STANDARD FORINTEGRATION DEFINITION FOR FUNCTION MODELING (IDEF 0)

Standard ini menjelaskan bahasa pemodelan (semantik dan sintak) IDEF0, beserta aturan dan teknik yang berkaitan dengannya, untuk membangun representasi grafik yang terstruktur dari sebuah sistem atau enterprise. Standar ini menjadi referensi bagi pembuat model sistem dan enterprise yang menggunakan teknik pemodelan IDEF0, bagi implementor dalam pembangunan *tool* untuk mengimplementasikan teknik ini, dan bagi profesional dalam bidang komputer dalam memahami sintak dan aturan semantik dari standar ini. Standar ini menjelaskan bahasa pemodelan (sintak dan semantik) yang mendukung teknik IDEF0 untuk membangun representasi grafis yang terstruktur dari sistem/subjek area.

Tujuan dari standar ini antara lain:

a. Secara lengkap dan konsisten memodelkan fungsi dari sebuah sistem/enterprise, hubungan fungsional dan data yang mendukung integrasi fungsi-fungsi tersebut.

b. Menyediakan teknik pemodelan yang independen dari metode dan *tools Computer-Aided S/W Engineering* (CASE), namun dapat digunakan bersama metode dan *tools* tersebut.

c. Menyediakan teknik pemodelan dengan karakteristik : generic (untuk analis sistem dari berbagai *purpose*, skope, dan kompleksitas), rigorous dan precise (untuk menghasilkan model yang *correct* dan *usable*, conceptual (merepresentasikan kebutuhan fungsional daripada fisikal atau implementasi organisasional), flexible (mendukung beberapa fase dari siklus projek).

Penggunaan standar ini direkomendasikan untuk projek yang: membutuhkan teknik pemodelan untuk analisis, development, re-engineering, integrasi, atau acquisition dari sistem informasi, *Incorporate* teknik pemodelan sistem atau enterprise ke dalam proses bisnis analisis atau metodologi S/W engineering.

7.1. Latar Belakang Teknik IDEF0

Sejak 1970-an, program U.S.Air Force untuk Manufacturing Computer Aided yang Terintegrasi (ICAM) memperlihatkan pertumbuhan produktifitas manufaktur melalui aplikasi yang sistematik dari teknologi komputer. Program ICAM membangun sebuah seri dari teknik yang dikenal dengan teknik IDEF (ICAM Definition) yang meliputi hal-hal dibawah ini:

1. IDEF0, digunakan untuk menghasilkan 'model fungsi'. Sebuah model fungsi adalah representasi terstruktur dari fungsi, aktivitas, atau proses dari sistem / subjek area yang dimodelkan.
2. IDEF1, digunakan untuk menghasilkan 'model informasi'. Sebuah model informasi merepresentasikan struktur dan semantik dari informasi dengan sistem/subjek area yang dimodelkan.
3. IDEF3, digunakan untuk menghasilkan 'model dinamik'. Sebuah model dinamik merepresentasikan perilaku karakteristik variasi-waktu dari sistem/subjek area yang dimodelkan.

7.2. Pendekatan IDEF0

IDEF0 (Integrated Definition Language 0) didasari oleh SADT™ (Structured Analysis and Design Technique™), yang dibangun oleh Douglas T.Ross dan Softech,Inc. Dalam bentuk originalnya, IDEF0 mencakup definisi dari bahasa pemodelan (sintak dan semantik) grafik

dan deskripsi dari metodologi yang komprehensif untuk pembangunan model.

IDEF0 dapat digunakan untuk memodelkan automated dan non-automated sistem yang bervariasi. Untuk sistem-sistem yang baru, IDEF0 dapat digunakan pertama kali saat mendefinisikan kebutuhan dan menspesifikasikan fungsi, dan kemudian mendesain implementasi yang memenuhi kebutuhan dan menampilkan fungsi yang diinginkan. Untuk sistem yang sudah berjalan, IDEF0 dapat digunakan untuk menganalisa fungsi yang ditampilkan oleh sistem dan merekam mekanisme pembuatannya.

Sebagai bahasa pemodelan, IDEF0 memiliki karakteristik sbb:

1. Komprehensif dan ekspresif, mampu merepresentasikan secar grafis dari berbagai variasi bisnis, manufaktur, dan enterprise lainnya sampai level yang detil
2. Bahasa yang koheren dan sederhana. Menyediakan ekspresi yang precise dan rigorous, dan mengajukan penggunaan interpretasi yang konsisten
3. Mendorong komunikasi antara analis sistem, developer, dan user melalui pembelajaran yang mudah
4. Dapat digenerate oleh berbagai tools grafik, banyak produk komersial secara spesifik mendukung pembangunan dan analisis dari diagram dan model IDEF0.

7.3. Model IDEF0

IDEF0 adalah teknik pemodelan yang didasari oleh kombinasi grafik dan teks, ditampilkan dengan cara yang terorganisasi dan sistematik untuk dimengereti, mendukung analisis, logik, untuk perubahan potential , menspesifikasikan kebutuhan, atau mendukung desain sistem dan aktivitas integrasi. Model IDEF0 dibentuk dari diagram yang disusun secara hirarki yang pada tahapannya menunjukkan fungsi dan hubungan fungsional melalui kotak dan sintak/semantik panah. Teks dan glossary diagram menyediakan informasi tambahan dalam mendukung diagram grafik.

IDEF0 adalah teknik engineering untuk menampilkan dan memanage analisis kebutuhan, analisis keuntungan, definisi permintaan, analisis fungsional, desain sistem, maintenance, dan baselines untuk pengembangan yang berkelanjutan. Model IDEF0 menyediakan '*blueprint*' dari fungsi dan interfacenya yang harus ditangkap dan dipahami. Model IDEF0 merefleksikan bagaimana fungsi-fungsi sistem saling berhubungan dan beroperasi sebagaiman sebuah *blueprint* dari sebah produk merefleksikan bagaimana bagian-bagian yang berbeda diintegrasikan.

