

BAB 6 PEMODELAN SISTEM (CONTEXT DIAGRAM, DATA FLOW DIAGRAM)

1. PENDAHULUAN

Tiga alasan yang menyebabkan sebaiknya dilakukan pemodelan sistem, yaitu:

- Dapat melakukan perhatian pada hal-hal penting dalam sistem tanpa mesti terlibat terlalu jauh.
- Mendiskusikan perubahan dan koreksi terhadap kebutuhan pemakai dengan resiko dan biaya minimal.
- Menguji pengertian penganalisa sistem terhadap kebutuhan pemakai dan membantu pendesain sistem dan pemrogram membangun sistem.

Tetapi ada banyak bentuk model yang dapat digunakan dalam perancangan sistem antara lain model narasi, model prototype, model grafis dan lain-lain. Dalam hal ini tidak jadi masalah model mana yang akan digunakan, yang jelas harus mampu merepresentasikan visualisasi bentuk sistem yang diinginkan pemakai, krena sistem akhir yagn dibuat bagi pemakai akan diturunkan dari model tersebut.

Perangkat yang digunakan untuk memodelkan suatu sistem diantaranya adalah:

1. Context Diagram
2. Data Flow Diagram
3. Kamus Data
4. Spesifikasi Proses

2. CONTEXT DIAGRAM (DIAGRAM KONTEKS)

Model berikutnya menjawab sejumlah pertanyaan yang muncul dalam pembuatan statement of purpose. Context Diagram merupakan kejadian tersendiri dari suatu diagram alir data. Dimana satu lingkaran merepresentasikan seluruh sistem. Context Diagram ini harus berupa suatu pandangan, yang mencakup masukan-masukan dasar, sistem-sistem dan keluaran.

Context Diagram merupakan tingkatan tertinggi dalam diagram aliran data dan hanya memuat satu proses, menunjukkan sistem secara keseluruhan. Proses tersebut diberi nomor nol. Semua entitas eksternal yang ditunjukkan pada diagram konteks berikut aliran data-aliran data utama menuju dan dari sistem. Diagram tersebut tidak memuat penyimpanan data dan tampak sederhana untuk diciptakan, begitu entitas-entitas eksternal serta aliran data-aliran daa menuju dan dari sistem diketahui penganalisis dari wawancara dengan user dan sebagai hasil analisis dokumen.

Context diagram menggarisbawahi sejumlah karakteristik penting dari suatu sistem:


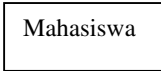

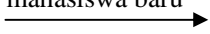
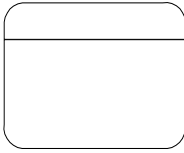
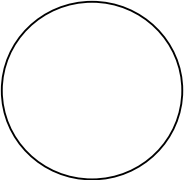

- **Kelompok pemakai**, organisasi, atau sistem lain dimana sistem kita melakukan komunikasi yang disebut juga sebagai terminator.

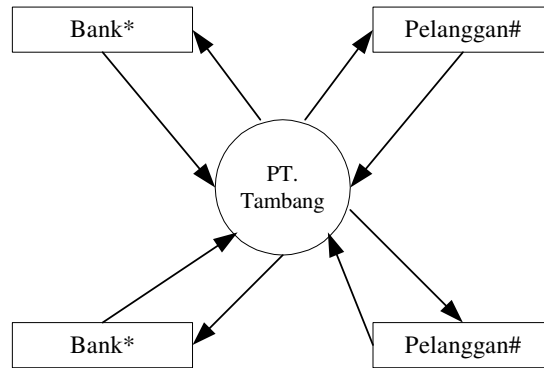
- **Data** dimana sistem kita menerima dari lingkungan dan harus diproses dengan cara tertentu.
- **Data** yang dihasilkan sistem kita dan diberikan ke dunia luar.
- **Penyimpanan data** yang digunakan secara bersama antara sistem kita dengan terminator. Data ini dibuat oleh sistem dan digunakan oleh lingkungan atau sebaliknya,, dibuat oleh lingkungan dan digunakan oleh sistem kita.
- **Batasan** antara sistem kita dan lingkungan.

Context Diagram dimulai dengan penggambaran terminator, aliran data, aliran kontrol penyimpanan, dan proses tunggal yang menunjukkan keseluruhan sistem. Bagian termudah adalah menetapkan proses (yang hanya terdiri dari satu lingkaran) dan diberi nama yang mewakili sistem. Nama dalam hal ini dapat menjelaskan proses atau pekerjaan atau dalam kasus ekstrim berupa nama perusahaan yang dalam hal ini mewakili proses yang dilakukan keseluruhan organisasi.

