



Rekayasa Perangkat Lunak

Pertemuan 5 Pemodelan Analisis

Pemodelan Fungsional

- Mendeskripsikan seluruh fungsi yang terlibat dalam PL
- Tool:
 - Context Diagram
 - ✓ Diagram level tertinggi dari DFD yang menggambarkan hubungan sistem dengan lingkungan luarnya.
 - DFD
 - ✓ Menggambarkan bagaimana data ditransformasikan pada PL
 - ✓ Menggambarkan Fungsi-fungsi yang mentransformasikan data
 - ✓ Berisi deskripsi dari setiap fungsi yang muncul pada DFD



Data Flow Diagram (DFD)

- DFD bukan flowchart
- Proses dalam DFD bisa berjalan secara paralel
- DFD menggambarkan aliran data dalam sebuah sistem
- DFD adalah Data yang tersimpan dan proses dengan proses yang terhubung dengan data tersebut
- Tidak ada loop ataupun cabang dalam DFD
- DFD menggambarkan semua proses, meskipun proses tersebut terjadi dalam waktu yang berbeda.

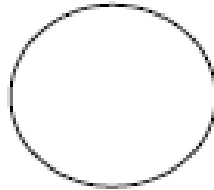


Komponen DFD

Menurut Yourdan dan DeMarco



Terminator



Proses



Data Store

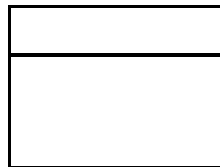


Alur Data

Menurut Gene dan Serson



Terminator



Proses



Data Store



Alur Data

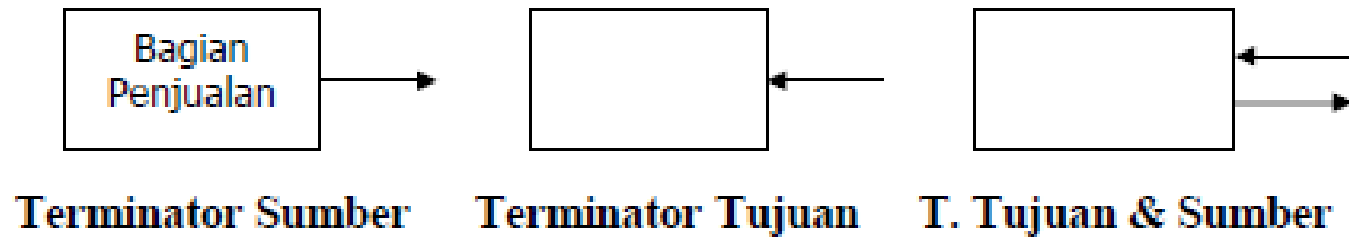


- Terminator

Biasanya terminator dikenal dengan nama entitas luar (external entity). Penulisannya menggunakan **kata benda**

- Terdapat dua jenis terminator :

1. Terminator Sumber (source) : merupakan terminator yang menjadi sumber.
2. Terminator Tujuan (sink) : merupakan terminator yang menjadi tujuan data / informasi sistem.



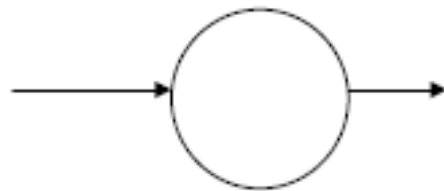
Ada tiga hal penting yang harus diingat tentang terminator :

1. Terminator merupakan bagian/lingkungan luar sistem. Alur data yang menghubungkan terminator dengan berbagai proses sistem, menunjukkan hubungan sistem dengan dunia luar.
2. Profesional Sistem Tidak berhak mengubah isi atau cara kerja organisasi atau prosedur yang berkaitan dengan terminator
3. Hubungan yang ada antar terminator yang satu dengan yang lain tidak digambarkan pada DFD.

