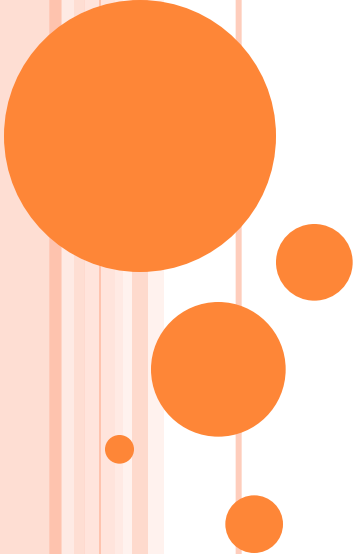


CONVENTIONAL METHODS FOR SOFTWARE ENGINEERING



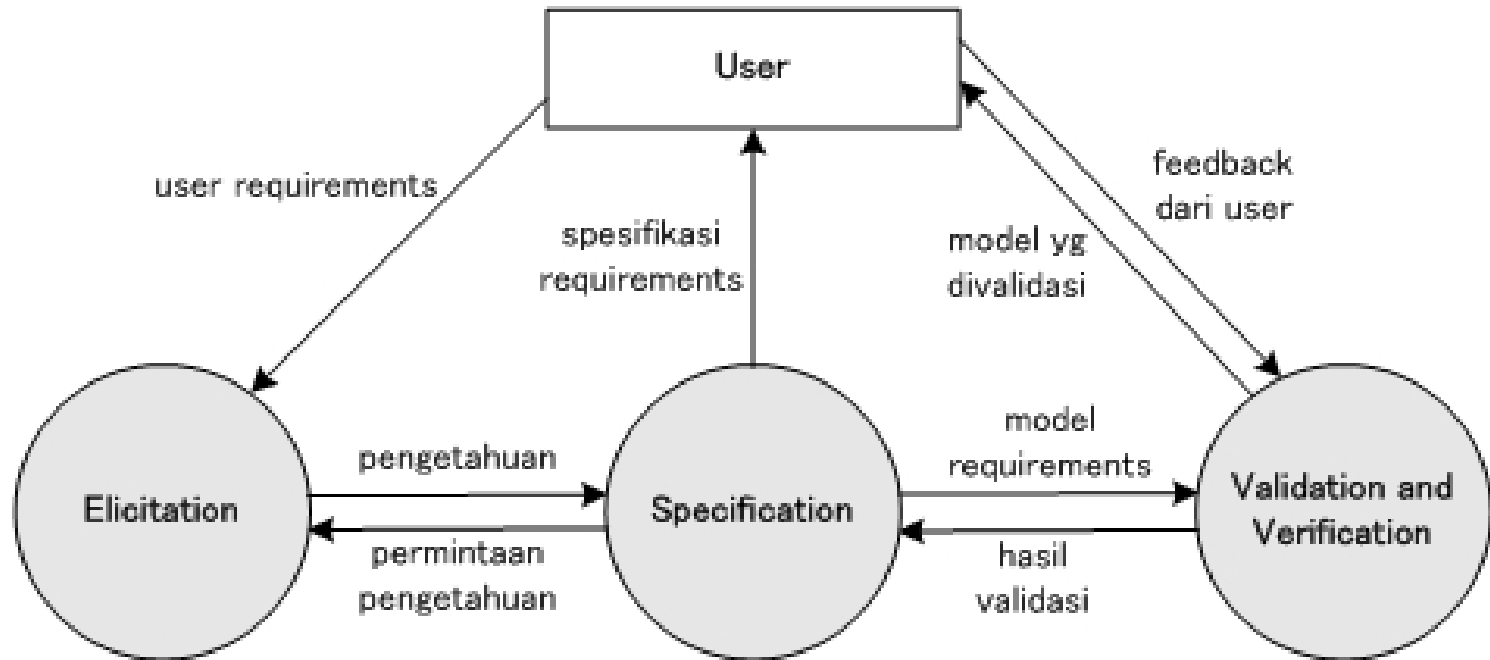
Overview of Requirements Analysis,
Requirements Analysis Methods,
Analysis Modeling Structure

REQUIREMENTS ENGINEERING

- Requirements engineering adalah fase terdepan dari proses software engineering, di mana software requirements (kebutuhan) dari user (pengguna) dan customer (pelanggan) dikumpulkan, dipahami dan ditetapkan.
- Kebanyakan kegagalan pengembangan software disebabkan karena adanya ketidakkonsistenan (inconsistent), ketidaklengkapan (incomplete), dan ketidakbenaran (incorrect) dari requirements specification (spesifikasi kebutuhan)



THE THREE DIMENSIONS OF REQUIREMENTS ENGINEERING



REQUIREMENTS ELICITATION

- Adalah proses mengumpulkan dan memahami (problem analysis & problem description) requirements dari user.
- Kadang masalah muncul berakar dari perbedaan disiplin ilmu yang dimiliki (gap knowledge domain).
- Customer adalah expert pada domain yang softwarena ingin dikembangkan (domain specialist) oleh pengembang (requirements analyst).
- Gap knowledge domain bisa diatasi dengan adanya interaksi terus menerus dan berulang (iterasi) antara pengembang dan customer.
- Proses interaksi tersebut kemudian dimodelkan menjadi beberapa teknik dan metodologi diantaranya adalah interviewing, brainstorming, prototyping, use case, dsb.





REQUIREMENTS ELICITATION

○ Techniques

- Interview / Meeting
- Survey / Questionnaire
- Observation
- Ethnography / Temporary Assignment
- Business Plans
- Review Internal / External Documents
- Review Software



REQUIREMENTS SPECIFICATION

- Mendeskripsikan masalah dalam bentuk dokumen spesifikasi dokumen.
- Berisi tentang fitur dan fungsi yang diinginkan oleh customer (tidak membahas bagaimana metode pengembangannya).
- IEEE mengeluarkan standard untuk dokumen spesifikasi requirements yang terkenal dengan nama IEEE Recommended Practice for Software Requirements Specifications.
- Dokumen spesifikasi requirements bisa berisi functional requirements, performance requirements, external interface requirements, design constraints, maupun quality requirements.