Terminator ditunjukkan dalam bentuk persegi panjang dan berkomunikasi langsung dengan sistem melalui aliran data atau penyimpanan eksternal. Antar terminator tidak diperbolehkan komunikasi langsung. Pada kenyataannya hubungan antar terminator dilakukan, tetapi secara definitif karena terminator adalah bagian dari lingkungan, maka tidak relevan jika dibahas dalam context diagram.

Tabel 1. Simbol-simbol Context Diagram

Simbol	Arti	Contoh
	Terminator	
	Aliran Data/ Data flow	Informasi mahasiswa baru 
 atau 	Proses/Process	



Gambar 6.1 Duplikasi terminator

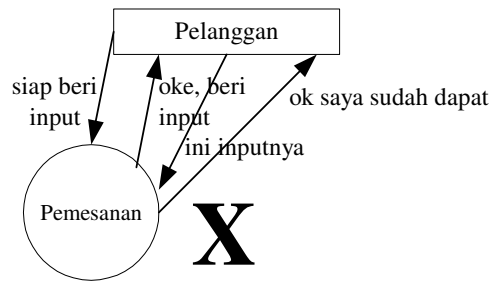
Context Diagram memiliki aturan sebagai berikut:

- a. Jika terdapat banyak 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 tersebut dapat berupa asterik (*) atau pagar (#).
- b. Jika terminator mewakili individu sebaiknya diwakili oleh peran yang dimainkan personil tersebut. Alasan pertama adalah personil yang berfungsi untuk melakukan itu dapat berganti sedang Context Diagram harus tetap akurat walaupun personil berganti. Alasan kedua adalah seorang personil dapat memainkan lebih dari satu peran.
- c. Karena fokus utama adalah mengembangkan model, maka penting untuk membedakan sumber (*resource*) dan pelaku (*handler*)., pelaku adalah mekanisme, perangkat atau media fisik yang mentransportasikan data ke/dari sistem, karena pelaku seringkali familier dengan pemakai dalam implementasi sistem berjalan, maka sering menonjol sebagai sesuatu yang harus digambarkan lebih dari sumber data itu sendiri. Sedangkan sistem baru dengan konsep pengembangan teknologinya membuat pelaku menjadi sesuatu yang tidak perlu digambarkan.

Aliran dalam context diagram memodelkan masukan 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 kita seharusnya menggambar context diagram dengan asumsi bahwa masukan disebabkan dan diawali oleh terminator, sedangkan keluaran disebabkan dan diawali oleh sistem.

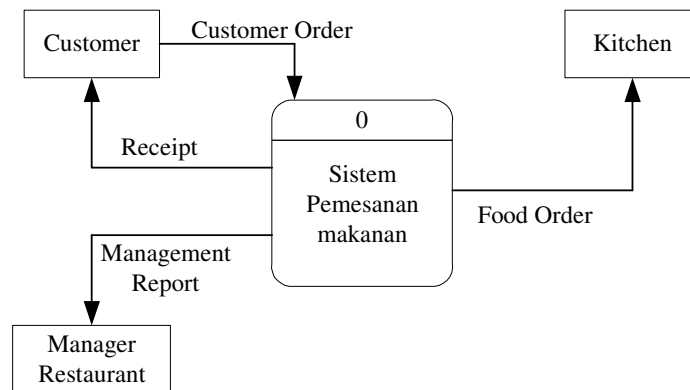
Dengan mencegah interaksi yang tidak perlu (*extraneous prompts*) yang berorientasi pada implementasi masukan-keluaran dan mengkonsentrasikan pemodelan pada jaringan aliran data.



Gambar 6.2 Interaksi yang tidak perlu

Kadang-kadang diperlukan dialog karena terminator tidak tahu sistem memerlukan masukan atau sistem tidak memberikan keluaran karena tidak tahu terminator membutuhkannya. Dalam hal ini interaksi menjadi diperlukan dan diasumsikan menjadi bagian esensi sistem sebagaimana gambar dibawah ini:

Contoh 6.3. sebuah Context Diagram untuk sistem pemesanan makanan ditunjukkan pada gambar di bawah ini.



Gambar 6.3. Diagram Konteks Sistem Pemesanan Makanan

3. DATA FLOW DIAGRAM

Empat simbol dasar yang digunakan untuk memetakan gerakan diagram aliran data adalah:

a. External Entity (Entitas)/terminator

Suatu entitas dapat disimbolkan dengan suatu notasi kotak sebagai berikut:

Kotak ini digunakan untuk menggambarkan suatu entitas eksternal (bagian lain, sebuah perusahaan, seseorang atau sebuah mesin) yang dapat mengirim data atau menerima data dari sistem. Entitas ini disebut juga sumber atau tujuan data, dan dianggap eksternal terhadap sistem yang sedang digambarkan. Setiap entitas diberi label dengan sebuah nama yang sesuai. Meskipun berinteraksi dengan sistem, namun dianggap di luar batas-batas sistem. Entitas-entitas tersebut harus diberi nama dengan suatu kata benda entitas yang sama bisa digunakan

