



Desain Software

Arna Fariza
PENS-ITS



Materi

- ☐ Apakah desain software itu?
- ☐ Apakah modularisasi itu?



Apakah Desain Software itu?

- ☐ **Desain** adalah proses mengubah kebutuhan sistem ke dalam produk yang lengkap
- ☐ **Desain Software** adalah output dari proses yang
 - o Dekomposisi sistem dalam beberapa bagian
 - o Menentukan tanggung jawab setiap bagian
 - o Memastikan bagian-bagian tersebut dapat mencapai tujuan global
- ☐ Desain terdiri dari aktifitas dan hasil dari aktifitas
- ☐ Desain adalah aktifitas iteratif
 - o Mulai dari titik inisial
 - o Berkembang dan beriterasi dengan menambahkan fitur/kapabilitas
 - o Mencapai state yang tepat berdasarkan
 - Obyektif/tujuan desain
 - Input/feedback dari konsumen

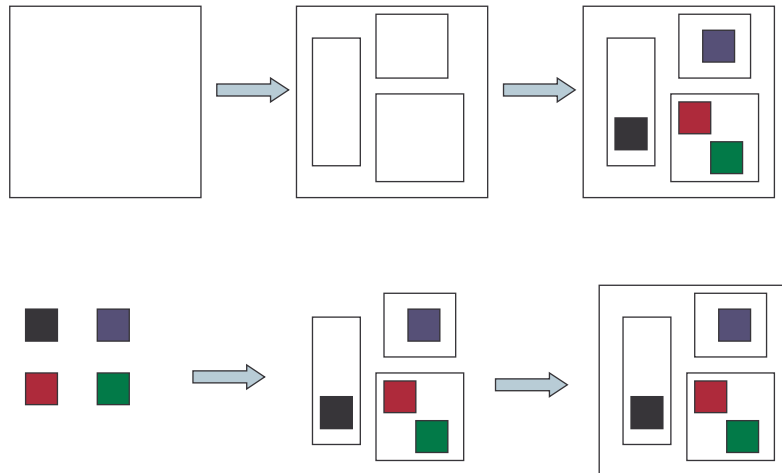


Apakah Desain Software itu?

- ☐ Software dikembangkan secara terinci sampai cukup detail untuk mendukung implementasi
 - o Sistem dibagi ke dalam modul- modul (komponen, paket, class dll)
 - o Dekomposisi Top-Down
 - Modul besar (komponen, paket, class) didekomposisi ke dalam bagian yang lebih kecil
 - Bagian yang lebih kecil tsb mengimplementasikan bagian yang lebih besar
 - Bagian yang lebih besar dikomposisi dari bagian yang lebih kecil
 - o Dekomposisi Bottom-Up
 - Modul level yang lebih rendah didefinisikan terlebih dahulu
 - Modul-modul tsb dikombinasikan untuk membentuk modul yang lebih besar
- ☐ Baik TD atau BU dapat mencapai hasil yang sama



Contoh Desain TD dan BU



Apakah Desain Software itu?

- ❑ Desain Software adalah
 - o Aktivitas yang menjembatani antara kebutuhan dan implementasi software
 - o Aktivitas yang memberikan struktur benda yang dihasilkan
 - Dokumen spesifikasi kebutuhan harus didesain
 - Struktur dapat memudahkan untuk mengerti dan mengembangkannya
- ❑ Aktivitas Desain Software
 - o Menentukan dekomposisi sistem ke dalam modul
 - o Menghasilkan dokumen desain software
 - Menggambarkan dekomposisi sistem ke dalam modul-modul
 - o Biasanya *Arsitektur Software* dihasilkan sebelum desain software



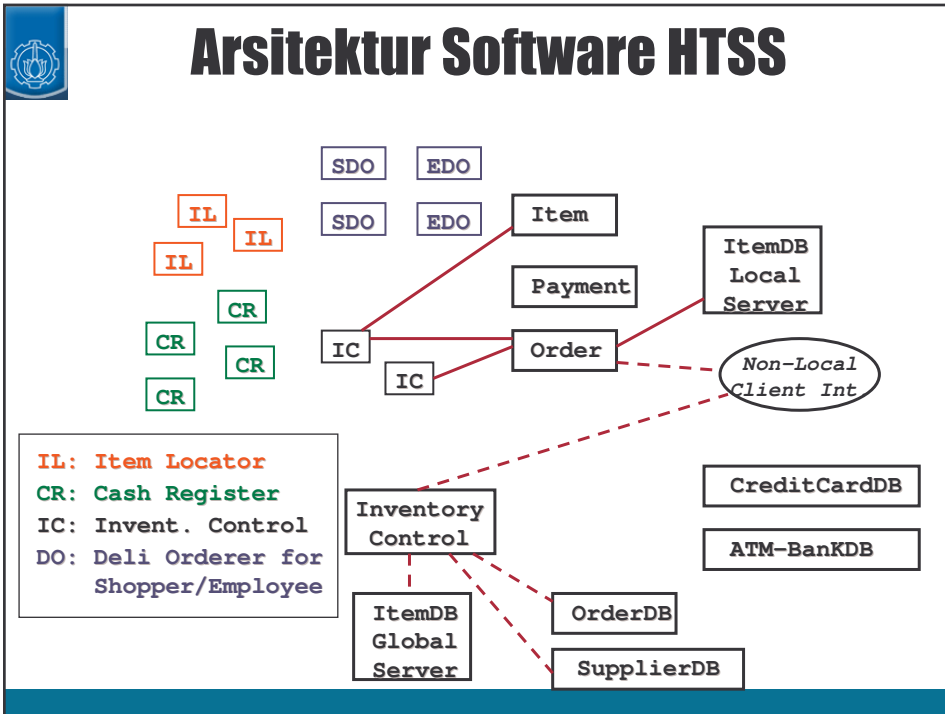
Apakah Arsitektur Software itu?

- ❑ Arsitektur Software
 - o Bentuk kasar dari sistem yang ditentukan
 - o Terdiri dari deskripsi
 - Komponen utama sistem
 - Relasi antar komponen
 - Alasan dekomposisi ke dalam komponen-komponen
 - Batasan yang harus dipatuhi dalam mendesain komponen
 - o Petunjuk pengembangan desain



Contoh Arsitektur Sistem High-Tech Supermarket System (HTSS)

- ❑ Otomatisasi fungsi dan aksi
 - o Update kasir dan inventori
 - o Meletakkan bahan makanan secara user friendly
 - o Pelacakan delivery order dengan cepat
 - o Mengontrol inventory
- ❑ Antar muka sistem user
 - o Scanner Cash Register
 - o GUI untuk kontrol Inventory
 - o Shopper menghubungkan Locator dan Orderer
 - o Antar muka deli untuk pekerja delivery order



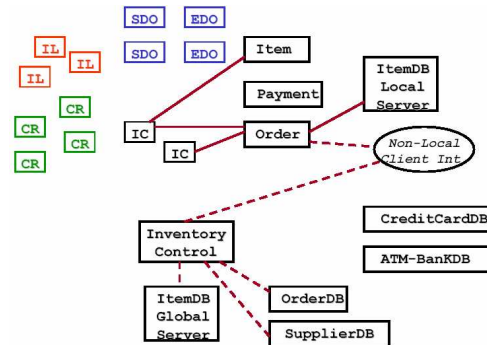
Apakah Modularisasi itu?

- ❑ Modularisasi mengungkit ide dari arsitektur software
 - o Arsitektur software mendekomposisi sistem ke dalam komponen yang berpengaruh
 - o Biasanya, komponen tersebut merupakan program independent terhadap platform hardware
 - o Modularisasi mendekomposisi komponen tersebut ke dalam modul-modul yang berpengaruh
- ❑ Modul:
 - o Adalah komponen Well-defined (terdefinisi dengan baik) dari sistem software
 - o Bagian dari sistem yang menyediakan layanan ke modul-modul lain
 - Layanan adalah elemen komputasi yang tersedia untuk digunakan oleh modul lain
 - Modul memberikan layanan dan mengambil layanan



Contoh Modularisasi

- ❑ Memanggil komponen Cash Register dari HTSS
- ❑ CR terdiri dari modul
 - o UPC Scanner
 - o Menyimpan Item yg terjual
 - Mengurangi inventory
 - Menambah ke Nota
 - Menambah profil konsumen
 - o Pembayaran
 - Kupon
 - Cash/Debit/Credit
 - o Nota
- ❑ Modul-modul tersebut berinteraksi dengan komponen HTSS lain



Spesifikasi Software

Arna Fariza
PENS-ITS



Materi

- ☐ Apakah Spesifikasi itu?
- ☐ Spesifikasi Operasional
 - o Data Flow Diagrams, Finite State Machines, Petri Nets, Entity Relationship Diagrams
 - o UML Diagrams (Briefly - Revisit in Total)



Apakah Spesifikasi itu?

