DATA FLOW DIAGRAM

1. 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 akam 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)

2. DATA FLOW DIAGRAM (DFD)

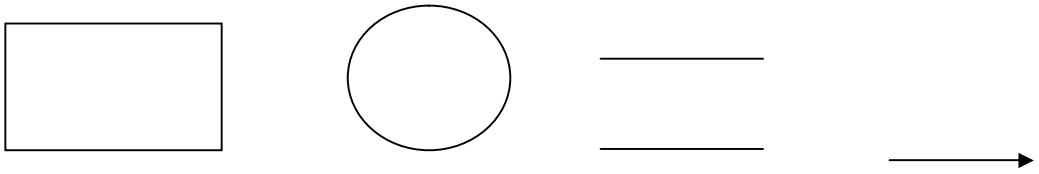
Data Flow Diagram (DFD) adalah alat pembuatan model yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi. DFD ini sering disebut juga dengan nama Bubble chart, Bubble diagram, model proses, diagram alur kerja, atau model fungsi.

DFD ini adalah salah satu alat pembuatan model yang sering digunakan, khususnya bila fungsi-fungsi sistem merupakan bagian yang lebih penting dan kompleks dari pada data yang dimanipulasi oleh sistem. Dengan kata lain, DFD adalah alat pembuatan model yang memberikan penekanan hanya pada fungsi sistem.

DFD ini 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.

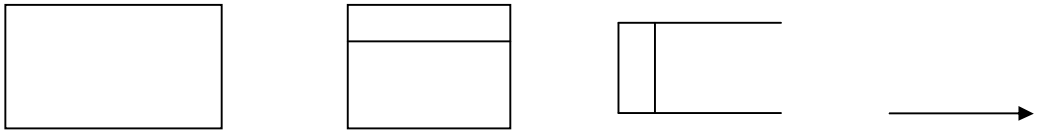
3. KOMPONEN DATA FLOW DIAGRAM

Menurut Yourdan dan DeMarco



Terminator **Proses** **Data Store** **Alur Data**

Menurut Gene dan Serson



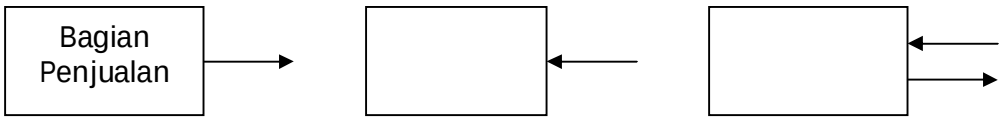
Terminator **Proses** **Data Store** **Alur Data**

3.1. 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.



Terminator Sumber **Terminator Tujuan** **T. Tujuan & Sumber**

Terminator dapat berupa orang, sekelompok orang, organisasi, departemen di dalam organisasi, atau perusahaan yang sama tetapi di luar kendali sistem yang sedang dibuat modelnya.

Terminator dapat juga berupa departemen, divisi atau sistem di luar sistem yang berkomunikasi dengan sistem yang sedang dikembangkan.

Komponen terminator ini perlu **diberi nama** sesuai dengan dunia luar yang berkomunikasi dengan sistem yang sedang dibuat modelnya, dan biasanya menggunakan **kata benda**, misalnya ***Bagian Penjualan, Dosen, Mahasiswa***.

Ada tiga hal penting yang harus diingat tentang terminator :

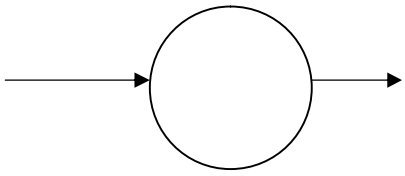
1. Terminator merupakan bagian/lingkungan luar sistem. Alur data yang menghubungkan terminator dengan berbagai proses sistem, menunjukkan hubungan sistem dengan dunia luar.
2. Profesional sistem tidak dapat mengubah isi atau cara kerja organisasi, atau prosedur yang berkaitan dengan terminator.
3. Hubungan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

3.2. Komponen Proses

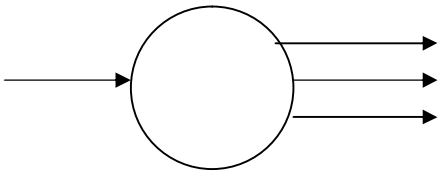
Komponen proses menggambarkan bagian dari sistem yang mentransformasikan input menjadi output.

Proses diberi nama untuk menjelaskan proses/kegiatan apa yang sedang/akan dilaksanakan. Pemberian nama proses dilakukan dengan menggunakan **kata kerja transitif** (kata kerja yang membutuhkan obyek), seperti ***Menghitung Gaji, Mencetak KRS, Menghitung Jumlah SKS***.