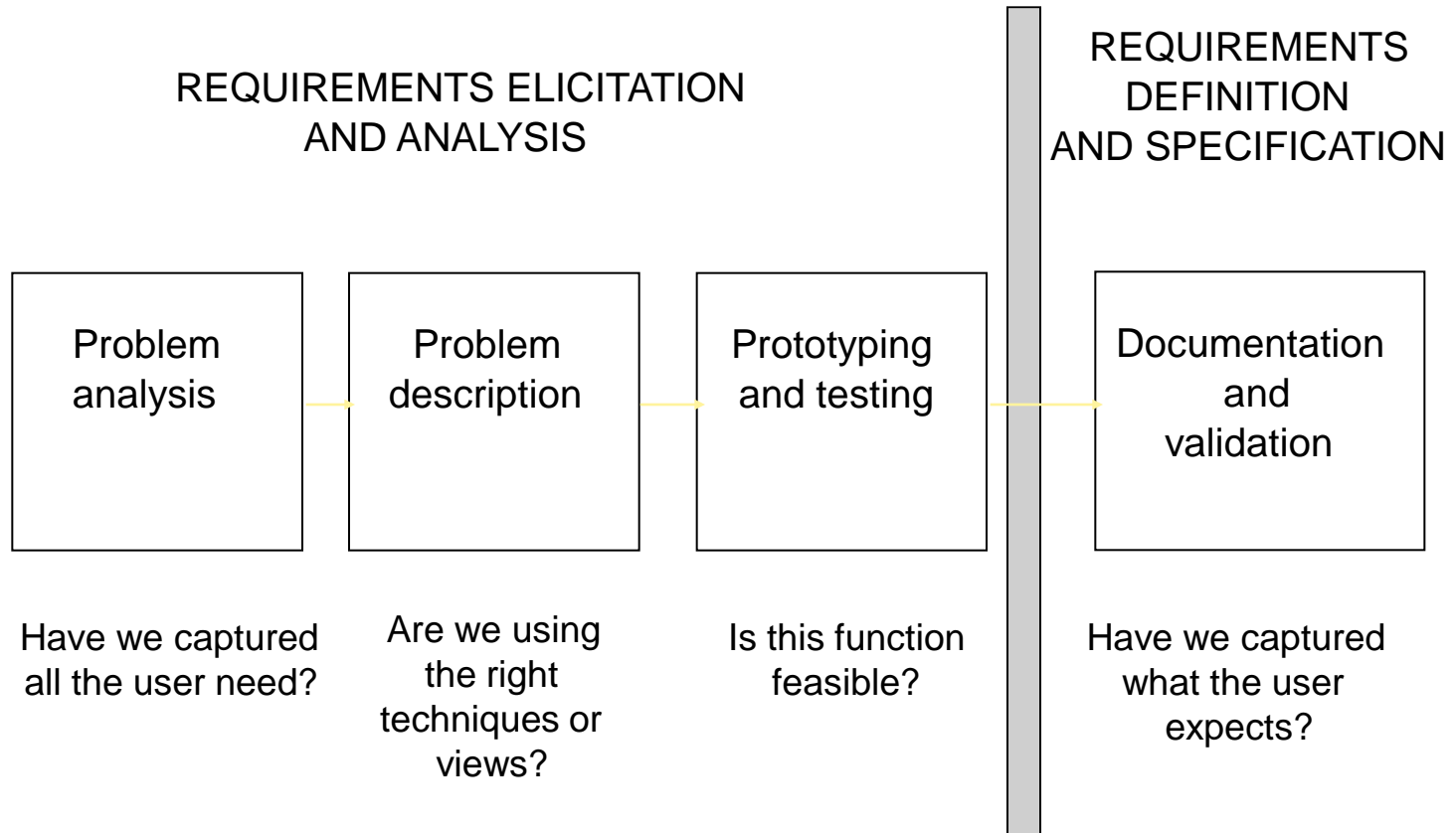


REQUIREMENTS VALIDATION AND VERIFICATION

- Validation (validasi), yaitu proses untuk memastikan bahwa requirements yang benar sudah ditulis.
- Verification (verifikasi), yaitu proses untuk memastikan bahwa requirements sudah ditulis dengan benar.
- Proses validasi dan verifikasi ini melibatkan customer (user) sebagai pihak yang menilai dan memberi feedback berhubungan dengan requirements



REQUIREMENTS ENGINEERING





REQUIREMENTS ELICITATION

○ Techniques

- Interview / Meeting
- Survey / Questionnaire
- Observation
- Ethnography / Temporary Assignment
- Business Plans
- Review Internal / External Documents
- Review Software





INFORMATION GATHERING

- Organization
- Processes
- Existing Systems
- Improvements




REQUIREMENTS ANALYSIS PRINCIPLES

- Domain informasi dari suatu masalah harus dipahami
- Fungsi-fungsi yang akan dilakukan oleh perangkat lunak harus didefinisikan
- Tingkah laku perangkat lunak harus direpresentasikan
- Model-model yang menggambarkan informasi, fungsi dan tingkah laku harus dipecah-pecah secara hirarki
- Proses analisis harus bergerak dari informasi dasar ke detail implementasi





REQUIREMENTS REVIEW?

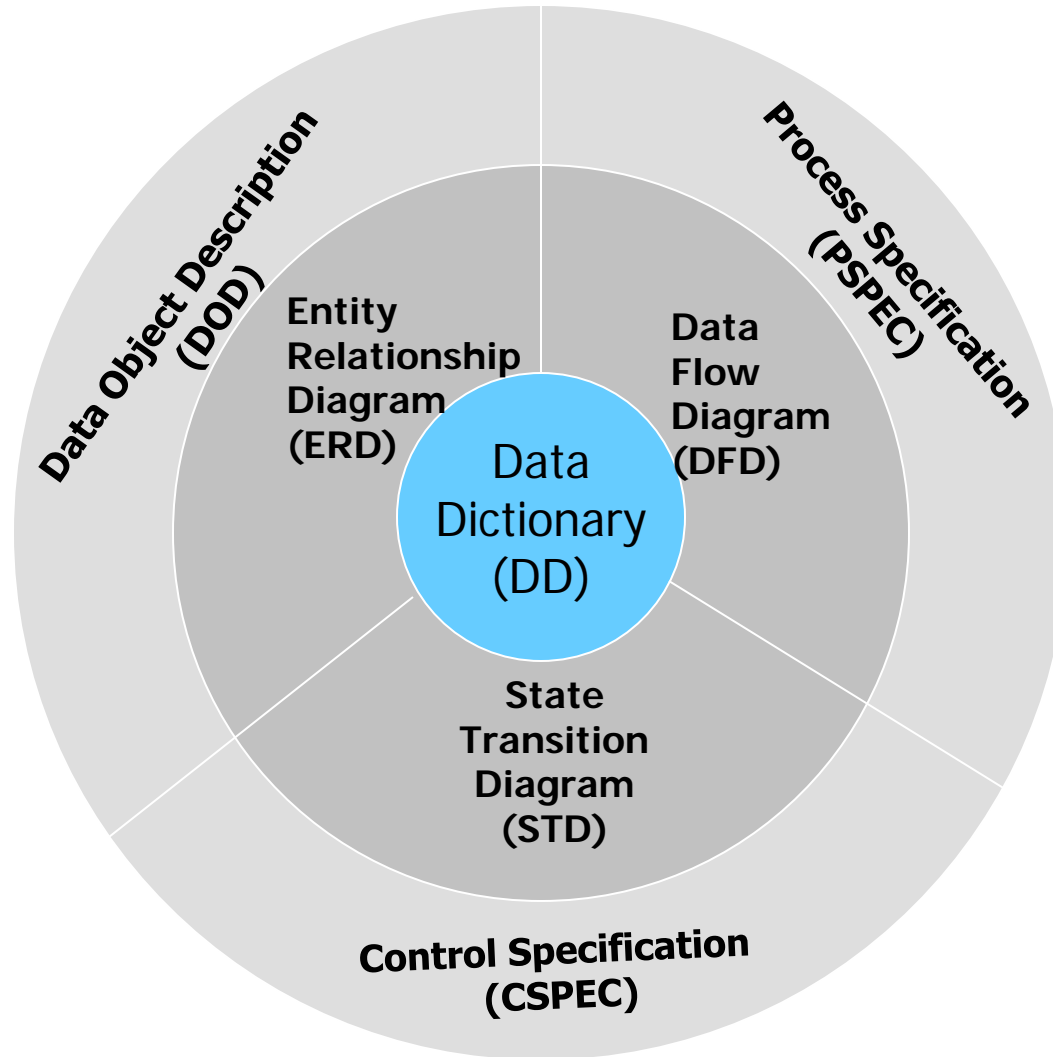
- Apakah requirements complete?
 - Apakah requirements concise?
 - Apakah requirements correct?
 - Apakah requirements consistent?
 - Apakah requirements modular? Dapatkah mengakomodasi perubahan?
 - Apakah requirements realistic?
 - Apakah requirement dibutuhkan oleh customer?
 - Apakah requirements traceable?
- 

ANALYSIS MODELING

- Pemodelan analisis merupakan representasi teknis yang pertama dari sistem
- Lanskap pemodelan analisis:
 - Analisis Terstruktur : adalah aktivitas pembangunan model dengan menggunakan notasi yang sesuai dengan prinsip analisis operasional.
 - Menciptakan model yang menggambarkan data dan aliran informasi
 - Membagi sistem secara fungsional dan secara behavioral
 - Menggambarkan esensi dari apa yang harus dibangun
 - Analisis Berorientasi Objek : menggambarkan informasi, fungsi, dan perilaku dalam konteks elemen model objek



ANALYSIS MODELING STRUCTURE



THE ELEMENT OF ANALYSIS MODEL

- Sasaran utama model analisis:
 - Untuk menggambarkan apa yang dibutuhkan oleh pelanggan
 - Untuk membangun dasar bagi pembuatan desain perangkat lunak
 - Untuk membatasi serangkaian persyaratan yang dapat divalidasi saat perangkat lunak dibangun
- Elemen model analisis:
 - DD - penyimpan yang berisi deskripsi dari semua objek data yang dikonsumsi oleh perangkat lunak
 - ERD – menggambarkan hubungan antara objek data
 - DFD – merepresentasikan bagaimana data ditransformasi dan fungsi-fungsi yang mentransformasi aliran data
 - STD – menunjukkan bagaimana sistem bertindak sebagai akibat dari kejadian eksternal
 - PSPEC – deskripsi setiap fungsi pada DFD
 - CSPEC – deskripsi aspek kontrol dari perangkat lunak



DATA MODELING

- Model data terdiri dari tiga informasi yang saling tergantung:
 - objek data,
 - atribut yang menggambarkan data tersebut, dan
 - hubungan objek data satu dengan yang lain.
- Objek data adalah representasi dari semua informasi gabungan yang harus dipahami oleh perangkat lunak
 - Objek data dapat berupa entitas eksternal (semua sumber data atau yang mengkonsumsi informasi), suatu benda (laporan atau tampilan), peristiwa (sambungan telepon) atau event (sebuah alarm), peran (tenaga penjualan), unit organisasi (bagian akuntansi), atau suatu struktur (file).
- Atribut adalah properti suatu objek data.
 - Atribut digunakan untuk:
 - Menamai sebuah contoh dari objek data,
 - Menggambarkan contoh,
 - Membuat referensi ke contoh yang lain pada tabel yang lain
- Hubungan adalah relasi antara objek data yang satu dengan yang lainnya.
 - Misal toko buku dan buku → hubungan : toko buku memesan buku



OBJEK DATA, ATRIBUT DAN HUBUNGAN

Objek:



Atribut:

Nama
Alamat
Umur
Lisensi Mengemudi
Nomor

Hubungan:



