

Designing the System

Oleh :

Ir. I Gede Made Karma, MT

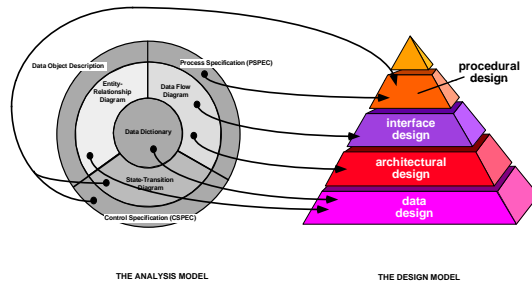
Posisi Perancangan dalam RPL

- Sebagai 'technical kernel' dari proses pembangunan PL.
- Merupakan proses kreatif dalam pembangunan PL untuk memecahkan persoalan.
- Filosofi proses perancangan:
 - mengemukakan suatu solusi
 - membangun model dari solusi
 - evaluasi model terhadap spesifikasi kebutuhan yang telah ada
 - menjabarkan kerincian atas solusi tersebut

Fungsi Proses Perancangan

- Translasi / pengembangan spesifikasi PL
- Penjabaran bagaimana PL dapat berfungsi
- Penjabaran bagaimana spesifikasi PL dapat diimplementasikan
- Input : SRS
- Output : SDD

Elemen Proses Perancangan (1)



Elemen Proses Perancangan (2)

- Perancangan Data
 - transformasi model data yang dihasilkan oleh proses analisis menjadi struktur data yang dibutuhkan pada saat implementasi
- Perancangan Arsitektur
 - definisi keterkaitan antar elemen-elemen utama yang akan membentuk program
- Perancangan Antarmuka
 - penjabaran komunikasi: internal PL, antara PL dengan sistem diluarnya, dan antara PL dengan usernya
- Perancangan Prosedur
 - transformasi elemen struktural dari arsitektur program menjadi deskripsi prosedur.

Perancangan vs Kualitas PL

- Selama proses perancangan, kualitas perancangan selalu dipantau melalui 'review teknis formal'.
- Petunjuk untuk evaluasi kualitas (McGlaughlin):
 - perancangan harus mengimplementasikan semua kebutuhan PL yang disebut eksplisit di SRS, sekaligus mengakomodasikan semua kebutuhan implisit dari SRS
 - harus readable, understandable khususnya bagi programmer, tester dan pelaku maintenance
 - harus menyajikan gambaran lengkap PL, meliputi: model data, fungsi, dan kelakuan PL dari sudut pandang perspektif implementasi.

Prinsip-prinsip Perancangan

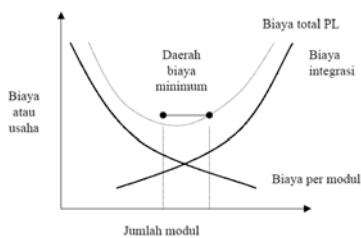
- Mempertimbangkan beberapa alternatif model solusi
- Traceable terhadap model analisis
- Mempertimbangkan dan menghasilkan komponen reusable
- Meminimasi kesenjangan antara PL dengan kondisi nyata
- Memperlihatkan keseragaman dan integrasi
- Mengakomodasi perubahan
- Mengakomodasi kondisi-kondisi insidental yang mungkin terjadi
- Abstraksi lebih detil dari analisis, tetapi lebih tinggi dari coding
- Dapat terukur kualitasnya
- Harus di-review untuk meminimasi kesalahan semantik

Konsep Perancangan (1)

- Abstraksi:
 - abstraksi data
 - abstraksi prosedur
 - abstraksi kontrol/kendali
- Refinement/elaborasi
 - merinci abstraksi tingkat tinggi menjadi representasi yang lebih mengarah pada struktur internal sistem.

