

Simulasi Monte Carlo dan Pemodelan Discrete Event

Chapter 2A

What We'll Do ...

- Pada bagian ini kita akan mempelajari bagaimana proses sampai muncul angka waktu kedatangan dan waktu proses dimana data tersebut merupakan hasil dari sampling yang juga menggunakan teknik monte carlo ini
- Pada bab ini akan dibahas beberapa terminologi umum yang akan kita pakai dalam membangun model yaitu “Activity cycle diagram”

Simulasi Monte Carlo

- Pendekatan monte carlo digunakan untuk menghasilkan variabel input dalam simulasi seperti waktu antar kedatangan, waktu proses dan variabel input lainnya yang sesuai dengan distribusi yang diinginkan.
- Teknik ini menggunakan bilangan random yang berdistribusi uniform untuk kemudian mengkonversinya menjadi distribusi probabilitas yang diinginkan.
- Sebelum kita dapat mengaplikasikan metode ini maka kita terlebih dahulu harus mengetahui distribusi beserta parameternya dari sistem nyata yang kemudian akan kita hasilkan kembali kedalam model.
- Hal ini biasanya memerlukan suatu observasi terhadap objek pengamatan misalnya dengan menerapkan studi pengukuran kerja dan waktu standard

Simulasi Monte Carlo

Secara manual metode ini mempunyai langkah-langkah sebagai berikut :

- 1. lakukan observasi terhadap parameter yang akan dimodelkan**
- 2. hitung frekwensi tiap-tiap nilai parameter**
- 3. hitung distribusi frekwensi kumulatif dan distribusi probabilitas kumulatif**
- 4. Pasangkan nilai kelas dari tiap parameter dengan bilangan random dengan range antara 00 sampai 99**
- 5. Tarik suatu bilangan random dengan menggunakan tabel random**
- 6. Dapatkan nilai parameter yang sesuai dengan memasang bilangan random yang dihasilkan.**

Simulasi Monte Carlo

Misalkan kita akan memodelkan waktu proses dari suatu stasiun kerja:

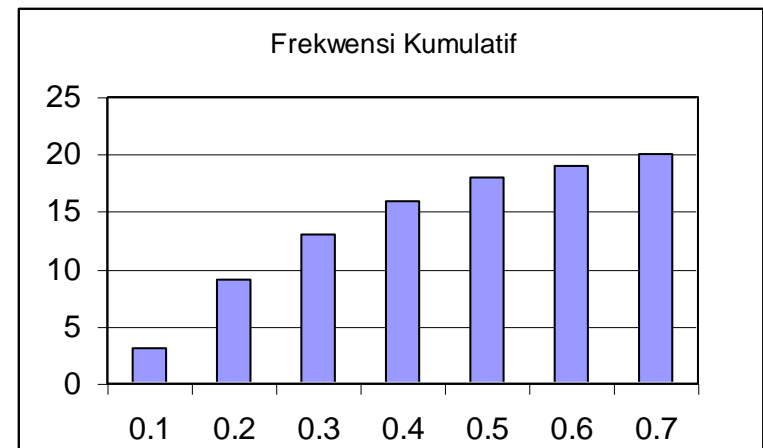
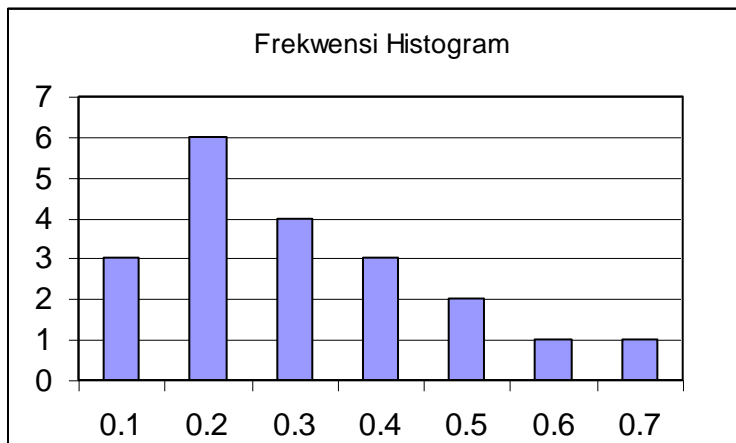
- Langkah 1. Hasil pengamatan menunjukkan waktu proses yang berbeda yaitu sebagai berikut :

0.4	0.1	0.2	0.4	0.7	0.3	0.2	0.5	0.2	0.6
0.1	0.3	0.5	0.1	0.3	0.2	0.3	0.2	0.5	0.4

Simulasi Monte Carlo

- Langkah 2 & 3. Frekwensi dan frekwensi kumulatif:

Nilai	Frekwensi	Frekwensi Kumulatif	Probabilitas kumulatif
0.1	3	3	0.15
0.2	6	9	0.45
0.3	4	13	0.65
0.4	3	16	0.80
0.5	2	18	0.90
0.6	1	19	0.95
0.7	1	20	1.00