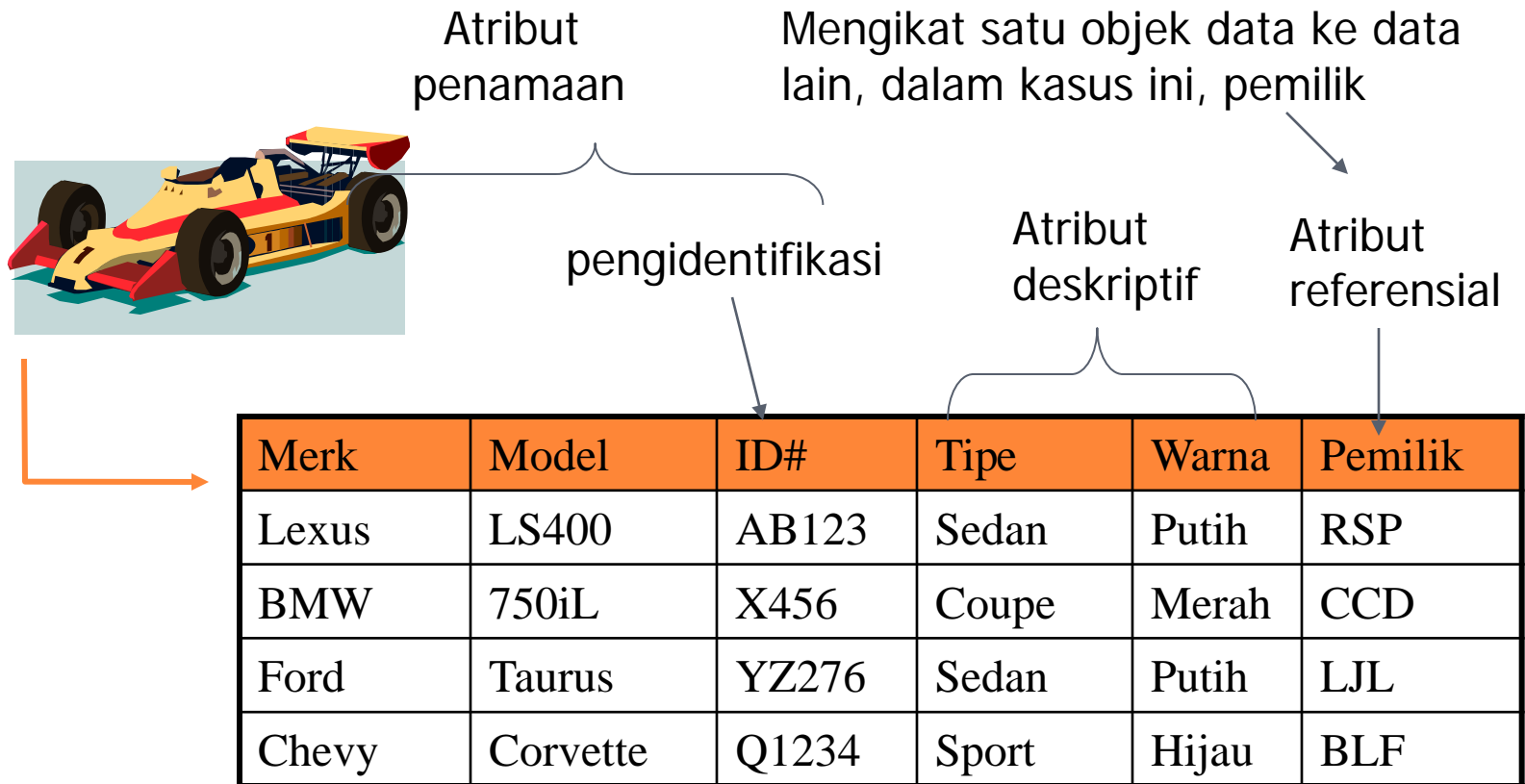
memiliki



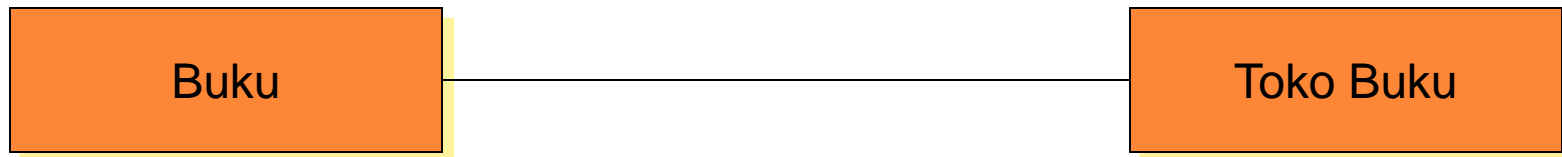
Merk
Model
Nomor ID
Tipe
Warna



REPRESENTASI TABULAR OBJEK DATA



HUBUNGAN OBJEK DATA



(a) Koneksi dasar antar objek



(b) Hubungan antar objek

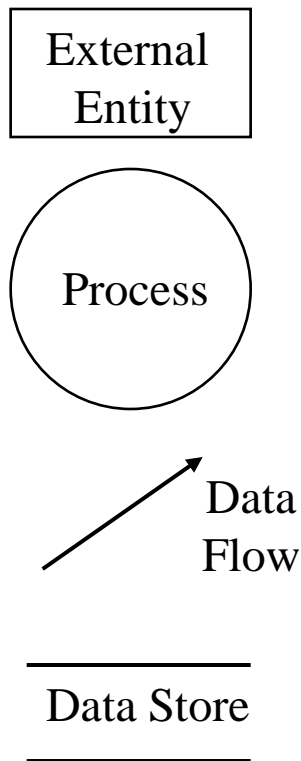


DATA FLOW DIAGRAM

- Data Flow Diagram (DFD) atau Diagram Arus Data (DAD) adalah gambaran grafis dari suatu sistem yang menggunakan sejumlah simbol untuk menggambarkan bagaimana data mengalir melalui suatu proses yang saling berkaitan.
- Walau nama diagram ini menekankan pada data, tetapi sebenarnya penekanannya ada pada proses.
- Diagram yang mendokumentasikan sistem pada tingkat yang paling ringkas disebut dengan Diagram Konteks (Context Diagram), sedangkan diagram yang lebih rinci disebut DAD Bertingkat (Levelled DFD).