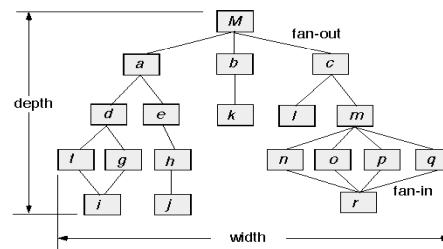
Konsep Perancangan (2)

- Modularitas



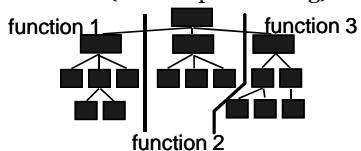
Konsep Perancangan (3)

- Arsitektur PL



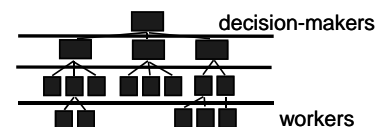
Konsep Perancangan (4)

- Hierarki kendali
 - struktur program merepresentasikan organisasi dari modul-modul, yang implikasinya merupakan kendali hierarki
- Partisi struktur (vertical partitioning)



Konsep Perancangan (5)

- Partisi struktur (horizontal partitioning)



Konsep Perancangan (6)

- Mengapa dipartisi ?
 1. Memudahkan pengujian PL
 2. Memudahkan perawatan PL
 3. Meminimalkan side efek
 4. Memudahkan pengembangan PL

Konsep Perancangan (7)

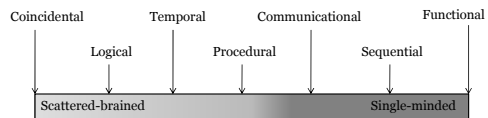
- Struktur Data:
 - scalar item
 - sequential vector (contiguous)
 - linked list
 - kombinasi dari beberapa format
- Penting karena mempengaruhi:
 - organisasi informasi
 - metode akses
 - tingkat keterkaitan informasi
 - alternatif pemrosesan informasi

Konsep Perancangan (8)

- Prosedur PL
 - menjabarkan kerincian pemrosesan setiap modul, termasuk urutan proses dan pengulangan operasi
- Information hiding
 - modul-modul harus dirancang sedemikian rupa sehingga informasi (prosedur dan data) yang terkandung, tidak dapat diakses oleh modul lain yang tidak berkepentingan dengan informasi tersebut.

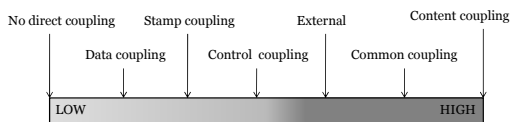
Perancangan Modular Efektif (1)

- Functional Independence
 - Diukur berdasarkan dua kriteria kualitatif
 - (1) Cohesion: derajat kekuatan fungsional relatif dalam suatu modul



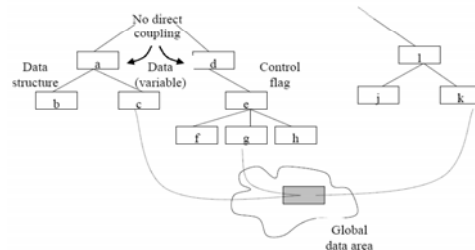
Perancangan Modular Efektif (2)

- Functional Independence
 - (2) Coupling: derajat kebergantungan relatif antar modul



Perancangan Modular Efektif (3)

- Contoh situasi coupling dalam perancangan PL:



Perancangan Heuristik

- Agar hasil perancangan menjadi modular dan efektif
- Menggunakan petunjuk-petunjuk teknis di setiap tahap perancangan (7 langkah heuristik):
 - pada iterasi pertama: kurangi coupling, naikkan cohesion
 - minimisasi fan-out, usahakan fan-in ketika depth bertambah
 - tetap perhatikan scope of control dari modul
 - evaluasi antarmuka modul untuk mereduksi kompleksitas dan menaikkan konsistensi
 - definisikan modul yang fungsinya dapat diprediksi, dan hindari modul yang sangat terbatas 'fungsinya'
 - usahakan modul yang memiliki "controlled entry", hindari 'pathological connections'
 - 'bungkus'lah PL sesuai kebutuhan portabilitas dan batasan perancangan.