- ☐ Spesifikasi adalah versi yang lebih luas dari ketentuan definisi
 - o Digunakan pada tingkat yang berbeda untuk pengembangan software untuk tujuan yang berbeda
 - o Pernyataan persetujuan (kontrak) antara
 - Produsen dan konsumen suatu servis
 - Implementer dan user
 - o Semua kualitas yang diperlukan harus ditentukan



Apakah Spesifikasi itu?

- ☐ Spesifikasi bisa berbeda untuk stakeholder yang berbeda (user, desainer, manager, dll)
- ☐ Spesifikasi kebutuhan:
 - o Persetujuan antara End User dan Desainer
- ☐ Spesifikasi Desain:
 - o Persetujuan antara Desainer dan Developer
- ☐ Spesifikasi Modul:
 - o Persetujuan antara SE yang menggunakan modul dan SE yang mengimplementasikan modul termasuk antar muka
- ☐ Spesifikasi Berorientasi Obyek
 - o Pernyataan kemampuan class menggunakan public interface
- ☐ Spesifikasi menyatakan “Apa”
- ☐ Implementasi menyatakan “Bagaimana”

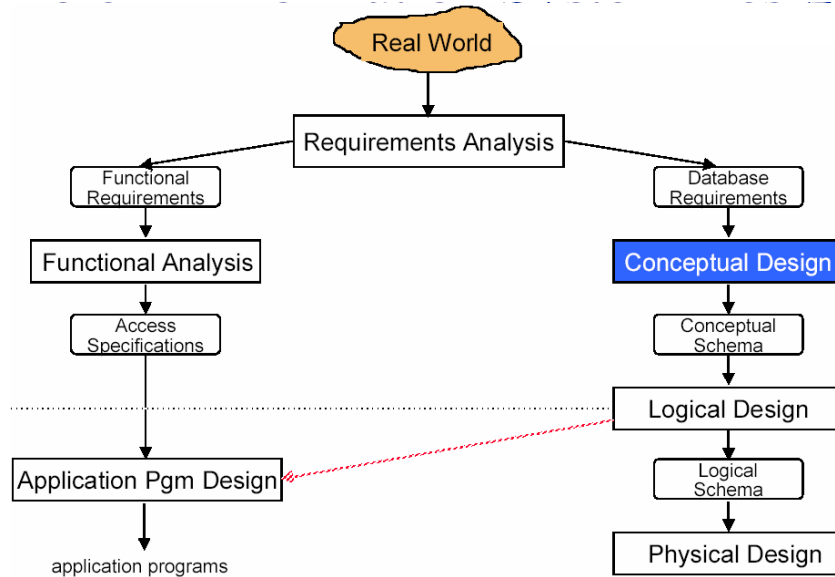


Penggunaan Spesifikasi

- ☐ Pernyataan apa yang diperlukan user
 - o Pernyataan dari antar muka antara mesin dan lingkungan terkontrol
- ☐ Isu kritis:
 - o Kebutuhan user mungkin tidak dimengerti oleh Developer
 - o Perlu adanya verifikasi spesifikasi
- ☐ Pernyataan kebutuhan implementasi
 - o Hardware, OS, Bahasa pemrograman, DBMS, dll
 - o Spesifikasi kebutuhan
 - Perilaku eksternal dari sistem
 - Perilaku fungsional dan non fungsional
 - Verifikasi spesifikasi desain vs spesifikasi kebutuhan
 - o Spesifikasi Desain
 - Menggambarkan arsitektur software
 - Kode harus diverifikasi vs spesifikasi desain



Desain Sistem Informasi Klasik



Spesifikasi Operasional

- ☐ Bagaimana perilaku sistem didefinisikan
- ☐ Beberapa Teknik
 - o Data Flow Diagrams
 - o UML Diagrams
 - o Finite State Machines
 - o Petri Nets
- ☐ Spesifikasi operasional menentukan pengertian sistem model
 - o Perspektif harus konsisten satu dengan lainnya
 - o Beberapa teknik targetnya untuk End-User/Konsumen



Data Flow Diagrams (DFD)

- ☐ Merupakan spesifikasi operasional Semi-Formal
- ☐ Sistem dipandang sebagai manipulasi data dengan “fungsi”
- ☐ Data bersifat persistent - disimpan dalam repositori
- ☐ State input merepresentasikan triger dari aliran data
- ☐ State output merepresentasikan hasil aliran data
- ☐ Data dapat mengalir dari input ke fungsi ke/dari repositori ke output
- ☐ DFD mempunyai notasi grafis
 - o Perangkat tersedia secara komersial



Notasi Grafis untuk DFD

- ☐ Bagaimana konstruksi pemodelan utamanya?
 - o Lingkaran merepresentasikan fungsi
 - o Panah merupakan aliran data (data flow)
 - o Kotak terbuka menyatakan penyimpanan persisten
 - o Kotak tertutup merupakan interaksi I/O
- ☐ Catatan: DFD tidak dapat merepresentasikan langkah sequensial



The function symbol



The data flow symbol



The data store symbol



The input device symbol

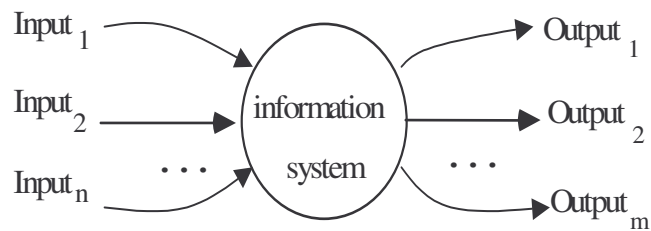


The output device symbol



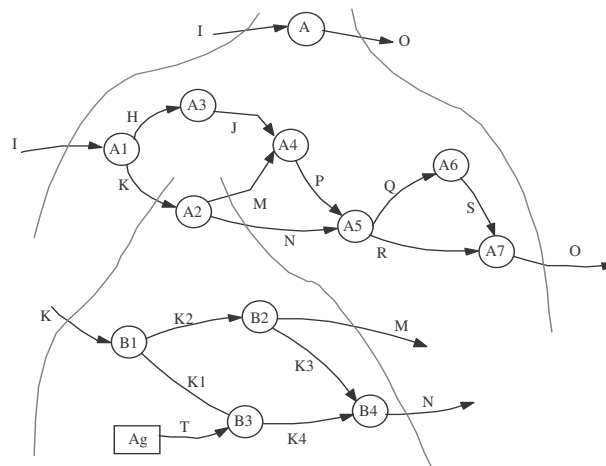
Metodologi Sistem Informasi

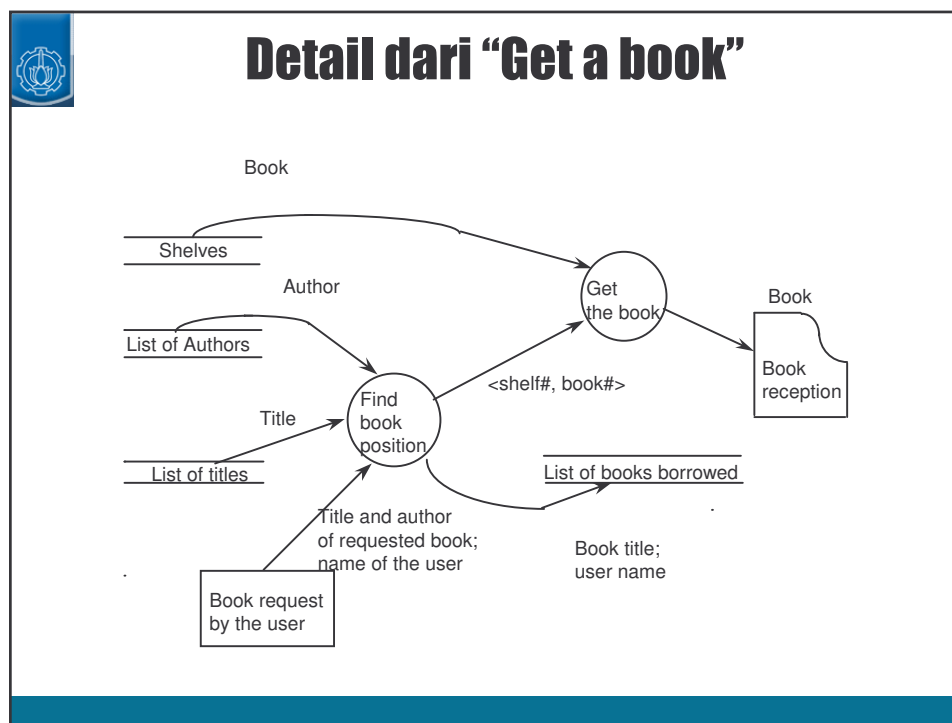
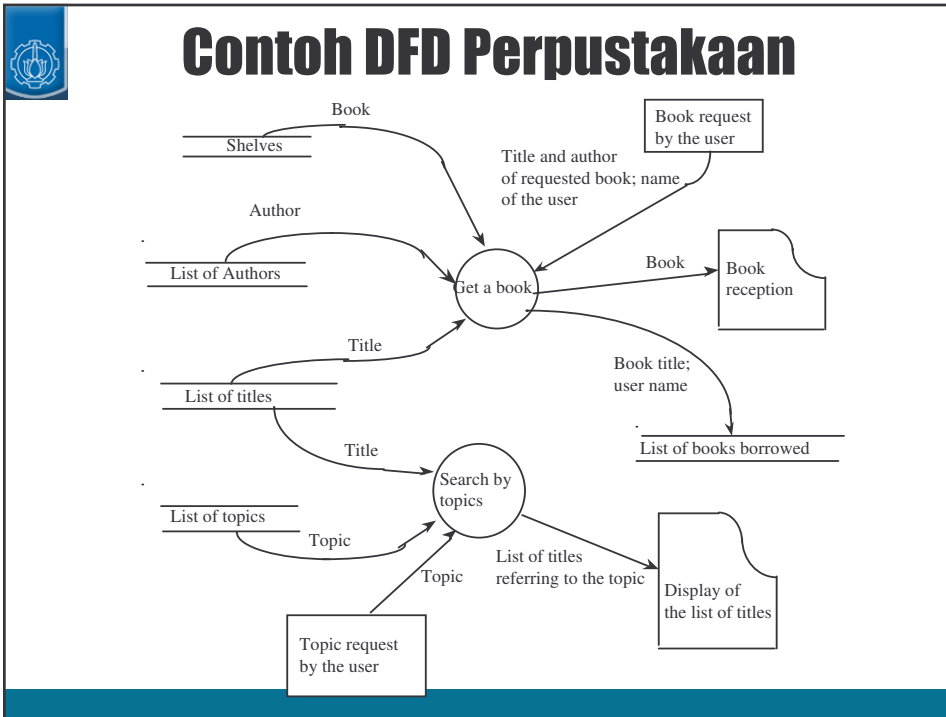
1. Mulai dari diagram context



