**Context Diagram Level**

**Menentukan Tujuan (Statement of Purpose atau STP)**

Berisi deskripsi tekstual fungsi sistem. Hal ini berguna bagi manajemen atas, menengah dan bawah yang terlibat secara langsung dalam pengembangan sistem. Contoh :

*Kegunaan sistem pemrosesan buku adalah menangani semua aspek detil pemesanan buku oleh pelanggan, seperti pengiriman, pembayaran dan pengembalian bukti pembayaran pelanggan. Informasi buku juga disediakan bagi sitem lain seperti pemasaran, penjualan, dam keuangan.*

STP tidak digunakan untuk menjelaskan sistem secara rinci tetapi garis besarnya saja. Apabila dijelaskan secara rinci dapat menimbulkan pertanyaanyang lebih luas dimana penjelasan rinci selanjutnya dilakukan pada langkah berikutnya. STP cenderung menjelaskan aspek keuntungan secara kuantitas sebagai akibat pengembagan sistem, misalnya mengurangi waktu pemrosesan menjadi setengahnya, biaya operasional dapat ditekan setengahnya, waktu lembur yang dibutuhkan berkurang setengah dari semula, dan sebagainya.

**Data Flow Diagram Context Diagram Level**

**(Context Diagram/CD)**

Context Diagram adalah bagian dari Data Flow Diagram (DF) yang berfungsi memetakan model lingkungan, yang dipresentasikan dengan lingkaran tunggal yang mewakili keseluruhan sistem. CD menyoroti sejumlah karakteristik penting sistem, yaitu :

1. Kelompok pemakai, organisasi atau sistem lain dimana sistem melakukan komunikasi (sebagai terminator).
2. Data masuk, yaitu data yang diterima sistem dari lingkungan dan harus diproses dengan cara tertentu.
3. Data keluar, yaitu data yang dihasilkan sistem dan diberikan ke dunia luar.
4. Penyimpanan data (storage), yaitu digunakan secara bersama antara sistem dengan terminator. Data ini dapat dibuat oleh sistem dan digunakan oleh lingkungan atau sebaliknya dibuat oleh lingkungan dan digunakan oleh sistem. Hal ini berarti pembuatan simbol data storage dalam CD dibenarkan, dengan syarat simbol tersebut merupakan bagian dari dunia diluar sistem.
5. Batasan, antara sistem dan lingkungan.

Simbol yang digunakan dalam Context Diagram (CD), antara lain :

1. Persegi panjang (terminator)

Untuk berkomunikasi langsung dengan sistem melalui aliran data. Antara terminator tidak diperbolehkan komunikasi langsung.

1. Lingkaran

Untuk menunjukkan adanya kegiatian proses dalam sistem.

Bentuk simbol dapat dilihat pada gambar 2.2.

Aturan-aturan CD :

1. Bila terdapat terminator yang mempunyai banyak masukan dan keluaran, diperbolehkan untuk digambarkan lebih dari satu kali sehingga mencegah penggambaran yang terlalu rumit, dengan ditandai secara khusus untuk menjelaskan bahwa terminator yang dimaksud adalah identik. Tanda dapat berupa asterisk (\*) atau tanda kres (#).
2. Bila terminator mewakili individu (personil) sebaiknya diwakili oleh peran yang dimainkan personil tersebut. Alasannya adalah : personil yang berfungsi untuk melakukan itu dapat berganti, sedangkan CD harus tetap akurat walaupun personil berganti dan mungkin seorang personil dapat memiliki lebih dari satu tugas (peran).
3. Karena model ini membedakan sumber (resources) dan pelaku (handler).Dimana pelaku adalah mekanisme, perangkat, atau media fisik yang mentransformasikan data ke/dari sistem, sehingga pelaku tidak perlu digambarkan.

Aliran dalam CD memodelkan masukkan ke sistem dan keluaran dari sistem, seperti halnya sinyal kontrol yang diterima atau dibuat sistem. Aliran data hanya digambarkan jika diperlukan untuk mendeteksi kejadian dalam lingkungan dimana sistem harus memberikan respon atau membutuhkan data untuk menghasilkan respon. Selain itu aliran data dibutuhkan untuk menggambarkan transportasi antara sistem dan terminator. Dengan kata lain aliran data digambarkan jika data tersebut diperlukan untuk menghasilkan respon pada kejadian tertentu.