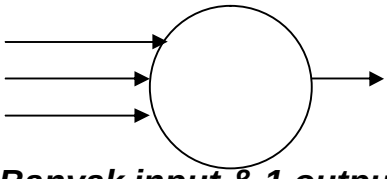
Ada empat kemungkinan yang dapat terjadi dalam proses sehubungan dengan input dan output :



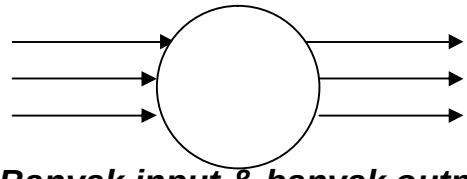
1 input & 1 output



1 input & banyak output



Banyak input & 1 output

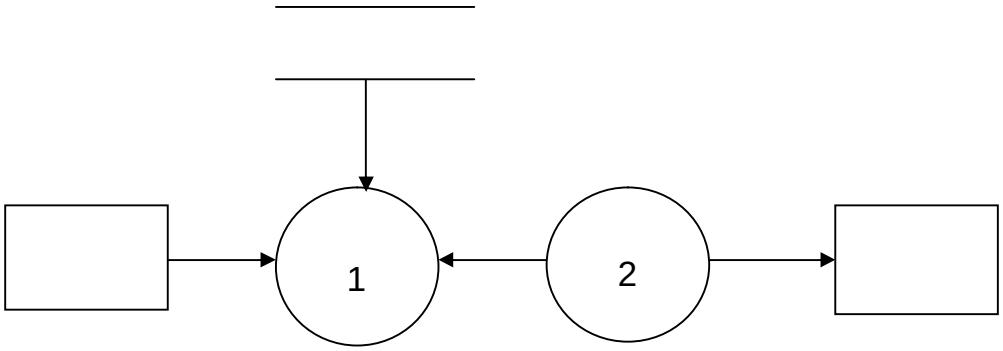


Banyak input & banyak output

Ada beberapa hal yang perlu diperhatikan tentang proses :

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

Berikut ini merupakan suatu contoh proses yang salah :



Gambar 1. Contoh 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*).

3.3. Komponen Data Store

Komponen ini digunakan untuk membuat model sekumpulan paket data dan **diberi nama** dengan **kata benda jamak**, misalnya ***Mahasiswa***.

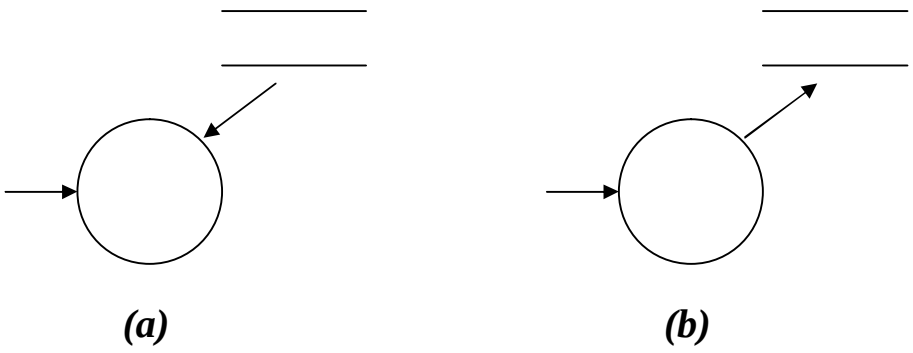
Data store ini biasanya berkaitan dengan penyimpanan-penyimpanan, seperti file atau database yang berkaitan dengan penyimpanan secara komputerisasi, misalnya file disket, file harddisk, file pita magnetik. Data store juga berkaitan dengan penyimpanan secara manual seperti buku alamat, file folder, dan agenda.

Suatu data store dihubungkan dengan alur data **hanya pada komponen proses**, tidak dengan komponen DFD lainnya. Alur data yang menghubungkan data store dengan suatu proses mempunyai pengertian sebagai berikut :

- **Alur data dari data store** yang berarti sebagai pembacaan atau pengaksesan satu paket tunggal data, lebih dari satu paket data, sebagian dari satu paket tunggal data, atau sebagian dari lebih dari satu paket data untuk suatu proses (*lihat gambar 2 (a)*).

- **Alur data ke data store** yang berarti sebagai pengupdatean data, seperti menambah satu paket data baru atau lebih, menghapus satu paket atau lebih, atau mengubah/memodifikasi satu paket data atau lebih (*lihat gambar 2 (b)*).

Pada pengertian pertama jelaslah bahwa data store tidak berubah, jika suatu paket data/informasi berpindah dari data store ke suatu proses. Sebaliknya pada pengertian kedua data store berubah sebagai hasil alur yang memasuki data store. Dengan kata lain, proses alur data bertanggung jawab terhadap perubahan yang terjadi pada data store.