Metodologi Sistem Informasi

2. Proses ditentukan sampai mencapai fungsi elementer

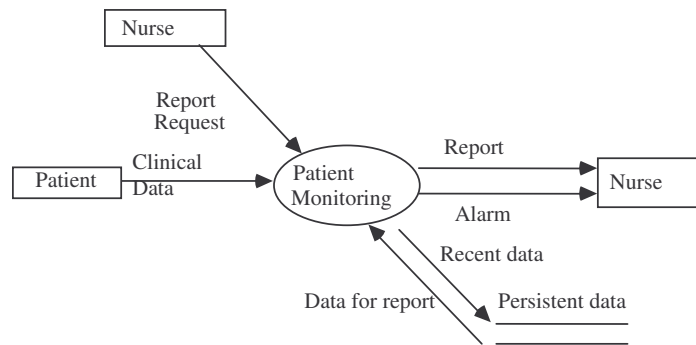




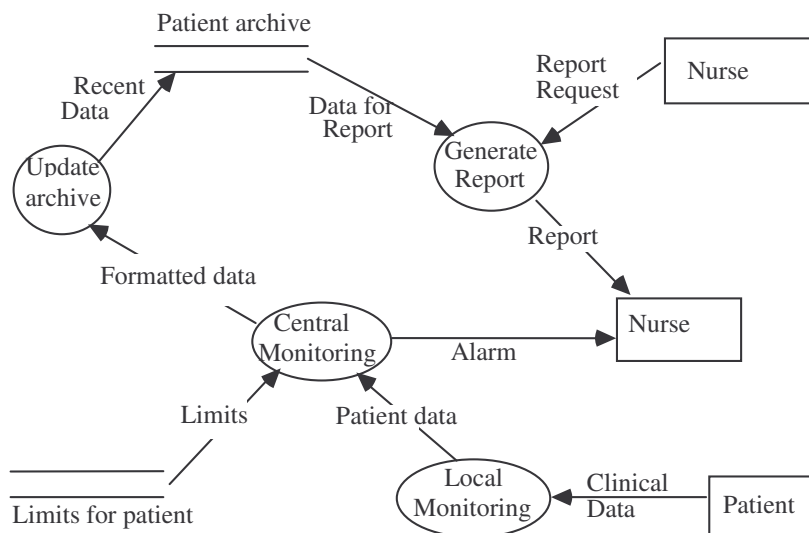


Sistem Monitoring Pasien

Tujuannya adalah untuk memonitor faktor vital dari pasien - darah, tekanan, suhu,...-- membacanya dalam waktu tertentu dari perangkat analog dan menyimpan dalam DB. Jika pembacaan gagal karena diluar jangkauan untuk pasien atau peralatan gagal maka alarm harus dikirim ke perawat. Sistem juga menyediakan laporan.

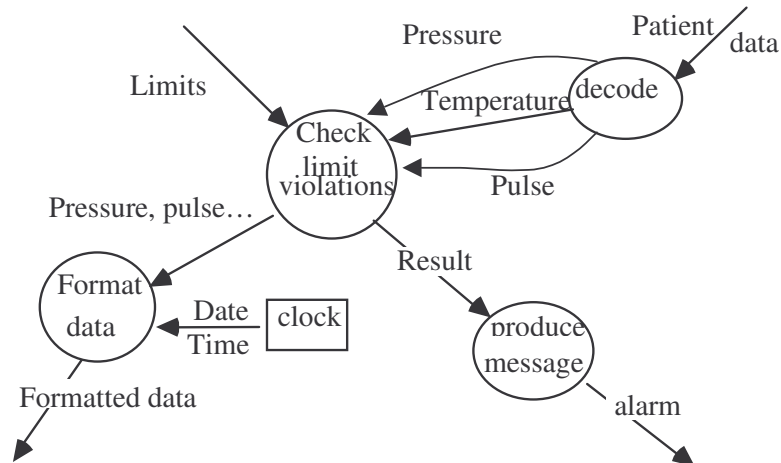


DFD Detail





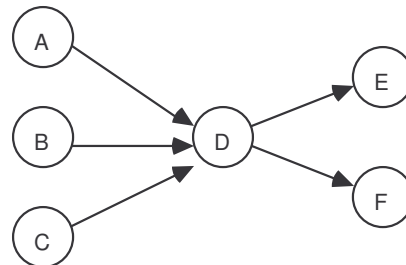
Lebih Detail Lagi



Evaluasi DFDs

- ☐ Mudah dibaca
- ☐ Semantik informal
 - o Bagaimana mendefinisikan fungsi akhir
 - o Kemenduaan aliran
- ☐ Misalnya DFD (fungsi) dibawah ini:

- Apakah output dari A, B, C semua diperlukan sebelum D di-enable?
- Bagaimana jika hanya ada A dan B?
- Apakah A, B, dan C diterima dengan urutan khusus?
- Output untuk E dan F

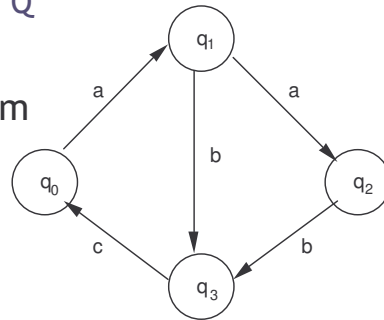


dibasilkan pada waktu yang sama?

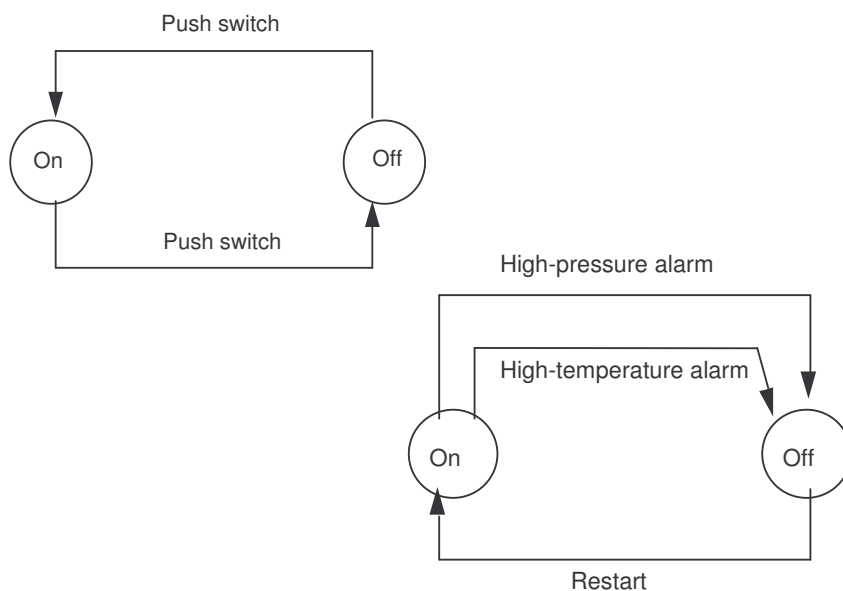


Finite state machine (FSM)