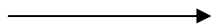
lebih dari sekali atas suatu diagram aliran data tertentu untuk menghindari persilangan antara jalur-jalur aliran data

Bentuk dari eksternal entity diantaranya adalah sebagai berikut:

- Suatu kantor, departemen atau divisi dalam perusahaan tetapi di luar sistem yang sedang dikembangkan.
- Orang/sekelompok orang di organisasi tetapi di luar sistem yang sedang dikembangkan
- Suatu organisasi atau orang yang berada di luar organisasi seperti misalnya langganan, pemasok, dll.
- Sistem informasi yang lain di luar sistem yang sedang dikembangkan
- Sumber asli dari suatu transaksi
- Penerima akhir dari suatu laporan yang dihasilkan oleh sistem

b. Data Flow/Arus data

Suatu arus data dapat disimbolkan dengan menggunakan suatu notasi tanda panah berikut .



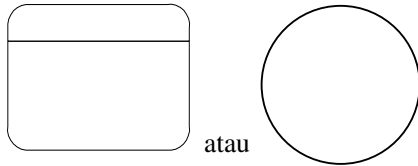
Tanda panah ini menunjukkan perpindahan data dari satu titik ke titik yang lain, dengan kepala tanda panah mengarah ke tujuan data. Karena sebuah tanda panah menunjukkan seseorang, tempat atau sesuatu, maka harus digambarkan dalam kata benda.

Bentuk dari arus data diantaranya adalah sebagai berikut:

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

c. Process/Proses

Bujur sangkar dengan sudut membulat/lingkaran digunakan untuk menunjukkan adanya proses transformasi. Proses-proses tersebut selalu menunjukkan suatu perubahan dalam data; jadi, aliran data yang meninggalkan suatu proses selalu diberi label yang berbeda dari aliran data yang masuk. Proses-proses yang menunjukkan hal itu di dalam sistem dan harus diberi nama menggunakan salah satu format berikut ini. Sebuah nama yang jelas memudahkan untuk memahami proses apa yang sedang dilakukan.

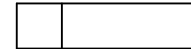


Pemberian nama pada proses:

1. Menetapkan nama sistem secara keseluruhan saat menamai proses pada level yang lebih tinggi. Contoh: *sistem kontrol inventaris*.
2. Menamai suatu subsistem utama, menggunakan nama-nama seperti : *Sistem pelaporan inventaris* atau *Sistem pelayanan konsumen internet*.
3. Menggunakan format kata kerja – kata sifat – kata benda untuk proses-proses yang mendetail. Kata kerja yang menggambarkan jenis kegiatan yang seperti ini, misalnya *menghitung, memverifikasi, menyiapkan, mencetak* atau *menambahkan*. Contoh-contoh nama proses yang lengkap adalah: *menghitung pajak penjualan, memverifikasi status rekening konsumen, menyiapkan invoice pengapalan, mencetak laporan yang diurutkan ke belakang, mengirim konfirmasi email ke konsumen, memverifikasi neraca kartu kredit dan menambah record inventaris*.

d. Data Store (Penyimpanan Data)

Data store dilambangkan dengan bujur sangkar dengan ujung terbuka.



yang menunjukkan penyimpanan data. Penyimpanan data menandakan penyimpanan manual, seperti lemari file/sebuah file/basisdata terkomputerisasi. Karena penyimpanan data mewakili seseorang, tempat atau sesuatu, maka diberi nama dengan sebuah kata benda. Penyimpanan data sementara seperti kertas catatan/sebuah file komputer sementara tidak dimasukkan ke dalam diagram aliran data.

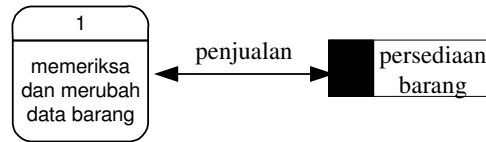
Bentuk dari penyimpanan data diantaranya adalah sebagai berikut:

- Suatu file atau database di sistem komputer
- Suatu arsip atau catatan manual
- Suatu kotak tempat data di meja seseorang
- Suatu tabel acuan manual
- Suatu agenda atau buku

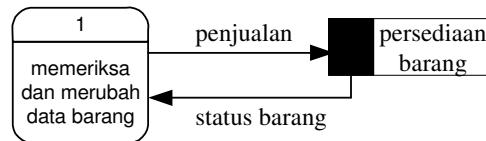
Dalam penggambaran penyimpanan data perlu diperhatikan beberapa hal, antara lain:

- a. Hanya proses saja yang berhubungan dengan simpanan data, karena yang menggunakan/merubah data di simpanan data adalah suatu proses.
- b. Arus data yang menuju ke simpanan data dari suatu proses menunjukkan proses update terhadap data yang tersimpan di simpanan data. Update dapat berupa proses:
 - Menambah/menyimpankan record baru atau dokumen baru ke dalam simpanan data.
 - Menghapus record atau mengambil dokumen dari simpanan data
 - Merubah nilai data di suatu record atau di suatu dokumen yang ada di simpanan data

