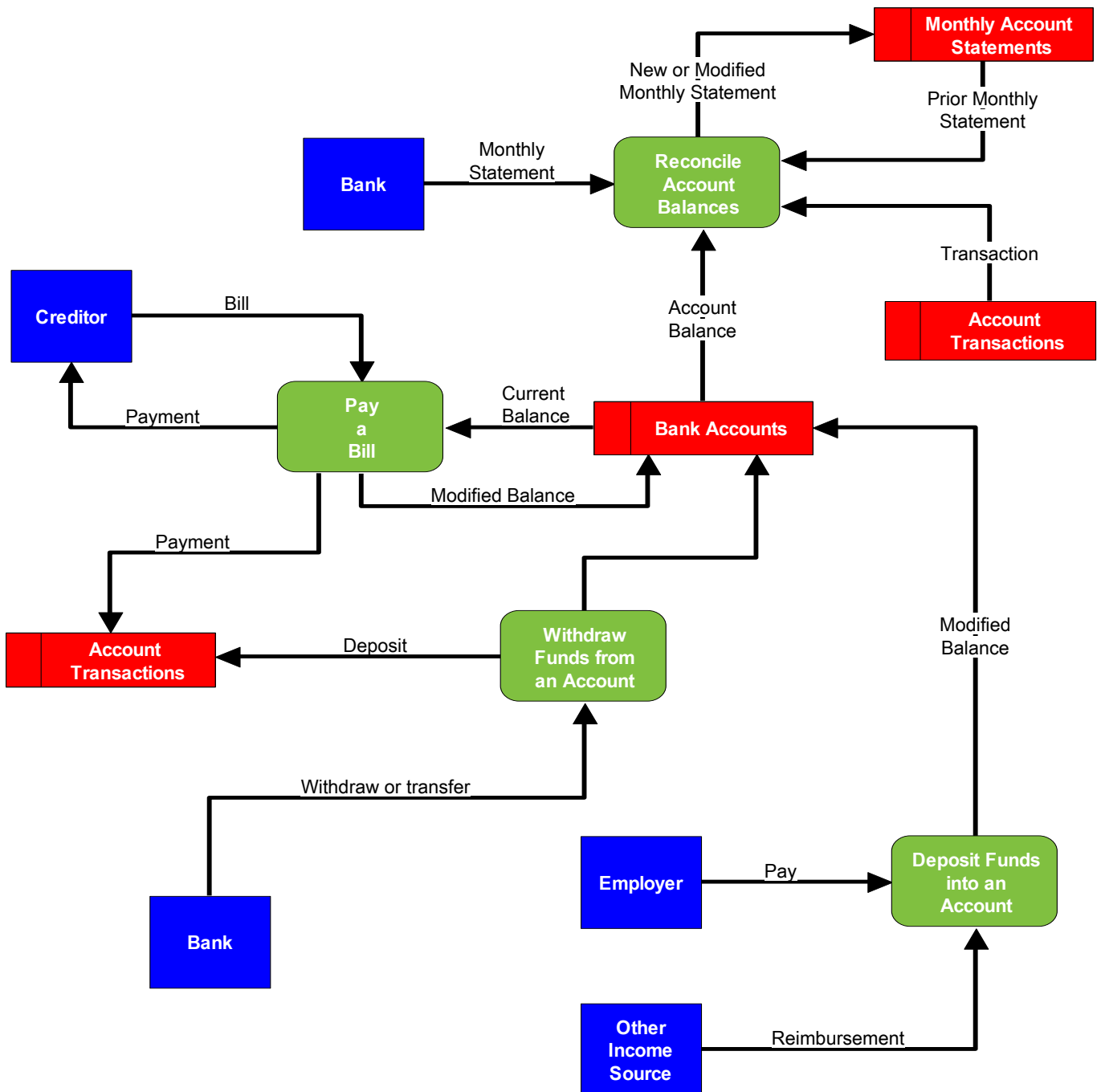


PEMODELAN PROSES

1. Pengertian

Permodelan Proses adalah suatu teknik menyusun dan mendokumentasikan struktur dan aliran data yang melalui proses sistem dan/atau logika, polisi dan prosedur yang dilaksanakan oleh proses sistem. Model proses analisis sistem terdiri dari *Diagram Aliran Data* (Data Flow Diagram) atau bubble chart, transformation graph, model proses.



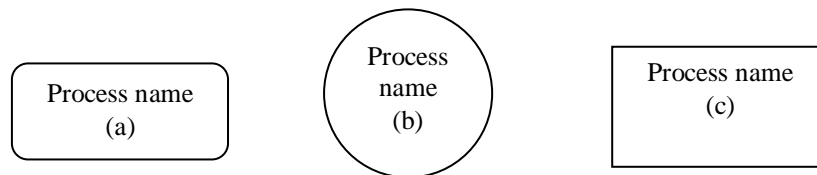
Gambar 1: A Simple Data Flow Diagram

2. Konsep Permodelan Proses

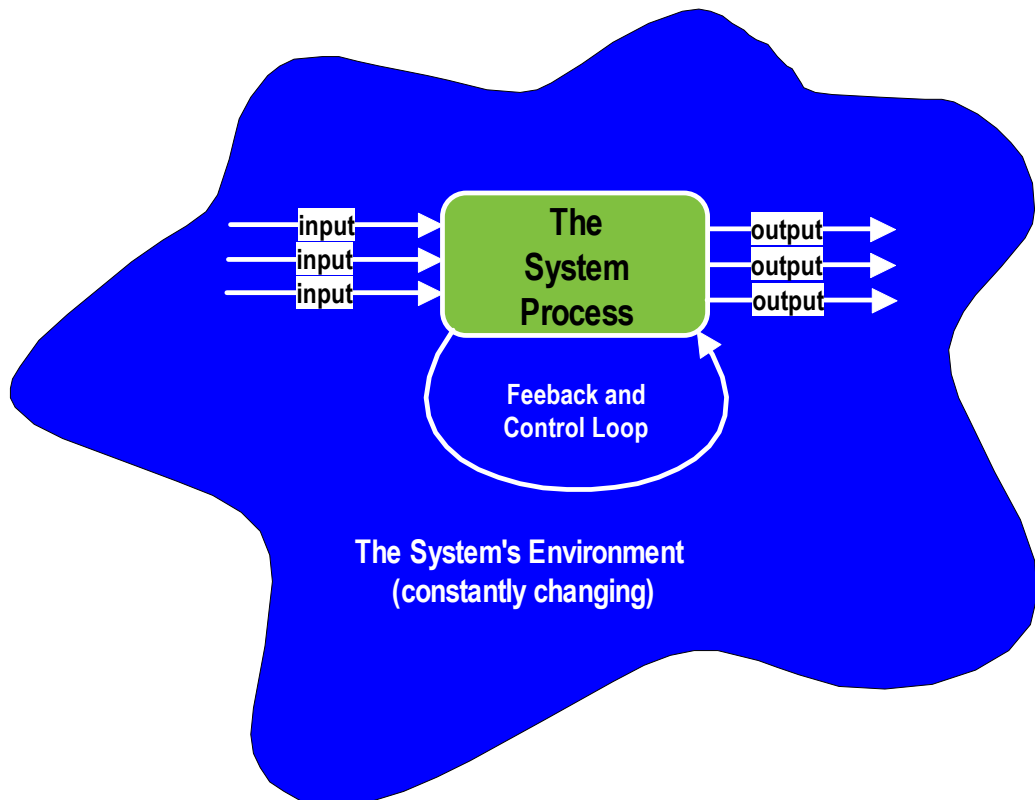
Proses atau *transform* adalah suatu kerja yang dilaksanakan pada, atau unpan balik dengan aliran data atau suatu keadaan. Model proses sistem yang paling mudah terdiri dari input, output dan sistem itu sendiri. Simbol proses mendefinisikan batasan sesuatu sistem. Sistem membuat pertukaran input dan output dengan lingkungannya.