- ❑ Menentukan aspek kontrol flow
- ❑ FSM terdiri dari:
 - o Finite set of states (node), Q ;
 - o Finite set of inputs (label), I ;
 - o Fungsi transisi $d : Q \times I \rightarrow Q$
- ❑ FSM cocok untuk merepresentasikan sistem dengan
 - o Multiple Known States
 - o Well-Defined Events untuk perubahan state



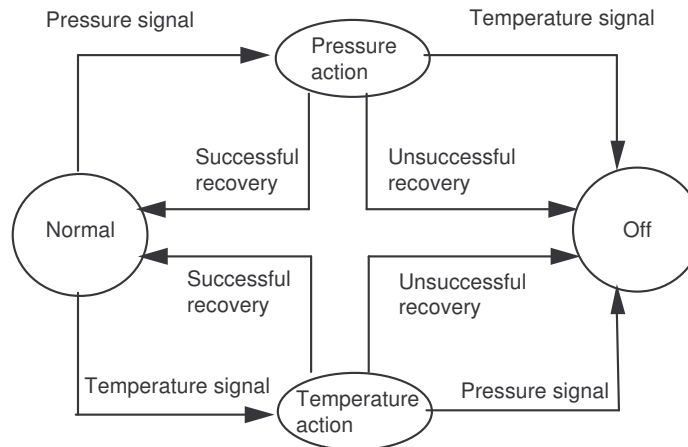
Contoh FSM Klasik





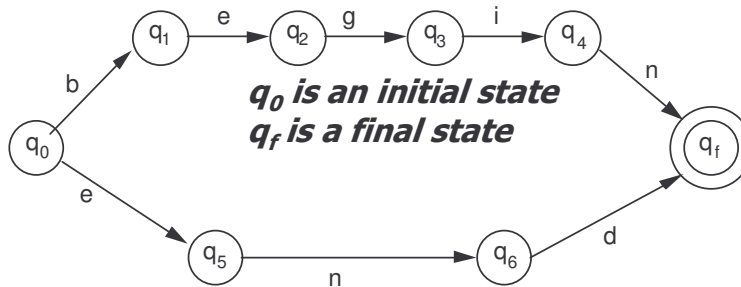
FSM dapat Memodelkan Real World

- ❑ Detail FSM untuk High Pressure/High Temperature pada slide sebelumnya



Class FSM

- ❑ Deterministic/Nondeterministic
- ❑ FSM sebagai Recognizer - Final State
- ❑ FSM
 - o Digunakan untuk analisis Lexical dalam Compiler
 - o Realisasi Regular Expression dalam Automata





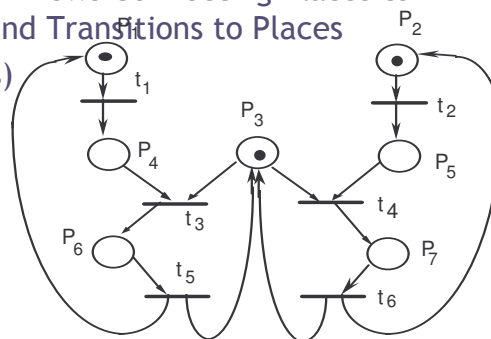
Penggunaan dan Keterbatasan FSM

- ❑ FSM sederhana dan digunakan secara luas
 - o Sistem kontrol, kompilasi
 - o Pattern Matching, Hardware Design
- ❑ Hampir semua aplikasi software dapat dimodelkan dengan FSM



Petri Net

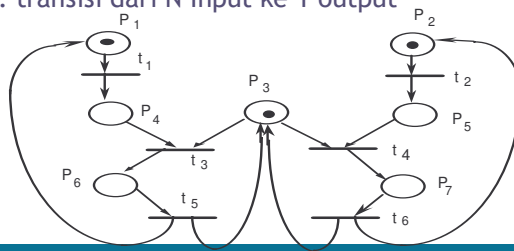
- ❑ Petri Net adalah bentuk grafis lain untuk spesifikasi sistem terdiri dari:
 - o Finite Set of Places (Circles - P_i 's)
 - o Finite set of Transitions (Horizontal Lines - t_j 's)
 - o Finite Set of Arrows Connecting Places to Transitions and Transitions to Places
 - o Tokens (Dots)



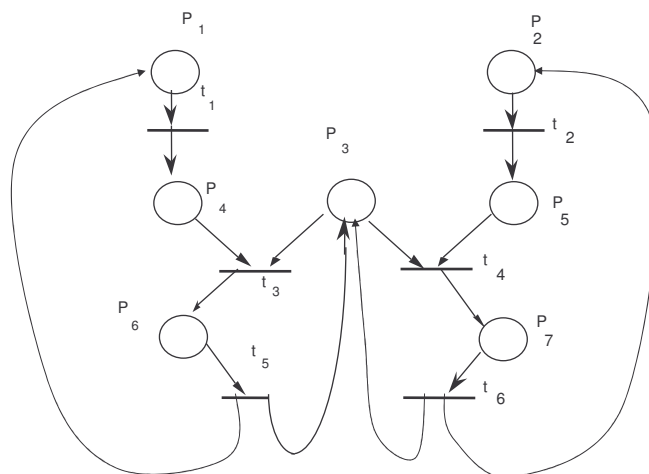


Pemodelan dengan Petri net

- ❑ Place merepresentasikan State terdistribusi
- ❑ Transisi merepresentasikan aksi atau event yang mungkin terjadi jika sistem berada pada state tertentu
- ❑ Dapat terjadi kondisi saat ini berada pada state
- ❑ Pemodelan fork dan join#
 - o Fork: transisi dari 1 input ke N output
 - o Join: transisi dari N Input ke 1 output



Representasi Grafis dari PN





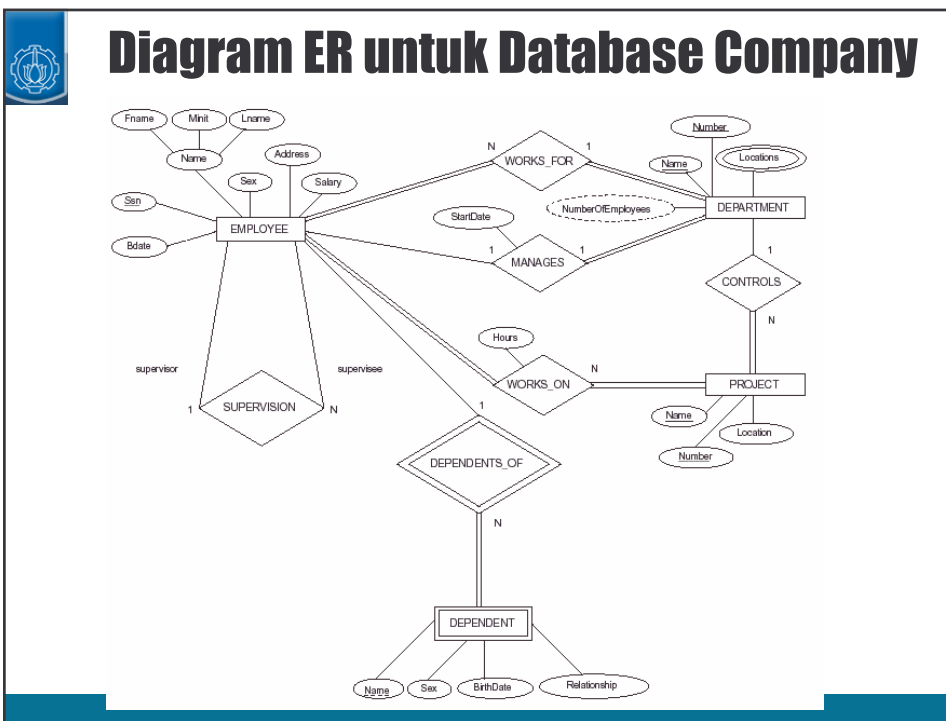
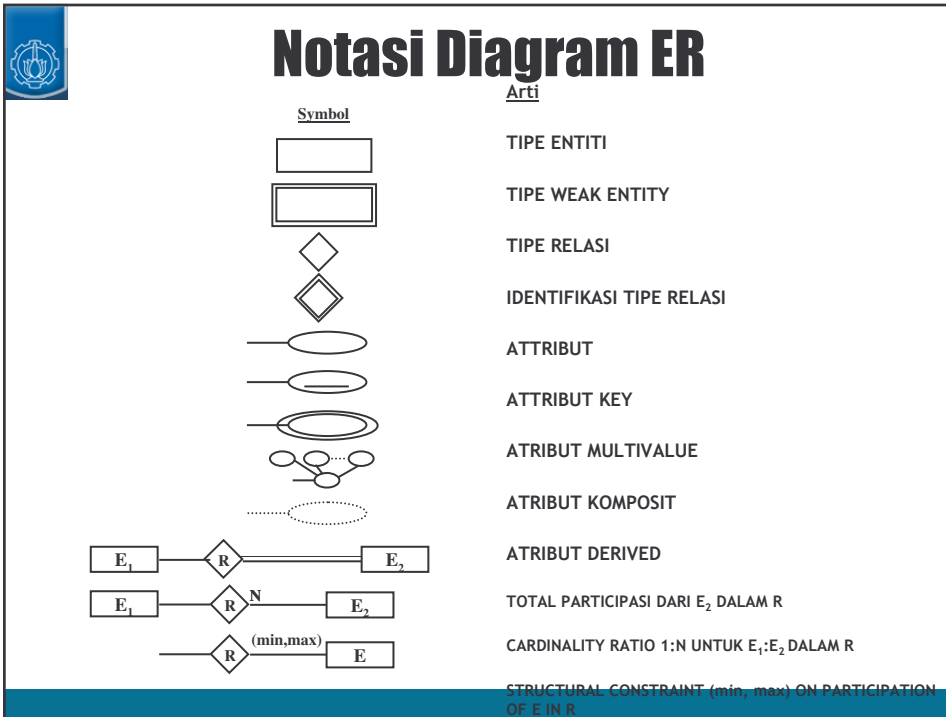
Keterbatasan Petri Net