Gambar 2. Implementasi data store

3.4. Komponen Data Flow / Alur Data

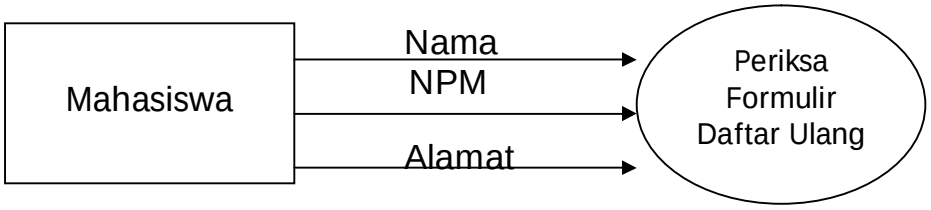
Suatu data flow / alur data digambarkan dengan anak panah, yang menunjukkan arah menuju ke dan keluar dari suatu proses. Alur data ini digunakan untuk menerangkan perpindahan data atau paket data/informasi dari satu bagian sistem ke bagian lainnya.

Selain menunjukkan arah, alur data pada model yang dibuat oleh profesional sistem dapat merepresentasikan bit, karakter, pesan, formulir, bilangan real, dan macam-macam informasi yang berkaitan dengan komputer. Alur data juga dapat merepresentasikan data/informasi yang tidak berkaitan dengan komputer.

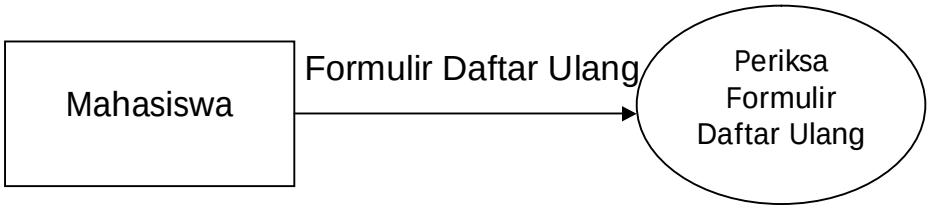
Alur data perlu **diberi nama** sesuai dengan data/informasi yang dimaksud, biasanya pemberian nama pada alur data dilakukan dengan menggunakan **kata benda**, contohnya ***Laporan Penjualan***.

Ada empat konsep yang perlu diperhatikan dalam penggambaran alur data, yaitu :

1. **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.



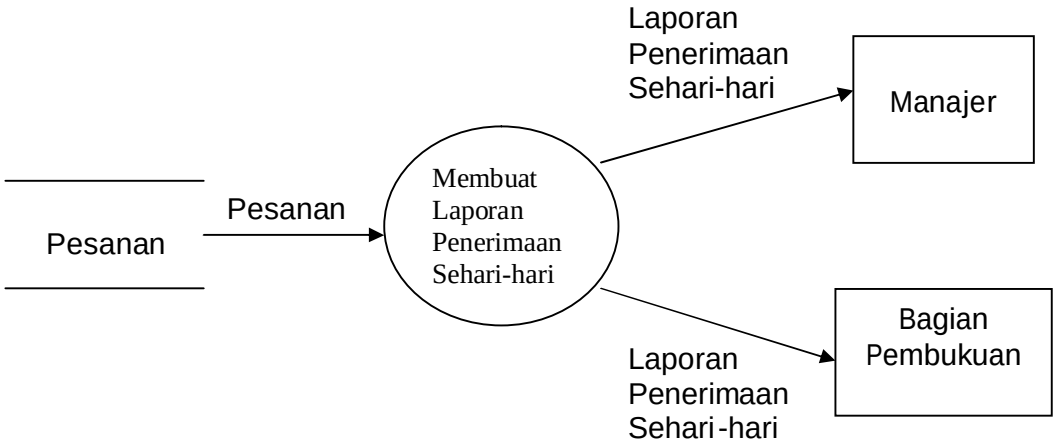
(a) Konsep paket data yang salah



(b) Konsep paket data yang benar

Gambar 3. Konsep paket data

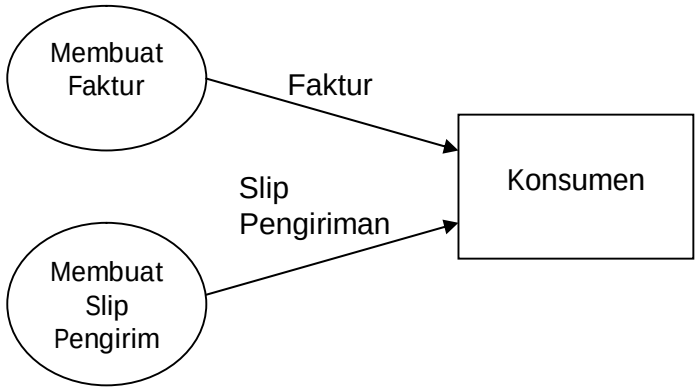
2. **Konsep Alur Data Menyebar (*Diverging Data Flow*)**
   Alur data menyebar menunjukkan sejumlah tembusan paket data yang yang berasal dari *sumber yang sama* menuju ke *tujuan yang berbeda*, atau paket data yang kompleks dibagi menjadi beberapa elemen data yang dikirim ke tujuan yang berbeda, atau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



Gambar 4. Konsep alur data menyebar

3. Konsep Alur Data Mengumpul (*Converging Data Flow*)

Beberapa alur data yang *berbeda sumber* bergabung bersama-sama menuju ke *tujuan yang sama*.