Simbol Proses :

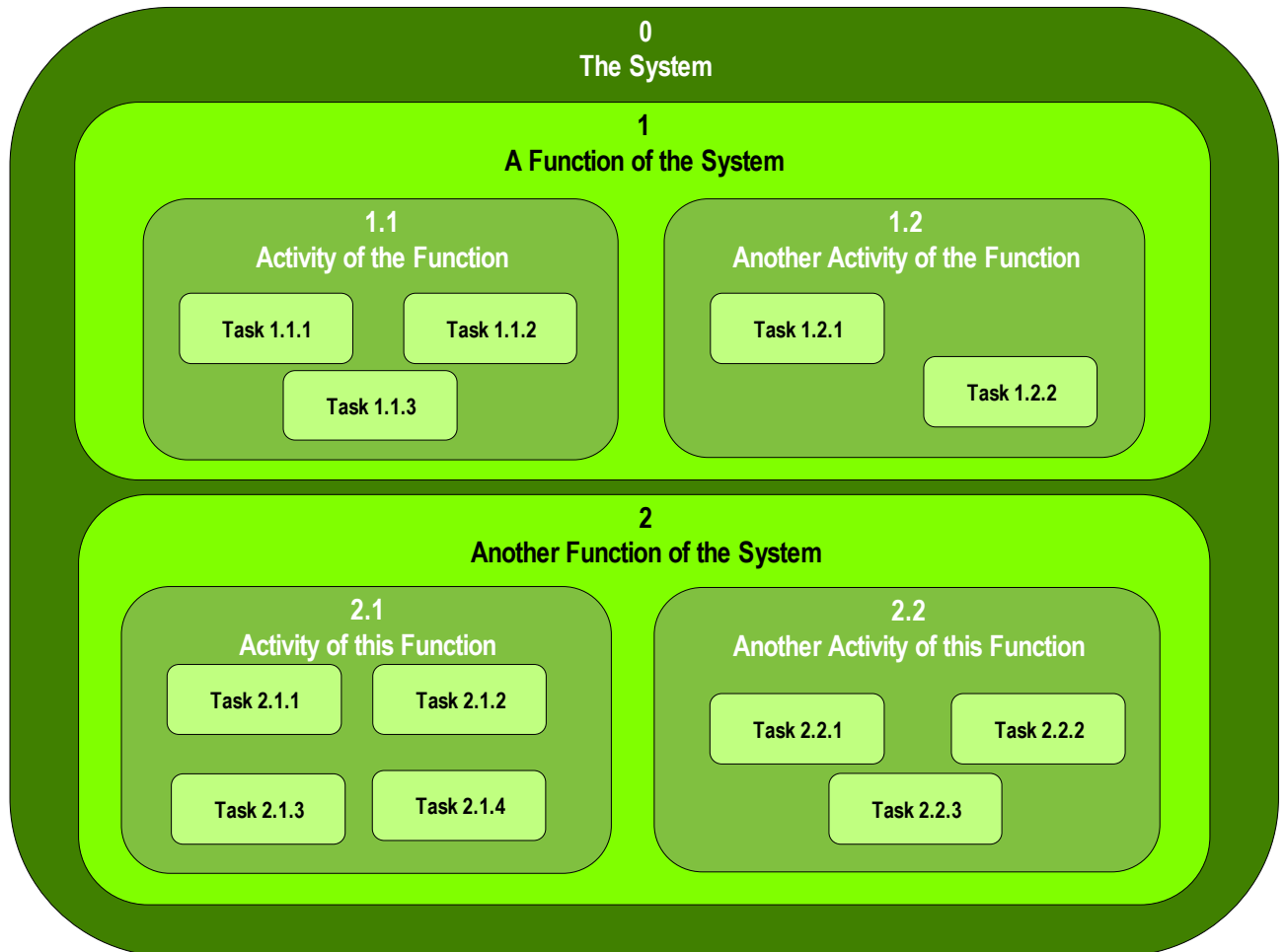
- (a) Rounded rectangle (notasi Gane and Sarson).
- (b) Bulatan (notasi Demarco/Yourdon).
- (c) Segiempat (notasi SSADM/IDEF0).



Gambar 2: Simbol Proses



Gambar 3: The Classical Process Model of a System



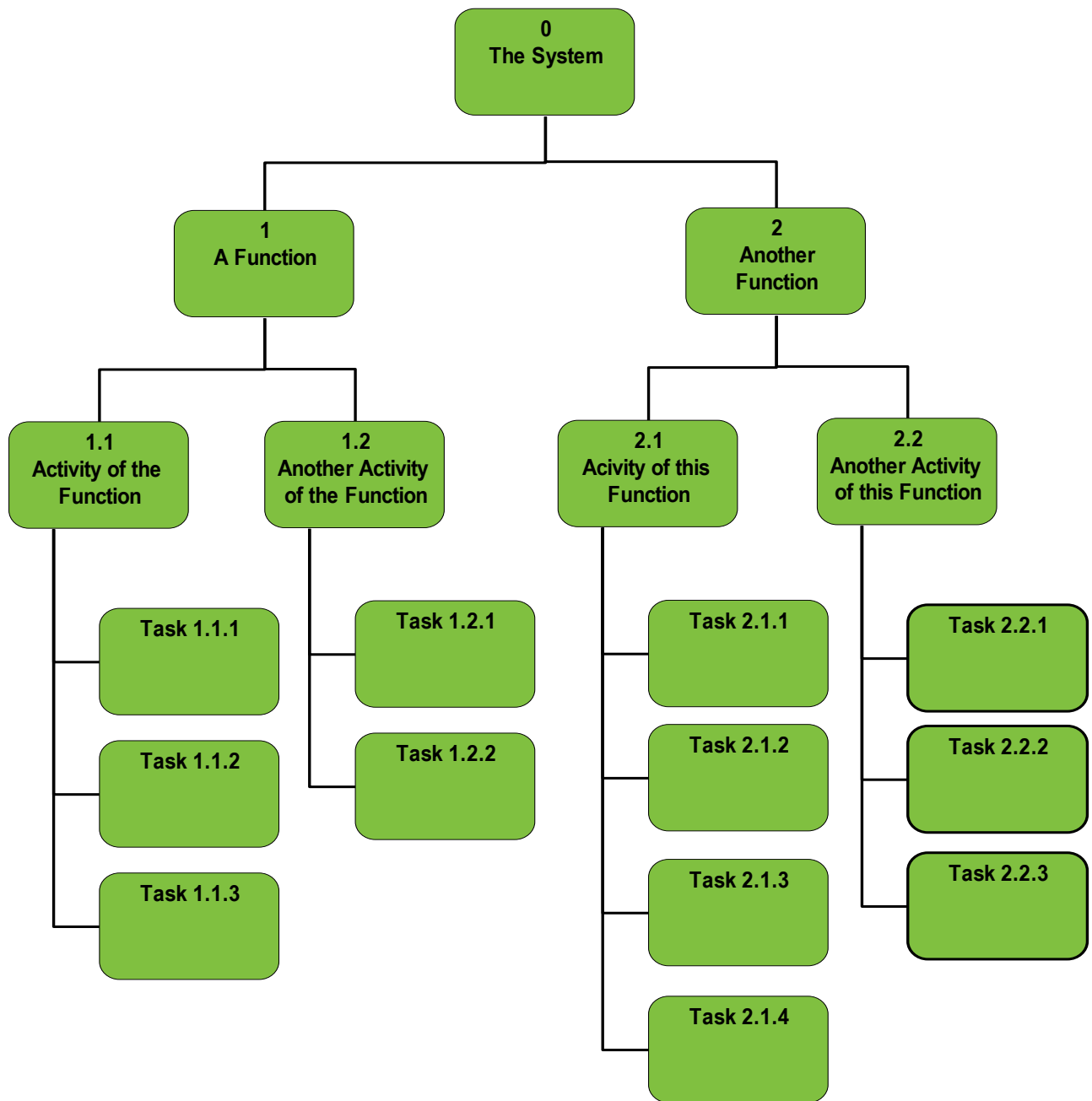
Gambar 4: A System Consists of Many Subsystems & Processes

2.1 Dekomposisi Proses

Dikomposisi adalah suatu teknik pemecahan sistem kepada subsistem, proses dan subproses. Setiap tingkat pengabstrakan menunjukkan perincian yang lebih tinggi/rendah tentang keseluruhan sistem atau subset sistem tersebut. Sedangkan decomposition diagram/*hierarchy chart* menunjukkan dekomposisi atas bawah dan struktur sebuah sistem.

