

MODUL 4

PENDEKATAN PERANCANGAN TERSTRUKTUR DAN DATA FLOW DIAGRAM

MATERI

1. Konsep Perancangan Terstruktur
2. Data Flow Diagram (DFD)
 - 2.1 Komponen DFD
 - 2.2 Bentuk DFD
 - 2.3 Syarat Pembuatan

TUJUAN INSTRUKSI UMUM

Memahami Konsep Pendekatan Perancangan Terstruktur dengan Salah Satu Alat Bantunya DFD, serta Dapat Menggunakan DFD Secara Tepat

TUJUAN INSTRUKSI KHUSUS

1. Mahasiswa mengetahui konsep DFD serta bagaimana cara menggunakannya
2. Mahasiswa dapat menentukan kapan menggunakan secara tepat berdasarkan kelebihan dan kekurangan DFD
3. Mahasiswa diharapkan dapat membuat model sistem yang akan mereka kembangkan dengan DFD

1. Konsep Perancangan Terstruktur

Karena banyak terjadi permasalahan-permasalahan di pendekatan klasik, maka kebutuhan akan pendekatan pengembangan sistem yang lebih baik mulai terasa dibutuhkan. Sayangnya sampai sekarang masih banyak orang yang tidak menyadari bahwa hanya dengan mengikuti tahapan di *life cycle* saja tidak akan membuat pengembangan sistem informasi menjadi berhasil. Oleh karena itu diperlukan suatu pendekatan pengembangan sistem yang baru yang dilengkapi dengan beberapa alat dan teknik supaya membuatnya berhasil.

Pendekatan ini yang dimulai dari awal tahun 1970 disebut dengan pendekatan terstruktur (*structured approach*). Pendekatan terstruktur dilengkapi dengan alat-alat (*tools*) dan teknik-teknik (*techniques*) yang dibutuhkan dalam pengembangan sistem, sehingga hasil akhir dari sistem yang dikembangkan akan didapatkan sistem yang strukturnya didefinisikan dengan baik dan jelas.

Konsep pengembangan sistem terstruktur bukan merupakan konsep yang baru. Teknik perakitan di pabrik-pabrik dan perancangan sirkuit untuk alat-alat elektronik adalah dua contoh dari konsep ini yang banyak digunakan di industri-industri. Konsep ini memang relatif masih baru digunakan dalam mengembangkan sistem informasi untuk dihasilkan produk sistem yang memuaskan pemakainya. Melalui pendekatan terstruktur, permasalahan-permasalahan yang kompleks di organisasi dapat dipecahkan dan hasil dari sistem akan mudah untuk dipelihara, fleksibel, lebih memuaskan pemakainya, mempunyai dokumentasi yang baik, tepat pada waktunya, sesuai dengan anggaran biaya pengembangannya, dapat meningkatkan produktivitas dan kualitasnya akan lebih baik (bebas kesalahan).

Salah satu *tools* dan teknik dalam pengembangan sistem terstruktur adalah menggunakan DFD (*Data Flow Diagram* = Diagram Arus Data, DAD).

2. Data Flow Diagram (DFD)

Ide dari suatu bagan untuk mewakili arus data dalam suatu sistem bukanlah hal yang baru. Pada tahun 1967, *Martin* dan *Estrin* memperkenalkan suatu algoritma program dengan menggunakan simbol lingkaran dan panah untuk mewakili arus data. *E. Yourdan* dan *L. L. Constantine* juga menggunakan notasi simbol ini untuk menggambarkan arus data dalam perancangan program. *G.E. Whitehouse* tahun 1973 juga menggunakan notasi semacam ini untuk membuat model-model sistem matematika. Penggunaan notasi dalam diagram arus data ini sangat membantu sekali untuk memahami suatu sistem pada semua tingkat kompleksitasnya seperti yang diungkapkan oleh *Chris Gane* dan *Trish Sarson*. Pada tahap analisis, penggunaan notasi ini sangat membantu sekali di dalam komunikasi dengan pemakai sistem untuk memahami sistem secara logika. Diagram yang menggunakan notasi-notasi ini untuk menggambarkan arus dari data sistem sekarang dikenal dengan nama diagram arus data (*data flow diagram, DFD*).

DFD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir (misalnya lewat telpon, surat dan sebagainya) atau lingkungan fisik dimana data tersebut akan disimpan (misalnya *file* kartu, *microfile*, *harddisk*, *tape*, *diskette* dan lain sebagainya). DFD merupakan alat yang cukup populer sekarang ini, karena dapat menggambarkan arus data di dalam sistem dengan terstruktur dan jelas. Lebih lanjut DFD juga merupakan dokumentasi dari sistem yang baik.

Komponen DFD

Beberapa komponen atau simbol yang digunakan DFD untuk maksu mewakili :

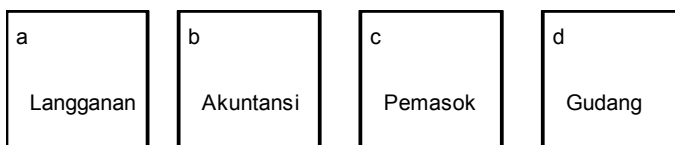
1. *external entity* (kesatuan luar) atau *boundary* (batas sistem)
2. *data flow* (arus data)
3. *process* (proses)
4. *data store* (simpanan data)

KESATUAN LUAR

Setiap sistem pasti mempunyai batas sistem (*boundary*) yang memisahkan suatu sistem dengan lingkungan luarnya. Sistem akan menerima *input* dan menghasilkan *output* kepada lingkungan luarnya. Kesatuan luar (*external entity*) merupakan kesatuan (*entity*) di lingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lainnya yang berada di lingkungan luarnya yang akan memberikan *input* atau menerima *output* dari sistem. Kesatuan luar ini kebanyakan adalah salah satu dari berikut ini :

- a. Suatu kantor, departemen, atau divisi dalam perusahaan teatpi di luar sistem yang sedang dikembangkan
- b. Orang atau sekelompok orang di organisasis tetapi di luar sistem yang sedang dikembangkan
- c. Suatu organisasi atau orang uang berada di luar organisasi seperti misalnya langganan, pemasok
- d. Sistem infromasi yang lain di luar sistem yang sedang dikembangkan
- e. Sumber asli dari suatu transaksi
- f. Penerima akhir dari suatu laporan yang dihasilkan oleh sistem