Dalam hal ini seharusnya menggambar dengan asumsi bahwa masukan disebabkan dan diinisiasi oleh terminator, sedangkan keluaran disebabkan dan diinisiasi oleh sistem. Hal itu dilakukan dengan mencegah interaksi yang tidak perlu(extraneous prompts) yang berorientasi pada implementasi masukan-keluaran,dan mengkonsentrasikan pemodelan pada aliran data yang esensial saja.

CD dimulai dengan penggambaran terminator, aliran data, aliran kontrol,penyimpanan, dan proses tunggal yang mempresentasikan keseluruhan sistem.Bagian termudah adalah menetapkan proses yang hanya terdiri dari satulingkaran dan diberi nama yang mewakili sistem. Nama harus dapat menjelaskan proses.

Langkah yang dapat membantu dalam menggambarkan CD :

1. Identifikasikan seluruh informasi yang dibutuhkan.

2. Identifikasikan seluruh data yang dibutuhkan proses/informasi.

3. Identifikasikan seluruh tujuan setiap informasi bagi penggunanya.

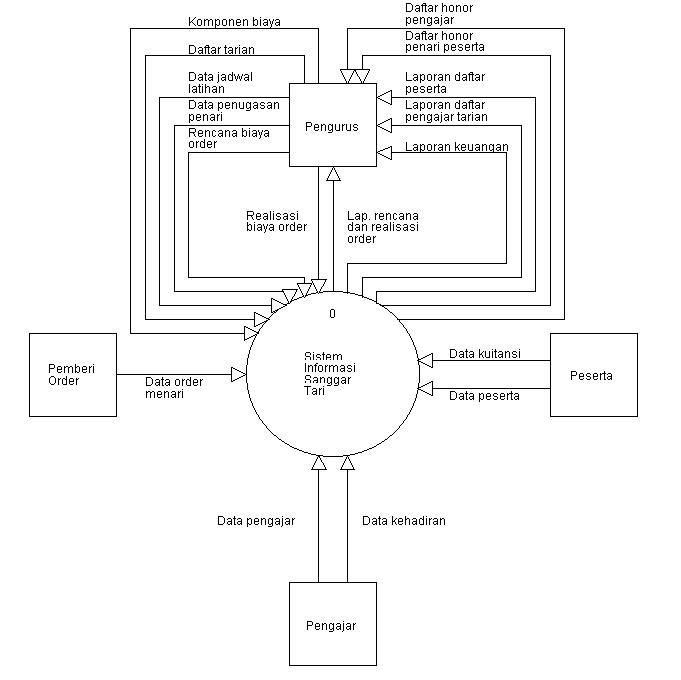
4. Identifikasikan seluruh sumber data yang dibutuhkan proses/informasi

Contoh :

|  |  |  |  |
| --- | --- | --- | --- |
| No | Nama Data atau Informasi | Arah aliran (Sumber/Tujuan) | Terminator |
| 1 | Daftar tarian | Menuju sistem | Pengurus |
| 2 | Data peserta | Menuju sistem | Peserta |
| 3 | Data kuitansi | Menuju sistem | Peserta |
| 4 | Data pengajar | Menuju sistem | Pengajar |
| 5 | Data jadwal latihan | Menuju sistem | Pengurus |
| 6 | Daftar kehadiran | Menuju sistem | Pengajar |
| 7 | Data order menari | Menuju sistem | Pengajar |
| 8 | Data penugasan penari | Menuju sistem | Pengurus |
| 9 | Data honor pengajar | Dari sistem | Pengurus |
| 10 | Daftar honor penari peserta | Dari sistem | Pengurus |
| 11 | Rencana biaya order | Menuju sistem | Pengurus |
| 12 | Realisasi biaya order | Menuju sistem | Pengurus |
| 13 | Laporan daftar peserta tarian | Dari sistem | Pengurus |
| 14 | Laporan daftar pengajar tarian | Dari sistem | Pengurus |
| 15 | Laporan rencana dan realisasi order | Dari sistem | Pengurus |
| 16 | Laporan keuangan | Dari sistem | Pengurus |
| 17 | Komponen biaya | Menuju sistem | Pengurus |

Tabel 2.1 : Inventarisasi data, informasi, sumber data, tujuan informasi

Sumber : Pengantar Perancangan Sistem, Husni Iskandar Pohan, dkk.