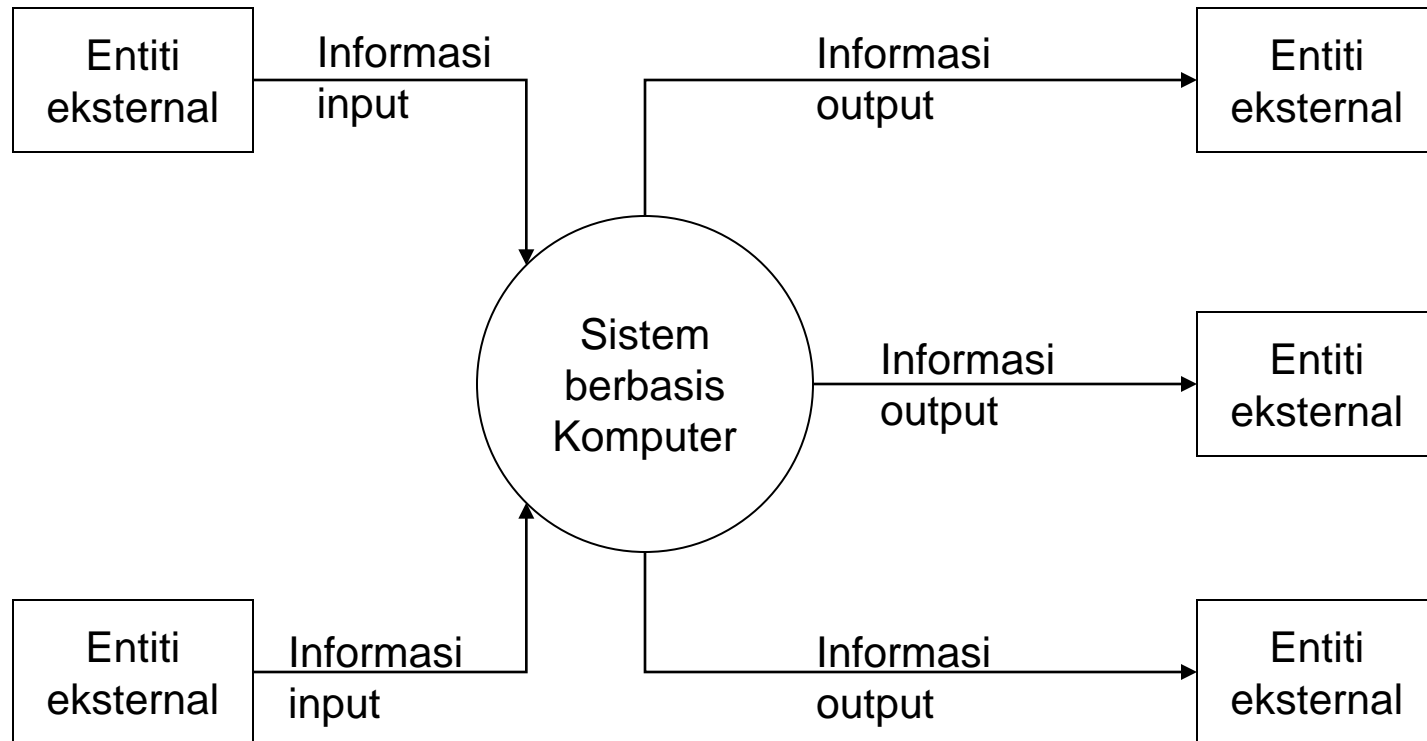
BASIC DFD NOTATION



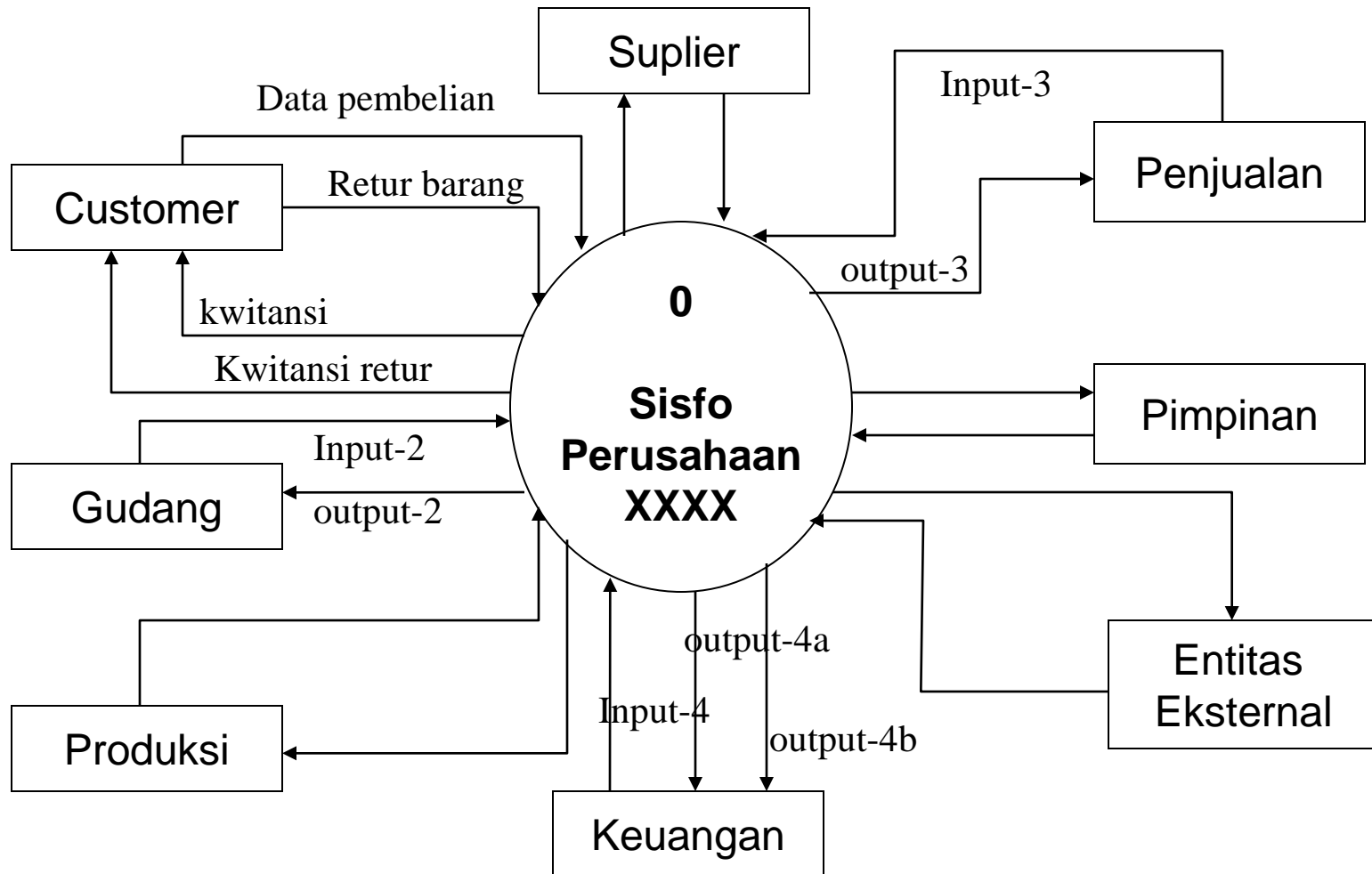
- **External Entity**
Elemen lingkungan yang berada di luar batas sistem, yang dapat memberikan masukan kepada sistem dan atau menerima keluaran dari sistem.
- **Data Process**
Data Process atau Proses adalah sesuatu yang mengubah input menjadi output.
- **Data Store**
Data store (penyimpanan data) adalah tempat penampungan suatu data
- **Data Flow**
Obyek data yang menunjukkan arah aliran data