Suatu kesatuan luar dapat disimbolkan dengan suatu notasi kotak atau suatu kotak dengan sisi kiri dan atasnya berbentuk garis tebal, juga dapat diberi identifikasi dengan huruf kecil di ujung kiri atas sehingga berbentu sebagai berikut :

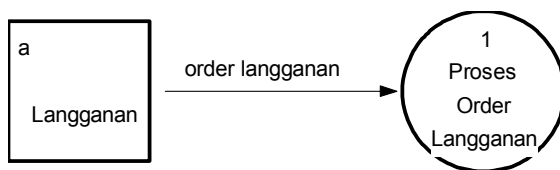


ARUS DATA

Arus data (*data flow*) di DFD diberi simbol suatu panah. Arus data ini mengalir diantara proses (*process*), simpanan data (*data store*) dan kesatuan luar (*external entity*). Arus data ini menunjukkan arus dari data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem dan dapat berbentuk sebagai berikut ini :

- a. Formulir atau dokumen yang digunakan di perusahaan
- b. Laporan tercetak yang dihasilkan oleh sistem
- c. Tampilan atau output di layar komputer yang dihasilkan oleh sistem
- d. Masukan untuk komputer
- e. Komunikasi ucapan
- f. Surat-surat atau memo
- g. Data yang dibaca atau direkamkan ke suatu file
- h. Suatu isian yang dicatat pada buku agenda
- i. Transmisi data dari suatu komputer ke komputer yang lain

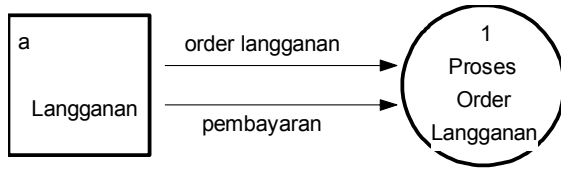
Arus data sebaiknya diberi nama yang jelas dan mempunyai arti. Nama dari arus data dituliskan disamping garis panahnya.



Gambar. arus data yang mengalir dari kesatuan luar *Langganan* ke *Proses Order Langganan* adalah dengan nama *order langganan*

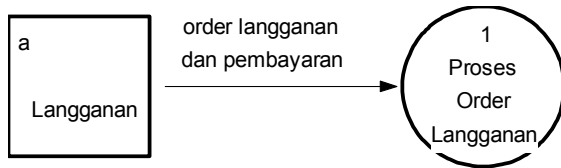
Di dalam menggambar arus data di DFD perlu diperhatikan beberapa konsep penggambarannya sebagai berikut :

1. Konsep paket dari data (*packet of data*)
 Bila dua atau lebih data mengalir dari suatu sumber yang sama ke tujuan yang sama, maka harus dianggap sebagai suatu arus data yang tunggal. Mengapa ? karena dua atau lebih data tersebut mengalir bersama-sama sebagai suatu paket. Data yang mengalir bersama-sama harus ditunjukkan sebagai satu arus data, walaupun misalnya terdiri dari beberapa dokumen. Contoh penggambaran arus data yang tidak benar



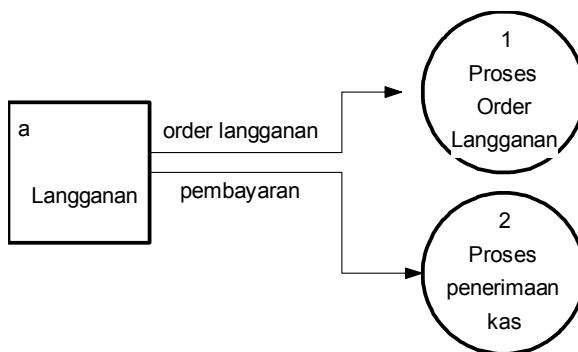
Gambar. arus data yang salah, karena dua buah data yang mengalir bersama harus ditunjukkan sebagai arus data yang tunggal

Dua buah arus data ini, yaitu *order langganan* dan *pembayaran* harus ditunjukkan sebagai arus data yang tunggal, yaitu sebagai arus data *order langganan dan pembayaran* sebagai berikut ini



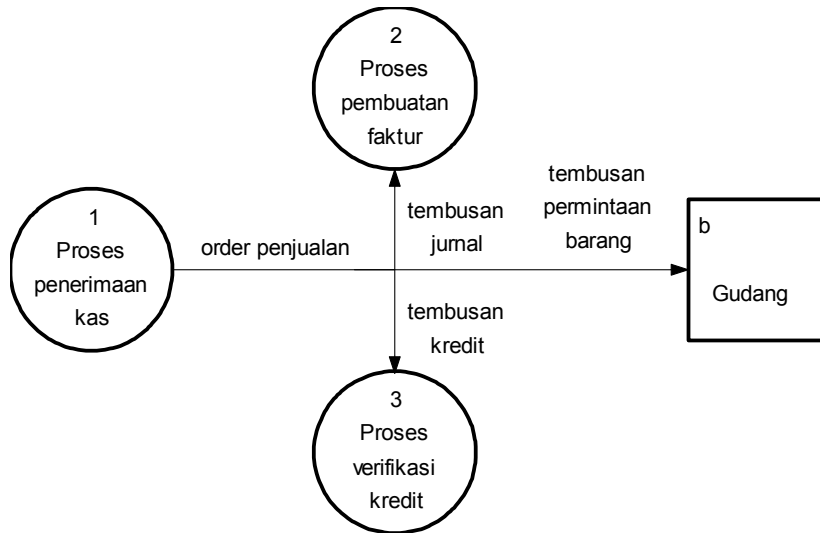
Gambar. arus data yang benar, karena dua buah data yang mengalir bersama ditunjukkan sebagai arus data yang tunggal.

Bila dua buah data ini akan ditangani oleh dua proses yang berlainan, berarti mempunyai tujuan yang berbeda, walaupun sumbernya sama, maka dapat digambarkan sebagai berikut ini :



2. Konsep arus data menyebar (*diverging data flow*)

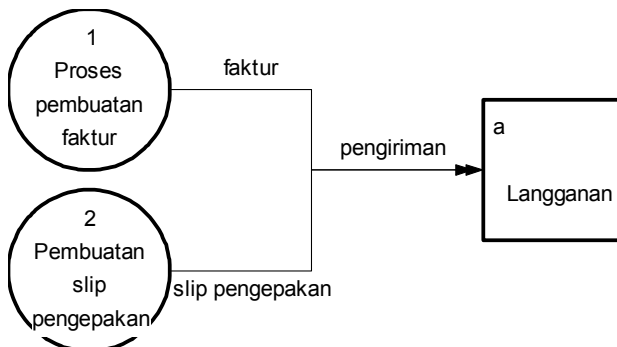
Arus data yang menyebar menunjukkan sejumlah tembusan dari arus data yang sama dari sumber yang sama ke tujuan yang berbeda.



