

Catatan Kuliah

Simulasi dan Pemodelan

Kata Pengantar

Puji syukur kehadiran Tuhan Yang Maha Esa, karena dengan rahmat dan berkat Hidayahnya catatan kuliah ini dapat diselesaikan.

Dalam suatu institusi pendidikan, proses utama yang sangat perlu diperhatikan dan merupakan tolok ukur dari kualitas institusi tersebut adalah proses belajar mengajar yang terjadi antara mahasiswa dan dosen. Guna menunjang proses tersebut kami team pengajar simulasi dan pemodelan menyusun catatan kuliah ini.

Selain diperuntukkan bagi mahasiswa, catatan kuliah ini juga diharapkan dapat digunakan sebagai acuan keseragaman materi antar dosen yang mengajar pada beberapa kelas parallel di Jurusan Teknik Informatika.

Kami sangat mengharapkan saran dan kritik membangun dari para mahasiswa, dosen dan pembaca guna kesempurnaan catatan kuliah ini.

Depok, 2003

Penyusun

Daftar Isi

I Gambaran Umum Simulasi, Prinsip-Prinsip Umum Sistem Simulasi Peristiwa Diskrit ix

1	Pengantar Studi Simulasi (Kuliah 1-2)	1
1.1	Definisi Simulasi	3
1.2	Model Simulasi	3
1.3	Dimana Simulasi Cocok digunakan?	4
1.4	Dimana Simulasi Tidak Cocok digunakan?	4
1.5	Bidang-Bidang Aplikasi	5
1.6	Sistem dan Lingkungan Sistem	5
1.7	Komponen Sistem	5
1.8	Sistem Diskrit dan Kontinyu	7
1.9	Tipe-Tipe Model	7
1.10	Klasifikasi Model Simulasi	7
1.11	Simulasi Sistem Peristiwa Diskrit	8
1.12	Langkah-Langkah Studi Simulasi	8
1.13	Verifikasi dan Validasi	12
1.14	Pembangunan Model	12
1.15	Kelebihan, Kekurangan, 'Pitfalls' dari Simulasi	13
1.15.1	Kelebihan	13
1.15.2	Kekurangan	13
1.15.3	Pitfalls	13
1.16	Fitur-fitur software simulasi yang dibutuhkan	14
2	Contoh-Contoh Simulasi (Kuliah 1-2)	15
2.1	Langkah-Langkah Dasar	15
2.2	Simulasi Sistem Antrian	16
2.2.1	Sistem Antrian	16
2.2.2	Keacakan dalam simulasi	18
2.3	Sistem Antrian Layanan Tunggal	19
2.4	Contoh-Contoh Lain	22
2.4.1	Masalah Able Baker Carhop: Dua Pelayan.	22

2.4.2	Sistem Inventory	24
2.4.3	Masalah Reabilitas	24
2.4.4	Masalah Militer	24
2.4.5	Lead-Time Demand	25
2.5	Ringkasan	25
3	Prinsip Umum SSPD (Kuliah 3)	26
3.1	Konsep dan Definisi	28
3.2	Time in Simulation	30
3.3	Algoritma Umum	30
3.3.1	Eksekutif Simulasi Sinkron	30
3.3.2	Eksekutif Event-Scanning	31
3.4	Mekanisme Eksekusi SSPD	31
3.5	Pendekatan-Pendekatan dalam SSPD	33
3.5.1	Pendekatan event-scheduling	34
3.5.2	Pendekatan process-interaction	34
3.5.3	Pendekatan activity-scanning	35
3.6	Contoh-Contoh lain	36
3.6.1	Contoh 3.1: Able and Baker, versi revisi.	36
3.6.2	Contoh 3.2: Antrian single-channel (Supermarket check-out counter).	36
3.6.3	Contoh 3.4: Simulasi check-out counter, lanjutan	37
3.6.4	Contoh 3.5: Masalah dump truck.	38
4	Bahasa-Bahasa Simulasi	39
4.1	Bahasa Simulasi Kontinyu dan Diskrit	39
4.2	Bahasa Simulasi Kontinyu	39
4.3	Bahasa Simulasi Sistem Diskrit.	40
4.3.1	Event-oriented languages.	41
4.3.2	Activity-oriented languages.	41
4.3.3	Process-oriented languages.	41
4.4	SIMSCRIPT.	42
4.5	GPSS (General Purpose Simulation System)	45
4.5.1	Single-Server Queue Simulation in GPSS/H	45
4.6	SIMULA (SIMULation Language)	47
4.7	Factors in selection of a discrete system simulation language.	48
II	Model Matematis dan Statistik	50
5	Model Statistik dalam Simulasi	51

5.1	Alasan Penggunaan Probabilitas dan Statistik dalam Simulasi	52
5.2	Tinjauan Ulang Terminologi dan Konsep	52
5.3	Model-Model Statistik yang Berguna	54
5.4	Distribusi Variabel Acak Diskrit	55
5.4.1	Bernoulli trials dan distribusi Bernoulli	55
5.4.2	Distribusi Binomial	56
5.4.3	Distribusi Geometrik	57
5.4.4	Distribusi Poisson	58
5.5	Distribusi Variabel Acak Kontinyu	58
5.5.1	Distribusi Uniform	58
5.5.2	Distribusi Eksponensial	59
5.5.3	Distribusi Gamma	60
5.5.4	Distribusi Erlang	61
5.5.5	Distribusi Normal	61
5.5.6	Distribusi Weibull	62
5.6	Estimasi Means dan Variances	62
5.7	Confidence Intervals dari Mean	63
5.8	Testing Hipotesis	63
5.9	Distribusi Empiris dan Ringkasan	64
6	Gambaran Umum Model-Model Antrian	65
6.1	Model-Model Antrian	65
6.2	Karakteristik Sistem Antrian	66
6.3	Sifat Antrian dan Displin Antrian	67
6.3.1	Sifat Antrian	67
6.3.2	Displin Antrian	67
6.4	Waktu Pelayanan dan Mekanisme Pelayanan	68
6.5	Notasi Antrian	68
6.6	Pengukuran Long-Run Kinerja Sistem Antrian	69
6.7	Steady-State Populasi Tak-Terbatas Model Markovian	71
6.7.1	$M=G=1$	72
6.8	Jaringan Antrian	73
6.9	Ringkasan	74
III	Bilangan dan Variabel Acak	75
7	Pembangkitan Bilangan Acak	76
7.1	Sifat-Sifat Bilangan Acak	76
7.2	Pembangkitan Bilangan Acak Pseudo	77
7.3	Teknik Pembangkitan Bilangan Acak	78

7.3.1	Metode Kongruen Linier	78
7.3.2	Metode Kongruen Linier Kombinasi	79
7.4	Test Bilangan Acak	79
7.4.1	Tes Frekuensi	81
7.4.2	Tes Runs	82
7.4.3	Tes Auto-correlation	85
7.4.4	Tes Gap	86
7.4.5	Tes Poker	88
8	Pembangkitan Variabel (Variate) Acak	89
8.1	Teknik Transformasi Balik	89
8.1.1	Distribusi Eksponensial	90
8.1.2	Distribusi Uniform	92
8.1.3	Distribusi Weibull	92
8.1.4	Distribusi Triangular	93
8.1.5	Distribusi Kontinyu Empiris	94
8.1.6	Distribusi Kontinyu tanpa invers bentuk tertutup	95
8.1.7	Distribusi Diskrit	95
8.2	Transformasi Langsung Distribusi Normal	98
8.3	Metode Konvolusi	98
8.4	Teknik Penerimaan Penolakan (Acceptance-Rejection)	99
IV	Analisis Data Simulasi	102
9	Pemodelan Masukan (Input Modeling)	103
9.1	Identi...kasi Distribusi dengan Data	103
9.1.1	Histogram	103
9.1.2	Penyeleksian Kelas Distribusi	104
9.1.3	Plot Quantile-Quantile	105
9.1.4	Estimasi Parameter	106
9.2	Tes Goodness-of-Fit	107
9.2.1	Tes Chi-Square	107
10	Veri...kasi dan Validasi Model Simulasi	110
10.1	Pembangunan, Veri...kasi dan Validasi Model	110
10.2	Veri...kasi Model Simulasi	111
10.3	Kalibrasi dan Validasi Model	111
10.3.1	Face Validity	112
10.3.2	Validasi Asumsi Model	112
10.3.3	Validasi Transformasi Input-Output	112

10.3.4 Validasi Input-Output menggunakan Data masukan historis	113
11 Analisis Keluaran Model Tunggal	115
11.1 Sifat Stokastik Data Keluaran	115
11.2 Jenis Simulasi menurut Analisis Keluaran	116
11.3 Pengukuran Kinerja dan Estimasi	116
11.3.1 Estimasi Titik	116
11.3.2 Estimasi Interval	117
11.4 Analisis Keluaran Simulasi Terminating	120
11.4.1 Estimasi Interval untuk Jumlah Replikasi yang tetap .	120
11.4.2 Estimasi Interval dengan Presisi Tertentu	121
11.5 Analisis Keluaran Simulasi Steady-State	122
11.5.1 Inisialisasi Bias pada Simulasi Steady-State	122
11.5.2 Metode Replikasi Simulasi Steady-State	123
11.5.3 Ukuran Sample Simulasi Steady-State	124
11.5.4 Batch Means untuk Estimasi Interval Estimation pada Simulasi Steady-State	125

Daftar Gambar

1.1	Cara mempelajari sebuah sistem	6
1.2	Langkah dalam studi simulasi	11
2.1	Sistem Antrian	17
2.2	Diagram Aliran Layan yang telah selesai	17
2.3	Diagram Aliran unit memasuki sistem	18
2.4	Sistem Antrian Pelayan Tunggal	20
2.5	Penentuan waktu antar ketibaan	20
2.6	Hasil Simulasi	21
2.7	Sistem Antrian Dua Pelayan	23
2.8	Hasil simulasi sistem antrian dua pelayan	24

Daftar Tabel

2.1	Tabel Simulasi	16
2.2	Aksi-Aksi Potensial saat kedatangan	17
2.3	Keluaran (outcomes) Pelayan setelah layanan selesai	18
2.4	Contoh hasil pembangkitan distribusi yang sederhana	19
2.5	Distribusi waktu antar ketibaan	22
2.6	Distribusi waktu pelayanan dari Able	23
2.7	Distribusi waktu pelayanan dari Baker	23

Bagian I

Gambaran Umum Simulasi, Prinsip-Prinsip Umum Sistem Simulasi Peristiwa Diskrit

Bab 1

Pengantar Studi Simulasi (Kuliah 1-2)

Bahasan:

- 2 Pendahuluan studi simulasi
 - Pengertian dan tujuan simulasi
 - Manfaat dan kelebihan pendekatan simulasi
 - Penerapan Simulasi
- 2 Sistem, Model & Simulasi
 - Definisi dari sistem dan model
 - Sistem, Model & Simulasi

TIU:

- 2 Mahasiswa mengerti arti dan manfaat studi simulasi, serta mendapatkan gambaran tentang cakupan studi simulasi
- 2 Mahasiswa dapat membangun model yang akan disimulasikan dan memahami definisi simulasi.

TIK:

- 2 Mahasiswa mampu mengikhtisarkan pentingnya simulasi sehingga lebih termotivasi untuk memahaminya lebih lanjut
- 2 Mahasiswa dapat menyebutkan manfaat dan kelebihan-kelebihan pendekatan simulasi.

- ² Mahasiswa dapat menyebutkan bidang-bidang atau ilmu-ilmu yang sering menggunakan pendekatan simulasi.
- ² Mahasiswa mampu membandingkan sistem dan model, dan menyimpulkan perlunya model untuk kebutuhan simulasi.
- ² Mahasiswa mampu menggolongkan model ...sis dan model matematis, baik yang statis maupun dinamis.
- ² Mahasiswa dapat menyimpulkan langkah-langkah dalam studi simulasi secara garis besar.

Deskripsi Singkat:

Pada perkuliahan ini gambaran umum studi simulasi akan diberikan, mulai dari pengertian, tujuan, manfaat, sampai penerapannya. Deskripsi sistem, model, komponen sistem serta kaitannya dengan simulasi akan dijelaskan.

Bahan Bacaan:

- ² [1] J. Banks, et.al., "Discrete Event System Simulation", 3ed., Prentice-Hall (Chap. 1)
- ² [2] Law & Kelton, et.al., "Simulation Modeling & Analysis", Mc. Graw-Hill Inc., Singapore (Chap. 1)

1.1 Definisi Simulasi

- ² Simulasi adalah peniruan operasi, menurut waktu, sebuah proses atau sistem dunia nyata.
 - 1. Dapat dilakukan secara manual maupun dengan bantuan komputer.
 - 2. Menyertakan pembentukan data dan sejarah buatan (artificial history) dari sebuah sistem, pengamatan data dan sejarah, dan kesimpulan yang terkait dengan karakteristik sistem-sistem.
- ² Untuk mempelajari sebuah sistem, biasanya kita harus membuat asumsi-asumsi tentang operasi sistem tersebut.
- ² Asumsi-asumsi membentuk sebuah model, yang akan digunakan untuk memahami sifat/perilaku sebuah sistem.
- ² Solusi Analitik: Jika keterkaitan (relationship) model cukup sederhana, sehingga memungkinkan penggunaan metode matematis untuk memperoleh informasi eksak dari sistem
- ² Langkah riil simulasi: Mengembangkan sebuah model simulasi dan mengevaluasi model, biasanya dengan menggunakan komputer, untuk mengestimasi karakteristik yang diharapkan dari model tersebut.

1.2 Model Simulasi

- ² Suatu representasi sederhana dari sebuah sistem (atau proses atau teori), bukan sistem itu sendiri.
- ² Model-model tidak harus memiliki seluruh atribut; mereka disederhanakan, dikontrol, digeneralisasi, atau diidealkan.
- ² Untuk sebuah model yang akan digunakan, seluruh sifat-sifat relevannya harus ditetapkan dalam suatu cara yang praktis, dinyatakan dalam suatu set deskripsi terbatas yang masuk akal (reasonably).
- ² Sebuah model harus divalidasi.
- ² Setelah divalidasi, sebuah model dapat digunakan untuk menyelidiki dan memprediksi perilaku-perilaku (sifat) sistem, atau menjawab "what-if questions" untuk mempertajam pemahaman, pelatihan, prediksi, dan evaluasi alternatif.

1.3 Dimana Simulasi Cocok digunakan?

- ² Mempelajari interaksi internal (sub)-sistem yang kompleks.
- ² Mengamati sifat model dan hasil keluaran akibat perubahan lingkungan luar atau variabel internal.
- ² Meningkatkan kinerja sistem melalui pembangunan/pembentukan model.
- ² Eksperimen desain dan aturan baru sebelum diimplementasikan.
- ² Memahami dan memverifikasi solusi analitik.
- ² Mengidentifikasi dan menetapkan persyaratan-persyaratan.
- ² Alat bantu pelatihan dan pembelajaran dengan biaya lebih rendah.
- ² Visualisasi operasi melalui animasi.
- ² Masalahnya sulit, memakan waktu, atau tidak mungkin diselesaikan melalui metode analitik atau numerik konvensional.

1.4 Dimana Simulasi Tidak Cocok digunakan?

- ² Jika masalah dapat diselesaikan dengan metode sederhana.
- ² Jika masalah dapat diselesaikan secara analitik.
- ² Jika eksperimen langsung lebih mudah dilakukan.
- ² Jika biaya terlalu mahal.
- ² Jika sumber daya atau waktu tidak tersedia.
- ² Jika tidak ada data yang tersedia.
- ² Jika verifikasi dan validasi tidak dapat dilakukan.
- ² Jika daya melebihi kapasitas (overestimated).
- ² Jika sistem terlalu kompleks atau tidak dapat didemonstrasikan.

1.5 Bidang-Bidang Aplikasi

- ² Perancangan dan analisis sistem manufacturing.
- ² Evaluasi persyaratan hardware dan software untuk sistem komputer.
- ² Evaluasi sistem senjata atau taktik militer yang baru.
- ² Perancangan sistem komunikasi dan message protocol.
- ² Perancangan dan pengoperasian fasilitas transportasi, mis. jalan tol, bandara, rel kereta, atau pelabuhan.
- ² Evaluasi perancangan organisasi jasa, mis. rumah sakit, kantor pos, atau restoran fast food.
- ² Analisis sistem keuangan atau ekonomi.

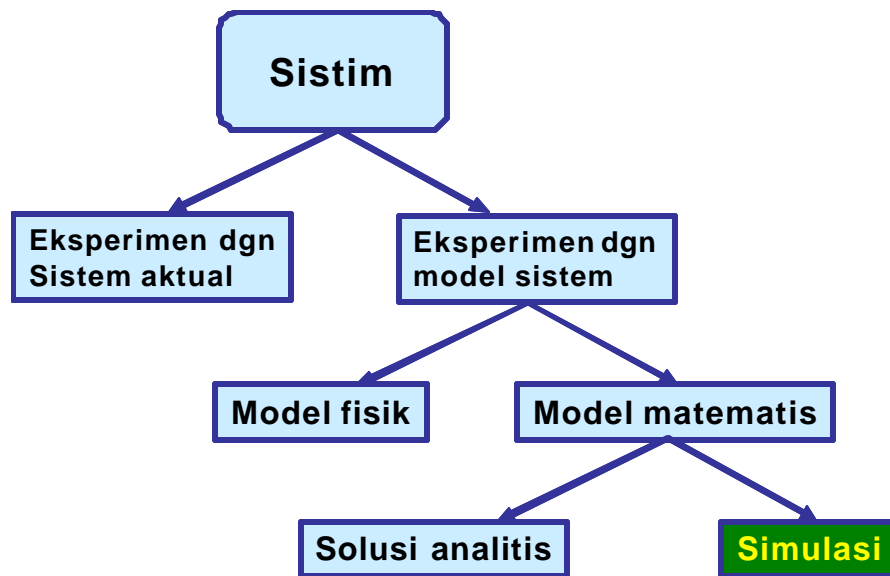
1.6 Sistem dan Lingkungan Sistem

- ² Sistem adalah sekumpulan obyek yang dihubungkan satu sama lain melalui beberapa interaksi reguler atau secara bebas untuk mencapai suatu tujuan.
- ² Sistem biasanya dipengaruhi oleh perubahan yang terjadi di luar sistem. Perubahan ini terjadi di lingkungan sistem. Dalam pemodelan sistem, perlu ditetapkan batas (boundary) antara sistem dan lingkungannya. Contoh, pada studi memori cache menggunakan, kita harus menetapkan dimana batas sistem. Batas ini dapat antara CPU dan cache, atau dapat memasukan memori utama, disk, OS, kompilator, ataupun program-program aplikasi.
- ² Cara mempelajari sebuah sistem

Mempelajari sistem dengan simulasi, secara numerik menjalankan model dengan memberi input dan melihat pengaruhnya terhadap output.

1.7 Komponen Sistem

- ² Entitas merupakan obyek dalam sistem. Contoh, customers pada suatu bank.



Gambar 1.1: Cara mempelajari sebuah sistem

- ² **Atribut** merupakan suatu sifat dari suatu entitas. Contoh, pengecekan neraca rekening customer.
- ² **Aktivitas** merepresentasikan suatu periode waktu dengan lama tertentu (specified length). Periode waktu sangat penting karena biasanya simulasi menyertakan besaran waktu. Contoh deposito uang ke rekening pada waktu dan tanggal tertentu.
- ² **Keadaan sistem** didefinisikan sebagai kumpulan variabel-variabel yang diperlukan untuk menggambarkan sistem kapanpun, relatif terhadap obyektif dari studi. Contoh, jumlah teller yang sibuk, jumlah customer yang menunggu dibaris antrian.
- ² **Peristiwa** didefinisikan sebagai kejadian sesaat yang dapat mengubah keadaan sistem. Contoh, kedatangan customer, pejumlahan jumlah teller, keberangkatan customer.

Contoh-contoh entitas, atribut, aktivitas, peristiwa dan variabel-variabel keadaan dari berbagai sistim dapat dilihat pada referensi [1] tabel 1.1. halaman 11.

1.8 Sistem Diskrit dan Kontinyu

² Sistem Diskrit: variabel-variabel keadaan hanya berubah pada set titik waktu yang diskrit.

- Contoh: jumlah customer yang menunggu diantrian

² Sistem Kontinyu: variabel-variabel berubah secara kontinyu menurut waktu.

- Contoh: arus listrik

1.9 Tipe-Tipe Model

² Model:

- Fisik: model rumah, model jembatan

- Matematis (symbolic): $\dot{x} = f(x, u)$

- Model simulasi

- † Statik (pada beberapa titik waktu) vs. Dinamik (berubah menurut waktu)

- † Deterministik (masukan diketahui) vs. Stokastik (variabel acak, masukan/keluaran)

- † Diskrit vs. Kontinyu

1.10 Klasifikasi Model Simulasi

² Model Simulasi Statik vs. Dinamik

- Model statik: representasi sistem pada waktu tertentu. Waktu tidak berperan di sini. Contoh: model Monte Carlo.

- Model dinamik: merepresentasikan sistem dalam perubahannya terhadap waktu. Contoh: sistem conveyor di pabrik.

² Model Simulasi Deterministik vs. Stokastik

- Model deterministik: tidak memiliki komponen probabilistik (random).

- Model stokastik: memiliki komponen input random, dan menghasilkan output yang random pula.

² Model Simulasi Kontinyu vs. Diskrit

- Model kontinyu: status berubah secara kontinu terhadap waktu, mis. gerakan pesawat terbang.
- Model diskrit: status berubah secara instan pada titik-titik waktu yang terpisah, mis. jumlah customer di bank.

Model yang akan dipelajari selanjutnya adalah diskrit, dinamik, dan stokastik, dan disebut model simulasi (sistem) peristiwa diskrit (discrete-event).

1.11 Simulasi Sistem Peristiwa Diskrit

- ² Pemodelan sistim dimana variabel keadaan berubah pada set waktu yang diskrit.

² Metode: numerik (bukan analitik)

- Analitik: alasan deduktif secara matematis; akurat
- Numerik: prosedur komputasional; aproksimasi

² Model simulasi di-run (bukan diselesaikan (solved)).