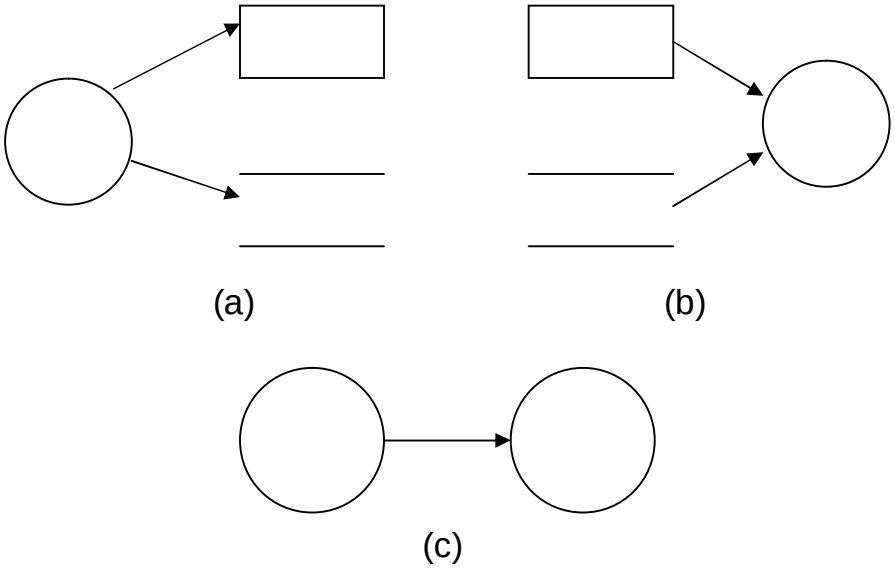


Gambar 5. Konsep alur data mengumpul

4. 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* (*lihat gambar 6 (a)*).
- Sutu alur data dihasilkan dari suatu *data store* dan/atau *terminator* dan menuju ke suatu *proses* (*lihat gambar 6 (b)*).
- Suatu alur data dihasilkan dari suatu *proses* dan menuju ke suatu *proses* (*lihat gambar 6 (c)*).



Gambar 6. Konsep sumber atau tujuan alur data

4. BENTUK DATA FLOW DIAGRAM

Terdapat dua bentuk DFD, yaitu **Diagram Alur Data Fisik**, dan **Diagram Alur data Logika**. Diagram alur data fisik lebih menekankan pada bagaimana proses dari sistem diterapkan, sedangkan diagram alur data logika lebih menekankan proses-proses apa yang terdapat di sistem.

4.1. 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.

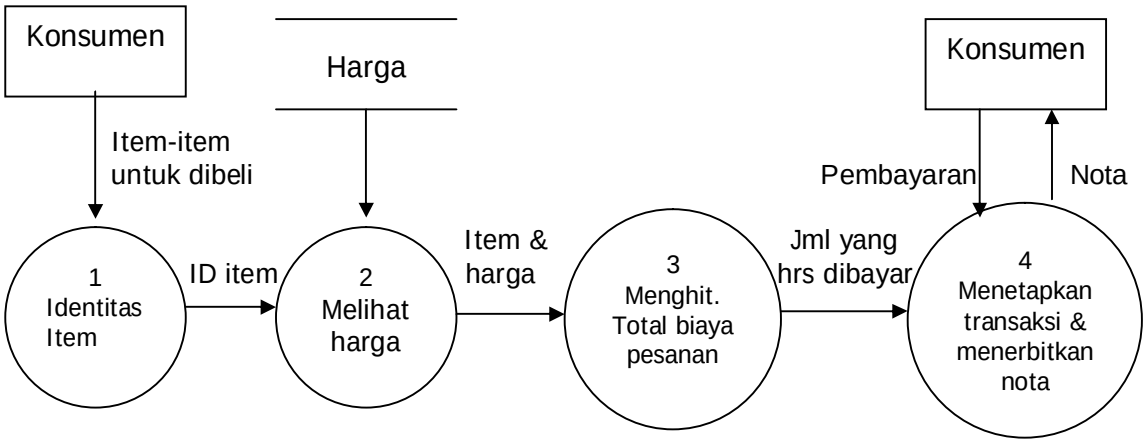
Untuk memperoleh gambaran bagaimana sistem yang ada diterapkan, DADF harus memuat :