- c. Arus data yang berasal dari simpanan data ke suatu proses menunjukkan bahwa proses tersebut menggunakan data yang ada di simpanan data.
- d. Untuk suatu proses yang melakukan kedua-duanya yaitu menggunakan dan update simpanan data dapat dipilih salah satu penggambaran.
 - Menggambarkan sebuah garis dengan panah mengarah kedua arah yang berlawanan dari simpanan data sebagai berikut:



- 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/kesatuan luar dapat digambar lebih dari sebuah.

4. MENGGAMBAR DIAGRAM 0 (LEVEL BERIKUTNYA)

Diagram 0 adalah pengembangan dari diagram konteks dan bisa mencakup sampai sembilan proses. Setiap proses diberi nomor bilangan bulat, umumnya dimulai dari sudut sebelah kiri atas diagram dan mengarah ke sudut sebelah kanan bawah. Penyimpanan data-penyimpanan data utama dari sistem (mewakili file-file master) dan semua entitas eksternal dimasukkan ke dalam diagram 0.

5. MENCIPTAKAN DIAGRAM ANAK (TINGKAT YANG LEBIH MENDETAIL)

Setiap proses dalam diagram 0 bisa dikembangkan untuk menciptakan diagram anak yang lebih mendetail. Proses oada diagram 0 yang dikembangkan itu disebut *parent process* (proses induk) dan diagram yang dihasilkan disebut *child diagram* (diagram anak). Aturan utaa untk menciptakan diagram anak adalah keseimbangan vertikal, dimana menyatakan bahwa suatu diagram anak tidak bisa menghasilkan keluaran/menerima masukan dimana proses induknya juga tidak menghasilkan atau menerima. Semua aliran data yang menuju atau keluar dari proses induk harus ditunjukkan mengalir ke dalam atau keluar dari diagram anak.

Entitas-entitas biasanya tidak ditunjukkan dalam diagram anak di bawah diagram 0. aliran data yang menyesuaikan aliran induknya disebut aliran data antarmuka dan ditunjukkan sebagai anak panah dari dan menuju area kosong dalam diagram anak. Bila proses indk memiliki aliran daa yang terhubung ke penyimpanan data, diagram anak bisa memasukkan penyimpanan data tersebut. Selain itu, diagram pada level yang lebih rendah ini bisa memasukkan penyimpanan data-penyimpanan data yang tidak ditunjukkan dalam proses induk. Sebagai contoh, sebuah file yang berisikan suatu tabel informasi, seperti tabel pajak atau file yang menghubungkan dua proses pada

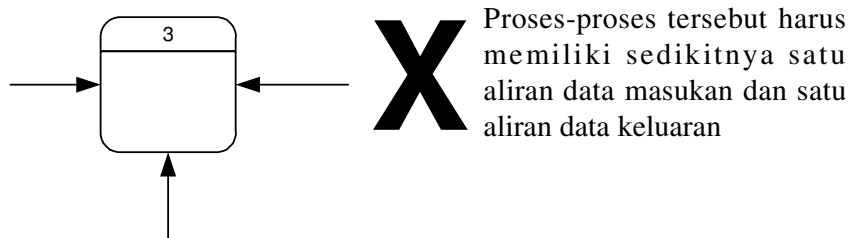
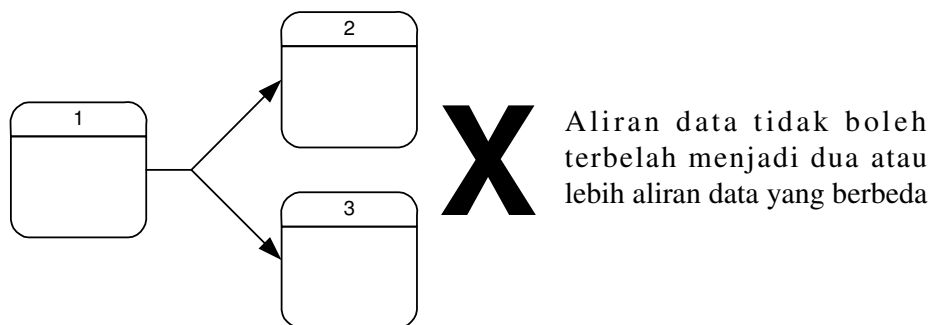
diagram anak bisa dimasukkan. Aliran data minor, seperti jalur kesalahan, bisa dimasukkan pada diagram anak dan bukan pada diagram induk.