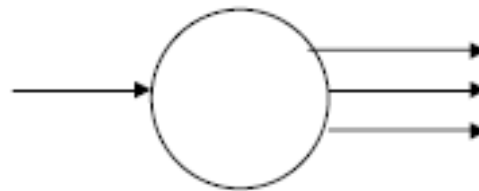


- Proses

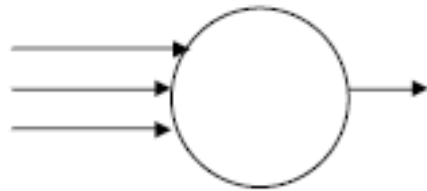
Menggambarkan bagian dari sistem yang mentransformasikan input menjadi output. Penulisannya menggunakan **kata kerja transitif**



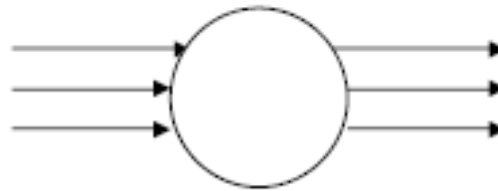
1 input & 1 output



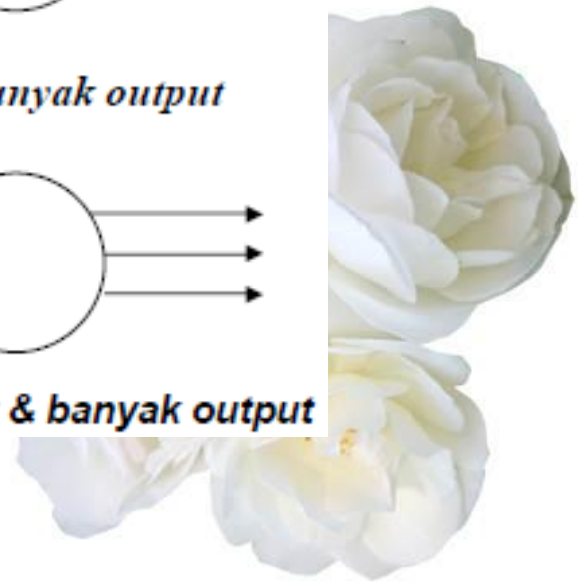
1 input & banyak output



Banyak input & 1 output



Banyak input & banyak output



Ada beberapa hal yang perlu diperhatikan tentang proses :

1. Proses harus memiliki input dan output.
2. Setiap proses harus diberi level no.urut yang jelas
3. Proses dapat dihubungkan dengan komponen terminator, data store atau proses melalui alur data.
4. Sistem/bagian/divisi/departemen yang sedang dianalisis oleh profesional sistem digambarkan dengan komponen proses.

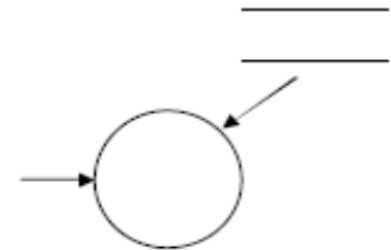


- Data Store

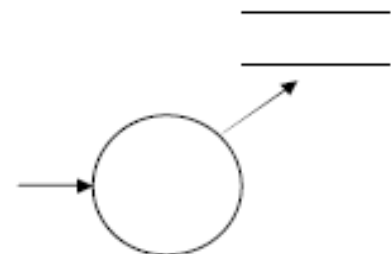
Digunakan untuk membuat model sekumpulan paket data dan diberi nama dengan **kata benda jamak**

- Implementasi Data Store

- ✓ **Alur data dari data store** : Pembacaan atau pengaksesan satu paket tunggal data, lebih dari satu paket data, dari satu paket tunggal data, atau sebagian dari lebih dari satu paket data untuk suatu proses.



- ✓ **Alur data ke data store** : pengupdatean data, seperti menambah satu paket data baru atau lebih, menghapus satu paket atau lebih, atau mengubah/memodifikasi satu paket data atau lebih.



- Alur Data

Digunakan untuk menerangkan perpindahan data atau paket data/informasi dari satu bagian sistem ke bagian lainnya.

- Konsep Alur Data

- ✓ Konsep Paket Data (*Packets of Data*)
- ✓ Konsep Alur Data Menyebar (*Diverging Data Flow*)
- ✓ Konsep Alur Data Mengumpul (*Converging Data Flow*)
- ✓ Konsep Sumber atau Tujuan Alur Data



Penggambaran DFD

1. Identifikasi terlebih dahulu semua entitas luar yang terlibat di sistem.
2. Identifikasi semua input dan output yang terlibat dengan entitas luar.
3. Buat Diagram Konteks (diagram context)
 - ✓ Tentukan nama sistemnya.
 - ✓ Tentukan batasan sistemnya.
 - ✓ Tentukan terminator apa saja yang ada dalam sistem.
 - ✓ Tentukan apa yang diterima/diberikan terminator dari/ke sistem.
 - ✓ Gambarkan diagram konteks.
4. Buat Diagram Level Zero
5. Buat Diagram Level Satu
6. DFD Level Dua, Tiga, ...