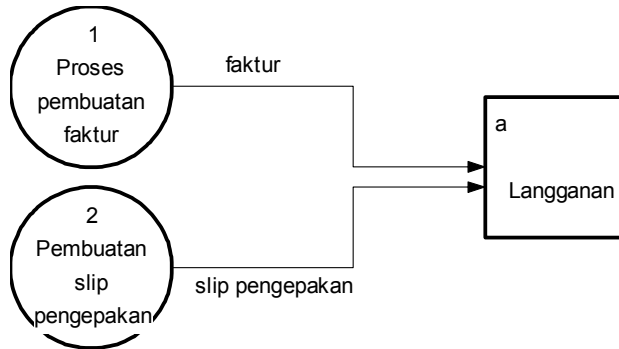
Pada gambar terlihat bahwa arus data *order penjualan* mempunyai sebanyak 3 tembusan, yaitu *tembusan untuk jurnal* yang mengalir ke proses pembuatan faktur, *tembusan permintaan barang* yang mengalir ke kesatuan luar gudang dan *tembusan kredit* yang mengalir ke proses verifikasi kredit. Konsep arus data yang menyebar ini menunjukkan bahwa arus data *tembusan jurnal*, *tembusan permintaan barang* dan *tembusan kredit* merupakan arus data yang mempunyai struktur elemen yang sama, karena merupakan hasil dari tembusan arus data *order penjualan*.

3. Konsep arus data mengumpul (*converging data flow*)

Arus data yang mengumpul menunjukkan beberapa arus data yang berbeda dari sumber yang berbeda bergabung bersama-sama menuju ke tujuan yang sama



Arus data *pengiriman* merupakan hasil dari gabungan arus data *faktur* dan *slip pengepakan*. Arus data mengumpul ini jarang dibuat di DFD dan sebagai penggantinya dapat digambarkan sebagai berikut ini

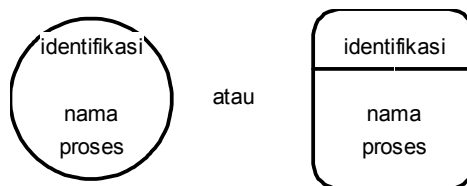


4. Konsep sumber dan tujuan arus data

Semua arus data harus dihasilkan dari suatu proses atau menuju ke suatu proses (*dapat salah satu atau kedua-duanya, yaitu berasal dari suatu proses menuju ke bukan suatu proses atau berasal dari bukan suatu proses tetapi menuju ke suatu proses atau berasal dari suatu proses dan menuju ke suatu proses*). Konsep ini penting karena arus data adalah salah satu dari hasil suatu proses atau akan digunakan untuk melakukan suatu proses.

PROSES

Suatu proses adalah kegiatan atau kerja yang dilakukan oleh orang, mesin atau komputer dari hasil suatu arus data yang masuk ke dalam proses untuk dihasilkan arus data yang akan keluar dari proses. Untuk *physical data flow diagram* (PDFD), proses yang dapat dilakukan oleh orang, mesin atau komputer, sedang untuk *logical data flow diagram* (LDFD), suatu proses hanya menunjukkan proses dari komputer. Suatu proses dapat ditunjukkan dengan simbol lingkaran atau dengan simbol empat persegi panjang tegak dengan sudut-sudutnya tumpul :



Gambar. Notasi proses di DFD

Setiap proses harus diberi penjelasan yang lengkap meliputi berikut ini :

1. Identifikasi proses

Identifikasi ini umumnya berupa suatu **angka** yang menunjukkan nomor acuan dari proses dan ditulis pada bagian atas di simbol proses.

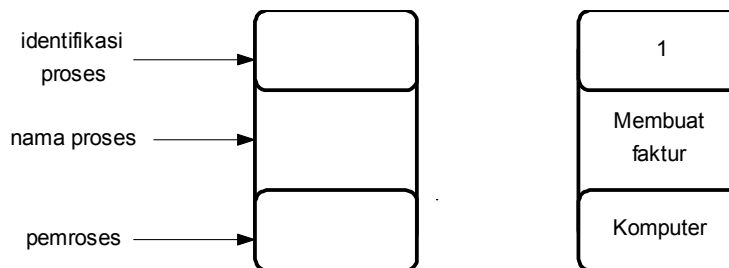
2. Nama proses

Nama proses menunjukkan apa yang dikerjakan oleh proses tersebut. Nama dari proses harus jelas dan lengkap menggambarkan kegiatan prosesnya. Nama dari proses biasanya berbentuk suatu kalimat diawali dengan **kata kerja** (misalnya *menghitung, membuat, membandingkan, memverifikasi, mempersiapkan, merekam* dan lain sebagainya). Nama dari proses diletakkan di bawah identifikasi proses di simbol proses.

3. Pemroses

Untuk PDFD yang menunjukkan proses tidak hanya proses dari komputer, tetapi juga proses manual, seperti proses yang dilakukan oleh orang, mesin dan lain sebagainya, maka pemroses harus ditunjukkan. Pemroses ini menunjukkan siapa atau dimana suatu proses dilakukan.

