 dengan kecepatan maksimum dari setiap kemungkinan komputer yang tersedia.
4. Tetapkan rancangan sistem komputer yang mampu mempertahankan aliran data.

CONTOH KASUS - 2

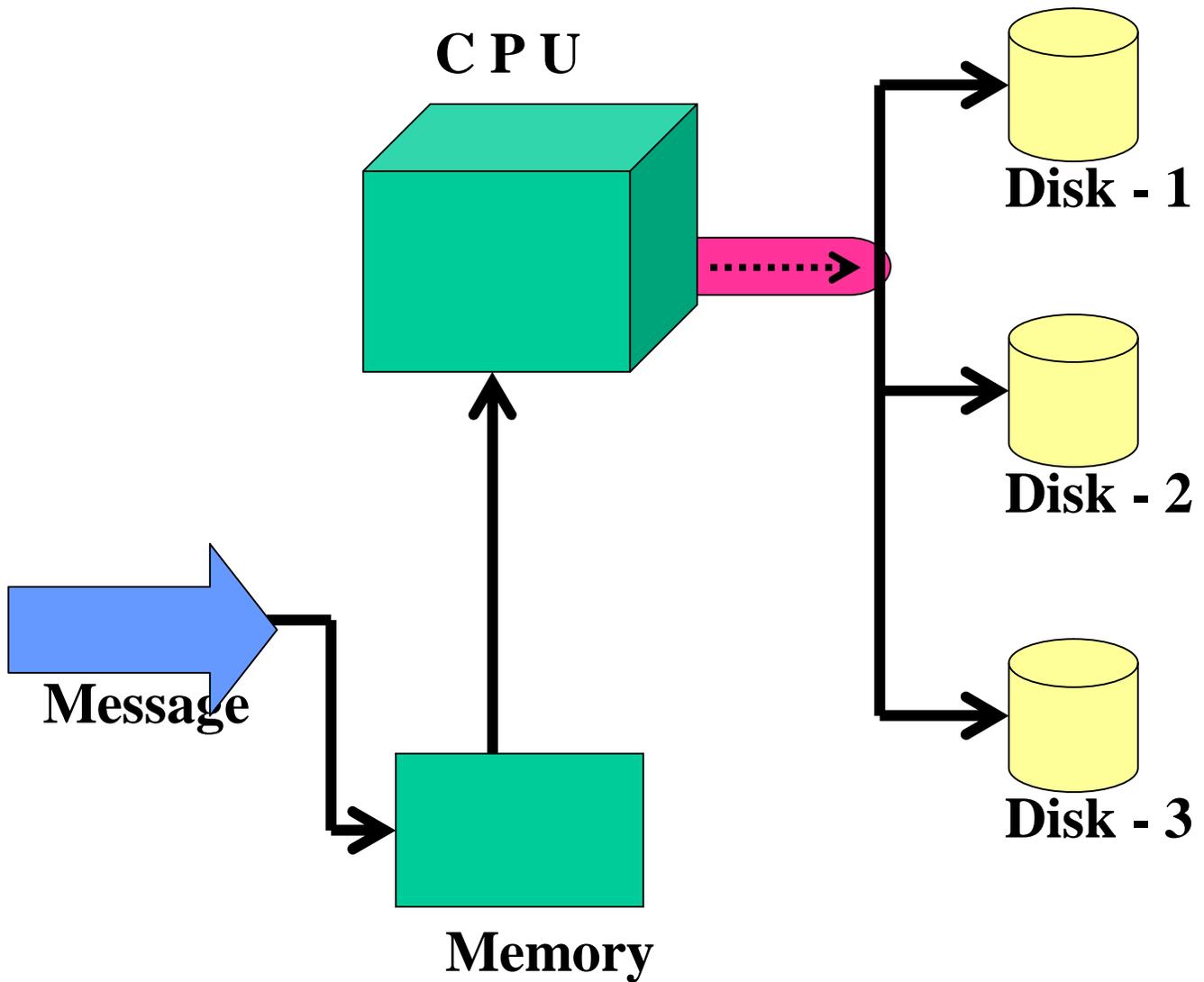
- **Kasus Komputer Real Time**

Diberikan suatu rancangan sebuah mesin pengolah data yang bekerja secara real time, yang terdiri dari komponen-komponen (lihat Gambar 2.1) :