7.3.1. Sintak dan semantik

Komponen dari sintak IDEF0 adalah kotak, panah, aturan, dan diagram. Kotak merepresenasikan fungsi, dibedakan menjadi aktivitas, proses, dan transformasi. Panah merepresentasikan data atau objek yang terkait dengan fungsi. Aturan menjelaskan bagaimana komponen-komponen tadi digunakan, dan diagram menyediakan format untuk menampilkan model secara verbal dan grafik.

Kotak

Sebuah kotak menyediakan deskripsi dari apa yang terjadi pada fungsi yang didesain. Setiap kotak harus memiliki nama dan nomor di dalamnya. 'Nama' harus berupa kata kerja katif yang menunjukkan fungsi. Setiap kotak dalam diagram harus memiliki nomor di pojok kanan bawah kotak. Nomor kotak digunakan untuk mengidentifikasikan subjek pada teks yang terkait.

- kotak harus cukup dalam ukuran, untuk meletakkan nama
- kotak harus segiempat
- kotak harus digambar dengan garis lurus

Panah

Panah dibentuk dari 1 atau lebih segmen garis, dengan ujung panah pada satu ujungnya. Panah dapat lurus, bercabang, atau dijoinkan. Panah tidak menunjukkan flow atau sequens seperti pada flow model tradisional. Panah menunjukkan data atau objek yang terkait dengan fungsi yang ditampilkan.

- Panah yang membelok harus dibentuk kurva 90 derajat

- Panah harus digambar oleh garis lurus
- Panah harus digambar vertikal atau horisontal, tidak boleh diagonal
- Ujung panah harus menyentuh sisi terluar dari kotak, dan tidak melintang di dalam
- Panah harus dihubungkan dengan sisi kotak, bukan dengan sudut kotak

Semantik mereferensi ke makna dari komponen sintak pada bahasa dan memberi interpretasi yang benar. Interpretasi mengacu ke items seperti notasi kotak dan panah dan interface hubungan fungsional.

Kotak dan panah

Sejak IDEF0 mendukung pemodelan fungsi, nama kotak haruslah verb/verb phrase. Langkah definitif dari penamaan kotak adalah sesuai dengan panah-panah di sisinya, yang menunjukkan kondisi-kondisi komplemen.

Makna Panah dengan mnun/noun phrase yang menunjukkan particular data atau objck yang dircprcscntasikan.

- Panah di sisi kiri kotak adalah input. Input ditransformasikan oleh fungsi untuk menghasilkan output.
- Panah di sisi atas kotak adalah kontrol. Kontrol menunjukkan kondisi yang dibutuhkan oleh fungsi untuk menghasilkan keluaran yang tepat.
- Panah di sisi kanan kotak adalah keluaran. Keluaran adalah hasil dari fungsi.
- Panah di sisi bawah kotak adalah mekanisme. Bisa berarti hal-hal yang mendukung eksekusi fungsi, bisa juga berarti inharited dari parent kotak. Panah mekanisme yang berarah ke bawah adalah Call Arrow.

Aturan:

1. Kotak harus dinamai dengan aktif verb/verbphrase
2. Setiap sisi kotak memiliki standar relationship (misal: input di kiri, kontrol di atas, dan lain-lain)

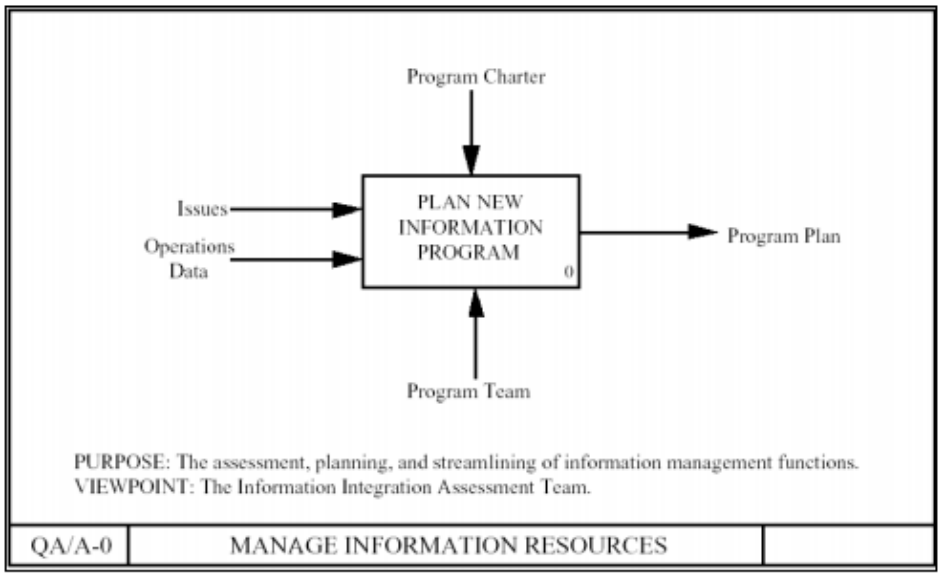
7.3.2. Diagram IDEF0

Model IDEF0 dibentuk oleh 3 tipe informasi: diagram grafik, teks, dan *glossary*. Diagram ini saling mereferensi satu sama lain. Diagram grafik adalah komponen utama dari model IDEF0, yang mengandung kotak, panah, kotak/panah *interconnection*, dan relationship yang terkait. Kotak merepresentasikan setiap fungsi dari subjek. Fungsi-fungsi ini didekomposisi menjadi diagram yang lebih detil. Top-level diagram dari model menampilkan deskripsi paling general dan abstrak dari subjek yang direpresentasikan. Diagram ini diikuti oleh diagram anak yang menampilkan detil dari subjek.

Top-Level Context Diagram

Setiap model harus punya *top-level diagram*, yang direpresentasikan oleh kotak tunggal dengan panah-panah yang mengitarinya. Ini disebut diaram A-0. *Context Diagram* ini juga harus memberikan statemen yang spesifik tentang *viewpoint* dari model dan tujuannya, yang memandu dan membatasi penciptaan model.