Peraturan decomposition diagram:

- Setiap proses merupakan *parent process*, *child process* (of a parent), atau kedua-uanya.
- Parent mesti mempunyai dua atau lebih anak (anak tunggal tidak menunjukkan perincian tambahan mengenai sistem).
- Anak mempunyai satu parent sahaja.



Gambar 5: A Decomposition Diagram

2.2 Proses logikal dan peraturan

Proses Logikal adalah suatu kerja/tindakan yang mesti dilakukan tanpa melihat cara melakukannya. Setiap proses logikal dilaksanakan/akan sebagai satu atau lebih proses fisik yang terdiri dari:

- kerja yang dilakukan oleh manusia

- kerja yang dilakukan oleh robot atau mesin
- kerja yang dilakukan oleh program komputer

Terdapat 3 jenis proses logikal yaitu: fungsi, peristiwa dan proses dasar.

- Fungsi : sekumpulan aktivitas bisnis yang berkaitan dan terus menerus. Fungsi tidak mempunyai awal dan akhir karena fungsi dilaksanakan terus menerus selama fungsi tersebut diperlukan.
 - Setiap fungsi mungkin terdiri dari banyak lagi proses lain yang perlu dilakukan untuk menunjang aktivitas/tugas tertentu.
 - Fungsi mengumpulkan aktivitas-aktivitas dan tugas-tugas yang berkaitan.
 - Fungsi dinamakan dengan kata nama untuk menggambarkan keseluruhan fungsi.
- Peristiwa: unit logikal kerja yang mesti dilengkapkan secara keseluruhan. Peristiwa dimulai dengan berbagai input dan diakhiri setelah proses unpan balik dengan output tertentu. Peristiwa kadang-kadang dipanggil transaksi. Proses peristiwa bisa dibagi lagi kepada proses-proses dasar yang menggambarkan secara terperinci bagaimana sistem perlu bertindak balas dengan peristiwa.
 - Fungsi terdiri dari proses-proses yang berunpan balik dengan peristiwa.
 - Setiap peristiwa mempunyai pelopor dan unpan balik yang bisa didefinisikan input dan outputnya.
 - Fungsi-fungsi sistem dibagi kepada peristiwa bisnis.
 - Setiap peristiwa bisnis diwakili oleh satu proses yang akan berunpan balik dengan peristiwa tersebut.
- Proses asas : beberapa aktivitas/tugas terperinci yang diperlukan untuk melengkapkan unpan balik kepada sesuatu peristiwa. Kedudukannya berada di tingkat perincian terbawah dalam model proses.

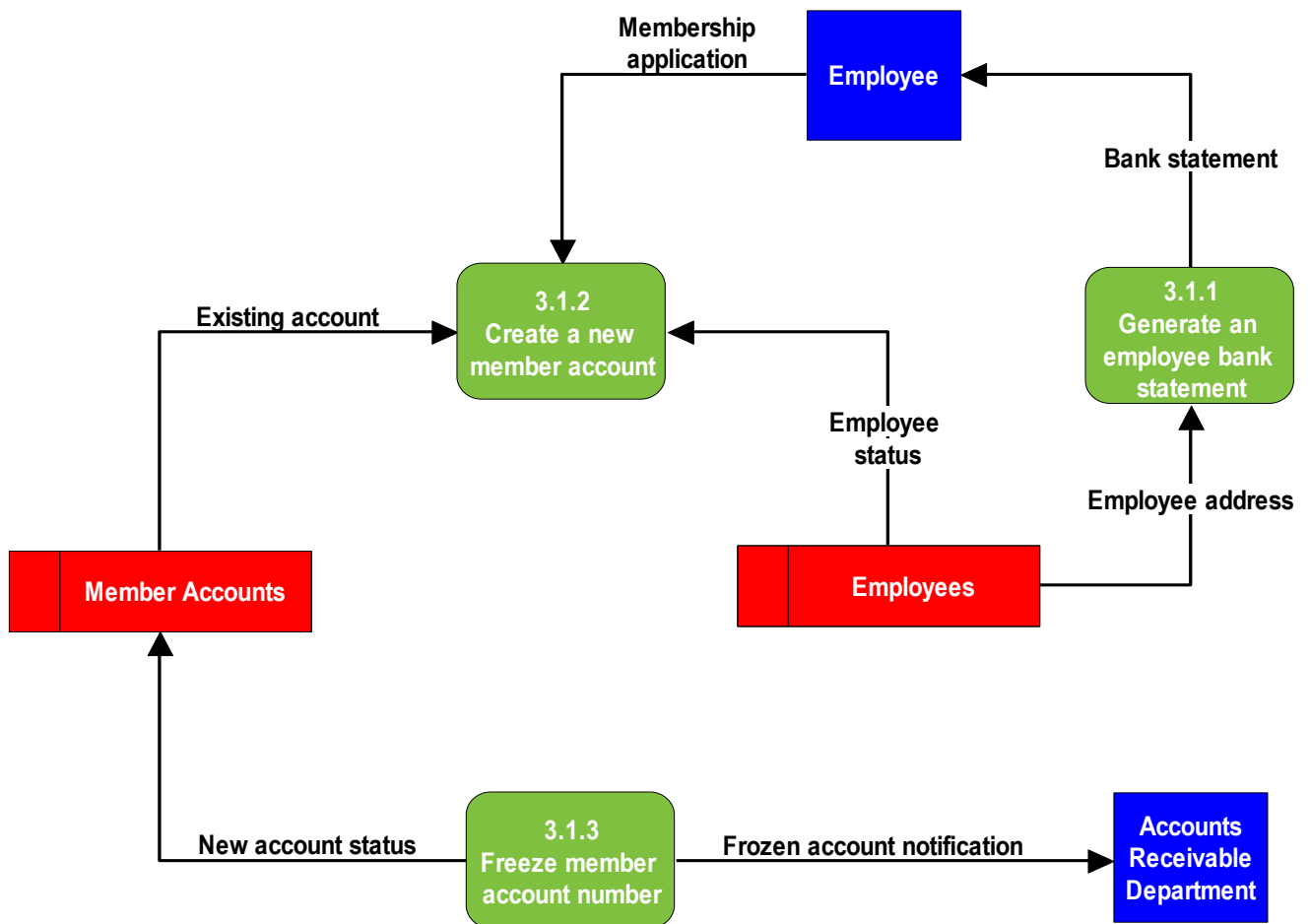
Model proses logikal tidak meliputi proses yang hanya memindahkan data (data tidak berubah). Proses logikal yang perlu dimodelkan :

- Membuat perhitungan (hitung CGPA)
- Membuat keputusan (menentukan kewujudan produk yang ditempah)

- Atur, saring dan ringkaskan data (mengetahui invois yang lewat)
- Mengelola data menjadi informasi yang berguna (buat laporan atau jawab pertanyaan)
- Menciptakan proses lain (menghidupkan mesin, mengarahkan robot)
- Menggunakan data yang disimpan (cipta, baca, rubah atau hapus data)

Hindari kesalahan-kesalahan mekanikal, seperti berikut:

- black hole : proses ada input tetapi tidak ada output. Data memasuki proses kemudian hilang.
- miracle : proses ada output tetapi tidak ada input.
- gray hole : inputs proses x mencukupi untuk menghasilkan output (paling biasa)



Gambar 6 : Common Errors on Data Flow Diagrams

2.3 Logika Proses

- Gunakan perintah terperinci dalam proses dasar (dalam DFD) untuk menunjukkan logika di dalam proses tersebut. Perintah dengan menggunakan bahasa dan sintaks yang berdasarkan kekuatan pemrograman berstruktur dan bahasa untuk menunjukkan logika proses dasar dalam model proses (seperti DFD).
- Struktur keseluruhan spesifikasi bahasa dibuat dengan syarat-syarat sebagai berikut:
 - Kalimat mudah dan berbentuk penjelasan terhadap suatu perkara.
 - Struktur bersyarat/keputusan yang menandakan bahwa proses mesti melaksanakan tindakan yang berlainan mengikuti syarat yang spesifik.
 - Struktur pengulangan menentukan bahwa sekelompok tindakan perlu diulang berdasarkan syarat yang telah ditetapkan.
 - Kalimat majemuk tidak dianjurkan karena menyebabkan kekeliruan.
 - Setiap kalimat menggunakan kata kerja yang jelas, seperti: cari, simpan, bangun, baca, baiki, hapus, hitung, tulis, atur.
 - formula bisa dimasukkan dalam kalimat (seperti: $\text{calculate gross pay} = \text{hours worked} \times \text{hourly wage.}$)
- Terdapat dua jenis pembuatan struktur keputusan atau bersyarat:
 - IF-THEN-ELSE & CASE
 - Bagi logika yang menggunakan banyak syarat dan gabungan syarat, gunakan *decision tables*.
- Terdapat dua jenis pembuatan pengulangan:
 - DO-WHILE
 - REPEAT-UNTIL
- Kebanyakan proses mempunyai peraturan yang dipanggil sebagai kebijakan bisnis.