- Observasi sistem riil, entitas, interaksi
- Asumsi model
- Pengumpulan data
- Analisis dan estimasi kinerja sistem

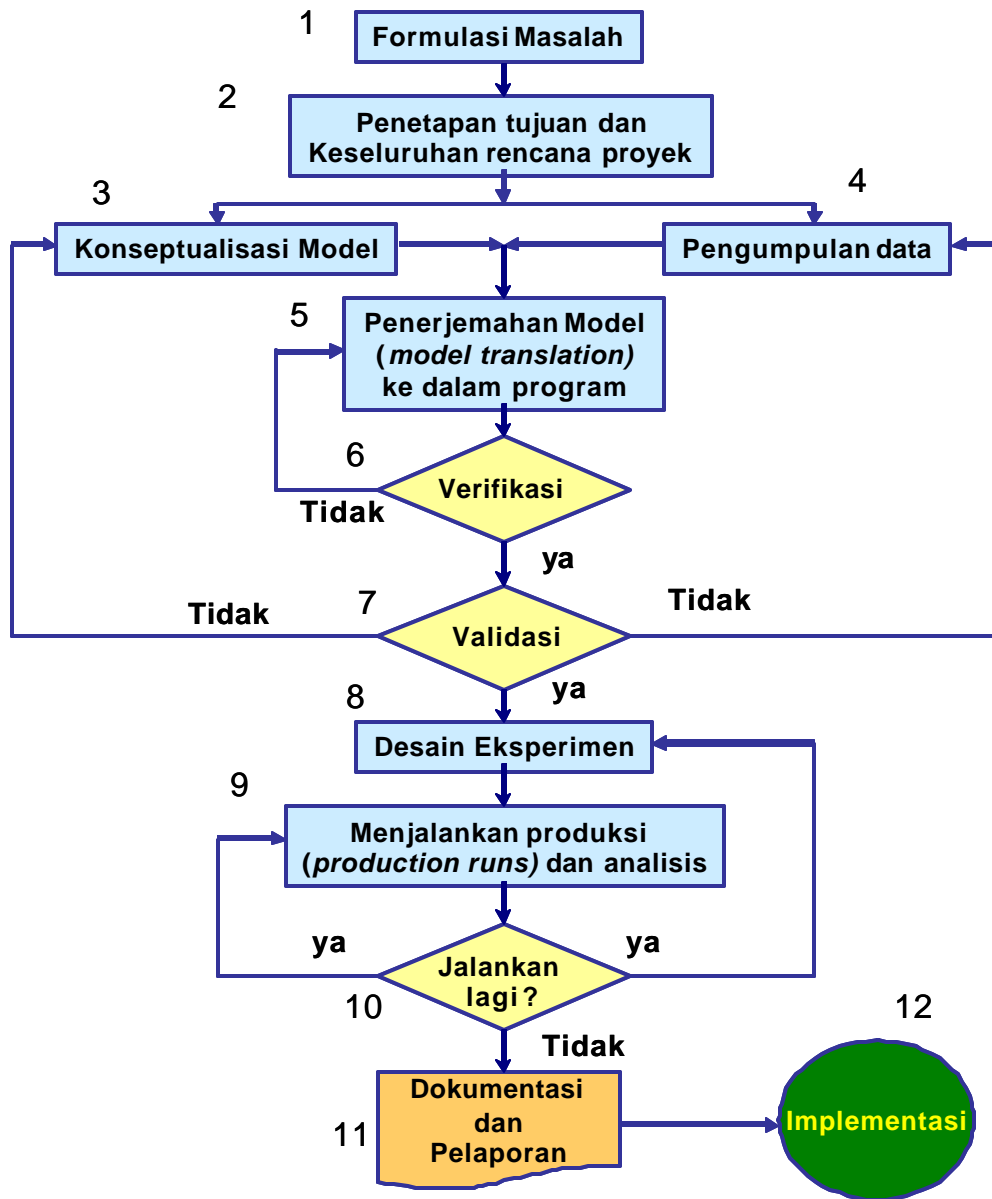
1.12 Langkah-Langkah Studi Simulasi

² Formulasi masalah:

- mengidentifikasi masalah yang akan diselesaikan
- mendeskripsikan operasi sistim dalam term-term obyek dan aktivitas dalam suatu layout ...sik

- mengidentifikasi sistem dalam term-term variabel input (eksogen), dan output (endogen)
 - mengkatagorikan variabel input sebagai decision (controllable) dan parameters (uncontrollable)
 - mendefinisikan pengukuran kinerja sistem (sebagai fungsi dari variabel endogen) dan fungsi obyek (kombinasi beberapa pengukuran)
 - mengembangkan struktur model awal (preliminary)
 - mengembangkan struktur model lebih rinci yang mengidentifikasi seluruh obyek berikut atribut dan interface-nya
- 2 Penetapan tujuan dan rencana proyek: pendekatan yang digunakan untuk menyelesaikan masalah.
- 2 Konseptualisasi model: membangun model yang masuk akal.
- memahami sistem
 - Pendekatan proses (atau pendekatan aliran fisik (physical flow approach)) didasarkan pada tracking flow dari entitas-entitas keseluruhan sistem berikut titik pemrosesan dan aturan keputusan percabangan
 - Pendekatan peristiwa (event) (atau pendekatan perubahan keadaan (state change approach)) didasarkan pada definisi variabel keadaan internal dan events sistem yang mengubahnya, diikuti oleh deskripsi operasi sistem ketika suatu event terjadi
 - konstruksi model
 - definisi obyek, atribut, metode
 - flowchart metode yang relevan
 - pemilihan bahasa implementasi
 - penggunaan random variates dan statistik kinerja
 - coding dan debugging
- 2 Pengumpulan data: mengumpulkan data yang diperlukan untuk merun simulasi (seperti laju kedatangan, proses kedatangan, disiplin layanan, laju pelayanan dsb.).
- observasi langsung dan perekaman manual variabel yang diseleksi(selected)

- time-stamping untuk men-track aliran suatu entitas keseluruhan sistem
 - menyeleksi ukuran sample yang valid secara statistik
 - menyeleksi suatu format data yang dapat diproses oleh komputer
 - analisis statistik untuk menetapkan distribusi dan parameter data acak
 - memutuskan data mana yang dipandang sebagai acak dan yang mana diasumsikan deterministik
- ² Penerjemahan Model: konversi model suatu bahas pemrograman.
- ² Verifikasi: Verifikasi model melalui pengecekan apakah program bekerja dengan baik.
- ² Validasi: Check apakah sistem merepresentasi sistem riil secara akurat.
- ² Desain Eksperimen: Berapa banyak runs? Untuk berapa lama? Jenis variasi masukannya seperti apa ?
- evaluasi statistik output untuk menetapkan beberapa level presisi yang diterima dari pengukuran kinerja
 - analisis terminasi digunakan jika interval waktu riil tertentu akan disimulasikan
 - steady state analysis digunakan jika obyek of interest merupakan rata-rata long-term
 -
- ² Produksi runs dan analisis: running aktual simulasi, mengumpulkan dan menganalisis keluaran.
- ² Jalankan lagi (More runs) ?: mengulangi eksperimen jika perlu.
- ² Dokumentasi dan pelaporan: Dokumen dan laporan hasil
- ² Implementasi



Gambar 1.2: Langkah dalam studi simulasi

1.13 Veri...kasi dan Validasi

- ² Langkah terpenting dalam studi simulasi: validasi
- ² Validasi bukan merupakan tugas tersendiri yang mengikuti pengembangan model, namun merupakan satu kesatuan yang terintegrasi dalam pengembangan model.
- ² Veri...kasi: Apakah kita membangun model yang benar?
 - Apakah model diprorgram secara benar (input parameters dan logical structure)?
- ² Validasi:
 - Apakah model merupakan representasi akurat dari sistim riil?
 - Proses interatif dari perbandingan model terhadap sifat sistem aktual dan memperbaiki model.

1.14 Pembangunan Model

Proses iteratif yang mengandung tiga langkah utama:

- ² Observasi sistim riil dan interaksi komponen dan pengumpulan data
 - Domain pengetahuan tertentu
 - Stakeholders: operator, teknisi, engineers
- ² Konstruksi model konseptual
 - Asumsi dan hipotesa komponen dan nilai-nilai parameter
 - Struktur sistim
- ² Penerjemahan model operasional ke bentuk yang dikenal oleh komputer

1.15 Kelebihan, Kekurangan, 'Pitfalls' dari Simulasi

1.15.1 Kelebihan

- ² Sebagian besar sistem riil dengan elemen-elemen stokastik tidak dapat dideskripsikan secara akurat dengan model matematik yang dievaluasi secara analitik. Dengan demikian simulasi seringkali merupakan satu-satunya cara.
- ² Simulasi memungkinkan estimasi kinerja sistem yang ada dengan beberapa kondisi operasi yang berbeda.
- ² Rancangan-rancangan sistem alternatif yang dianjurkan dapat dibandingkan via simulasi untuk mendapatkan yang terbaik.
- ² Pada simulasi bisa dipertahankan kontrol yang lebih baik terhadap kondisi eksperimen.
- ² Simulasi memungkinkan studi sistem dengan kerangka waktu lama dalam waktu yang lebih singkat, atau mempelajari cara kerja rinci dalam waktu yang diperpanjang.

1.15.2 Kekurangan

- ² Setiap langkah percobaan model simulasi stokastik hanya menghasilkan estimasi dari karakteristik sistem yang sebenarnya untuk parameter input tertentu. Model analitik lebih valid.
- ² Model simulasi seringkali mahal dan makan waktu lama untuk dikembangkan.
- ² Output dalam jumlah besar yang dihasilkan dari simulasi biasanya tampak meyakinkan, padahal belum tentu modelnya valid.

1.15.3 Pitfalls

- ² Gagal mengidentifikasi tujuan secara jelas
- ² Desain dan analisis eksperimen simulasi tidak memadai
- ² Pendidikan dan pelatihan yang tidak memadai

1.16 Fitur-...tur software simulasi yang dibutuhkan

- ² Membangkitkan bilangan random dari distribusi probabilitas $U(0,1)$.
- ² Membangkitkan nilai-nilai random dari distribusi probabilitas tertentu, mis. eksponensial.
- ² Memajukan waktu simulasi.
- ² Menentukan event berikutnya dari daftar event dan memberikan kontrol ke blok kode yang benar.
- ² Menambah atau menghapus record pada list.
- ² Mengumpulkan dan menganalisa data.
- ² Melaporkan hasil.
- ² Mendeteksi kondisi error.

Bab 2

Contoh-Contoh Simulasi (Kuliah 1-2)

Bahan Bacaan:

- ² [1] J. Banks, et.al., "Discrete Event System Simulation", 3ed., Prent-Hall (Chap.2)
- ² [2] Law & Kelton, et.al., "Simulation Modeling & Analysis", Mc. Graw-Hill Inc., Singapore (Chap.1)

2.1 Langkah-Langkah Dasar

- ² Menetapkan karakteristik masukan.
 - Biasanya dimodelkan sebagai distribusi probabilitas
- ² Menkonstruksi tabel simulasi.
 - Spesi...kasi masalah
 - Biasanya terdiri dari sekumpulan masukan dan lebih dari satu respon
 - Pengulangan
- ² Membangkitkan nilai secara berulanag untuk setiap masukan dan mengeval-uasi fungsi.

	Masukan							Respon
Pengulangan	$X_{i;1}$	$X_{i;2}$	\vdots	$X_{i;j}$	\vdots	\vdots	$X_{i;p}$	y_i
1								
2								
3								
\vdots								
\vdots								
n								

Tabel 2.1: Tabel Simulasi

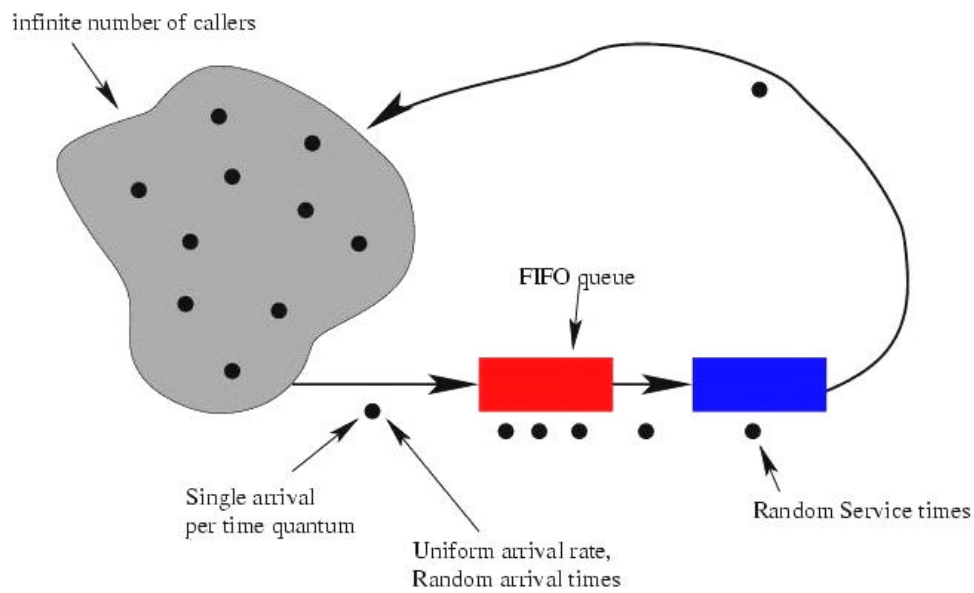
2.2 Simulasi Sistem Antrian

Sistem antrina terdiri dari:

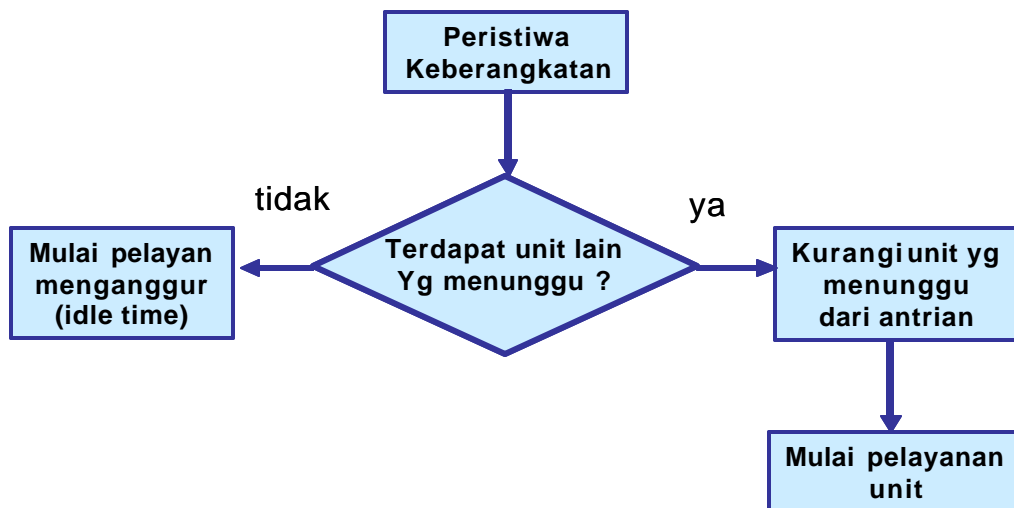
- ² Pemanggilan populasi (Calling population): Biasa tidak terbatas: jika sebuah unit keluar, tidak ada perubahan pada laju ketibaan/kedatangan.
- ² Kedatangan/ketibaan: terjadi secara acak.
- ² Mekanisme pelayanan: Sebuah unit akan dilayani dalam panjang waktu yang acak berdasarkan suatu distribusi probabilitas.
- ² Kapasitas sistem: tidak ada batasan
- ² Displin antrian
 - Urutan layanan, misal, FIFO.

2.2.1 Sistem Antrian

- ² Kedatangan dan pelayanan didefinisikan melalui distribusi probabilitas waktu antara kedatangan dan distribusi waktu pelayanan.
- ² Laju pelayanan vs. laju kedatangan: tidak stabil atau eksplosif
- ² Keadaan: jumlah unit dalam sistem dan status dari pelayan
- ² Peristiwa: Stimulan yang menyebabkan keadaan sistem berubah.
- ² Clock simulasi: Trace waktu simulasi.



Gambar 2.1: Sistem Antrian



Gambar 2.2: Diagram Aliran Layan yang telah selesai

		Status Antrian	
status pelayan	sibuk	tidak kosong antri	kosong antri
	idle	tidak mungkin	masuk layanan

Tabel 2.2: Aksi-Aksi Potensial saat kedatangan



Gambar 2.3: Diagram Aliran unit memasuki sistem

		Status Antrian	
keluaran pelayan	sibuk	tidak kosong ya	kosong tidak mungkin
	idle	tidak mungkin	ya

Tabel 2.3: Keluaran (outcomes) Pelayan setelah layanan selesai

2.2.2 Keacakan dalam simulasi

² Contoh aplikasi:

- Waktu pelayanan
- Waktu antar kedatangan

² Bilangan acak: terdistribusi secara uniform dalam interval (0,1)

² Digit acak: terdistribusi secara uniform pada himpunan {0; 1; 2; ...; 9}

² Bilangan acak yang sebenarnya sangat sulit dibuat:

- Bilangan acak bayangan (pseudo-random numbers)
- Membangkitkan bilangan acak dari tabel digit acak.

² Pembangkitan suatu distribusi sederhana

Pseudo-code:

```
int service_time( void )
```

```
    r = rand()/RAND_MAX /* pseudo random on (0,1) */
```

Waktu Layanan (menit)	Probabilitas	Probabilitas Kumulatif
1	0.10	0.10
2	0.20	0.30
3	0.30	0.60
4	0.25	0.85
5	0.10	0.95
6	0.05	1.00

Tabel 2.4: Contoh hasil pembangkitan distribusi yang sederhana

```

if( r < .1 ) return 1;
  else if( r < .3 ) return 2;
  else if( r < .6 ) return 3;
  else if( r < .85 ) return 4;
  else if( r < .95 ) return 5;

```

```

return 6;

```

2.3 Sistem Antrian Layanan Tunggal

- 2 Entitas apa ? Keadaan apa ? Persitiwa seperti apa ?
- 2 Kapan peristiwa terjadi atau bagaimana memodelkan peristiwa ?
- 2 Bagaimana keadaan berubah ketika peristiwa terjadi?

Contoh: Simulasi kedatangan, pelayanan 20 customer
Statistik dan analisis contoh sistem antrian tunggal

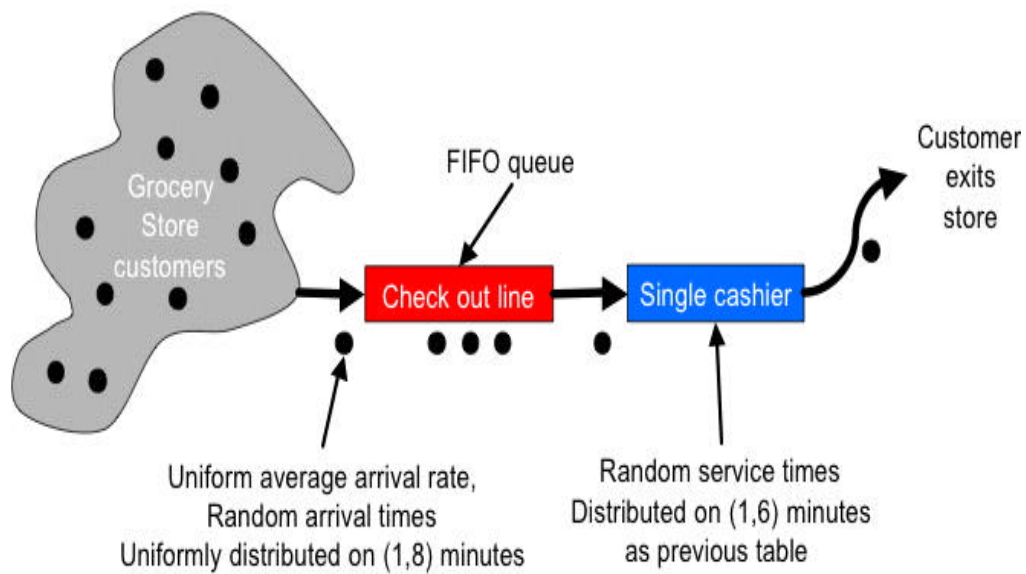
- 2 Rata-rat waktu tunggu = (Total waktu tunggu customer)/(total jumlah customers)

$$= 56/20 = 2.8 \text{ menit}$$

- 2 Probabilitas customer yang harus menunggu di antrian = P(menunggu)

$$P(\text{menunggu}) = (\text{jumlah customer yang menunggu}) / (\text{total jumlah customers})$$

$$P(\text{menunggu}) = 13/20 = 0.65$$



Gambar 2.4: Sistem Antrian Pelayan Tunggal

Customer	Random Digits	Time between Arrivals (minutes)	Customer	Random Digits	Time between arrivals (minutes)
1	-	-	11	109	1
2	913	8	12	093	1
3	727	6	13	607	5
4	015	1	14	738	6
5	948	8	15	359	3
6	309	3	16	888	8
7	922	8	17	106	1
8	753	7	18	212	2
9	235	2	19	493	4
10	302	3	20	535	5

Gambar 2.5: Penentuan waktu antar ketibaan

Customer	Time Since Last Arrival	Arrival Time	Service Time Lookup	Time Service Begins	Time Service Ends	Time Customer Waits in Queue	Time Customer Spends in System (minutes)	Idle Time of Server (minutes)	Customer Wait?
1		0	4	0	4	0	4	0	0
2	8	8	1	8	9	0	1	4	0
3	6	14	4	14	18	0	4	5	0
4	1	15	3	18	21	3	6	0	1
5	8	23	2	23	25	0	2	2	0
6	3	26	4	26	30	0	4	1	0
7	8	34	5	34	39	0	5	4	0
8	7	41	4	41	45	0	4	2	0
9	2	43	5	45	50	2	7	0	1
10	3	46	3	50	53	4	7	0	1
11	1	47	3	53	56	6	9	0	1
12	1	48	5	56	61	8	13	0	1
13	5	53	4	61	65	8	12	0	1
14	6	59	1	65	66	6	7	0	1
15	3	62	5	66	71	4	9	0	1
16	8	70	4	71	75	1	5	0	1
17	1	71	3	75	78	4	7	0	1
18	2	73	3	78	81	5	8	0	1
19	4	77	2	81	83	4	6	0	1
20	5	82	3	83	86	1	4	0	1
			68			56	124	18	13

Gambar 2.6: Hasil Simulasi

2 Fraksi waktu menganggur pelayan:

$$P(\text{idle}) = (\text{total waktu idle}) / (\text{total waktu run simulasi})$$

$$P(\text{idle}) = 18/86 = 0.21$$

2 Rata-rata waktu pelayanan = (total waktu pelayanan) / (total jumlah customers)

$$= 68/20 = 3.4 \text{ menit}$$

2 Waktu layanan yang diharapkan (expectation):

$$E(S) = \sum (\text{waktu layanan}) \times p(\text{waktu layanan})$$

$$= 1 \times (0:10) + 2 \times (0:20) + 3 \times (0:30) + 4 \times (0:25) + 5 \times (0:10) + 6 \times (0:05) = 3.2 \text{ menit}$$

Waktu layanan yang diharapkan lebih kecil ketimbang hasil simulasi. Semakin lama waktu simulasi akan semakin dekat ke nilai ekspektasi $E(S)$.