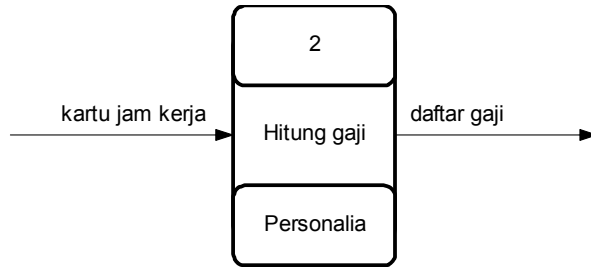
Untuk LDFD yang prosesnya hanya menunjukkan proses komputer saja, maka pemroses dapat tidak disebutkan. Untuk LDFD bila pemroses akan disebutkan dapat juga untuk menyebutkan nama dari program yang melakukan prosesnya. Keterangan pemroses ini di simbol proses dapat dituliskan dibawah nama proses sebagai berikut :



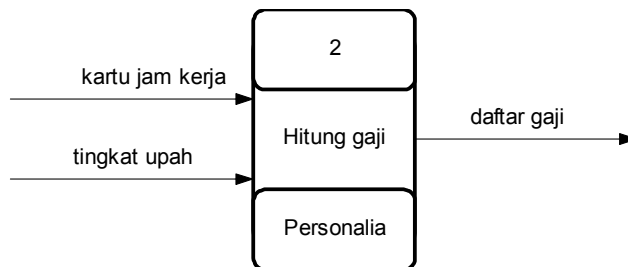
Gambar. Penjelasan di simbol proses

Suatu proses terjadi karena adanya arus data yang masuk dan hasil dari proses adalah juga merupakan arus data lain yang mengalir. Berikut ini adalah berbagai kemungkinan arus data dalam suatu proses

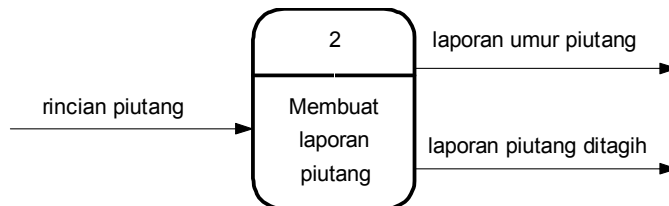
- a. Suatu proses yang menerima sebuah arus data dan menghasilkan sebuah arus data



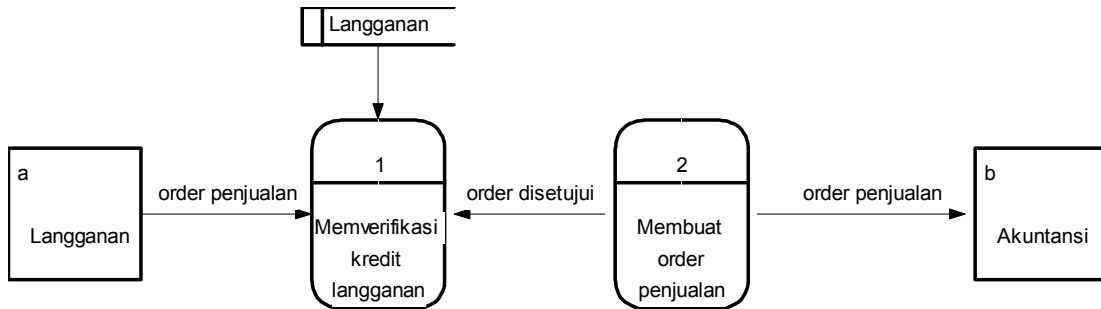
- b. Suatu proses yang menerima lebih dari satu arus data dan menghasilkan sebuah arus data



- c. Suatu proses yang menerima satu arus data dan menghasilkan lebih dari sebuah arus data



suatu proses harus menerima arus data dan menghasilkan arus data. Berikut ini merupakan suatu proses yang salah :



dimanakah letak kesalahannya ? umumnya kesalahan proses di DFD adalah sebagai berikut :

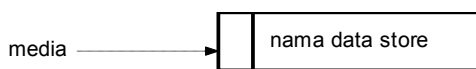
1. Proses mempunyai *input* tetapi tidak menghasilkan output. Kesalahan ini disebut dengan **black hole** (lubang hitam), karena data masuk kedalam proses dan lenyap tidak berbekas seperti dimasukkan ke dalam lubang hitam yang dalam sekali. Kesalahan lubang hitam tampak pada proses nomor 1.
2. Proses menghasilkan *output* tetapi tidak pernah menerima input dan kesalahan ini disebut dengan **miracle** (ajaib), karena secara ajaib dihasilkan *output* tanpa pernah menerima *input*. Proses nomor 2 merupakan kesalahan jenis ini.

SIMPANAN DATA

Simpanan data (*data store*) merupakan simpanan dari data yang dapat berupa sebagai berikut ini :

- a. Suatu *file* atau *database* di sistem komputer
- b. Suatu arsip atau catatan manual
- c. Suatu kotak tempat data di meja seseorang
- d. Suatu tabel acuan manual
- e. Suatu agenda atau buku

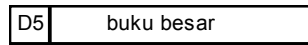
Simpanan data di DFD dapat disimbolkan dengan sepasang garis horisontal paralel yang tertutup di salah satu ujungnya



Gambar. Simbol dari simpanan data di DFD

Nama dari *data store* menunjukkan nama dari *filenya*, misalnya *file* langganan, *file* hutang, *file* arsip faktur dan lain sebagainya. Untuk PDFD, supaya memperjelas simpanan data ini, penjelasan mengenai media dari simpanan data perlu dicantumkan seperti misalnya buku atau arsip, atau suatu kotak dan lain sebagainya. Sedang untuk

LDFD, penjelasan ini dapat digunakan untuk identifikasi dari simpanan data yang berguna sebagai acuan dalam merancang *database*.



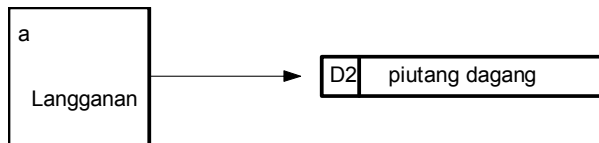
Gambar. Simpanan data yang menunjukkan *file* buku besar dengan nomor acuan D5

Di dalam penggambaran simpanan data di DFD perlu diperhatikan beberapa hal, sebagai berikut :

1. Hanya proses saja yang berhubungan dengan simpanan data, karena yang menggunakan atau merubah data di simpanan data adlah suatu proses.