A SIMPLE POLICY STATEMENT

CHECK CASHING IDENTIFICATION CARD
<p>A customer with check cashing privileges is entitled to cash personal checks of up to \$75.00 and payroll checks of from companies pre-approved by <i>LMART</i>. This card is issued in accordance with the terms and conditions of the application and is subject to change without notice. This card is the property of <i>LMART</i> and shall be forfeited upon request of <i>LMART</i>.</p>
<p>SIGNATURE <i>Charles C. Parker,</i> <i>Jr.</i></p>
<p style="color: red; font-weight: bold;">EXPIRES May 31, 1998</p>

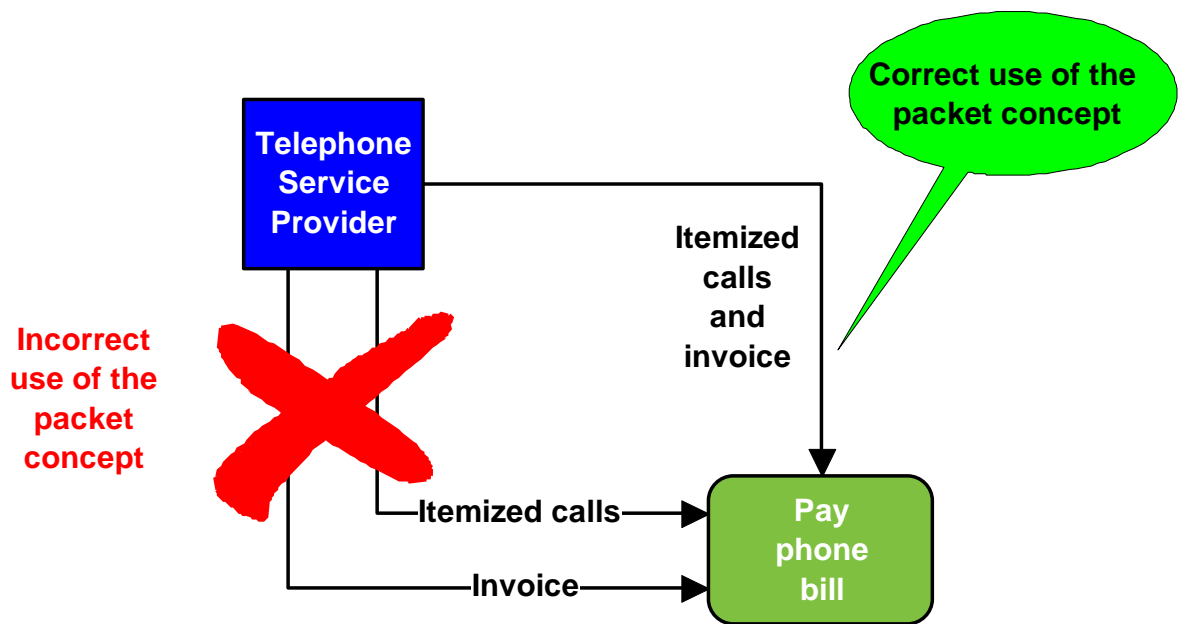
THE EQUIVALENT POLICY DECISION TABLE

Conditions and Actions	Rule 1	Rule 2	Rule 3	Rule 5
C1: Type of check	personal	payroll	personal	payroll
C2: Check amount less than or equal to \$75.00	yes	doesn't matter	no	doesn't matter
C3: Company accredited by <i>LMART</i>	doesn't matter	yes	doesn't matter	no
A1: Cash the check	X	X		
A2: Don't cash the check			X	X

Tabel 1: a sample Decision Table

2.4 Aliran Data

Aliran data merupakan pergerakan data dalam sistem. Aliran data di antara sistem dan lingkungannya, atau di antara dua proses dalam sistem merupakan komunikasi di antara data. Aliran data mewakili data input terhadap proses atau data output (informasi) dari proses. Aliran data juga mewakili penambahan, penghapusan atau perbaikan data dlm fail/pangkalan data (storan data dlm DFD). Kesemua aliran data tersebut diwakili oleh garisan dengan anak panah.