2 Rata-rata waktu antar kedatangan = (jumlah seluruh waktu antar kedatangan) / (jumlah ketibaan - 1)

$$82=19 = 4:3$$

- ² Waktu yang diharapkan antara kedatangan (mean distribusi uniform diskrit, yang memiliki titik ujung $a = 1; b = 8$).

$$E(A) = (1 + 8)/2 = 4.5 \text{ menit}$$

Rata-rata nilai ekspektasi berbeda

- ² Rata-rata waktu tunggu bagi pelanggan yang menunggu:

(total waktu customers menunggu di antrian) / (total customer yang menunggu)

$$= 56/13 = 4.3 \text{ menit}$$

- ² Rata-rata waktu berada di sistem:

(total waktu customers berada di sistem) / (total jumlah customers)

$$= 124/20 = 6.2 \text{ menit}$$

$$(\text{waktu rata di antrian}) + (\text{waktu rata dalam pelayanan}) = 2.8 + 3.4 = 6.2 \text{ menit}$$

2.4 Contoh-Contoh Lain

2.4.1 Masalah Able Baker Carhop: Dua Pelayan.

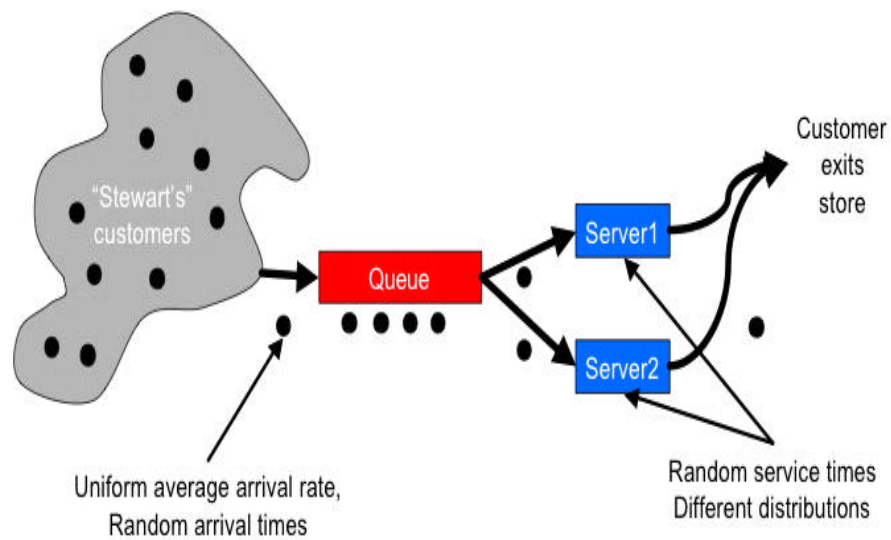
Able kemampuan kerjanya lebih baik dan bekerja lebih cepat ketimbang Baker. Penyederhanaan aturan (rule) — Able mendapat customer jika kedua carhops menganggur.

Waktu antar kedatangan (mnt)	Probabilitas
1	0.25
2	0.40
3	0.20
4	0.15

Tabel 2.5: Distribusi waktu antar ketibaan

Analisis hasil simulasi:

Able sibuk 90% dari waktu yang ada. Baker sibuk hanya 69 %. Sembilan dari 26 kedatangan (35%) harus menunggu. Rata-rata waktu tunggu untuk



Gambar 2.7: Sistem Antrian Dua Pelayan

Waktu layanan (menit)	Probabilitas
2	0.30
3	0.28
4	0.25
5	0.17

Tabel 2.6: Distribusi waktu pelayanan dari Able

Waktu layanan (menit)	Probabilitas
2	0.35
3	0.25
4	0.20
5	0.20

Tabel 2.7: Distribusi waktu pelayanan dari Baker

Customer Digits for Arrival	Time Between Arrivals	Clock Time of Arrival	Able			Baker			Time in Queue
			Time Service Begins	Service Time	Time Service Ends	Time Service Begins	Service Time	Time Service Ends	
1		0	0	5		5			0
2	2	2					3	5	0
3	4	6	6	3	9	9			0
4	4	10	10	5	15	15			0
5	2	12					6	18	0
6	2	14	15	3	18	18			1
7	3	17	18	2	20	20			0
8	3	20	20	4	24	24			0
9	3	23					4	27	0
10	1	24	24	3	27	27			0
11	2	26	27	3	30	30			1
12	2	28					4	32	0
13	2	30	30	5	35	35			0
14	1	31					3	35	1
15	2	33	35	4	39	39			2
16	2	35					4	39	0
17	2	37	39	4	43	43			2
18	3	40					5	45	0
19	2	42	43	2	45	45			1
20	2	44	45	4	49	49			1
21	4	48					3	51	0
22	1	49	49	3	52	52			0
23	2	51					5	56	0
24	3	54	54	3	57	57			0
25	1	55					6	62	1
26	4	59	59	3	62	62			0
			56			43			11

Gambar 2.8: Hasil simulasi sistem antrian dua pelayan

seluruh hanya sekitar 0.42 menit (25 detik), sangat kecil. Kesembilan customer yang harus menunggu, hanya menunggu rata-rata 1,22 menit, cukup rendah. Kesimpulan, sistem ini tampak seimbang dengan baik. Satu pelayan tidak cukup, tiga pelayan mungkin akan terlalu banyak.

2.4.2 Sistem Inventory

- Simulasikan sistem inventory (M; N).

2.4.3 Masalah Reabilitas

- Evaluasi alternatif

2.4.4 Masalah Militer

- Bilangan normal acak

2.4.5 Lead-Time Demand

- Histogram

2.5 Ringkasan

² Konsep Dasar Simulasi:

- Menetapkan karakteristik data masukan.
- Mengkonstruksi tabel simulasi.
- Membangkitkan variabel acak berdasarkan model masukan dan menghitung nilai respon.
- Menganalisis hasil-hasil.

² Masalah utama dengan pendekatan tabel simulasi:

- Tidak dapat digunakan atau mengatasi ketergantungan yang kompleks antar entitas.

Bab 3

Prinsip Umum SSPD (Kuliah 3)

Bahasan:

- 2 Konsep dan Definisi Simulasi Sistem Peristiwa Diskrit (SSPD)
- 2 Mekanisme Eksekusi Simulasi Sistem Peristiwa Diskrit (SSPD)
- 2 Pendekatan untuk SSPD

TIU:

- 2 Mahasiswa memahami prinsip umum SSPD

TIK:

- 2 Mahasiswa mampu menjelaskan konsep dan definisi dalam SSPD
- 2 Mahasiswa mampu menjelaskan mekanisme eksekusi SSPD
- 2 Mahasiswa mengerti dan dapat menggunakan pendekatan untuk SSPD

Deskripsi Singkat:

- 2 Pada bab ini akan dibahas kerangka kerja umum untuk memodelkan sistem yang kompleks dengan menggunakan simulasi peristiwa diskrit. Didiskusikan blok pembangun dasar simulasi peristiwa diskrit: entitas dan atribut, aktivitas dan peristiwa, keadaan

Bahan Bacaan:

- ² [1] J. Banks, et.al., "Discrete Event System Simulation", 3ed., Prent-Hall (Chap.3)
- ² [2] Law & Kelton, et.al., "Simulation Modeling & Analysis", Mc. Graw-Hill Inc., Singapore (Chap.1)

3.1 Konsep dan Definisi

² Sistem:

- Sekumpulan entitas (misal orang dan mesin) yang berinteraksi satu sama lain dalam waktu tertentu untuk mencapai tujuan tertentu

² Model:

- Suatu representasi abstrak dari suatu sistem, biasanya mengandung hubungan struktural, logikal, atau matematis yang menggambarkan suatu sistem dari segi keadaan, entitas, atribut, proses, peristiwa, aktivitas dan delay

² Keadaan sistem:

- Kumpulan variabel yang mengandung seluruh informasi penting untuk menggambarkan sistem setiap saat. Contoh Carhop Able-Baker (Able-sibuk, Baker-sibuk, jumlah mobil yang menunggu).

² Entitas:

- Setiap objek atau komponen dalam sistem yang memerlukan representasi eksplisit dalam model.

² Atribut:

- Sifat-sifat dari entitas.

² List:

- Kumpulan entitas yang terkait, diurut menurut gambaran logis.

² Event:

- Suatu kejadian sesaat yang mengubah keadaan sistem (seperti kedatangan atau keberangkatan seorang customer).

² Event notice:

- Rekord dari sebuah peristiwa yang akan terjadi saat ini atau pada waktu berikutnya, bersamaan dengan informasi penting yang terkait untuk mengeksekusi peristiwa. Minimal, rekord mengandung jenis peristiwa dan waktu peristiwa. Misal pada simulasi operasi airport, kita dapat mempunyai dua jenis peristiwa, take-off dan landing. Dengan dua peristiwa ini, suatu event notice yang mungkin memiliki bentuk sbb:.

- ✧ Jenis peristiwa (mis. landing atau take-off)
- ✧ Waktu peristiwa (mis. 134)
- ✧ Nomor penerbangan
- ✧ Jenis pesawat (mis. Boeing 737-200, DC-10)
- ✧ Jumlah penumpang dalam penerbangan (mis. 125)
- ✧ Pointer ke informasi penerbangan lain
- ✧ Pointer ke spesifikasi pesawat
- ✧ Pointer ke informasi crew

² Event list:

- Suatu daftar event notices untuk peristiwa yang akan datang, diurut menurut waktu kejadian; biasa disebut sebagai future event list (FEL).

² Aktivitas:

- Durasi waktu dengan panjang tertentu, yang diketahui ketika suatu peristiwa di mulai.
- Catatan bahwa term waktu tidak harus selalu pembacaan jam, bisa saja berupa suatu proses.
Misal take-off time: sebuah pesawat akan take-off dalam 3 menit setelah mesin dihidupkan.

² Delay:

- Durasi waktu dengan panjang tidak dispesifikasi dan didenisikan, yang tidak diketahui sampai akhir peristiwa.
Contoh delay customer pada antrian LIFO yang tergantung pada kedatangan berikutnya, sejak dari awal antrian (contoh. procedure call stack).

² Clock:

- Suatu variabel yang merepresentasikan waktu simulasi. Variabel clock dapat terpusat atau terdistribusi

3.2 Time in Simulation

Pada suatu simulasi terkait dua notasi waktu:

- simulation time – waktu pada clock simulasi – waktu virtual dalam dunia simulasi
- run time – lama waktu prosesor yang dikonsumsi

Kita memerlukan run times sekecil mungkin untuk memperoleh suatu hasil dalam kerangka sumber daya yang tersedia.

Akan tetapi, waktu simulasi lebih penting jika dipandang dari segi hasil dan bagaimana simulasi tersebut diorganisasi.

- event time: simulation time dimana sebuah peristiwa (event) terjadi

Teknik-teknik mengubah waktu pada clock simulasi:

- fixed time increment – time slicing – periodic scan
- variable time increment – event scan

3.3 Algoritma Umum

3.3.1 Eksekutif Simulasi Sinkron

Algoritma umum:

```
while simulation time not at end
    increment time by predetermined unit
    if events occurred during time interval
        simulate those events
```

Batasan-batasan:

- Peristiwa (event) dieksekusi pada akhir interval, bukan pada waktu presisi tertentu
- Beberapa interval bisa saja bukan peristiwa (events)

3.3.2 Eksekutif Event-Scanning

Algoritma umum:

```
while event list not empty and simulation time not at end
    get unsimulated event with earliest time
    advance simulation clock to time of event
    simulate the event
```

Baik teknik sinkron maupun event-scanning mensimulasi paralelisme proses-proses.

3.4 Mekanisme Eksekusi SSPD

Operasional, suatu SPD merupakan urutan kronologi yang nondecreasing dari kejadian-kejadian peristiwa (event occurrences).

event record: pasangan event dengan waktu.

future event list (FEL): sebuah daftar yang diurut menurut waktu simulasi yang nondecreasing

event (list) driven simulation: simulasi dimana waktu ditambahkan ke waktu dimana event record (dari event list) pertama dilaksanakan

Persyaratan pendukung SPD:

- memelihara future event list
- enable event record creation dan insertion into dan deletion dari event list
- memelihara clock simulasi
- menyediakan utilitas untuk membangkitkan bilangan acak dari distribusi probabilitas

Bagaimana sebuah tipikal SSPD dieksekusi? Untuk menggambarkan proses ini, kita akan menggunakan sebuah contoh: simulasi airport (take-off dan landing) dengan satu landasan pacu (run way).

1. Persitiwa-peristiwa yang mungkin: landing-request, landing, landing-complete, take-off-request, take-off, take-off-complete, idle.

2. Keadaan awal simulasi di-set.
 - ² Set runway menjadi idle.
 - ² No landing atau take-off terjadi.
 - ² Clock simulasi adalah nol.
 - ² Landing-request pertama dijadual pada waktu 3; take-off-request pertama dijadualkan pada waktu 5.
3. Pada moment ini, FEL memiliki dua event notices, landing-request pada 3 dan take-off-request pada 5.
4. Baik landing maupun take-off berlangsung 3 menit.
5. Mengambil event notice pertama dari FEL, memprosesnya (peristiwa landing-request). Pemrosesan event notice biasanya menyertakan aktivitas pemrograman yang terkait dengan aplikasi-aplikasi. Contoh,
 - ² Set run way menjadi sibuk sehingga tidak ada landing dan take-off lainnya berlangsung.
 - ² Bangkitkan event notice berikut dengan tipe yang sama (landing-request). Waktu dari peristiwa berikut ditentukan baik melalui interval yang ...x atau melalui pembangkitan bilangan acak. Asumsikan waktu adalah 4. Event notice baru ini dimasukkan ke FEL, sebelum event notice kedua dari take-off pada saat 5!
 - ² Bangkitkan event notice landing-complete pada waktu 6. Masukkan ke dalam FEL
 - ² Kumpulkan informasi: berapa penumpang dalam penerbangan, nomor penerbangan dll.
6. Ambil event notice berikut. Pada saat ini, landing-request lain pada waktu 4. Tetapi runway sibuk. Sehingga kita harus meletakkan event notice ini ke antrian tunggu untuk runway. (Catatan bahwa kita tidak meletakkan event notice ini kembali ke FEL dalam hal ini.)
7. Ambil event notice berikut, yaitu sebuah take-off-request event pada waktu 5. Runway masih sibuk. Letak event ini ke antrian tunggu untuk runway.
8. Ambil event notice berikut, yaitu sebuah landing-complete event pada waktu 6. Pemrosesan event ini:

- 2 Set the runway menjadi bebas.
 - 2 Updating statistik (yaitu.landing lain selesai).
 - 2 Check apakah terdapat airplanes yang menunggu di antrian runway. Jika ya, ambil event notice dari antrian tunggu dan memprosesnya. Pada contoh ini, event notice pertama dari antrian tunggu adalah sebuah landing-request, waktunya adalah 4. Waktu event aktual adalah 6. Pada saat pemrosesan event ini, sebuah landing-complete event pada waktu 9 dimasukkan ke FEL.
9. Proses ini diulang sampai kondisi yang telah ditetapkan dijumpai seperti total waktu simulasi dicapai, atau total jumlah landing, take-off dicapai.
 10. Setiap kali sebuah event notice diproses, nilai CLOCK diset menjadi nilai waktu event. Hal ini biasa disebut simulation clock, atau waktu simulasi.
 11. Metode pembangkitan event berikut pada saat pemrosesan sebuah event saat ini dengan tipe yang sama biasa disebut bootstrapping.
 12. Events yang muncul dalam FEL disebut events utama. Yang lainnya disebut events kondisional seperti event landing dan take-off, yang tidak muncul dalam FEL.
 13. Cara lain yang mungkin untuk membangkitkan events utama adalah pertukaran suatu keadaan. Contoh, airport mungkin "shut down" secara acak/kebetulan. Selain itu ia beroperasi normal. Untuk mensimulasi fakta ini, kita dapat menjadwalkan suatu "end-of-normal event" untuk waktu yang akan datang. Jika waktu tersebut dicapai, airport menjadi shut-down. Pada saat pemrosesan "end-of-normal event", sesuatu "end-of-shut-down event" harus dibangkitkan pada waktu berikut.

3.5 Pendekatan-Pendekatan dalam SSPD

Bagaimana menggambarkan sebuah simulasi, atau dari sudut pandang mana mengamati sebuah simulasi? Terdapat beberapa mode-mode populer, dan biasa disebut "world views"/pendekatan umum.

3.5.1 Pendekatan event-scheduling

Kita memandang simulasi sebagai urutan peristiwa terjadual menurut waktunya. Simulasi diproses menurut urutan snap-shots dari sistim. Setiap snap-shot dipicu oleh sebuah peristiwa dari event list.

Hanya satu snap-shot dipertahankan dalam memori komputer. Suatu snap-shot baru dapat di-derived hanya dari snap-shot sebelumnya, nilai-nilai variable acak terbaru dibangkitkan, dan logika peristiwa. snap-shots yang lalu diabaikan pada saat pertambahan clock. Snap-shot saat ini harus memuat seluruh informasi penting untuk melanjutkan simulation.

Karakterik pendekatan event-scheduling:

- Blok pembangun dasar adalah event
- Segmen kode program model terdiri dari rutin-rutin event yang menunggu untuk dieksekusi
- Rutin event terkait dengan setiap jenis event – melakukan operasi yang diperlukan untuk jenis-jenis tersebut
- Eksekutif simulasi bergerak dari event ke event yang mengeksekusi rutin event yang terkait

3.5.2 Pendekatan process-interaction

Pada pendekatan ini, simulasi dianggap sebagai kumpulan interaksi diantara proses-proses. Hal ini mirip dengan paradigma pemrograman berbasis obyek. Proses-proses berinteraksi dengan yang lain melalui pesan-pesan.

Lihat gambar 3.4 pada hal. 69 [1] sebagai contoh. Dari gambaran ini, dua proses berinteraksi satu dengan yang lainnya.

Kadang paket simulasi tertentu mendukung gambaran ini. Paket simulasi ini memperhatikan "time advancing issues" bagi programmer. Pemrogramman pada bahasa tingkat tinggi general purpose sulit menggunakan gambaran process-interaction, karena proses ini cukup rumit bagi para programmer untuk menspesifikasi secara detail seluruhnya.

Karakteristik pendekatan ini:

- Blok pembangun dasar adalah proses
- Sistem terdiri dari kumpulan proses yang berinteraksi
- Kode program model untuk setiap proses memuat operasi yang berlangsung selama waktu hidupnya (lifetime)
- Urutan event sistem terdiri dari penggabungan urutan-urutan untuk seluruh proses

- Future event list terdiri dari urutan node-node event (atau notices)
- Setiap node event mengindikasikan event time dan proses yang terkaitnya
- Eksekutif simulasi melakukan tugas-tugas berikut:
 - menempatkan proses-proses pada titik waktu tertentu ke dalam list
 - membuang proses-proses dari event list
 - mengaktifasi proses yang terkait dengan event node berikut dari event list
 - rescheduling proses-prose di event list
- Tipikal, obyek proses dapat berada di salah satu dari beberapa keadaan:
 - active – proses yang sedang dieksekusi
Hanya ada satu proses yang demikian dalam suatu sistem.
 - ready – proses di event list, menunggu untuk aktivasi pada waktu tertentu
 - idle (blocked) – proses tidak ada di event list, tapi eligible untuk di-reactivated oleh beberapa entitas lain
 - terminated – proses telah menyelesaikan uruttan aksinya, tidak lagi berada di event list, dan tidak dapat di-reactivated

3.5.3 Pendekatan activity-scanning

Dengan pendekatan activity-scanning, seorang modeler terkonsentrasi pada aktivitas sebuah model dan kondisi-kondisi yang memungkinkan sebuah aktivitas dimulai. Pada setiap penambahan clock, kondisi setiap aktivitas di-check dan jika kondisi adalah benar, aktivitas yang terkait dimulai. Sebagai contoh lihat gambar 3.4 pada hal 69 [1].

Karakteristik pendekatan ini:

- Blok dasar pembangun adalah aktivitas
- Segmen kode program model terdiri dari urutan aktivitas-aktivitas (operasi-operasi) yang menunggu untuk dieksekusi
- Kondisi urutan aktivitas harus dipenuhi sebelum dapat dieksekusi
- Eksekutif simulasi bergerak dari event ke event yang mengeksekusi urutan-urutan aktivitas tersebut yang kondisinya terpenuhi

3.6 Contoh-Contoh lain

3.6.1 Contoh 3.1: Able and Baker, versi revisi.

² Keadaan sistim:

- $L_Q(t)$: jumlah mobil yang menunggu untuk dilayani pada saat t .
- $L_A(t)$: variabel boolean yang menunjukkan Able idle atau sibuk pada waktu t .
- $L_B(t)$: variabel boolean yang menunjukkan Baker idle atau sibuk pada waktu t .

² Entitas: mobil dan dua pelayan.

² Peristiwa (events):

- Peristiwa kedatangan
- Peristiwa layanan selesai oleh Able.
- Peristiwa layanan selesai oleh Baker.

² Aktivitas:

- Waktu antar kedatangan
- Waktu layanan Able
- Waktu layanan Baker

² Delay: waktu tunggu customer di antrian sampai Able atau Baker bebas.

3.6.2 Contoh 3.2: Antrian single-channel (Supermarket check-out counter).

Contoh suatu simulasi event-scheduling, suatu tabel simulasi digunakan untuk me-rekord snap-shots sistem secara suksesif ketika waktu bertambah.

Tabel simulasi lihat table 3.1 pada hal 72 [1]

² Keadaan sistem:

$(L_Q(t); L_S(t))$ dimana $L_Q(t)$ adalah jumlah customer pada baris tunggu, dan $L_S(t)$ jumlah customers dilayani pada saat t .

² Entitas:

Pelayan dan customers.

² Peristiwa:

- Kedatangan (A)
- Keberangkatan (D)
- Stopping event (E), dijadualkan terjadi pada waktu 60.

² Event notices:

- (A; t) menunjukkan sebuah peristiwa kedatangan terjadi pada waktu yang kan datang t.
- (D; t) menunjukkan sebuah peristiwa keberangkatan terjadi pada waktu yang kan datang t.
- (E; 60) menunjukkan stopping event terjadi pada waktu yang kan datang 60.

² Aktivitas:

- Waktu antar kedatangan, Tabel 2.6 hal 28 [1].
- Waktu pelayanan, Tabel 2.7 hal 28 [1].

3.6.3 Contoh 3.4: Simulasi check-out counter, lanjutan

Kelanjutan dari contoh 3.3, kita akan mengumpulkan beberapa statistik, mean waktu respon dan mean bagian customer yang menghabiskan waktu 4 menit atau lebih dalam sistim (waktu dalam sistim termasuk waktu tunggu dan waktu pelayanan).