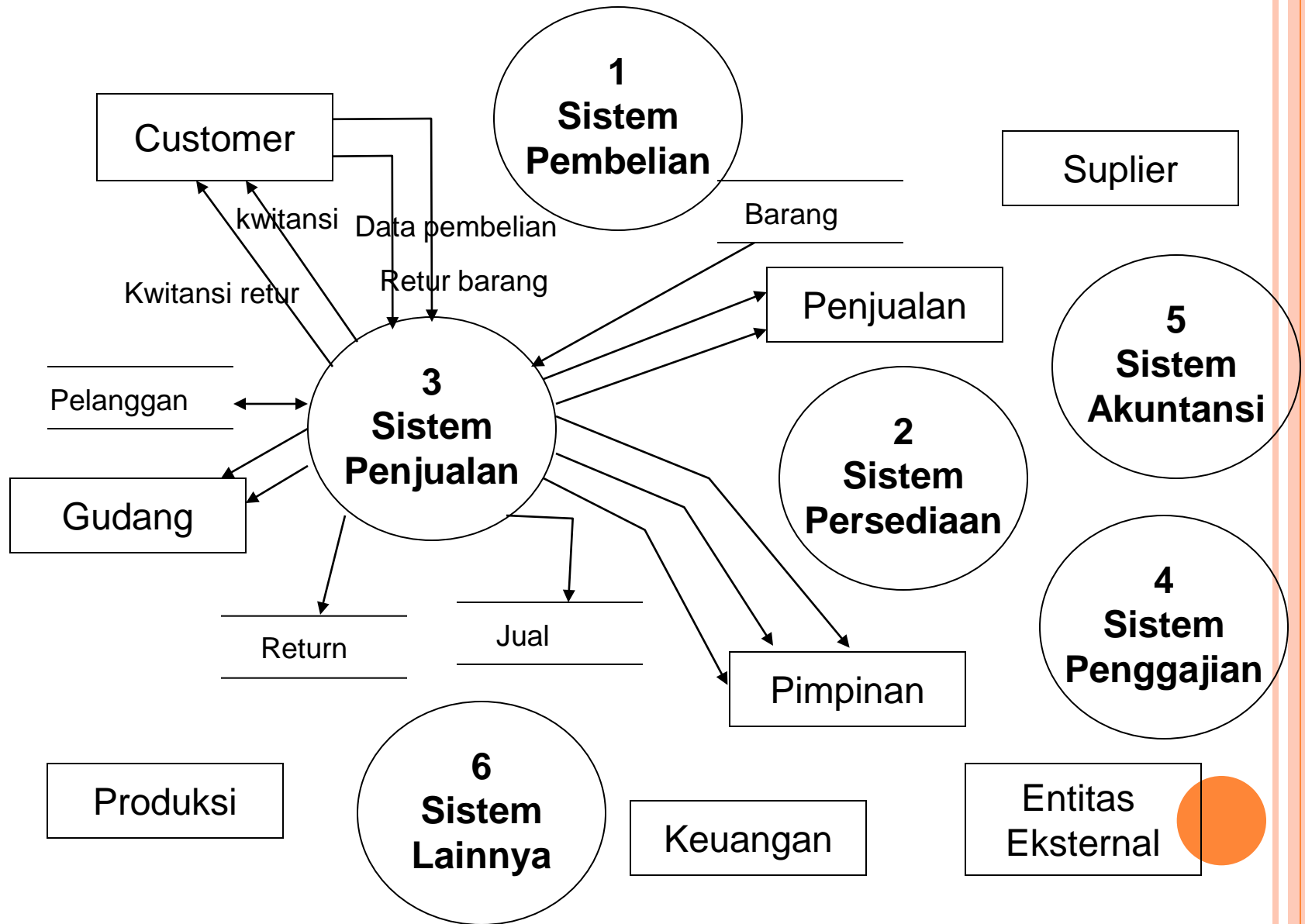
INFORMATION FLOW MODEL



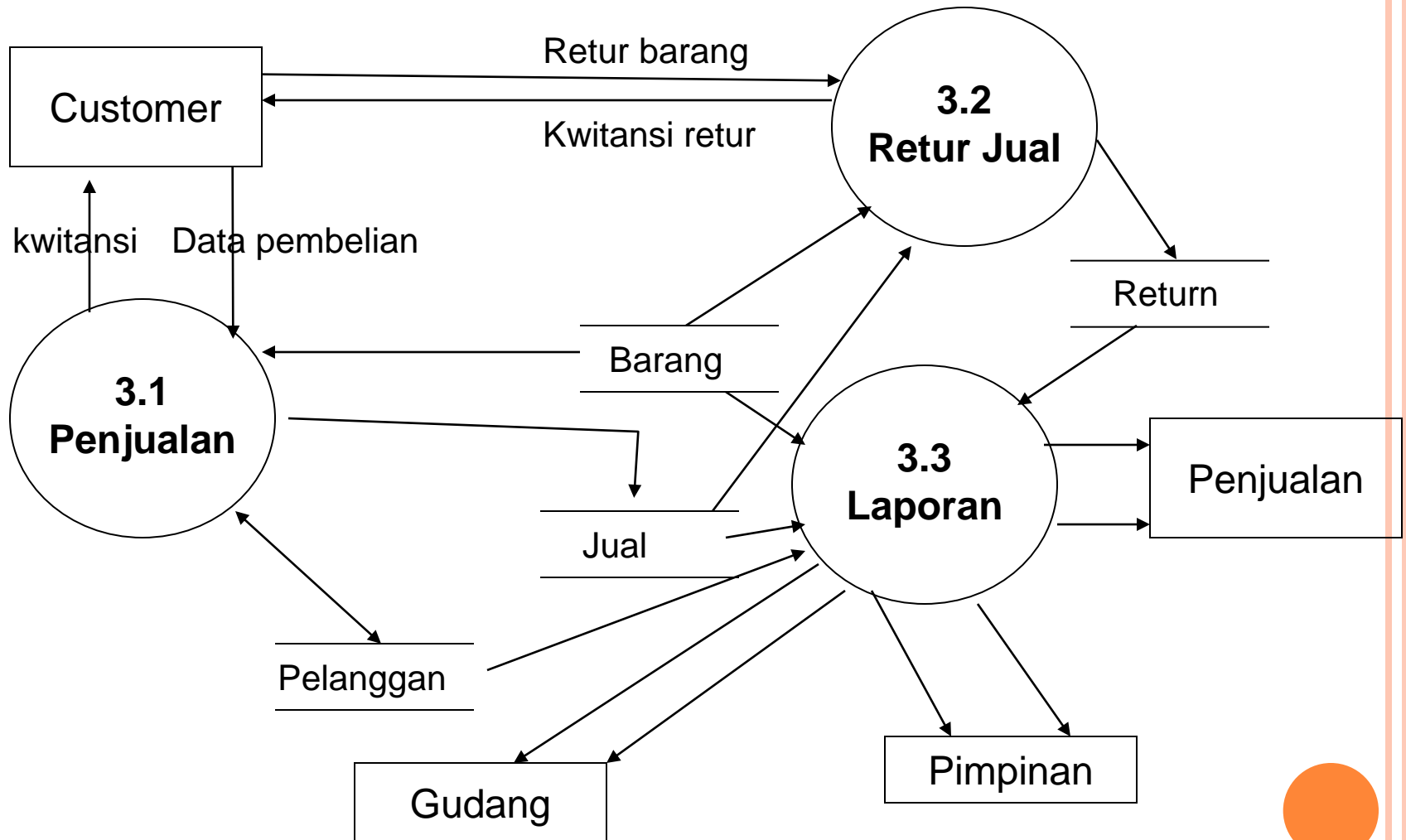
DFD EXAMPLE (CONTEXT DIAGRAM)



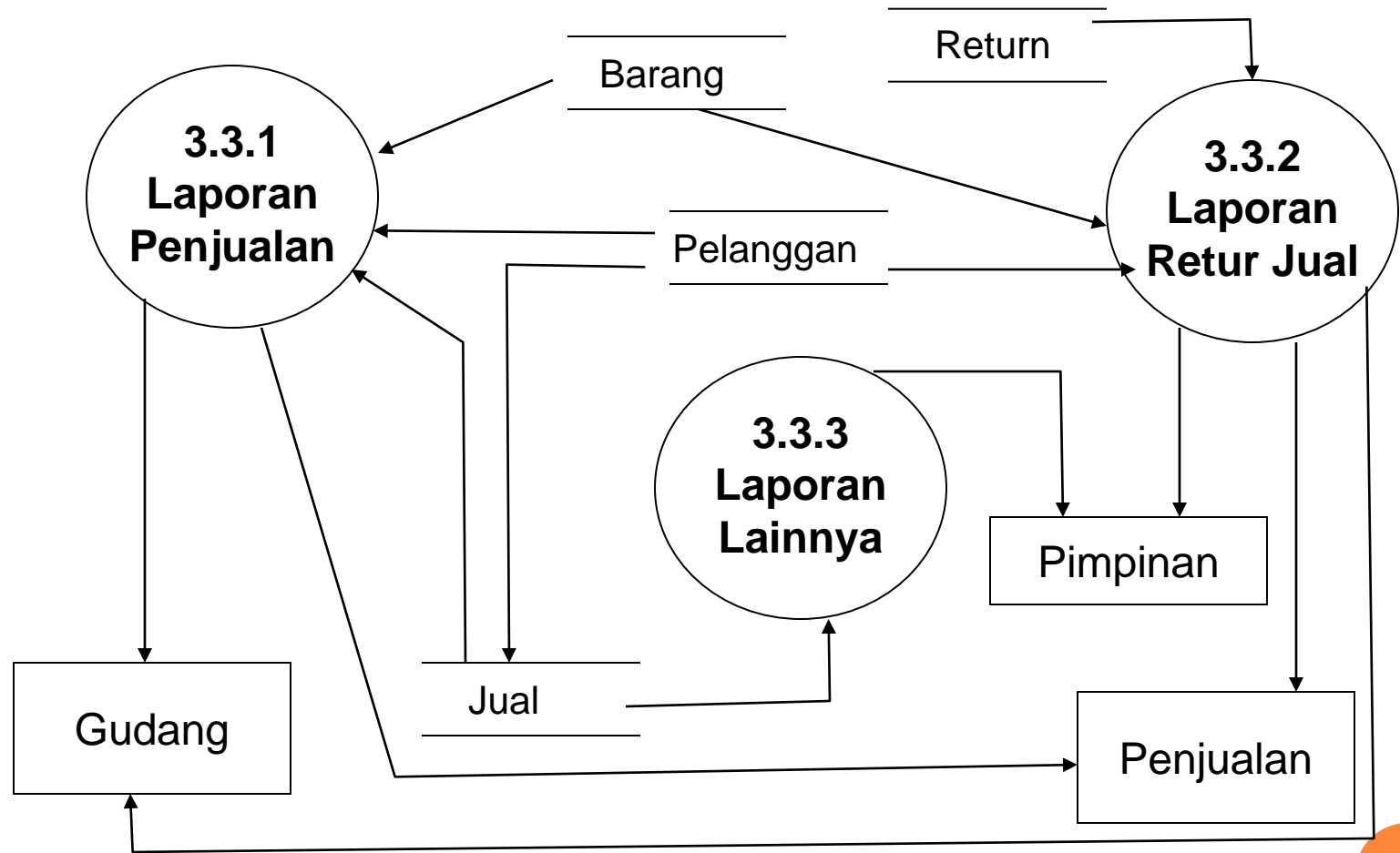
LEVEL 1 DFD



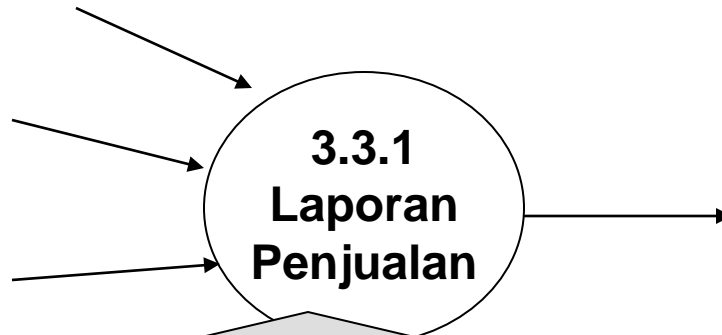
LEVEL 1 DFD FOR PROCESS 3



LEVEL 3 DFD FOR PROCESS 3.3

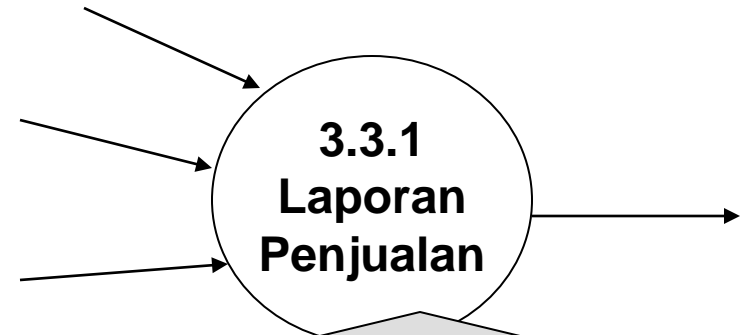


PROCESS SPECIFICATION



PSPEC (Spesifikasi Proses) :
Naratif Proses untuk laporan penjualan

Proses laporan penjualan menggunakan tabel Pelanggan, Barang dan Jual. Proses diawali dengan masukan periode penjualan, selanjutnya sistem akan mencetak ke layar judul laporan, periode dan header, dilanjutkan dengan mencetak semua penjualan yang terjadi pada periode tersebut. Data yang tercetak adalah : nomor urut (counter), nomor kwitansi (jual), kode pelanggan (jual), nama pelanggan (pelanggan), tanggal penjualan (jual), dll... Setiap halaman tercetak 20 data penjualan,dst