Gambar. Penggambaran yang salah, karena simpanan data tidak dapat menggunakan data di simpanan data yang lain

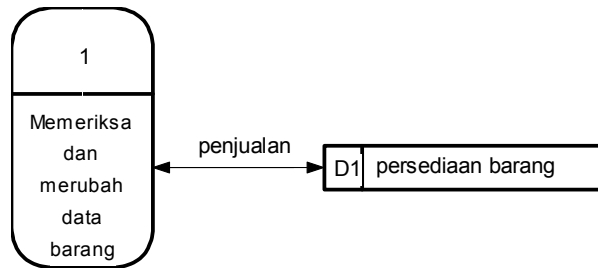


Gambar. Penggambaran yang juga salah, karena tidak jelas data di simpanan data *piutang dagang* akan digunakan untuk apa oleh kesatuan luar *langganan*

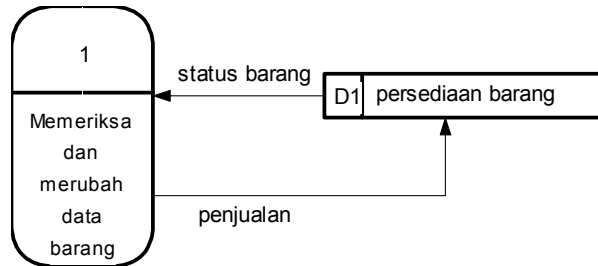
2. Arus data yang menuju ke simpanan data dari suatu proses menunjukkan proses *update* terhadap data yang tersimpan di simpanan data. *Update* dapat berupa proses :
 - a. menambah atau menyimpankan *record* baru atau dokumen baru ke dalam simpanan data
 - b. menghapus *record* atau mengambil dokumen dari simpanan data
 - c. merubah nilai data di suatu *record* atau di suatu dokumen yang ada di simpanan data
3. Arus data yang berasal dari simpanan data ke suatu proses menunjukkan bahwa proses tersebut menggunakan data yang ada di simpanan data. Untuk media simpanan data berupa simpanan luar komputer (*disk* atau *tape*) berarti membaca

data dari suatu *record* di file sedang untuk simpanan data berupa media manual berarti mengambil suatu formulir atau dokumen untuk dilihat isinya dari suatu simpanan data.

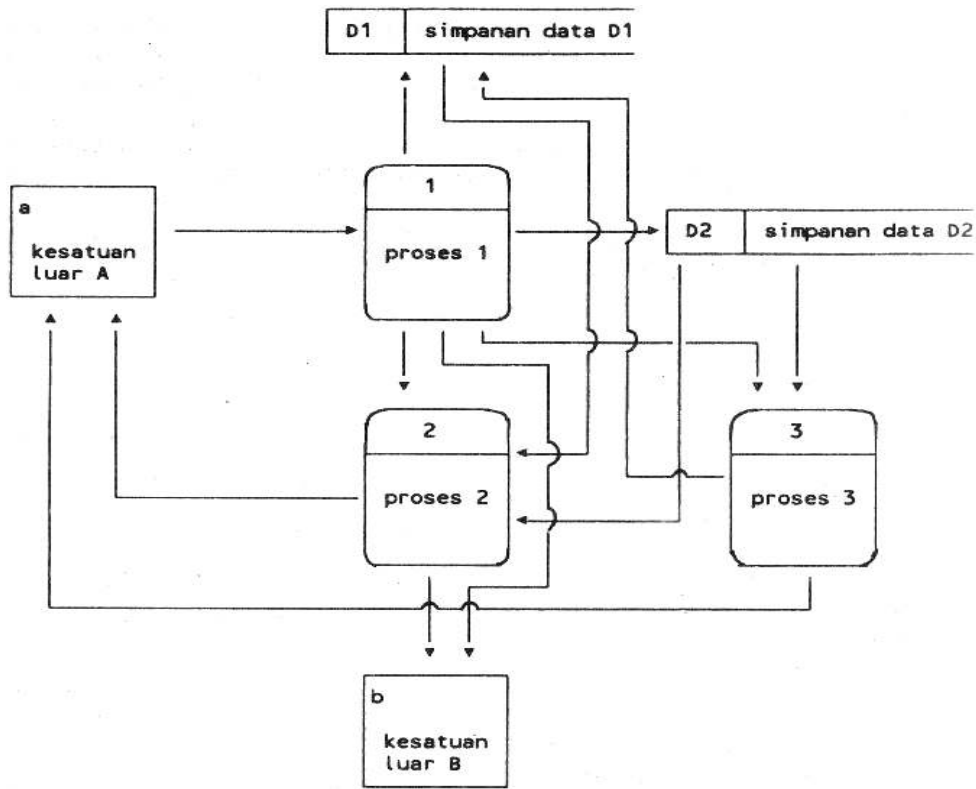
4. Untuk suatu proses yang melakukan kedua-duanya, yaitu menggunakan dan *update* simpanan data dapat dipilih salah satu penggambaran sebagai berikut ini
 - a. Menggunakan sebuah garis dengan panah mengarah kedua arah yang berlawanan dari simpanan data sebagai berikut



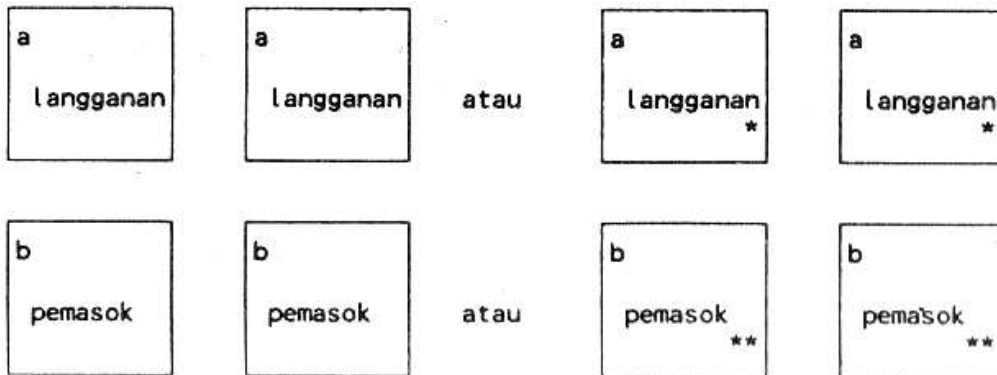
- b. Menggunakan arus data yang terpisah sebagai berikut :



