

Mata Kuliah

ANALISIS & DESAIN SISTEM (ADS)

System Analysis & Design

Dosen:

Budi Nugroho, S.Kom, M.Kom

Materi 4

Structured Modeling: Detailed Design



**Program Studi S1 Informatika
Fakultas Ilmu Komputer UPN “Veteran” Jawa Timur**

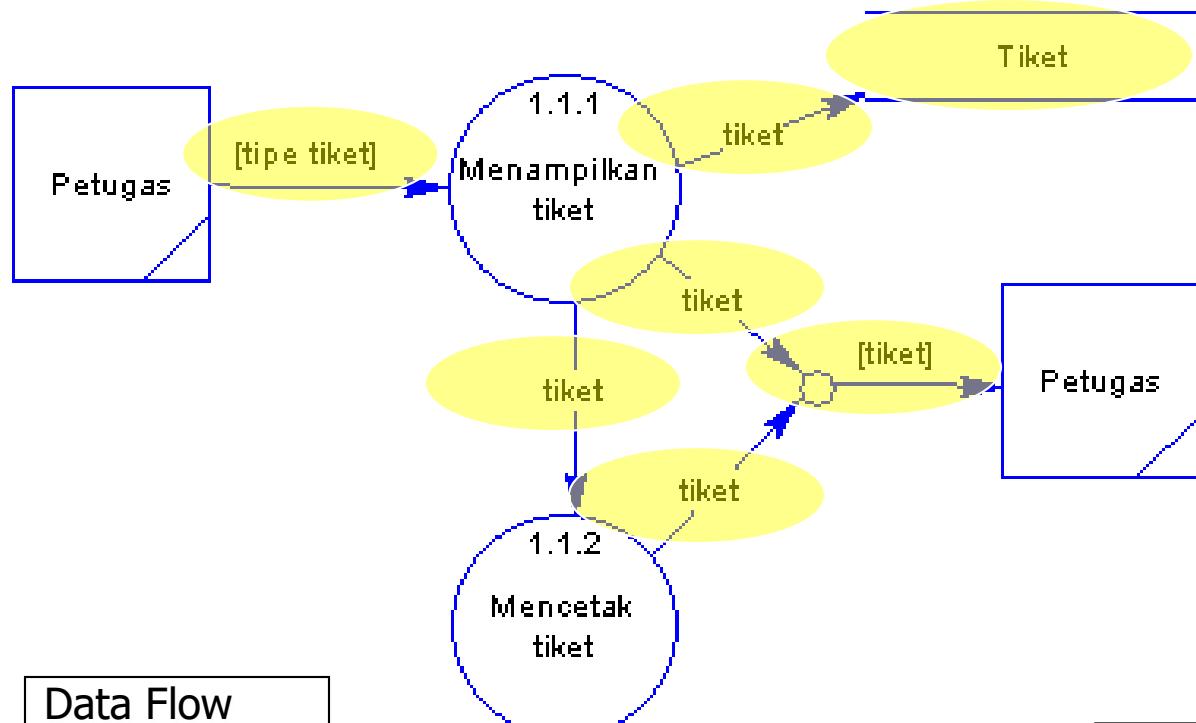
TA 2019/2020 Semester Genap

Detailed Design

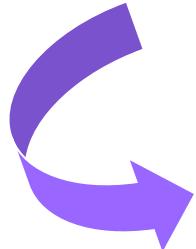
- Data Dictionary
- Process Specification
- User Interface
- Data Modeling

Data Dictionary (Kamus Data)

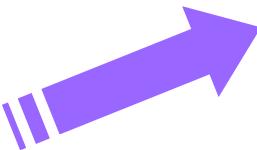
DFD Level 2 : Proses 1.1 Membuat Tiket



Data Flow
Data Store



Bagaimana
Komposisi
elemen datanya ?



Data Dictionary
(Data Library)

- Menjelaskan detail dari elemen data pada DFD (Data Flow & Data Store) agar semua stakeholder memiliki persepsi yang sama.

Data Dictionary untuk Data Flow

Nama : Nama dari Data Flow di DFD

Deskripsi : Penjelasan rinci (apa maksudnya)

Aliran : Sumber dan tujuan (darimana dan kemana)

Periode : Kapan terjadinya (waktu atau kejadian lain yang memicu)

Struktur : Komposisi data

Data Dictionary untuk Data Flow

Struktur

Notasi	Keterangan
=	Terdiri atas ; diuraikan menjadi
+	Dan (komposisi dari beberapa data)
&	Dan (gabungan dari beberapa data)
	Memilih salah satu dari beberapa data
[...]	Data atau Sekelompok data yang harus ada
(...)	Optional (boleh ada atau boleh tidak ada)

Contoh :

Pelanggan = nama_pel + alamat_pel

nama_pel = karakter huruf = [A-Z | a-z]

alamat_pel = nama_jalan + nomor_rumah + nama_kota + (kode_pos)

nama_jalan = karakter huruf = [A-Z | a-z]

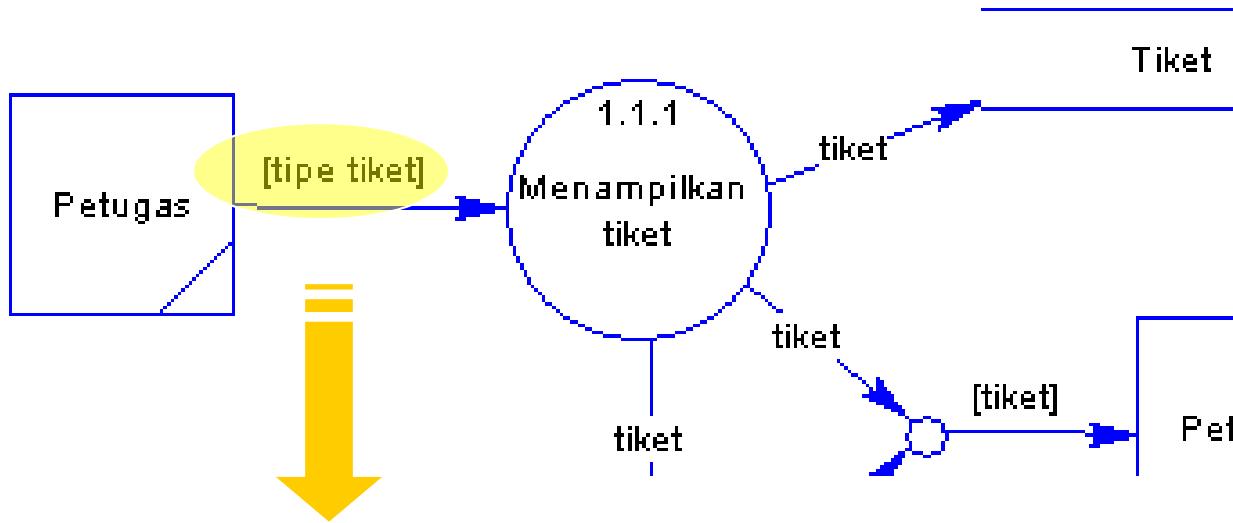
nomor_rumah = karakter angka + (karakter huruf)
= [0-9] + (A-Z)

nama_kota = karakter huruf = [A-Z | a-z]

kode_pos = 5 digit karakter angka

Contoh

DFD Level 2 : Proses 1.1 Membuat Tiket



Data Dictionary :

- Nama : tipe tiket
- Desripsi : tipe dari tiket yang ingin dibeli pelanggan
- Aliran : dari Petugas ke Proses 1.1.1 (Menampilkan tiket)
- Periode : saat Petugas ingin membuat tiket
- Struktur : tipe tiket = karakter_huruf
= [A-Z]

Data Dictionary untuk Data Store

Nama : Nama dari Data Store di DFD

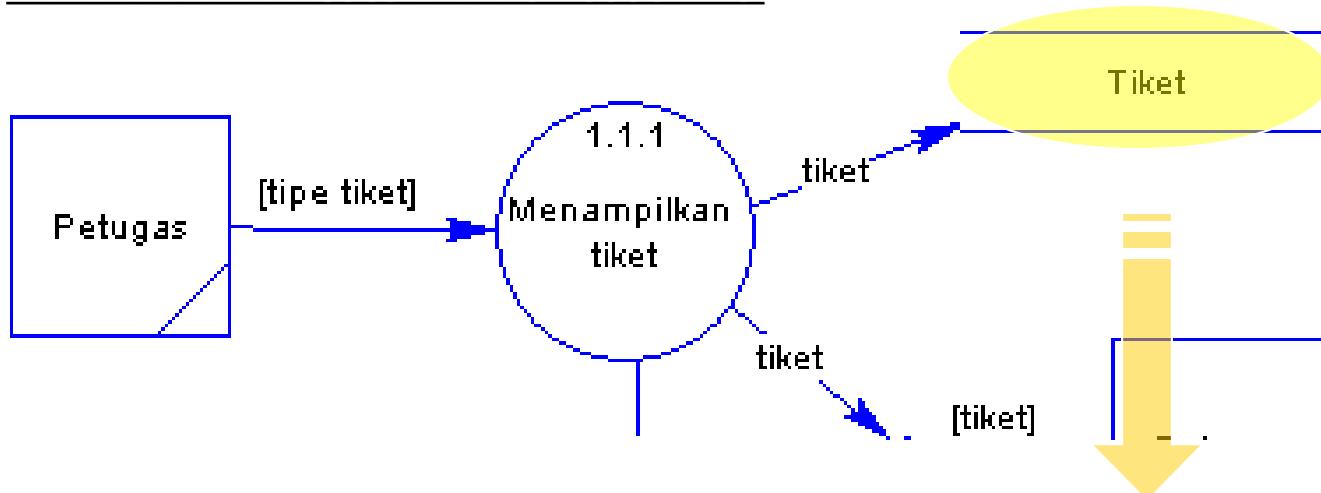
Deskripsi : Penjelasan rinci (apa maksudnya)

Struktur : Komposisi Tabel pada DBMS

Nama	Tipe Data	Keterangan

Contoh

DFD Level 2 : Proses 1.1 Membuat Tiket



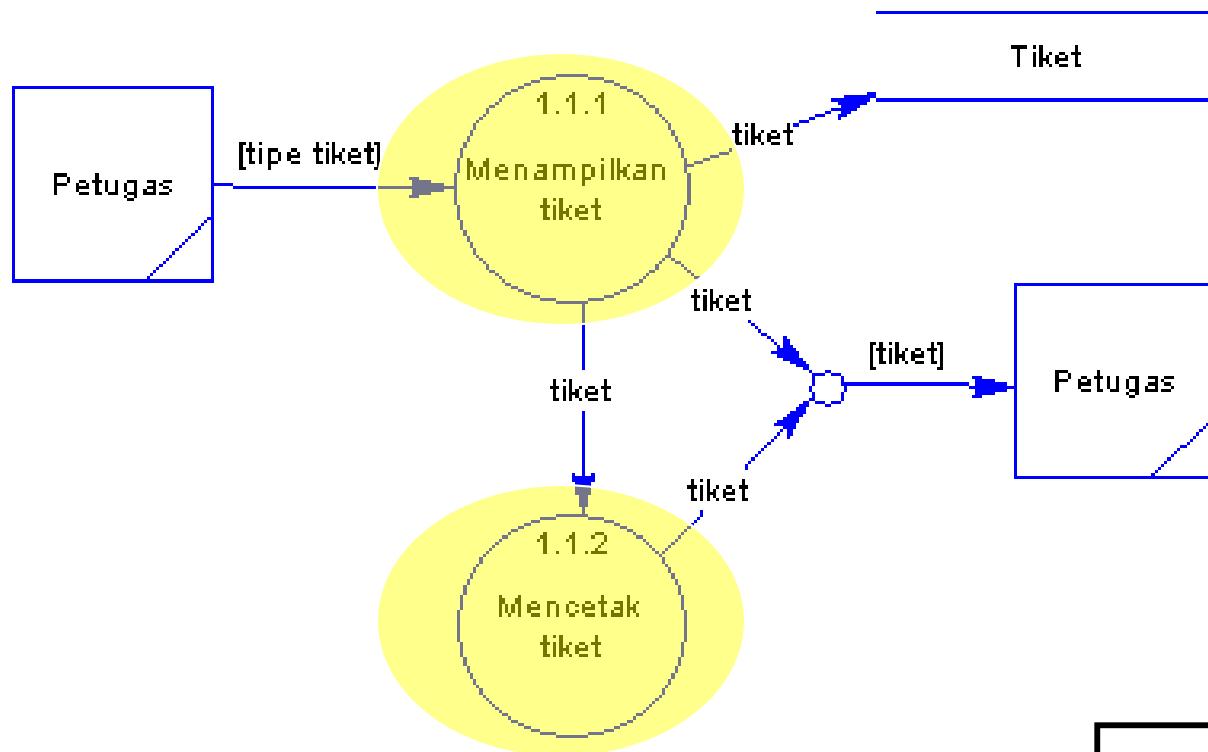
Data Dictionary :

- Nama : Tiket
- Deskripsi : penyimpanan detail tiket
- Struktur :

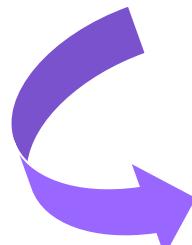
Nama	Tipe Data	Keterangan
IdTiket	Text	Primary Key (No Duplicates)
Tipe	Text	-
Status	Boolean	Yes/No ; Yes = sudah terbeli, No = belum terbeli
Waktu	Date/Time	Format : MM/DD/YYYY HH:MM:SS

Process Specification (Spesifikasi Proses)

DFD Level 2 : Proses 1.1 Membuat Tiket

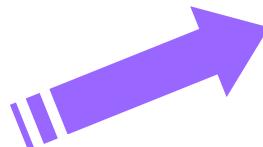


Proses di level terendah



Bagaimana
detil
prosedurnya ?