1. Proses-proses manual juga digambarkan.
2. 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.

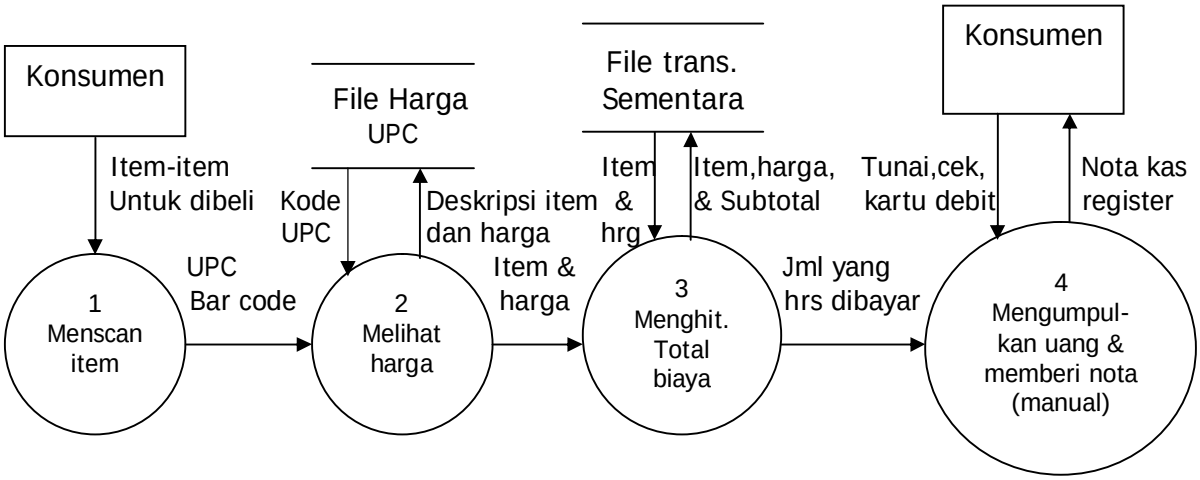
5. Proses harus menunjukkan nama dari pemroses, yaitu orang, departemen, sistem komputer, atau nama program komputer yang mengakses proses tersebut.

4.2. Diagram Alur Data Logika (DADL)

DADL lebih tepat digunakan untuk menggambarkan sistem yang akan diusulkan (sistem yang baru). Untuk 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.



(a) Diagram Alur Data Logika



(b) Diagram Alur Data Fisik

Gambar 7. DADF dan DADL

5. SYARAT-SYARAT PEMBUATAN DATA FLOW DIAGRAM

Syarat pembuatan DFD ini akan menolong profesional sistem untuk menghindari pembentukkan DFD yang salah atau DFD yang tidak lengkap atau tidak konsisten secara logika. Beberapa syarat pembutan DFD dapat menolong profesional sistem untuk membentuk DFD yang benar, menyenangkan untuk dilihat dan mudah dibaca oleh pemakai.

Syarat-syarat pembuatan DFD ini adalah :
1. Pemberian nama untuk tiap komponen DFD
2. Pemberian nomor pada komponen proses
3. Penggambaran DFD sesering mungkin agar enak dilihat
4. Penghindaran penggambaran DFD yang rumit
5. Pemastian DFD yang dibentuk itu konsiten secara logika

5.1. Pemberian Nama untuk Tiap komponen DFD

Seperti yang telah dijelaskan sebelumnya, komponen terminator mewakili lingkungan luar dari sistem, tetapi mempunyai pengaruh terhadap sistem yang sedang dikembangkan ini. Maka agar pemakai mengetahui dengan lingkungan mana saja sistem mereka berhubungan, komponen terminator ini harus diberi nama sesuai dengan lingkungan luar yang mempengaruhi sistem ini. Biasanya komponen terminator diberi nama dengan kata benda.
Selanjutnya adalah komponen proses. Komponen proses ini mewakili fungsi sistem yang akan dilaksanakan atau menunjukkan bagaimana fungsi sistem dilaksanakan oleh seseorang, sekelompok orang atau mesin. Maka sangatlah jelas bahwa komponen ini perlu diberi nama yang tepat, agar siapa yang membaca DFD khususnya pemakai akan merasa yakin bahwa DFD yang dibentuk ini adalah model yang akurat.

Pemberian nama pada komponen proses lebih baik menunjukkan aturan-aturan yang akan dilaksanakan oleh seseorang dibandingkan dengan memberikan nama atau identitas orang yang akan melaksanakannya. Ada dua alasan mengapa bukan nama atau identitas orang (yang melaksanakan fungsi sistem) yang digunakan sebagai nama proses, yaitu :
1. Orang tersebut mungkin diganti oleh orang lain saat mendatang, sehingga bila tiap kali ada pergantian orang yang melaksanakan fungsi tersebut, maka sistem yang dibentuk harus diubah lagi.

2. Orang tersebut mungkin tidak melaksanakan satu fungsi sistem saja, melainkan beberapa fungsi sistem yang berbeda. Daripada menggambarkan beberapa proses dengan nama yang sama tetapi artinya berbeda, lebih baik tunjukkan dengan tugas/fungsi sistem yang sebenarnya akan dilaksanakan.