Simulasi Monte Carlo

- Langkah 4. Pasangkan tiap kelas dengan suatu bilangan random

Nilai	Probabilitas kumulatif	Range Bilangan Random
0.1	0.15	00 – 14
0.2	0.45	15 – 44
0.3	0.65	45 – 64
0.4	0.80	65 – 79
0.5	0.90	80 – 89
0.6	0.95	90 – 94
0.7	1.00	95 – 99

- Selanjutnya kita dapat menggunakan tabel random untuk menghasilkan waktu proses yang sesuai. Misalkan dari tabel random kita dapatkan angka 27 maka nilai ini ada range 15-44 yang menunjuk pada nilai 0.2

Simulasi Monte Carlo

- Langkah 5 dan 6. Menghasilkan bilangan random dan mendapatkan nilai parameter
- Dengan menggunakan tabel random maka kita bisa mendapatkan nilai waktu proses dengan distribuis yang sesuai dengan hasil pengamatan. Jika kita baca tabel random tersebut pada kolom kiri dan kebawah maka kita dapatkan

Bilangan random	:	81	94	21	35	60	97	92	dst
Nilai Parameter	:	0.5	0.6	0.2	0.2	0.4	0.7	0.2	dst

- Selanjutnya dengan telah mempunyai nilai-nilai variabel input tersebut maka kita akan mendapatkan data lengkap untuk input simulasi dan kita tinggal melakukan simulasi

Terminologi : Komponen Sistem

- **Entiti** : adalah elemen sistem yang akan disimulasikan dan dapat diidentifikasi secara individu dan dapat dikenai proses.
- **Ada dua macam entiti yaitu** :
 - *Permanen Entiti* : entiti yang tetap berada dalam sistem seperti mesin dalam pabrik
 - *Temporer Entiti* : entiti yang keberadaannya dalam sistem tidak tetap, tetapi hanya lewat saja dalam sistem kemudian keluar misalnya job yang diproses

Terminologi : Komponen Sistem

- **Classes** : kumpulan entiti yang mempunyai kesamaan type dikelompokkan dalam satu group yang disebut **classes**.
- **Atribut** : adalah sifat atau informasi yang melekat pada tiap-tiap entiti atau **classes**. Sebagai contoh mobil, yang mana mobil mempunyai atribut warna, cc, type, merk. Atribut juga dapat digunakan untuk mengendalikan perilaku dari entiti dan mengendalikan disiplin dari antrian