Process
Specification



- Menjelaskan detail prosedur elemen proses pada DFD level terendah agar semua stakeholder memiliki persepsi yang sama.

Process Specification

Nomor : Nomor Proses

Nama : Nama Proses

Deskripsi : Penjelasan rinci (apa maksudnya)

Input : Data Flow Input dan Asalnya

Output : Data Flow Output dan Tujuannya

Struktur : Algoritma / prosedur detail dari Proses

Spesifikasi Proses : Struktur Program

- Tujuan :

Memudahkan Programmer untuk membuat kode-kode program sesuai dengan yang dimaksud oleh System Analyst.
- Bentuk :
 1. Common Style

Bentuk umum / bebas yang mudah dipahami (bisa menggunakan bahasa apapun).
 2. Pseudocode

Bentuk yang mirip dengan kode program, menggunakan bahasa inggris yang terstruktur (Structured English).
 3. Program Flowchart

Bentuk diagram yang menggambarkan alur program (bagan alur).

Struktur 1 : Common Style

- Contoh 1 :

Mencatat Nama Mahasiswa

Mencatat NRP Mahasiswa

Untuk setiap mata kuliah yang diambil

Tampilkan nama dan sks mata kuliah

Hitung total sks sampai saat ini

Tampilkan Total Sks

Jika total sks lebih besar dari 18

Maka Tampilkan peringatan

Selainnya

Tampilkan persetujuan

Struktur 1 : Common Style

- Contoh 2 :

MENCATAT Nama Mahasiswa

MENCATAT NRP Mahasiswa

UNTUK SETIAP mata kuliah yang diambil

TAMPILKAN nama dan sks mata kuliah

HITUNG total sks sampai saat ini

TAMPILKAN Total Sks

JIKA total sks lebih besar dari 18

MAKA TAMPILKAN peringatan

SELAINNYA

TAMPILKAN persetujuan

Struktur 2 : Pseudocode

- Contoh 1 :

Read Nama Mahasiswa

Read NRP Mahasiswa

For each mata kuliah yang diambil

 Print nama dan sks mata kuliah

 total_sks = total_sks + sks

Print total sks

If total sks > 18

 Then Print Peringatan

Else

 Print Persetujuan

Struktur 2 : Pseudocode

- Contoh 2 :

Read (Nama Mahasiswa)

Read (NRP Mahasiswa)

For each (mata kuliah yang diambil)

 Print (nama dan sks mata kuliah)

 total_sks = total_sks + sks

Print (total sks)

If (total sks > 18)

 Then Print (Peringatan)

Else

 Print (Persetujuan)

Alur Program : Sequence

langkah 1
langkah 2
langkah n

Contoh :

Read (Nama Mahasiswa)
Read (NRP Mahasiswa)
Print (Nama & NRP Mahasiswa)

Alur Program : Condition

If (kondisi 1) Then

 urutan langkah jika kondisi 1 benar

If Else (kondisi 2) Then

 urutan langkah jika kondisi 2 benar

Else

 urutan langkah jika kondisi 1 & 2 salah

End If

Do Case

Case (kondisi 1)

 urutan langkah jika kondisi 1 benar

Case (kondisi 2)

 urutan langkah jika kondisi 2 benar

.....

Otherwise

 urutan langkah jika semua kondisi salah

End Case

Contoh :

If (nilai>55) Then

 Lulus = true

Else

 Lulus = false

End If

Do Case

Case (NA<40)

 NilaiHuruf = E

Case (NA>=40 & NA<55)

 NilaiHuruf = D

Case (NA>=55 & NA<70)

 NilaiHuruf = C

Case (NA>=70 & NA<80)

 NilaiHuruf = B

Otherwise

 NilaiHuruf = A

End Case

Alur Program : Repetition

For (inisialisasi ; kondisi ; increment)
Urutan langkah

Inisialisasi
Do While (kondisi)
Urutan langkah
increment
End Do

inisialisasi
Repeat
Urutan langkah
increment
Until (kondisi)

Contoh :

For ($i=1$; $i < 5$; $i++$)
Nilai = Nilai + 5

Contoh :

$i=1$
Do While ($i < 5$)
Nilai = Nilai + 5
 $i=i+1$
End Do

Contoh :

$i=1$
Repeat
Nilai = Nilai + 5
 $i=i+1$
Until ($i < 5$)

Contoh Terminologi untuk Komputasi

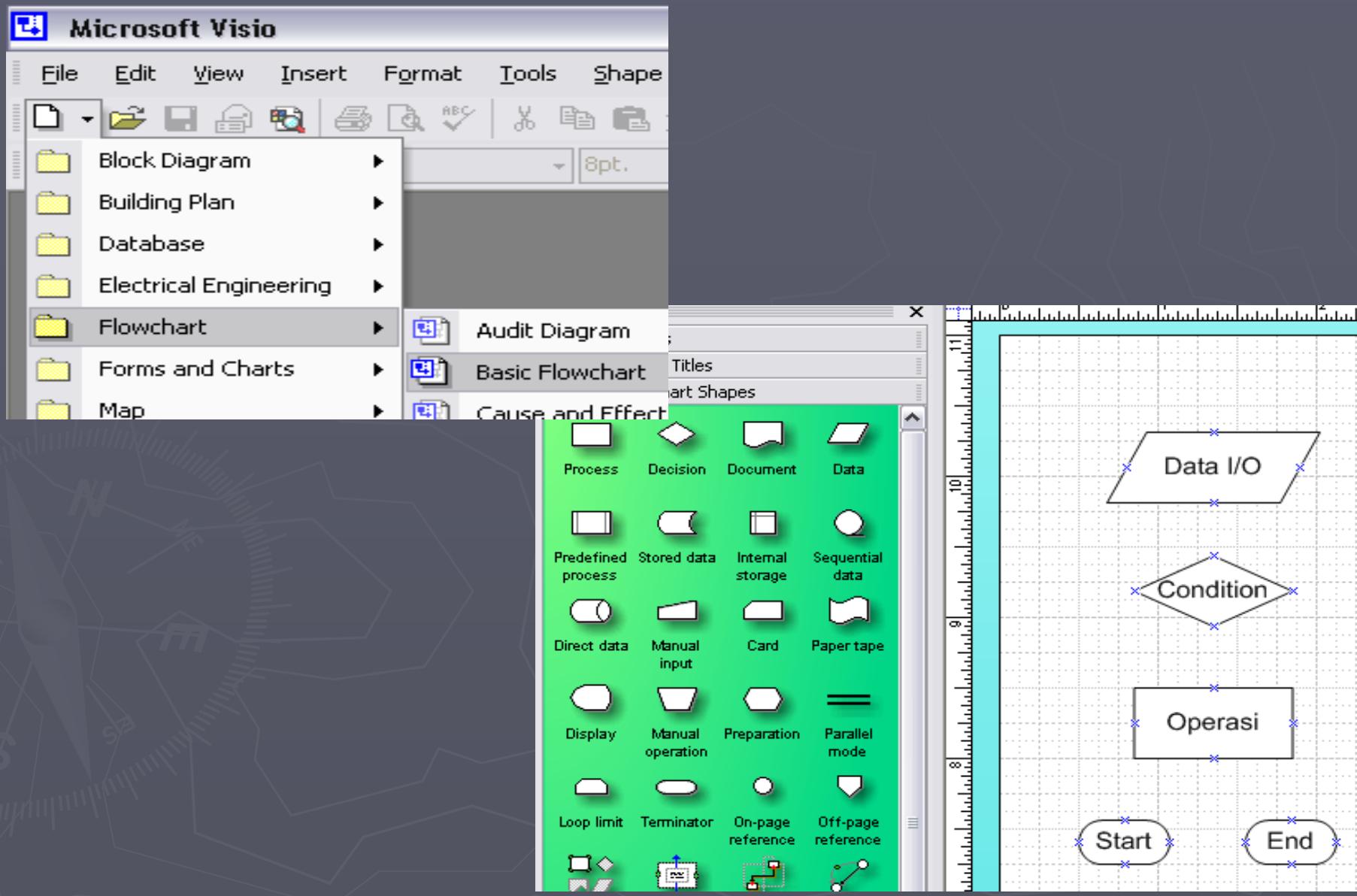
Indonesian Version

Cari
Jumlahkan
Kalikan
Kurangi
Bagi
Ambil
Tampilkan, Tulis
Hitung
Hapus
Cek
Pindahkan
Gantikan
Set
Urutkan
Buka
Gandakan

English Version

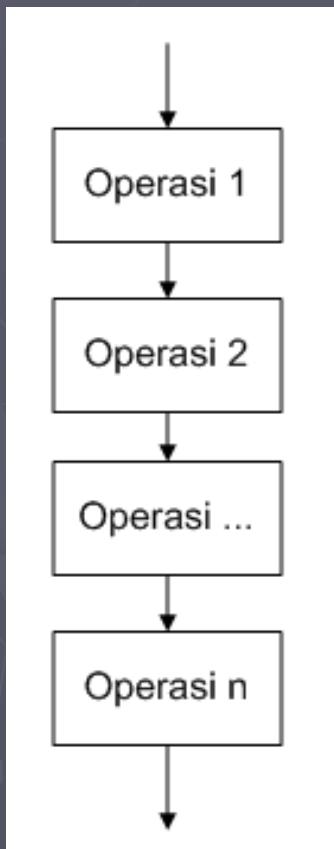
find, search, locate
add
multiply
subtract
divide
get, read, accept
display, write
compute
delete
validate
move
replace
set
sort
open
copy