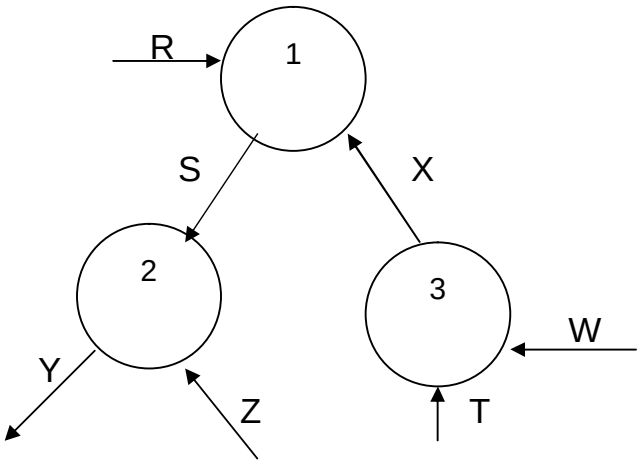
Karena nama untuk komponen proses lebih baik menunjukkan tugas/fungsi sistem yang akan dilaksanakan, maka lebih baik pemberian nama ini menggunakan kata kerja transitif.

Pemberian nama untuk komponen data store menggunakan kata benda, karena data store menunjukkan data apa yang disimpan untuk kebutuhan sistem dalam melaksanakan tugasnya. Jika sistem sewaktu-waktu membutuhkan data tersebut untuk melaksanakan tugasnya, maka data tersebut tetap ada, karena sistem menyimpannya.

Begitu pula untuk komponen alur data, namanya lebih baik diberikan dengan menggunakan kata benda. Karena alur data ini menunjukkan data dan infiormasi yang dibutuhkan dan yang dikeluarkan oleh sistem dalam pelaksanaan tugasnya.

5.2. Pemberian Nomor pada Komponen Proses

Biasanya profesional sistem memberikan nomor dengan bilangan terurut pada komponen proses sebagai referensi. Tidak jadi masalah bagaimana nomor-nomor proses ini diberikan. Nomor proses dapat diberikan dari kiri ke kanan, atau dari atas ke bawah, atau dapat pula dilakukan dengan pola-pola tertentu selama pemberian nomor ini tetap konsisten pada nomor yang dipergunakan.



Gambar 8. Contoh Pemberian nomor pada proses

Nomor-nomor proses yang diberikan terhadap komponen proses ini tidak dimaksudkan bahwa proses tersebut dilaksanakan secara berurutan. Pemberian nomor ini dimaksudkan agar pembacaan suatu proses dalam suatu diskusi akan lebih mudah dengan hanya menyebutkan prosesnya saja jika dibandingkan dengan menyebutkan nama prosesnya, khususnya jika nama prosesnya panjang dan sulit.

Maksud pemberian nomor pada proses yang lebih penting lagi adalah untuk menunjukkan referensi terhadap skema penomoran secara hirarki pada levelisasi DFD. Dengan kata lain, nomor proses ini merupakan dasar pemberian nomor pada levelilasi DFD (*lihat gambar 11*).

5.3. Penggambaran DFD sesering mungkin

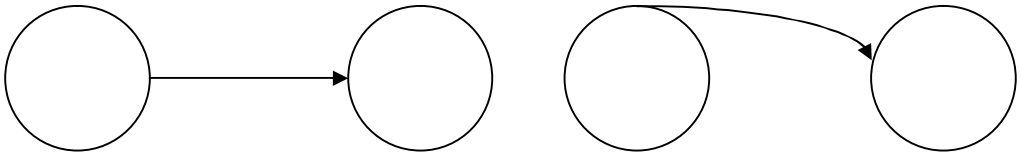
Penggambaran DFD dapat dilakukan berkali-kali sampai secara teknik DFD itu benar, dapat diterima oleh pemakai, dan sudah cukup rapih sehingga profesional sistem tidak merasa malu untuk menunjukkan DFD itu kepada atasannya dan pemakai.

Dengan kata lain, penggambaran DFD ini dilakukan sampai terbentuk DFD yang enak dilihat, dan mudah dibaca oleh pemakai dan profesional sistem lainnya. Keindahan penggambaran DFD tergantung pada standar-standar yang diminta oleh organisasi tempat profesional sistem itu bekerja dan perangkat lunak yang dipakai oleh profesional sistem dalam membuat DFD.

Penggambaran yang enak untuk dilihat dapat dilakukan dengan memperhatikan hal-hal berikut ini :

- ***Ukuran dan bentuk proses***.
  Beberapa pemakai kadang-kadang merasa bingung bila ukuran proses satu berbeda dengan proses yang lain. Mereka akan mengira bahwa proses dengan ukuran yang lebih besar akan diduga lebih penting dari proses yang lebih kecil. Hal ini sebenarnya hanya karena nama proses itu lebih panjang dibandingkan dengan proses yang lain. Jadi, sebaiknya proses yang digambarkan memiliki ukuran dan bentuk yang sama.