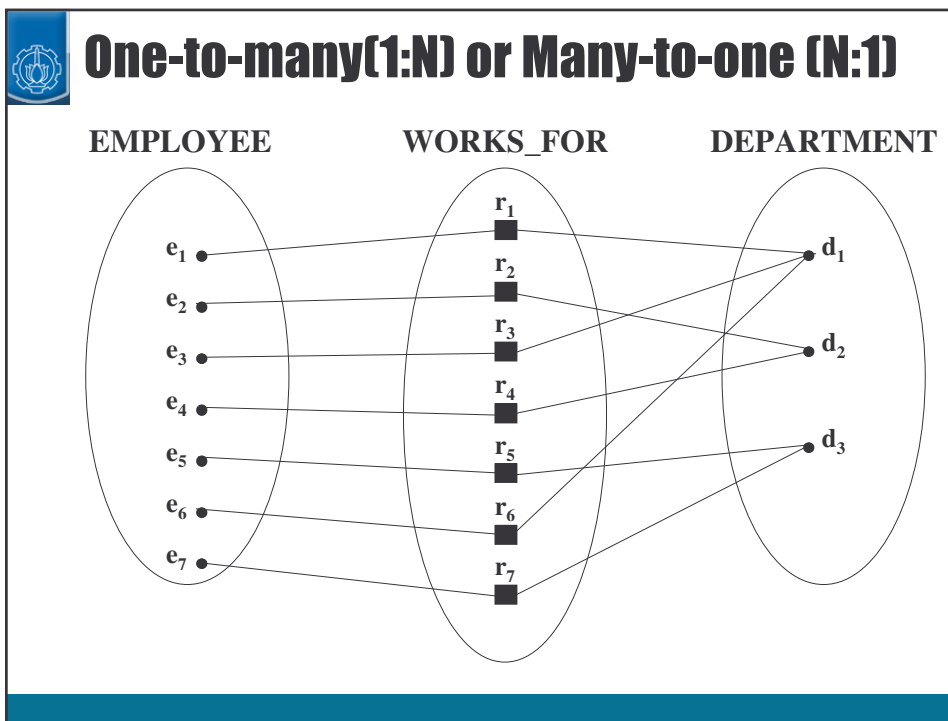
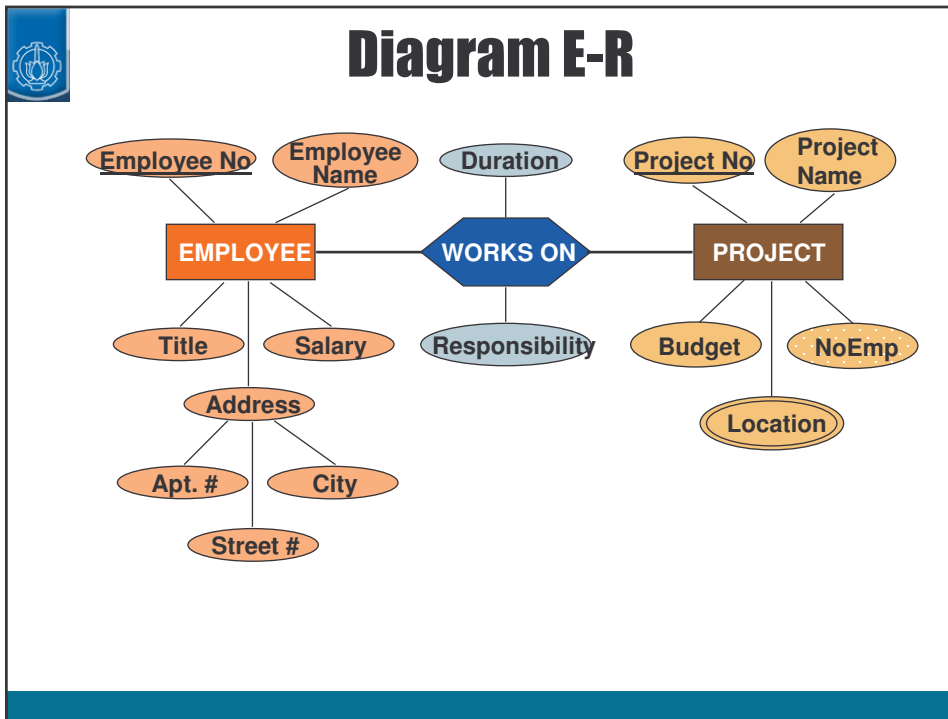
- ☐ Petri Net terbatas pada kapabilitas pemodelan
 - o Non-Deterministic
- ☐ Terdapat beberapa kapabilitas yang mungkin sangat berguna untuk memodelkan perilaku sistem
 - o Penambahan predikat dan fungsi yang digunakan untuk evaluasi kondisi dimana terjadi transisi dan hasilnya



Diagram Entity Relationship

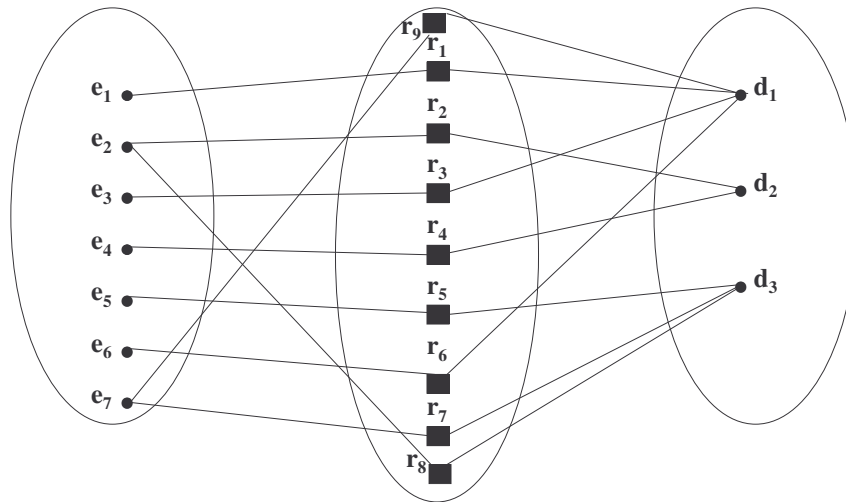
- ☐ Konsep ER Model
 - o Entiti dan atribut
 - o Tipe entiti, himpunan nilai dan atribut key
 - o Relasi dan tipe relasi
 - o Tipe weak entity
 - o Aturan dan atribut dan tipe relasi
- ☐ Relasi dari tingkat lebih tingg
- ☐ Notasi berdasarkan :
 - o R. Elmasri and S.B. Navathe, “ Fundamentals of Database Systems,” Ed. 3., Addison Wesley, 2000, Chapters 3 and 4.