Struktur 3 : Program Flowchart



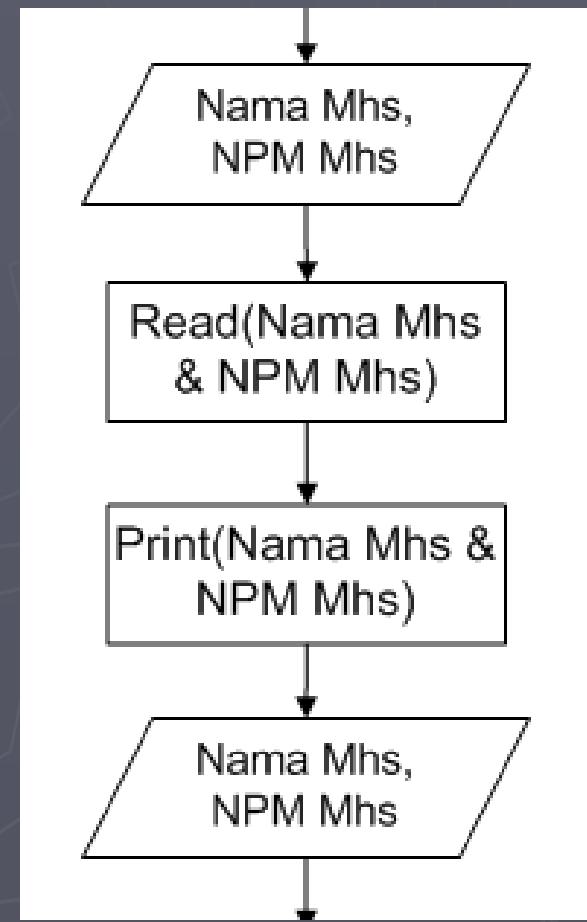
Alur Program : Sequence

langkah 1
langkah 2
langkah n



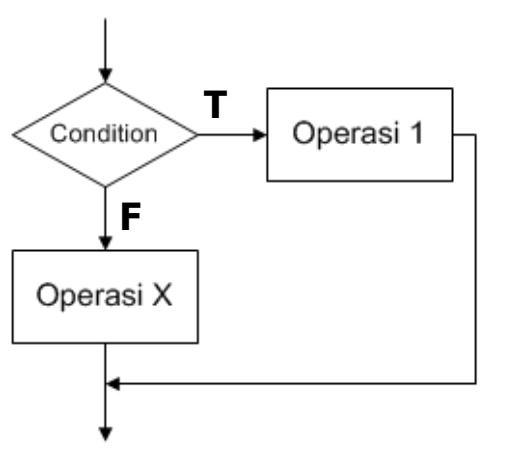
Contoh :

Read (Nama Mhs)
Read (NRP Mhs)
Print (Nama & NRP Mhs)

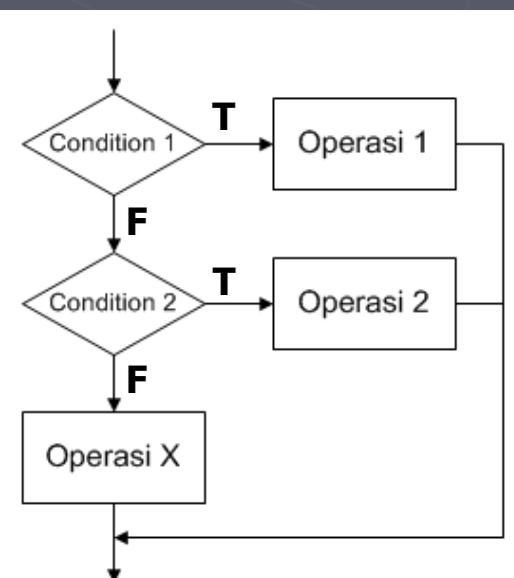


Alur Program : Condition (1)

```
If (kondisi) Then  
    Operasi 1  
Else  
    Operasi X  
End If
```

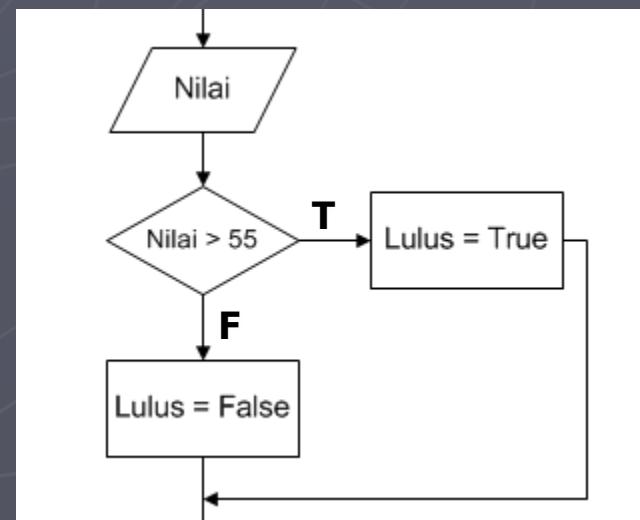


```
If (kondisi 1) Then  
    Operasi 1  
If Else (kondisi 2) Then  
    Operasi 2  
Else  
    Operasi X  
End If
```



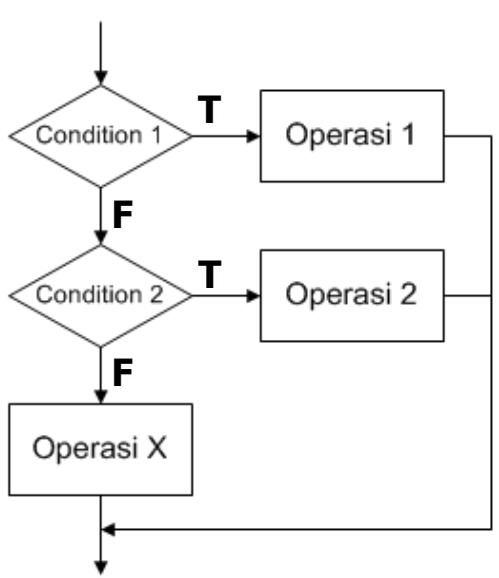
Contoh :

```
If (nilai>55) Then  
    Lulus = true  
Else  
    Lulus = false  
End If
```



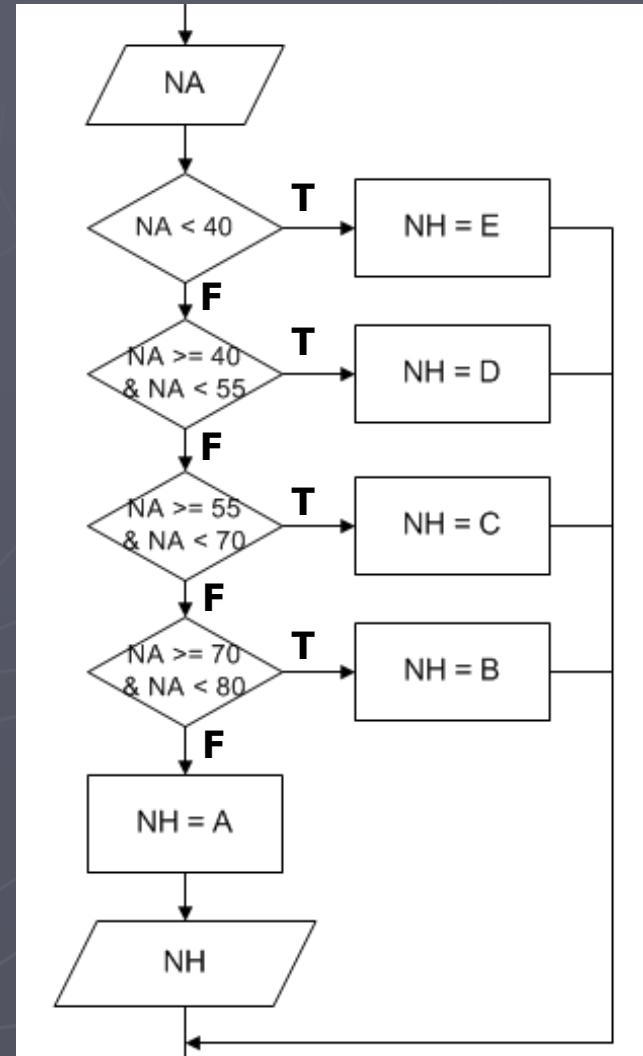
Alur Program : Condition (2)

```
Do Case  
Case (kondisi 1)  
    operasi 1  
Case (kondisi 2)  
    operasi 2  
.....  
Otherwise  
    operasi X  
End Case
```



Contoh :

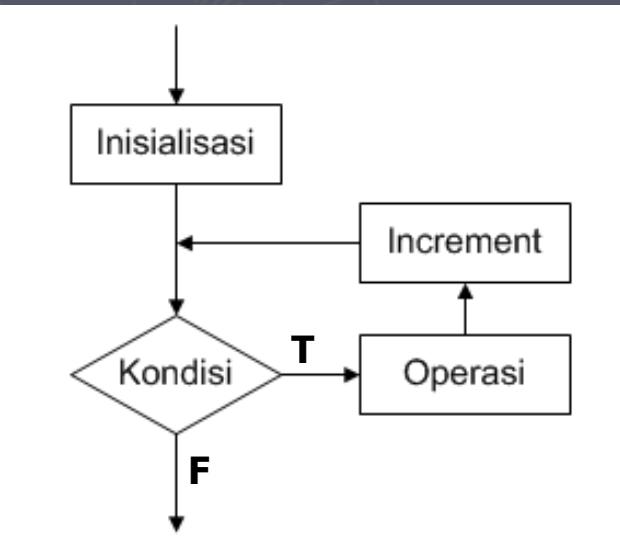
```
Do Case  
Case (NA<40)  
    NH= E  
Case (NA>=40 & NA<55)  
    NilaiHuruf = D  
Case (NA >=55 & NA<70)  
    NilaiHuruf = C  
Case (NA>= 70 & NA<80)  
    NH=B  
Otherwise  
    NH=A  
End Case
```



Alur Program : Repetition (1)

For (inisialisasi ; kondisi ; increment)
Urutan langkah

Inisialisasi
Do While (kondisi)
Urutan langkah
increment
End Do

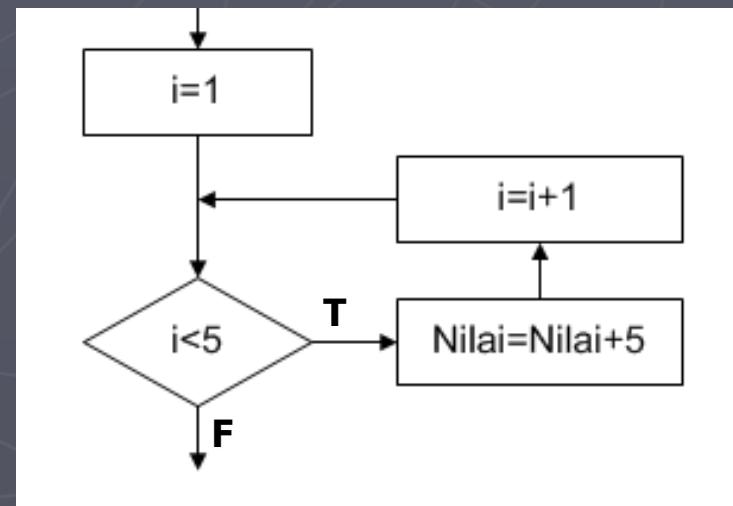


Contoh :

For (i=1 ; i<5 ; i++)
Nilai = Nilai + 5

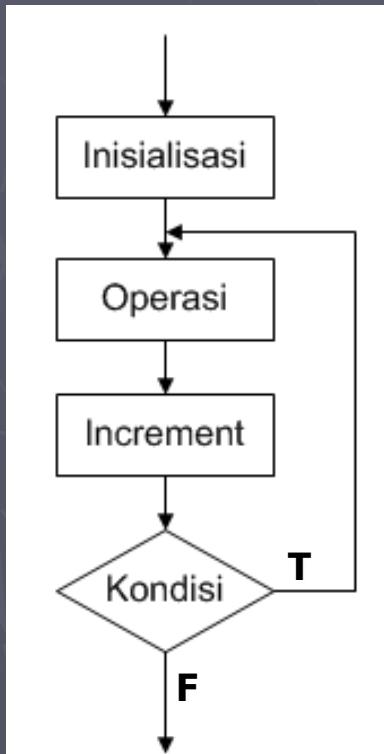
Contoh :

i=1
Do While (i<5)
Nilai = Nilai + 5
i=i+1
End Do



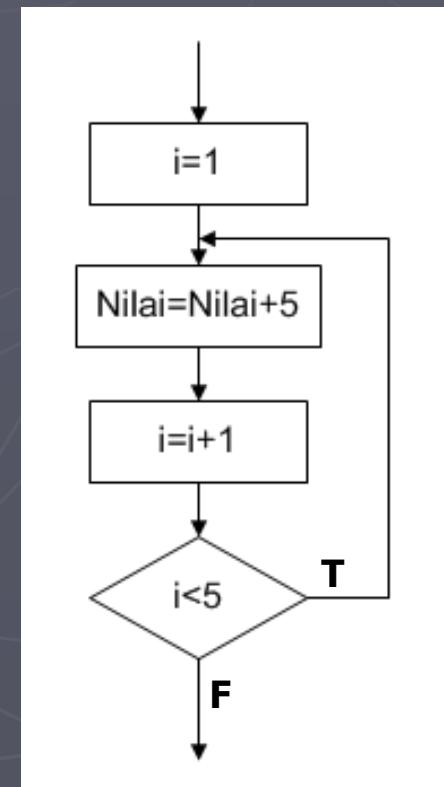
Alur Program : Repetition (2)

inisialisasi
Repeat
Urutan langkah
increment
Until (kondisi)