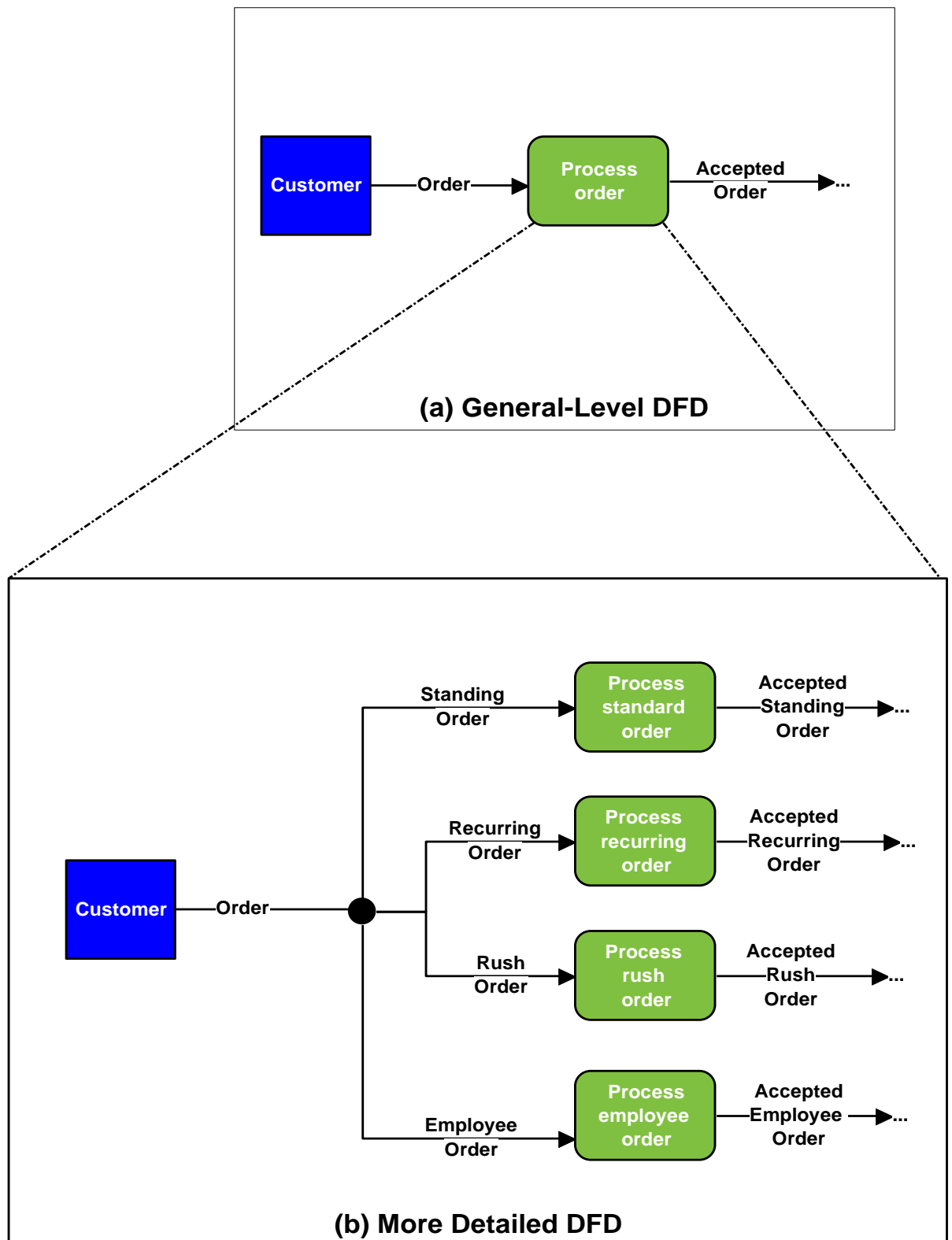


Gambar 7: The Data Flow Packet Concept

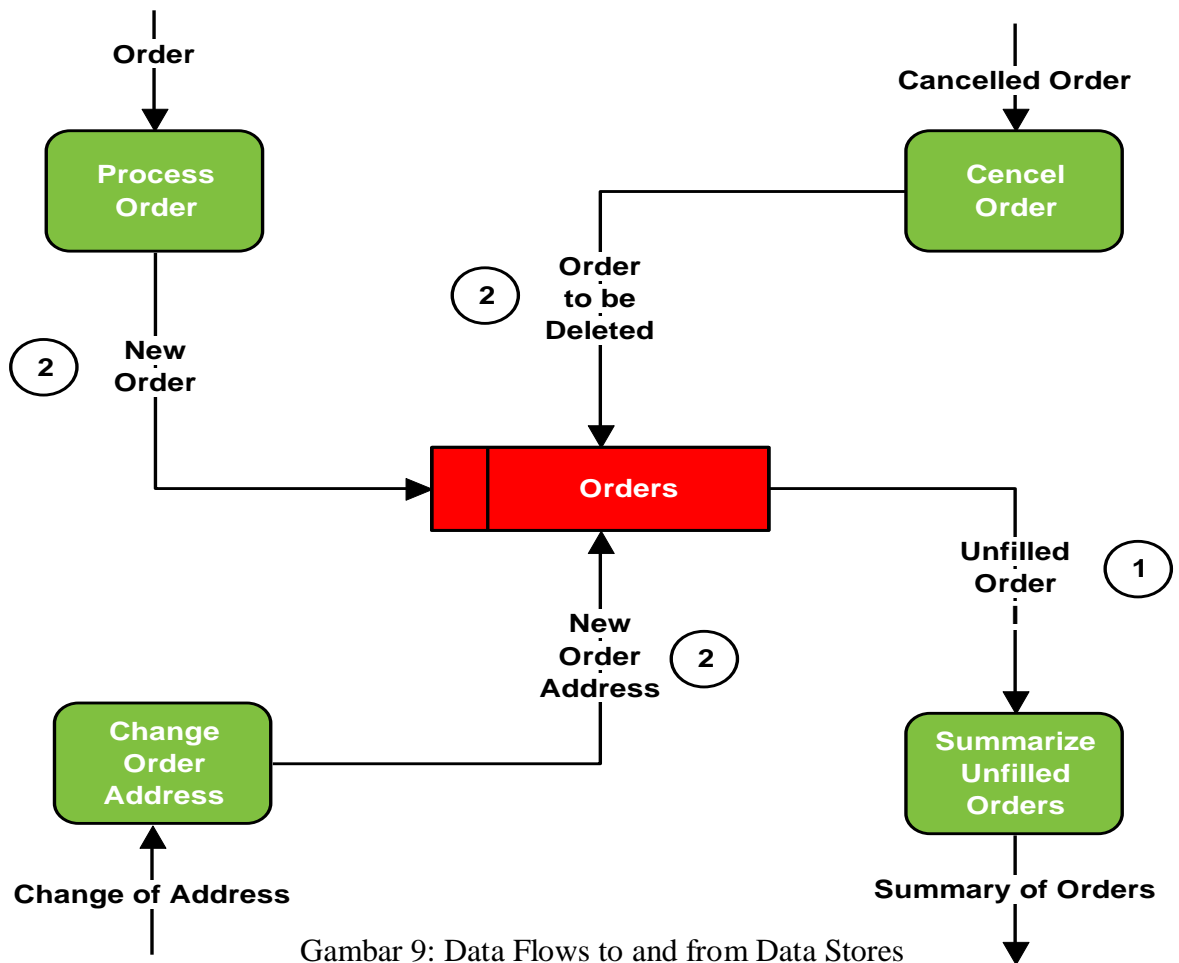
Aliran data dibuat dari atribut data yang benar-benar data dalam sistem (dipanggil struktur data) atau aliran data lain.

- Aliran data komposit : aliran data yang memiliki aliran-aliran data yang digunakan untuk menggabungkan aliran data yang sama pada DFD tingkat umum supaya aliran mudah difahami.
- Aliran kawalan : aliran disebut juga aliran bukan-data yang mewakili syarat/peristiwa bukan data yang melahirkan proses. Seperti: keadaan yang perlu dokontrol sewaktu proses sistem berfungsi. Apabila sistem menyadari sesuatu syarat dipenuhi, proses yg menggunakannya sebagai input dimulai. Biasanya digambarkan dengan menggunakan garisan putus-putus dengan anak panah

Aliran data perlu diberi nama atau label. Nama aliran data: biasanya dibuat dengan menggunakan nama yang unik. Menggunakan kata kerja/kata nama untuk menerangkan bagaimana pemrosesan mengubah aliran data. Semua aliran data mesti dimulai atau diakhiri pada satu proses karena aliran data merupakan suatu input dan output proses.



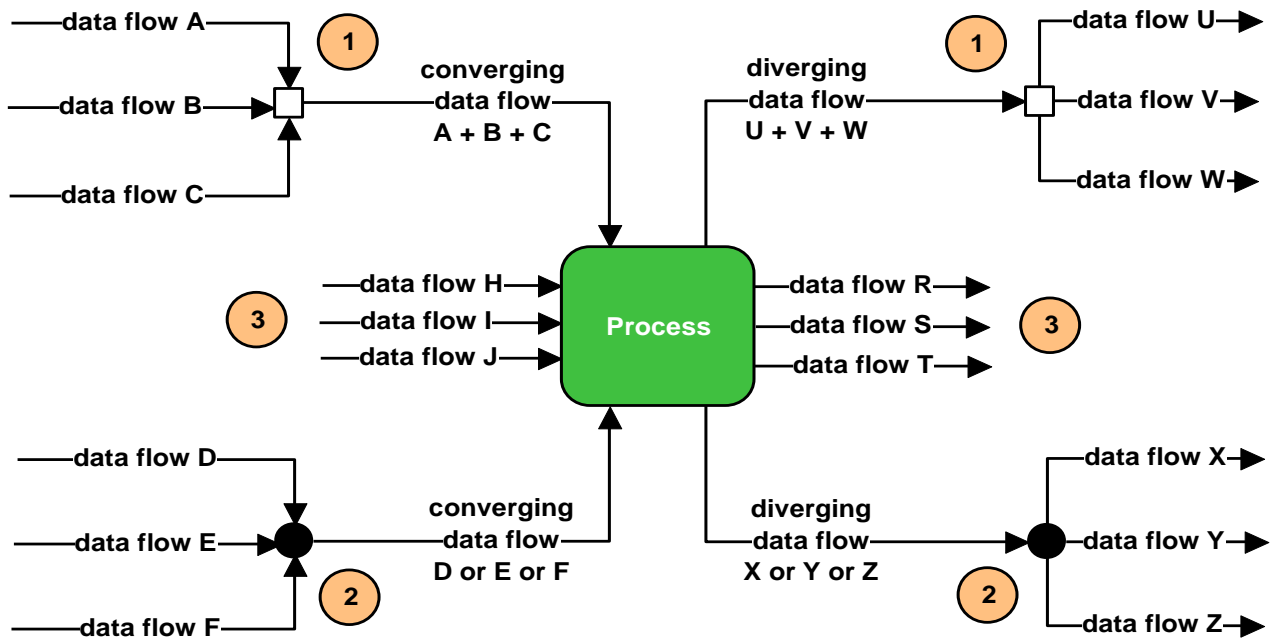
Gambar 8: Composite & Elementary Data Flows



Gambar 9: Data Flows to and from Data Stores

Aliran penghantaran data bisa dilakukan kepada beberapa aliran data, yaitu:

- Diverging data flow : penghantaran aliran data semua atau sebagian aliran data tunggal dialirkan ke tempat yang berlainan.
- Converging data flow: mencantumkan beberapa aliran data ke dalam satu aliran data. Aliran data dari berbagai sumber bisa digabungkan sebagai satu paket untuk pemrosesan seterusnya.