Deliverable Tahap Perancangan

- Dokumen Software Design Document (SDD):
 - Ruang Lingkup
 - Perancangan Data
 - Perancangan Arsitektural
 - Perancangan Antarmuka
 - Perancangan Prosedur
 - Kebutuhan Lainnya
 - Persiapan Pengujian
 - Catatan Khusus

Kesimpulan

- Proses Perancangan:
 - 'technical kernel' dari RPL
 - proses utama: refinement atas isi SRS
 - hasil rancangan harus dapat diukur kualitasnya
 - dapat menggunakan beberapa pendekatan metode perancangan
 - sebelum masuk ke tahap coding, harus dilakukan review atas SDD yang dihasilkan

Tahap Perancangan

- Perancangan Data
- Perancangan Arsitektural
- Perancangan Antarmuka
- Perancangan Prosedural

Perancangan Data (1)

- Memilih representasi logik dari objek data yang ditemukan pada proses analisis
- Refinement terhadap Data Dictionary menjadi:
 - struktur data (array, list, dll.)
 - struktur file / basis data lengkap dengan fieldnya
- Penentuan struktur data maupun struktur file /basis data membutuhkan kreativitas dan sistematika yang rapi agar tidak menyulitkan proses berikutnya, khususnya perancangan prosedur.

Perancangan Data (2)

- Petunjuk teknis perancangan data:
 - menerapkan prinsip-prinsip analisis sistematis (pada tahap analisis)
 - mengidentifikasi semua struktur data dan prosedur yang akan digunakan untuk mengakses data tsb
 - me-refine isi data dictionary
 - menunda perancangan data yang 'low-level' sampai di akhir-akhir proses perancangan
 - merepresentasikan struktur data sedemikian rupa sehingga hanyamodul yang menggunakan data tersebut yang dapat mengaksesnya
 - membangun pustaka untuk struktur data dan prosedur yang sering digunakan
 - mendukung spesifikasi dan realisasi ADT

Perancangan Data (3)

- Hasil perancangan data adalah:
 - struktur data siap diprogram
 - struktur basis data siap dibuat oleh pemrogram
 - prosedur/operasi untuk mengakses data, siap diprogram

Perancangan Arsitektural (1)

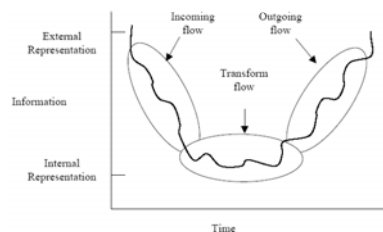
- Objektif utama:
 - membangun struktur program modular dan merepresentasikan keterkaitan kendali antar modul; dan
 - memadukan struktur program dan struktur data, dan mendefinisikan antarmuka yang memungkinkan data dapat mengalir pada seluruh program.

Perancangan Arsitektural (2)

- Proses:
 - perubahan dari aliran informasi (direpresentasikan dengan DFD) menjadi struktur PL (direpresentasikan dengan Structure Chart).
- Langkah:
 - (1) menentukan jenis aliran informasi
 - (2) menentukan batas aliran informasi
 - (3) pemetaan dari DFD ke struktur program
 - (4) menentukan hierarki kendali dengan cara faktorisasi
 - (5) menghaluskan struktur program yang terbentuk, dengan mempertimbangkan faktor-faktor pengukuran PL dan heuristik (fan-in, fan-out, coupling, cohesion, dll)

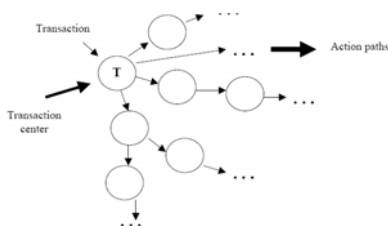
Perancangan Arsitektural (3)

- Jenis aliran informasi:
 - (1) aliran transformasional

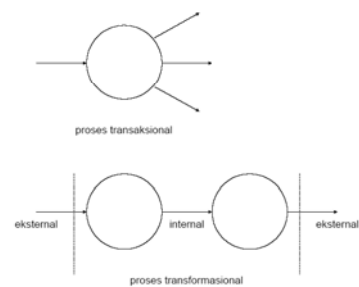


Perancangan Arsitektural (4)