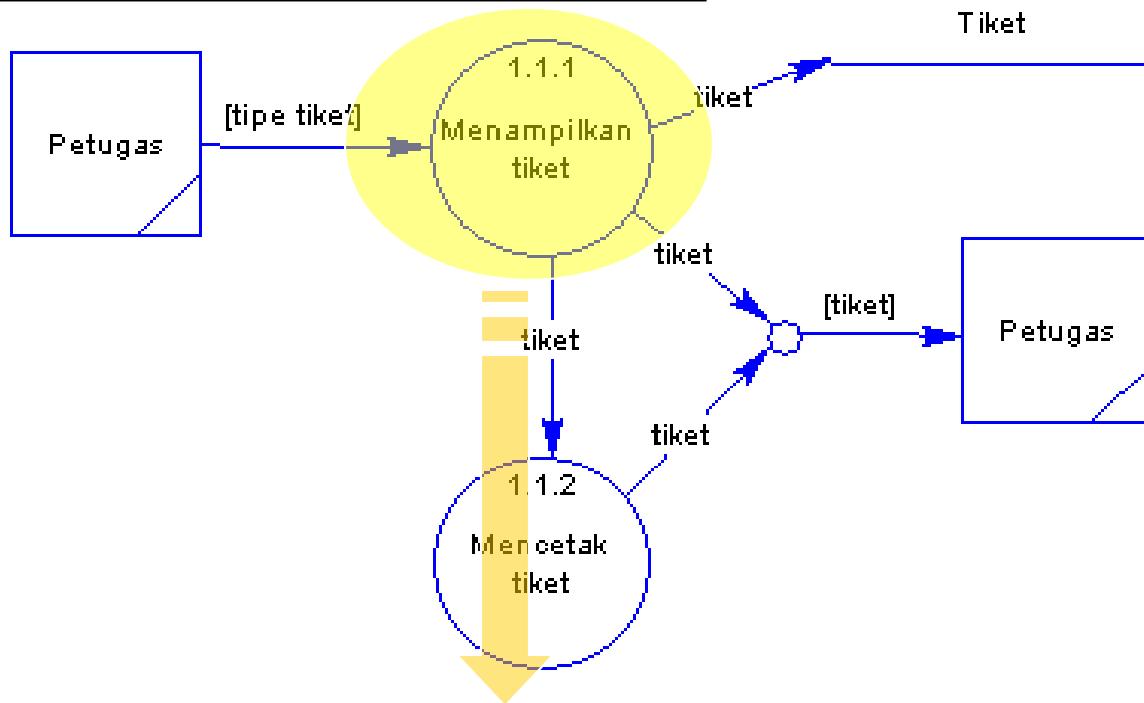
Contoh :

$i=1$
Repeat
 $Nilai = Nilai + 5$
 $i=i+1$
Until ($i < 5$)



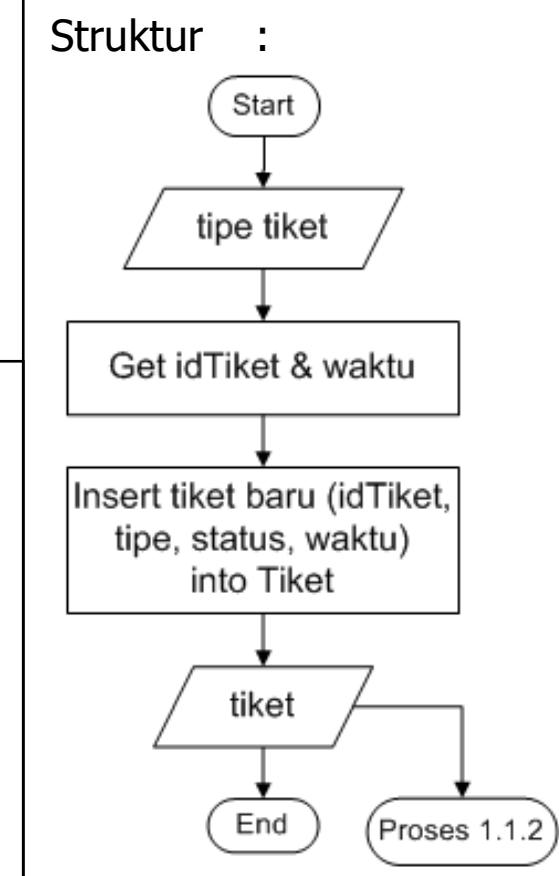
Contoh 1

DFD Level 2 : Proses 1.1 Membuat Tiket



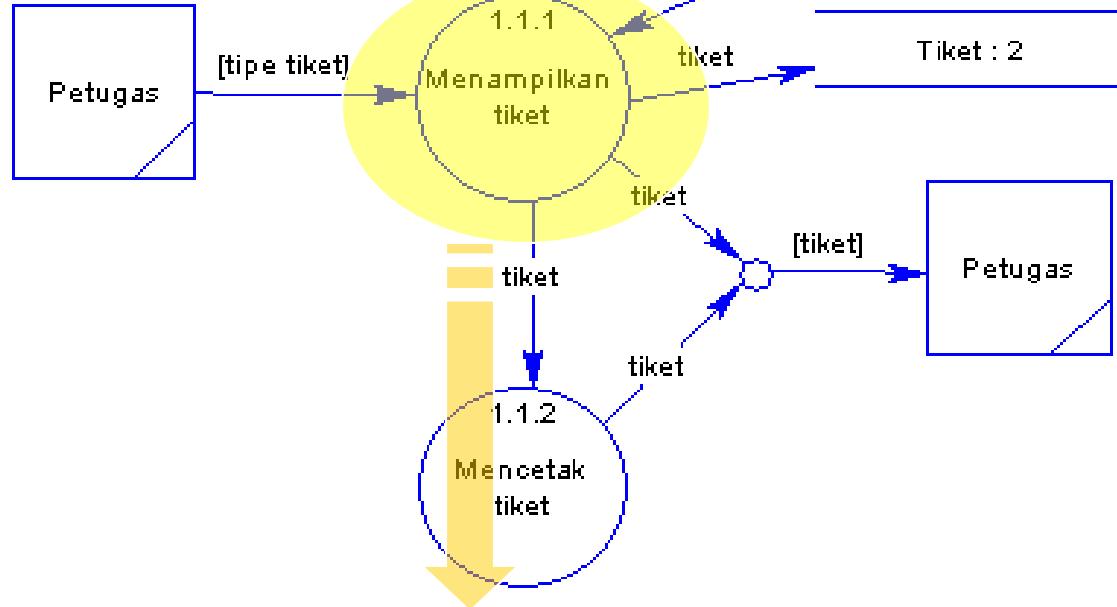
Spesifikasi Proses :

- Nomor : 1.1.1
Nama : Menampilkan tiket
Deskripsi : Menyimpan data tiket baru dan menampilkan di layar monitor
Input : tipe tiket (dari Petugas)
Output : tipe tiket (diterima oleh proses 1.1.2)
 tiket (diterima oleh data store Tiket)



Contoh 2

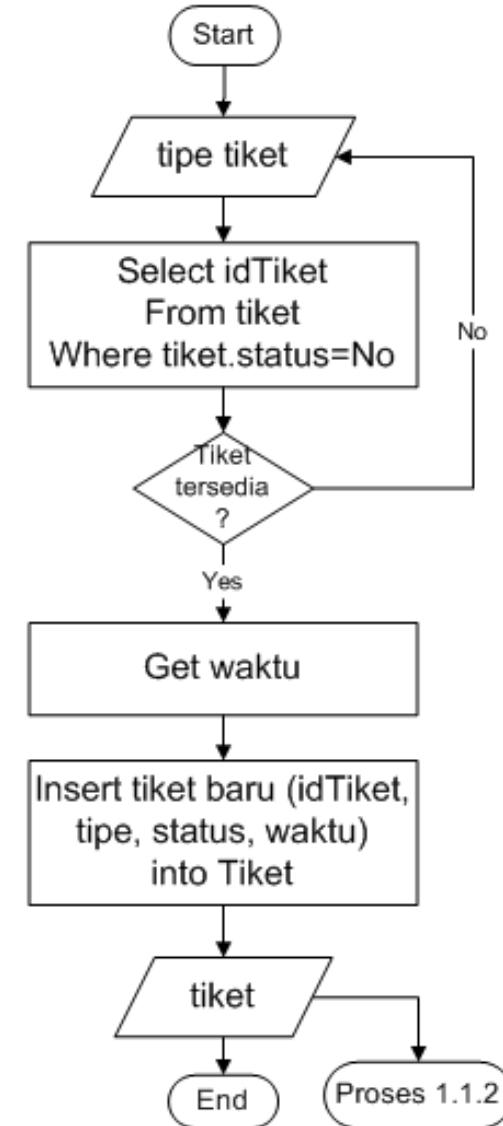
DFD Level 2 : Proses 1.1 Membuat Tiket



Spesifikasi Proses :

- Nomor : 1.1.1
Nama : Menampilkan tiket
Deskripsi : Memeriksa ketersediaan tiket baru, menyimpan data tiket baru, dan menampilkan di layar monitor
Input : tipe tiket (dari Petugas)
 tiket (dari data store Tiket)
Output : tipe tiket (ke petugas & proses 1.1.2)
 tiket (ke data store Tiket)

Struktur :

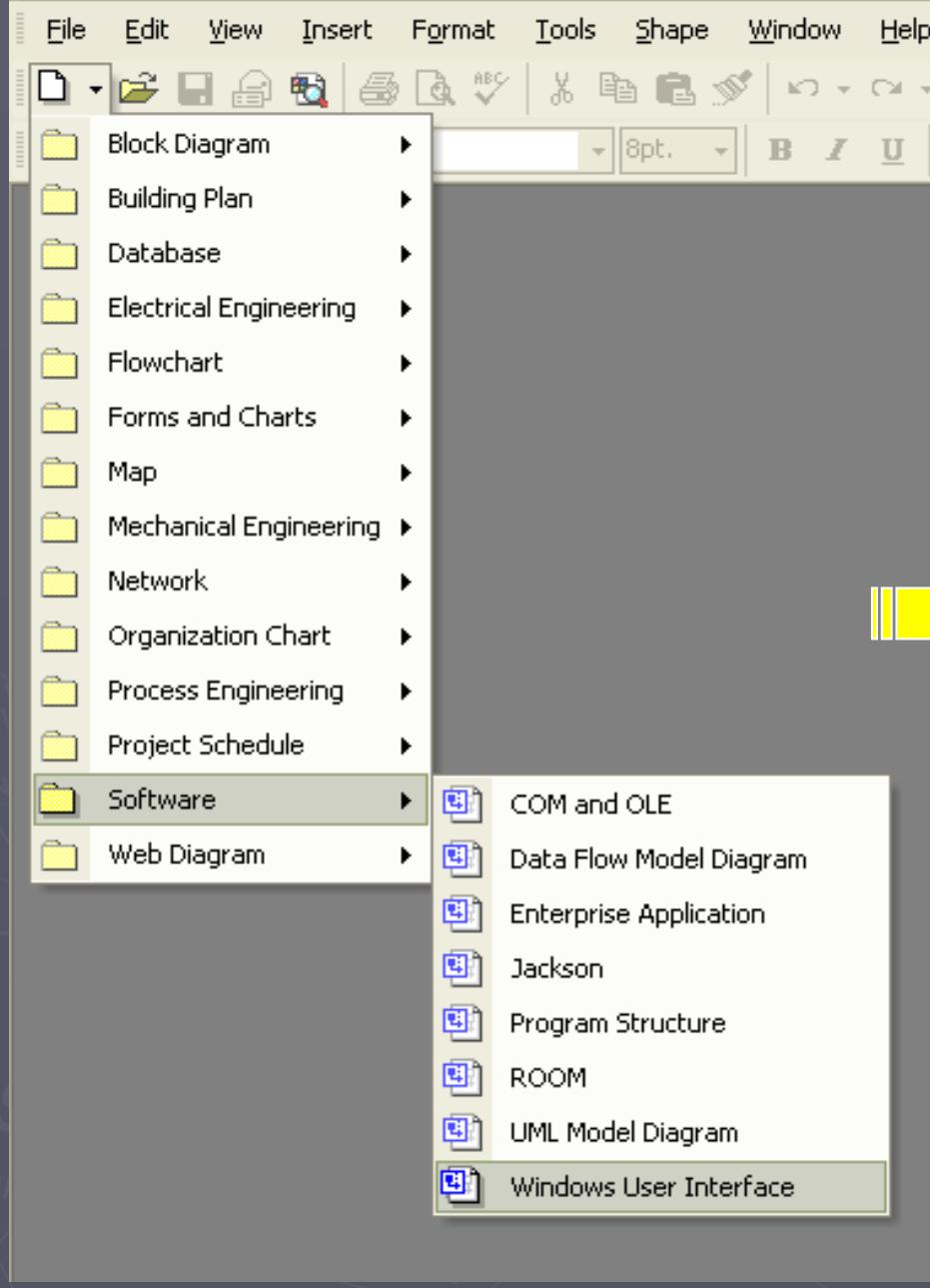


User Interface (Antarmuka Pengguna)