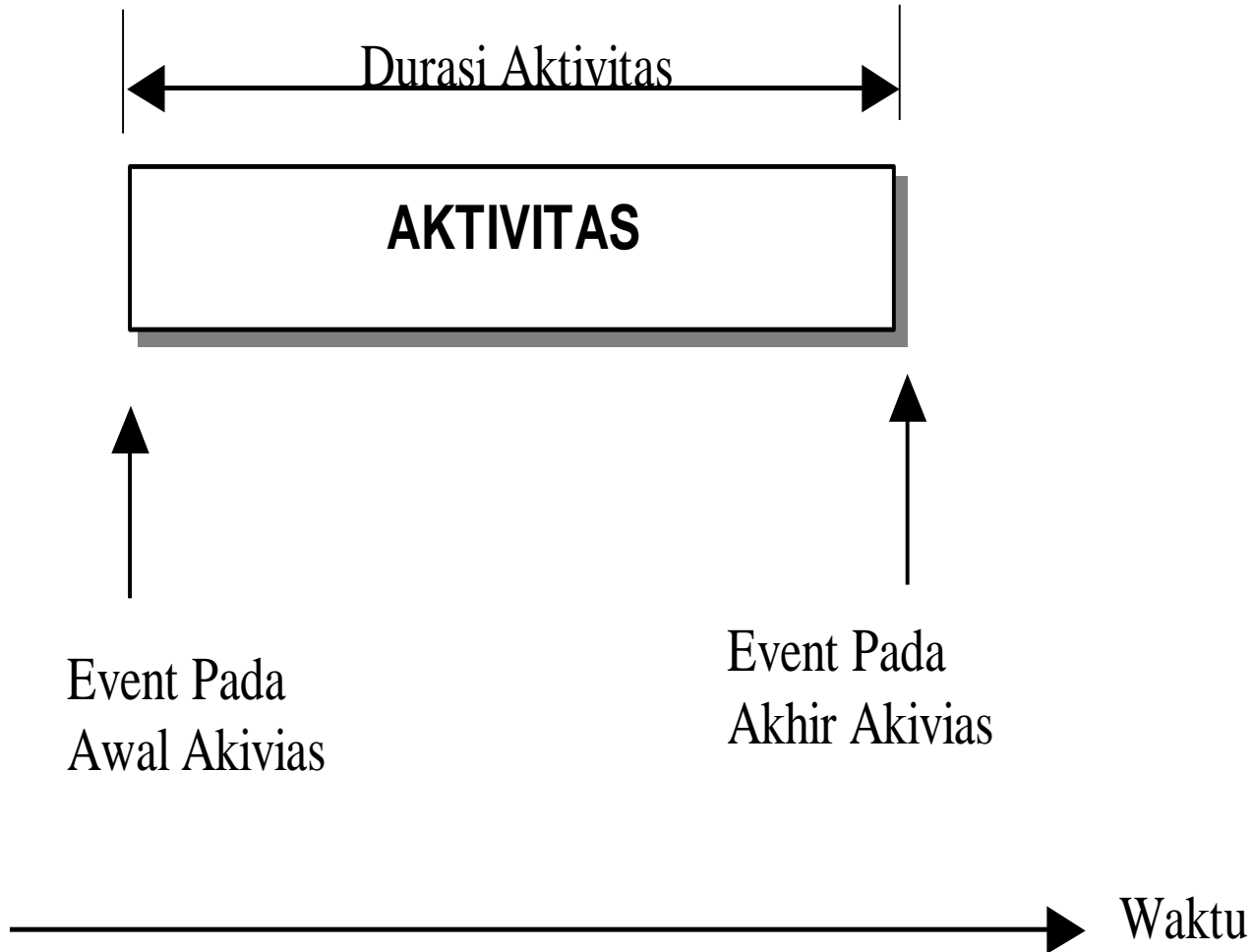
Terminologi : Komponen Sistem

- **Sets** : adalah sekumpulan **variable state** dari suatu sistem yang didalamnya berisi entiti.
 - Selama simulasi, meskipun entiti telah menjadi anggota dari *classes*, entiti tersebut bisa berubah status dan secara temporer ia akan bergerak dari satu *set* ke *set* lainnya.
 - Misalnya dalam suatu industri manufaktur, suatu job secara temporer akan menjadi anggota dari set entiti yang menunggu diproses, setelah selesai diproses ia akan ditransfer ke bagian finishing dan ia menjadi anggota dari set komponen yang menunggu finishing.
 - Keanggotaan terhadap set dapat digunakan untuk identifikasi status sistem pada suatu waktu tertentu

Terminologi : Operasi dari Entiti

- **Ketika simulasi berjalan, seluruh entiti akan saling berinteraksi dan melakukan perubahan state dari sistem. Beberapa terminologi untuk menjelaskan operasi dari entity:**
- **Event : adalah waktu sesaat yang secara significant menyebabkan terjadinya perubahan state dari sistem. Seperti ketika entiti masuk atau meninggalkan suatu set, atau pada saat operasi dimulai.**
- **Aktivitas : suatu entiti bergerak dari set satu ke set lainnya karena operasi yang mereka alami. Operasi dan prosedur yang mengawali suatu pada tiap event disebut aktivitas**

Terminologi : Operasi dari Entiti



***Terminologi* : Operasi dari Entiti**

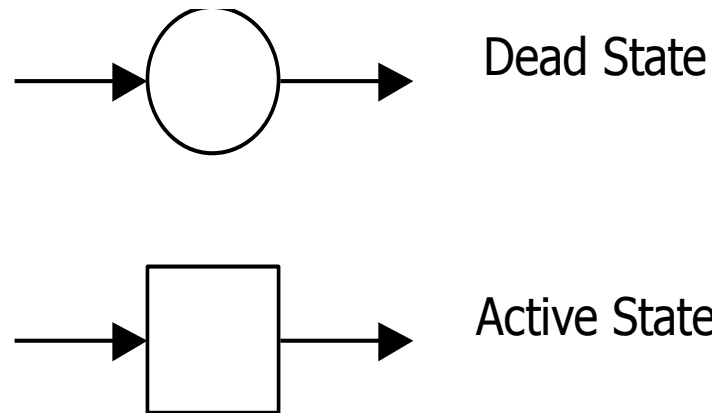
- **Proses:** adalah kumpulan event yang berurutan atau diurutkan secara kronologis.
- **Jam Simulasi (*simulation clock*):** adalah yang mengendalikan jalannya simulasi. Jam inilah yang akan dijalankan untuk menentukan kapan suatu event terjadi

Activity Cycle Diagram

- **Untuk membangun model yang cocok maka harus dilakukan beberapa hal berikut :**
 - Identifikasi entiti
 - Aktivitas tiap entiti
 - Menghubungkan aktifitas-aktifitas tersebut
- **Dengan mendetailkan ketiga hal tersebut maka kita dapat membangun model yang representatif.**
- **Activity cycle diagram adalah salah satu cara untuk dapat memodelkan interaksi dari entiti-entiti dan sangat berguna khusus untuk sistem yang mempunyai banyak struktur antrian**

Activity Cycle Diagram

- Activity cycle diagram dibentuk dari dua simbol seperti pada gambar dibawah ini. Diagram ini akan menggambarkan bagaimana sejarah hidup suatu entiti selama simulasi secara grafis. Tiap classes dari entiti mempunyai suatu life cycle yang terdiri dari sederatan state, dimana entiti bergerak dari satu state ke state lainnya