- Jenis aliran informasi:
 - (2) aliran transaksional



Perancangan Arsitektural (5)



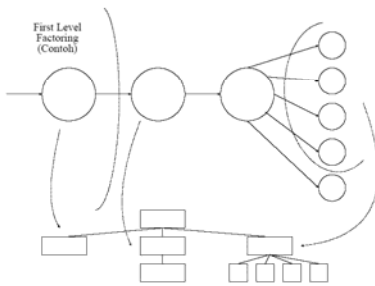
Perancangan Arsitektural (6)

- Langkah Pemetaan (untuk jenis aliran transformasional):
 - kaji ulang model sistem dasarnya
 - kaji ulang dan perhalus DFD-nya
 - tentukan apakah DFD memiliki jenis aliran transformasional dan atau aliran transaksional
 - isolasi pusat transaksi dengan menentukan batas aliran incoming dan outgoing
 - lakukan faktorisasi level satu
 - lakukan faktorisasi level dua
 - perhalus struktur program yang diperoleh dari iterasi tahap pertama ini dengan heuristik.

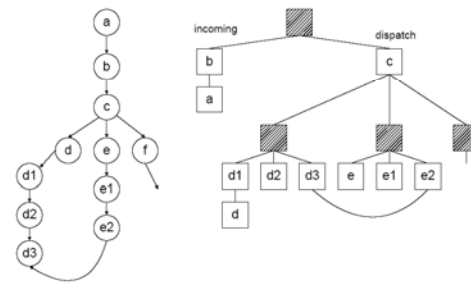
Perancangan Arsitektural (7)

- Langkah Pemetaan (untuk jenis aliran transaksional):
 - kaji ulang model sistem dasarnya
 - kaji ulang dan perhalus DFD-nya
 - tentukan apakah DFD memiliki jenis aliran transformasional dan atau aliran transaksional
 - tentukan pusat transaksi dan jenis aliran di sepanjang setiap jalur aksi (action paths)
 - petakan DFD ke dalam struktur program sesuai proses transaksinya
 - faktorisasi dan perhalus struktur transaksi dan juga struktur disetiap jalur aksi
 - perhalus struktur program yang diperoleh dari iterasi tahap pertama ini dengan heuristik.

Perancangan Arsitektural (8)



Perancangan Arsitektural (9)



Perancangan Arsitektural (10)

- Optimasi rancangan arsitektural
 - kesederhanaan struktur seringkali mencerminkan keindahan dan efisiensi PL
 - optimasi rancangan harus diarahkan pada sesedikit mungkin modul yang terbentuk (dengan tetap mempertimbangkan kriteria perancangan modular efektif) dan sesedikit mungkin struktur data yang rumit/kompleks

Perancangan Arsitektural (11)

- Hasil perancangan arsitektural:
 - Structure Chart yang merepresentasikan gambaran menyeluruh struktur PL / arsitektur PL, beserta seluruh hierarki kendali / passing parameter, yang siap dituliskan dalam bentuk modul program

Perancangan Antarmuka (1)

- Fokus perancangan antarmuka:
 - antarmuka antar modul-modul PL
 - antarmuka antara PL dengan sumber informasi, selain manusia (external entities)
 - antarmuka antara manusia (user) dengan komputernya

Perancangan Antarmuka (2)

- Jenis antarmuka yang diperlukan adalah:
 - antarmuka untuk input parameter process → layar
 - antarmuka untuk output proses → layar
 - antarmuka untuk input data → layar maupun parameter passing
 - antarmuka untuk output data → layar maupun parameter passing
 - antarmuka untuk pesan-pesan → layar

Perancangan Antarmuka (3)