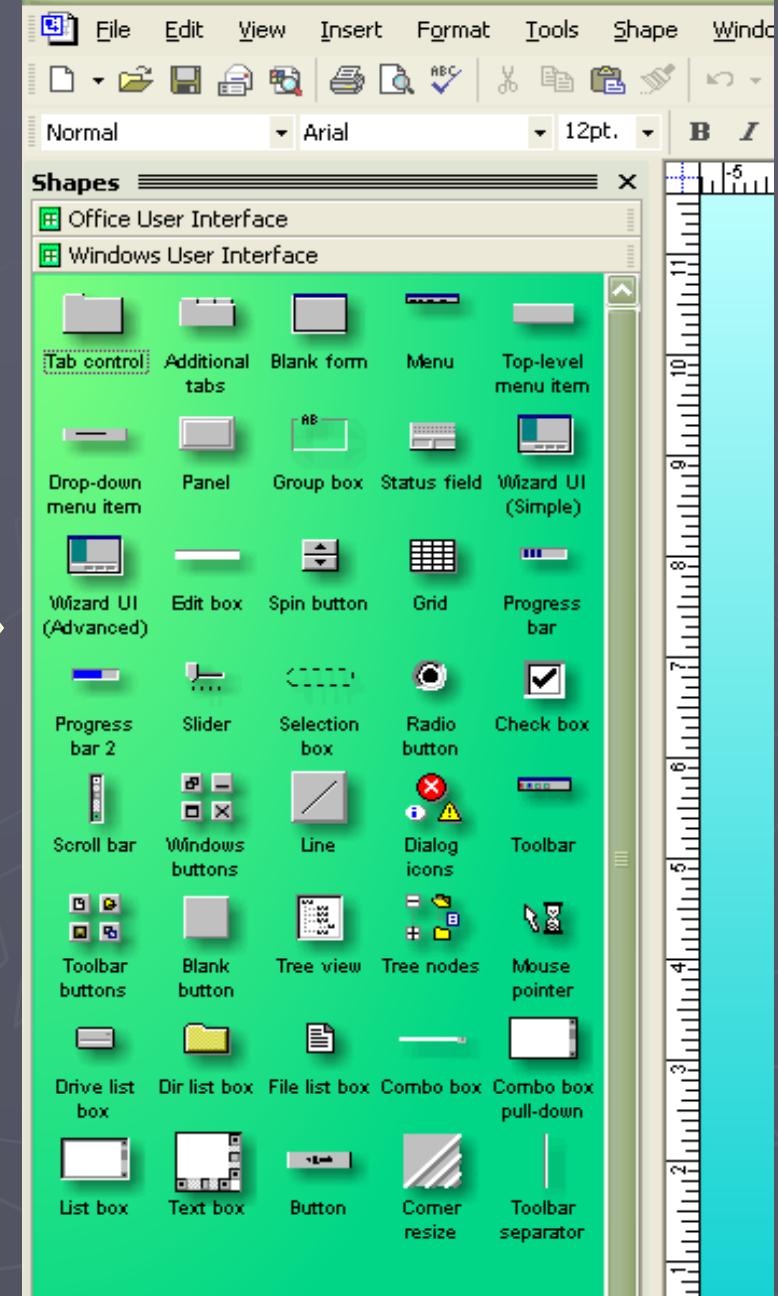
Data Dictionary : User Interface

- Interaksi antara Sistem dengan User memerlukan User Interface.
 - ↳ Data Flow dari User ke Proses atau sebaliknya → Tidak semua data flow dari/ke Terminator memerlukan user interface.
 - ↳ Data Flow yang memerlukan peran User (misalnya: kebutuhan User untuk memicu terjadinya input data)
- Elemen User Interface untuk Input ke Sistem :
 - Title : Nama user interface
 - Label : Keterangan mengenai Data apa yang perlu dimasukkan user.
 - Data Input : Data yang dimasukkan oleh User (diterima oleh Sistem).
 - Event : Tindakan user untuk memicu terjadinya proses input.
- Elemen User Interface untuk Output dari Sistem :
 - Title : Nama user interface
 - Label : Keterangan mengenai Data apa yang ditampilkan sistem.
 - Data Output : Data yang diterima oleh User (dihasilkan oleh Sistem).

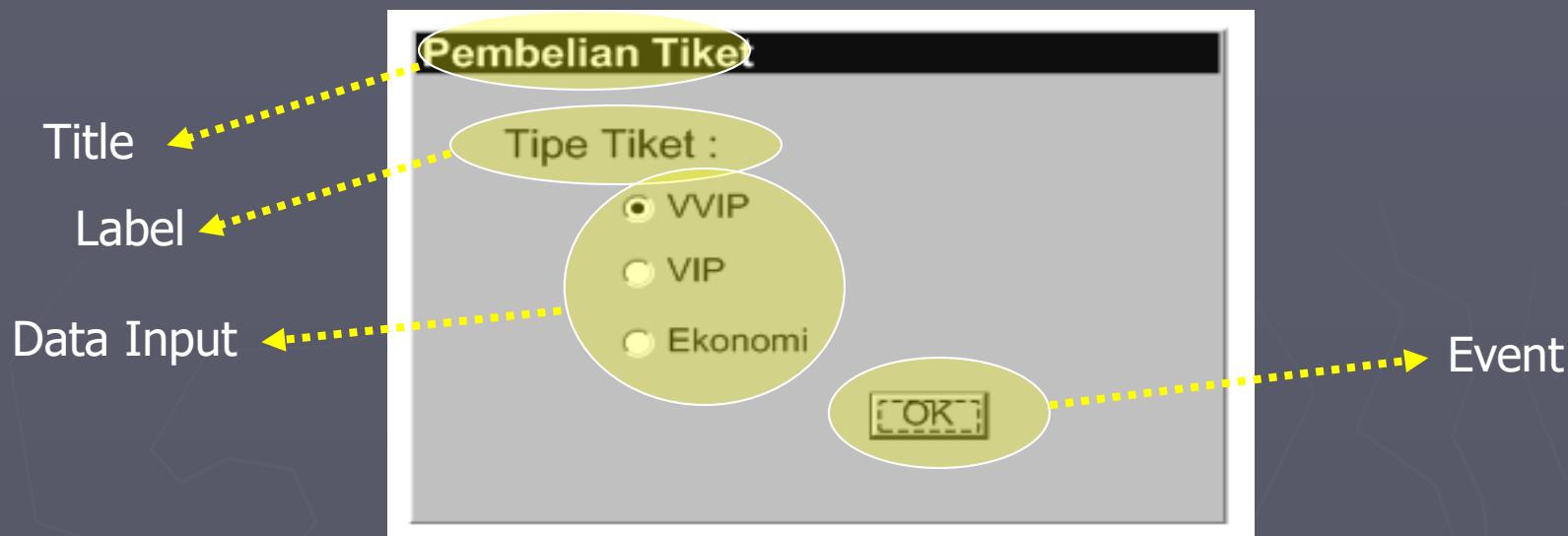
Microsoft Visio



Microsoft Visio - [Drawing5:Page-1]

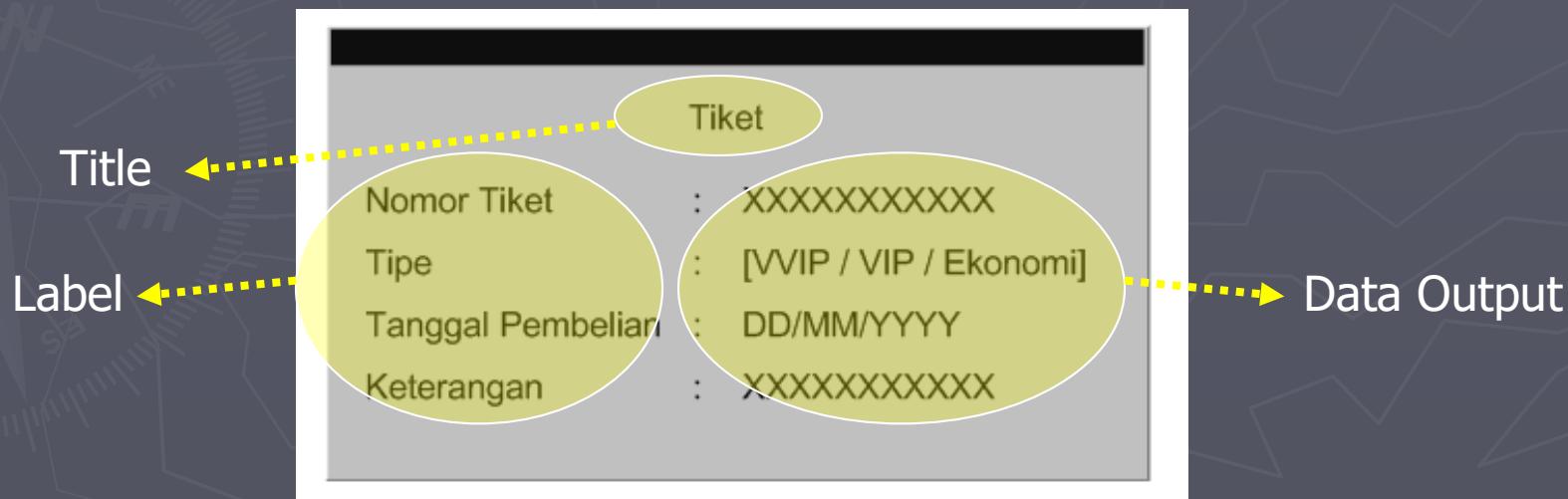


User Interface : Input ke Sistem



Input tipe tiket pada Proses 1.1.1 (Menampilkan tiket)