² mean waktu repson =

$$\frac{\sum_{i=0}^{\infty} \text{Depature}_i - \text{Arrive}_i}{\text{jumlah total pelanggan}}$$

² mean bagian customers yang menghabis waktu 4 menit atau lebih =

$$\frac{\text{Jumlah pelanggan yang menunggu 4 mnt atau ebih}}{\text{jumlah total pelanggan}}$$

3.6.4 Contoh 3.5: Masalah dump truck.

- ² Enam dump trucks mengangkut batubara mulai dari pintu masuk tambang kecil ke railroad.
- ² Setiap truck diisi dengan satu atau dua beban.
- ² Setelah pemuatan, truck segera bergerak ke timbangan sesegera mungkin.
- ² Dua beban dan timbangan memiliki antrian FCFS.
- ² Waktu aktual pemuatan dan penimbangan diabaikan.
- ² Setelah ditimbang, truck mundur dan akan kembali ke baris yang sama untuk memperoleh "more coal". Proses ini berulang.

Model memiliki komponen-komponen berikut.

- ² Keadaan sistim: $[L_Q(t); L(t); w_Q(t); w(t)]$
 - $L_Q(t)$ = jumlah trucks pada antrian pemuatan
 - $L(t)$ = jumlah trucks pada pemuatan (0, 1, 2)
 - $w_Q(t)$ = jumlah truck pada antrian tunggu timbangan.
 - $w(t)$ = jumlah truck yang ditimbang (0, atau 1).
- ² Event notices
 - $(AL_Q; t; DT_i)$ dump truck i tiba di antrian pemuatan
 - $(EL; t; DT_i)$ dump truck i mengakhiri pemuatan EL pada saat t .
 - $(EW; t; DT_i)$ dump truck i mengakhiri penimbangan pada saat t .
- ² Entitas: enam dump trucks ($DT_1; DT_2; \dots; DT_6$)
- ² Lists: antrian pemuatan dan antrian timbangan, keduanya FIFO.
- ² Aktivitas: Waktu pemuatan, waktu penimbangan, dan waktu perjalanan.
- ² Delay: Delay pada antrian pemuatan, dan antrian timbangan.
- ² Lihat tabel 3.6 [1], tabel simulasi operasi dump truck

Bab 4

Bahasa-Bahasa Simulasi

As we saw in the previous chapters, computer simulation is essentially an experimental technique, used for studying a wide variety of problems. The abstract model of the system being simulated takes the form of a computer program, and the behavior is given by the output, as the program runs. A programmer who has to perform simulation frequently would be better off learning a higher-level special purpose language, which facilitates simulation programming.

Unfortunately there is a bewildering variety of simulation languages. But there is no single language which can be termed as the "best", "most useful", or "universally" available. Most of these languages are suited for narrow class of applications. In addition to the nature of a system being simulated, the availability of the hardware and the software also dictates the choice of a language.

4.1 Bahasa Simulasi Kontinyu dan Diskrit

As discussed earlier, simulation is divided into two categories: discrete and continuous. Accordingly, most of the simulation languages also fall into one of the two classes. Continuous simulation languages are designed for simulating continuous models and discrete simulation languages are designed for discrete models. A few languages have been designed which are suitable for both discrete as well as continuous models.

4.2 Bahasa Simulasi Kontinyu

Before the digital computers came into widespread use, analog computers were being used for simulating continuous dynamic systems. The system

being simulated was generally engineering systems, described by differential equations, for which analytic solutions were hard to obtain. As soon as digital computers arrived, some of its advantages (such as greater accuracy, freedom from scaling) over the analog computers become obvious. The continuous simulation languages are divided into two:

- 2 Block-structured simulation languages
- 2 Expression-based languages.

4.3 Bahasa Simulasi Sistem Diskrit.

Although many small as well as large programs for simulating discrete systems have been and being written in general purpose languages, such as FORTRAN, and C, languages designed specially for simulating discrete systems are popular. These languages offer many convenient facilities such as automatic generation of streams of pseudo-random numbers for any desired statistical distribution; automatic data collection; their statistical analysis and report generation; good diagnostics; automatic handling of queues; etc.

Discrete system simulation languages are highly problem-oriented. A language very natural and convenient for simulating one class of discrete system may not be so natural for another class of systems.

Every discrete system simulation language must provide the concepts and statements for

1. representing the state of a system at a single point in time (static modeling)
2. moving a system from state to state (dynamic modeling)
3. performing relevant chores, such as, random number generation, data analyses, and report generation.

Based on the above point of view, discrete system simulation languages can be classified into three main categories.

- 2 event-oriented languages
- 2 activity-oriented languages
- 2 process-oriented languages

4.3.1 Event-oriented languages.

In an event oriented language each event is represented by an instantaneous occurrence in simulated time and must be scheduled to occur (in advance) when a proper set of conditions exists. The state of the system changes at the occurrence of an event. The language in this category are used to model processes that are characterized by a large number of entities. The two well known of this group of languages are SIMSCRIPT and GASP.

4.3.2 Activity-oriented languages.

In an activity oriented language the discrete occurrences are not scheduled in advance. They are created by a program which contains descriptions of conditions under which any activity can take place. These conditions are scanned before each simulation time advance and if all necessary conditions are met , the proper actions are taken. An activity-oriented language should be considered for use if the model has the following characteristics :

- 2 the model uses a next event or variable time increment type of timing
- 2 the process simulated is highly interactive but involves a fixed number of entities with events happening irregularly
- 2 event occurrence is controlled by cyclic scanning activity programs.

An example for this type of language is CSL .

4.3.3 Process-oriented languages.

A key feature of a process orientation is that of a single process routine ,composed of a number of segments describing a sequence of activities. Each segment behaves as an independently controlled program. On receiving control, only the statements composing the segments are executed, and then control is returned. Thus the model is defined as a series of occurrences(called processes) and an example for this type of language is SIMULA.

Transaction-flow oriented language: Transaction-flow oriented languages form a subcategory of process-oriented languages except that the flow of activities passes through specially defined blocks. The system model is represented by a flow chart consisting of the language blocks. The program creates transactions , executes them in the blocks and moves them along the flow chart. Writing a program is reduced to specifying a flow-chart representation . The best known language in this class is GPSS.

4.4 SIMSCRIPT.

SIMSCRIPT was developed by RAND corporation in the early 1960's and was first released in 1962. It has undergone many revisions and improvements, including the development of SIMSCRIPT 1.5. A completely new version SIMSCRIPT II was released by the RAND corporation in 1968. The latest version is SIMSCRIPT II.5, which was released in 1972.

The language is very FORTRAN like in appearance; in fact initially it was implemented with FORTRAN as an intermediate language. SIMSCRIPT II (II.5) can be viewed as a general programming language with extra features for discrete-event simulation. Because of this general power and its FORTRAN base, SIMSCRIPT has been widely implemented and used discrete simulation language. It has been implemented on IBM 7090, 7094, 360,370 series and many others.

- 2 Level 1 of SIMSCRIPT II is comparable to a very simple algorithmic language such as BASIC. An inexperienced user would appreciate such features as : format free data and ease of programming, simplified output options and automatic mode conversion.

- 2 For example consider, the following simple statements in SIMSCRIPT II.5

```
2 READ X, Y AND N
2 PRINT 1 LINE WITH X, X/Y, N**2 AS FOLLOWS
2 X = **. *, Y= **. *, X/Y= **. *, N SQUARED = ***
```

- 2 The first statement reads the values of the three variables X, Y, and N. The second statement evaluates X, Y, X/Y and N² and display their values.
- 2 Level 2 of SIMSCRIPT II.5, provides additional facilities in the area of data structures. The language allows construction of piecemeal arrays(eg; two-dimensional arrays with different number of elements in each row.)and complex tree structures.
- 2 Level 3 provides ALGOL like statements. The first three levels view SIMSCRIPT II as an algebraic procedural language.
- 2 Level 4 gives SIMSCRIPT II a power to define and manipulate entities, attributes and sets. An entity is a program element similar to a subscripted variable. Individual entities have values, called attributes,

which define a particular state of the entity. Attributes are named, not numbered. For example, we may define EMPLOYEE to be an entity and AGE, SALARY as attributes of EMPLOYEE.

Entities can be of two types:

permanent (specified for all time of program execution) and temporary (specified dynamically as the program proceeds). Temporary entities are physically created and destroyed through special statements. Permanent entities have their attributes stored as arrays. Entities may be declared as temporary or permanent in the PREAMBLE of a SIMSCRIPT II program.

SIMSCRIPT II requires a model to be specified only in terms of separate events, and scheduling an event means issuing an "event notice" in the "event set".

Ex; SCHEDULE A DEPARTURE AT TIME.V+8.5

Schedules an event called DEPARTURE to occur at the current simulated time, given by the variable TIME.V, plus 8.5 time units.

SIMSCRIPT II provides eleven functions for generating pseudo-random samples from the following statistical distributions: Uniform, Normal, Poisson, exponential, beta, Erlang, lognormal, binomial, gamma, Weibull, and uniformly distributed integers.

Two statements, ACCUMULATE and TALLY, are used in the PREAMBLE to instruct the compiler to generate automatic data collection and analysis statements at appropriate places in the program.

Example: Program for simulation of single-server queue using SIMSCRIPT II.5

SIMSCRIPT II.5 programs start with a preamble that describes the components of the model. These are :

- (a) processes
- (b) resources
- (c) variables and statistics gathering variables.

The preamble serves a double purpose. Declaring all of the components of the program causes the SIMSCRIPT II.5 compiler to check each component mentioned in the preamble for consistent and correct usage in the program code. The system keywords are written in lower or mixed case and user declared variables in upper case.

Note that, once the two processes are declared, the preamble adds an attribute,

CUST.MEAN.IT, to the CUSTOMER.GENERATOR process and defines it as a real variable.

Resources are a built-in capability in SIMSCRIPT II.5 , so it is only necessary to declare them.

After the processes and resources are declared in the preamble , the program variables are also declared . Note that the keywords " Tally" and " Accumulate " are also declared in the preamble. The Accumulate keyword is the same as the Tally keyword except that it weighs its average with respect to the simulation time which has elapsed.

The Tally keyword specifies that TIME.IN.SYSTEM will be monitored. Whenever an assignment is made to the variable TIME.IN.SYSTEM, statistics will be automatically gathered. The particular statistic that will be kept in this case is the average of all assignments to this variable. The average will be called MEAN.TIME.IN.SYSTEM.

N.Q.SERVER is one of the attributes of the SERVER resource that is automatically generated and managed by the system to keep track of the SERVER resource's queue statistics. MAX.QUEUE is the highest number ever waiting in the queue for service.

N.X.SERVER is an attribute of SERVER resource, automatically generated by the system to keep track of resource utilization statistics.

The program starts executing in the Main section where program variables are initialized, processes created and activated and the simulation is started. After one resource called SERVER is created then one customer generator process is created. Once the simulation has run its course and finished , the report generator is called and the program terminates at the End statement.

The customer process code , illustrates how readable SIMSCRIPT II.5 code helps the reader to understand a model.

First , a local variable called ARRIVAL.TIME is declared and used to note the current simulation time which is taken from the system variable called TIME.V. The customer process then requested for on unit of SERVER resource. If one unit of resource is not available , the customer process blocks and waits for an indefinite time until the resource is available. It then continues to work statement, and , once finish with the resource , the customer process relinquishes it , computes statistics and terminates.

The next part of the program, generates the inter arrival time from the exponential distribution. The last part is the report generator .

4.5 GPSS (General Purpose Simulation System)

GPSS , one of the earliest discrete simulation languages, was developed by Geoffrey Gordon and presented in two papers in 1961 and 1962. The first release of this language was implemented on the IBM 704, 709 and 7090 computers. Since then improved and more powerful versions have been developed and implemented.

GPSS is suited for modeling traffic and queueing systems well. A GPSS programmer does not write a program in the same sense as a SIMSCRIPT programmer does. Instead , he constructs a block diagram- a network of interconnected blocks, each performing a special simulation -oriented function.

Moving through the system of blocks are entities called transactions. Examples of transactions are : customers, message, machine parts, vehicles, etc. Typical blocks are :

- (a) GENERATE, creates transactions
- (b) QUEUE , creates a queue of transactions and maintains certain queueing statistics;
- (c) TABULATE , tabulates the time it took the transactions to reach that point from the time it entered the simulated system
- (d) TERMINATE , destroys transactions and removes them from the system.
- (e) ADVANCE , when transaction enters this block, an action time is computed and added to the current time to produce a block departure time.

Simple mathematical calculations can be carried out with the use of variable statements. Unlike in SIMSCRIPT , there are no elementary mathematical functions in GPSS, such as the trigonometric or logarithmic functions. GPSS can , however , generate a number of basic random variates.

The latest release of GPSS/H is version 2.0 . It added a floating point clock, built- in math functions , and built in random variate generators.

4.5.1 Single-Server Queue Simulation in GPSS/H

In the following fig we can see the GPSS block diagram for single server queue simulation and after that we can see the GPSS program . Note that the program is a translation of the block diagram together with additional definition and control statements.

In ...g , the GENERATE block represents the arrival event, and also denotes the inter arrival times by the specification RVEXPO(1,&1AT).

RVEXPO indicates " random variable, exponentially distributed " , the 1 indicates the random number stream to use, and &1AT indicates that the mean time for the exponential distribution comes from an ampervariable &1AT. Ampervariable have an "&" before them , these are defined as integer or real by control statements INTEGER and REAL.

The next block is a QUEUE with a dummy queue named SYSTIME. It should be noted that the QUEUE block is not needed for queues to form in GPSS/H. This is just an anomaly of the language. The true purpose of the QUEUE block is the commencement of data collection. By placing a QUEUE block at the point that transactions enter the system, and the counter part of the QUEUE block , the DEPART block, at the point that the transactions complete their processing, the response times will be automatically collected. The measure will be associated with the dummy SYSTIME. The purpose of the DEPART block is to signal the end of data collection.

The next QUEUE block begins data collection for the queue before the cashier. This queue is given the name LINE. The customers may or may not have to wait for the cashier. At some time , the customer captures the cashier as indicated by the SEIZE block with the resource given the name CHECKOUT.(single unit resources called facilities are captured by a SEIZE block, multiple unit resources are captured by a STORE block). Once this capture begins, the data collection for the queue ends as represented by the DEPART block, with queue LINE indicated. The transactions delay at the cashier is given by the ADVANCE block. RVNORM indicates " random variable , normally distributed" . Again random number stream 1 is being used, the time for the normal distribution is given by the ampervariable &MEAN , and its standard deviation is given by the ampervariable &STDEV,

Next the customer gives up the use of the facility CHECKOUT with a RELEASE block.(for a multiple unit resource , the analogous block is the RETURN block,) The end of the data collection for response times is indicated by the DEPART block for the dummy queue SYSTIME.

Next there is a TEST block that checks to see if the time in the system , M1 , is greater than or equal to 4 minutes.(Note that M1 is a reserved word in GPSS/H). Thus , if the customer has been in the system four minutes or longer , add one to the counter &COUNT in the BLET (for Block LET) block. If not true , the escape route is to the block with the label TER. That label appears before the TERMINATE block whose purpose is the removal of the transaction from the system. The TERMINATE block has a value "1" indicating that one more customer is added toward the limiting value , or "transaction to go".

There are eleven blocks in this program. The control statements that begin with an "*" are comments, some of which are used for spacing purposes. The control statement SIMULATE tells GPSS/H to conduct a simulation. If this control statement is omitted, the model will be compiled only. The ampers variables are defined as integer or real by control statements INTEGER and REAL. The four LET statements provide data for the simulation.

To insure that the model data is correct, and for the purpose of managing different scenarios simulated, it is good practice to echo the input data. This is accomplished with a PUTPIC(for "put picture") control statement. It says to put the 5 lines that follow in a file with the name OUT and that the data going in to that file is given by the four ampers variables indicated. There are five lines, the last of which is just a blank to separate it from additional information that will be written to the file OUT.

The first output in the parenthesized list is the server utilization.

- ² FR(CHECKOUT)/1000 Indicates that the fractional utilization of the facility CHECKOUT.
- ² QM(LINE) is the maximum value in the queue LINE during the simulation.
- ² QT(SYSTIME) is the average time in the queue SYSTIME.
- ² &COUNT/N(TER) is the number that had a response time of four or more minutes divided by the number of customers that went through the block with label TER, or N(TER).
- ² AC1 is the clock time, whose last value gives the length of the simulation.

4.6 SIMULA (SIMULATION Language)

SIMULA 67 is a true extension of ALGOL 60, and is therefore a general purpose programming language, despite its name. The language was designed to provide simulation facilities without losing advantage of a powerful general purpose language. Just as GPSS and SIMSCRIPT are the two most popular simulation languages in the USA, SIMULA appears to be most popular in Europe. The instructions have the form of ALGOL statements, and therefore it is necessary that the user has some knowledge of ALGOL.

The basic idea behind SIMULA is add to ALGOL the concept of a collection of programs, called processes, conceptually operating in parallel. The processes perform their operation in groups called active phases. A process

carries data and executes action. The operation rule specifies the actions to be taken.

In a SIMULA program, the definition of a process class is accomplished in a block, which is SIMULA's fundamental way for program decomposition. A block is an independent, self-contained part of the program.

SIMULA 67 provides 10 random number procedures for drawing samples from different probability distributions, including uniform, exponential, normal, Poisson, and Erlang. Although there is no automatic collection of statistics in SIMULA 67, a procedure "accum" is available to accumulate the simulated time integral of any variable desired.

4.7 Factors in selection of a discrete system simulation language.

The criteria to select a discrete system simulation language can be grouped as

Language generality and programming power:

The language should be able to handle a wide area of applications. This implies that it should offer strong algorithmic capabilities, as well as, special simulation oriented features.

Ease of use:

Ease of use includes the natural translation of the system under study to a corresponding computer program, reduction in debugging effort, convenient documentation and flexible design of experiments.

Machine efficiency:

Machine efficiency is to a large extent a property of the quality of compiler implementation, which in turn depends on the language structure and content. The two measures of machine efficiency are: execution time and memory requirement.

Availability:

The language must be available and supported on the most important type of computers in order to be useful.

Salient features (both good and bad) of the three languages.

GPSS.

1. restricted to simple queueing problems ;
2. poor computational facilities;
3. inflexible input-output;
4. no language extension possible;
5. easy to learn and use;
6. good debugging facilities; and
7. machine efficiency is often poor since GPSS is an interpretative system.

SIMULA.

- (a) algorithmic capabilities comparable to ALGOL;
- (b) collecting statistical data over the system behavior quite adequate but inferior to SIMSCRIPT
- (c) input-output facilities comparable to SIMSCRIPT
- (d) language extension possibilities allow us to construct special problem-oriented languages;
- (e) SIMULA is a complex language , but once mastered it is easy to use;
- (f) Very good debugging features;
- (g) Good program readability and structuring;
- (h) Possibilities for experimental design comparable to SIMSCRIPT

SIMSCRIPT.

- (a) machine-independent general-purpose simulation language;
- (b) algorithmic capabilities comparable to those of ALGOL or PL/I
- (c) simulation concepts are relatively few and very general
- (d) data collection facilities are excellent
- (e) input-output facilities are good
- (f) security(error detection) is poor
- (g) flexibility for experimental design is good
- (h) machine efficiency is high

Bagian II

Model Matematis dan Statistik

Bab 5

Model Statistik dalam Simulasi

² In this chapter, we discuss some of the most commonly used statistical models.

² Motivation:

- So far we used only the uniformly distributed random numbers. They only ...t a fraction of the real applications.
- We need other kind of statistical models to describe various types of applications.
- These models can be used in other situations than simulation. But we discuss them in the context of simulation.

² There are two major categories of random numbers:discrete and continuous.

² The discrete distributions we will discuss inculde:

1. Bernoulli distribution
2. Binomial distribution
3. Geometric distribution
4. Poisson distribution

² Continuous distributions include:

1. Uniform distribution
2. Exponential distribution
3. Gamma distribution

4. Erlang distribution
5. Normal distribution
6. Weibull distribution
7. Triangle distribution

- ² Note that here we discuss random variables that are discretely or continuously distributed. This doesn't have a direct connection to discrete simulation vs. continuous simulation where the concept is how the simulation clock is ticked.
- ² We will talk about some of the basics of probabilities ...rst. Then we will discuss various distributions. Then we will study the Poisson process, a very important, yet relatively simple distribution.

5.1 Alasan Penggunaan Probabilitas dan Statistik dalam Simulasi

- ² Understand how to model a probabilistic system
- ² Validate the simulation model
- ² Choose the input probability distributions
- ² Generate random samples from these distributions
- ² Perform statistical analyses of simulation output data
- ² Design the simulation experiments
- ² Evaluate & compare alternatives

5.2 Tinjauan Ulang Terminologi dan Konsep

- ² Random variable: A variable that assumes values in a irregular pattern (or no particular pattern).
- ² Discrete random variables: Let X be a random variable. If the number of possible values of X is ...nite, or countably in...nite, X is called a discrete random variable.
- ² Let x_i be all possible values of X , and $p(x_i) = P(X = x_i)$ be the probability that $X = x_i$, then $p(x_i)$ must meet the following conditions.

1. $p(x_i)$ for all i .

$$2. \sum_{i=1}^{\infty} p(x_i) = 1$$

Examples: example 6.1 and 6.2 on p. 186

² Continuous random variables: If the values of X is an interval or a collection of intervals, then X is called a continuous random variable.

For continuous random variable, its probability is represented as

$$P(a \leq X \leq b) = \int_a^b f(x)dx$$

The function $f(x)$ is called the probability density function (pdf) of the random variable X , which has to meet the following condition.

(a) $f(x) \geq 0$ for all x :

(b) $\int_{-\infty}^{\infty} f(x)dx = 1$

(c) $f(x) = 0$ if x is not in R_x .

² Example 6.3 on page 187.

² Cumulative distribution function. The cumulative distribution function (cdf), denoted by $F(x)$, measures the probability that the random variable X assumes a value less than or equal to x , $F(x) = P(X \leq x)$.

$$\begin{aligned} \text{If } X \text{ is discrete, then } F(x) &= \sum_{\text{all } x_i \leq x} p(x_i) \\ \text{If } X \text{ is continuous, the } F(x) &= \int_{-\infty}^x f(x)dx \end{aligned}$$

Some properties of cdf include:

(a) F is a non-decreasing function. If $a < b$ then $F(a) \leq F(b)$.

(b) $\lim_{x \rightarrow \infty} F(x) = 1$.

(c) $\lim_{x \rightarrow -\infty} F(x) = 0$.

² Example: 6.4, 6.5 on page 189.

² Expectation and variance. Expectation essentially is the expected value of a random variable. Variance is a measure how a random variable varies from its expected value.

- For discrete random variables

$$E(x) = \sum_{\text{all } i} x_i p(x_i)$$

- For continuous random variables

$$E(x) = \int_0^{\infty} x f(x) dx$$

- For discrete or continuous random variables, its variance is

$$V(x) = E[(X - E[X])^2]$$

which has an identity

$$V(x) = E(X^2) - [E(X)]^2$$

- A more frequently used practical measure is standard deviation of a random variable, which is expressed as the same units as that of expectation.

$$\sigma = \sqrt{V(X)}$$

Examples: 6.6, 6.7 on page 191.