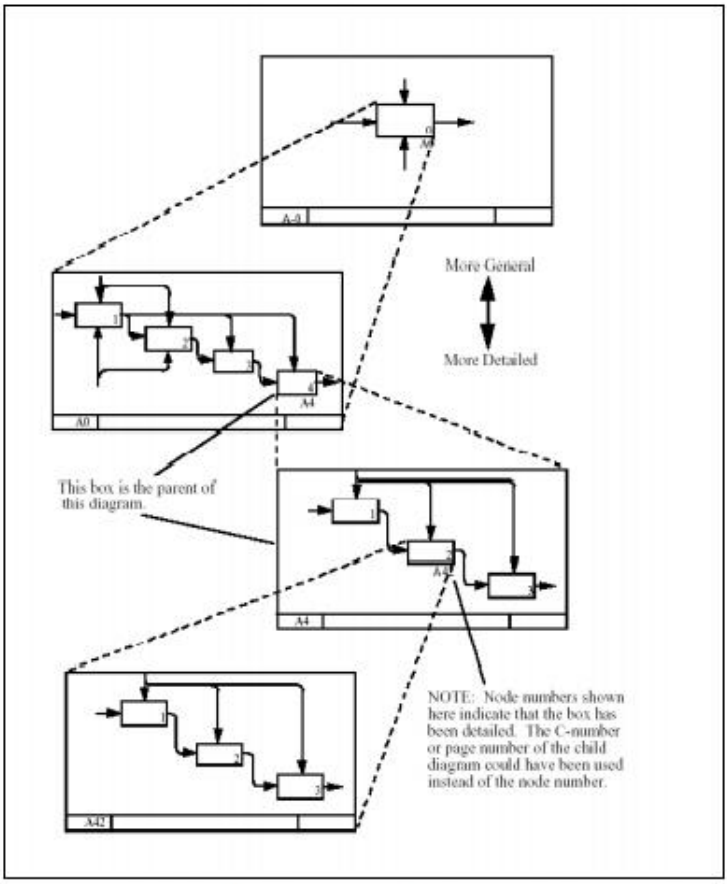
Contoh dari *Top-Level Context Diagram* ditunjukkan pada gambar berikut ini :



Gambar 1. *Top-Level Context Diagram*

Child Diagram

Fungsi pada *top-level* dapat didekomposisi menjadi sub-fungsi dengan membuat diagram anak (*child diagram*). Diagram anak ini dapat didekomposisi lagi menjadi *lower-level child diagram*. Sebuah kotak dapat menjadi *parent* kotak yang didetilkan pada *child diagram*, atau *child kotak* yang muncul pada *child diagram*. *Hirarki parent* kotak dan *child diagram* dapat dilihat pada gambar berikut:



Gambar 2. *Hirarki parent* kotak dan *child diagram*

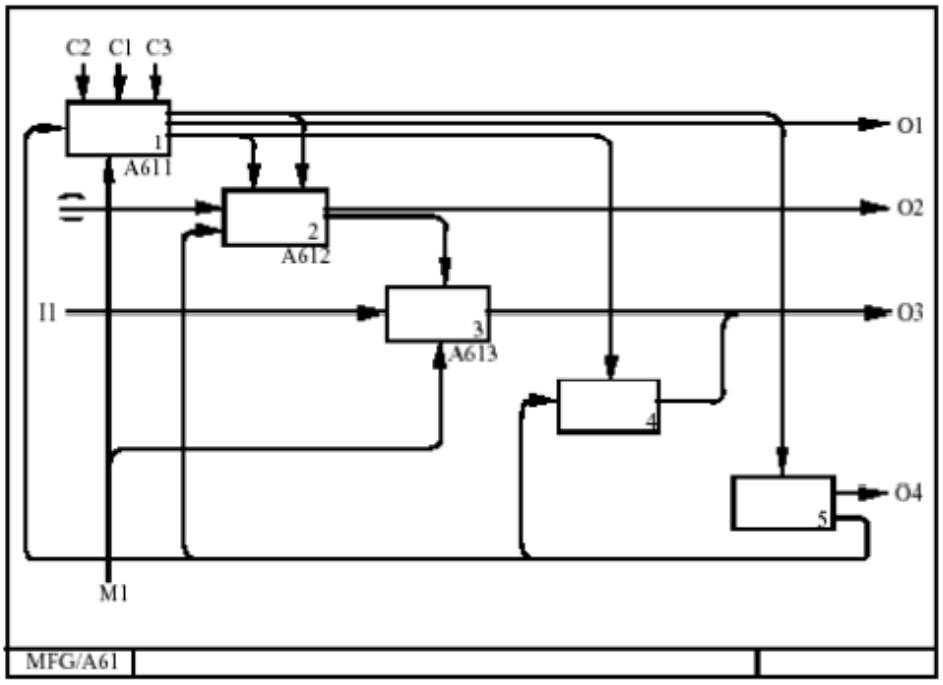
Diagram *Parent*

Diagram *parent* adalah diagram yang mengandung satu atau lebih *box parent*. Setiap diagram *ordinary* (*non-context*) adalah juga diagram *child* yang secara definisi digunakan untuk menjelaskan box parent. Jadi

sebuah diagram dapat berfungsi sebagai diagram *parent* (mengandung *box-box parent*) dan atau diagram *child* (menjelaskan secara detail *box parent*-nya). Demikian juga sebuah box dapat berfungsi sebagai *box parent* (dijelaskan oleh diagram child) dan atau box child (muncul dalam diagram child).

Dalam kenyataannya sebuah *box child* dapat dijelaskan lebih detil sehingga box tersebut juga merupakan box parent, hal ini diindikasikan dengan adanya *Detail Reference Expression* (DRE). DRE adalah kode singkat yang dituliskan di bawah bagian pojok kanan bawah dari sebuah detail box parent yang menunjuk ke diagram child dari box tersebut.

Gambar di bawah ini mengilustrasikan menggunakaan nomor node sebagai DRE. Pada gambar tersebut DRE digambarkan dibawah box 1, 2, dan 3 yang menyatakan box-box tersebut dijelaskan secara detail oleh diagram child yang ditunjuk.