Gambar 2.1 : Context Diagram Sistem Informasi Sanggar Tari

Sumber : Pengantar Perancangan Sistem, Husni Iskandar Pohan, dkk.

**Even List**

Adalah daftar narasi daftar kejadian (stimuli) yang terjadi dalam lingkungan dan mempunyai hubungan dengan respon yang diberikan sistem. Secara umum setiap aliran data dalam CD adalah kejadian (event), tepatnya aliran data mengindikasikan terjadinya kejadian, atau aliran data dibutuhkan oleh sistem untuk melakukan proses. Aturan-aturan dalam even list, antara lain daftar kejadian yang didesain dan digambarkan dalam bentuk tekstual sederhana yang berfungsi memodelkan kejadian dalam lingkungan dimana sistem harus memberikan respon. Ketika membuat EL harus yakin perbedaan antara event dan kejadian yang berelasi dengan aliran (event-related flow)

Contoh :

1. Administrasi peserta, terdiri atas kejadian :

1.1 Pencatatan daftar tarian.

1.2 Pencatatan peserta.

1.3 Pencatatan pembayaran biaya latihan peserta.

1.4 Pembuatan laporan daftar peserta tarian.

2. Administrasi pengajar, terdiri atas kejadian :

2.1 Pencatatan daftar pengajar.

2.2 Pencatatan jadwal dan kelompok latihan.

2.3 Pencatatan absensi pengajar/latihan.

2.4 Pembuatan laporan daftar pengajar tarian.

3. Order menari, terdiri atas kejadian :

3.1 Pencatatan order menari.

3.2 Pencatatan realisasi biaya menari.

3.3 Pembuatan laporan realisasi biaya.

4. Penugasan dan honor, terdiri atas kejadian :

4.1 Pencatatan penugasan penari.

4.2 Pencatatan pembayaran honor pengajar.

4.3 Pencatatan pembayaran honor penari peserta.

5. Laporan keuangan

**Data Flow Diagram Levelled (DFDL)**

Model ini berfungsi untuk menggambarkan sistem sebagai jaringan kerja antar fungsi yang berhubungan satu sama lain dengan aliran dan penyimpanan data. DFD pada dasarnya sebuah diagram yang menjelaskan bagaimana hubungan bersama dari bagian file, laporan, sumber dokumen dan sebagainya. DFD termasuk alat komunikasi mediun yang baik antara designer dan pemakai karena mudah dipahami (hanya berisi 4 simbol). Tujuan dari DFD adalah membuat/mengetahui aliran (track) aliran data seluruhnya dari sistem. Data dan proses adalah hal yang kritis untuk dipahami. DFD berbeda dengan flow sistem (systems flowcharts) dan flow program (program flowcharts) karena keduanya lebih mengarah ke hasil (orientation).

System flowcharts adalah device-specific flowchart yang melacak (track)urutan pengolahan data dan mengidentifikasikan komponen fisik sebagai *auxiliary storage devices*.

Program flowcharts menyangkut primitive level logic dari sebuah program. Level yang rinci sangat baik untuk *system oriented view*. Programmer dan analis yang menggunakan program flowcharts sering memiliki kesulitan membuat peralihan (transition) ke DFD. Namun kedua alat tersebut memiliki kesamaan, yaitu penggunaan simbol proses dan flow lines, tetapi masing-masing memiliki tujuan yang sangat berbeda.

Rancangan (blueprint) mencakup satu atau lebih DFD yang menunjukkan hubungan antara file, input dokumen, output dokumen dan laporan.

Empat komponen dalam DFDL :

1. Proses (fungsi)

Dipresentasikan dalam bentuk lingkaran (circle) atau bujursangkar dengan sudut melengkung (a rounded rectangle). Setiap proses ditandai dengan nomor. Nomor berfungsi menjelaskan tingkatan proses dari hierarchy chat. Setiap proses dinamai dengan sepasang kata kerja yang simple, contoh : screen customer-order, record customer-order. Dalam physical DFD, lokasi atau program yang memproses seringkali dituliskan dibagian bawah simbol, tetapi dalam logical DFD tidak perlu disebutkan. Untuk menunjukkan transformasi dari masukan menjadi keluaran, dalam hal ini sejumlah masukan dapat menjadi hanya satu keluaran ataupun sebaliknya. Proses umumnya didefinisikan dengan kata tunggal, atau kalimat sederhana. Fokus simbol ini adalah apa yang dikerjakan atau tindakan yang dilakukan (proses), bukan orang atau melakukan kegiatan apa.