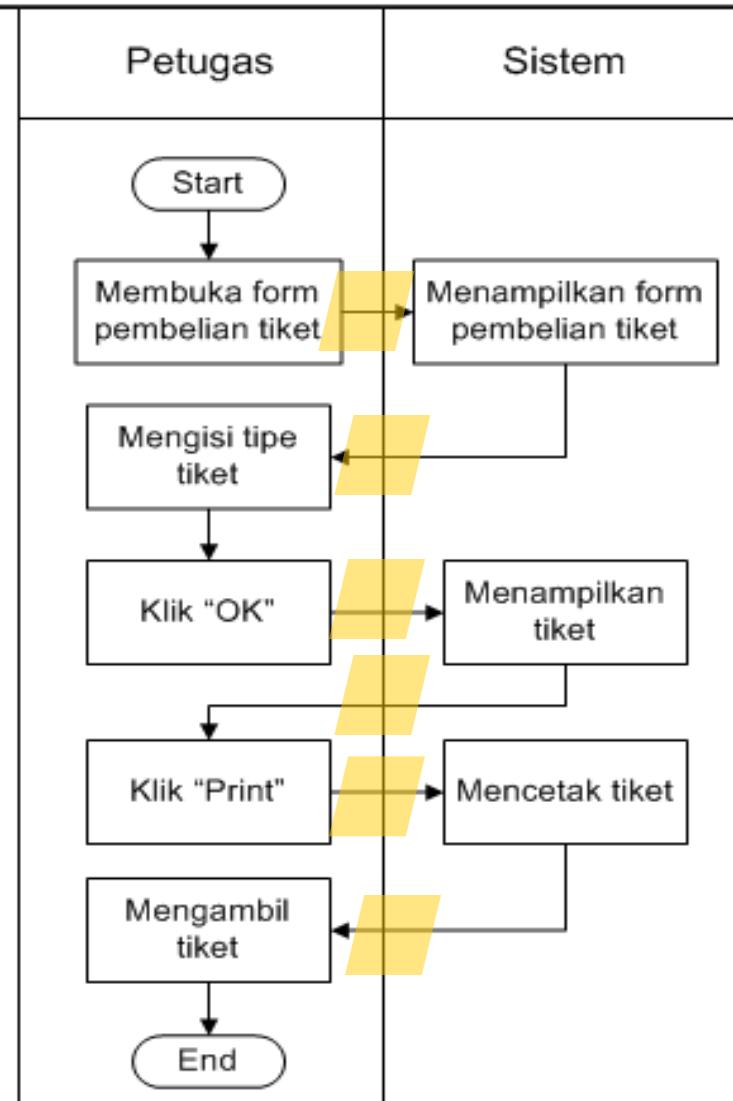
User Interface : Output dari Sistem



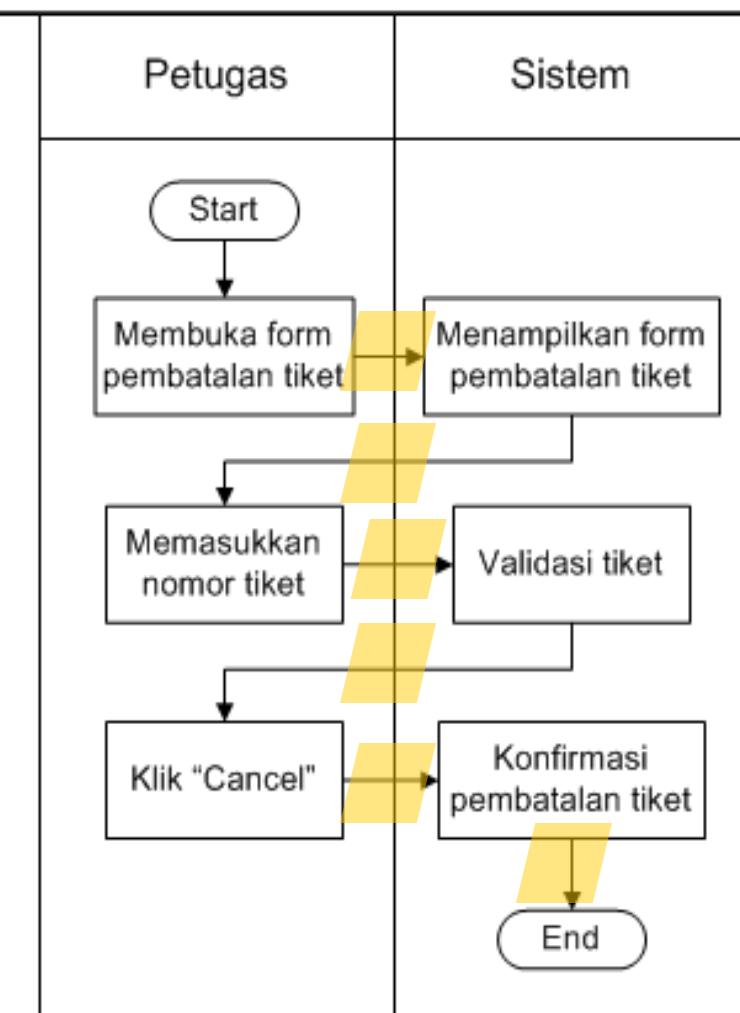
Output tiket pada Proses 1.1.1 (Menampilkan tiket)

Kebutuhan terhadap User Interface didasarkan pada System Workflow (interaksi User & Sistem), bukan DFD (interaksi Terminator & Sistem).

Membuat Tiket

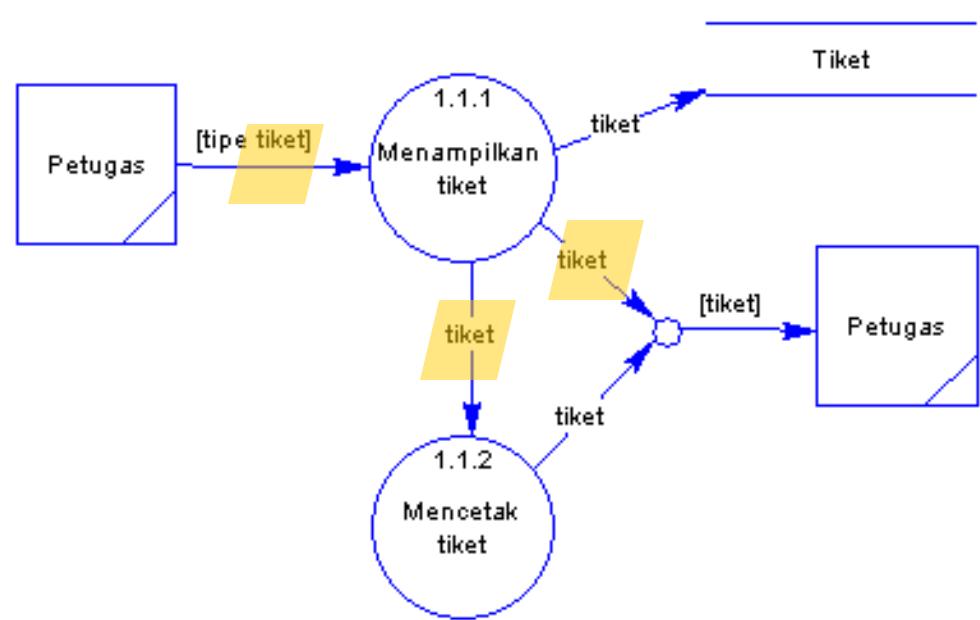
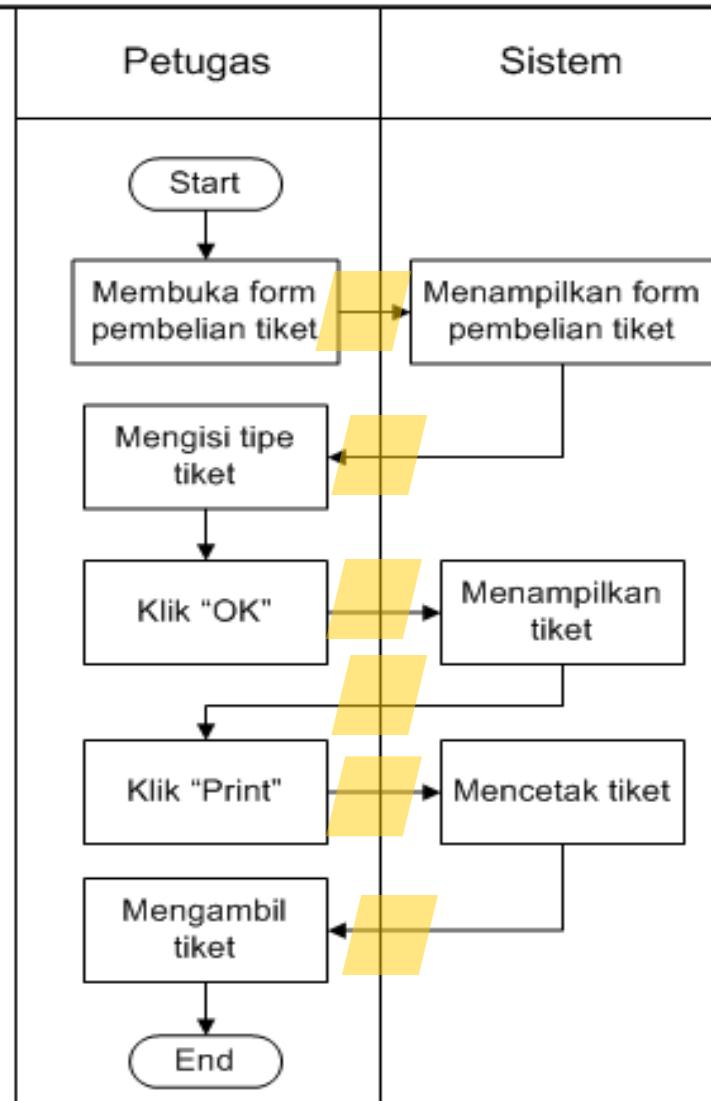


Membatalkan Tiket



Perbandingan kebutuhan terhadap user interface dari aktivitas I/O pada System Workflow dan DFD :

Membuat Tiket



- Dari perbandingan kedua model, kebutuhan terhadap user interface terlihat lebih jelas & lebih lengkap pada System Workflow.
- Desain user interface perlu dibuat untuk setiap aktivitas I/O pada System Workflow.

Data Modeling (Pemodelan Data)

Data Modeling

- Data Modeling dibuat sebelum DFD & Data Dictionary
 - ☞ Analisa dulu semua elemen Basisdata yang dibutuhkan berkenaan dengan Sistem secara keseluruhan → Buat ERD / CDM dan PDM-nya.
 - ☞ Data Store pada DFD ditentukan berdasarkan pada setiap Entitas pada ERD / CDM (Tabel pada PDM).
 - ☞ Cara ini lebih memudahkan dalam pembuatan DFD, tetapi memerlukan analisa basisdata yang lebih rumit (karena harus mengidentifikasi semua elemen basisdata tanpa mengetahui secara detail aliran data yang berlangsung).
- Data Modeling dibuat setelah DFD & Data Dictionary
 - ☞ Analisa dulu seluruh proses dan aliran data pada Sistem → Buat DFD.
 - ☞ Identifikasi setiap Data Store yang dibutuhkan → nantinya digunakan sebagai elemen Entitas pada ERD / CDM (Tabel pada PDM).
 - ☞ Cara ini lebih sulit dalam pembuatan DFD (karena harus cermat dalam mengidentifikasi setiap Data Store yang dibutuhkan), tetapi memudahkan dalam analisa basisdata (dengan sudah tersedianya elemen Data Store pada DFD).

Data Modeling

- **Pada Perkuliahan ini :**
 - ☞ Data Modeling dibuat setelah DFD dan Data Dictionary.
- **Entity Relationship Diagram (ERD)**
 - ☞ Merupakan bentuk organisasi data yang masih bersifat konseptual, berisi sejumlah elemen data yang belum diimplementasikan dalam basisdata secara riil / fisik.
 - ☞ Implementasi pada Power Designer : Conceptual Data Model (CDM).
- **Physical Data Model (PDM)**
 - ☞ Merupakan bentuk organisasi data yang sudah bersifat aktual, berisi sejumlah elemen data yang riil (bentuk implementasi elemen data dalam basisdata secara fisik).
- **Perangkat Lunak Bantu :**
 - ☞ CDM : PowerDesigner 6 >> DataArchitect (atau versi diatas 6)
 - ☞ PDM : Generating from CDM

Data Modeling

▪ Komponen ERD:

☞ Entity (Entitas)

- ✓ Suatu obyek yang dapat diidentifikasi berkaitan dengan Sistem.
Contoh: Mahasiswa, Mata Kuliah, dsb.



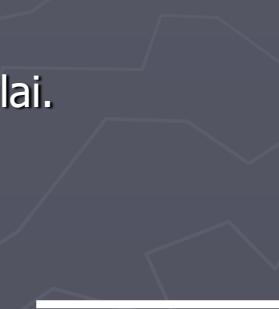
☞ Attribute (Atribut)

- ✓ Karakter atau sifat yang dimiliki oleh Entitas.
- ✓ **Simple Attribute** → tidak bisa dibagi lagi.
Contoh: Usia, Status Pernikahan, Jenis Kelamin, dsb.
- ✓ **Composite Attribute** → bisa dibagi menjadi beberapa sub-attribute.
Contoh: Alamat → terdiri atas Nama Jalan, Nomor Rumah, Kota, dsb.
- ✓ **Single-Valued Attribute (Key Attribute)** → memiliki nilai tunggal / unik.
Contoh: NoKTP, NPM, SerialNumber, dsb.
- ✓ **Multi-Valued Attribute** → bisa memiliki beberapa nilai.
Contoh: NoTelp, Alamat, Pekerjaan, dsb.