[Contoh](#)

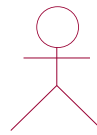
Pemodelan Tingkah Laku

- Mendeskripsikan status sistem yang dapat muncul ketika perangkat lunak digunakan
- Mendeskripsikan kelakuan sistem
- Tool:
 - ✓ UML
- Umumnya digunakan pada saat waktu nyata

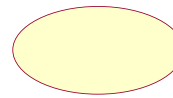


Use Case Diagram

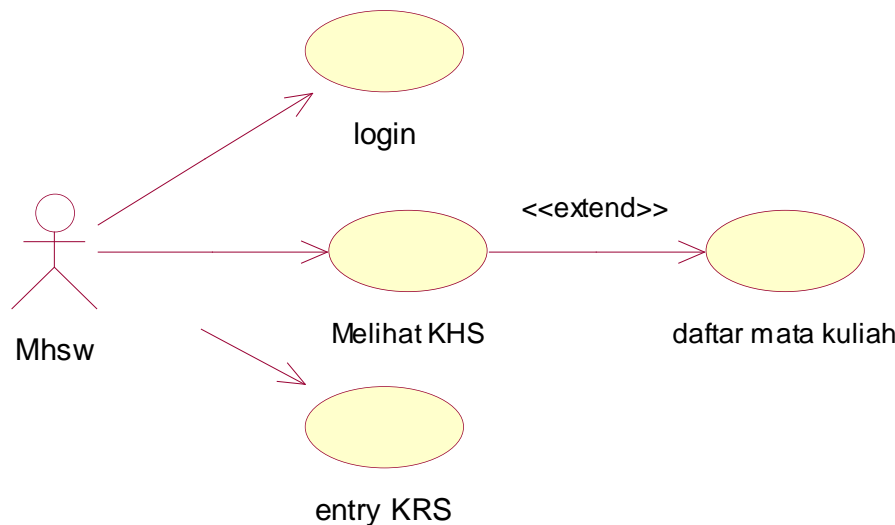
- Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem.
- Komponen Use Case