Gambar 9: Diverging & Converging Data Flows

2.5 Agen Luar

Sistem informasi berinteraksi dengan peristiwa/situasi yang ada di lingkungan yang terdiri dari agen luar yang membentuk batas sistem dan mendefinisikan tempat di mana sistem berinteraksi dengan lingkungan. Agen luaran atau entiti luaran mendefinisikan individu, unit organisasi, sistem/ organisasi lain yang berada di luar ruang lingkup proyek, tetapi berinteraksi dengan sistem yang sedang dikaji. Agen luaran memberi input kepada sistem dan menerima output dari sistem.

2.6 Storan Data

Sistem informasi menyimpan data untuk digunakan kemudian, data tersebut disimpan dalam *storan data* (inventori data, fail & pangkalan data). Storan data diwakili oleh open-end box. Jika aliran data adalah data yang sedang bergerak, storan data adalah data yang sedang beristirahat. Biasanya setiap entiti dalam *End-Relationship diagram* (ERD) akan mengadakan satu storan data. Hindarkan istilah fisikal seperti *fail*, *pangkalan data*, *kabinet fail*, *file folder*, dll.

3. Proses Permodelan Proses Logikal

3.1 Perancangan Sistem Strategik

Perancangan strategik menghasilkan perancangan strategik sistem informasi (ISSP). ISSP mendefinisikan arkitektur sistem informasi yang biasanya meliputi model proses enterprise yang mengetahui kontrol bisnes dan fungsi saja. ISSP ditunjukkan oleh decomposition diagram/DFD tingkat tinggi. Disimpan dalam corporate repository.

3.2 Permodelan Proses bagi Business Process Redesign (BPR)

Proyek BPR projects menganalisis proses bisnes dan mendesain kembali untuk menghapuskan ketidakcekapan dalam aplikasi teknologi informasi pada saat ini. Untuk menganalisis proses bisnes, proses pada saat ini mesti dikaji dan didokumenkan menggunakan model proses.

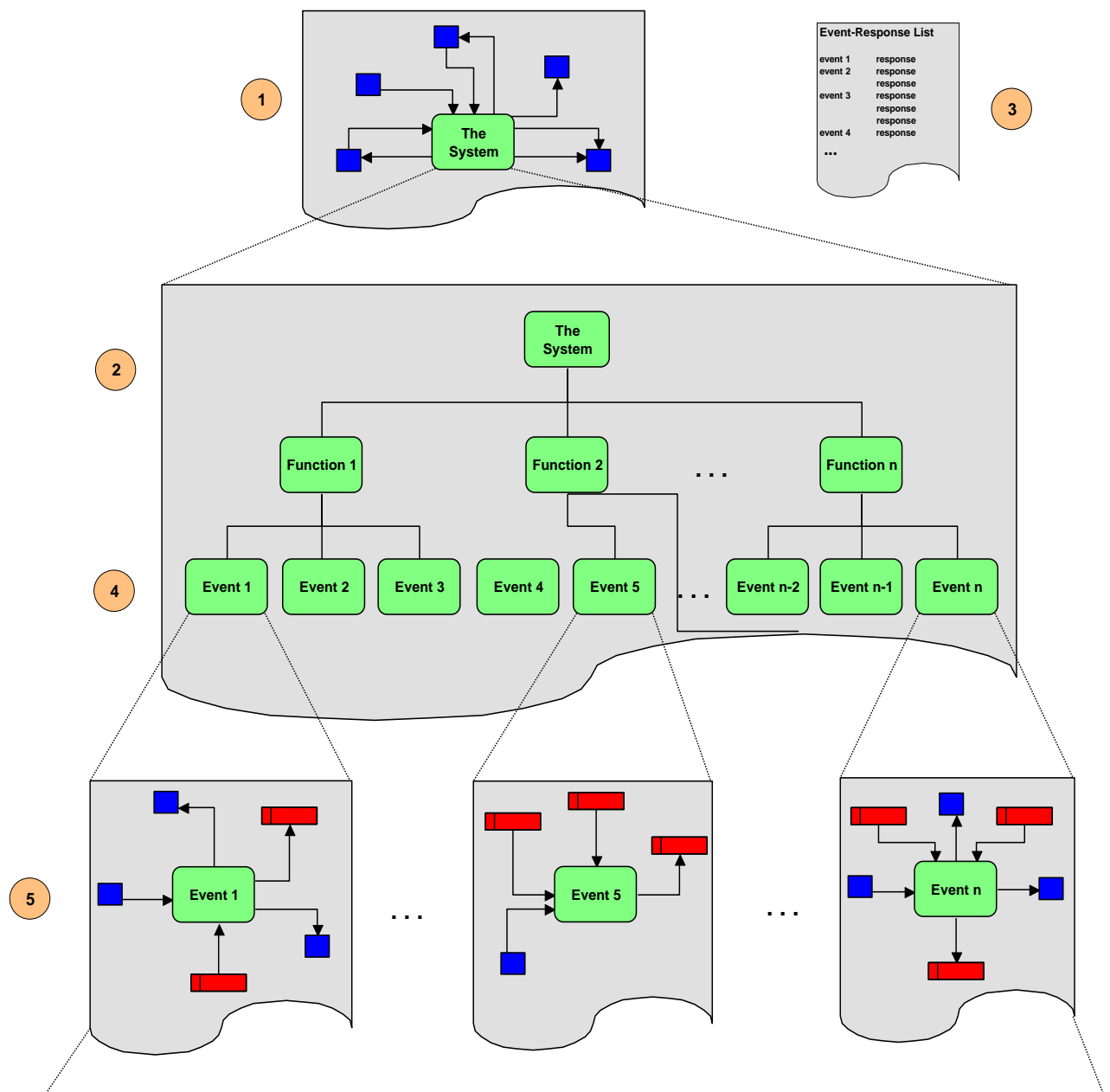
3.3 Permodelan Proses Sewaktu Analisis Sistem

Model proses logikal analisis sistem dipanggil model proses aplikasi. Kebanyakan strategi analisis berstruktur memberi fokus khusus kepada model logikal bagi sistem yang sedang dibangunkan. Model proses logikal disusun mengikut strategi yang dipanggil pemecahan peristiwa (event partitioning) yaitu membahagikan sistem kepada subsistem berdasarkan peristiwa-peristiwa bisnes dan unpan balik kepada peristiwa-peristiwa tersebut.