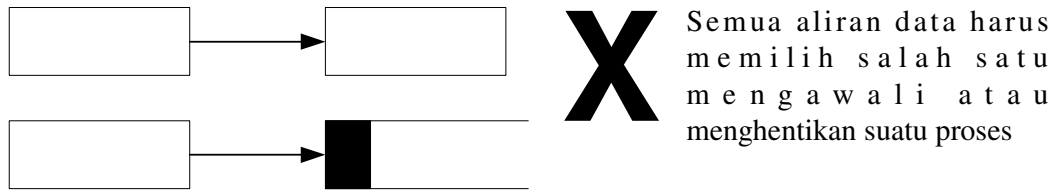
6. MENGECEK KESALAHAN DIAGRAM

Sejumlah kesalahan bisa terjadi saat menggambar diagram aliran data. Beberapa kesalahan yang umumnya terjadi adalah:

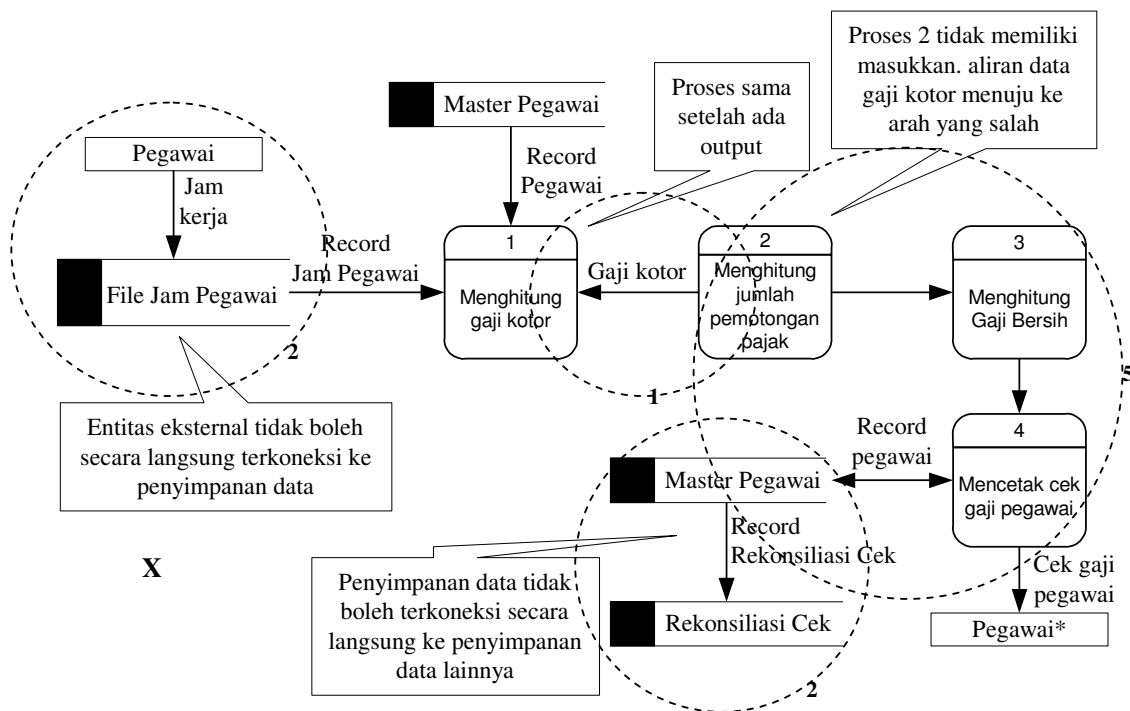
1. Lupa memasukkan suatu aliran data/mengarahkan kepala anak panah pada arah yang salah. Contohnya adalah sebuah proses gambaran yang menunjukkan semua aliran data sebagai masukan atau sebagai keluaran saja. setiap proses mentransformasikan data dan harus menerima dan menghasilkan keluaran.
2. Menghubungkan penyimpanan data dan entitas-entitas eksternal secara langsung satu sama lain. Penyimpanan data-penyimpanan data serta entitas juga tidak perlu dikonesikan satu sama lain; penyimpanan data dan entitas eksternal hanya terhubung dengan suatu proses. Suatu file tidak ditampilkan dengan file yang lain tanpa bantuan suatu program atau seseorang untuk memindahkan data.

Dua entitas eksternal yang terkoneksi secara langsung menunjukkan bahwa mereka ingin berkomunikasi satu sama lain. Koneksi ini tidak termasuk dalam diagram aliran data kecuali bila sistem memfasilitasi komunikasi tersebut. Menghasilkan sebuah laporan adalah contoh dari jenis komunikasi ini. sebuah proses masih harus ditempatkan di antara entitas sehingga menghasilkan laporan.