Gambar 3. Penggunaan Detail Reference Expression (DRE)

4. Teks dan *Glossary*

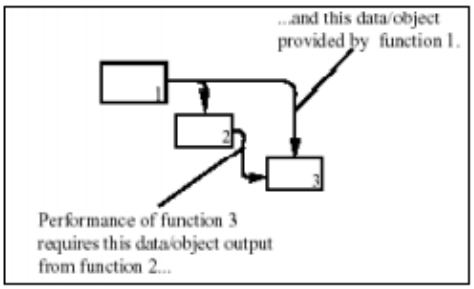
Sebuah diagram mungkin memiliki suatu teks yang terstruktur yang digunakan untuk menyediakan penjelasan singkat dari diagram tersebut. Teks biasanya digunakan utnuk menandakan fitur, flow, dan koneksi inter-box untuk menunjukkan tujuan dari item-item dan pola yang dianggap akan menunjukkan kejelasan. Teks tidak hanya digunakan untuk mendeskripsikan, secara redundan, arti dari box-box dan panah-panah yang ada.

Glossary digunakan untuk mendefinisikan akronim, kata kunci, dan frase yang digunakan dalam hubungannya dengan grafik diagram. *Glosarry* mendefinisikan kata-kata dalam model yang harus memberikan pengertian agar interpretasi yang didapatkan dari model. tersebut adalah benar.

7.3.3. Fitur-fitur Diagram

Panah sebagai Constraints

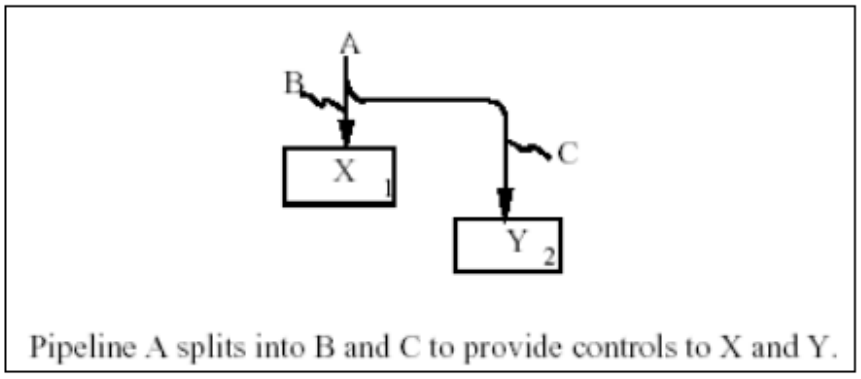
Panah dalam sebuah diagram IDEF0 merepresentasikan data atau objek sebagai constarint. Hanya level-level rendah dari detail yang dapat menunjuk flow atau sequence yang direpresentasikan. Menghubungkan output dari sebuah box ke inputnya, kontrol, atau mekanisme dari box lain menunjukkan bahwa fungsi yang dimodelkan oleh box selanjutnya dibutuhkan, dan oleh karenanya hal ini dibatasi oleh adanya hubungan antara output dari box former. Tipe keterhubugan ini diilustrasikan pada Gambar di bawah. Panah yang terhubung ke sebuah box menunjukkan semua data dan objek yang dibutuhkan agar fungsi yang ada berfungsi secara komplit.



Gambar 4. Panah sebagai Constraints

Panah sebagai Pipelines

Adalah sangat berguna untuk memikirkan panah pada level tinggi sebagai *pipeline* atau saluran. Panah pada level tinggi memiliki label yang umum, sedangkan panah pada diagram level rendah memiliki label yang lebih spesifik. Jika sebuah panah dipisahkan, membentuk dua atau lebih segmen panah, setiap segmen panah mungkin memliki label yang lebih spesifik, seperti yang ditunjukkan pada Gambar berikut.:



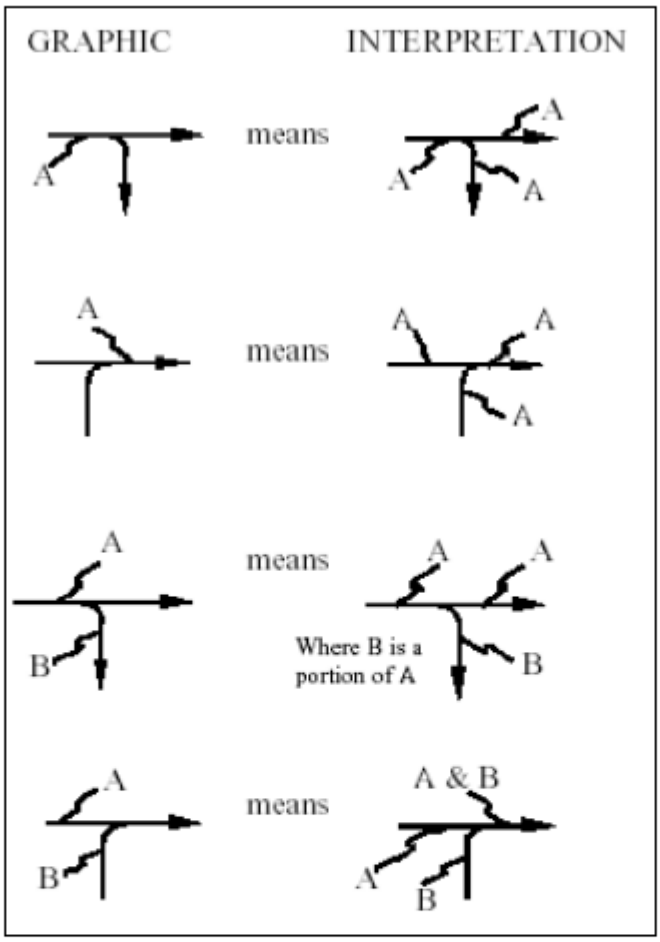
Gambar 5. : *Pipeline* Panah dengan Pencabangan

Panah Bercabang

Sebuah panah mungkin bercabang (pemecahan atau penggabungan), mengindikasikan bahwa data atau objek yang sama dibutuhkan atau diproduksi oleh lebih dari satu fungsi. Masing-masing cabang mungkin merepresentasikan hal yang sama atau bagian-bagian dari sebuah hal yang sama. Karena label menspesifikasikan segmen panah mana yang merepresentasikan, label pada segmen cabang panah menyediakan detail dari isi panah seperti diagram level rendah menyediakan detail dari box parent.