1. Aliran data (data arrow).

Dipresentasikan delam bentuk anak panah yang menuju ke atau dari proses. Digunakan untuk menggambarkan gerakan paket data atau informasi dari satu bagian ke bagian lain dari sistem dimana penyimpanan mewakili lokasi penyimpanan data. Nama berfungsi untuk mendefinisikan arti dari aliran dan ditulis untuk mengidentifikasi aliran tersebut. Ujung panah menunjukkan kemana data bergerak ke atau dari proses, penyimpanan ataupun terminator atau keduanya. Aliran yang digambarkan sebagai panah dengan dua ujung menggambarkan terjadinya dialog. Aliran dapat juga menyebar atau menyatu, misalnya sejumlah atribut dapat membentuk satu aliran, atau satu aliran menyebar menjadi sejumlah atribut. Atribut dalam hal ini dapat merupakan bagian atau duplikasi dari aliran. Nama data yang digambarkan dalam aliran disebut *data packet* dan dituliskan diatas garis panah. Arus data menunjukkan arus dari data yang dapat berupa masukan untuk sistem atau hasil proses sistem dan dapat berbentuk sebagai berikut :

a. Formulir atau dokumen yang digunakan di perusahaan.

b. Laporan tercetak yang dihasilkan oleh sistem.

c. Tampilan atau output dilayar komputer yang dihasilkan oleh sistem.

d. Masukan untuk komputer.

e. Komunikasi lisan / ucapan.

f. Surat-surat atau memo.

g. Data yang dibaca atau direkamkan ke suatu file.

h. Suatu sistem yang dicatat pada buku agenda.

i. Transmisi data dari suatu komputer ke komputer yang lain.

1. Penyimpanan (data stores)

Komponen ini digunakan untuk memodelkan kumpulan data atau paket data. Notasi yang digunakan adalah garis sejajar, segiempat dengan sudut melengkung atau persegi panjang, atau open-ended rectangle on the rightside. Nama (database atau file) dari penyimpanan disebutkan dalam symbol. Dalam kasus notasi ini dapat mendefinisikan file atau basis data seperti tape magnetic, disk, dan model DBMS lainnya, atau seringkali mendefinisikan bagaimana penyimpanan diimplementasikan dalam sistem komputer. Penyimpanan kadangkala didefinisikan sebagai suatu mekanisme di antara dua proses yang dibatasi oleh jangka waktu tertentu.

1. Terminator (external or internal entities)

Dipresentasikan dengan simbol persegi panjang yang mewakili entiti luar atau dalam dimana sistem berkomunikasi dan disebut dengan sources (data, masukan ke sistem) atau destinations (informasi, keluarah dari sistem).Seringkali sulit untuk menentukan mana external dan internal entity.

Pertanyaan kunci untuk menjawabnya adalah "apakah pelaku (person)atau group ini melakukan proses dari bagian sistem ? Jika jawabannnya tidak, entitiy tersebut adalah external. External entity jika jelas bahwa entity tersebut diluar sistem organisasi, seperti customers, vendors, pemerintah, orang, kelompok orang, perusahaan, departemen. Contoh dalam order entry system adalah customer, order inventory system adalah suppliers. Internal entity jika mewakili unit-unit fungsi suatu organisasi, yang meliputi pemakai. External dan internal entity juga dapat didefinisikan sebagai batasan sistem.

