

KONSEP PERANCANGAN TERSTRUKTUR

- Pendekatan perancangan terstruktur dimulai dari awal 1970.
- Pendekatan terstruktur dilengkapi dengan alat-alat (tools) dan teknik-teknik (techniques) yang dibutuhkan dalam pengembangan sistem, sehingga hasil akhir dari sistem yang dikembangkan akan diperoleh sistem yang strukturnya didefinisikan dengan baik dan jelas.
- Melalui pendekatan terstruktur, permasalahan yang komplek di organisasi dapat dipecahkan dan hasil dari sistem akan mudah untuk dipelihara, fleksibel, lebih memuaskan pemakainya, mempunyai dokumentasi yang baik, tepat waktu, sesuai dengan anggaran biaya pengembangan, dapat meningkatkan produktivitas dan kualitasnya akan lebih baik (bebas kesalahan)

Data Flow Diagram (DFD)

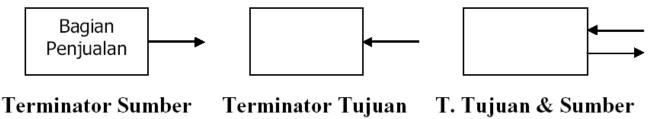
- Adalah alat pembuatan model yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi.
- Sering disebut juga dengan nama Bubble chart, Bubble diagram, model proses, diagram alur kerja, atau model fungsi.
- Merupakan alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisa maupun rancangan sistem yang mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program.

KOMPONEN DATA FLOW DIAGRAM

Gane/Sarson	Yourdon/De Marco	Keterangan
Entitas Eksternal	Entitas Eksternal	Entitas eksternal, dapat berupa orang/unit terkait yang berinteraksi dengan sistem tetapi diluar sistem
Proses	Proses	Orang, unit yang mempergunakan atau melakukan transformasi data. Komponen fisik tidak diidentifikasikan.
Aliran data	Aliran data	Aliran data dengan arah khusus dari sumber ke tujuan
Data store	Data Store	Penyimpanan data atau tempat data direfer oleh proses.

Komponen Terminator / Entitas Luar

- Terminator mewakili entitas eksternal yang berkomunikasi dengan sistem yang sedang dikembangkan. Biasanya terminator dikenal dengan nama entitas luar (external entity).
- 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.

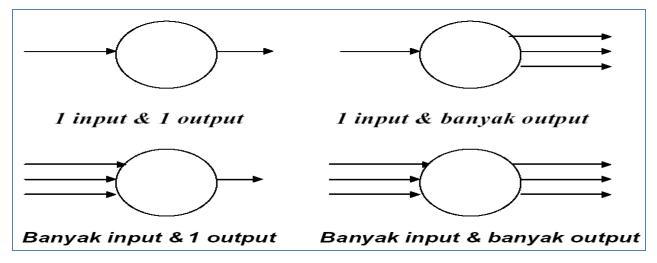


Komponen Terminator / Entitas Luar

- tiga hal penting yang harus diingat tentang terminator :
 - Terminator merupakan bagian/lingkungan luar sistem.
 Alur data yang menghubungkan terminator dengan berbagai proses sistem, menunjukkan hubungan sistem dengan dunia luar.
 - Profesional Sistem Tidak berhak mengubah isi atau cara kerja organisasi atau prosedur yang berkaitan dengan terminator
 - Hubungan yang ada antar terminator yang satu dengan yang lain tidak digambarkan pada DFD.

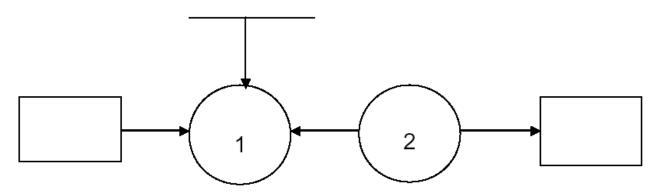
Komponen Proses

- Komponen proses menggambarkan bagian dari sistem yang mentransformasikan input menjadi output.
- Ada empat kemungkinan yang dapat terjadi dalam proses sehubungan dengan input dan output:



- Ada beberapa hal yang perlu diperhatikan tentang proses :
 - Proses harus memiliki input dan output.
 - Proses dapat dihubungkan dengan komponen terminator, data store atau proses melalui alur data.
 - Sistem/bagian/divisi/departemen yang sedang dianalisis oleh profesional sistem digambarkan dengan komponen proses.