Activity Cycle Diagram

- **Active state :**
 - menggambarkan suatu kerjasama antar entiti yang berbeda.
 - lama waktu aktivitas tersebut adalah ditentukan atau terjadwalkan yang biasanya dilakukan secara sampling dari suatu distribusi probabilitas tertentu jika sistemnya bersifat stochastic.
 - proses pelayanan merupakan active state karena melibatkan dua entiti yaitu customer dan server dan lama prosesnya dapat ditentukan
- **Dead State :**
 - suatu keadaan dimana tidak terjadi kerjasama antara dua atau lebih entiti berbeda.
 - aktifitas menunggu lama waktunya tidak dapat dijadwalkan sehingga kita tidak tahu sampai kapan menunggu tersebut selesai, karena selesainya proses menunggu adalah bergantung pada kondisi sistem misalnya pada sistem antrian customer akan menunggu sampai ada server yang idle.
 - contoh, customer yang menunggu (antri) dilayani pada sistem antrian

Activity Cycle Diagram

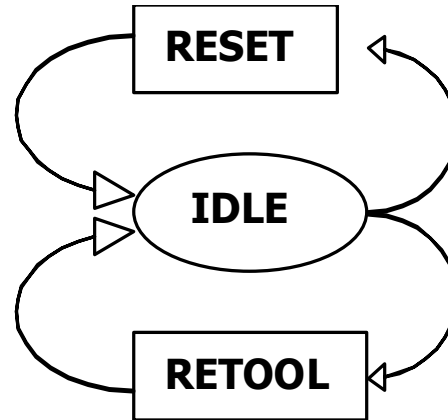
Contoh : Bengkel Permesian Sederhana

- Misalkan suatu bengkel mesin terdiri dari beberapa mesin yang identik. Tiap mesin bertipe 'general purpose' yang dapat memproses seluruh job yang datang. Job yang datang diproses pada mesin yang tersedia pertama kali. Waktu prosesnya adalah variable tapi tidak bergantung pada jenis mesin. Bengkel ini mempunyai satu orang operator yang bertugas menangani baik mesin maupun jobnya sendiri.
- Tugas operator tersebut adalah :
 - Melakukan RESET mesin jika proses telah selesai dan pahat mesin dalam keadaan baik
 - Mengganti pahat mesin jika pahatnya telah aus.
- Sistem ini terdiri dari tiga classes entiti :
 - Operator
 - Mesin
 - Job

Activity Cycle Diagram

Activity Cycle Diagram Operator

- Operator bertugas mengganti pahat (RETOOL) dan RESET mesin untuk pengerjaan job berikutnya. Berdasarkan informasi ini maka diagram dari operator adalah pada gambar berikut ini :

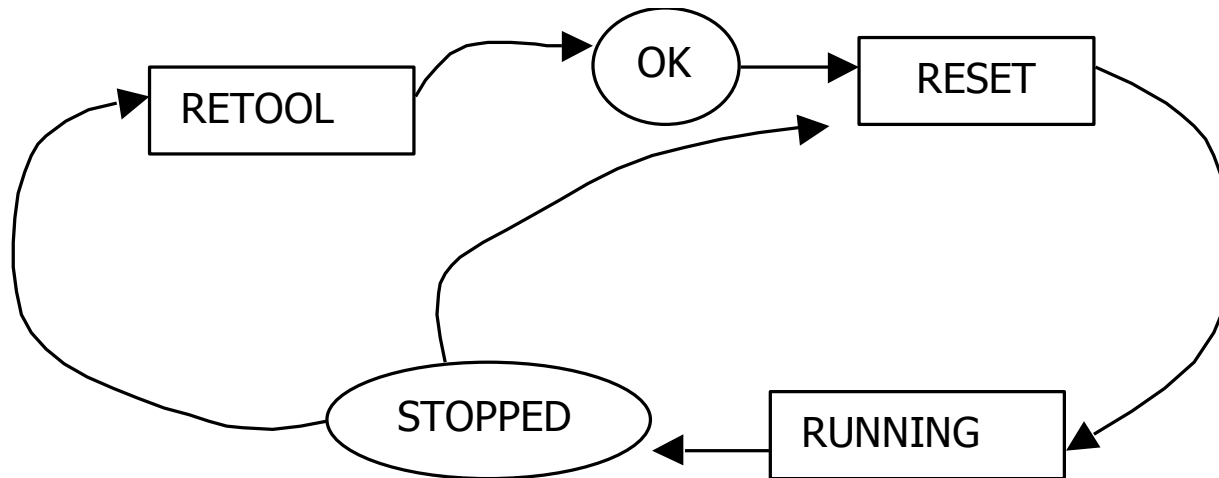


- Diagram diatas terdiri dari dua active state : RESET dan RETOOL . Lama waktu dari aktivitas tersebut ditentukan dengan melakukan sampling dari distribusi probabilitas tertentu