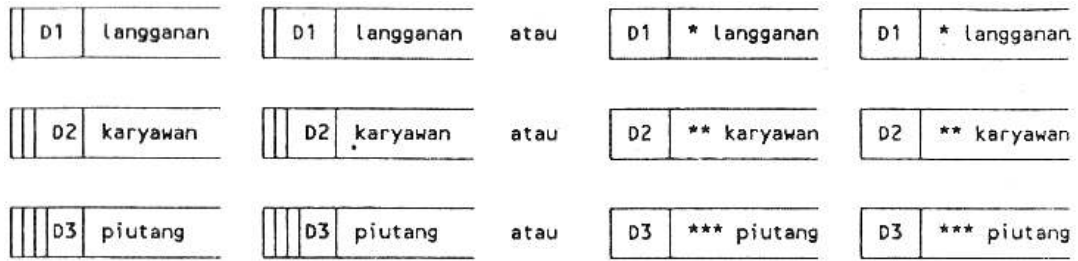
Untuk menghindari garis arus data yang saling berpotongan sehingga membuat gambar di DFD menjadi ruwet, maka simpanan data atau kesatuan luar dapat digambarkan lebih dari sebuah. Misalnya DFD sebagai berikut :



Supaya gambar dari DFD tidak ruwet karena banyaknya garis arus data yang saling berpotongan, maka kesatuan luar dan simpanan data dapat digambarkan lebih dari sebuah. Duplikasi dari kesatuan luar dapat diidentifikasi dengan suatu garis miring (/) atau dengan asterik (*). Sedang duplikasi dari simpanan data dapat diidentifikasi dengan garis vertikal (|) atau dengan asterik (*).

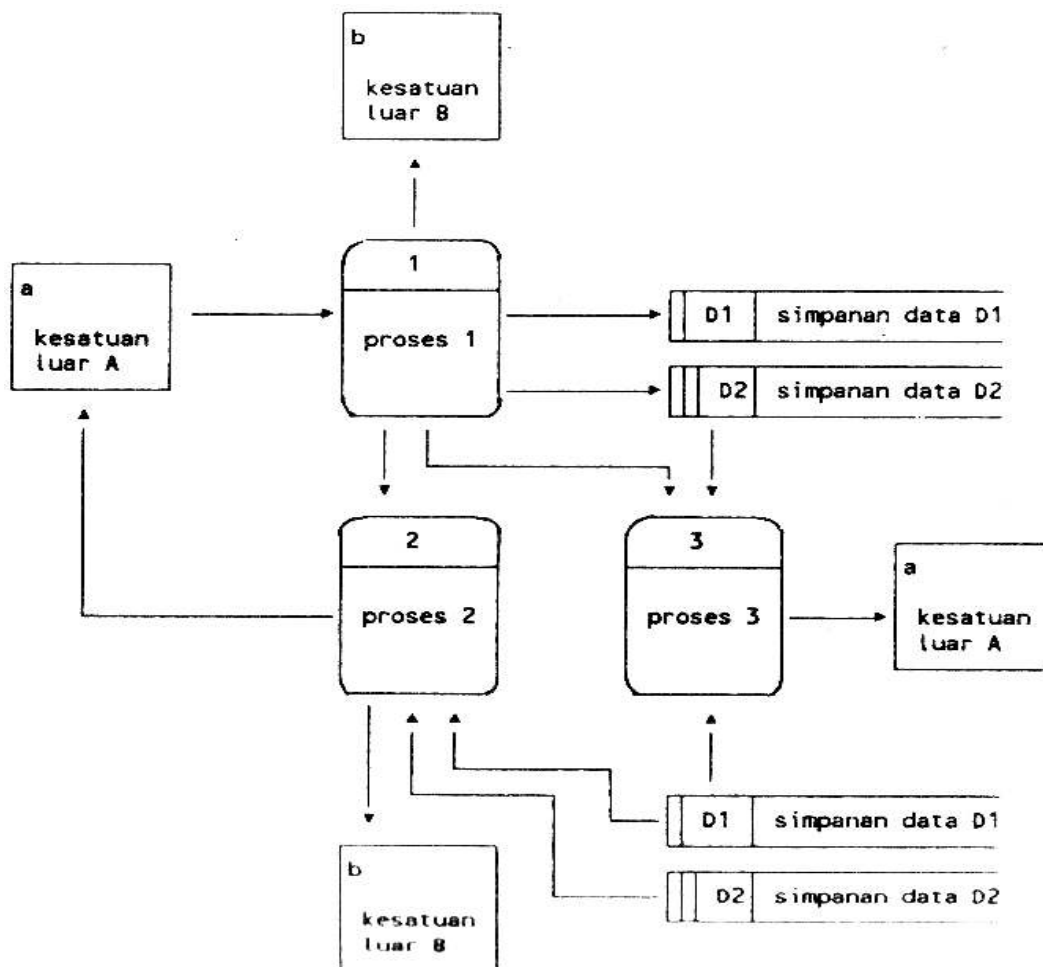


Gambar. Duplikasi kesatuan luar



Gambar. Duplikasi simpanan data

Dengan menggambarkan ulang kesatuan luar dan simpanan data lebih dari sebuah, maka garis-garis arus data yang berpotongan dapat dihindari, sehingga DFD akan tampak tidak terlalu ruwet, seperti sebagai berikut ini :



Bentuk DFD

Terdapat 2 bentuk DFD, yaitu DFD fisik (*Physical Data Flow Diagram*) dan DFD logika (*Logical Data Flow Diagram*). DFD fisik lebih menekankan pada bagaimana proses dari sistem diterapkan sedang DFD logika lebih menekankan proses-proses apa yang terdapat di sistem.

PHYSICAL DATA FLOW DIAGRAM (PDFD)

PDFD lebih tepat digunakan untuk menggambarkan sistem yang ada (sistem yang lama). Penekanan dari PDFD adalah bagaimana proses-proses dari sistem diterapkan (*dengan cara apa, oleh siapa dan dimana*), termasuk proses-proses manual. Dengan menggunakan PDFD, bagaimana proses sistem yang ada akan lebih dapat digambarkan dan dikomunikasikan kepada pemakai sistem, sehingga analisis sistem akan dapat memperoleh gambaran yang jelas bagaimana sistem tersebut bekerja. Untuk memperoleh gambaran bagaimana sistem yang ada diterapkan, PDFD harus memuat sebagai berikut :

1. Proses-proses manual juga digambarkan
2. Nama dari arus data harus menunjukkan fakta penerapannya semacam nomor formulir dan medianya (misalnya telpon atau surat). Nama arus data mungkin juga menerangkan tentang waktu mengalirnya (misalnya harian atau mingguan). Dengan kata lain, nama dari arus data harus memuat keterangan yang cukup terinci untuk menunjukkan **bagaimana** pemakai sistem memahami kerja dari sistem.
3. Simpanan data dapat menunjukkan simpanan non komputer, misalnya kotak *in/out* yang berfungsi sebagai *buffer* dari proses serentak yang beroperasi dengan kecepatan berbeda, sehingga ada sebuah data yang harus menunggu di *buffer*.
4. Nama dari simpanan data harus menunjukkan tipe penerapannya apakah secara manual atau komputerisasi. Secara manual misalnya dapat menunjukkan buku catatan, meja pekerja atau kotak *in/out*. Sedang secara komputerisasi misalnya menunjukkan *file* urut, *file ISAM*, *file database* dan lain sebagainya.
5. Proses harus menunjukkan nama dari pemroses (*processor*), yaitu orang, departemen, sistem komputer atau nama program komputer yang mengeksekusi proses tersebut.