Actor

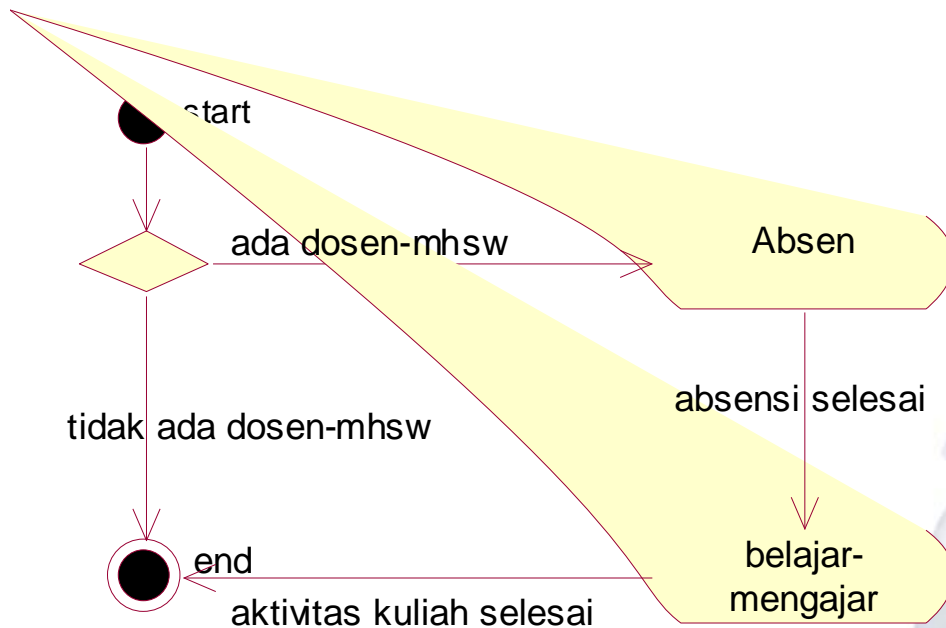


Use Case



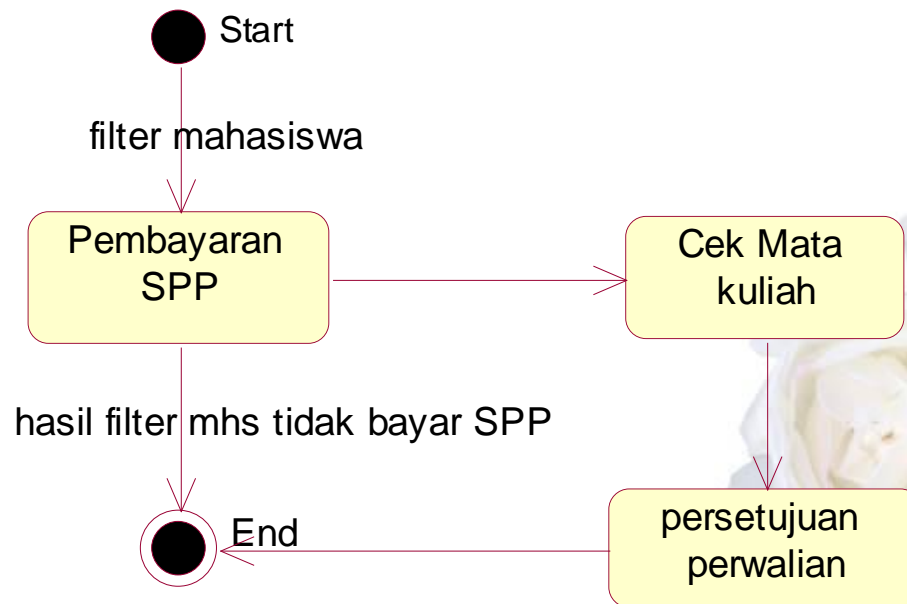
Activity diagram

- Menggambarkan proses-proses yang terjadi mulai aktivitas dimulai sampai aktivitas berhenti
- Activity diagram ini mirip dengan flowchart diagram



Statechart Diagram

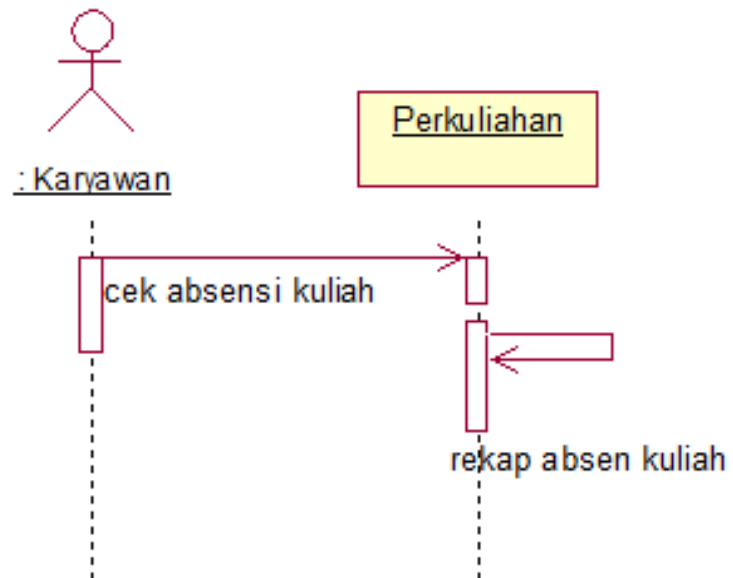
- Merupakan diagram yang menggambarkan perubahan state dari state yang satu ke state lainnya
- Terdiri dari status, transisi, kejadian, dan aktivitas



Sequence Diagram

- Merupakan interaksi antara objek-objek dalam suatu sistem dan terjadi komunikasi yang berupa pesan serta parameter waktu
- Biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang men-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan





Collaboration Diagram

Collaboration Diagram hampir sama dengan sequence diagram tetapi berbeda pada objek yang di titik tekankan , collaboration lebih menekankan pada pemunculan objek itu sendiri sedangkan sequence diagram lebih pada penyampaian message dengan parameter waktu