Activity Cycle Diagram

Activity Cycle Diagram Mesin

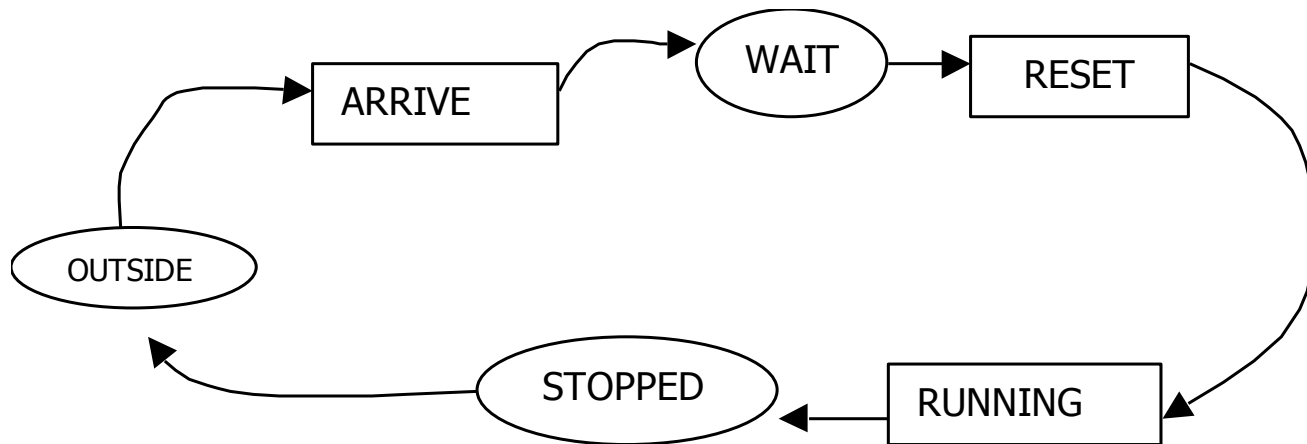
- Mesin bekerja sebagai berikut : setelah mesin tersebut selesai memproses suatu job (RUNNING) maka mesin tersebut menuju ke state berhenti (STOPPED) untuk menunggu pelayanan operator. Jika operator telah ada (dalam keadaan idle) maka operator akan melakukan penggantian pahat (RETOOL) terlebih dahulu jika pahatnya telah aus jika tidak maka operator langsung melakukan RESET pada mesin untuk mengerjakan job berikutnya



Activity Cycle Diagram

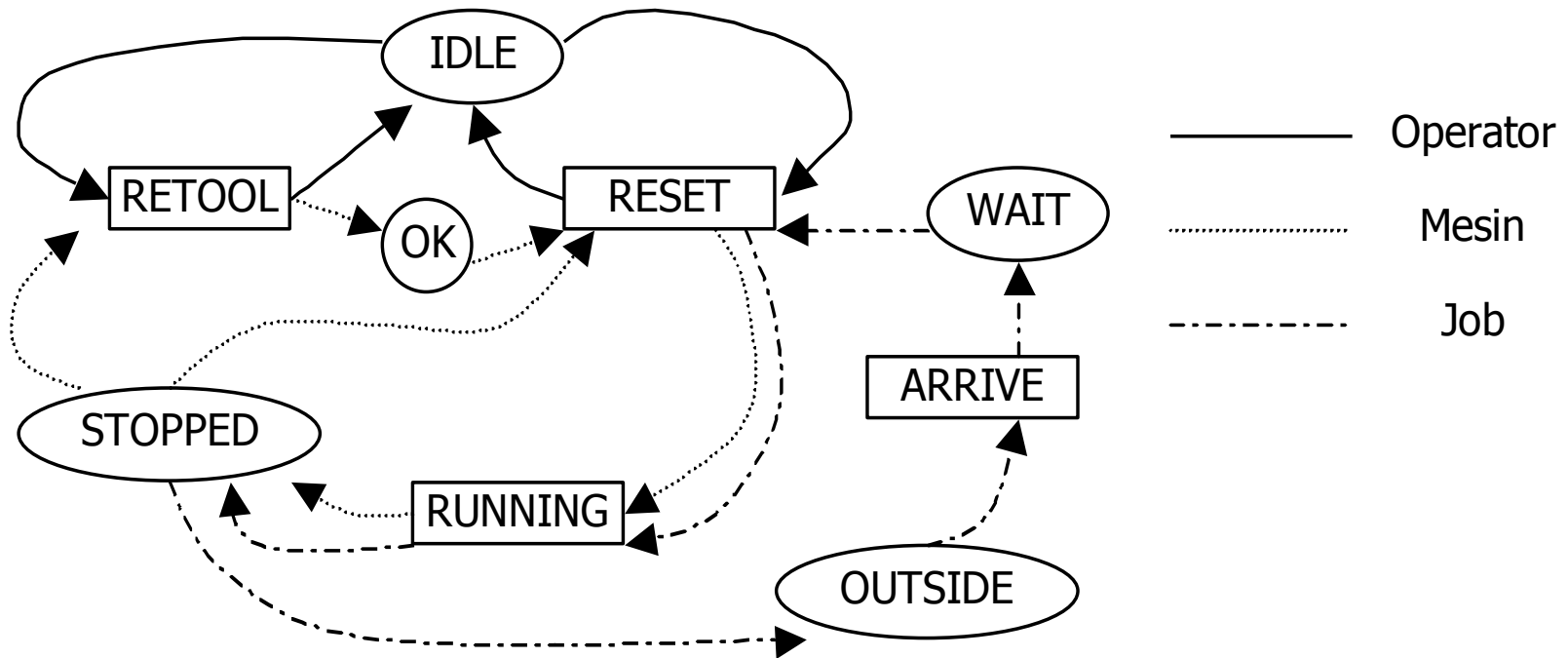
Activity Cycle Diagram Job

- Job datang dari luar dengan waktu antar kedatangan berdistribusi tertentu. Job yang datang akan menunggu sampai ada mesin yang idle dan operator juga dalam keadaan idle