² The mode. The mode is used to describe most frequently occurred values in discrete random variable, or the maximum value of a continuous random variable.

5.3 Model-Model Statistik yang Berguna

1. Queueing system. This is one of the most often used models. We will study queueing system in great detail later. Queueing system can have a lot of variations that can be used to model many real life situation. We have seen a few examples in earlier chapters.

Queueing systems often include random variables such as service time at various server(s); inter-arrival time for various customer streams; routing distribution of customers.

Example 6.8, 6.9 on page 193. Emphasize on the case where empirical data can produce histogram which shows the trend, often close to a mathematical model.

2. **Inventory system.** This is another model used to simulation storage problems, inventory problems. A typical inventory has three random variables: the number of units demanded per order or per time period; the time between demands; the lead time.
3. **Reliability and maintainability.** Time to failure of a component, of a system.
4. **Limited data.** When complete data is not available, three distributions can be used to model the system, uniform, triangular, and beta distribution.
5. **Other distributions.** A few other distributions can be used to model systems that are not covered above: Bernoulli and binomial distributions are for discrete system; hyperexponential may be useful in other instances.

5.4 Distribusi Variabel Acak Diskrit

Here we are going to study a few discrete random variable distributions.

5.4.1 Bernoulli trials dan distribusi Bernoulli

- ² A Bernoulli trial is an experiment with result of success or failure.
- ² We can use a random variable to model this phenomena. Let $X_j = 1$ if it is a success, $X_j = 0$ if it is a failure.
- ² A consecutive n Bernoulli trials are called a Bernoulli process if
 - the trials are independent of each other;
 - each trial has only two possible outcomes (success or failure, true or false, etc.); and
 - the probability of a success remains constant

- ² The following relations hold.

$$p(x_1; x_2; \dots; x_n) = p_1(x_1) \cdot p_2(x_2) \cdot \dots \cdot p_n(x_n)$$

which means the probability of the result of a sequence of events is equal to the product of the probabilities of each event.

Because the events are independent and the probability remains the same,

$$p_j(x_j) = p(x_j) = \begin{cases} p & x_j = 1; j = 1; 2; \dots; n \\ q & x_j = 0; j = 1; 2; \dots; n \end{cases}$$

- ² Note that the "location" of the p_i 's don't matter. It is the count of p_i 's that is important.
- ² Examples of Bernoulli trials include: a consecutive throwing of a "fair" coin, counting heads and tails; a pass or fail test on a sequence of a particular components of the "same" quality; and others of the similar type.
- ² For one trial, the distribution above is called the Bernoulli distribution. The mean and the variance is as follows.

$$E(X_j) = 0 \cdot q + 1 \cdot p = p$$

$$V(X_j) = E(X_j^2) - (E(X_j))^2 = 0^2 \cdot q + 1^2 \cdot p - p^2 = p(1 - p)$$

5.4.2 Distribusi Binomial

- ² The number of successes in n Bernoulli trials is a random variable, X .
- ² What is the probability that m out of n trials are success?

$$p_{m;n} = p^m q^{n-m}$$

- ² There are

$$\frac{n!}{x!(n-x)!}$$

- ² So the total probability of m successes out of n trials is given by

$$p(x) = \begin{cases} \frac{n!}{x!(n-x)!} p^x q^{n-x} & x = 0; 1; 2; \dots; n \\ 0 & \text{otherwise} \end{cases}$$

- ² Mean and variance: consider the binomial distribution as the sum of n independent Bernoulli trials. Thus

$$E[X] = p + p + \dots + p = np$$

$$V(X) = pq + pq + \dots + pq = npq$$

- ² Example 6.10 on page 198

5.4.3 Distribusi Geometrik

- ² The number of trials needed in a Bernoulli trial to achieve the first success is a random variable that follows the geometric distribution.
- ² The distribution is given by

$$p(x) = \begin{cases} q^{x-1}p & x = 1; 2; \dots \\ 0 & \text{otherwise} \end{cases}$$

- ² The mean is calculated as follows.

$$E(X) = \sum_{x=1}^{\infty} xq^{x-1}p = \sum_{x=1}^{\infty} xq^{x-1}p$$

$$E(X) = p \sum_{x=1}^{\infty} \frac{d}{dq} q^x$$

because the sequence converges, so we can exchange the order of summation and the differentiation.

$$\begin{aligned} E(X) &= p \frac{d}{dx} \sum_{x=0}^{\infty} q^x = p \frac{d}{dq} \frac{1}{1 - q} \\ &= p \frac{1}{(1 - q)^2} = \frac{1}{p} \end{aligned}$$

- ² Variance

$$V(X) = \frac{q}{p^2}$$

- ² Example 6.11 on page 199

5.4.4 Distribusi Poisson

The Poisson distribution has the following pdf

$$p(x) = \begin{cases} \frac{e^{-\lambda} \lambda^x}{x!} & x = 0, 1, 2, \dots \\ 0 & \text{otherwise} \end{cases}$$

where $\lambda > 0$

² Poisson distribution property: mean and variance are the same

$$E(X) = \lambda = V(X)$$

² The cdf of the Poisson distribution

$$F(X) = \sum_{i=0}^X \frac{e^{-\lambda} \lambda^i}{i!}$$

because λ is a constant and the increase rate of $i!$ is much faster than that of λ^i , the distribution is mostly determined by the first a few items.

² Poisson distribution is most useful in Poisson process which is used most frequently in describing such phenomena as telephone call arrivals.

² Example 6.12, 6.13, 6.14 on page 200, page 201

5.5 Distribusi Variabel Acak Kontinyu

Continuous random variables can be used to describe phenomena where the values of a random variable can be any value in an interval: the time to failure, or the length of a broken rod. Seven distributions will be discussed.

5.5.1 Distribusi Uniform

² pdf

$$f(x) = \begin{cases} \frac{1}{b-a} & a \leq x \leq b \\ 0 & \text{otherwise} \end{cases}$$

² cdf:

$$F(x) = \begin{cases} 0 & x < a \\ \frac{x - a}{b - a} & a \leq x \leq b \\ 1 & x > b \end{cases}$$

² mean:

$$E(X) = \frac{a + b}{2}$$

² variance:

$$V(X) = \frac{(b - a)^2}{12}$$

² the interval where X can assume value can be arbitrarily long, but it cannot be infinite.

² Note that

$$P(x_1 < X < x_2) = F(x_2) - F(x_1) = \frac{x_2 - x_1}{b - a}$$

is proportional to the length of the interval, for all x_1 dan x_2 satisfying $a \leq x_1 < x_2 \leq b$.

² Example 6.15 and 6.16 on page 202, 203

5.5.2 Distribusi Eksponensial

Exponential distributed random variable is one of most frequently used distribution in computer simulation. It is widely used in simulations of computer network and others.

² pdf

$$f(x) = \begin{cases} e^{-x} & x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

² cdf

$$F(x) = \begin{cases} 1 - e^{-x} & x \geq 0 \\ 0 & x < 0 \end{cases}$$

² mean

$$E(X) = 1$$

² variance

$$V(X) = 1$$

- ² memoryless property of the exponential distributed random variables: the future values of the exponentially distributed values are not affected by the past values. Compare this to, for example, a uniformly distributed random variable, one can see the difference. For example, when throwing a fair coin, we can consider the probability of head and tail is the same which has the value of 0.5. If, after a result of head, we would expect to see a tail (though it may not happen). In exponentially distributed random variable, we cannot have this type of expectation. In another word, we know nothing about the future value of the random variable given a full history of the past.

Mathematical proof.

$$\begin{aligned} P(X > s + t | X > s) &= \frac{P(X > s + t)}{P(X > s)} = \frac{e^{-\lambda(s+t)}}{e^{-\lambda s}} \\ &= e^{-\lambda t} = P(X > t) \end{aligned}$$

- ² Example 6.17 and 6.18 on page 204 and 205 where Example 6.18 demonstrates the memoryless property of the exponential distribution.

5.5.3 Distribusi Gamma

- ² pdf

$$f(x) = \begin{cases} \frac{\lambda^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\lambda x} & x > 0 \\ 0 & \text{otherwise} \end{cases}$$

where $\Gamma(\alpha) = (\alpha - 1)!$ when α is an integer.

- ² When $\alpha = 1$, this is the exponential distribution. In another word, the Gamma distribution is a more general form of exponential distribution.

- ² mean

$$E(X) = 1/\lambda$$

- ² variance

$$V(X) = 1/\lambda^2$$

5.5.4 Distribusi Erlang

- 2 When the parameter λ in Gamma distribution is an integer, the distribution is referred to as Erlang distribution.
- 2 When $\lambda = k$, a positive integer, the cdf of Erlang distribution is (using integration by parts)

$$F(x) = \begin{cases} 1 - \sum_{i=0}^{k-1} \frac{e^{-\lambda x} (\lambda x)^i}{i!} & x > 0 \\ 0 & x \leq 0 \end{cases}$$

which is the sum of Poisson terms with mean

- 2 mean

$$E(X) = 1/\lambda$$

- 2 variance

$$V(X) = \frac{1}{\lambda^2}$$

- 2 Example 6.19, 6.20 on page 208, 209.

5.5.5 Distribusi Normal

- 2 pdf

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

- 2 cdf

$$F(X) = P(X \leq x) = \int_{-\infty}^x \frac{1}{\sigma\sqrt{2\pi}} \exp \left\{ -\frac{1}{2} \frac{(t-\mu)^2}{\sigma^2} \right\} dt$$

This value is very difficult to calculate. Often a table is made for $X \sim N(0; 1)$. Because an $X \sim N(\mu; \sigma^2)$ can be transformed into $X \sim N(0; 1)$ by let

$$Z = \frac{X - \mu}{\sigma}$$

- 2 To calculate $P(X \leq x)$ for $X \sim N(\mu; \sigma^2)$, we use $\Phi \left(\frac{x - \mu}{\sigma} \right)$

Example: to calculate $F(56)$ for $N(50; 9)$, we have

$$F(56) = \Phi \left(\frac{56 - 50}{3} \right) = \Phi(2) = 0.9772$$

- 2 mean μ
- 2 variance σ^2
- 2 Notation: $X \sim N(\mu; \sigma^2)$
- 2 The curve shape of the normal pdf is like a "bell".
- 2 properties:
 - $\lim_{x \rightarrow -\infty} f(x) = 0$ and $\lim_{x \rightarrow \infty} f(x) = 0$
 - $f(\mu - x) = f(\mu + x)$ the pdf is symmetric about μ because of this, $\Phi(\mu - x) = 1 - \Phi(x)$
 - the maximum value of the pdf occurs at $x = \mu$ (thus, the mean and the mode are equal).
- 2 Example 6.21 and 6.22 on page 211, 6.23 and 6.24 on page 213.

5.5.6 Distribusi Weibull

The random variable X has a Weibull distribution if its pdf has the form

$$f(x) = \begin{cases} \frac{\sigma}{\mu} \left(\frac{x - \mu}{\sigma} \right)^{\sigma-1} \exp \left(- \left(\frac{x - \mu}{\sigma} \right)^\sigma \right) & x \geq \mu \\ 0 & \text{otherwise} \end{cases}$$

- 2 Weibull distribution has the following three parameters:
 - μ which has the range of which is the location parameter
 - σ which is greater than zero which is the scale parameter
 - σ which is a positive value determines the shape

5.6 Estimasi Means dan Variances

- 2 Suppose X_1, X_2, \dots, X_n are independent and identically distributed RVs (sharing the same probability distributions) with mean μ and variance σ^2 .
- 2 To estimate μ , we use the sample mean:
- 2 An unbiased estimator of σ^2 is:

- ² To access the precision of the sample mean as an estimator of μ is to construct a confidence interval of μ , which involves estimating the variance of the sample mean.

5.7 Confidence Intervals dari Mean

- ² X_i is the same as before; Z is RV:
- ² The classical central limit theorem:
- If n is sufficiently large, the random variable Z will be approximately distributed as a standard normal random variable.
 - Difficult to use the above, since the variance is generally unknown.
 - Use the sample variance
 - For n sufficiently large, we can construct an approximate confidence interval for μ .
 - If one constructs a very large # of independent $(1 - \alpha) \times 100\%$ CIs, each based on n observations, where n is sufficiently large, the proportion of these CIs that contain (cover) μ should be $(1 - \alpha)$.
- ² What does n is sufficiently large mean?
- ² Alternate CI expression
- Begins with the assumption that the observations are normally distributed and then uses the t distribution:
 - The new CI will also be approximate in coverage but it will be less peaked and has longer tails than the normal distribution.

5.8 Testing Hypothesis

- ² Assuming the observations are normally distributed (or are approximately so) then test the null hypothesis H_0 that $\mu = \mu_0$, for some fixed, hypothesized μ_0 .
- ² If $\sqrt{n}(\bar{X} - \mu_0)$ is large, (recall that \bar{X} is the point estimator for μ), H_0 is not likely to be true.
- ² The form of two-tailed hypothesis test for $\mu = \mu_0$:
- If $|t_n| > t_{n-1, \alpha/2}$ reject H_0
- If $|t_n| \leq t_{n-1, \alpha/2}$ accept H_0

5.9 Distribusi Empiris dan Ringkasan

- ² Used when a RV has no known distribution.
- ² Discrete: histogram and cdf
- ² Continuous: generate cdf by defining a continuous, pairwise-linear distribution function.

A major task in simulation is the collection and analysis of input data. One of the first steps in this task is hypothesizing a distributional form for the input data.

How to do this?

- ² Compare the shape of the pdf or pmf to a histogram of the data.
- ² Understand that certain physical processes give rise to specific distributions.

Bab 6

Gambaran Umum Model-Model Antrian

6.1 Model-Model Antrian

- ² Whether solved mathematically or analyzed through simulation, QM provide useful info for designing & evaluating the performance of queuing systems.
- ² Typical measures:
 - Server utilization
 - Length of waiting lines
 - Delays of customers as a function of input parameters such as arrival rate, service demands, service rate, # of servers.
- ² Involves trade-offs analysis between server utilization, customer satisfaction (line lengths & delays)
- ² For simple systems, measures can be computed mathematically or analytically.
 - Output measures are related to input parameters by a set of mathematical formulae.
 - Easier to solve
 - More restrictive assumptions
 - Rough-cut estimates
- ² For more complex systems, simulation is usually required.

6.2 Karakteristik Sistem Antrian

² Beberapa key elements: customers, servers dan network of queues

Calling population:

The population of potential customers is referred to as the calling population. Use examples from Table 7.1

² The calling population can be finite or infinite.

² In the systems with large population, we usually assume the population is infinite.

² The key difference between "finite" and "infinite" population model is how the arrival rate is defined.

- In an infinite population model, the arrival rate is not affected by the number of customers in the system. Usually the system is viewed as an open system, customers come from outside the system and leave the system after finishing the work.

- In a finite population model, the arrival rate at a server is affected by the population in the system. Usually the system is viewed as a closed system, customers (with a fixed population) don't leave the system, they merely move around system from one server to another, from one queue to another.

System capacity:

the maximum number of customers allowed in the system is called the system capacity. Examples:

² Telephone booth: capacity is one. When the server is busy (the booth occupied), the incoming customer is turned away.

² A typical TCP/IP packet buffer has a limit of 4096 bytes.

² The number of simultaneous connections allowed at a web server usually is a fixed constant.

The arrival process:

for infinite population

- inter-arrival time characterized by a random distribution, e.g. Poisson distribution

- scheduled arrivals, such as patients to a physician's office; airline flight arrives at an airport. The interarrival time may be a constant, or a constant with a small random fluctuation.
- at least one customer is assumed to always be present in the queue, so the server is never idle because of the lack of the customers, e.g. parts in an assembly line.

for infinite population:

- Define a customer is pending when it is outside queueing system, and it is a member of the potential calling population. E.g. when simulating a local area network, if a particular computer is powered on, we say it is pending. As soon as it is powered on, the customer (computer) will demand service from the network.
- Define runtime of a given customer as the length of the time from departure from the queueing system until the customer's next arrival to the queue. Runtime essentially is the time when the customer is being serviced.

The arrival rate in a infinite population model is a function of the number of pending customers.

6.3 Sifat Antrian dan Displin Antrian

6.3.1 Sifat Antrian

how customers react to the situation of the queue.

balk: Customers leave when they see the queue is too long.

renege: Customers leave after being in the queue when they see the queue is moving too slowly.

jockey: Customers move from one queue to another if they think they have chosen a slow line.

6.3.2 Displin Antrian

how customers are served in the queue.

FIFO: first-in-first-out

LIFO: last-in-first-out

SIRO: service in random order

SPT: shortest processing time first

PR: service according to priority

RR: round robin

6.4 Waktu Pelayanan dan Mekanisme Pelayanan

- ² Service time of successive arrivals are denoted by S_1, S_2, S_3, \dots . They may be constants or random variables. If they are random variables, they follow certain distribution.
- ² A queueing system consists of service centers and inter-connecting queues.
- ² Each service center contains a number of servers, c .
 - $c = 1$ single server
 - $1 < c < \infty$ multiple servers
 - $c = \infty$ unlimited servers
- ² Transition state and steady state: simulation needs some time to “warm up” from a initial state to a steady state. The status in a transition state should not be considered as indicative as typical measures. One should always wait until certain time elapses before collecting statistics.
- ² Examples 7.1 and 7.2 on page 239

6.5 Notasi Antrian

- ² Kendall's notation

$A=B=c=N=K$

- A: interarrival time distribution
 - M (Markovian, memoryless, exponential)
 - G (general or arbitrary)
 - D (constant or deterministic)

- E_k (Erlang of order k)
- B : service time distribution, same as A
- c : number of servers
 - 1, multiple, infinite
- N : system capacity or queue size
 - Finite, infinite
- K : size of calling population
- Table 6.2

6.6 Pengukuran Long-Run Kinerja Sistem Antrian

² Main long-run measures:

- L : long-run time-average number of customers in the system
- L_Q : in the queue
- w : long-run time average time spent in the system
- w_Q : in the queue
- ρ : server utilization

² Other measures:

- long-run proportion of customers who are delayed in queue
- longer than t_0 time units
- long-run proportion of customers turned away
- long-run proportion of time the waiting line contains more than k_0 customers

Time-average number in system L
influenced by initial conditions at time 0 and the run length T .

- $L(t)$: number of customers in system at time t .
- T_i : total time during $[0; T]$ in which the system contained exactly i customers.
- \hat{L} : time-weighted-average number in system:

- As $T \rightarrow \infty$; $\hat{L} = L$ (long-run time-average number in system)
- $L_Q(t)$: number of customers waiting in line
- L_Q : long-run time-average number waiting in line.
- \hat{L}_Q : observed time-average number of customers in line from time 0 to time T .
- T_i^Q : total time during $[0; T]$ in which exactly i customers are waiting in line.

Average time spent in system per customer,
 is similar to L

Conservation equation (Little's Law)

$$L = \lambda W$$

- λ : long-run average arrival rate;
- $\hat{\lambda}$: observed average arrival rate;
- $\hat{\lambda} = \lambda$ as $T \rightarrow \infty$ and $N \rightarrow \infty$
- The average number of customers in the system at an arbitrary point in time = the average number of arrivals per time unit * average time spent in the system.

Server utilization

proportion of time that the server is busy.

Server utilization in $G/G/1$ is 1

- arrival rate: λ (customers per time unit)
- service rate: μ (customers per time unit)
- server can be considered as a queuing system in itself, so $L = \lambda W$ can be applied.
- What is W for the server subsystem, i.e., average server time?
 $W = 1/\mu$
- \hat{L} : observed average number in server subsystem
- L_s : average number in server subsystem or busy servers
- In general, for a single-server queue, $L_s = \lambda W = \lambda / \mu = \rho$
 ρ also called the offered load; a measure of workload

Server utilization in $G/G/c$ is ρ/c

- c identical servers in parallel: the choice of server might be made at random.
- Maximum service rate for is $c\mu$: all servers are busy.
- $L_s = \rho E(S) = \rho/\mu = 1$ (average # of busy servers = c)
- $\rho = L_s/c = \lambda/c\mu = 1$ (long-run average server utilization = 1: proportion of time an arbitrary server is busy in the long run.)
- For the system to be stable, $c > \rho = 1$
- A stable queue can still have long lines.
- Trade-off analysis: server utilization vs. customer satisfaction

Costs in Queuing Problems

- Cost can be associated with various aspects of the queue or servers.
- $\$x$: cost per hour per customer; $\$y$ per hour while busy
- Average cost per customer: $\$x \times W_Q$
- Average cost per hour: $\$x \times L_Q$ / hour
- Cost for a set of c parallel servers
 - Server is busy: $\$y \times c\rho$
 - Server is idle: $\$y \times c(1 - \rho)$

² Objective: minimize total costs by varying above parameters (# of servers, service rate, system capacity)

6.7 Steady-State Populasi Tak-Terbatas Model Markovian

² Arrival: Poisson process with λ arrivals.

² Assumptions:

- Arrivals occur one at a time with FIFO discipline.
- The distribution of the #s of arrivals between t and $t+s$ depends only on the length of the interval s and not on the starting point t .

- The #s of arrivals during nonoverlapping time intervals are independent random variables. Or future arrivals occur completely at random, independent of the #s of arrivals in the past time intervals.
 - It has been shown that if interarrival times are exponentially & independently distributed, then the #s of arrivals is a Poisson process. (see section 5.5)
- ² These models are called Markovian models because of the exponential distribution assumptions
- ² $P_n(t)$: probability of n customers in system at time t .
- ² P_n : steady-state probability of having n customers in system
- ² A queuing system is in steady state if the system in a given state is independent of time t , i.e., $P_n(t) = P_n$:
- ² Sections deal with math models to get a rough guide of the system behaviors, model parameter, e.g. L , instead of simulation models which delivers a statistical estimate (e.g. \hat{L}) of the parameter hat.
- ² Two properties are important to consider for steady state: starting state and remaining in steady state once it is reached.

6.7.1 $M=G=1$

(when N and K are infinite, they may be dropped from notation)

- ² Assumptions:
- mean service times $1/\mu$ and variance $1/\mu^2$, terdapat one server
 - $\rho = 1/\mu < 1$ dan $M=G=1$ has a steady-state probability distribution.
 - Steady-state characteristics: Table 6.3.
 - What is P_0 ?
 - What is L ?
 - What is L_Q ?

Antrian $M=M=1$

(The service times are exponentially distributed)

2 Assumptions:

- exponentially distributed
- mean: $1/\lambda$ = standar deviasi dari distribusi ekponesial
- variance: $1/\lambda^2$
- steady-state parameters shown in Table 6.4.
- effect of λ , and L and w
 - examples 7.11 & 7.12

effect of utilization & service variability

- coefficient of variation (cv):
 - $(cv)^2 = V(x)/[E(x)]^2$
 - Figure 7.12

Relationship between M/G/1 and M/M/1, and M/G/c and M/M/c

Correction factor for L_Q and w_Q :

Rewrite L_Q and w_Q for M/G/1 queue (Table 7.3) in terms of the coefficient of variation and compare it with that for M/M/1 (Table 7.4). Correction factor is $(1 + (cv)^2)/2$, see equation (7.19). The same correction factor can also be applied to M/G/c and M/M/c to obtain L_Q and w_Q .