- ***Alur data melingkar dan alur data lurus.***
  Alur data dapat digambarkan dengan melingkar atau hanya garis lurus. Mana yang lebih enak dipandang tergantung siapa yang akan melihat DFD tersebut.



(a). Alur data dengan garis lurus (b). Alur data dengan melingkar

Gambar 9

Ø ***DFD dengan gambar tangan dan gambar menggunakan mesin*.**

DFD dapat digambarkan secara manual atau dengan menggunakan bantuan mesin, tergantung pilihan pemakai atau profesional sistem.

5.4. Penghindaran Penggambaran DFD yang rumit

Tujuan DFD adalah untuk membuat model fungsi yang harus dilaksanakan oleh suatu sistem dan interaksi antar fungsi. Tujuan lainnya adalah agar model yang dibuat itu mudah dibaca dan dimengerti tidak hanya oleh profesional sistem yang membuat DFD, tetapi juga oleh pemakai yang berpengalaman dengan subyek yang terjadi. Hal ini berarti DFD harus mudah dimengerti, dibaca, dan menyenangkan untuk dilihat.

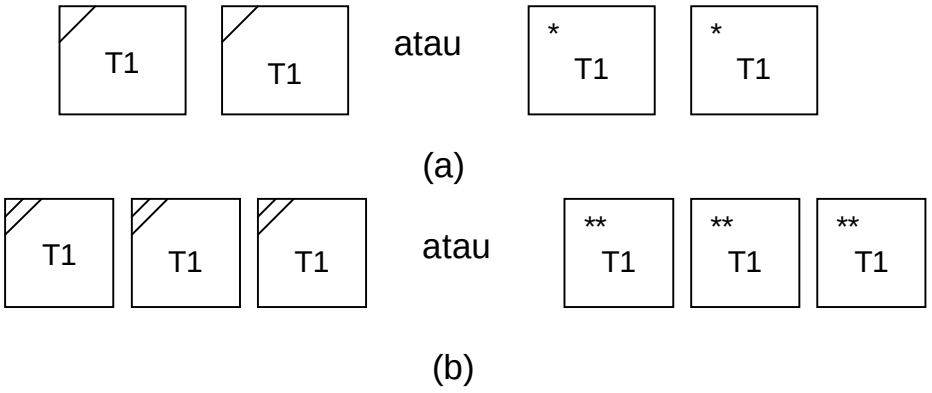
Pada banyak masalah, DFD yang dibuat tidak memiliki terlalu banyak proses (maksimal enam proses) dengan data store, alur data, dan terminator yang berkaitan dengan proses tersebut dalam satu diagram.

Bila terlalu banyak proses, terminator, data store, dan alur data digambarkan dalam satu DFD, maka ada kemungkinan terjadi banyak persilangan alur data dalam DFD tersebut. Persilangan alur data ini menyebabkan pemakai akan sulit membaca dan mengerti DFD yang terbentu. Jadi semakin sedikit adanya persilangan data pada DFD, maka makin baik DFD yang dibentuk oleh profesional sistem.

Persilangan alur data ini dapat dihindari dengan menggambarkan DFD secara bertingkat-tingkat (levelisasi DFD), atau dengan menggunakan pemakaian duplikat terhadap komponen DFD.

Komponen DFD yang **dapat menggunakan duplikat** hanya **komponen store** dan **terminator**. Pemberian duplikat ini juga tidak dapat diberikan sesuka profesional sistem yang membuat DFD, tetapi makin sedikit pemakaian duplikat, makin baik DFD yang terbentuk.

Pemberian duplikat terhadap data store dilakukan dengan memberikan simbol garis lurus (ξ) atau asterik (*), sedangkan untuk terminator menggunakan simbol garis miring (/) atau asterik (*). Banyaknya pemberian simbol duplikat pada duplikat yang digunakan tergantung banyaknya duplikat yang digunakan.



Gambar 10. Contoh pemakaian simbol duplikat
pada komponen terminator
(a) Satu duplikat yang digunakan
(b) Dua duplikat yang digunakan

5.5. Penggambaran DFD yang Konsisten

Penggambaran DFD harus konsisten terhadap kelompok DFD lainnya. Profesional sistem menggambarkan DFD berdasarkan tingkatan DFD dengan tujuan agar DFD yang dibuatnya itu mudah dibaca dan dimengerti oleh pemakai sistem. Hal ini sesuai dengan salah satu tujuan atau syarat membuat DFD.