PSPEC (Spesifikasi Proses) :
Structure English (Pseudo Code)/PDL

```
Procedure Cetak Laporan:  
  Open file pelanggan, barang, jual  
  Relation ketiga file tersebut  
  Input periode penjualan  
  Nomor=0;halaman=0  
  Do while not eof  
    If halaman=0 then  
      Increment halaman  
      Print judul dan header  
    endif  
  Cari data pada tabel jual untuk periode tst  
  If ketemu  
    Increment nomor  
    Cetak nomor, nomor kwitansi, dll  
  Endif ; dst..
```

LATIHAN DFD

Latihan 1

Untuk meningkatkan pelayanan terhadap pasiennya, suatu poliklinik akan membangun sistem pelayanan pasien yang dibantu oleh komputer. Anda diminta untuk membuat perangkat lunak tersebut. Dari survey yang Anda lakukan, Anda mendapatkan abstraksi proses sebagai berikut :

Ketika pasien mendaftar, maka akan dilakukan pengecekan terhadap data yang telah disimpan untuk mencari apakah pasien tersebut pernah terdaftar ataukah belum. Bila belum terdaftar, maka data lengkap pasien akan dimasukkan.

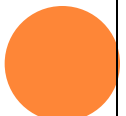
Sebagai salah satu bahan masukan bagi dokter dalam melakukan diagnosa adalah catatan kesehatan pasien.

Setelah selesai pemeriksaan, dokter memberikan data tentang diagnosa terhadap pasien tersebut, serta menuliskan resep. Dokter juga harus melaporkan tentang injeksi apa yang telah diberikan.

Dari resep yang dibuat dokter tersebut, maka dikeluarkan biaya tentang obat yang harus dibayar. Selanjutnya petugas poliklinik akan menyerahkan obat tersebut beserta rincian biaya tersebut (biaya obat dan biaya pemeriksaan). Data tentang diagnosa dan obat yang diberikan dicatat oleh petugas poliklinik sebagai catatan kesehatan pasien.

Kepala poliklinik berharap agar sistem yang akan dibuat dapat memberikan laporan keuangan dan laporan pemakaian obat.

Dari deskripsi tersebut, susunlah DFD hingga level paling rinci



LATIHAN DFD

Latihan 2

Diberikan Term of Reference (TOR) dari Sistem Reservasi Hotel sebagai berikut:

Bagian pengecekan ketersediaan kamar (reservasi hotel) berdasarkan jenis dan tanggal tertentu akan digunakan sebagai panduan di dalam reservasi dan pemesanan kamar. Pemesanan ini dapat dilakukan oleh customer sendiri maupun melalui agen perjalanan.

Ketentuan pemesanan kamar hotel adalah sebagai berikut:

- Dilakukan minimal 1 hari sebelum check-in dan minimal 1 kamar
- Pemesanan lebih dari 1 minggu sebelumnya ataupun lebih dari 1 kamar dan pada musim liburan, wajib membayar down payment (uang muka) secara langsung atau melalui transfer rekening bank.
- Pemesanan 10 kamar atau lebih mendapat diskon 5 %
- Pemesanan 2 minggu atau lebih mendapat diskon 10 %

LATIHAN DFD (LANJUTAN LATIHAN 2)

Pada saat customer check-out, mereka harus menyelesaikan administrasi pada front office untuk hal-hal sebagai berikut:

- Pembayaran sewa kamar selama tinggal di hotel
- Pembayaran fasilitas dan servis ekstra yang dikenai biaya

Hotel memiliki standar bintang 4 sebanyak 200 kamar terdiri dari:

- Junior Suite 85 kamar bertarif U\$ 45
- Bisnis 94 kamar bertarif U\$ 55
- Deluxe 11 kamar bertarif U\$85
- Executive 10 kamar bertarif U\$ 125

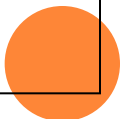
Servis-servis yang diberikan hotel adalah : Dokter, Laundry, Restaurant, Telephone, Money Changer

Sedangkan fasilitas-fasilitas yang diberikan hotel adalah: Kolam renang, Fitness Centre, Meeting Room, Pub & Karaoke.

Untuk informasi mengenai kamar hotel terbagi menjadi:

- Keterhubungan antar kamar dengan lainnya.
- View kamar.

Dari deskripsi tersebut, susunlah DFD hingga level paling rinci



ENTITY RELATIONSHIP DIAGRAM

- ERD adalah alat bantu diagramatik untuk mendiskripsikan :
 - Relasi/hubungan antar entitas
 - Atribut dari semua entitas yang berhubungan
- Komponen utama ERD adalah objek data, atribut, hubungan dan tipe indikator.
- Tujuan utama dari ERD adalah untuk mewakili objek data dan hubungan mereka.
- Objek data diwakili oleh sebuah persegi panjang yang diberi label
- Hubungan ditunjukkan dengan garis yang diberi label yang menghubungkan objek.
- Sambungan antara data dan objek dibangun dengan menggunakan berbagai simbol khusus yang menunjukkan kardinalitas dan modalitas.

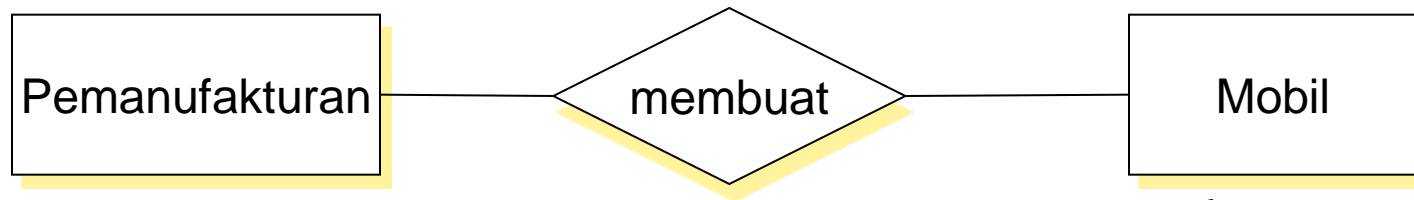


CARDINALITY

- Kardinalitas merupakan spesifikasi dari sejumlah peristiwa dari satu objek yang dapat dihubungkan ke sejumlah peristiwa dari objek lain.
 - Satu ke satu (1:1) – suatu peristiwa dari objek 'A' dapat berhubungan dengan satu dan hanya satu kejadian dari objek 'B', dan sebaliknya.
 - Satu ke banyak (1:N) – suatu kejadian dari objek 'A' dapat berhubungan dengan satu atau lebih kejadian dari objek 'B', tetapi sebuah kejadian 'B' dapat berhubungan dengan hanya satu kejadian 'A'.
 - Banyak ke banyak (N:N) - suatu kejadian dari objek 'A' dapat berhubungan dengan satu atau lebih kejadian dari objek 'B' dan sebaliknya.
- Kardinalitas mendefinisikan 'jumlah maksimum dari hubungan objek yang ikut serta di dalam sebuah hubungan', tetapi tidak memberikan sebuah indikasi apakah objek data tertentu berpartisipasi di dalam hubungan atau tidak.