Semua atau sebagian dari isi dari sebuah panah mungkin mengikuti sebuah cabang. Panah bercabang menyebar dapat berarti "unbundling" (pemecahan ikatan) dalam arti sebelumnya dikombinasikan dalam sebuah label yang umum. Penggabungan dua segmen panah dapat berarti "bundling" (pengikatan), sebagai contoh, penggabungan beberapa arti yang terpisah menjadi sebuah kategori yang lebih umum.

Semua konten disediakan melalui semua cabang, jika tidak diindikasikan dengan adanya label spesial pada setiap segmen panah.

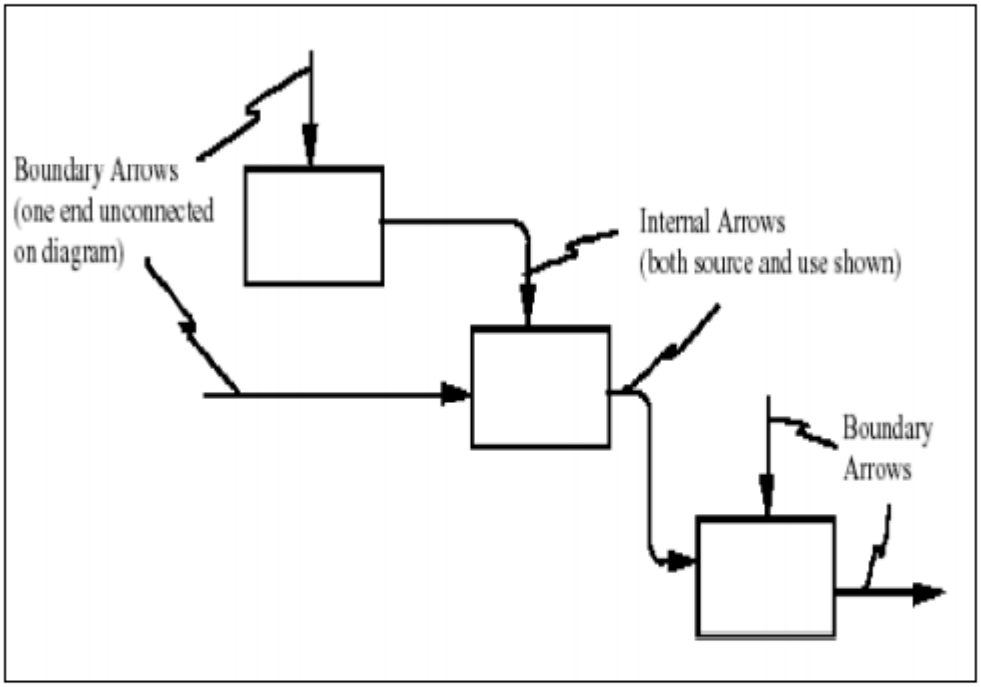


Gambar 6. Struktur Percabangan Panah

Keterhubungan *Inter-box*

Kecuali untuk diagram konteks box tunggal A-0, sebuah diagram grafik dapat mengandung minimum tiga dan maksimum enam box. Box-box secara normal disusun secara diagonal dari pojok kiri atas ke pojok kanan bawah. Panah output dapat menyediakan beberapa atau semua input, kontrol, atau mekanisme data atau objek untuk box lain. Sebuah panah output mungkin menyediakan data atau objek untuk beberapa box melalui mekanisme pencabangan.

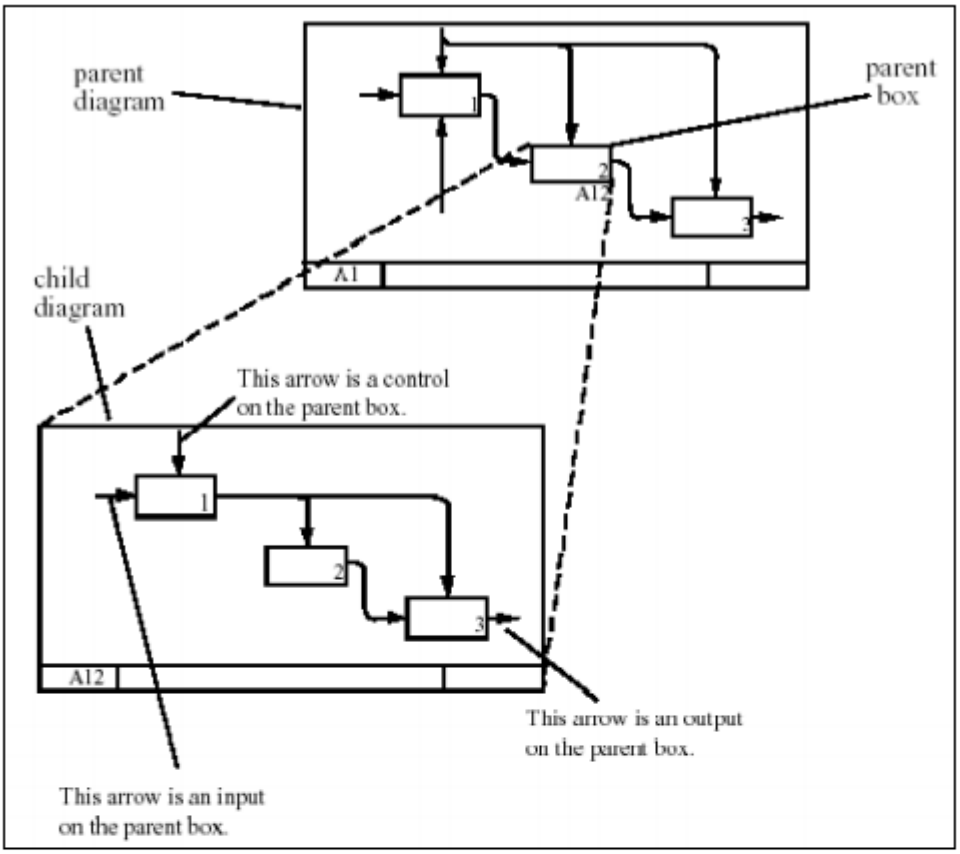
Jika sebuah box dalam diagram dijelaskan melalui sebuah diagram *child*, masing-masing panah yang terhubung ke box *parent* akan muncul di diagram anak, jika tidak berarti panah tersebut di-*tunnel* ke sebelah *box parent*. Dalam sebuah diagram, data atau objek dapat direpresentasikan oleh panah internal, dengan kedua ujungnya terhubung ke box, atau oleh panah pembatas dengan hanya satu ujung yang terhubung.