- Terminal Masukkan,
- CPU,
- Memory,
- Sebuah saluran komunikasi dari CPU ke media penyimpanan (disk), dan
- 3 buah media penyimpanan.

CONTOH KASUS - 2

lanjutan



Gambar 2.1. Rancangan Mesin - Real Time

CONTOH KASUS - 2 lanjutan

Spesifikasi perangkat keras dari komponen-komponen sistim :

- Kecepatan CPU mengolah data bervariasi antara 6 milidetik (mdet) sampai dengan 14 mdet.
- Waktu yang dibutuhkan CPU untuk memasukkan perintah (message) ke dalam memori adalah 1 mdet.
- Panjang perintah berkisar antara 10 sampai 20 unit.
- Kapasitas memori 2000 unit.
- Waktu untuk mencari lintasan dari disket berkisar antara 40 - 2000 mdet.

CONTOH KASUS - 2 lanjutan

- Waktu untuk mencari sektor media penyimpanan berkisar antara 0 - 50 mdet.
- Berkas yang dibutuhkan agar suatu perintah dapat dilayani terdistribusi secara merata ke ketiga media penyimpanan.
- Waktu yang dibutuhkan oleh berkas untuk menempuh saluran komunikasi adalah 2 mdet.

CONTOH KASUS - 2 lanjutan

Asumsi dari komponen-komponen sistim :

- Waktu kedatangan perintah berdistribusi eksponensial dengan rata-rata 50 mdet.
- Berkas yang dibutuhkan oleh suatu perintah tersebar ke ketiga media penyimpanan dengan kemungkinan yang sama.
- Panjang perintah berdistribusi uniform dari 10 - 20 unit.
- Waktu proses berdistribusi normal dengan rata-rata 10 mdet dan standar deviasi 4 mdet.

CONTOH KASUS - 2 lanjutan

Pengendalian kerja sistim :

- Perintah yang dibaca oleh CPU dimasukkan dalam antrian di memori.
- Kemudian CPU mendecode perintah yang berada terdepan pada antrian di memori.
- Bila saluran komunikasi dalam keadaan tidak terpakai maka berkas yang dibutuhkan oleh perintah yang sedang dilayani dapat ditransfer ke CPU untuk diolah.
- Sebaliknya, bila saluran komunikasi dalam keadaan terpakai maka pelayanan ditunda sampai saluran komunikasi siap dan permintaan akan ditransfer berkas dimasukkan dalam antrian saluran komunikasi.

CONTOH KASUS - 2 lanjutan

Permasalahan :

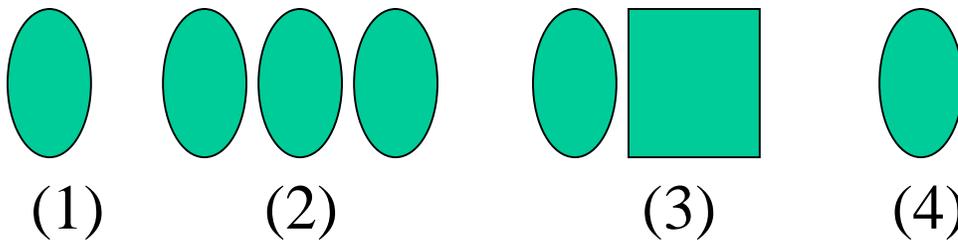
Bagaimana unjuk kerja dari sistim melalui pengukuran besaran-besaran sebagai berikut :

- Waktu tunggu rata-rata, waktu tunggu maksimum, waktu tunggu minimum dari suatu perintah untuk mendapat layanan CPU,
- Panjang antrian yang terjadi,
- Utilisasi dari komponen-komponen sistim (waktu sibuk CPU, waktu sibuk saluran komunikasi, rata-rata pemakaian memori).