Activity Cycle Diagram

Activity cycle diagram lengkapnya:



ACD Sistem Penjualan Tiket

Contoh: Sistem Penjualan Tiket

- Suatu gedung bioskop sehari-harinya mempekerjakan seorang petugas. Petugas ini bertugas untuk menjual tiket dan menjawab pesanan yang masuk. Kursi dapat dipesan hanya jika pembeli tiket datang dan membayar harga tiketnya. Pemesanan bisa datang dari orang yang datang ke bioskop tersebut ataupun melalui telepon. Petugas tersebut diintruksikan untuk memberi prioritas pada pengunjung yang datang langsung. Sehingga jika telepon pemesan berdering pada saat ada pengunjung yang datang maka yang datang ini yang dilayani terlebih dahulu. Telepon yang masuk dan belum terjawab akan antri secara FIFO, dimana diasumsikan tidak ada penelepon yang membatalkan teleponnya karena tidak dijawab

ACD Sistem Penjualan Tiket

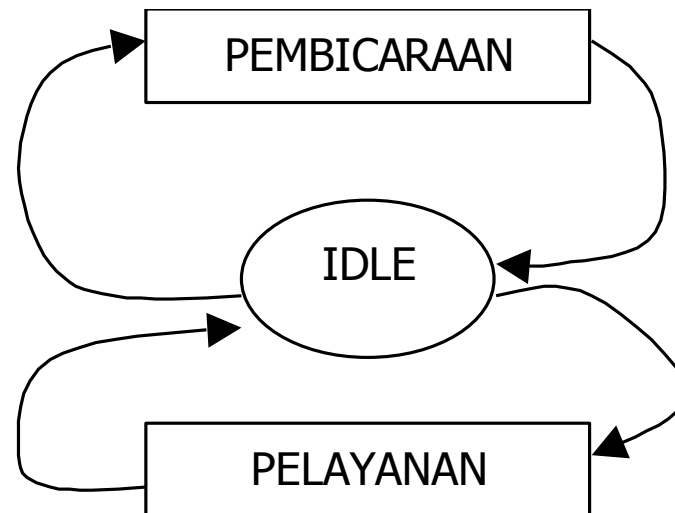
Contoh: Sistem Penjualan Tiket

- Suatu gedung bioskop sehari-harinya mempekerjakan seorang petugas. Petugas ini bertugas untuk menjual tiket dan menjawab pesanan yang masuk. Kursi dapat dipesan hanya jika pembeli tiket datang dan membayar harga tiketnya. Pemesanan bisa datang dari orang yang datang ke bioskop tersebut ataupun melalui telepon. Petugas tersebut diintruksikan untuk memberi prioritas pada pengunjung yang datang langsung. Sehingga jika telepon pemesan berdering pada saat ada pengunjung yang datang maka yang datang ini yang dilayani terlebih dahulu. Telepon yang masuk dan belum terjawab akan antri secara FIFO, dimana diasumsikan tidak ada penelepon yang membatalkan teleponnya karena tidak dijawab
- Dalam hal ini ada tiga class entiti:
 - petugas tiket
 - pengunjung langsung
 - panggilan telepon

ACD Sistem Penjualan Tiket

Petugas tiket

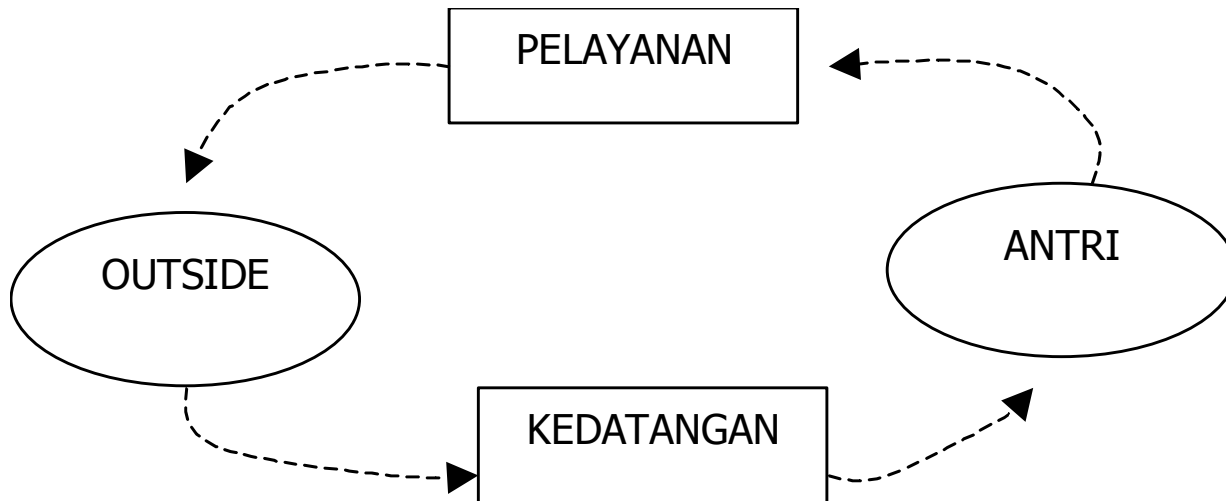
- **Petugas ini mempunyai dua active state :**
 - PELAYANAN: Melayani konsumen yang datang langsung
 - PEMBICARAAN: Berbicara dengan konsumen yang memesan lewat telepon
- **Jika tidak sedang melakukan kedua aktivitas tersebut maka petugas tersebut berada dalam keadaan dead state yaitu IDLE. Sehingga activity cycle diagram untuk petugas dapat dilihat pada gambar dibawah ini**