ERD (EXAMPLE)

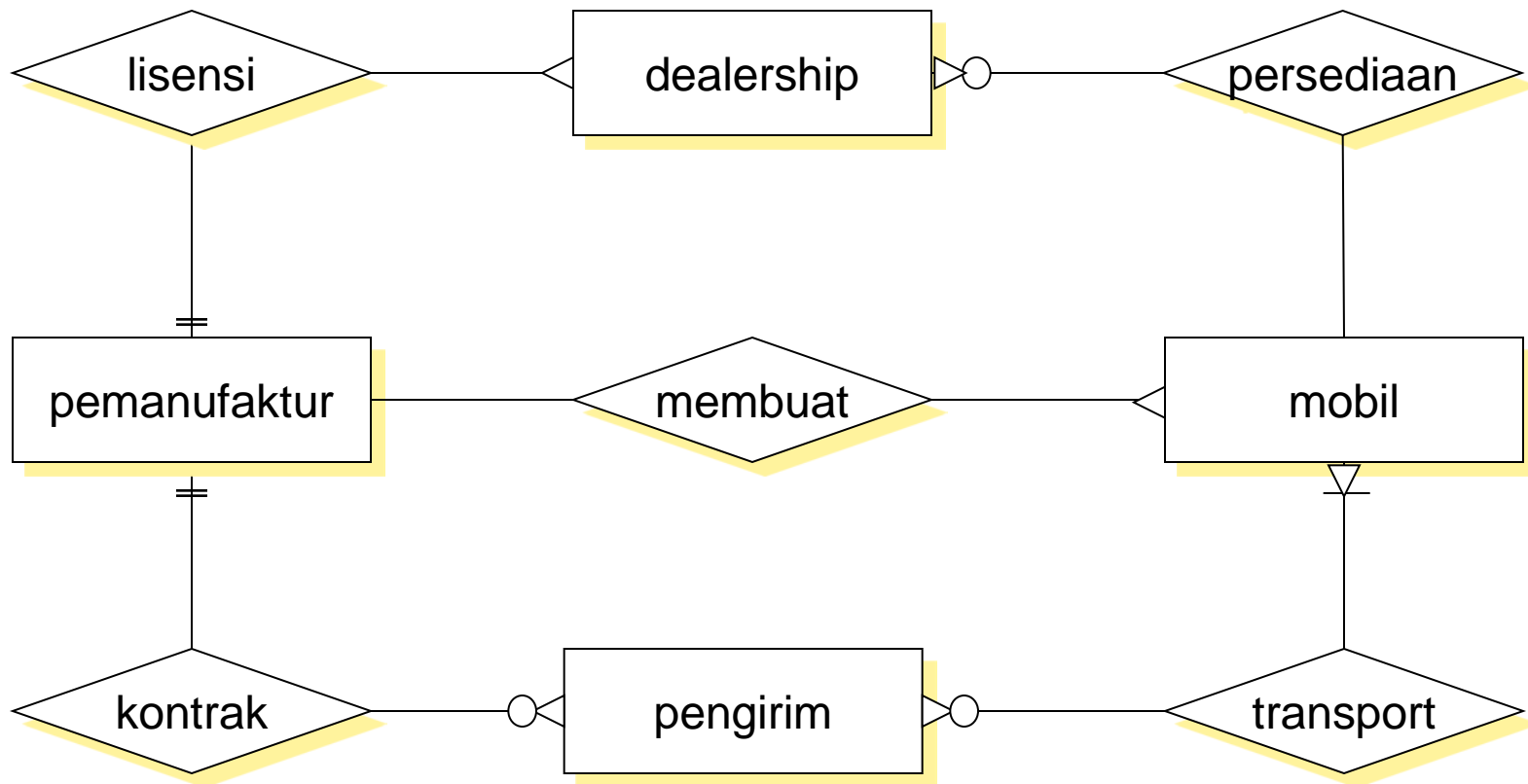


Tabel Objek Data

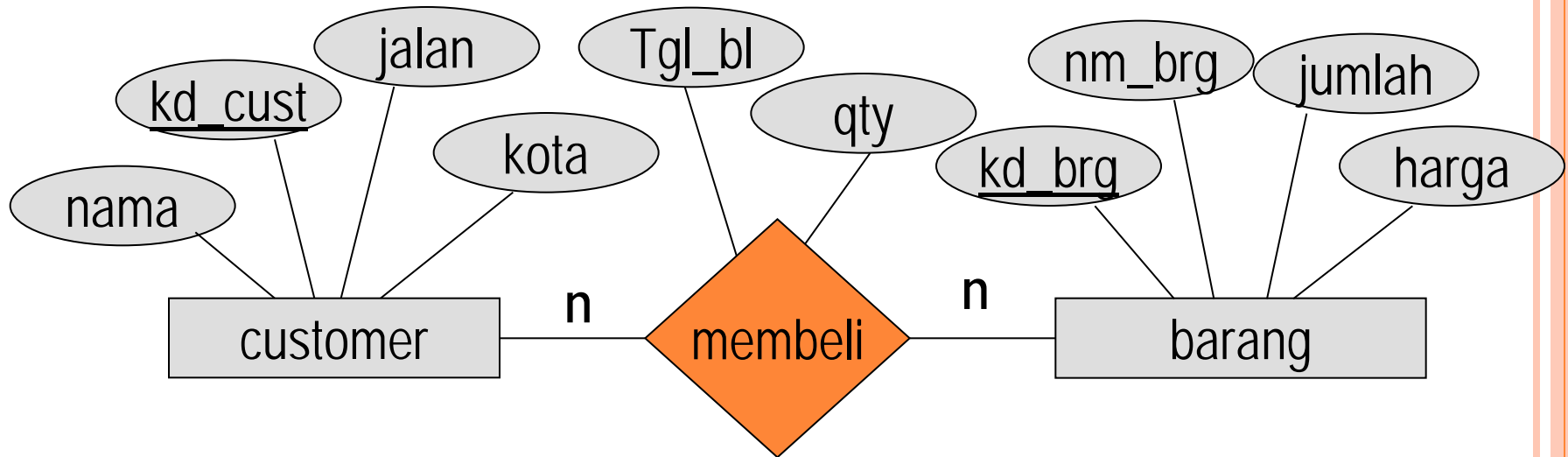
ID#	model	Tipe body	mesin	transmisi	...



ERD (EXAMPLE CONT)



ERD (EXAMPLE CONT.)



Skema basis data:

Customer = (kd_cust, nama, jalan, kota)

Barang = (kd_brg, nm_brg, harga, jumlah)

Beli = (kd_cust, kd_brg, tgl_bl, qty)



ERD CONT.

- Dari ERD yang dihasilkan, disusun skema basis data (database scheme)
- Dilanjutkan uji normalisasi untuk mendapatkan basis data yang baik
 - Jumlah tabel hasil normalisasi dapat lebih banyak atau lebih sedikit dari tabel yang sebelum dinormalisasi
- Dari tabel normal, dibuat struktur fisik tabel
- Untuk keperluan validasi domain dari atribut di setiap tabel, dibuat kamus data