Contoh kesalahan proses

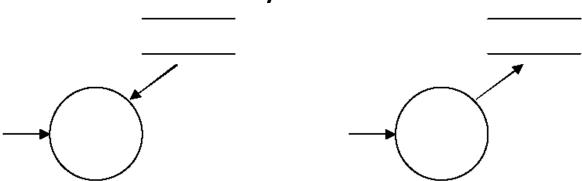


Umumnya kesalahan proses di DFD adalah:

- 1. Proses mempunyai input tetapi tidak menghasilkan output. Kesalahan ini disebut dengan *black hole* (lubang hitam), karena data masuk ke dalam proses dan lenyap tidak berbekas seperti dimasukkan ke dalam lubang hitam (*lihat proses 1*).
- 2. Proses menghasilkan output tetapi tidak pernah menerima input. Kesalahan ini disebut dengan *miracle* (ajaib), karena ajaib dihasilkan output tanpa pernah menerima input (*lihat proses 2*).

Komponen Data Store

- biasanya berkaitan dengan penyimpananpenyimpanan, seperti file atau database yang berkaitan dengan penyimpanan secara komputerisasi, misalnya file disket, file harddisk, file pita magnetik. Data store juga berkaitan dengan penyimpanan secara manual seperti buku alamat, file folder, dan agenda.
- Suatu data store dihubungkan dengan alur data hanya pada komponen proses, tidak dengan komponen DFD lainnya.



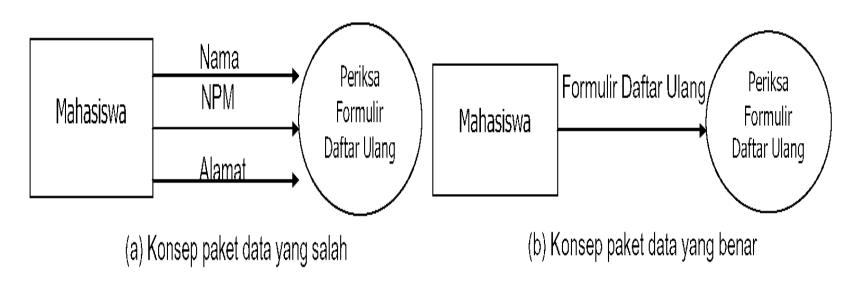
Komponen Data Flow / Alur Data

 Suatu data flow / alur data digambarkan dengan anak panah, yang menunjukkan arah menuju ke dan keluar dari suatu proses. Alur data ini digunakan untuk menerangkan perpindahan data atau paket data/informasi dari satu bagian sistem ke bagian lainnya.

Konsep yang perlu diperhatikan dalam alur data

Konsep Paket Data (Packets of Data)

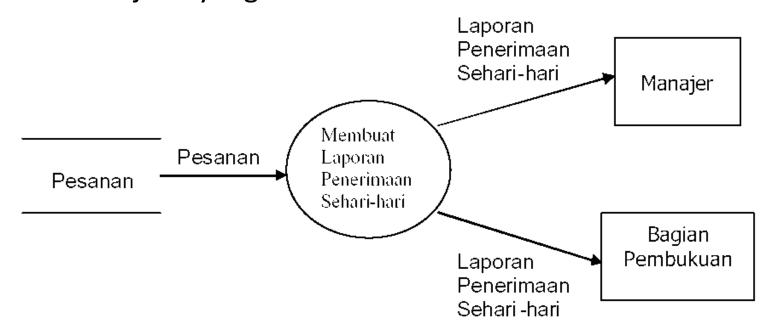
Apabila dua data atau lebih mengalir dari suatu sumber yang sama menuju ke tujuan yang sama dan mempunyai hubungan, dan harus dianggap sebagai satu alur data tunggal, karena data itu mengalir bersama-sama sebagai satu paket.