ACD Sistem Penjualan Tiket

Pengunjung Langsung

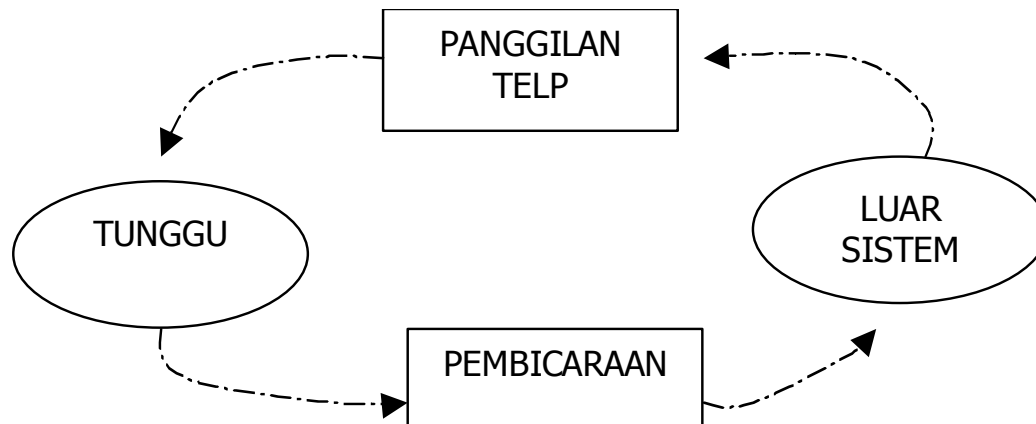
- Pengunjung langsung datang dari suatu populasi dengan waktu antar kedatangan beridistribusi tertentu. Setelah KEDATANGAN dan masuk ke sistem maka konsumen tersebut ANTRI sampai petugas dalam keadaan idle, kemudian baru bisa dimulai PELAYANAN dan setelah selesai maka konsumen tersebut keluar dari sistem.
- Activity cycle diagram untuk pengunjung langsung ini dapat kita lihat pada gambar berikut ini