- Aliran data-aliran data atau proses-proses pemberian label yang tidak tepat. Periksalah diagram aliran data tersebut untuk memastikan bahwa setiap objek atau aliran data diberi label yang sesuai. Sebuah proses harus menunjukkan nama sistem atau menggunakan format kata kerja – kata sifat – kata benda. Masing-masing aliran data harus bisa digambarkan dengan sebuah kata benda

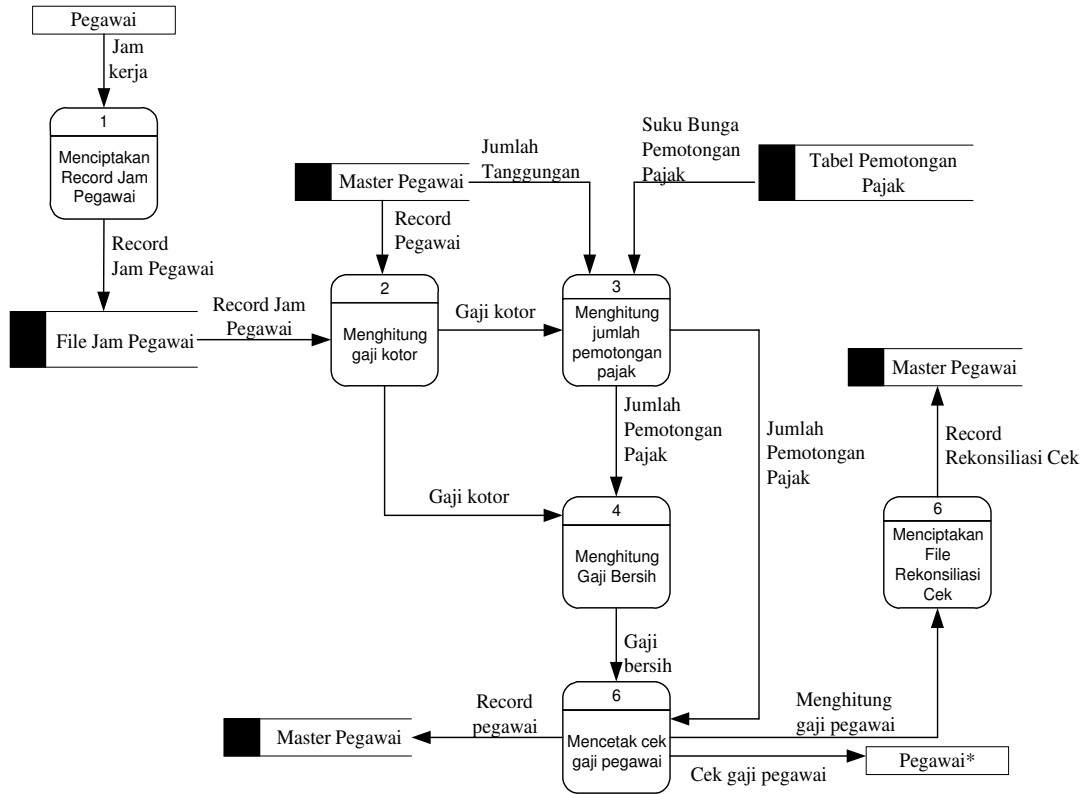


- Memasukkan lebih dari sembilan proses pada diagram aliran data. Memiliki terlalu banyak proses yang menciptakan suatu diagram yang kacau akan memusingkan untuk dibaca dan malah menghalangi komunikasi. Bila melibatkan lebih dari sembilan proses dalam suatu sistem kelompokkan beberapa proses yang bekerja bersama-sama di dalam suatu subsistem dan letakkan mereka pada suatu diagram anak.
- Mengabaikan aliran data. Aliran data pada anak sangat mendetail, sehingga aliran data linier sangat jarang ditemukan. Contohnya dari gambar di bawah ini, pada proses menghitung Jumlah Pemotongan Pajak memerlukan Jumlah Tanggungan yang dimiliki

pegawai dan Suku Bunga Pemotongan Pajak sebagai masukan Sama halnya dengan proses Gaji Bersih

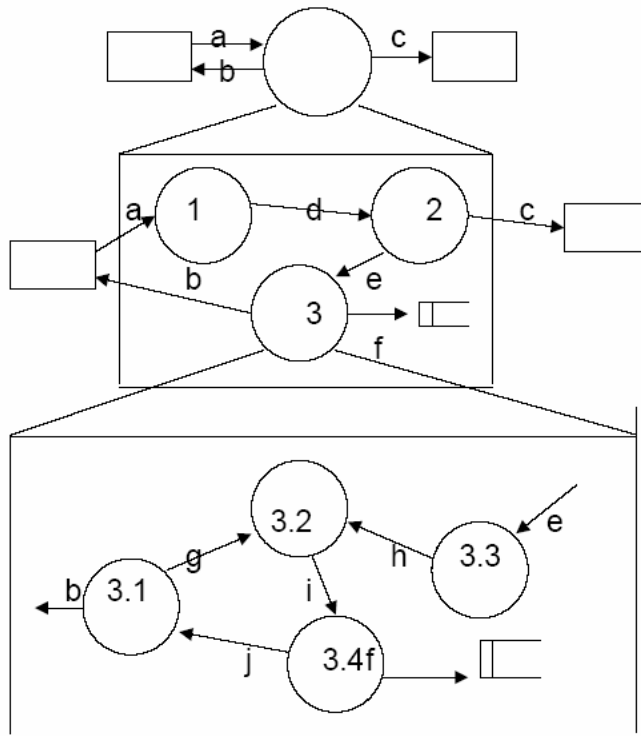
- Menciptakan analisis yang tidak seimbang. Masing-masing diagram anak harus memiliki masukan dan aliran data keluaran yang sama seperti proses induk. Pengecualiannya adalah keluaran minor, seperti jalur-jalur kesalahan yang hanya dimasukkan pada diagram anak.