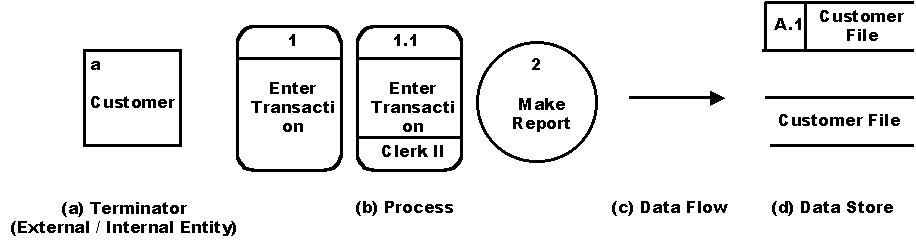
Untuk mengidentifikasi terminator tidak sulit. Kadangkala terminator berupa pemakai sistem itu sendiri. Hal itu bisa terjadi jika pemakai punya sikap sikap seperti berikut :

*"Saya menyediakan data x, y, z bagi sistem, dan saya mengharapkan system dapat memberikan informasi/data a, b, dan c"*

Pada kasus lain pemakai berpikir bahwa dia merupakan bagian dari sistem sehingga mempermudah pemodelan untuk mengidentifikasi terminator yang relevan. Hal itu bisa terjadi jika pemakai bersikap :

*"Kami menerima transaksi penerimaan dari Departemen Akuntasi, dan kamiharus menyerahkan laporan mingguan biaya ke Komisi Keuangan".*

Simbol untuk entity external adalah sebuah bujur sangkar (square).Untuk mengidentifikasinya adalah dengan memberikan tanda alfabet huruf kecil yang dituliskan pada sudut kiri atas dari bujur sangkar, selanjutnya diberi sebuah nama tunggal.

Gambar 2.2 : Simbol untuk Context Diagram dan Data Flow Diagram Level

Sumber : Analysis and Design of Business Information System,Merle P.Martin

Ada hal penting tentang terminator :

1. Terminator merupakan bagian luar sistem, dan aliran data (panah) yang dihubungkan dengan terminator (ke/dari proses, ke/dari penyimpanan) dalam sistem memodelkan hubungan antara sistem dengan dunia luar.
2. Penganalisa sistem punya kemungkinan untuk memodifikasi esensi terminator dengan mengubah cara kerja sistem, karena seorang penganalisa sistem bertujuan membuat sistem sefleksibel mungkin dengan kebebasan memilih yang terbaik, misalnya karena efisiensi atau karena lebih handal, bagi implementasi yang paling mungkin dilakukan.

Dengan demikian, komponen DFD terdiri dari sejumlah komponen yang sederhana, yaitu proses (process), aliran (flows), penyimpanan (stores) dan terminator. Dengan komponen tersebut, dapat digunakan untuk berkomunikasi dengan pemakai yang berorientasi pada fungsi (function-oriented view).Ketika membuat DFD harus teliti, agar DFD tidak terlalu sederhana, salah

## dan tidak konsisten.

Beberapa petunjuk untuk membuat DFD yang jelas, dan mudah dibaca :

1. Pilih nama yang jelas maksudnya (bagi proses, aliran, penyimpanan,

dan terminator

a. Untuk proses sebaiknya menggunakan nama yang mengacu pada fungsi, yaitu gabungan antara kata kerja yang spesifik dan obyek, misalnya memproses laporan inventori, validasi nomer telepon dan lain-lain.

Untuk terminator lebih mengacu pada orang atau kelompok orang, sedangkan untuk aliran dan penyimpanan mengacu pada paket data atau informasi yang terkandung didalamnya.

b. Jangan menggunakan nama yang terlalu umum, misalnya proses data, tangani masukan dan lain-lain.

c. Gunakan nama yang familiar bagi pemakai.

2. Menomori proses untuk memperjelas sistematika.

a. Tidak jadi masalah bagaimana urutan ditempatkan.

b. Nomor tidak menunjukkan urutan.

c. Penomoran dimaksudkan sebagai identifikasi proses dan memudahkan penurunan ke level yang lebih rendah atau ke proses berikutnya.

3. Menggambar kembali DFD hingga beberapa kali, sehingga cukup estetik. Penggambaran kembali yang kadang-kadang bahkan lebih dari sepuluh kali digunakan untuk menjaga secara teknis gambar tersebut sudah benar, dapat diterima oleh pemakai, misalnya tingkat operasional yang akan mengoperasikan dan dapat diterima oleh pimpinan yang dalam hal ini menentukan strategi organisasi. Ketika penggambaran dilakukan ada beberapa hal yang perlu diperhatikan, yaitu :

a. Ukuran dan bentuk lingkaran sebaiknya tetap sama, karena jika tidak dapat menimbulkan kesan lingkaran yang lebih besar memproses sesuatu yang juga lebih besar.

b. Panah yang melengkung dan lurus tidak jadi masalah tetapi sebaiknya tidak menggunakan kedua cara tersebut pada gambar yang sama.

c. Menggambar dengan tangan atau menggambar dengan perangkat tertentu, tidak merupakan masalah. Menggambar dengan tangan seringkali memberikan sesuatu yang lain katakanlah sentuhan seni, tetapi menggambar dengan mesin, seringkali lebih memudahkan modifikasi.