Strategi permodelan proses berdasarkan peristiwa ialah :

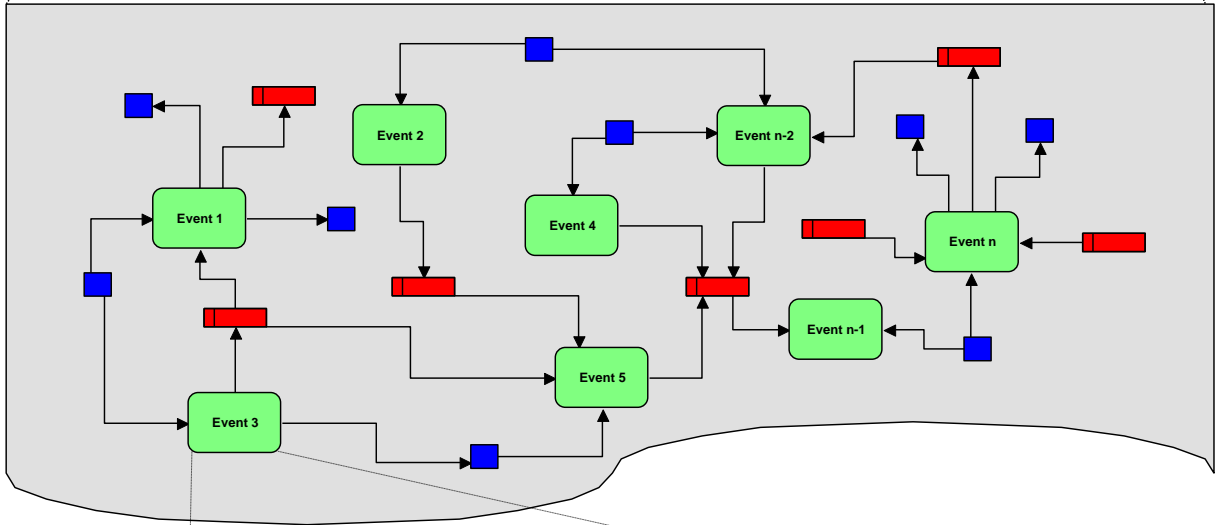
1. Diagram Konteks sistem dibina utk menunjukkan skop awal projek.
2. Functional decomposition diagram dilukis utk memecahkan sistem kpd subsistem/fungsi logikal.
3. Senarai tindakbalas peristiwa dikompil utk mengenalpasti dan mengesahkan peristiwa bisnes di mana sistem perlu bertindakbalas. .
4. Satu proses yg dipanggil pengendali peristiwa ditambah kpd setiap peristiwa dlm decomposition diagram.
5. Diagram peristiwa dibina dan disahkan utk setiap peristiwa.
6. Rajah(2) sistem dibina dgn menggabungkan rajah2 peristiwa.
7. Diagram primitif dibina utk setiap proses peristiwa. DFD ini menunjukkan semua proses asas, storan data dan aliran data utk setiap peristiwa.

Model proses logikal menerangkan keperluan proses bisnes sistem, bukan menyelesaikan masalah teknikal. Fase konfigurasi menentukan cara terbaik melaksanakan kebutuhan tersebut dengan menggunakan teknologi. Sewaktu desain sistem, model proses logikal diterjemah menjadi model proses fizikal (dipanggil *skema aplikasi*) untuk arsitektur teknikal yang dipilih. Model proses logikal ini menunjukkan kemampuan&kekuatan teknikal teknologi yang dipilih.



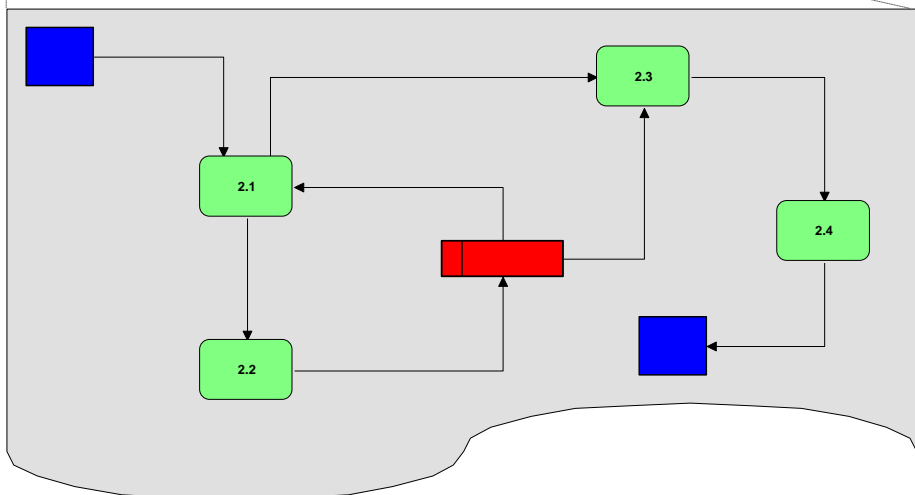
Gambar 10a: Event-Driven Process Modeling Strategy

6



...

7



Gambar 10b: Event-Driven Process Modeling Strategy

3.4 Pencarian Fakta dan Pengumpulan Maklumat bg Permodelan Proses

Model proses tidak bisa dibuat tanpa informasi dan fakta yang mencukupi. Fakta dikumpul melalui: formulir dan fail pada saat ini, kajian sistem yang serupa, angket/wawancara pengguna dan pengelola, joint application development (JAD).

4. Membuat Model Proses

4.1 Diagram Konteks

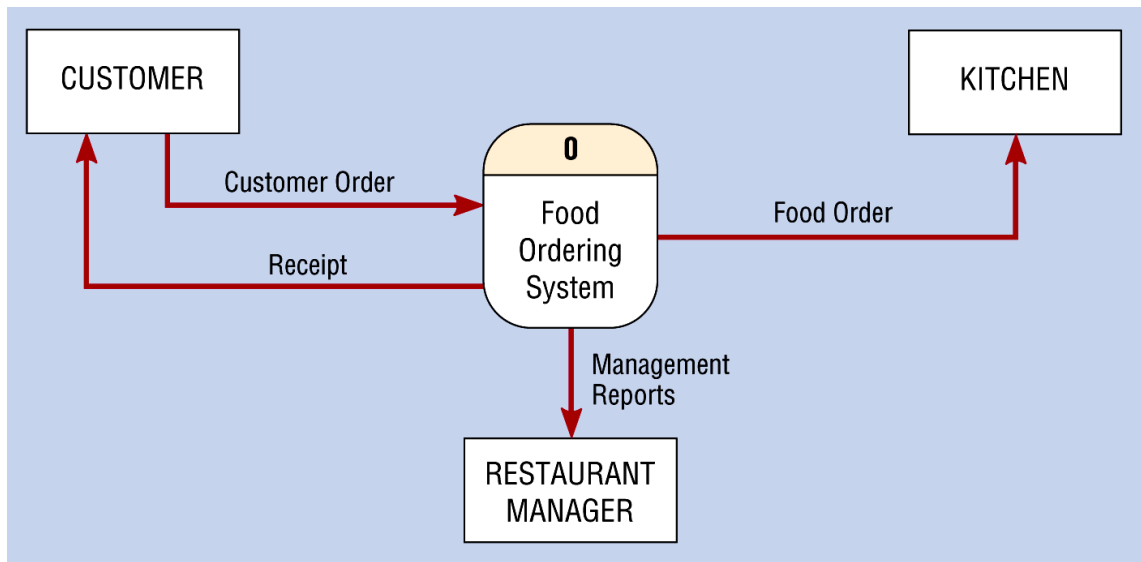
Diagram Konteks mempunyai hanya satu proses (proses 0). Entiti luaran dan storan data luaran dilukis di sekeliling proses. Aliran data mendefinisikan interaksi sistem dengan batasan serta storan data luaran. Ruang lingkup adalah salah satu bahasan yang penting dalam diagram konteks.

- Ruang lingkup perlu diketahui sebelum kita membuat model proses yang sebenarnya.
- Ruang lingkup projek mendefinisikan aspek yang perlu ditunjang oleh sistem bisnes/ aplikasi.
- Ruang lingkup proyek juga mendefinisikan bagaimana sistem/aplikasi yang dimodelkan harus berinteraksi dengan sistem-sistem lain bisnes secara keseluruhan.
- Ruang lingkup projek didokumentasikan dengan menggunakan *diagram konteks*.

Diagram konteks mendefinisikan ruang lingkup dan batasan sistem. Oleh kerana ruang lingkup projek bisa berubah, diagram konteks juga berubah. Strategi menentukan ruang lingkup dan batasan sistem :

1. Fikirkan sistem sebagai sebuah ‘container’ untuk membedakan dalam dan luaran sistem.
2. Tanya pengguna akhir tentang peristiwa/transaksi bisnes di mana sistem perlu unpan balik.
 - Ini merupakan *jumlah input* bagi sistem.
 - Untuk setiap input, tentukan sumbernya.
 - Sumber-sumber akan menjadi *entiti luaran* bagi diagram konteks.
3. Tanya pengguna-akhir mengenai unpan balik yang mesti dihasilkan oleh sistem.
 - Ini merupakan jumlah output sistem.
 - Untuk setiap output, tentukan destinasi.
 - Destinasi bisa terdiri dari *entiti luaran*.