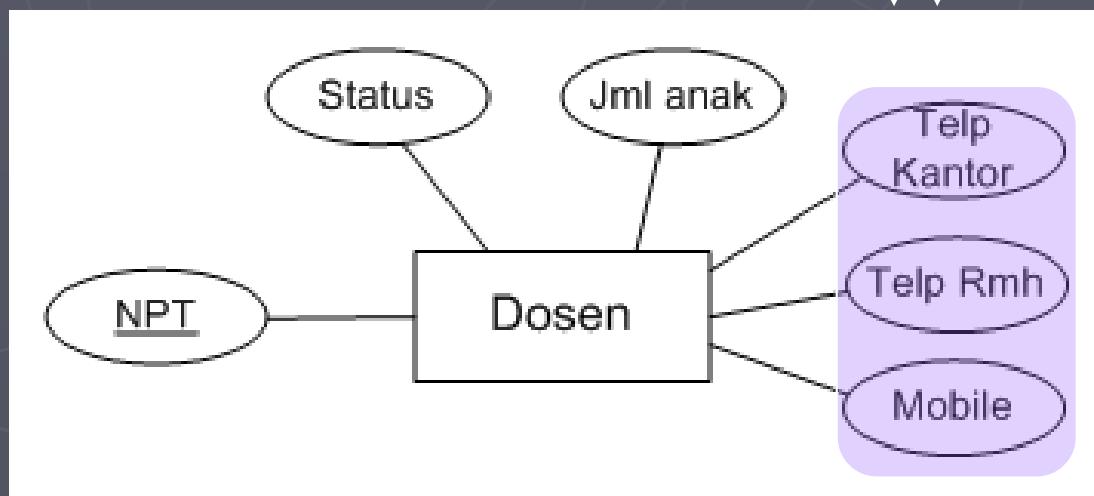
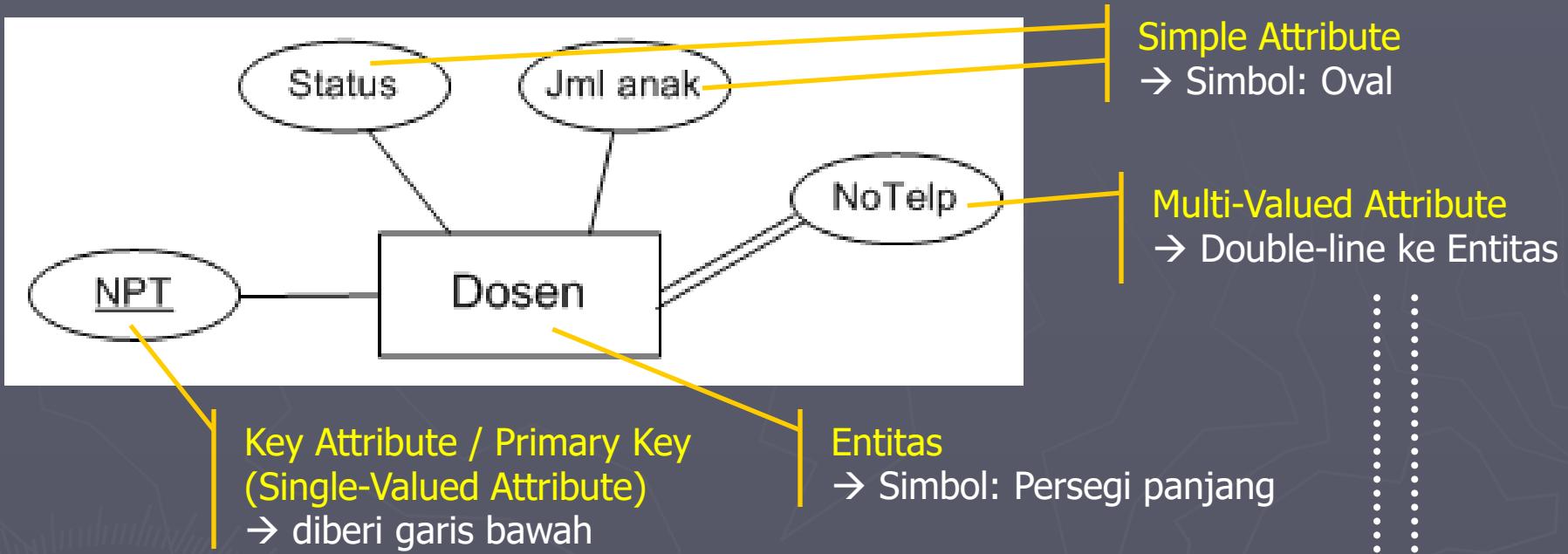


☞ Relationship (Relasi)

- ✓ Hubungan antar entitas yang berbeda.



ERD



ERD

▪ Connectivity

- ☞ Mendeskripsikan tipe relasi antar entitas : one-to-one, one-to-many, dan many-to-many.



1 Dosen bisa mengajar banyak MK, dan 1 MK hanya diajar oleh 1 Dosen.



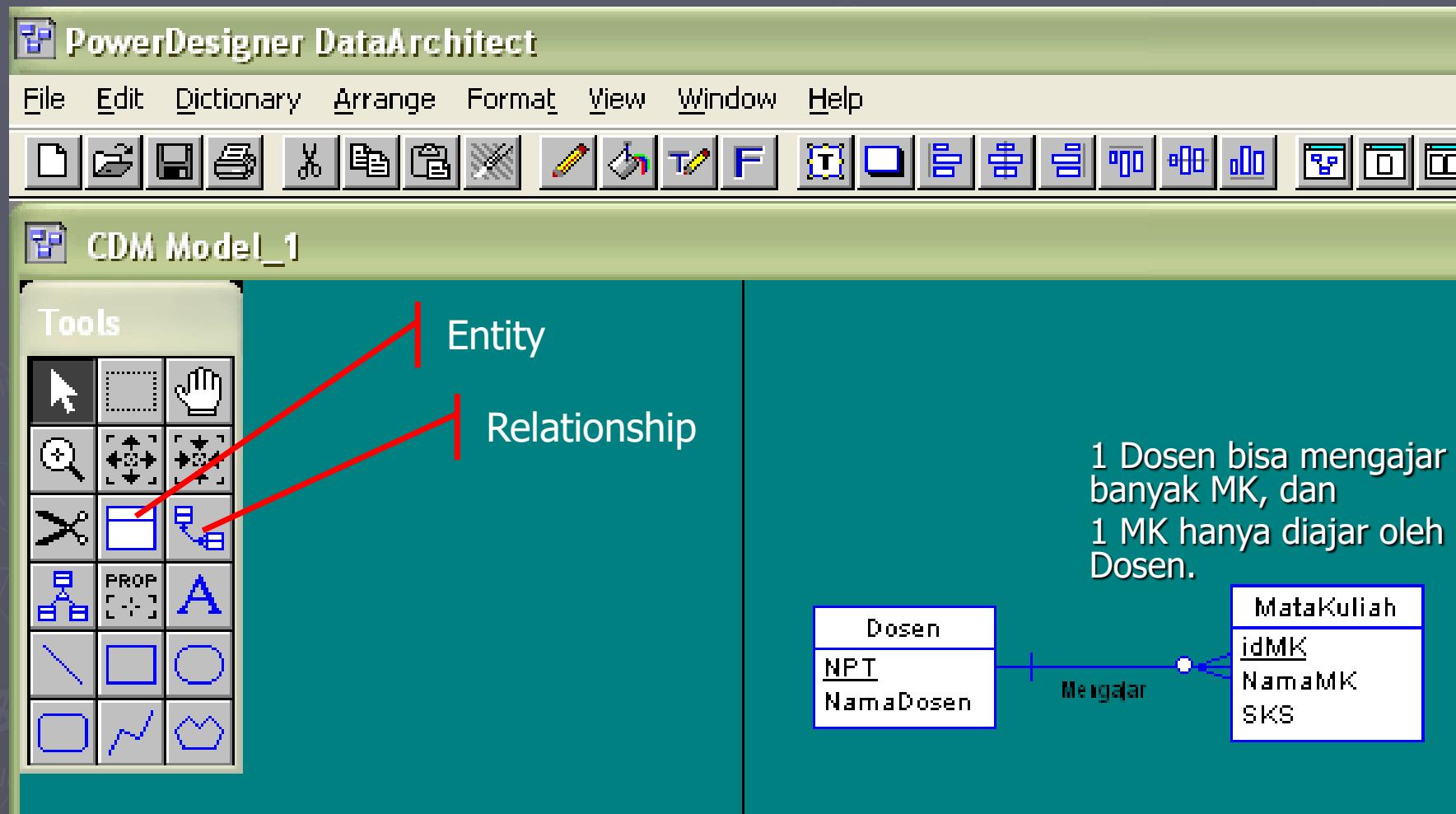
1 Mahasiswa bisa mengambil banyak MK, dan 1 MK bisa diambil oleh banyak Mahasiswa.



1 Mahasiswa hanya memiliki 1 Transkrip Nilai, dan 1 Transkrip Nilai hanya dimiliki oleh 1 Mahasiswa.

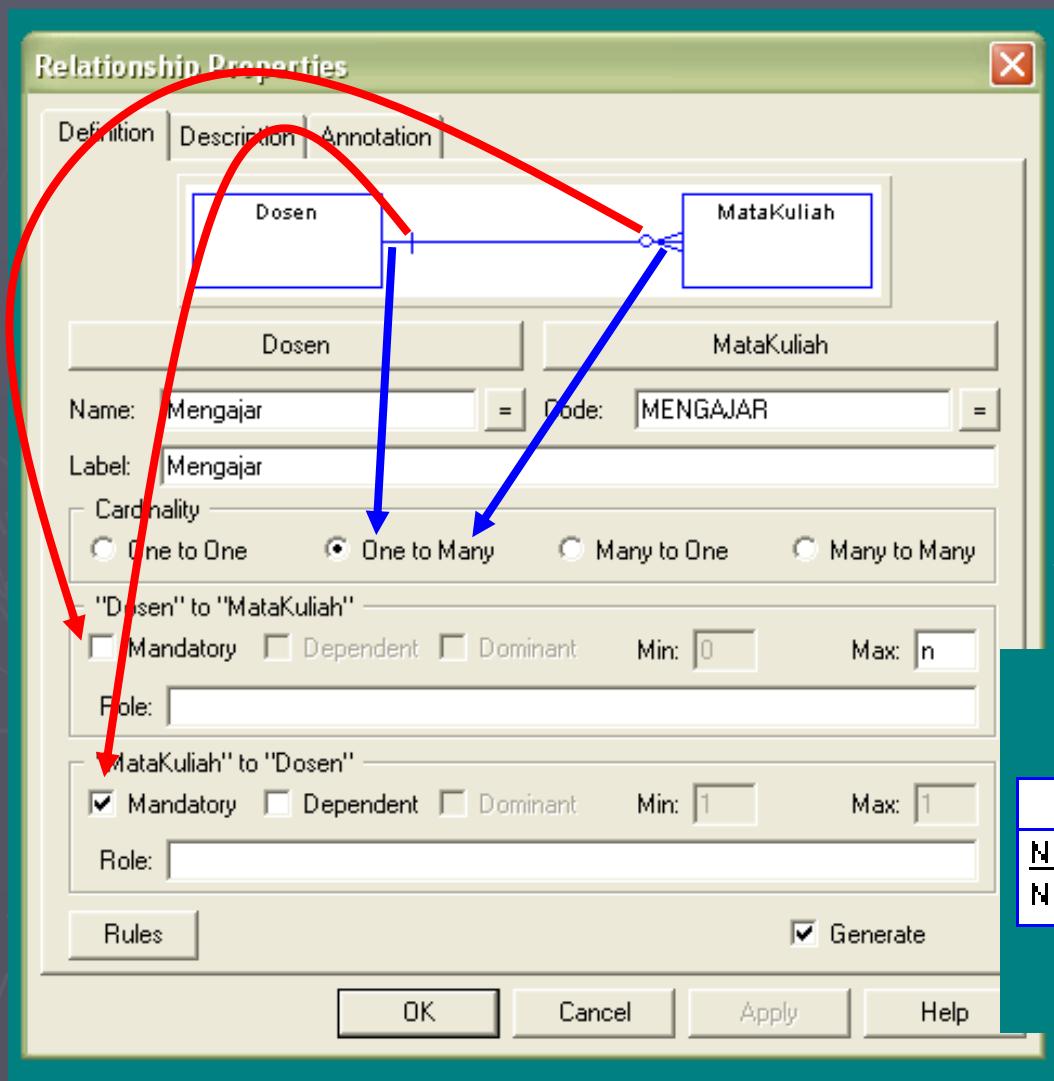
CDM

- Membuat CDM dengan Power Designer – Data Architect



CDM

- Membuat CDM dengan Power Designer – Data Architect



Keterlibatan semua Entitas dalam relasinya dengan Entitas lain.