6.8 Jaringan Antrian

In practice, many systems are modeled as networks of single queues in which customers departing one queue may be routed to another.

Some basic principles:

- (a) If no customers are created or destroyed in the queue, then
- (b) departure rate = arrival rate over the long run.
- (c) Arrival rate for queue i is λ_i and $0 \leq p_{ij} \leq 1$ of them are routed to queue j , then the arrival rate from queue i to j is $\lambda_i p_{ij}$.
- (d) Overall rate into queue j is the sum of the arrival rate from all sources.

- (e) If queue j has $c_j < 1$ parallel servers with service rate m_j , then the utilization of each server is ?

$$\rho_j = \frac{\lambda_j}{c_j \mu_j}$$

- (f) If, for queue j , arrivals from outside the network is Poisson process with rate λ_j and there are c_j servers with exponentially distributed service times of mean $1/\mu_j$, then in steady state queue j behaves like an $M/M/c_j$ queue with arrival rate as stated in point 3.

6.9 Ringkasan

Queuing models are widely used to estimate desired performance measures of the system. The estimate contains random error, and thus a proper statistical analysis is required to assess the accuracy of the estimate.

Mathematical models/solutions may not be practical, but provide a rough estimate of a performance measure. An important application of mathematical queuing models is determining the minimum number of servers needed at a service centre. $\rho = \lambda / c\mu < 1$ can be used to provide an initial estimate for the # of servers, c .

Performance bottleneck or congestion may be detected using queuing modeling. Congestion may be decreased by adding more servers or by reducing the mean value and variability of service times. Queuing models can also be used to evaluate alternative system designs.

Bagian III

Bilangan dan Variabel Acak

Bab 7

Pembangkitan Bilangan Acak

Random numbers play a key role in discrete event simulation. We have used the uniformly distributed random numbers in many programming assignments before simulation. In simulation, we need random numbers with other distribution besides the uniformly distributed one.

7.1 Sifat-Sifat Bilangan Acak

A sequence of random numbers, $R_1; R_2; R_3; \dots$ must have two important properties:

1. Uniformity, i.e. they are equally probable every where
2. Independence, i.e. the current value of a random variable has no relation with the previous values

Each random number R_i is an independent sample drawn from a continuous uniform distribution between zero and one.

² pdf

$$f(x) = \begin{cases} \frac{1}{2} & 0 \leq x \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

² expectation

$$E(R) = \int_0^1 x dx = \frac{1}{2}$$

² variance

$$V(R) = \int_0^1 x^2 dx - [E(R)]^2 = \frac{1}{12}$$

Some consequences of the uniformity and independence properties

1. If the interval (0; 1) is divided into n sub-intervals of equal length, the expected number of observations in each interval is N/n where N is the total number of observations. Note that N has to be sufficiently large to show this trend.
2. The probability of observing a value in a particular interval is independent of the previous values drawn.

7.2 Pembangkitan Bilangan Acak Pseudo

- ² In computer simulation, we often do not want to have pure random numbers because we would like to have the control of the random numbers so that the experiment can be repeated.
- ² In general, a systematic way to generate pseudo-random number is used to generate the random numbers used in simulation. Some algorithms are needed.
- ² We generate the uniformly distributed random numbers ...rst; then we use this to generate random numbers of other distribution.
- ² Some desired properties of pseudo-random number generators:
 - The routine should be fast.
 - The routine should be portable across hardware platforms and programming languages.
 - The routine should have sufficiently long cycle.
 - A cycle length represents the length of the random number sequence before previous numbers begin to repeat themselves in an earlier order. For example

4; 9; 5; 6; 9; 3; 8; 4; 9; 5; 6; 9; 3; 8; 4; 9; 5; 6; 9; 3; 8; ::

appears to have a cycle length of 7 (this is just an example of cycle, a random number of cycle 7 is completely unacceptable!).

- A special case of cycling is degenerating where the same random numbers appear repeatedly.
- Because we use an algorithm to generate random number, cycling cannot be avoided. But long cycles (e.g. a few millions or a few billions) serve the purpose of general simulations.
- The random numbers should be replicable.
- Most importantly, the generated random numbers should closely approximate the ideal statistical properties of uniformity and independence.

7.3 Teknik Pembangkitan Bilangan Acak

Many different methods of generating pseudo-random numbers are available. This text introduces two of them, with one in great detail.

7.3.1 Metode Kongruen Linier

The linear congruential method produces a sequence of integers $X_1; X_2; X_3; \dots$ between zero and $m-1$ according to the following recursive relationship:

$$X_{i+1} = (aX_i + c) \bmod m; \quad i = 0; 1; 2; \dots$$

- ² The initial value X_0 is called the seed;
- ² a is called the constant multiplier;
- ² c is the increment
- ² m is the modulus

The selection of a , c , m and X_0 drastically affects the statistical properties such as mean and variance, and the cycle length.

When $c \neq 0$, the form is called the mixed congruential method; When $c = 0$, the form is known as the multiplicative congruential method.

Contoh 8.1 on page 292

Issues to consider:

- ² The numbers generated from the example can only assume values from the set $I = \{0; 1=m; 2=m; \dots; (m-1)=m\}$. If m is very large, it is of less problem. Values of $m = 2^{31} - 1$ and $m = 2^{48}$ are in common use.

² To achieve maximum density for a given range, proper choice of a ; c ; m and X_0 is very important. Maximal period can be achieved by some proven selection of these values.

- For m a power of 2, i.e. , $m = 2^b$ and $c \neq 0$, the longest possible period is $P = m = 2^b$, when c is relatively prime to m and $a = 1 + 4k$ where k is an integer.
- For m a power of 2, i.e. , $m = 2^b$ and $c = 0$, the longest possible period is $P = m/4 = 2^{b-2}$, when X_0 is odd and the multiplier, a is given by $a = 3 + 8k$ or $a = 5 + 8k$ where k is an integer.
- For m a prime number and $c = 0$, the longest possible period is $P = m - 1$ when a satisfies the property that the smallest k such that $a^k - 1$ is divisible by m is $k = m - 1$. For example, we choose $m = 7$ and $a = 3$, the above conditions satisfy. Here k has to be 6.
 - when $k = 6$, $a^k - 1 = 728$ which is divisible by m
 - when $k = 5$, $a^k - 1 = 242$ which is not divisible by m
 - when $k = 4$, $a^k - 1 = 80$ which is not divisible by m
 - when $k = 3$, $a^k - 1 = 26$ which is not divisible by m

Of course, the longest possible period here is 6, which is of no practical use. But the example shows how the conditions can be checked.

Examples 8.2, 8.3 and 8.4 on page 294 and page 295.

7.3.2 Metode Kongruen Linier Kombinasi

By combining two or more multiplicative congruential generators may increase the length of the period and results in other better statistics. See Example 8.5 on page 297.

7.4 Test Bilangan Acak

When a random number generator is devised, one needs to test its property. The two properties we are concerned most are uniformity and independence. A list of tests will be discussed. The first one tests for uniformity and the second to fifth ones test independence.

1. Frequency test
2. Runs test
3. Autocorrelation test

4. Gap test
5. Poker test

The algorithms of testing a random number generator are based on some statistics theory, i.e. testing the hypotheses. The basic ideas are the following, using testing of uniformity as an example.

We have two hypotheses, one says the random number generator is indeed uniformly distributed. We call this H_0 , known in statistics as null hypothesis. The other hypothesis says the random number generator is not uniformly distributed. We call this H_1 , known in statistics as alternative hypothesis.

We are interested in testing result of H_0 , reject it, or fail to reject it.

To see why we don't say accept H null, let's ask this question: what does it mean if we had said accepting H null? That would have meant the distribution is truly uniform. But this is impossible to state, without exhaustive test of a real random generator with infinite number of cases. So we can only say failure to reject H null, which means no evidence of non-uniformity has been detected on the basis of the test. This can be described by the saying "so far so good".

On the other hand, if we have found evidence that the random number generator is not uniform, we can simply say reject H null.

It is always possible that the H_0 is true, but we rejected it because a sample landed in the H_1 region, leading us to reject H_0 . This is known as Type I error. Similarly if H_0 is false, but we didn't reject it, this also results in an error, known as Type II error.

With these information, how do we state the result of a test? (How to perform the test will be the subject of next a few sections)

- ² A level of statistical significance has to be given. The level is the probability of rejecting the H null while the H null is true (thus, Type I error).

$$\alpha = P(\text{reject } H_0 | H_0 \text{ true})$$

- ² We want the probability as little as possible. Typical values are 0.01 (one percent) or 0.05 (five percent).
- ² Decreasing the probability of Type I error will increase the probability of Type II error. We should try to strike a balance.

For a given set of random numbers produced by a random number generator, the more tests are, the more accurate the results will be.

7.4.1 Tes Frekuensi

- ² The frequency test is a test of uniformity.
- ² Two different methods available, Kolmogorov-Smirnov test and the chi-square test. Both tests measure the agreement between the distribution of a sample of generated random numbers and the theoretical uniform distribution.
- ² Both tests are based on the null hypothesis of no significant difference between the sample distribution and the theoretical distribution.

Tes Kolmogorov-Smirnov

This test compares the cdf of uniform distribution $F(x)$ to the empirical cdf of the sample of N observations.

$$^2 F(x) = x, 0 \leq x \leq 1$$

$$^2 S_N(x) = \frac{\text{number of } R_1; R_2; \dots; R_N \leq x}{N}$$

- ² As N becomes larger, $S_N(x)$ should be close to $F(x)$

- ² Kolmogorov-Smirnov test is based on the statistic

$$D = \max |F(x) - S_N(x)|$$

that is the absolute value of the differences.

- ² Here D is a random variable, its sampling distribution is tabulated in Table A.8.
- ² If the calculated D value is greater than the ones listed in the Table, the hypothesis (no disagreement between the samples and the theoretical value) should be rejected; otherwise, we don't have enough information to reject it.
- ² Following steps are taken to perform the test.

1. Rank the data from smallest to largest

$$R_{(1)} \leq R_{(2)} \leq \dots \leq R_{(N)}$$

2. Compute

$$D^+ = \max_{1 \leq i \leq N} \left(\frac{i-1}{N} - R_{(i)} \right)^{3/4}$$

$$D^- = \max_{1 \leq i \leq N} \left(R_{(i)} - \frac{i}{N} \right)^{3/4}$$

3. Compute

$$D = \max(D^+, D^-)$$

4. Determine the critical value, D_{α} , from Table A.8 for the specified significance level and the given sample size N .

5. If the sample statistic D is greater than the critical value D_{α} , the null hypothesis that the sample data is from a uniform distribution is rejected; if $D \leq D_{\alpha}$, then there is no evidence to reject it.

² Example 8.6 on page 300.

Chi-Square test

The chi-square test looks at the issue from the same angle but uses different method. Instead of measure the difference of each point between the samples and the true distribution, chi-square checks the "deviation" from the "expected" value.

$$\chi^2 = \sum_{i=1}^n \frac{(O_i - E_i)^2}{E_i}$$

where n is the number of classes (e.g. intervals), O_i is the number of samples observed in the interval, E_i is expected number of samples in the interval. If the sample size is N , in a uniform distribution,

$$E_i = \frac{N}{n}$$

See Example 8.7 on page 302.

7.4.2 Tes Runs

Runs up dan down

The runs test examines the arrangement of numbers in a sequence to test the hypothesis of independence.

See the tables on page 303. With a closer look, the numbers in the first table go from small to large with certain pattern. Though these numbers will pass the uniform tests in previous section, they are not quite independent.

A run is defined as a succession of similar events preceded and followed by a different event.

E.g. in a sequence of tosses of a coin, we may have

H T T H H T T T H T

The first toss is preceded and the last toss is followed by a "no event". This sequence has six runs, first with a length of one, second and third with length two, fourth length three, fifth and sixth length one.

A few features of a run

- two characteristics: number of runs and the length of run
- an up run is a sequence of numbers each of which is succeeded by a larger number; a down run is a sequence of numbers each of which is succeeded by a smaller number

If a sequence of numbers have too few runs, it is unlikely a real random sequence. E.g. 0.08, 0.18, 0.23, 0.36, 0.42, 0.55, 0.63, 0.72, 0.89, 0.91, the sequence has one run, an up run. It is not likely a random sequence.

If a sequence of numbers have too many runs, it is unlikely a real random sequence. E.g. 0.08, 0.93, 0.15, 0.96, 0.26, 0.84, 0.28, 0.79, 0.36, 0.57. It has nine runs, five up and four down. It is not likely a random sequence.

If a is the total number of runs in a truly random sequence, the mean and variance of a is given by

$$\bar{a} = \frac{2N - 1}{3}$$

and

$$s_a^2 = \frac{16N - 29}{90}$$

For $N > 20$, the distribution of a is reasonably approximated by a normal distribution, $N(\bar{a}, s_a^2)$. Converting it to a standardized normal distribution by

$$Z_0 = \frac{a - \bar{a}}{s_a}$$

that is

$$Z_0 = \frac{a - [(2N - 1)/3]}{\sqrt{(16N - 29)/90}}$$

Failure to reject the hypothesis of independence occurs when $|Z_0| < Z_{\alpha/2}$, where α is the level of significance. See Figure 8.3 on page 305. See Example 8.8 on page 305

Runs above dan below the mean.

The previous test for up runs and down runs are important. But they are not adequate to assure that the sequence is random.

Check the sequence of numbers at the top of page 306, where they pass the runs up and down test. But it display the phenomenon that the first 20 numbers are above the mean, while the last 20 are below the mean.

Let n_1 and n_2 be the number of individual observations above and below the mean, let b the total number of runs.

For a given n_1 and n_2 the mean and variance of b can be expressed as

$$\mu_b = \frac{2n_1n_2}{N} + \frac{1}{2}$$

and

$$\sigma_b^2 = \frac{2n_1n_2(2n_1n_2 - N)}{N^2(N - 1)}$$

For either n_1 or n_2 greater than 20, b is approximately normally distributed

$$Z_0 = \frac{b - \mu_b}{\sigma_b} = \frac{b - \frac{2n_1n_2}{N} - \frac{1}{2}}{\sqrt{\frac{2n_1n_2(2n_1n_2 - N)}{N^2(N - 1)}}}$$

Failure to reject the hypothesis of independence occurs when $|Z_0| \leq z_{\alpha/2}$, where α is the level of significance. See Example 8.9 on page 307

Runs test: length of runs.

The example in the book:

0.16, 0.27, 0.58, 0.63, 0.45, 0.21, 0.72, 0.87, 0.27, 0.15, 0.92, 0.85,...

If the same pattern continues, two numbers below average, two numbers above average, it is unlikely a random number sequence. But this sequence will pass other tests.

We need to test the randomness of the length of runs.

Let Y_i be the number of runs of length i in a sequence of N numbers. E.g. if the above sequence stopped at 12 numbers ($N = 12$), then $Y_1 = Y_3 = Y_4 = \dots = Y_{11} = 0$ dan $Y_2 = 6$

Obviously Y_i is a random variable. Among various runs, the expected value for runs up and down is given by

$$E(Y_i) = \frac{2}{(i+3)!} N (i^2 + 3i + 1) - (i^3 + 3i^2 - i - 4)^{\frac{1}{i+3}}, \text{ for } i \leq N - 2$$

and

$$E(Y_i) = \frac{2}{N!}, \text{ for } i = N - 1$$

The number of runs above and below the mean, also random variables, the expected value of Y_i is approximated by

$$E(Y_i) = \frac{Nw_i}{E(I)}; N > 20$$

where $E(I)$ the approximate expected length of a run and w_i is the approximate probability of length i . w_i is given by

$$w_i = \frac{n_1}{N} \frac{n_2}{N} + \frac{n_2}{N} \frac{n_1}{N}$$

$E(I)$ is given by

$$E(I) = \frac{n_1}{n_2} + \frac{n_2}{n_1}; N > 20$$

The approximate expected total number of runs (of all length) in a sequence of length N is given by

$$E(A) = \frac{N}{E(I)}; N > 20$$

(total number divided by expected run length).

The appropriate test is the chi-square test with O_i being the observed number of runs of length i

$$\hat{A}_0^2 = \sum_{i=1}^L \frac{[O_i - E(Y_i)]^2}{E(Y_i)}$$

where $L = N - 1$ for runs up and down, $L = N$ for runs above and below the mean. See Example 8.10 on page 308 for length of runs up and down. See Example 8.11 on page 311 for length of above and below the mean.

7.4.3 Tes Auto-correlation

The tests for auto-correlation are concerned with the dependence between numbers in a sequence.

The list of the 30 numbers on page 311 appears to have the effect that every 5th number has a very large value. If this is a regular pattern, we can't really say the sequence is random.

The test computes the auto-correlation between every m numbers (m is also known as the lag) starting with the i th number.

Thus the autocorrelation $\frac{1}{2}r_{im}$ between the following numbers would be of interest.

$$R_i; R_{i+m}; R_{i+2m}; \dots; R_{i+(M+1)m}$$

The value M is the largest integer such $i + (M + 1)m \leq N$ that where N is the total number of values in the sequence. E.g. $N = 17; i = 3; m = 4$, then the above sequence would be 3; 7; 11; 15 ($M = 2$). The reason we require $M + 1$ instead of M is that we need to have at least two numbers to test ($M = 0$) the autocorrelation.

Since a non-zero autocorrelation implies a lack of independence, the following test is appropriate

$$H_0 : \frac{1}{2}r_{im} = 0$$

$$H_1 : \frac{1}{2}r_{im} \neq 0$$

For large values of M , the distribution of the estimator $\frac{1}{2}r_{im}$, denoted as b_{im} , is approximately normal if the values $R_i; R_{i+m}; R_{i+2m}; \dots; R_{i+(M+1)m}$ are uncorrelated.

Form the test statistic

$$Z_0 = \frac{b_{im}}{\sqrt{\frac{1}{3}b_{im}}}$$

which is distributed normally with a mean of zero and a variance of one.

The actual formula for and the standard deviation is

$$b_{im} = \frac{1}{M+1} \sum_{k=0}^M R_{i+km} R_{(k+1)m} \quad i = 0:25$$

and

$$b_{im} = \frac{\sqrt{\frac{1}{12M+7}}}{12(M+1)}$$

After computing Z_0 , do not reject the null hypothesis of independence if $|Z_0| \leq z_{\alpha/2}$

$$z_{\alpha/2} \leq Z_0 \leq z_{\alpha/2}$$

where α is the level of significance. See Example 8.12 on page 312.

7.4.4 Tes Gap

The gap test is used to determine the significance of the interval between recurrence of the same digit. A gap of length x occurs between the recurrence of some digit.

See the example on page 313 where the digit 3 is underlined. There are a total of eighteen 3's in the list. Thus only 17 gaps can occur.

The probability of a particular gap length can be determined by a Bernoulli trial.

$$P(\text{gap of } n) = P(x \neq 3)P(x \neq 3) \cdots P(x \neq 3)P(x = 3)$$

If we are only concerned with digits between 0 and 9, then

$$P(\text{gap of } n) = 0.9^n(0.1)$$

The theoretical frequency distribution for randomly ordered digits is given by

$$P(\text{gap of } n) = F(x) = 0.1 \sum_{n=0}^x (0.9)^n = 1 - 0.9^{x+1}$$

Steps involved in the test.

Step 1.

Specify the cdf for the theoretical frequency distribution given by Equation (8.14) based on the selected class interval width (See Table 8.6 for an example).

Step 2.

Arrange the observed sample of gaps in a cumulative distribution with these same classes.

Step 3.

Find D ; the maximum deviation between $F(x)$ and $S_N(x)$ as in Equation 8.3 (on page 299).

Step 4.

Determine the critical value, D_{α} , from Table A.8 for the specified value of α and the sample size N :

Step 5.

If the calculated value of D is greater than the tabulated value of D_{α} , the null hypothesis of independence is rejected.

² See the Example 8.13 on page 314

7.4.5 Tes Poker

The poker test for independence is based on the frequency in which certain digits are repeated in a series of numbers. For example 0.255, 0.577, 0.331, 0.414, 0.828, 0.909, 0.303, 0.001... In each case, a pair of like digits appears in the number.

In a three digit number, there are only three possibilities.

1. The individual digits can be all different. Case 1.
2. The individual digits can all be the same. Case 2.
3. There can be one pair of like digits. Case 3.

$P(\text{case 1}) = P(\text{second differ from the first}) * P(\text{third differ from the first and second}) = 0.9 * 0.8 = 0.72$; $P(\text{case 2}) = P(\text{second the same as the first}) * P(\text{third same as the first}) = 0.1 * 0.1 = 0.01$ $P(\text{case 3}) = 1 - 0.72 - 0.01 = 0.27$

² See Example 8.14 on page 316

Bab 8

Pembangkitan Variabel (Variate) Acak

Now that we have learned how to generate a uniformly distributed random variable, we will study how to produce random variables of other distribution using the uniformly distributed random variable.

The techniques discussed include inverse transform and convolution. Also discussed is the acceptance-rejection technique.

All work here assume the existence of a source of uniform (0,1) random numbers,

² pdf

$$f_R(x) = \begin{cases} \frac{1}{2} & 0 \leq x \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

² cdf

$$F_R(x) = \begin{cases} 0 & x < 0 \\ x & 0 \leq x \leq 1 \\ 1 & x > 1 \end{cases}$$

8.1 Teknik Transformasi Balik

² The inverse transform technique can be used to sample from exponential, the uniform, the Weibull and the triangle distributions.

² The basic principle is to find the inverse function of F , F^{-1} such that

$$F(F^{-1}) = F^{-1}(F) = 1$$

F^{-1} denotes the solution of the equation $r = F(x)$ in terms of r , not $1=F$. For example, the inverse of $y = x$ is $x = y$, the inverse of $y = 2x+1$ is $x = (y - 1)/2$.

8.1.1 Distribusi Eksponensial

pdf

$$f(x) = \begin{cases} \frac{1}{2} e^{-x}; & x \geq 0 \\ 0; & x < 0 \end{cases}$$

cdf

$$F(x) = \int_{-\infty}^x f(t) dt = \begin{cases} 1 - e^{-x}; & x \geq 0 \\ 0; & x < 0 \end{cases}$$

The idea is to solve $y = 1 - e^{-x}$ for x where y is uniformly distributed on $(0; 1)$ because it is a cdf. Then x is exponentially distributed.

This method can be used for any distribution in theory. But it is particularly useful for random variates that their inverse function can be easily solved.

Steps involved are as follows:

Step 1.

Compute the cdf of the desired random variable X .