Contoh DFD yang benar dari suatu proses pembayaran gaji



Gambar 6.4 Contoh DFD level 0

Gambar Pembuatan CD dan DFD Level



PENGAMBARAN DFD

Tidak ada aturan baku untuk menggambarkan DFD, tapi dari berbagai referensi yg ada, secara garis besar:

1. Buat diagram context

Diagram ini adalah diagram level tertinggi dari DFD yg menggambarkan hubungan sistem dengan lingkungan luarnya. Cara :

- Tentukan nama sistemnya.
- Tentukan batasan sistemnya.
- Tentukan terminator apa saja yg ada dalam sistem.
- Tentukan apa yg diterima/diberikan terminator dari/pada sistem.
- Gambarkan diagram context.

2. Buat diagram level Zero

Diagram ini adalah dekomposisi dari diagram Context. Cara :

- Tentukan proses utama yg ada pada sistem.
- Tentukan apa yg diberikan/diterima masing-masing proses pada/dari sistem sambil memperhatikan konsep keseimbangan (alur data yang keluar/masuk dari suatu level harus sama dgn 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).

3. Buat diagram level Satu

Diagram ini merupakan dekomposisi dari diagram level zero. Cara :

- Tentukan proses yg lebih kecil (sub-proses) dari proses utama yg ada di level zero.
- Tentukan apa yg diberikan/diterima masing-masing sub-proses pada/dari sistem dan perhatikan konsep keseimbangan.
- Apabila diperlukan, munculkan data store (transaksi) sbg 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

4. DFD level dua, tiga, ..

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

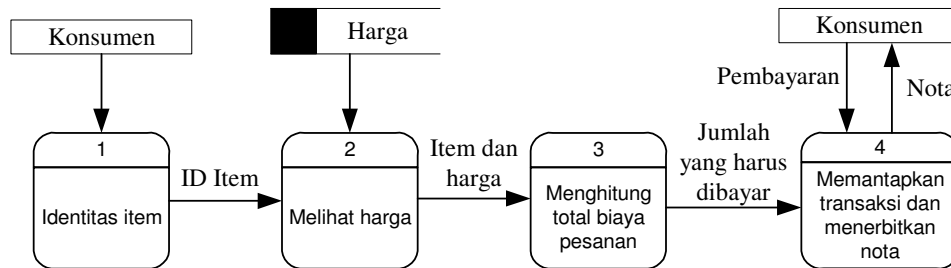
a. Diagram Aliran Data Logika dan Fisik

Diagram aliran data dikategorikan baik sebagai logika maupun fisik. Diagram aliran data logika memfokuskan pada bisnis serta bagaimana sistem tersebut beroperasi dan tidak berhubungan dengan bagaimana sistem tersebut dibangun. Melainkan, menggambarkan peristiwa-peristiwa bisnis yang dilakukan serta data-data yang diperlukan dan dihasilkan setiap peristiwa tersebut. Sebaliknya diagram aliran data fisik menunjukkan bagaimana sistem tersebut akan diimplementasikan, termasuk perangkat keras, perangkat lunak, file-file dan orang-orang yang terlibat dalam sistem tersebut.