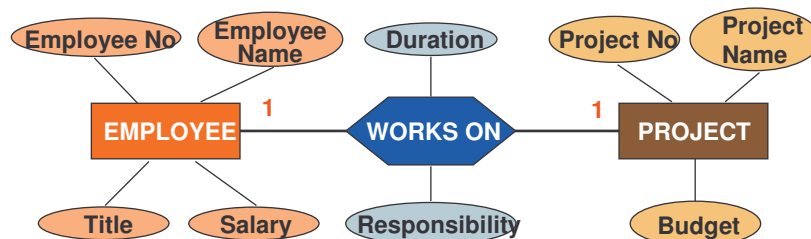


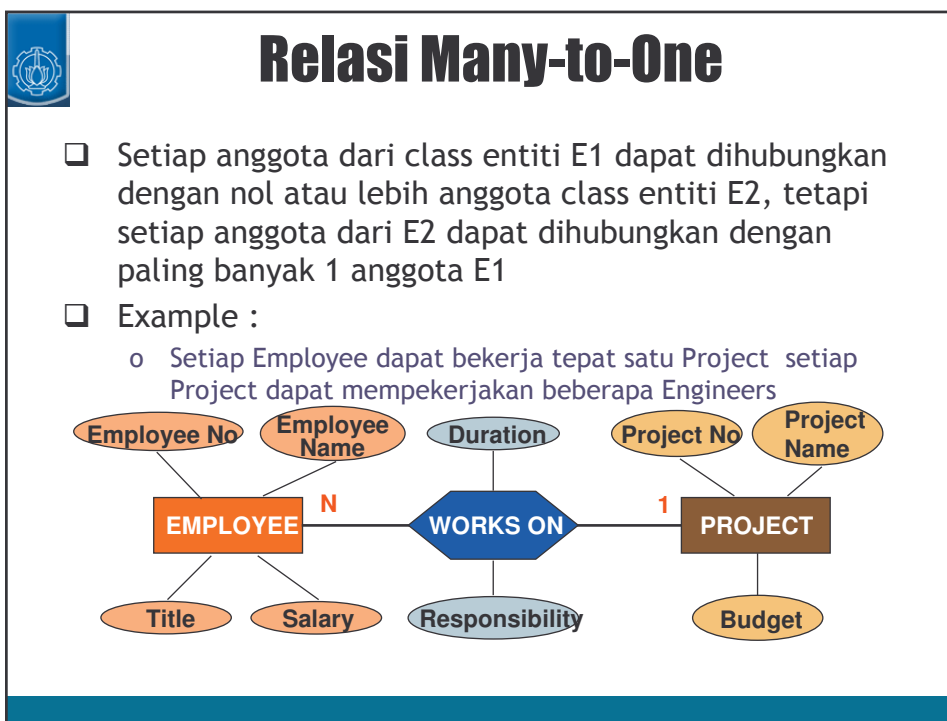
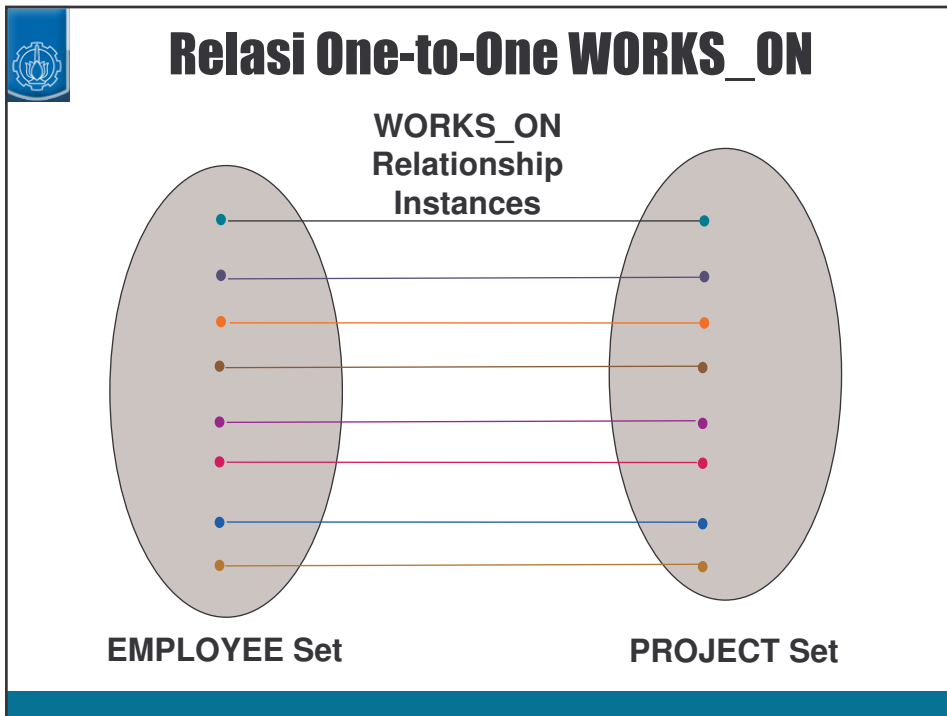
MANY-TO-MANY(M:N)



Relasi One-to-One

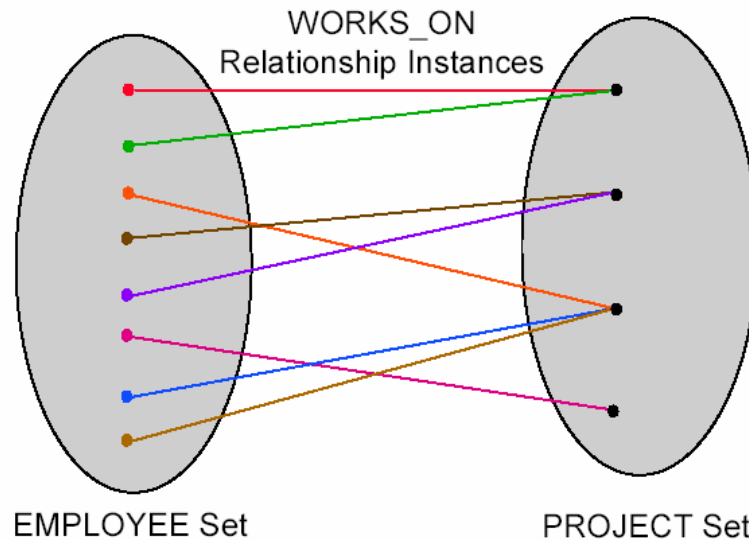
- ❑ Setiap anggota class entiti E1 dapat dihubungkan tepat satu anggota class entiti E2 dan sebaliknya.
- ❑ Contoh:
 - o Setiap Employee dapat bekerja tepat satu project dan setiap project mempekerjakan tepat satu Employee





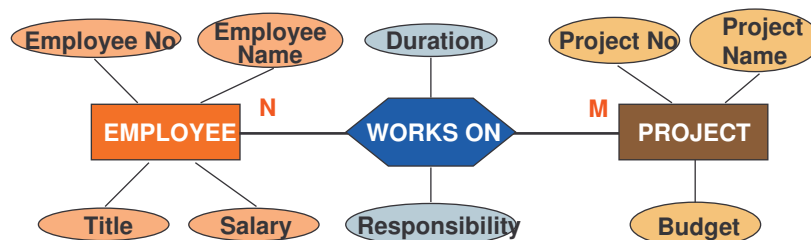


Relasi One-to-Many WORKS_ON



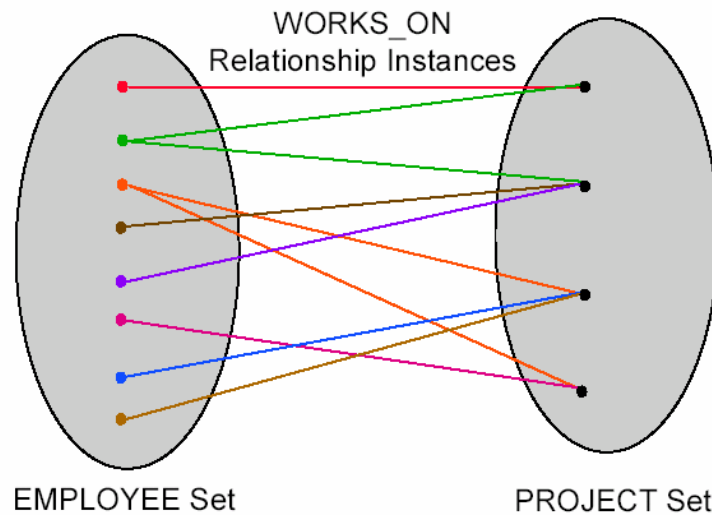
Relasi Many-to-Many

- ❑ Setiap anggota satu class entiti dapat dihubungkan dengan beberapa anggota dari class entiti lain dan sebaliknya
- ❑ Contoh:
 - o Setiap Employee dapat bekerja pada beberapa project
 - o setiap project dapat mempekerjakan beberapa Employees