Algoritma dan Program Sistem Antrian Pelayan Tunggal Sederhana

- **Contoh antrian** : *car wash*, kantor pos, *mobile host*,

- **Gambaran Masalah**



(1) kedatangan entitas

(2) antrian

(3) pelayanan

(4) keberangkatan

Komponen Simulasi :

- Ukuran *buffer* (ruang antrian)
- Skema pelayanan (FIFO, LIFO, dll)
- Variabel acak

Variabel Acak dan Fungsi Probabilitas :

- Pola kedatangan : selang waktu antar dua kedatangan

berurutan equivalen dengan waktu setiap kedatangan

$\sim f_A =$ fungsi probabilitas waktu kedatangan

- Pola pelayanan : durasi pelayanan

$\sim f_S =$ fungsi probabilitas durasi pelayanan

Contoh Soal :

Perkirakan jumlah rata-rata entitas dalam sebuah sistem antrian pelayan tunggal dengan *buffer* berukuran tak hingga dan skema pelayanan FIFO.

Membangun model dari contoh soal :

- Spesifikasi input : bentuk fungsi f_A dan f_S
- Spesifikasi ukuran kinerja : jumlah rata-rata entitas dalam sebuah sistem antrian :

$$Q_{av} = (1/t) \sum_t Q(u) du, \text{ dimana } Q(u)$$

adalah jumlah entitas dalam antrian pada waktu u

- Variabel output : jumlah kumulatif entitas dalam antrian :

$$Q = \sum_t Q(u) du$$

- Hubungan Masukan-keluaran : Dilakukan oleh simulator

(a) Peristiwa : - entitas masuk antrian

- entitas masuk pelayanan

- entitas meninggalkan pelayanan

Daftar_Peristiwa $E(t_A, t_D)$: daftar peristiwa berikut

t_A : waktu kedatangan berikut,

t_D : waktu keberangkatan berikut

(b) Keadaan : n jumlah entitas dalam sistem antrian

- Kriteria akhir simulasi : T (durasi simulasi keseluruhan)

(bisa juga yang lain, misalnya n_A , atau $M =$ kapasitas *buffer*)

Algoritma_Utama

(Computer code dapat dilihat di *lecture note*)

*/*Inisialisasi*/*

$n = 0; t = 0; Q = 0; E(t_A = \infty, t_D = \infty)$

$t_A = t + X (X \sim f_A)$

*/*loop utama*/*

while ($t \leq T$)

$t_E = \min(t_A, t_D)$

if ($t_E = t_A$) call Prosedur_Kedatangan

else call Prosedur_Keberangkatan

end

return $Q_{av} = Q/t$

Prosedur_Kedatangan

*/*Update Variabel*/*

$$Q = Q + n \times (t_E - t)$$

/ Update Keadaan*/*

$$n = n - 1$$

/ Update Waktu*/*

$$t = t_E$$

/ Penjadwalan Peristiwa Baru*/*

$$t_A = t + X (X \sim f_A)$$

$$\text{if } (n = 1) t_D = t + Y (Y \sim f_S)$$

Prosedur_Keberangkatan

*/*Update Variabel*/*

$$Q = Q + n \times (t_E - t)$$

/ Update Keadaan*/*

$$n = n + 1$$

/ Update Waktu*/*

$$t = t_E$$

/ Penjadwalan Peristiwa Baru*/*

if ($n \geq 1$) $t_D = t + Y$ ($Y \sim f_S$)

else $t_D = \infty$

***Mobile Host* sebagai Sistem
Antrian Pelayan Tunggal
Sederhana**

(‘ntar dulu, dech)