4. Mencegah DFD yang terlalu kompleks dan tidak perlu. Kegunaan DFD bukan hanya menggambarkan fungsi dan interaksinya dalam sistem secara akurat tetapi juga untuk dibaca dan dimengerti oleh bukan hanya penganalisa sistem, tetapi juga pemakai yang berpengalaman dalam sistem yang dimodelkan. Hal ini berarti : jangan membuat DFD dengan terlalu banyak proses, aliran, penyimpanan dan terminator.

5. Menjamin konsistensi DFD tersebut secara intern ataupun yang berkualitas dengan DFD. Konsistensi dalam hal ini juga menyangkut konsistensi dengan model lain (misalnya : entity relationship diagram, state transition diagram, data dictionary dan process specification).

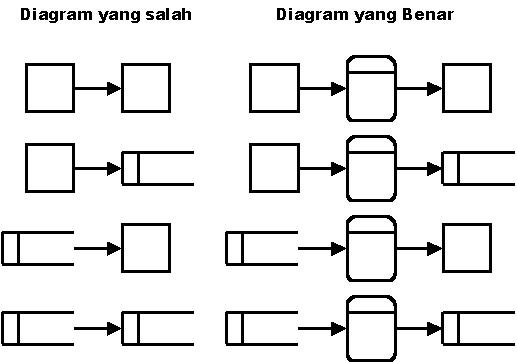
Hal-hal penting yang harus diperhatikan / diingat adalah :

a. Mencegah proses yang mempunyai masukan tetapi tidak mempunyai keluaran yang dikenal dengan **lubang hitam**

b. Mencegah proses yang mempunyai keluaran tetapi tidak punya masukan, misalnya penghasil bilangan acak.

c. Hati-hati dengan aliran dan proses yang tidak dinamakan karena dapat mengakibatkan elemen data yang saling tidak berhubungan menjadi satu.

d. Hati-hati dengan penyimpanan yang punya status hanya dapat dibaca atau hanya dapat ditulis dan berkaitan dengan proses yang hanya memproses masukan atau hanya memproses keluaran.



Gambar 2.4 : Bentuk penggambaran diagram yang salah (tidak menggunakan proses)

Sumber : System Analysis & Design Methods, Whitten, Bentley, Barlow)

Sebagaimana analogi dalam bahasan sebelumnya yang membahas tentang peta dan penurunnya dalam peta yang lebih detail dan detil lagi, maka demikian juga dengan DFD. Level paling tinggi dalam suatu DFD hanya punya sebuah lingkaran (proses) yang memodelkan seluruh sistem, sedangkan aliran memodelkan hubungan antara sistem dengan terminal diluar system (external terminator). Level ini disebut Context Diagram.

Dalam hal ini berperan pemberian nomor pada setiap lingkaran pada DFD yang berguna untuk memudahkan penurunan DFD ke level yang lebih rendah. Penurunan ini mengacu pada status tertentu, yaitu :

a. Setiap penurunan ke level yang lebih rendah harus mampu mempresentasikan proses tersebut dalam spesifikasi proses yang jelas, sehingga seandainya

belum cukup jelas maka seharusnya diturunkan ke level yang lebih rendah.