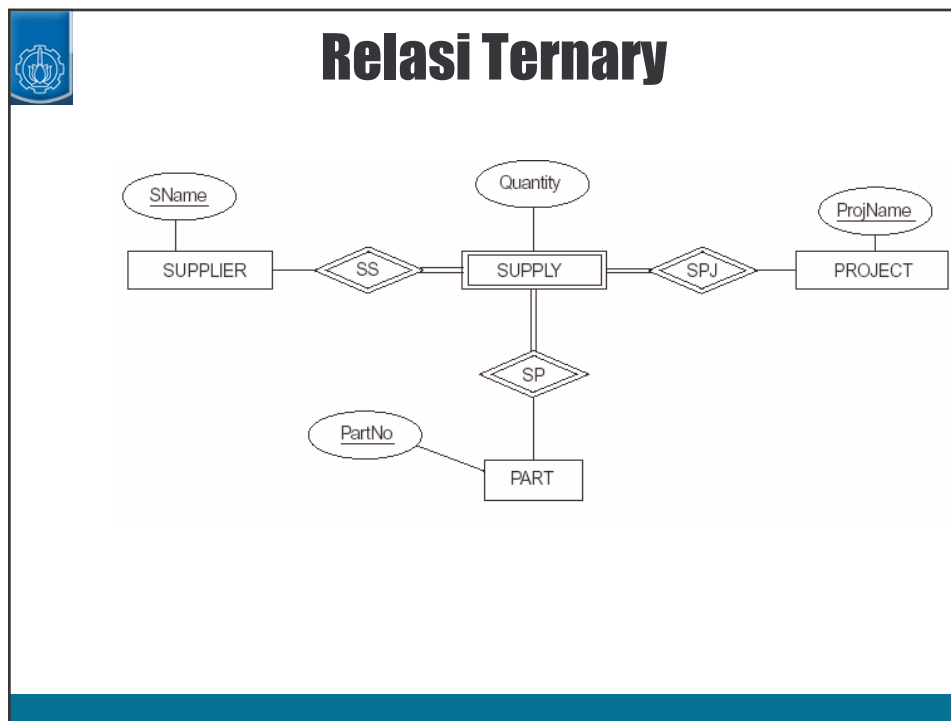
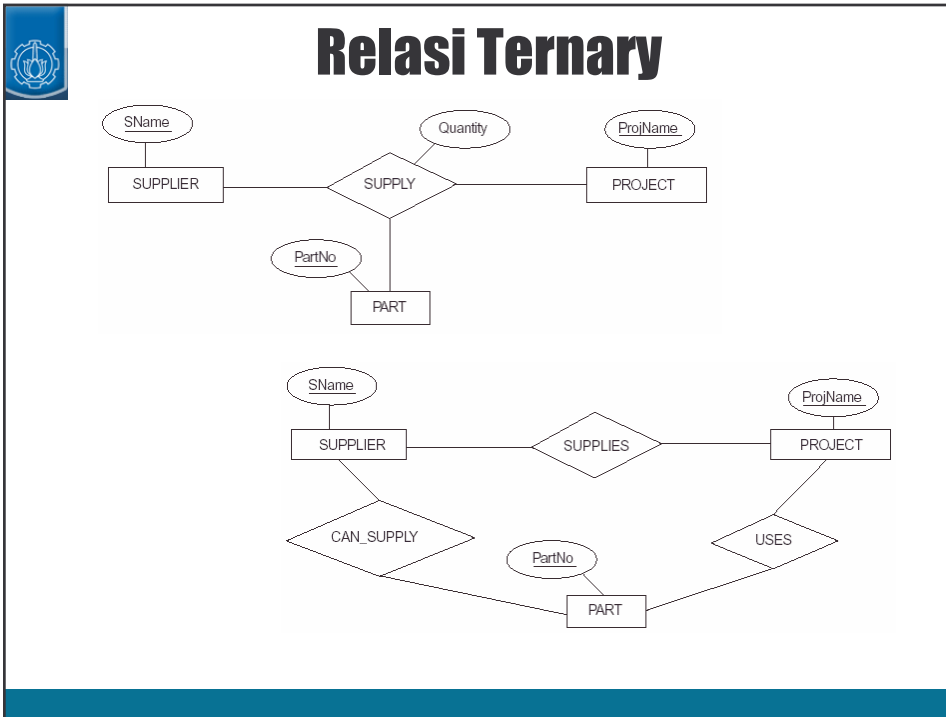


Relasi Many-to-Many WORKS_ON



Relasi Tingkat Lebih Tinggi

- ☐ Tipe relasi tingkat 2 disebut Binary
- ☐ Tipe relasi tingkat 3 disebut Ternary
 - o Terdapat anggota relasi yang terlibat pada ketiga tipe entiti
- ☐ Tipe relasi tingkat N disebut N-ary
- ☐ Secara umum, relasi N-ari tidak ekuivalen ke relasi binary N



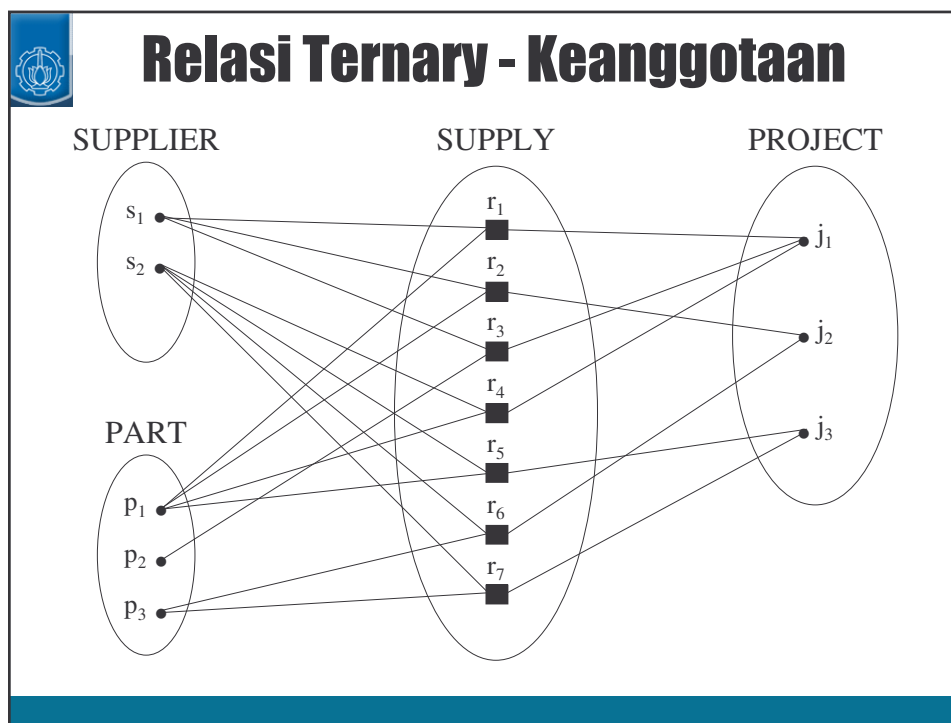
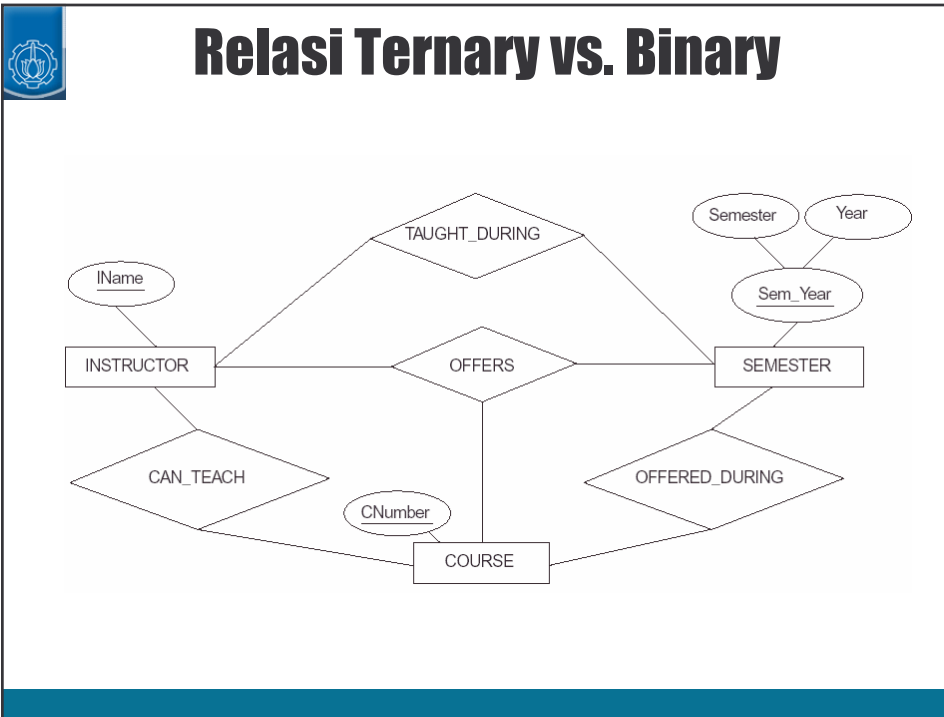




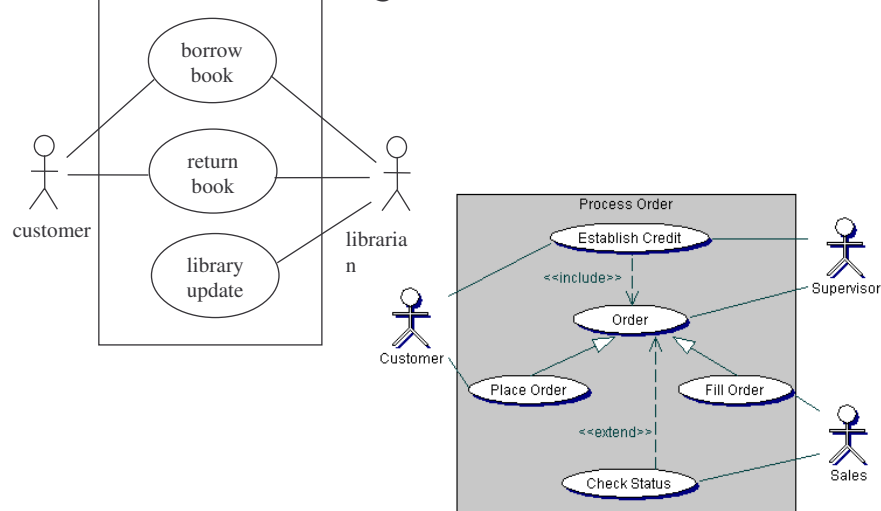
Diagram UML

- ❑ UML adalah bahasa untuk Spesifikasi, Visualisasi, Konstruksi dan Dokumentasi Alat Software
- ❑ UML diformalisasikan dari teknik sebelumnya (DFD, ER, FSM, PN, dll.) ke dalam lingkungan yang seragam (unified)
- ❑ UML mempunyai 13 diagram yang berbeda (2.0)