6. PENGGAMBARAN DFD

Tidak ada aturan baku untuk menggambarkan DFD. Tapi dari berbagai referensi yang ada, secara garis besar langkah untuk membuat DFD adalah :

1. Identifikasi terlebih dahulu semua entitas luar yang terlibat di sistem.

2. Identifikasi semua input dan output yang terlibat dengan entitas luar.

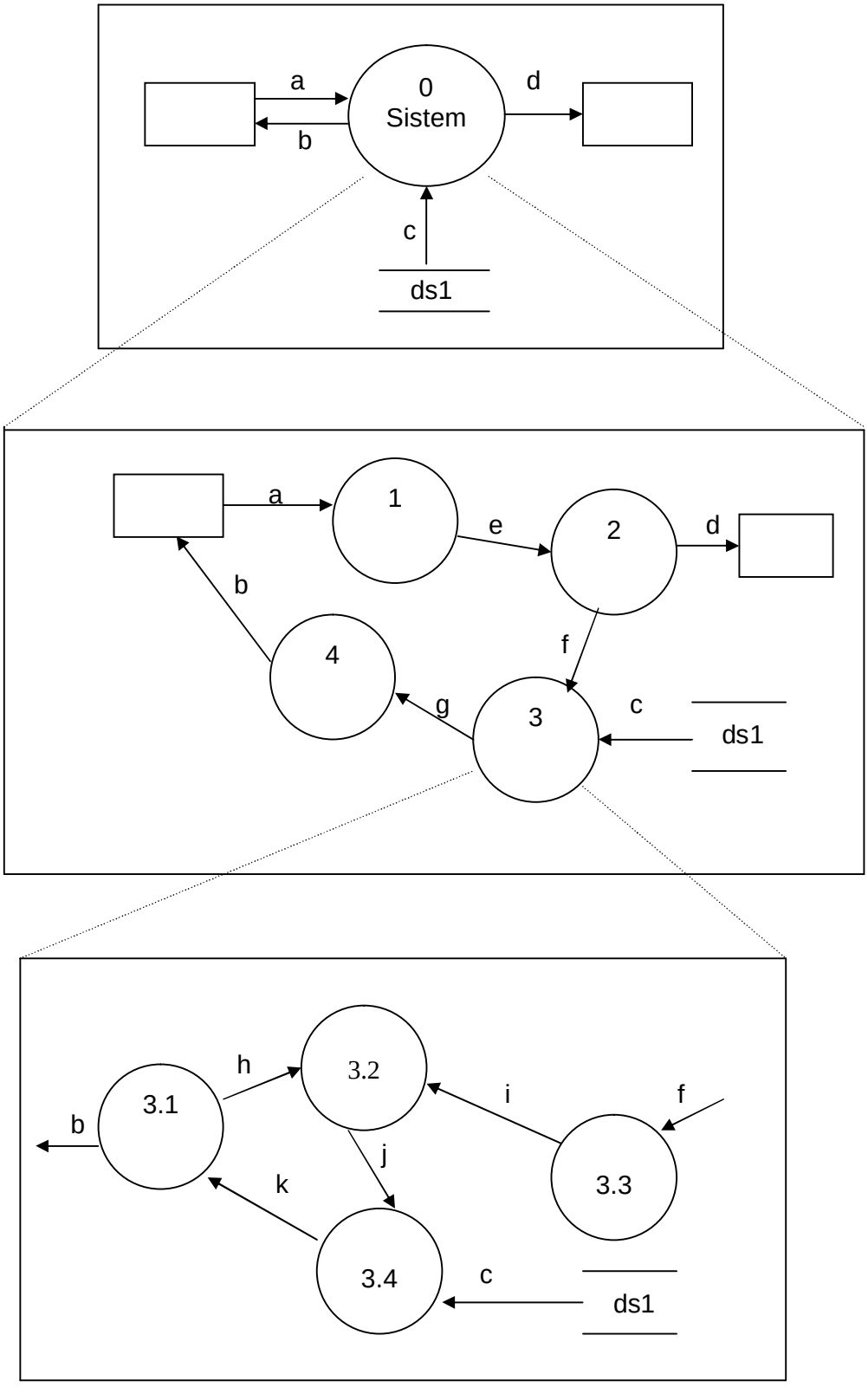
3. ***Buat Diagram Konteks (diagram context)***
   Diagram ini adalah diagram level tertinggi dari DFD yang menggambarkan hubungan sistem dengan lingkungan luarnya.
   Caranya :
   - Tentukan nama sistemnya.
   - Tentukan batasan sistemnya.
   - Tentukan terminator apa saja yang ada dalam sistem.
   - Tentukan apa yang diterima/diberikan terminator dari/ke sistem.
   - Gambarkan diagram konteks.

4. ***Buat Diagram Level Zero***
   Diagram ini adalah dekomposisi dari diagram konteks.
   Caranya :
   - 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).
   - Apabila diperlukan, munculkan data store (master) sebagai sumber maupun tujuan alur data.
   - 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.
   Caranya :
   - 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. 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.



Gambar 11. Levelisasi DFD