LOGICAL DATA FLOW DIAGRAM (LDFD)

LDFD lebih tepat digunakan untuk menggambarkan sistem yang akan diusulkan (sistem yang baru). LDFD tidak menekankan pada bagaimana sistem diterapkan, tetapi penekanannya hanya pada logika dari kebutuhan-kebutuhan sistem, yaitu

proses-proses apa secara logika yang dibutuhkan oleh sistem. Karena sistem yang diusulkan belum tentu diterima oleh pemakai sistem dan biasanya sistem yang diusulkan terdiri dari beberapa alternatif, maka penggambaran sistem secara logika terlebih dahulu tanpa berkepentingan dengan penerapannya secara fisik akan lebih mengena dan menghemat waktu penggambarannya dibandingkan dengan PDFD. Untuk sistem komputerisasi, penggambaran LDFD yang hanya menunjukkan kebutuhan proses dari sistem yang diusulkan secara logika, biasanya proses-proses yang digambarkan hanya merupakan proses-proses secara komputer saja.

Syarat Pembuatan

Pedoman bagaimana menggambar DFD baik PDFD ataupun LDFD adalah sebagai berikut ini :

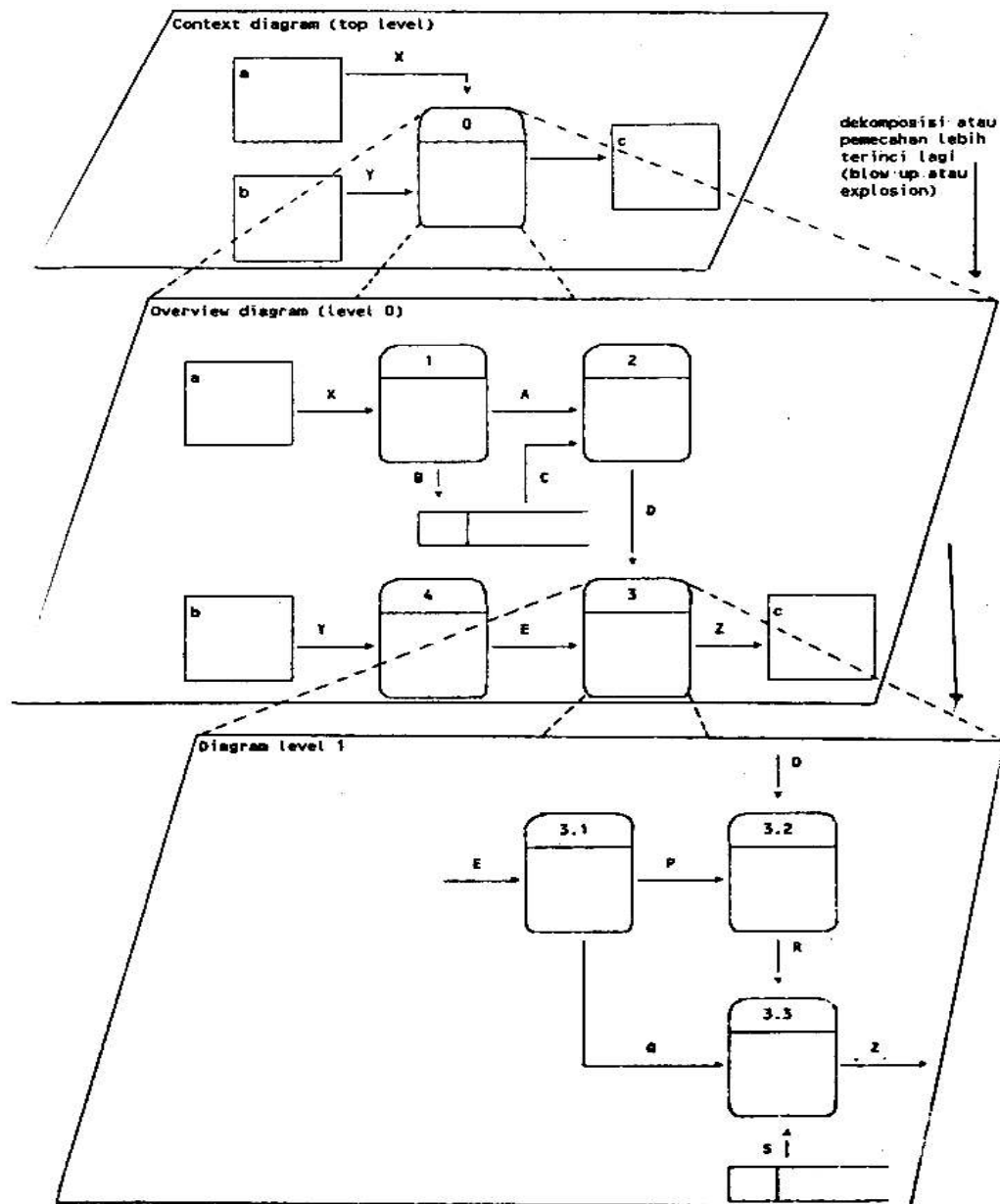
1. Identifikasikan terlebih dahulu semua kesatuan luar (*external entity*) yang terlibat di sistem. Kesatuan luar ini merupakan kesatuan (*entity*) di luar sistem, karena di luar bagian pengolahan data (sistem informasi). Kesatuan luar ini merupakan sumber arus data ke sistem informasi serta tujuan penerima arus data hasil dari proses sistem informasi, sehingga merupakan kesatuan di luar sistem informasi.
2. Identifikasikan semua *input* dan *output* yang terlibat dengan kesatuan luar.