UML Diagram Use-Case

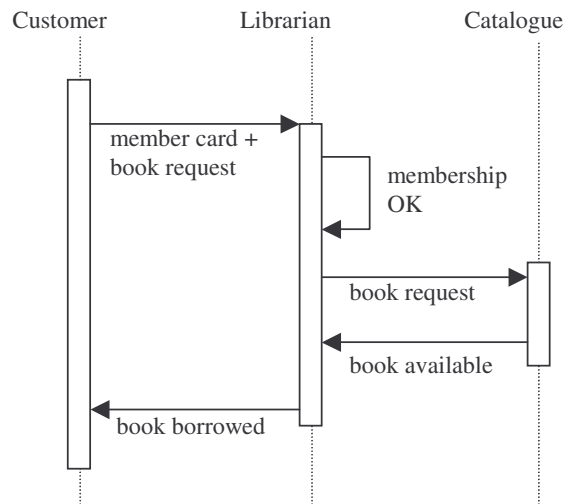
- ❑ Mendefinisikan fungsi berbasis aktor dan aksi



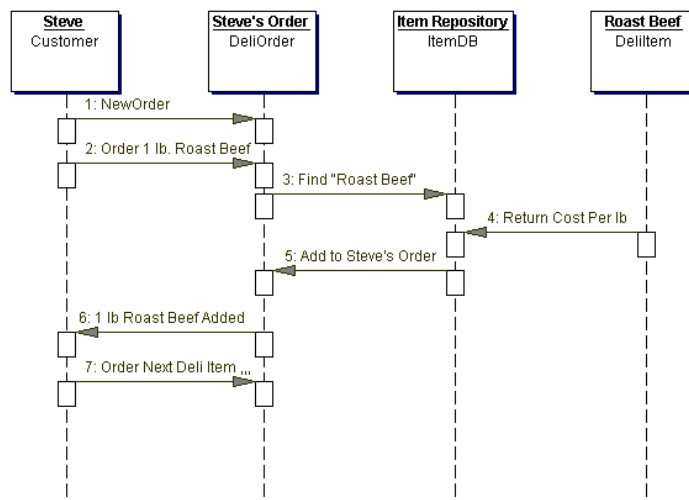


UML Diagram Sequence

- Menggambarkan interaksi obyek dengan pertukaran pesan



UML Diagram Sequence



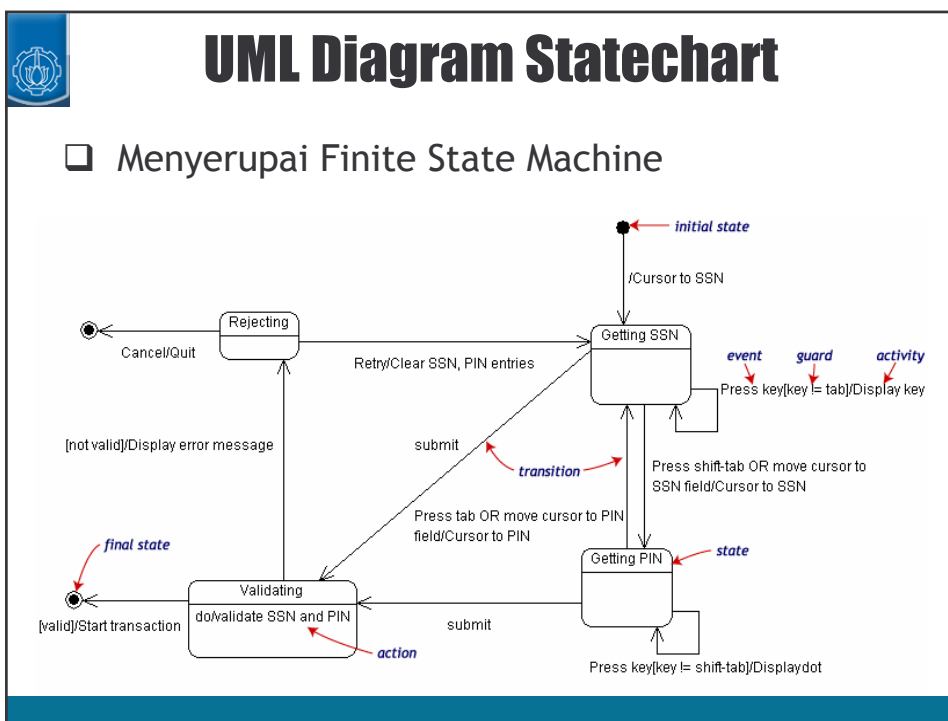
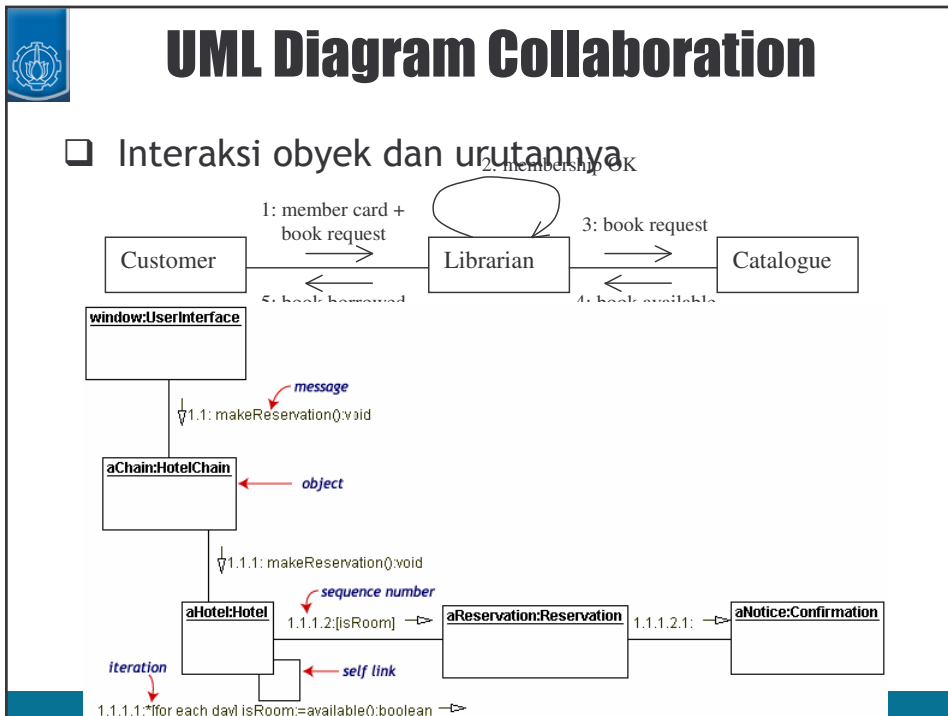
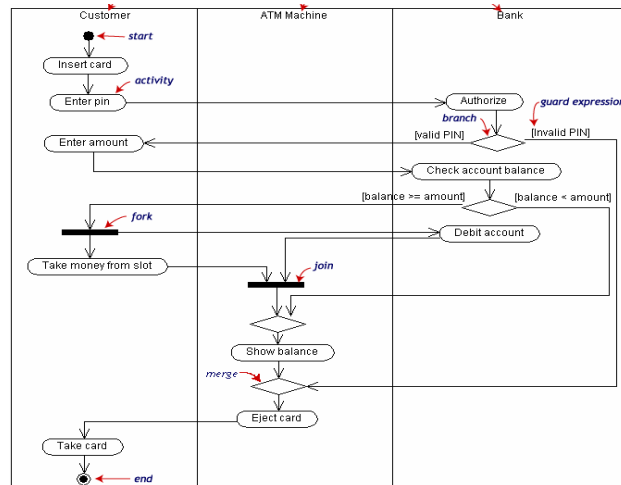




Diagram Activity

❑ Menyerupai Petri Net



Spesifikasi Berbasis Matematika

- ❑ Model Queueing dan Simulation
 - o Prediksi dan simulasi perilaku sistem
 - o CSE221
- ❑ Spesifikasi deklaratif:
 - o Spesifikasi logika
 - o Spesifikasi algebra
 - o CSE237
- ❑ Bahasa untuk Spesifikasi Modular
 - o Statechart dan Z