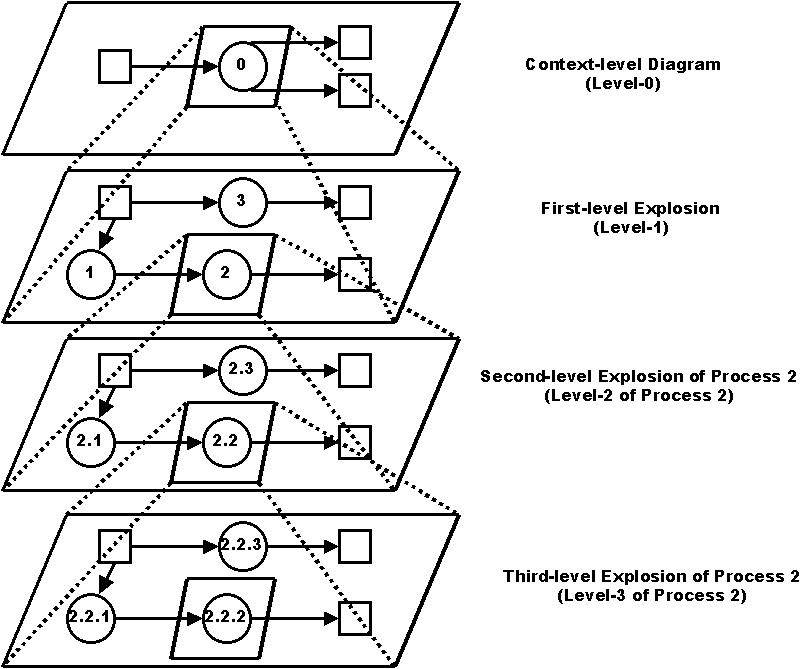
b. Setiap penurunan harus dilakukan hanya jika perlu.

c. Tidak semua bagian dari sistem harus diturunkan dengan jumlah level yang sama karena yang kompleks bisa saja diturunkan, dan yang sederhana mungki tidak perlu diturunkan. Selain itu, karena tidak semua proses dalam level yang sama punya derajat kompleksitas yang sama juga.

d. Konfirmasikan DFD yang telah dibuat pada pemakai dengan cara top-down.

e. Aliran data yang masuk dan keluar pada suatu proses di level harus berhubungan dengan aliran data yang masuk dan keluar pada level x+1. Dimana level x+1 tersebut mendefinisikan sub proses pada level x tersebut.

f. Ketika mulai menurunkan DFD dari level tertinggi, cobalah untuk mengindentifikasi external events dimana sistem harus memberikan respon. External events dalam hal ini berarti suatu kejadian yang berkaitan dengan pengolahan data di luar sistem, dan menyebabkan sistem kita memberikan respon.



Gambar 2.3 : Penurunan Context Daigram menjadi Data Flow Diagram Level

Sumber : Analysis and Design of Business Information System,Merle P.Martin

DFD sebagai perangkat pemodelan bisa dikatakan sederhana, tetapi sangat berguna untuk memodelkan fungsi dalam sistem. Jika yang akan dimodelkan punya ketergantungan dengan waktu (misalnya real time system), maka harus digunakan tambahan model lain atau kadang-kadang tambahan notasi lain, artinya tidak bisa hanay dengan DFD. Tetapi ironisnya, banyak penganalisa sistem mengira hanya dengan DFD, mereka sudah tahu segala-galanya tentang analisa terstruktur. Pada kenyatannya telah dibuktikan bahwa tanpa perangkat pemodelan tambahan, DFD masih sesuatu yang serba kekurangan.

Perbedaan antara Physical DFD dan Logical DFD :

a. Physical DFD

Diagram phisik memiliki atribut yang menjelaskan tentang, lokasi dimana sebuah proses dilakukan, siapa yang melakukan proses tersebut, perangkat apa yang digunakan untuk melakukan proses tersebut.

Atribut tersebut biasanya dituliskan pada bagian bawah simbol proses (bawah garis horisontal)

b. Logical DFD

Diagram logik hanya berisi data yang diarahkan menuju/dari sistem, yaitu data yang benar-benar dibutuhkan proses.

**Daftar Pustaka**

1. HM, Jogiyanto, *Analysis and Disain Sistem Informasi (Pendekatan*

*terstruktur)*, Penerbit Andi Offset, Yogyakarta, 1995.

2. Martin, Merle P., *Analysis and Design of Business Information System*,

Macmillan Publishing Company, New York, 1991.

2. Pohan, Husni Iskandar, *Pengantar Perancangan Sistem*, Penerbit

Erlangga, Jakarta, 1997.

3. Whitten, Bentley, Barlow, *Systems Analysis & Design Methods*,

Penerbit IRWIN, USA, 1989.