4. Ketahui storan data *luaran*. Kebanyakan sistem memerlukan capaian kepada fail/pangkalan data sistem-sistem lain.
5. Lukis Diagram Konteks berdasarkan informasi yang diperolehi.



Gambar 11: The Context Diagram

4.2 The Functional Decomposition Diagram

The Functional Decomposition Diagram menunjukkan dekomposisi fungsian atas-bawah/struktur sistem. Item-item dalam decomposition diagram, adalah:

- Item 1:** Proses akar bagi keseluruhan sistem.
- Item 2:** Sistem mula dipecahkan kepada subsistem/fungsi.
- Item 3:** Pecahkan subsistem kepada aspek operasi dan laporan.

4.3 Daftar Tindak Balas Peristiwa

Untuk menentukan peristiwa bisnes di mana sistem perlu berunpan balik dan jenis unpan balik yang sesuai. Beberapa input pada diagram konteks berkaitan dengan peristiwa.

Terdapat tiga jenis peristiwa, yaitu:

- Peristiwa luaran : dicituskan oleh entiti luaran. Digambarkan sebagai aliran data input.

- Peristiwa Sementara : mencetus proses mengikuti masa. Digambarkan oleh *aliran kawalan*.
- Peristiwa Keadaan : mencetus proses berdasarkan pertukaran sistem dari satu keadaan kepada keadaan yang lain. Digambarkan oleh *aliran kawalan*.

Setiap peristiwa perlu dinamakan mengikut jenisnya, seperti:

Peristiwa luaran : Nama entiti luaran + sebab bagi aliran data. Seperti: CUSTOMER REQUESTS ACCOUNT BALANCE.

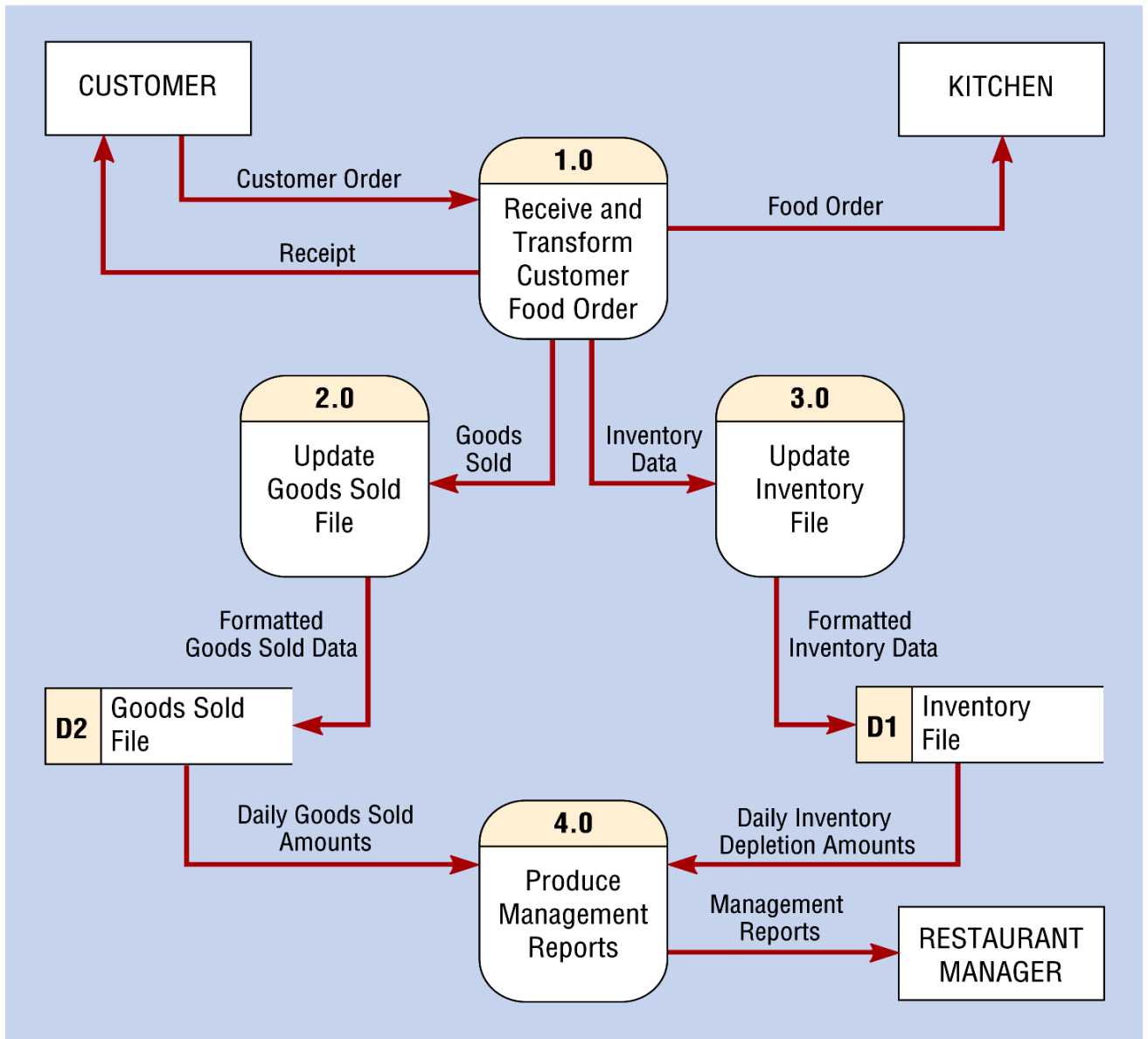
Peristiwa Sementara : Waktu untuk + tindakan yang perlu diambil, seperti: TIME TO BILL CUSTOMER ACCOUNTS.

4.4 Diagram Aliran Data Tingkat 0

Dengan menggunakan functional decomposition diagram sebagai panduan, buat DFD tingkat 0 yang memiliki semua subsistem/fungsi asas sistem keseluruhan. DFD tingkat 1 adalah DFD yang dihasilkan dari perincian Diagram Konteks

Untuk setiap peristiwa, tunjukkan:

- Input dan sumbernya (entiti luaran).
- Output dan destinasi (entiti luaran).
- Storan data (yang memiliki simpanan yang hendak dibaca, ditambah, dihapus atau diubah). Tunjukkan dan namakan Aliran Data untuk menunjukkan jenis data tsbt.



Gambar 12: Level-0 Data Flow Diagram

4.5 Diagram Dekomposisi Peristiwa

Diagram dekomposisi peristiwa untuk memecahkan fungsi dalam *decomposition* diagram. Hanya dengan menambah proses pengendalian peristiwa (satu bagi setiap peristiwa) kepada dekomposisi. Jika diagram menggunakan lebih dari satu halaman, tambah halaman lain. Proses akar pada halaman seterusnya perlu diambil dari halaman sebelumnya untuk menunjukkan hubungan.

Dekomposisi DFD

Berdasarkan Diagram Dekomposisi Peristiwa, bagian DFD tingkat 0 kepada tingkat 1, 2,... n Misalnya: Bagi Proses 4, terdapat 3 lagi subproses. Oleh itu, buat DFD tingkat 1 untuk Proses 1 yang memiliki subproses 4.1, 4.2 dan 4.3.

Synchronization : mengimbangi DFD di tingkat yang berlainan untuk mengekalkan konsistensi dan kelengkapan.

Setiap aliran data, storan data dan entiti luaran pada DFD di tingkat yang lebih tinggi mesti dimasukkan pada DFD di tingkat yang lebih rendah. Ini dipanggil sebagai keseimbangan (*balancing*). Kebanyakan peralatan CASE bisa menyemak perbaikan keseimbangan. Jangan gabungkan storan data kerana dapat menimbulkan perbaikan keseimbangan

4.6 Diagram Primitif

- DFD primitif menunjukkan pemprosesan terperinci bagi peristiwa.
- DFD peringkat terendah.
- Diagram primitif menunjukkan beberapa proses dasar.
- Setiap proses dasar melaksanakan hanya satu tugas. Seperti: capai, perbaiki, baca, hapus.
- Setiap proses dasar bisa diterangkan menggunakan Structured Language/decision tables.