- Hal-hal yang perlu diperhatikan dalam merancang antarmuka di layar:
 - harus konsisten (warna, font, bahasa, dll)
 - memberikan umpan balik ke pengguna
 - meminta verifikasi untuk semua aksi destruktif penting
 - memungkinkan aksi reversal
 - mengurangi jumlah informasi yang harus diingat antar aksi
 - efisiensi dialog, gerak, dan pikiran pengguna (dekomposisi, nilai default, layout)
 - mengelompokkan aktivitas berdasarkan fungsi & mengatur layarnya sesuai dengan pengelompokan tersebut
 - sediakan bantuan (help) yang context sensitif
 - tampilkan info yang sesuai dengan konteks

Perancangan Antarmuka (4)

- Hal-hal yang perlu diperhatikan dalam merancang antarmuka di layar (lanjutan):
 - perhatikan presentasi data:
 - sedikit: form/tabel
 - banyak: graph/chart
 - pelihara konteks visual
 - pesan kesalahan harus berarti
 - gunakan analog display untuk hal yang sesuai, misalnya bar chart
 - minimalkan jumlah aksi masukan yang diperlukan
 - sesuaikan dengan kebutuhan/kebiasaan user, misalnya:
 - clerk - keyboard, manager - mouse
 - clerk - input, decision makers - update/delete

Perancangan Antarmuka (5)

- Perancangan antarmuka dapat dilakukan secara:
 - manual, dilakukan pada kertas
 - bantuan alat bantu, alat bantu pemrograman atau dengan CASEtools (misalnya dengan AppModeller pada Power Designer)
- Hasil perancangan antarmuka adalah:
 - definisi antarmuka modul yang siap untuk diprogram
 - definisi / format rancangan layar yang siap diimplementasikan

Perancangan Prosedural (1)

- Tahapan terakhir pada proses perancangan
- Membentuk algoritma siap program dengan menggunakan dan mengacu pada:
 - struktur data yang terbentuk pada perancangan data
 - struktur modul dan kendali PL yang terbentuk pada Structure Chart yang diperoleh pada saat perancangan arsitektural
 - struktur dan rancangan menu / format tampilan layar yang diperoleh pada perancangan antarmuka

Perancangan Prosedural (2)

- Pada intinya perancangan prosedural harus memperhatikan dua hal utama yaitu:
 - coupling (a measure of the interdependence among software module), yaitu ukuran kekuatan saling kebergantungan antar modul-modul software
 - cohesion (a measure of the relative functional strength of a module), yaitu ukuran kekuatan modul-modul perangkat lunak secara fungsional relatif terhadap modul perangkat lunak itu sendiri

Perancangan Prosedural (3)

- Untuk merancang prosedur/modul alat bantu yang dapat digunakan adalah:
 - flow-chart
 - algoritma/pseudocode/program design language
- Alat bantu tersebut dapat dicapai secara:
 - manual, dilakukan pada kertas atau pada text editor/shape editor
 - dengan alat bantu menggunakan CASE tools misalnya PSpec pada Process Analyst (Power Designer)

Hasil Proses Perancangan

- Hasil proses perancangan perangkat lunak dinyatakan dengan sebuah dokumen yang diberi nama: Software Design Description (SDD).
- Isi SDD adalah paparan sistem yang umumnya disusun berdasarkan:
 - Tujuan
 - tuntutan umum (general requirement)
 - Batasan
 - Asumsi
 - Rancangan data
 - basis data
 - external file
 - Rancangan arsitektural, yaitu faktorisasi modul & deskripsinya
 - Rancangan antarmuka
 - Rancangan prosedural
- SDD menjadi patokan dalam melakukan pemrograman agar modul-modul yang dihasilkan terarah dan dapat dipertanggungjawabkan.

Kesimpulan

- Perancangan PL:
 - perancangan data
 - hasilnya: struktur data dan basis data
 - perancangan arsitektural
 - hasilnya: structure chart
 - perancangan antarmuka
 - hasilnya: definisi parameter passing dan format layar/menu
 - perancangan prosedural
 - hasilnya: algoritma program
- Semua hasil itu harus dituangkan dalam SDD