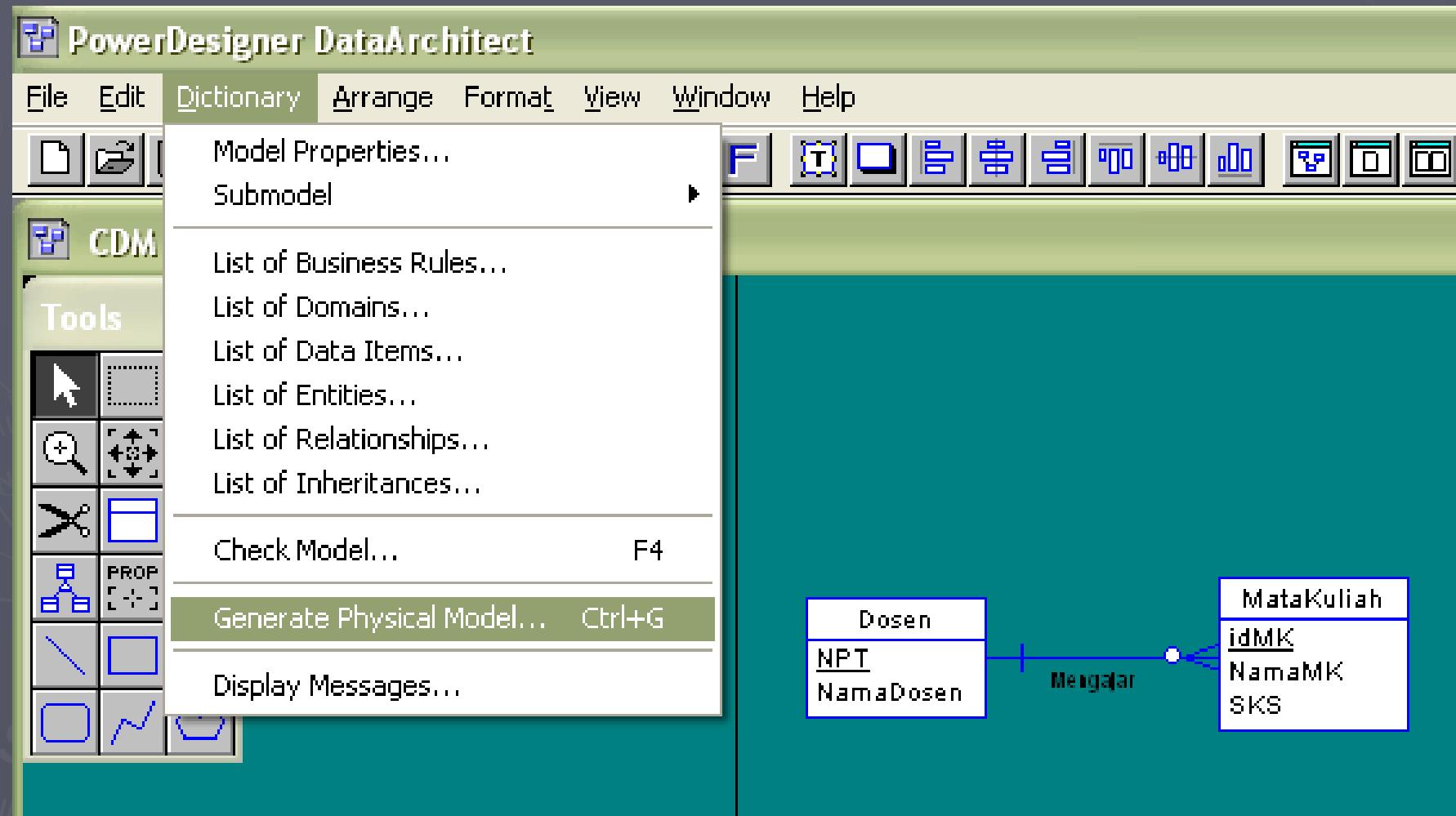


1 Dosen bisa mengajar banyak MK, dan
1 MK hanya diajar oleh 1 Dosen.



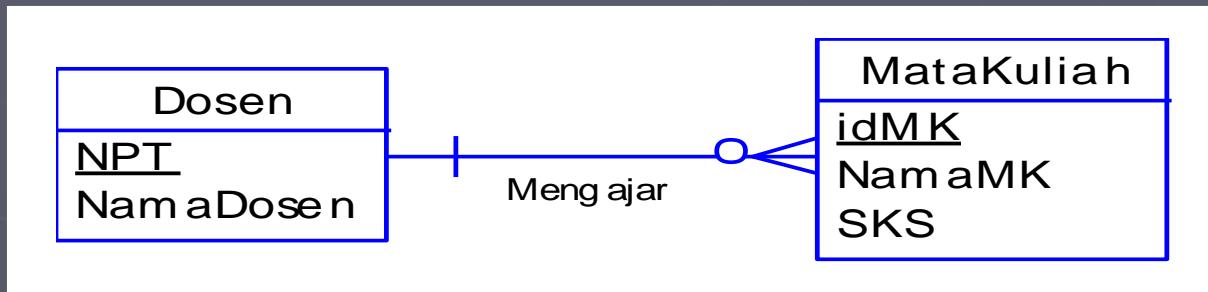
PDM

- Membuat PDM : Generating CDM

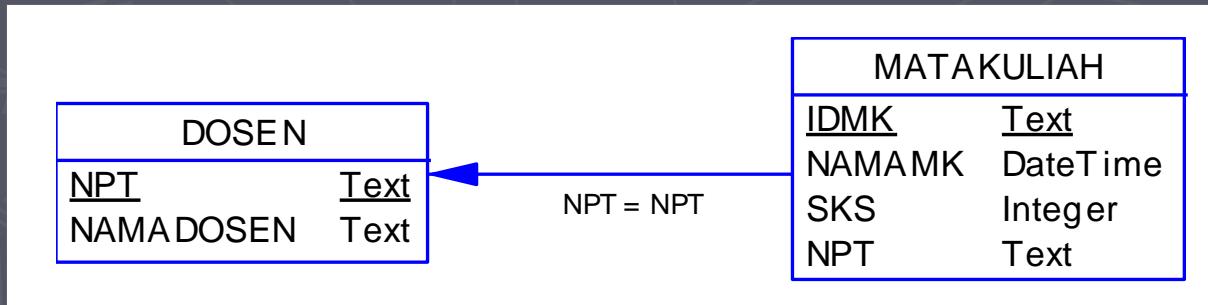


CDM & PDM

- Conceptual Data Model (CDM)



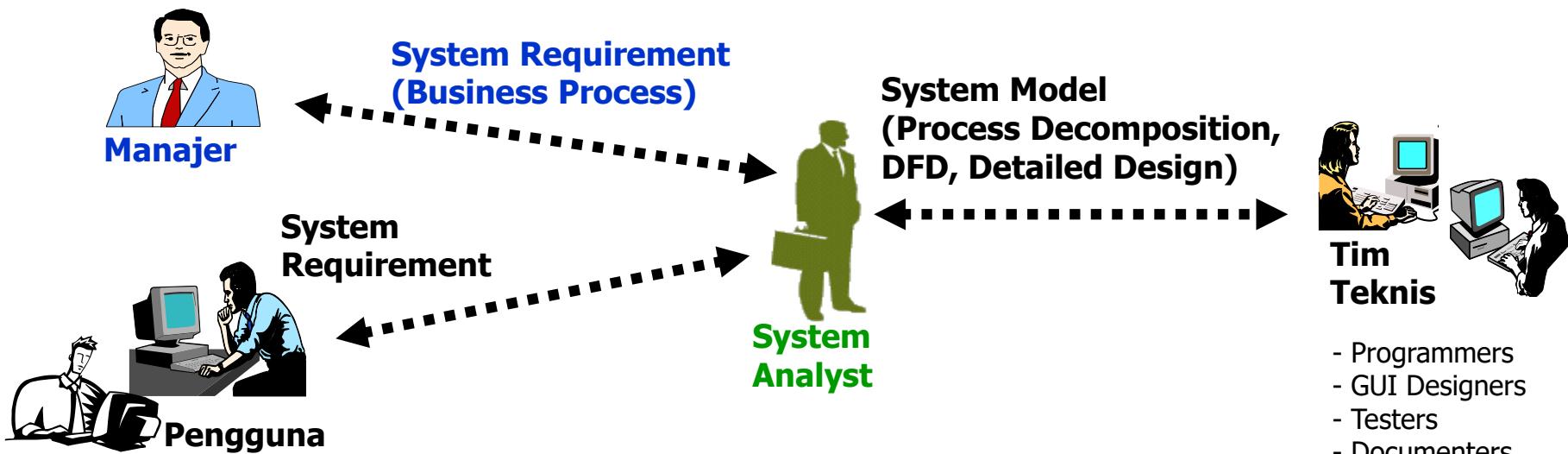
- Physical Data Model (PDM)



Hasil Analisis dan Desain

Hasil Analisis & Desain

- System Analyst membutuhkan komunikasi dengan berbagai pihak untuk menyamakan persepsi, persetujuan, pengendalian, & evaluasi.
- Dokumentasi hasil analisis dan desain perlu dibuat sebagai bahan komunikasi yang valid dan resmi.
Komunikasi lisan tanpa disertai dokumentasi tertulis tidak bisa dipertanggungjawabkan (secara legal formal) dan bisa menimbulkan kesalahan persepsi / pemahaman.



Software Engineering : Modeling & Programming

- Forward Engineering (from Model to Source Code)
 - ✓ Pemodelan Sistem dibuat terlebih dahulu sebagai acuan untuk pembuatan kode-kode program.
 - ✓ Dilakukan jika waktu dan kebutuhan sumber daya memadai.
- Reverse Engineering (from Source Code to Model)
 - ✓ Pemodelan Sistem dibuat berdasarkan kode-kode program yang telah tersedia.
 - ✓ Dilakukan pada saat mendesak (waktu dan sumber daya terbatas).

Sistem Baru

Pendekatan strategi untuk menghasilkan sistem baru :

1. Modifikasi (Perbaikan)

- Menambah fungsi-fungsi baru, memperbaiki / menghilangkan fungsi-fungsi yang bermasalah.
- Jika proses bisnis tidak berubah atau perubahan tidak signifikan.
- Jika teknologi (penyimpanan data, kecepatan proses, keamanan, dsb) masih mencukupi kebutuhan sistem.

2. Substitusi (Penggantian)

- Membuat sistem baru (memiliki persamaan atau bahkan berbeda sama sekali dengan sistem lama) dan membuang sistem lama.
- Jika proses bisnis berubah secara signifikan.
- Jika teknologi tidak mencukupi kebutuhan sistem.

Kesuksesan Pengembangan Sistem

- ✓ Tepat waktu (*on time*).
- ✓ Biaya minimal (*minimum cost*), baik dalam pembangunan sistem maupun pemeliharaan.
- ✓ Kemampuan dan keamanan sistem baru yang optimal (*optimal performance and security*).
- ✓ Sistem baru sesuai dengan kebutuhan pengguna (*appropriate user requirement*).
- ✓ Operasional dan Pemeliharaan Sistem berjalan sesuai dengan prosedur yang benar (good *operation and maintenance*).



Terima
kasih

Thank You!
;)!