LATIHAN ERD

Buat ERD untuk latihan 1 dan 2 pada kasus latihan DFD sebelumnya

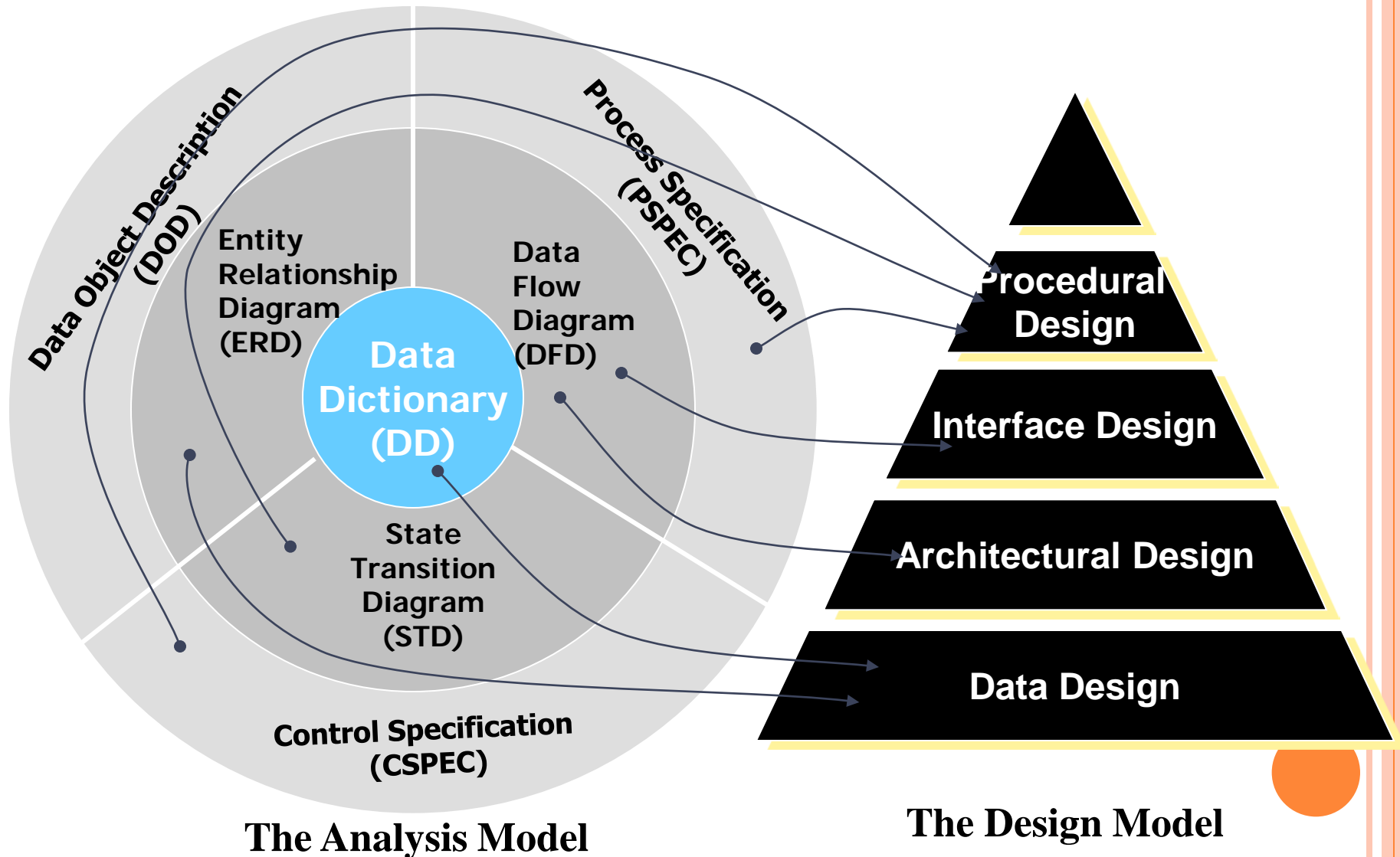


DESIGN CONCEPTS AND PRINCIPLES

- Merupakan langkah pertama dalam fase pengembangan produk bagi setiap produk (sistem) yang direkayasa.
- Desain adalah proses aplikasi berbagai teknik dan prinsip bagi tujuan pendefinisian suatu perangkat lunak (proses/sistem) dalam detail untuk digunakan sebagai pedoman realisasi fisiknya (coding)



TRANSLATING THE ANALYSIS MODEL INTO A SOFTWARE DESIGN



DATA DESIGN

Desain data adalah tahapan pemilihan representasi logis dari objek data yang telah teridentifikasi dalam Analisis Persyaratan dan Spesifikasi. Prinsip²nya :

- Prinsip² analisis sistem yang diterapkan pada fungsi dan perilaku PL juga perlu diterapkan pada data.
- Semua struktur data dan operasinya harus teridentifikasi.
- Kamus data harus dibangun untuk merepresentasikan hub antarobjek data dan batasan²nya.



PRINCIPLES

- Keputusan desain data tingkat rendah harus dilakukan diakhir proses desain data.
- Konsep information hiding (penyembunyian informasi suatu modul agar tidak diakses oleh modul lain yang tidak berkepentingan) dan perangkataian sangat penting bagi kualitas PL.
- Pustaka struktur data dan operasi yang berguna harus dikembangkan agar dapat digunakan kembali.
- Desain PL dan bahasa pemrograman harus mendukung spesifikasi dan realisasi tipe data abstrak.



ARCHITECTURAL DESIGN

Arsitektur merupakan struktur hirarkhi dari komponen program (modul), interaksinya, dan struktur data yang digunakan. Terdapat bbrp model desain arsitektur :

- Model Struktural : bdsk struktur komponen program
- Model Kerangka Kerja : berupa peningkatan abstraksi desain berdasarkan kerangka kerja
- Model Dinamis : menggambarkan perilaku berdasarkan external events
- Model Proses : fokus pada proses
- Model Fungsional : menggambarkan hirarkhi fungsional



MODELING TOOLS

- ✉ *Alat process modeling : Flow Chart, Flow of Document, Data Flow Diagram, Structured Chart, HIPO Diagram, Pseudocode, Nassi-Shneiderman Chart, Warnier-Orr Diagram, Michael Jackson Diagram, Functional Decomposition, Action Diagram, Data Navigation Diagram, HOS Chart, dsb.*
- ✉ *Alat data modeling : Entity Relationship Diagram, Network Diagram, Hierarchical Diagram, Table Normalization, atau gabungannya.*



MODELING TOOLS (CONT.)

✉ *Alat object modeling : Object Diagram* (Coad/Yourdon, Rumbaugh, Firesmith, Booch, dsb.) yang dapat dibangun dengan Oracle Designer/2000, Microsoft Visual Studio – Enterprise Tools (Modeler), dsb.

✉ *Alat working modeling : Excelerator, Easycase, Oracle Designer/2000, Microsoft Visual Studio - Enterprise Tools (Modeler), dsb.*

*Studi perbandingan tentang berbagai macam alat tersebut dapat dijumpai di buku *Structured Techniques* (James Martin & Carma McClure).*



MODULARITY & PROBLEM COMPLEXITY

Modularitas diterapkan melalui pembagian PL ke dalam komponen – komponen PL yang dapat dipanggil terpisah sehingga bila terdapat problem, maka problem tsb akan lebih mudah diselesaikan.

Kompleksitas (C) problem (p1 dan p2) yang bergabung menjadi satu, lebih besar dibandingkan bila problem tersebut dipandang secara terpisah.

$$C(p1+p2) > C(p1) + C(p2)$$

Oleh sebab itu modularitas penting dalam desain arsitektur PL. Namun berkaitan dengan biaya, sebaiknya dihindari kondisi undermodularity maupun overmodularity. Semakin banyak modul semakin rendah biaya per-modul tetapi semakin tinggi biaya integrasinya.



FUNCTIONAL INDEPENDENCE

Konsep ini mrpk pertumbuhan langsung dari modularitas, konsep abstraksi PL, dan Information Hiding. Independensi diukur melalui Kohesi dan Kopling.

- Kohesi

Modul yang kohesif seharusnya hanya melakukan satu hal saja (kohesi tinggi = fungsional <> koinsidental).

- Kopling

Sehubungan dengan perangkaian dengan modul lain, maka modul yang baik seharusnya memiliki hubungan yang sederhana (kopling rendah)



INTERFACE DESIGN

1. **Internal** : merupakan desain **interface antarmodul** dalam PL yang dikendalikan oleh data yang harus mengalir di antara modul-modul. Aliran transformasi dalam DFD merupakan pijakan utama dalam desain ini selain kemampuan bahasa pemrograman.
2. **Eksternal** : merupakan interface untuk **entitas eksternal** (tidak termasuk manusia), misalnya sensor pada PL Safehome.
3. **Manusia – Mesin** : merupakan interface antara manusia dengan PL (Human – Computer Interface). Interface ini memiliki tantangan besar karena berkaitan dengan pengguna dengan **berbagai karakter** yang lebih sulit untuk dipelajari.



INTERFACE DESIGN GUIDELINES

a. Interaksi Umum

- **Format konsisten**
- **Berikan umpan balik**
- **Konfirmasi untuk aksi destruktif (misal Hapus)**
- **Ijinkan pembatalan (misal Undo)**
- **Kurangi jumlah informasi yang harus diingat**
- **Efisiensi dalam dialog, gerakan (tangan), pemikiran**
- **Perlindungan thd kegagalan**
- **Kategorikan aktivitas sejenis dan posisinya di layar**
- **Sediakan Help yang sensitif konteks**
- **Berikan petunjuk singkat (tools tips) pada setiap button / ikon / nama**
- **Gunakan perintah dan nama² yang pendek**



INTERFACE DESIGN GUIDELINES (CONT.)

b. Output

- Tampilkan informasi yang relevan dg konteks
- Jangan membanjiri pemakai dg informasi
- Gunakan label, singkatan, warna yg standar dan konsisten
- Peliharalah konteks visual saat pengguna melakukan zoom-in / zoom-out
- Pesan kesalahan harus memiliki arti yang jelas
- Gunakan variasi huruf, indentasi, pengelompokan untuk memudahkan pemahaman
- Gunakan jendela untuk tipe-tipe informasi yang berbeda
- Gunakan tampilan alami (bukan angka/grafik) bila memungkinkan
- Geografi layar dioptimalkan shg tidak ada jendela yang 'hilang' / sulit ditemukan
- Berikan kemungkinan kustomisasi output (utk advance user)
- Jangan ada informasi/data yang tidak lengkap/hilang sebagian



INTERFACE DESIGN GUIDELINES (CONT.)

c. Input

- Minimalkan jumlah aksi input (combo box, list, dsb.)
- Konsisten
- Berikan kemungkinan kustomisasi input (utk advance user)
- Mode input harus fleksibel (mouse / keyboard)
- Non-aktifkan button / ikon yang tidak relevan dengan aksi
- Berikan kesempatan untuk mengontrol aliran interaksi (mengubah, membetulkan, mengulang)
- Sediakan Help
- Jangan meminta aktivitas manual (perhitungan, tanggal, waktu, dsb) bila dapat dilakukan oleh PL



PROCEDURAL DESIGN

Spesifikasi prosedural ditetapkan setelah desain data, arsitektur, dan interface selesai guna penyusunan algoritma PL.

Desain prosedural diterapkan melalui teknik pemrograman terstruktur yang didasarkan pada struktur logika algoritma :

- Sekuensial
- Percabangan
- Perulangan

Alat-alat yang dapat digunakan :

Flow Chart, HIPO Diagram, Pseudocode, Nassi-Shneiderman Chart, Warnier-Orr Diagram, Action Diagram, HOS Chart, dsb.