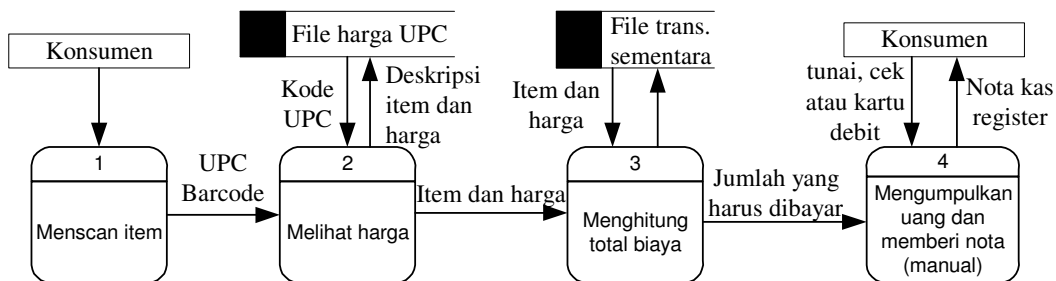
Fitur Desain	Logika	Fisik
Apa yang digambarkan model tersebut	Bagaimana bisnis tersebut beroperasi	Bagaimana sistem tersebut diimplementasikan atau bagaimana sistem yang ada beroperasi
Apa yang diwakili proses tersebut	Kegiatan-kegiatan bisnis	Program, modul-modul program dan prosedur-prosedur manual
Apa yang diwakili penyimpanan data tersebut	Pengumpulan data yang berhubungan dengan bagaimana data tersebut disimpan	Basisdata dan file-file fisik, file-file manual
Jenis penyimpanan data	Menunjukkan penyimpanan data-penyimpanan data yang mewakili pengumpulan data	File-file master, file-file transisi. Setiap proses yang beroperasi pada dua waktu yang berbeda harus dihubungkan dengan

	permanen	penyimpanan
Kontrol sistem	Menunjukkan kontrol-kontrol bisnis	Menunjukkan kontrol-kontrol untuk memvalidasi data-data masukkan agar memperoleh suatu record (record yang menemukan status) untuk memastikan penyelesaian suatu proses yang berhasil dan untuk keamanan sistem.

Contoh Diagram Aliran Data Logika



Contoh Diagram Aliran Data Fisik



b. Mengembangkan Diagram Aliran Data Logika

Keuntungan-keuntungan dengan menggunakan diagram aliran data logika:

- **Komunikasi yang lebih baik dengan pengguna**

Suatu model logika lebih mudah digunakan saat berkomunikasi dengan pengguna sistem karena dipusatkan pada kegiatan bisnis. Pengguna akan semakin mengenal kegiatan-kegiatan intinya serta beberapa syarat-syarat informasi dari setiap kegiatan.

- **Sistem yang lebih stabil**

Sistem yang dibentuk oleh diagram aliran data logika biasanya lebih stabil, hal ini dikarenakan diagram aliran data logika merepresentasikan fitur-fitur sistem yang harus ada tidak peduli apa yang dilakukan di dalam bisnis secara fisik.

- **Pemahaman yang lebih baik mengenai bisnis bagi penganalisis sistem**

Diagram aliran data logika memiliki suatu penekanan bisnis dan membantu penganalisa dalam memahami bisnis yang sedang dipelajari, memahami mengapa prosedur-prosedur

tertentu ditampilkan serta menentukan hasil-hasil yang diharapkan dari suatu tugas yang dilakukan.

- **Fleksibilitas dan pemeliharaan**

Sistem baru akan lebih fleksibel dan mudah mempertahankannya bila desainnya didasarkan atas suatu model logika. Fungsi-fungsi bisnis tidak terlalu sering mengalami perubahan. Aspek-aspek fisik dari sistem lebih sering berubah dibanding fungsi-fungsi bisnis.

- **Pengurangan redundansi dan kreasi yang lebih mudah mengenai model fisik**

Mengamati suatu model logika akan membantu anda menciptakan sistem yang lebih baik dengan mengurangi redundansi dan metode yang tidak efisien di dalam sistem lama. Selain itu, model logika memudahkan untuk menciptakan dan menyederhanakan penggunaan karena tidak terlalu sering memuat penyimpanan data dibandingkan dengan file master/basis data.

c. **Menciptakan Diagram Aliran Data Fisik**

Diagram aliran data fisik menunjukkan bagaimana sistem akan dikonstruksi.

Kelebihan-kelebihan dari diagram aliran data fisik:

- Mengklarifikasikan proses-proses mana yang manual dan mana yang otomatis
- Menggambarkan proses-proses secara lebih mendetail dibanding DAD Logika
- Mengurutkan proses-proses yang harus dilakukan menurut urutan tertentu
- Mengidentifikasi penyimpanan data sementara
- Menetapkan nama-nama aktual dari file-file dan printoutnya
- Menambahkan kontrol-kontrol untuk memastikan bahwa proses-proses tersebut sudah dilakukan secara tepat.

Diagram aliran data fisik yang memuat beberapa item yang tidak ditemukan dalam diagram aliran data logika.

- Proses-proses manual
- Proses –proses untuk penambahan, penghapusan, pengubahan, dan perbaharuan record
- Proses-proses masukkan data dan verifikasi
- Proses-proses validasi untuk memastikan keakuratan masukkan data
- Proses-proses pengurutan untuk mengatur kembali urutan record
- Proses-proses untuk memproduksi setiap keluaran sistem yang unik melanjutkan penyimpanan data
- Nama-nama file aktual yang digunakan untuk menyimpan data
- Kontrol untuk menandai selesainya tugas atau kondisi-kondisi kesalahan.