Gambar 7 : Boundary Arrows dan Panah Internal

Boundary Arrows

Boundary arrow dalam sebuah diagram grafik *ordinari* merepresentasikan input, kontrol, atau mekanisme dari *box parent* diagram tersebut. Ujung atau pangkan dari panah ini dapat ditemukan hanya dengan menelaah diagram parent. Semua boundary arrow dalam diagram child (kecuali untuk panah yang ditunnel) seharusnya terhubung ke panah yang terhubung ke box parent. Seperti ditunjukkan ada Gambar berikut. :



Gambar 8. Keterhubungan Boundary Arrow

7.3.4. Pengkodean ICOM dari *Boundary Arrows*

Kode ICOM menghubungkan boundary arrow pada panah diagram *child* yang terhubung ke box parent-nya. Sebuah notasi spesifik, yang disebut kode ICOM, menspesifikasikan koneksi yang bersesuaian. Huruf I, C, O, atau M ditulis didekat ujung yang tidak terhubung pada masing-masing *boundary arrow* pada diagram *child*. Kode ini mengidentifikasi panah sebagai sebuah input, kontrol, output, atau mekanisme dari box parent. Huruf-huruf ini diikutu oleh nomor yang menggambarkan posisi relatif dimana panah tersebut terhubung ke box parent, penomoran dilakukan dari kiri ke kanan atau atas ke bawah.

Pengkodean ini menghubungkan masing-masing diagram *child* ke *box parent*-nya. Jika box-box dalam diagram *child* dijelaskan pada diagram *child* berikutnya, kode ICOM yang baru ditentukan pada masing-masing

diagram *child*, menghubungkan *boundary arrow* dari diagram dengan panah pada box parent yang ada.

Dengan menggunakan skema kesesuaian penomoran huruf (letter-numbering) dari pengkodean ICOM, aturan panah (input, kontrol, mekanisme) dapat berbeda antara diagram parent dan diagram *child*. Sebagai contoh perubahan aturan, sebuah panah kontrol pada *box parent* dapat berupa input atau kontrol untuk box-box yang ada pada diagram *child*. Demikian juga, sebuah kontrol untuk *box parent* dapat berupa input untuk satu atau lebih *box child*.

Panah *Tunnel* (*Tunneled* Arrows)

Sebuah panah tunnel digunakan untuk menyediakan informasi pada level dekomposisi yang spesifik yang tidak dibutuhkan untuk dimengerti pada level lain. Sebuah panah dapat *ditunnel* pada setiap level yang dipilih. Panah *tunnel* ditandai dengan adanya tanda kurung pada ujung panah. Sebuah panah *tunnel* pada ujung yang terhubung dapat dihilangkan dari satu atau lebih level dekomposisi dan kemudian muncul kembali pada level lain, dalam satu atau lebih tempat, di-*tunnel* pada ujung yang tidak terhubung.

Men-*tunnel*-kan sebuah panah pada ujung yang tidak terhubung berarti data atau objek tidak dibutuhkan pada level selanjutnya yang lebih tinggi (parent) dan karenanya tidak perlu ditampilkan hubungan ke box parent. Hal ini ditunjukkan pada gambar. Karena panah ini tidak terhubung ke salah satu box pada diagram parent, panah tersebut tidak memiliki kode ICOM. Panah tersebut mungkin memiliki model tambahan yang mengandung referensi node dan kode ICOM yang berlokasi pada ujung lainnya dari panah tunnel. Pengkodean ICOM untuk panah mewakili semua diagram child yang ada selanjutnya.

Panah Panggil (*Call Arrows*)

Sebuah *call arrow* adalah kasus spesil dari panah mekanisme. Panah ini menunjutkkan bahwa *box* yang memanggil tidak memiliki diagram child sendiri untuk menjelaskan secara detail akan dirinya, tetapi dijelaskan detail secara keseluruhan oleh box lain (dan turunannya) dalam model yang sama atau berbeda.

Box pemanggil yang multiple mungkin memanggil box yang sama. Call arrow diberi label dengan *reference* ke node dari diagram yang mengandung box yang dipanggil, bersama dengan nomor dari box yang dipanggil.Sebuah box pemanggil mungkin hanya memanggil satu box dalam sebuah aktivasi. Bagaimanapun juga, bergantung pada kondisi yang dispesifikasikan dalam model note yang ditambahkan ke *call arrow*, box pemanggil dapat memilih satu dari beberapa box yang mungkin dipanggil. Dalam kasus ini, label dari call arrow dapat mengandung list dari node yang direferensikan dari semua box yang mungkin dipanggil.

Panah dari box yang dipanggil dapat tidak terhubung langsung dengan box yang dipanggil, baik dalam hal nomor atau arti. Dalam kasus ini, model note yang ditambahkan pada *call arrow* harus menspesifikasikan hubungannya sehingga memberikan interpretasi yang benar pada data dan objek yang dishare.

7.3.5. Aturan Sintaks Diagram

Secara umum aturan sintaks dalam pembuatan diagram adalah sebagai berikut :

1. Diagram context memiliki nomor node A-n, dimana n>=0.
2. Modelnya harus mengandung sebuah diagram context A-0, yang hanya mengandung satu box.
3. Nomor box dari box tunggal dalam diagram context A-0 adalah 0.
4. Sebuah diagram non-context setidaknya memiliki tiga box dan tidak lebih dari enam box.