For the exponential distribution, the cdf is $F(x) = 1 - e^{-x}$.

Step 2.

Set $R = F(X)$ on the range of X .

For the exponential distribution, $R = 1 - e^{-x}$ on the range of $x \geq 0$.

Step 3.

Solve the equation $F(X) = R$ for X in terms of R . For the exponential distribution, the solution proceeds as follows.

$$\begin{aligned} 1 - e^{-X} &= R \\ e^{-X} &= 1 - R \\ -X &= \ln(1 - R) \\ X &= -\ln(1 - R) \end{aligned}$$

Step 4.

Generate (as needed) uniform random numbers $R_1; R_2; \dots$ and compute the desired random variates by

$$X_i = F^{-1}(R_i)$$

In the case of exponential distribution

$$X_i = \frac{1}{\lambda} \ln(1 - R_i)$$

for $i = 1; 2; 3; \dots$ where R_i is a uniformly distributed random number on $(0; 1)$.

In practice, since both R_i AND $1 - R_i$ are uniformly distributed random number, so the calculation can be simplified as

$$X_i = \frac{1}{\lambda} \ln R_i$$

To see why this is correct, recall

$$X_i = F^{-1}(R_i)$$

and

$$R_i = F(X_i)$$

Because $X_i \leq x_0$ is equivalent to $F^{-1}(R_i) \leq x_0$, and F is a non-decreasing function (so that if $x \leq y$ then $F(x) \leq F(y)$) we get $X_i \leq x_0$ is equivalent to $F^{-1}(R_i) \leq x_0$, which implies that $F(F^{-1}(R_i)) \leq F(x_0)$ which is equivalent to $R_i \leq F(x_0)$.

This means

$$P(X_i \leq x_0) = P(R_i \leq F(x_0))$$

Since R_i is uniformly distributed on $(0; 1)$ and $F(x_0)$ is the cdf of exponential function which is between 0 and 1, so

$$P(R_i \leq F(x_0)) = F(x_0)$$

which means

$$P(X_i \leq x_0) = F(x_0)$$

This says the $F(x_0)$ is the cdf for X and X has the desired distribution.

Once we have this procedure established, we can proceed to solve other similar distribution for which a inverse function is relatively easy to obtain and has a closed formula.

8.1.2 Distribusi Uniform

- ² If we want a random variate X uniformly distributed on the interval $[a; b]$; a reasonable guess for generating X is given by

$$X = a + (b - a)R$$

where R is uniformly distributed on $(0; 1)$.

- ² If we follow the steps outlined in previous section, we get the same result.

- ² pdf for X

$$f(x) = \begin{cases} \frac{1}{b-a} & a \leq x \leq b \\ 0 & \text{otherwise} \end{cases}$$

- ² Steps:

Step 1.

the cdf

$$F(x) = \begin{cases} 0 & x < a \\ \frac{x-a}{b-a} & a \leq x \leq b \\ 1 & x > b \end{cases}$$

Step 2.

Set $F(X) = (X - a)/(b - a) = R$

Step 3.

Solve for X in terms of R yields

$$X = a + (b - a)R$$

which is the same as the earlier guess.

Step 4.

Generate R_i as needed, calculate X_i using the function obtained.

8.1.3 Distribusi Weibull

- ² pdf

$$f(x) = \begin{cases} \frac{1}{\theta} x^{\theta-1} e^{-x^\theta} & x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

2 Steps:

Step 1.

cdf

$$F(x) = 1 - e^{-(x-\theta)^+}; \text{ for } x \geq 0$$

Step 2.

let

$$F(X) = 1 - e^{-(X-\theta)^+} = R$$

Step 3.

Solve for X in terms of R yields

$$X = \theta + [-\ln(1 - R)]^{1/2}$$

Step 4.

Generate uniformly distributed from (0;1) feeding them to the function in Step 3 to get X.

8.1.4 Distribusi Triangular

2 cdf

$$f(x) = \begin{cases} \frac{2}{b-a}x & 0 \leq x \leq 1 \\ \frac{2}{b-a}(2-x) & 1 < x \leq 2 \\ 0 & \text{otherwise} \end{cases}$$

Its curve is shown on page 328

2 Steps

Step 1.

cdf

$$F(x) = \begin{cases} 0 & x \leq 0 \\ \frac{x^2}{2} & 0 < x \leq 1 \\ 1 - \frac{(2-x)^2}{2} & 1 < x \leq 2 \\ 1 & x > 2 \end{cases}$$

Step 2.

let

$$R = \frac{X^2}{2}; \text{ for } 0 < X \leq 1$$

$$R = 1 - \frac{(2 - X)^2}{2}; \text{ for } 1 < X \leq 2$$

Step 3.

Solve X in terms of R

$$X = \begin{cases} \sqrt{2R} & 0 \leq R \leq 1 \\ 2 - \sqrt{2(1-R)} & 1 \leq R \leq 2 \end{cases}$$

8.1.5 Distribusi Kontinyu Empiris

- ² If the modeler has been unable to find a theoretical distribution that provides a good model for the input data, it may be necessary to use the empirical distribution of the data.
- ² A typical way of resolving this difficult is through “curve fitting”.
- ² Steps involved: (see Example 9.2 on page 328)
 - Collect empirical data and group them accordingly.
 - Tabulate the frequency and cumulative frequency.
 - Now assume the value of cumulative frequency as a function of the empirical data, i.e. $F(x) = r$
 - Establish a relation between x and r using linear interpolation

$$X_1 = x_a + \frac{R_1 - y_a}{y_b - y_a} (x_b - x_a)$$

for each of the intervals.

- ² Example 9.3 on page 332: note that in this example, five response time are used which are all distinct. If there are values that are the same, or if the number of samples are large, we can certainly group them in the different ways.

8.1.6 Distribusi Kontinyu tanpa invers bentuk tertutup

- ² Many continuous distributions don't have a closed-form inverse function. Strictly speaking, the inverse transform method discussed earlier cannot be applied.
- ² If we are willing to accept numeric solution, inverse functions can be found. Here we have an example for normal distribution.
- ² One of the inverse cdf of the standard normal distribution was proposed by Schmeiser:

$$X = F^{-1}(R) \approx \frac{R^{0.135} - (1 - R)^{0.135}}{0.1975}$$

for $0.0013499 \leq R \leq 0.9986501$ which matches the true normal distribution with one digit after decimal point. See Table 9.6 on page 335.

8.1.7 Distribusi Diskrit

- ² All discrete distributions can be generated using the inverse transform technique.
- ² This section discusses the case of empirical distribution, (discrete) uniform distribution, and geometric distribution.
- ² Empirical discrete distribution. The idea is to collect and group the data, then develop the pdf and cdf. Use this information to obtain F^{-1} so that $X = F^{-1}(R)$ will be the random number function that we look for. Example 9.4 on page 336.
- ² Discrete Uniform Distribution (Example 9.5 on page 338)

– pdf

$$p(x) = 1/k; \quad x = 1, 2, \dots, k$$

– cdf

$$F(x) = \begin{cases} 0 & x < 1 \\ \frac{i}{k} & 1 \leq x \leq i \\ 1 & x \geq k \end{cases}$$

– Let $F(X) = R$

- Solve X in terms of R . Since x is discrete,

$$r_{i+1} = \frac{i+1}{k} < R \cdot r_i = \frac{i}{k}$$

thus,

$$\frac{i+1}{k} < Rk \cdot \frac{i}{k}$$

Consider the fact that i and k are integers and R is between $(0; 1)$. For the above relation to hold, we need

$$X = dRke$$

- For example, to generate a random variate X , uniformly distributed on $f1; 2; :::; 10g$ (thus $k = 10$)

$$\begin{aligned} R_1 &= 0.78 & X_1 &= d7.8e = 8 \\ R_2 &= 0.03 & X_2 &= d0.3e = 1 \\ R_3 &= 0.23 & X_3 &= d2.3e = 3 \\ R_4 &= 0.97 & X_4 &= d9.7e = 10 \end{aligned}$$

² Example 9.6 on page 339 gives us another favor. When an inverse function has more than one solution, one has to choose which one to use. In the example, one results in positive value and the other results in negative value. The choice is obvious.

² Example 9.7 on page 340: Geometric distribution.

- pmf

$$p(x) = p(1-p)^x; x = 0; 1; 2; \dots$$

where $0 < p < 1$

- cdf

$$\begin{aligned} F(x) &= \sum_{j=0}^x p(1-p)^j \\ &= \frac{p(1-p)^{x+1}}{1-p(1-p)} \\ &= 1-p(1-p)^{x+1} \end{aligned}$$

- Let $R = F(x)$, solve for x in term of R . Because this is a discrete random variate, use the inequality (9.12) on page 337,

$$F(x_{i+1}) = r_{i+1} < R \leq r_i = F(x_i)$$

that is

$$F(x_{i+1}) = r_{i+1} = 1 - (1 - p)^{x+1} < R \leq 1 - (1 - p)^x = r_i = F(x_i)$$

$$(1 - p)^{x+1} \cdot 1 - R < (1 - p)^x \\ (x + 1) \ln(1 - p) \cdot \ln(1 - R) < x \ln(1 - p)$$

Notice that $\ln(1 - p) < 0$

$$\frac{\ln(1 - R)}{\ln(1 - p)} \leq x < \frac{\ln(1 - R)}{\ln(1 - p)}$$

Consider that x must be an integer, so

$$X = \left\lceil \frac{\ln(1 - R)}{\ln(1 - p)} \right\rceil$$

- Let $d_i = \lceil \frac{\ln(1 - R)}{\ln(1 - p)} \rceil$ the equation above becomes

$$X = d_i - \ln(1 - R) \cdot \frac{1}{\ln(1 - p)}$$

The item in the ceiling function before subtracting one is the function to generate exponentially distributed variate.

- Thus one way to generate geometric distribution is to
 - let $\lambda = -\ln(1 - p)$ as the parameter to the exponential distribution,
 - generate an exponentially distributed variate by

$$-\frac{1}{\lambda} \ln(1 - R)$$

- subtract one and take the ceiling

- Example 9.8 on page 341

8.2 Transformasi Langsung Distribusi Normal

In the previous section, a simple, but less accurate method of generating a normal distribution was presented. Here we consider a direct transformation.

- ² Let Z_1, Z_2 be two standard normal random variates.
- ² Plot the two as a point in the plane and represent them in a polar coordinate system as

$$\begin{aligned} Z_1 &= B \cos \mu \\ Z_2 &= B \sin \mu \end{aligned}$$

- ² It is known that $B^2 = Z_1^2 + Z_2^2$ has the chi-square distribution with 2 degrees of freedom, which is equivalent to an exponential distribution with mean 2

$$\begin{aligned} Y &= \sum_{i=1}^{\infty} e^{-t} \cdot t^i / i!; \quad t \geq 0 \\ E[Y] &= 2 = \sum_{i=1}^{\infty} 2 \cdot e^{-t} \cdot t^i / i! \end{aligned}$$

thus the radius B can be generated using (9.3)

$$B = \sqrt{-2 \ln R}$$

- ² So a normal distribution can be generated by any one of the following.

$$\begin{aligned} Z_1 &= \sqrt{-2 \ln R_1} \cos(2\pi R_2) \\ Z_2 &= \sqrt{-2 \ln R_1} \sin(2\pi R_2) \end{aligned}$$

where R_1 and R_2 are uniformly distributed over $(0,1)$.

- ² To obtain normal variates X_i with mean μ and variance σ^2 , transform

$$X_i = \mu + \sigma Z_i$$

8.3 Metode Konvolusi

- ² The probability distribution of a sum of two or more independent random variables is called a convolution of the distributions of the original variables.
- ² Erlang distribution.

- an Erlang random variable X with parameters $(K; \mu)$ can be shown to be the sum of K independent exponential random variables X_i ($i = 1; 2; \dots; K$), each having a mean $1/\mu$

$$X = \sum_{i=1}^K X_i$$

- Using equation (9.3) that can generate exponential variable, an Erlang variate can be generated by

$$\begin{aligned} X &= \sum_{i=1}^K \frac{1}{\mu} \ln R_i \\ &= \frac{1}{\mu} \ln \left(\prod_{i=1}^K R_i \right) \end{aligned}$$

² Example 9.9 on page 343

8.4 Teknik Penerimaan Penolakan (Acceptance-Rejection)

² Example: use following steps to generate uniformly distributed random numbers between $1/4$ and 1 .

Step 1.

Generate a random number R

Step 2a.

If $R \geq 1/4$, accept $X = R$, goto Step 3

Step 2b.

If $R < 1/4$, reject R , return to Step 1

Step 3.

If another uniform random variate on $[1/4; 1]$ is needed, repeat the procedure beginning at Step 1. Otherwise stop.

- Do we know if the random variate generated using above methods is indeed uniformly distributed over $[1/4, 1]$? The answer is Yes. To prove this, use the definition. Take any $1/4 \leq a < b \leq 1$,

$$P(a < R \leq b | 1/4 \leq R \leq 1) = \frac{P(a < R \leq b)}{P(1/4 \leq R \leq 1)} = \frac{b - a}{3/4}$$

which is the correct probability for a uniform distribution on $[0, 1]$.

- The efficiency: use this method in this particular example, the rejection probability is $1/4$ on the average for each number generated. The number of rejections is a geometrically distributed random variable with probability of “success” being $p = 3/4$, mean number of rejections is $(1-p)/p = 4/3 - 1 = 1/3$ (i.e. $1/3$ waste).
- For this reason, the inverse transform ($X = 1/4 + (3/4)R$) is more efficient method.

2 Poisson Distribution

- pmf

$$p(n) = P(N = n) = \frac{e^{-\lambda} \lambda^n}{n!}; \quad n = 0, 1, 2, \dots$$

where N can be interpreted as the number of arrivals in one unit time.

- From the original Poisson process definition, we know the inter-arrival time A_1, A_2, \dots are exponentially distributed with a mean of λ , i.e. λ arrivals in one unit time.
- Relation between the two distribution:

$$N = n$$

if and only if

$$A_1 + A_2 + \dots + A_n \leq 1 < A_1 + A_2 + \dots + A_n + A_{n+1}$$

essentially this means if there are n arrivals in one unit time, the sum of interarrival time of the past n observations has to be less than or equal to one, but if one more interarrival time is added, it is greater than one (unit time).

- The A_i s in the relation can be generated from uniformly distributed random number $A_i = (-\ln R_i)$, thus

$$\sum_{i=1}^n \frac{1}{\lambda} \ln R_i \leq 1 < \sum_{i=1}^{n+1} \frac{1}{\lambda} \ln R_i$$

both sides are multiplied by λ

$$\ln \prod_{i=1}^n R_i = \sum_{i=1}^n \ln R_i \leq -1 < \sum_{i=1}^{n+1} \ln R_i = \ln \prod_{i=1}^{n+1} R_i$$

that is

$$\sum_{i=1}^N R_i \leq e^{\lambda} > \sum_{i=1}^{N+1} R_i$$

- Now we can use the Acceptance-Reject method to generate Poisson distribution.

Step 1.

Set $n = 0$; $P = 1$.

Step 2.

Generate a random number R_{n+1} and replace P by $P R_{n+1}$.

Step 3.

If $P < e^{-\lambda}$, then accept $N = n$, meaning at this time unit, there are n arrivals. Otherwise, reject the current n , increase n by one, return to Step 2.

- Efficiency: How many random numbers will be required, on the average, to generate one Poisson variate, N ? If $N = n$, then $n + 1$ random numbers are required (because of the $(n + 1)$ random numbers product).

$$E(N + 1) = \lambda + 1$$

- Example 9.10 on page 346, Example 9.11 on page 347
- When λ is large, say $\lambda = 15$, the acceptance-rejection technique described here becomes too expensive. Use normal distribution to approximate Poisson distribution. When λ is large

$$Z = \frac{N - \lambda}{\sqrt{\lambda}}$$

is approximately normally distributed with mean 0 and variance 1, thus

$$N = \lambda + \sqrt{\lambda} Z$$

can be used to generate Poisson random variate.

Bagian IV

Analisis Data Simulasi

Bab 9

Pemodelan Masukan (Input Modeling)

Collect and analyze input data so that the simulation can be fed with appropriate data. This is a very important task. Without proper input data, the good simulation model won't generate correct, appropriate result.

9.1 Identifikasi Distribusi dengan Data

9.1.1 Histogram

² A frequency distribution or histogram is useful in identifying the shape of a distribution.

² A histogram is constructed as follows.

1. Divide the range of the data into intervals (intervals are usually of equal width).
2. Label the horizontal axis to conform to the intervals selected.
3. Determine the frequency of occurrences within each interval.
4. Label the vertical axis so that the total occurrences can be plotted for each interval.
5. Plot the frequencies on the vertical axis.

² The issue of intervals being too ragged, too coarse ... See Figure 10.1 on page 360.

² Example 10.2 on page 359, Example 10.3 on page 360

9.1.2 Penyeleksian Kelas Distribusi

There are many different distributions that may suit into a specific simulation task. Though exponential, normal and Poisson distributions are the ones used most often, others such as gamma and Weibull distributions are useful and important as well.

Here is a list of commonly used distributions.

Binomial

Models the number of successes in n trials, when the trials are independent with common success probability, p .

Negative Binomial including the geometric distribution

Models the number of trials required to achieve k successes.

Poisson

Models the number of independent events that occur in a fixed amount of time or space.

Normal

Models the distribution of a process that can be thought of as the sum of a number of component processes.

Log-normal

Models the distribution of a process that can be thought of as the product of a number of component processes.

Exponential

Models the time between independent events, or a process time which is memoryless.

Gamma

An extremely flexible distribution used to model non-negative random variables.

Beta

An extremely flexible distribution used to model bounded random variables.

Erlang

Models processes that can be viewed as the sum of several exponentially distributed processes.

Weibull

Models the time-to-failure for components.

Discrete or Continuous Uniform

Models complete uncertainty, since all outcomes are equally likely.

Triangular

Models a process when only the minimum, most-likely, and maximum values of the distribution are known.

Empirical

Resamples from the actual data collected.

9.1.3 Plot Quantile-Quantile

² If X is a random variable with cdf F , then the q -quantile of X is the value such that $F(x) = P(X \leq x) = q$ for $0 < q < 1$.

² When F has an inverse, we write $x = F^{-1}(q)$

² Now we discuss how to use $Q-Q$ plots to draw distributions.

- Let $x_i; i = 1; 2; \dots; n$ be a sample data from X .
- Re-order the x_i s as y_i s such that $y_1 \leq y_2 \leq \dots \leq y_n$.
- The $Q-Q$ plot is based on the fact that y_i is an estimate of the $(j_i - 1)/n$ quantile of X , that is,

$$y_i \approx F^{-1}\left(\frac{j_i - 1}{n}\right)$$

where F is the distribution function under the consideration that fits the samples.

- Plot y_i against $F^{-1}\left(\frac{j_i - 1}{n}\right)$. If the distribution fits, the line would form a straight line. Use the measurement of deviation to decide whether or not the set of samples fits the distribution function.

² Example 10.4 on page 365.

9.1.4 Estimasi Parameter

² After a family of distribution has been selected such as Poisson, Normal, Geometric ..., the next step is to estimate the parameters of the distribution.

² Sample mean and sample variance can be used to estimate the parameters in a distribution.

- Let $X_1; X_2; \dots; X_n$ be the sample of size n .
- The sample mean is

$$\bar{X} = \frac{\sum_{i=1}^n X_i}{n}$$

- The sample variance is

$$S^2 = \frac{\sum_{i=1}^n X_i^2 - n\bar{X}^2}{n-1}$$

- If the data are discrete and grouped in a frequency distribution, then we can re-write the equations as

$$\bar{X} = \frac{\sum_{j=1}^n f_j X_j}{n}$$

and

$$S^2 = \frac{\sum_{j=1}^n f_j X_j^2 - n\bar{X}^2}{n-1}$$

- Example 10.5 on page 368
- If the data are continuous, we “discretize” them and estimate the mean

$$\bar{X} = \frac{\sum_{j=1}^n f_j m_j}{n}$$

and the variance

$$S^2 = \frac{\sum_{j=1}^n f_j m_j^2 - n\bar{X}^2}{n-1}$$

- where is the observed frequency in the j th class interval, m_j is the midpoint of the j th interval, and c is the number of class intervals.
 - Example 10.6 on page 369
- ² A few well-established, suggested estimators are listed in Table 10.3 on page 370, followed by examples. They come from theory of statistics.
- ² The examples include Poisson Distribution, Uniform Distribution, Normal Distribution, Exponential Distribution, and Weibull Distribution.

9.2 Tes Goodness-of-Fit

- ² Earlier we introduced hypothesis test to examine the quality of random number generators. Now we will apply these tests to hypothesis about distributional forms of input data.
- ² Goodness-of-fit tests provide helpful guidance for evaluating the suitability of a potential input model.
- ² The tests depends heavily on the amount of data. If very little data are available, the test is unlikely to reject any candidate distribution (because not enough evidence to reject); if a lot of data are available, the test will likely reject all candidate distributions (because none ...ts perfectly).
- ² Failing to reject a candidate should be viewed as a piece of evidence in favor of that choice; while rejecting an input model is only one piece of evidence against the choice.

9.2.1 Tes Chi-Square

- ² Chi-square test is for large sample sizes, for both discrete and continuous distributional assumptions, when parameters are estimated by maximum likelihood.
- Arranging the n observations into a set of k class intervals or cells.
 - The test statistic

$$\hat{A}_0^2 = \sum_{i=1}^k \frac{(O_i - E_i)^2}{E_i}$$

where O_i is the observed frequency in the i th class interval and E_i is the expected frequency in that class interval.

- The \hat{A}_0^2 approximately follows the chi-square distribution with $k - s - 1$ degrees of freedom, where s represents the number of parameters of the estimated distribution. E.g Poisson distribution has $s = 1$, normal distribution has $s = 2$.
- The hypothesis
 H_0 : the random variable, X , conforms to the distributional assumption with the parameter(s) given by the parameter estimate(s)
 H_1 : the random variable X does not conform the distribution
- The critical value $x_{\alpha; k-s-1}^2$ is found in Table A.6. H_0 is rejected if $x_0^2 > x_{\alpha; k-s-1}^2$.
- The choice of k , the number of class intervals, see Table 10.5 on page 377.
- o Example 10.13 on page 377.

² Chi-square test with equal probabilities:

- If a continuous distributional assumption is being tested, class intervals that are equal in probability rather than equal in width of interval should be used.
- Example 10.14: Chi-square test for exponential distribution (page 379)
 - test with intervals of equal probability (not necessary equal width)
 - number of intervals less than or equal to $n=5$
 - $n = 50$, so $k = 10$, according to recommendations in Table 10.5, 7 to 10 class intervals be used.
 - Let $k = 8$, thus $p = 0.125$
 - The end points for each interval are computed from the cdf for the exponential distribution

$$F(a_i) = 1 - e^{-a_i}$$

where a_i represents the end point of the i th interval.