ACD Sistem Penjualan Tiket

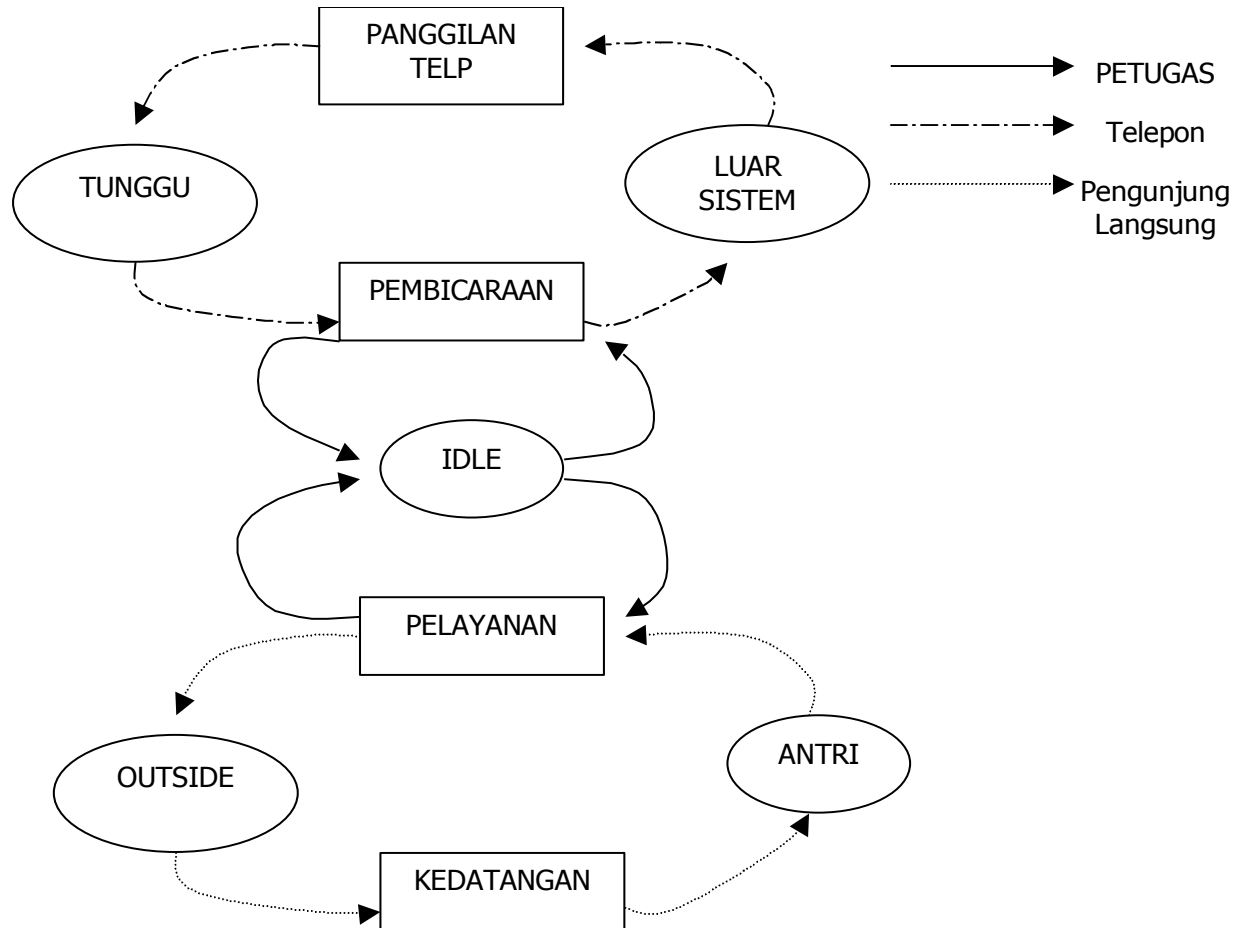
Panggilan Telepon

- Siklus aktivitas dari entiti ini adalah paralel dengan entiti konsumen langsung. Kedatangan pemesan lewat telepon adalah berupa panggilan telepon yang masuk kedalam sistem yang mana harus antri sampai panggilan telepon tersebut dijawab. Diagram siklus aktivitasnya digambarkan seperti nampak pada gambar 3.8. Disini ada dua active state yaitu PANGGILAN TELP dan PEMBICARAAN. Active state PANGGILAN TELP berasal dari luar lingkungan sistem dimana penelepon tersebut berasal. Aktivitas PEMBICARAAN dilakukan secara bersama-sama dengan entiti petugas



ACD Sistem Penjualan Tiket

Activity Cycle Diagram Sistem Penjualan Tiket :



Menggunakan Activity Cycle Diagram

- Activity cycle diagram disamping untuk menggambarkan secara sederhana interaksi antar entiti dalam sistem juga dapat digunakan pada tahapan awal dalam membangun model simulasi diskrit yaitu menentukan event-event apakah yang ada dalam model
- Tiap aktivitas dalam diagram adalah diawali dan diakhiri oleh sebuah event. Sehingga untuk contoh job shop diatas event-event yang dapat merubah variabel state system adalah :

ARRIVAL BEGIN	RESET BEGIN	RUNNING BEGIN	RETOOL BEGIN
ARRIVAL END	RESET END	RUNNING END	RETOOL END

- **Sebagai contoh event RESET BEGIN akan menyebabkan :**
 - operator menjadi tidak idle lagi
 - antrian mesin berhenti yang membutuhkan RESET berkurang satu
- **atau event ini akan terjadi jika:**
 - ada operator idle
 - ada mesin antri yang membutuhkan RESET

Menggunakan Activity Cycle Diagram

- Dari kedelapan event diatas ada dua diantaranya yang kejadiannya selalu bersamaan yaitu pada active state ARRIVAL yang mempunyai dua event yaitu ARRIVAL BEGIN dan ARRIVAL END. Dimana ARRIVAL END untuk job ke N akan terjadi bersamaan dengan ARRIVAL BEGIN untuk job ke N+1, sehingga kedua event tersebut sebaiknya dijadikan satu yaitu cukup event ARRIVAL saja. Dengan demikian total jumlah event adalah 7 buah event.
- Penentuan event ini adalah suatu hal yang sangat penting dalam simulasi discrete event karena jam simulasi (simulation clock) digerakkan kedepan dari event ke event. Satu kesulitan yang mungkin akan timbul adalah jika beberapa event harus terjadi pada waktu yang bersamaan, yakni beberapa operasi harus berjalan secara paralel. Padahal karakteristik dari komputer pada umumnya adalah mampu melakukannya secara serial bukan paralel

Menggunakan Activity Cycle Diagram

- **Masalah ini dapat diatasi dengan membuat program simulasi yang mampu melakukan dalam dua tahapan sebagai berikut:**
 - Program menggerakkan simulasi ke waktu dari perubahan event berikutnya. Kemudian jam simulasi ditahan atau dihentikan sementara.
 - Operasi-operasi yang terjadi juga pada saat ini juga dilakukan dengan urutan yang sederhana.
- **Setelah semuanya selesai maka jam simulasi selanjutnya dijalankan seperti semula.**
- **Sehingga dengan demikian operasi serial dapat dilakukan untuk mensimulasikan proses paralel**