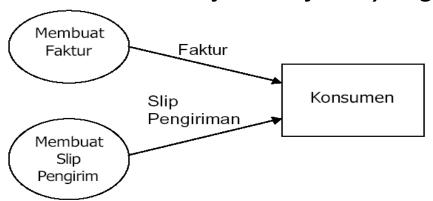
Konsep yang perlu diperhatikan dalam alur data

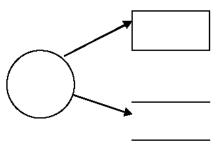
Konsep Alur Data Menyebar (Diverging Data Flow)

Alur data menyebar menunjukkan sejumlah tembusan paket data yang yang berasal dari *sumber yang sama* menuju ke *tujuan yang berbeda*, atau paket data yang kompleks dibagi menjadi beberapa elemen data yang dikirim ke tujuan yang berbeda, atau alur data ini membawa paket data yang memiliki nilai yang berbeda yang akan dikirim ke tujuan yang berbeda.

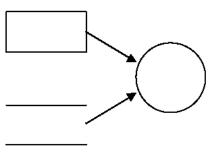


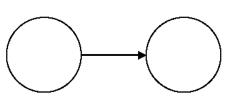
Konsep Alur Data Mengumpul (Converging Data Flow)
 Beberapa alur data yang berbeda sumber bergabung bersama-sama menuju ke tujuan yang sama.





- Konsep Sumber atau Tujuan Alur Data
 Semua alur data harus minimal mengandung satu proses. Maksud kalimat ini adalah :
 - Suatu alur data dihasilkan dari suatu proses dan menuju ke suatu data store dan/atau terminator
 - Satu alur data dihasilkan dari suatu data store dan/atau terminator dan menuju ke suatu proses
 - Suatu alur data dihasilkan dari suatu proses dan menuju ke suatu proses





Bentuk DFD

Diagram Alur Data Fisik (DADF)

DADF lebih tepat digunakan untuk menggambarkan sistem yang ada (sistem yang lama). Penekanan dari DADF adalah bagaimana proses-proses dari sistem diterapkan (dengan cara apa, oleh siapa dan dimana), termasuk proses-proses manual.

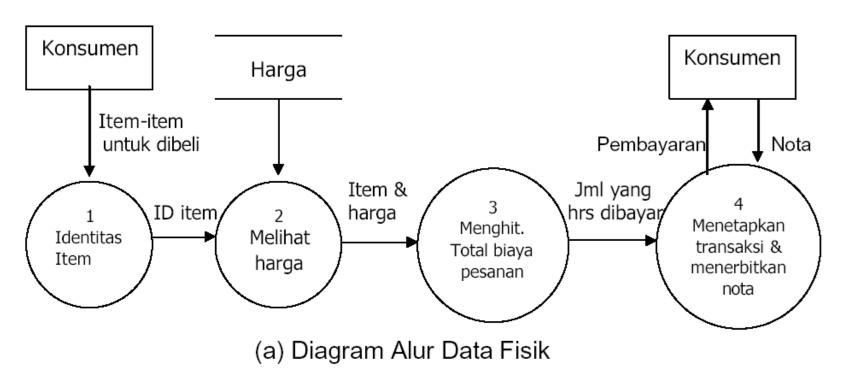
DADF harus memuat:

- 1. Proses-proses manual juga digambarkan.
- Nama dari alur data harus memuat keterangan yang cukup terinci untuk menunjukkan bagaimana pemakai sistem memahami kerja sistem.
- 3. Simpanan data dapat menunjukkan simpanan non komputer.
- 4. Nama dari simpanan data harus menunjukkan tipe penerapannya apakah secara manual atau komputerisasi. Secara manual misalnya dapat menunjukkan buku catatat, meja pekerja. Sedang cara komputerisasi misalnya menunjukkan file urut, file database.
- Proses harus menunjukkan nama dari pemroses, yaitu orang, departemen, sistem komputer, atau nama program komputer yang mengakses proses tersebut.