- Since is the cumulative area from zero to a_i , thus $F(a_i) = ip$

$$ip = 1 - e^{-a_i}$$

thus

$$a_i = -\ln(1 - ip); i = 0; 1; \dots; k$$

regardless the value of p , $a_0 = 0$ and $a_k = 1$.

- With $\lambda = 0.084$ in this example and $k = 8$,

$$a_1 = \lambda \frac{1}{0.084} \ln(1 - 0.125) = 1.590$$

continue with $i = 2, 3, \dots, 7$ results in 3.425, 5.595, 8.252, 11.677, 16.503, and 24.755.

- See page 379 and 380 for completion of the example.
- Example 10.15 (Chi-square test for Weibull distribution) on page 380
- Example 10.16 (Computing intervals for the normal distribution) on page 381
 - For the given data, using suggested estimator in Table 10.3 on page 370, we know (the original data was from Example 10.3 on page 360)

$$\hat{\lambda} = 11.90$$

- Kolmogorov-Smirnov Goodness-of-fit test
 - Chi-square test heavily depends on the class intervals. For the same data, different grouping of the data may result in different conclusion, rejection or acceptance.
 - The K-S goodness-of-fit test is designed to overcome this difficulty. The idea of K-S test is from $q_i - q$ plot.
 - The K-S test is particularly useful when sample size are small and when no parameters have been estimated from the data.
 - Example 10.7 on page 383, using the method described in Section 8.4.1 on page 299. A few notes:
 - † If the interarrival time is exponentially distributed, the arrival times are uniformly distributed on $(0; T]$

Bab 10

Verifikasi dan Validasi Model Simulasi

So far, we have discussed how to run simulation, how to generate and test random number generators, how to build input data models. This chapter discusses how one might verify and validate a simulation model.

The goal of validation process

- 2 to produce a model that represents true system behavior close enough for the model to be used as a substitute for the physical system
- 2 to increase to an acceptable level of credibility of the model

The verification and validation process consists of the following components.

1. Verification is concerned with building the model right.
2. Validation is concerned with building the right model.

10.1 Pembangunan, Verifikasi dan Validasi Model

- 2 Steps involved
 - 2 Observe the real system and the interaction among its various components, and collect data on its behavior.
 - 2 Construct a conceptual model - a collection of assumptions on the components and the structure of the system, plus hypotheses on the values of model input parameters.
 - 2 Translate the conceptual model into a computerized model.

10.2 Verifikasi Model Simulasi

² Make sure the conceptual model is translated into a computerized model properly and truthfully. Typically three intuitive mechanisms can be used.

(a) Common sense

- i. Have the computerized representation checked by someone other than its developer.
- ii. Make a flow diagram which includes each logically possible action of a system can take when an event occurs, and follow the model logic for each action for each event type.
- iii. Closely examine the model output under a variety of settings of input parameters.
- iv. Interactive Run Controller (IRC) or debugger is an essential component of successful simulation model building, which allows the analyst or the programmer to test the simulation program interactively.
- v. If one is modeling certain queueing system, the simulation results can be compared with theoretical results as upper or lower bound. For example, in a queueing system, if the increase in arrival rate or service time does not affect the queue length or waiting time, then something is suspicious.

(b) Thorough documentation: by describing in detail what the simulation programmer has done in design and implementing the program, one can often identify some problems.

(c) Trace: go through a few steps in an actual program to see if the behavior is reasonable (Example 11.1 on page 404)

² Verification of a simulation model is very similar in what a typical software product would have to go through.

10.3 Kalibrasi dan Validasi Model

² For a given program, while common sense verification is possible, strict verification of a model is intractable, very much similar to the proof of correctness of a program.

² Validation is a process of comparing the model and its behavior to the real system and its behavior.

- ² Calibration is the iterative process of comparing the model with real system, revising the model if necessary, comparing again, until a model is accepted (validated).
- ² Three-step approach in the validation process.
 - (a) Build a model that has high face validity.
 - (b) Validate model assumptions
 - (c) Compare the model input-output transformations to corresponding input-output transformation for the real system.

The three-step approach is discussed next.

10.3.1 Face Validity

- ² Build a “reasonable model” on its face to model users who are knowledgeable about the real system being simulated.
- ² Do some “sanity check”

10.3.2 Validasi Asumsi Model

- ² Model assumptions fall into two categories: structural assumptions and data assumptions.
- ² Structural assumptions deal with such questions as how the system operates, what kind of model should be used, queueing, inventory, reliability, and others.
- ² “ Data assumptions: what kind of input data model is? What are the parameter values to the input data model?

10.3.3 Validasi Transformasi Input-Output

- ² View the model as a black box
 - ² Feed the input at one end and examine the output at the other end
 - ² Use the same input for a real system, compare the output with the model output
 - ² If they ...t closely, the black box seems working ...ne
 - ² Otherwise, something is wrong
- The bank example (Example 11.2 on page 411)

10.3.4 Validasi Input-Output menggunakan Data masukan historis

- ² In the previous bank example and other places, we used some artificial data as input to a simulation model.
- ² An alternative is to use past data as input (historical input data). Sometimes this is called trace-driven simulation.
- ² To conduct a validation test using historical input data, it is important that all the input data and all the system response data be collected during the same time period.
- ² Example 11.3 (The Candy Factory)
 - The Candy Factory has three machines, the Candy maker, the Candy packer, and the Box maker.
 - The three machines are connected by two conveyors.
 - The machines have random break-down time.
 - When a machine breaks down, it causes its incoming conveyors to be full and out-going conveyors to be empty.
 - The time-to-failure (operating time) and down-time of each machine is recorded. For machine i

$$T_{i1}, D_{i1}, T_{i2}, D_{i2}, \dots$$

- The system responses: the production level Z_1 , the number Z_2 and the time of occurrence Z_3 of operator interventions were recorded.
- The model responses ($Y_i; i = 1; 2; 3$) for the same parameters were collected for comparison to the corresponding system responses.
- The differences between Y_i and Z_i can be used to build test statistics (in this case the Student Test t) to see if the model is statistically the same as the real system.
- If there is only one set of input data, only one set of output can be obtained. The result doesn't carry much statistical significance.
- If K sets of input data are available, K runs of tests can be conducted. See Table 11.6 for an example.
- Let $Z_{ij}; j = 1; 2; \dots; K$ be the system responses and $W_{ij}; j = 1; 2; \dots; K$ be the model responses, the differences $d_j = Z_{ij} - W_{ij}$ are approximately normally distributed with mean μ_d and variance $\frac{3}{4}\sigma_d^2$

- If the mean μ_d is zero, then statistically W_{ij} are the same as Z_{ij} . A t test is conducted

$$H_0 : \mu_d = 0$$

and

$$H_1 : \mu_d \neq 0$$

- The t statistic is computed as

$$t_0 = \frac{\bar{d} - \mu_d}{S_d / \sqrt{K}}$$

- The critical value $t_{\alpha/2; K-1}$ where K is the degree of freedom. In our example, K is the number of input data sets (thus, the number of experiments run).
- If $|t_0| > t_{\alpha/2; K-1}$ do not reject H_0 , otherwise reject H_0 .

² Example 11.4 (The Candy Factory continued) Use actual numbers (e.g. $K = 5$, etc.) to show how the above example works.

Bab 11

Analisis Keluaran Model Tunggal

- ² Output analysis is the analysis of data generated by a simulation.
- ² How do we know if the result of a simulation is statistically significant?

11.1 Sifat Stokastik Data Keluaran

- ² Simulation result is generated for a given input data. Essentially a model is an input/output transformation.
- ² Since the input variables are random variables, the output variables are random variables as well, thus they are stochastic (probabilistic).
- ² Example 12.1 (Able and Baker, revisited)
 - Instead of a single run of simulation, four runs were conducted. The results are shown in Table 12.1 on page 431.
 - The utilization u_r and w_r system time were listed as 0.808, 0.875, 0.708, 0.842, and 3.74, 4.53, 3.84, 3.98.
 - There are two general questions we have to address by a statistical analysis of the observed utilization u_r
 1. Estimation of the true utilization $u = E(u_r)$ by a single value, called a point estimate.
 2. Estimation of the error in our point estimate, either in the form of a standard error or confidence interval. This is called an interval estimate.

- ² Example 12.2 (Effect of correlation and initial condition) We will see actual measurement of performance in Section 12.3 and 12.4.
- ² Example 12.3 (Fifth National Bank of Jasper, revised)

11.2 Jenis Simulasi menurut Analisis Keluaran

- ² Terminating simulation and steady-state simulations:
 - A terminating simulation is one that runs for some duration of time, where E is a specified event or set of events which stops the simulation.
 - A steady-state simulation is a simulation whose objective is to study long-run, or steady-state, behavior of a non-terminating system.
- ² Examples of terminating simulation (Example 12.4 - 12.7) on pages 435-436.
- ² Examples of steady-state simulation (Example 12.8 - 12.9) on page 437.

11.3 Pengukuran Kinerja dan Estimasi

Consider a set of output values for the same measure $Y_1; Y_2; \dots; Y_n$ (e.g. delays of n different runs, or waiting times of n different runs). We want to have

- ² a point estimate to approximate the true value of Y_i , and
- ² an interval estimate to outline the range where the true value lies.

11.3.1 Estimasi Titik

- ² The point estimator of based on the data $Y_1; Y_2; \dots; Y_n$ is defined by

$$\bar{y} = \frac{1}{n} \sum_{i=1}^n Y_i$$

- ² The point estimator $\hat{\mu}$ is said to be unbiased for μ if

$$E(\hat{\mu}) = \mu$$

In general

$$E(\hat{\mu}) = \mu + b$$

i.e. there is a drifting, or bias.

- ² For continuous data, the point estimator \bar{A} of based on data $fY(t); 0 \leq t \leq T_E$, where T_E is the simulation run length, is defined by

$$\bar{A} = \frac{1}{T_E} \int_0^{T_E} Y(t) dt$$

and is called a time average of $Y(t)$ over $[0; T_E]$.

- ² In general

$$E(\bar{A}) = \bar{A} + b$$

if $b = 0$, \bar{A} is said to be unbiased for \bar{A}

- ² One performance measure of these estimators (point or interval) is a quantile or a percentile.

- Quantiles describe the level of performance that can be delivered with a given probability p
- Assume Y represents the delay in queue a customer experiences, then the 0.85 quantile (or 85% percentile) of Y is the value α such that

$$P(Y \leq \alpha) = p = 0.85$$

11.3.2 Estimasi Interval

- ² Valid interval estimation typically requires a method of estimating the variance of the point estimator $\hat{\mu}$ or \bar{A} .

- ² Let $\sigma^2(\hat{\mu}) = \text{var}(\hat{\mu})$ represent the true variance of a point estimator $\hat{\mu}$, and let $s^2(\hat{\mu})$ represent an estimator of $\sigma^2(\hat{\mu})$ based on the data $Y_i; i = 1; 2; \dots; n$

- ² Suppose that

$$E \sum_{i=1}^h \sum_{j=1}^i \mu^2(\hat{\mu}) = B \sum_{i=1}^h \sum_{j=1}^i \mu^2(\mu)$$

where B is called the bias in the variance estimator.

- ² It is desirable to have

$$B = 1$$

in which case $\sum_{i=1}^h \sum_{j=1}^i \mu^2(\hat{\mu})$ is said to be an unbiased estimator of variance, $\sum_{i=1}^h \sum_{j=1}^i \mu^2(\mu)$.

- ² If it is an unbiased estimator, the statistic

$$t = \frac{\hat{\mu}_i - \mu}{\sqrt{\sum_{i=1}^h \sum_{j=1}^i \mu^2(\hat{\mu})}}$$

is approximately t distribution with some degree f of freedom.

- ² An approximate $100(1 - \alpha)\%$ confidence interval for μ is given by

$$\hat{\mu}_i - t_{\alpha/2; f} \sqrt{\sum_{i=1}^h \sum_{j=1}^i \mu^2(\hat{\mu})} \leq \mu \leq \hat{\mu}_i + t_{\alpha/2; f} \sqrt{\sum_{i=1}^h \sum_{j=1}^i \mu^2(\hat{\mu})}$$

- ² This relation involves three parameters, estimator for mean, estimator for variance, and the degree of freedom. How to determine these values?

– Estimator for mean is calculated as above as a point estimator

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^n Y_i$$

– Estimator for the variance and for the degree of freedom has to consider two separate cases

1. If Y_i s are statistically independent observations then use

$$S^2 = \frac{\sum_{i=1}^n (Y_i - \hat{\mu})^2}{n - 1}$$

1. to calculate

$$\sum_{i=1}^h \sum_{j=1}^i \mu^2(\hat{\mu}) = \frac{S^2}{n}$$

- with the degree of freedom $f = n - 1$.

1. If Y_i s are not statistically independent, then the above estimator for variance is biased. Y_i s is an autocorrelated sequence, sometimes called a time series.

In this case,

$$\hat{\sigma}^2(\hat{\mu}) = \text{var}(\hat{\mu}) = \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n \text{cov}(Y_i; Y_j)$$

i.e. one needs to calculate co-variance for every possible pair of observations. Too expensive.

If the simulation is long enough to have passed the transient phase, the output is approximately covariance stationary. That is Y_{i+k} depends on Y_{i+1} in the same way as Y_k depends on Y_1 . For a covariance stationary time series s , define the lag k autocovariance by

$$\gamma_k = \text{cov}(Y_1; Y_{1+k}) = \text{cov}(Y_i; Y_{i+k})$$

For $k = 0$, γ_0 becomes the population variance

$$\gamma_0 = \text{cov}(Y_{0+i}; Y_i) = \text{var}(Y_i)$$

The lag k autocorrelation is the correlation between any two observations k apart.

$$\rho_k = \frac{\gamma_k}{\gamma_0}$$

and has the property

$$\rho_{-k} = \rho_k \quad k = 1, 2, \dots$$

If a time series is covariance stationary, then the calculation of sample variance can be substantially simplified.

$$\hat{\sigma}^2(\hat{\mu}) = \frac{\gamma_0}{n} \left(1 + 2 \sum_{k=1}^{\lfloor n/2 \rfloor} \left(1 - \frac{k}{n} \right) \rho_k \right)$$

So all we need is to calculate covariance between one sample and every other samples, but not every sample with every other samples.

- Some discussions about why autocorrelation make it difficult to estimate $\hat{\sigma}^2(\hat{\mu})$ are skipped

11.4 Analisis Keluaran Simulasi Terminating

- ² Use independent replications, i.e. the simulation is repeated a total of R times, each run using a different random number stream and independently chosen initial conditions.
- ² Let Y_{ri} be the i th observations within replication r , for $i = 1; 2; \dots; n_r$ and $r = 1; 2; \dots; R$:
- ² For a fixed r , $Y_{r1}; Y_{r2}; \dots$ is an auto correlated sequence. For different replications $r \neq s$; Y_{ri} and Y_{sj} are statistically independent.
- ² Define a sample mean

$$\bar{p}_r = \frac{1}{n_r} \sum_{i=1}^{n_r} Y_{ri}; \quad r = 1; 2; \dots; R$$

- ² There are R samples, so R sample means, the overall point estimate is

$$\bar{p} = \frac{1}{R} \sum_{r=1}^R \bar{p}_r$$

11.4.1 Estimasi Interval untuk Jumlah Replikasi yang tetap

- ² The sample mean can be calculated as above
- ² The sample variance

$$s^2(\bar{p}) = \frac{1}{R} \sum_{r=1}^R \frac{\sum_{i=1}^{n_r} (p_{ri} - \bar{p})^2}{n_r - 1}$$

- ² When the output data are of the form $fY_r(t); 0 \leq t \leq T_E$ for $r = 1; 2; \dots; R$

$$\bar{h}_r = \frac{1}{T_E} \int_0^{T_E} Y_r(t) dt; \quad r = 1; 2; \dots; R$$

$$\bar{h} = \frac{1}{R} \sum_{r=1}^R \bar{h}_r$$

and

$$s^2(\bar{x}) = \frac{1}{R} \sum_{r=1}^R \frac{\sum_{i=1}^I x_{ri}^2}{R_i - 1}$$

- ² The sample variance is essentially a measure of error. In statistics, we use its square root as the standard error.
- ² Example 12.10 on page 446 (The Able-Baker carhop problem, continued): calculate the confidence interval for Able's utilization and average waiting time.
- ² Example 12.11 on page 447: the more runs, the more accurate the result is.

11.4.2 Estimasi Interval dengan Presisi Tertentu

- ² Previous section discusses for a given set of replications to calculate the confidence interval and error. Sometimes we need to do the inverse, given a level of error and confidence, how many replications are needed?
- ² The half-length (h.l.) of a $100(1 - \alpha)\%$ confidence interval for a mean μ , based on the t distribution, is

$$h.l. = t_{\alpha/2; R-1} s(\bar{x}) \quad (*)$$

where $s(\bar{x}) = \frac{S}{\sqrt{R}}$, S is the sample standard deviation, R is the number of replications.

- ² Assume an error criterion is specified with a confidence level $1 - \alpha$, it is desired that a sufficiently large sample size R be taken such that

$$P(|\bar{x} - \mu| < \frac{h}{2}) \geq 1 - \alpha$$

- ² Since we have the relation (*), the desired the error control condition can be written as

$$h.l. = \frac{t_{\alpha/2; R-1} S_0}{\sqrt{R}} \leq \frac{h}{2}$$

- ² Solve the above relation, we have

$$R \geq \left(\frac{t_{\alpha/2; R-1} S_0}{h/2} \right)^2$$

since $t_{\alpha/2; R_i - 1} \leq Z_{\alpha/2}$ the above relation can be written

$$R \geq \frac{\mu_{Z_{\alpha/2} S_0}^2}{2}$$

For $R \geq 50$; $t_{\alpha/2; R_i - 1} \approx Z_{\alpha/2}$ the inequality with standard normal distribution holds.

This says we need to run that many (R) replications to satisfy the error requirement.

- ² The true value of μ is in the following range with probability of $100(1 - \alpha)\%$

$$\mu \in \left[\bar{p} - \frac{t_{\alpha/2; R_i - 1} S}{\sqrt{R}}, \bar{p} + \frac{t_{\alpha/2; R_i - 1} S}{\sqrt{R}} \right]$$

- ² Example 12.12 on page 449

11.5 Analisis Keluaran Simulasi Steady-State

- ² Consider a single run of a simulation model whose purpose is to estimate a steady state, or long run, characteristics of the system.
- ² Assume $Y_1; Y_2; \dots$ are observations, which in general are samples of an autocorrelated time series.
- ² The steady-state measure of performance μ to be estimated is defined by

$$\mu = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n Y_i$$

- ² This result is independent of initial conditions, random number streams, ...

11.5.1 Inisialisasi Bias pada Simulasi Steady-State

There are several methods of reducing the point estimator bias which is caused by using artificial and unrealistic initial conditions in a steady-state simulation.

1. Initialize the simulation in a state that is more representative of long-run conditions. E.g. use a set of real data as initial condition.

2. Divide the simulation into two phases, warm-up phase and steady state phase. Data collection doesn't start until the simulation passes the warm-up phase.

Consider the example on page 452 (Example 12.13)

² A set of 10 independent runs, each run was divided into 15 intervals. The data were listed in Table 12.5 on page 453.

² Typically we calculate average within a run. Since the data collected in each run is most likely autocorrelated, a different method is used to calculate the average across the runs.

² Such averages are known as ensemble average.

Several issues:

1. Ensemble average will reveal a smoother and more precise trend as the number of replications, R , is increased.
2. Ensemble average can be smoothened further by plotting a moving average. In a moving average each plotted point is actually the average of several adjacent ensemble averages.
3. Cumulative averages become less variable as more data are averaged. Thus, it is expected that the curve at left side (the starting of the simulation) of the plotting is less smooth than the right side.
4. Simulation data, especially from queueing models, usually exhibits positive autocorrelation. The more correlation present, the longer it takes for the average to approach steady state.
5. In most simulation studies the analyst is interested in several measures such as queue length, waiting time, utilization, etc. Different performance measures may approach steady state at different rates. Thus it is important to examine each performance measure individually for initialization bias and use a deletion point that is adequate for all of them.

11.5.2 Metode Replikasi Simulasi Steady-State

- ² If initialization bias in the point estimator has been reduced to a negligible level, the method of independent replications can be used to estimate point-estimator variability and to construct a confidence interval.

- ² if significant bias remains in the point estimator and a large number of replications are used to reduce point estimator variability, the result confidence interval can be misleading.
 - The bias is not affected by the number of replications R , but by deleting more data (i.e. increasing d) or extending the length of each run (i.e. increasing n).
 - Increasing the number of replications R may produce shorter confidence intervals around a “wrong point”, rather than
- ² If d is the number of observations to delete from a total of n observations, a rough rule is $n-d$ should be at least $10d$, or should be at least $n \geq 11d$.
- ² Given the run length, the number of replications should be as many as possible. Kelton in 1986 established that there is little value to run more than 25 replications. So if time is available, make the simulation longer, instead of making more replications.
- ² See Example 12.14 on page 460 and Example 12.15 on page 461, where Example 12.15 demonstrate the cases where few observations were deleted.

A couple of notes as fewer observations were deleted (d is smaller):

1. The confidence interval shifts downward, reflecting the greater downward bias in as d decreases. This can be attributed as the result of a “cold start”.
2. The standard error of \bar{y} , namely decreases as d decreases. As d decreases, the number of samples included in the statistics increases, reducing the error range.

11.5.3 Ukuran Sample Simulasi Steady-State

In a steady-state simulation, a specified precision may be achieved either by increasing the number of replications R , or by increasing the run length n .

Example 12.16 on page 462 shows an example of calculating R for a given precision.

Example 12.17 on page 463 shows an example of increasing n for the given precision (recall the general rule: should be at least $n \geq 11d$).

11.5.4 Batch Means untuk Estimasi Interval Estimation pada Simulasi Steady-State

- ² One disadvantage of replication is that data must be deleted on each replication.
- ² One disadvantage of a single-replication is its data tend to be autocorrelated.
- ² The method of batch mean divides the output data from one replication into a few large batches.
- ² Treat the means of these batches as if they were independent. See the formula on page 463 and 464.
- ² The key issue is that no commonly accepted method for choosing an acceptable batch size m . This is actually one of the research areas in simulation.
 - o Schmeiser found for a fixed total sample size there is little benefit from dividing it into more than $k = 30$ batches.
 - o Although there is typically autocorrelation between batch means at all lags, the lag-1 autocorrelation is usually studied to assess the dependence between batch means.
 - o The lag-1 autocorrelation between batch means can be estimated using the method described earlier. They should not be estimated from a small number of batch means, i.e. we need to have large number of batches, though the size of batches could be small.
 - o If the total sample size is to be chosen sequentially (i.e. choose one for one experiment, choose another one for improvement etc.), then it is helpful to allow the batch size and number of batches to grow as the run length increases.
- ² Example 12.18 on page 465.