5. Setiap box dalam diagram non-context harus diberi nomorpada bagian bawah kanan didalam kotak, dengan nomor 1 sampai 6 (dari kiri atas ke kanan bawah dari diagram).
6. Setiap box yang sudah dijelaskan harus memiliki DRE dari diagram child yang ditulisan di bawah bagian pojok kanan bawah box.
7. Panah digambarkan sebagai garis lurus horizontal dan vertikal. Garis diagonal sebaiknya tidak digunakan.
8. Setiap box setidaknya memiliki satu panah kontrol dan satu panah output.
9. Sebuah box memiliki 0 atau lebih panah input.
10. Sebuah box memiliki 0 atau lebih panah mekanisme non-call.
11. Sebuah box memiliki 0 atau 1 call arrow.
12. Kontrol terhadap umpan balik digambarkan sebagai “up dan over”. Input umpan balik digambarkan sebagai “down dan under”. Mekanisme umpan balik digambarkan sebagai “down dan under”.
13. Boundary arrow yang ujungnya tidak terhubung memiliki kode ICOM yang benar yang menspesifikasikan koneksinya ke box parent atau dapat di tunnel.
14. Boundary arrow open-ended yang merepresentasikan data atau objek yang sama harus dihubungkan melalui percabangan untuk menunjukan semua tempat yang terpengaruh, jika tidak hasilnya adalah diagram yang tidak bisa dibaca. Sumber multiple yang merepresentsikan data atau objek yang sama harus digabungkan untuk membentuk satu boundary arrow output.
15. Nama box dan label panah dapat tidak hanya mengandung kata-kata function, activity, process, input, output, control atau mechanism.

7.3.6. Diagram Reference Expressions (DRE)

Ekspresi reference menggunakan kode yang ditetapkan untuk fitur dari model seperti diagram, box, panah, dan note. Ekspresi reference yang dapat digunakan dalam berbagai konteks untuk mengacu dengan tepat pada semua aspek dari model.

Unit dasar dari reference adalah nomor node, yang digunakan pada tempat dimana fungsional dekomposisi dimodelkan oleh detail dari box parent dalam diagram child. Kode reference lainnya berdasarkan nomor node.

Nomor Box

Setiap box dalam diagram harus dinomori pada bagian bawah kanan dalam kotak. Sistem penomoran ini dibutuhkan untuk mengidentifikasi box yang ada di dalam diagram secara unik, dan untuk men-generate nomor node. Selain itu digunakan juga sebagai masukan cross-reference dalam teks dan glossary box pada diagram.

Nomor Node

Sebuah nomor node didasarkan pada posisi dari sebuah box dalam model hirarki. Secara normal sebuah nomor node dibentuk dengan menggabungkan nomor box dengan nomor node dari diagram. Ketika sebuah box dijelaskan detail oleh sebuah diagram child, nomor node dari box parent digunakan sebagai nomor node dari diagram, oleh karena itu box parent dan diagram child-nya memiliki nomor node yang sama.

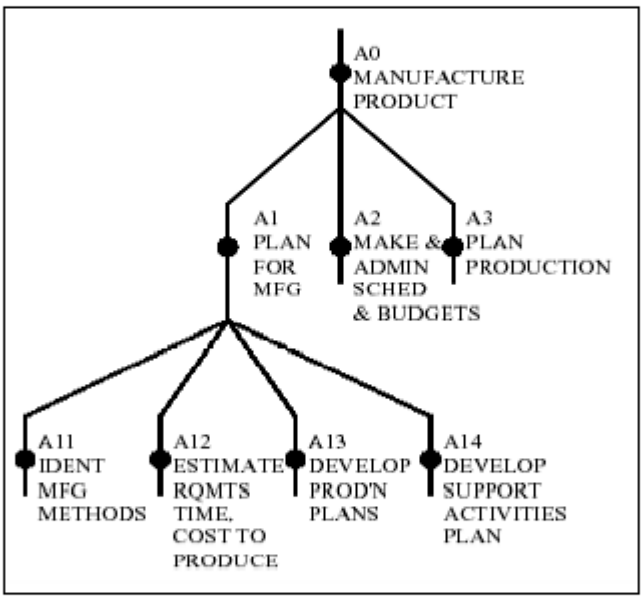
Diagram context dan diagram child pada level atas mendapatkan pengecualian dari aturan penomoran node. Setiap model IDEF0 memiliki diagram context level atas, diagram A-0. Diagram ini mengadung sebuah top box yang merupakan parent yang unik dari keseluruhan model. Nomor node dapat juga digunakan untuk menjelaskan ekspresi reference untuk mengindikasikan detail dari sebuah box parent dari diagram child.

Indeks Node

box, harus dipresentasikan dalam bentuk yang identik yang menunjukkan struktur hirarki bersarang dari model tersebut. Semua ini ditempatkan bersama dalam sebuah Daftar Isi.

7.3.7. Pohon Node (Node Tree)

Model pembangunan IDEF0 dengan struktur dekomposisinya menyediakan basis untuk menggambarkan komposisi lengkapnya dalam bentuk pohon node dalam diagram kecil. Penggunaan dari pohon node adalah opsional. Isi dari pohon node adalah indentik dengan indeks node atau hal lain yang sesuai. Tidak ada format standar untuk menampilkan informasi node, kecuali hirarkinya harus digambarkan sebagai akar pohon dari node yang dipilih.



Gambar 11. *Node Tree*

Reference Node

Setiap diagram dalam sebuah model memiliki sebuah reference node, yang secara unik mengidentifikasi dirinya dan posisinya dalam model hirarki. Reference node dibentuk dari singkatan nama model dan nomor node diagram, dipisahkan dengan slash (/). Reference ke diagram dalam model yang sama dapat menghilangkan singkatan nama model jadi

hanya menggunakan nomor node. Reference node juga dapat memiliki awalan.