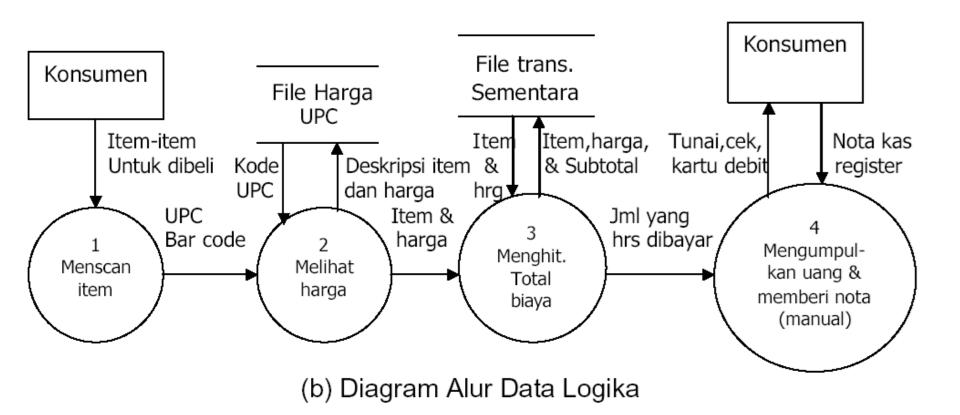
Bentuk DFD (Lanjutan...)

Diagram Alur Data Logika (DADL)

DADL lebih tepat digunakan untuk menggambarkan sistem komputerisasi, penggambaran DADL hanya menunjukkan kebutuhan proses dari sistem yang diusulkan secara logika, biasanya proses-proses yang digambarkan hanya merupakan proses-proses secara komputer saja.



Bentuk DFD (Lanjutan...)



Syarat Pembuatan DFD

- 1. Pemberian nama untuk tiap komponen DFD
- 2. Pemberian nomor pada komponen proses
- 3. Penggambaran DFD sesering mungkin agar enak dilihat
 - 1. Ukuran dan bentuk proses
 - 2. Alur data melingkar atau lurus
 - 3. DFD gambar tangan atau menggunakan mesin
- 4. Penghindaran penggambaran DFD yang rumit
- Pemastian DFD yang dibentuk itu konsiten secara logika

Langkah 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)
 - 1. Tentukan nama sistemnya.
 - 2. Tentukan batasan sistemnya.
 - 3. Tentukan terminator apa saja yang ada dalam sistem.
 - 4. Tentukan apa yang diterima/diberikan terminator dari/ke sistem.
 - 5. Gambarkan diagram konteks.

Langkah Penggambaran DFD

4. Buat Diagram Level Zero

Diagram ini adalah dekomposisi dari diagram konteks.

- 1. Tentukan proses utama yang ada pada sistem.
- Tentukan apa yang diberikan/diterima masing-masing proses ke/dari sistem sambil memperhatikan konsep keseimbangan (alur data yang keluar/masuk dari suatu level harus sama dengan alur data yang masuk/keluar pada level berikutnya).
- 3. Apabila diperlukan, munculkan data store (master) sebagai sumber maupun tujuan alur data.
- 4. Gambarkan diagram level zero.
 - Hindari perpotongan arus data
 - Beri nomor pada proses utama (nomor tidak menunjukkan urutan proses).

5. Buat Diagram Level Satu

Diagram ini merupakan dekomposisi dari diagram level zero.

- Tentukan proses yang lebih kecil (sub-proses) dari proses utama yang ada di level zero.
- Tentukan apa yang diberikan/diterima masing-masing sub-proses ke/dari sistem dan perhatikan konsep keseimbangan.
- Apabila diperlukan, munculkan data store (transaksi) sebagai sumber maupun tujuan alur data.
- Gambarkan DFD level Satu
 - Hindari perpotongan arus data.
 - Beri nomor pada masing-masing sub-proses yang menunjukkan dekomposisi dari proses sebelumnya.

Contoh: 1.1, 1.2, 2.1

6. 6. DFD Level Dua, Tiga, ...

Diagram ini merupakan dekomposisi dari level sebelumnya. Proses dekomposisi dilakukan sampai dengan proses siap dituangkan ke dalam program. Aturan yang digunakan sama dengan level satu.

Contoh DFD Level Nol