Kesatuan luar	Input	Output
Langganan	Order langganan	-
Bagian gudang	-	Tembusan permintaan persediaan
Bagian pengiriman	Tembusan jurnal	Faktur, tembusan kredit dan tembusan jurnal
Manajer kredit	-	Status piutang

3. Gambarlah terlebih dahulu suatu diagram konteks (*context diagram*). DFD merupakan alat untuk *structured analysis*. Pendekatan terstruktur ini mencoba untuk menggambarkan sistem pertama kali secara garis besar (disebut dengan *top level*) dan memecah-mecahnya menjadi bagian yang lebih terinci (disebut dengan *lower level*).

DFD yang pertama kali digambar adalah level teratas (*top level*) dan diagram ini disebut *context diagram*. Dari *context diagram* ini kemudian akan digambar dengan lebih terinci lagi yang disebut dengan *overview diagram (level 0)*. Tiap-tiap proses di *overview diagram* akan digambar secara lebih terinci lagi dan disebut dengan *level 1*. Tiap-tiap proses di *level 1* akan digambar kembali dengan

lebih terinci lagi dan disebut dengan *level 2* dan seterusnya sampai tiap-tiap proses tidak dapat digambar lebih terinci lagi.



PERBEDAAN DFD DENGAN BAGAN ALIR

DFD sangat berbeda dengan bagan alir (*flow-chart*). Perbedaannya adalah sebagai berikut :

1. proses di DFD dapat beroperasi secara paralel, sehingga beberapa proses dapat dilakukan serentak. Hal ini merupakan kelebihan DFD dibandingkan dengan bagan alir yang cenderung hanya menunjukkan proses yang urut. Kenyataannya kegiatan-kegiatan proses dapat dilakukan secara tidak urut, yaitu secara paralel atau serentak, sehingga DFD dapat menggambarkan proses semacam ini dengan lebih mengena.
2. DFD lebih menunjukkan arus data di suatu sistem, sedang bagan alir sistem lebih menunjukkan arus dari prosedur dan bagan alir program lebih menunjukkan arus dari algoritma.
3. DFD tidak menunjukkan proses perulangan (*loop*) dan proses keputusan (*decision*), sedang bagan alir menunjukkannya.

KETERBATASAN DFD

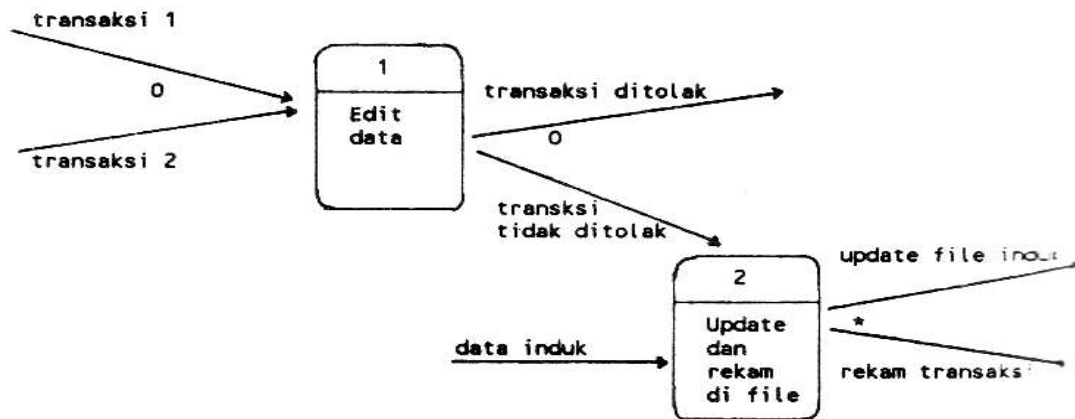
Walaupun DFD mempunyai kebaikan-kebaikan, yaitu dapat menggambarkan sistem secara terstruktur dengan memecah-mecah menjadi level lebih rendah (*decomposition*), dapat menunjukkan arus data di sistem, dapat menggambarkan proses paralel di sistem, dapat menunjukkan simpanan data, dapat menunjukkan kesatuan luar, tetapi DFD juga mempunyai keterbatasan. Keterbatasan DFD adalah sebagai berikut :

1. DFD tidak menunjukkan proses perulangan (*loop*)
2. DFD tidak menunjukkan proses keputusan (*decision*)
3. DFD tidak menunjukkan proses perhitungan

A. Ziya Aktas memberikan pemecahan untuk keterbatasan DFD ini, yaitu dengan menambahkan penggunaan operational operator (operator hubungan), sehingga kemampuan DFD dapat lebih ditingkatkan. Operator hubungan ini adalah :

- * menunjukkan hubungan suatu logika AND
- menunjukkan hubungan suatu logika OR
- ⊕ menunjukkan hubungan suatu logika XOR

Contoh penggunaan dari operator hubungan ini DAD tampak sebagai berikut ini.



EVALUASI

1. Jelaskan definisi dari simbol-simbol yang digunakan di DFD disertai dengan notasi-notasi yang mewakilinya !
2. Jelaskan perbedaan utama antara DFD fisik dengan DFD Logika !
3. Jelaskan hal-hal apa saja yang harus dilakukan ketika menggambar DFD !
4. Apa yang menjadi perbedaan antara DFD dengan bagan alir (flow-chart) ? jelaskan !
5. Apa saja keterbatasan-keterbatasan dari DFD !
6. Sebagai latihan buatlah DFD sistem administrasi penerimaan siswa baru di unit anda !

DAFTAR PUSTAKA

1. Burch, J.G., **System, Analysis, Design, and Implementation**, Boyd & Fraser Publishing Company, 1992.
 2. Jogiyanto, **Analisis dan Disain Sistem Informasi**, ANDI OFFSET Yogyakarta, 1990.
 3. John G. Burch, Jr, Felix R. Strater, Gary Grudnitski, **Information Systems : Theory and Practice**, Second Edition, John Wiley & Sons, 1979
 4. Meilir Page-Jones, **The Practical Guide to Structured Systems Design**, Second Edition, Yourdon Press, Prentice Hall, 1988
 5. I.T. Hawryszkiewicz, **Introduction Systems Analysis and Design**, Second Edition, Prentice Hall, 1991
 6. Raymond McLeod, Jr, **Management Information System : A Study of Computer-Based Information Systems**, Sixth Edition, Prentice Hall, 1979
 7. A. Ziya Aktas, **Structured Analysis & Design of Information Systems**, NJ: Prentice Hall, 1987, hal. 65
-