Note Model

Note model adalah opsional. Digambarkan dengan sebuah integer "n" dalam sebuah kotak kecil. Untuk diagram yang diberikan, nomor note membentuk sebuah keterurutan, dimulai dari 1. Garis vertikal diantara nomor note (|n|), dapat digunakan sebagai alternatif notasi. Note model menyediakan informasi yang relevan dari sebuah pesan diagram, tetapi tidak persis dengan sintaks dari box dan panah.

Notasi Pengacuan

Notasi standar digunakan dalam menulis text dan notes untuk mengacu diagram dan bagian spesifk dari diagram. Pengacuan didasarkan pada jumlah box, nomor node, kode ICOM, dan nomor note. Tabel berikut ini menampilkan contoh notasi pengacuan.

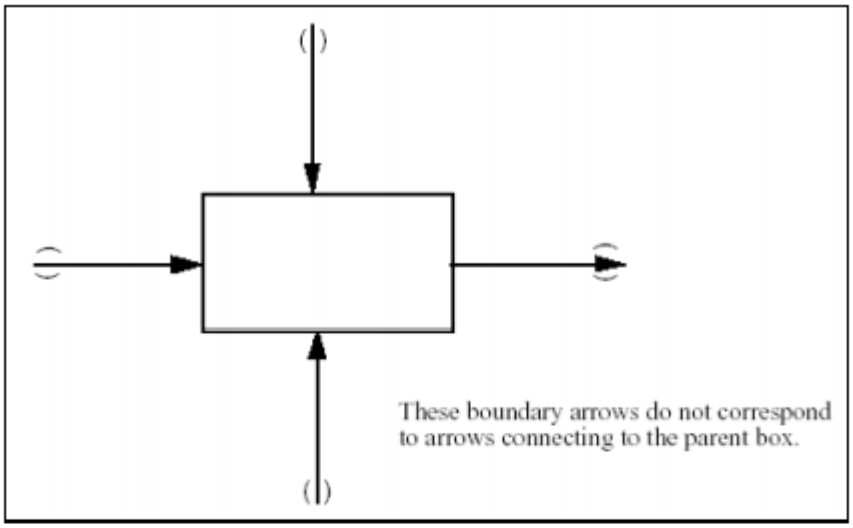
Notasi pengacuan	Arti
2I1	Box 2 Input 1
O2	Boundary arrow yang memiliki kode ICOM 02
2O2 ke 3C1 atau 2o2 ke 3c1	Panah dari 2O2 ke 3C1 (I,O,C, atau M mungkin huruf besar atau huruf kecil)
I2 ke 2I3 ke 2O2 ke (3C1 dan 4C2)	Dari boundary arrow dengan kode ICOM 12 ke Box 2 input 3, melalui aktifasi Box 2 output 2 menuju availability (dengan forking branch) output tersebut sebagai control 1 pada box 3 dan control 2 pada box 4
A21.3C2	Pada diagram A21 dalam model ini, lihat Box 3 Control 2
A42. 3	Pada diagram A42, lihat model note 3
A42.|3|	Sama dengan atas, menggunakan notasi optional (dengan pipa vertikal di samping model note, bukan boxed note)
A42.3	Pada diagram A42 dalam model ini, lihat Box 3
MFG/A42.1	Pada diagram A42 dalam model dengan singkatan MFG, lihat Box 1

Ringkasan

IDEF0 dapat digunakan untuk memodelkan automated dan non-automated sistem yang bervariasi. Untuk sistem yang baru, IDEF0 dapat digunakan pertama kali saat mendefinisikan kebutuhan dan menspesifikasikan fungsi, dan kemudian mendesain implementasi yang memenuhi kebutuhan dan menampilkan fungsi yang diinginkan. Untuk sistem yang sudah eksis, IDEF0 dapat digunakan untuk menganalisa fungsi yang ditampilkan oleh sistem dan merekam mekanisme pembuatannya.

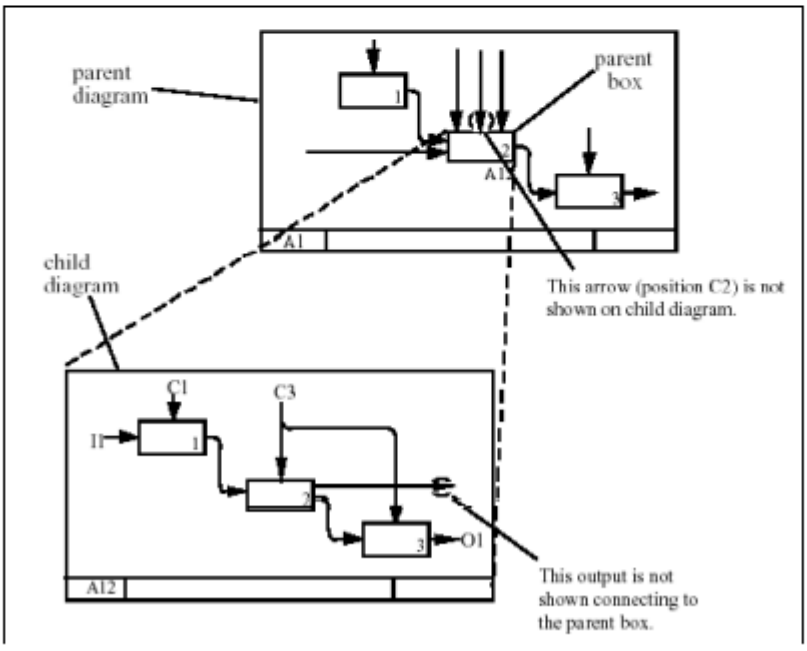
Latihan Soal

1. Tuliskan prinsip pemodelan proses Bisnis menggunakan IDEF0
2. Tuliskan satu contoh proses bisnis dan buatlah pemodelan menggunakan IDEF0.



Gambar 9. Arrows Tunneled at Unconnected End

Secara lebih rinci penggunaan tunnel digambarkan pada gambar berikut :



Gambar 10. Example of Tunneled Arrows