

Konsep dan Prinsip Desain

Tujuannya adalah untuk menghasilkan suatu model atau representasi dari entitas yang kemudian akan dibangun.

Desain Perangkat Lunak dan Rekayasa Perangkat Lunak

Fase pengembangan terdiri dari tiga langkah yaitu design, code generation (manual or automatic) dan testing.

Setiap langkah melakukan transformasi informasi dengan suatu cara yang akhirnya menghasilkan software komputer yang valid.

Software Requirements

Dijelaskan dengan information domain, functional and performance requirements, dan feed the design step.

Dengan menggunakan satu dari sejumlah metode desain, langkah desain menghasilkan :

- a. Desain data (difokuskan pada definisi dari struktur data)
- b. Desain arsitektur (mendefinisikan hubungan antara elemen struktur utama dari program)
- c. Desain interface
- d. Desain prosedural (mengubah struktur elemen kedalam prosedur software)

Selama desain, kita dapat membuat keputusan yang akan mempengaruhi kesuksesan konstruksi software dan kemudahan maintenance-nya. Desain sangat penting karena dapat menentukan kualitas dari suatu software.

Proses Desain

Software design dibagi dalam 2 tahap :

1. Preliminary Design, Pada tahap ini difokuskan dengan transformasi dari keperluan/kebutuhan ke dalam data dan arsitektur software
2. Detail Design, Difokuskan pada penghalusan representasi arsitektur yang berisi struktur data detail dan algoritma untuk software

Agar dihasilkan desain dengan kriteria yang baik, maka suatu desain haruslah :

1. Memperlihatkan organisasi hirarki yang mengontrol elemen-elemen software
2. Berkenaan dengan modul. Software secara logika terbagi dalam elemen-elemen yang membentuk fungsi dan sub fungsi
3. Berisi representasi yang berbeda dan terpisah dari data dan prosedur
4. Membentuk modul (contoh subroutine dan procedure) yang memperlihatkan karakteristik fungsi yang tidak saling bergantung
5. Diturunkan dengan menggunakan metode perulangan yang didukung oleh informasi yang ada selama analisa kebutuhan software

Evolusi Desain Software

Merupakan suatu proses kontinu yang terus berlangsung selama tiga dekade. Beberapa metodologi telah tumbuh, dan secara umum memiliki karakteristik sebagai berikut :

- a. Mekanisme penerjemahan suatu model analisis ke dalam representasi desain.
- b. Notasi untuk merepresentasikan komponen-komponen fungsional dan interface-nya.
- c. Heuristik bagi penyaringan dan partisi
- d. Pedoman bagi penilaian kualitas

Prinsip desain

Desain perangkat lunak berupa model dan proses. Proses desain adalah serangkaian langkah iteratif yang memungkinkan desainer menggambarkan semua aspek perangkat lunak yang dibangun. Model desain adalah ekivalen rencana arsitek untuk sebuah "rumah", yang dimulai dengan menyajikan totalitas dari hal yang akan dibangun.

Konsep-konsep desain

1. Abstraksi

Jika kita menggunakan suatu solusi modular untuk beberapa masalah, maka beberapa level / tingkat abstraksi dapat ditampilkan / diperlihatkan.

- Pada level tertinggi, suatu solusi berada pada term yang umum dengan menggunakan bahasa natural
- Level yang lebih rendah lebih berorientasi pada prosedur-prosedur

Contoh :

Abstraksi 1

The software will incorporate a computer graphics interface that will enable visual communication with the drafts person and a digitizer interface that replace the drafting board and square. All line and curve drawing, all geometric computation, and all sectioning and auxiliary views will be performed by the CAD Comp.

Abstraksi 2

CAD Software tasks :
user interaction task ;
2-D drawing creation task ;
graphics display task ;
drawing file management task ;
end.

Abstraksi 3

procedure : 2-D drawing creation ;
repeat until (drawing creation task terminates)
do while (digitizer interaction occurs)
digitizer interface task ;
determine drawing request case ;
line : line drawing task ;
circle : circle drawing task ;
...
...

```

end ;
do while (keyboard interaction occurs)
  keyboard interaction task ;
  process analysis / computation case ;
  view : auxiliary view task ;
  section : cross sectioning task ;
  ...
end
...
end repetition ;
end procedure.

```

2. Penyaringan

Penyaringan stepwise adalah strategi desain top-down yang diusulkan oleh Wiklaus Wirth.

Kajian dari konsep tersebut adalah “Pada setiap langkah (penyaringan), satu atau beberapa instruksi dari program yang diberikan didekomposisi ke dalam instruksi-instruksi yang lebih detail. Dekomposisi berurutan atau penyaringan spesifikasi berhenti bila semua instruksi diekspresikan dalam bentuk bahasa pemrograman atau komputer yang mendasar. Jika tugas-tugas disaring, maka data harus disaring juga, didekomposisi atau distruktur, dan adalah wajar untuk menyaring program dan spesifikasi data secara paralel” .

Abstraksi dan penyaringan adalah konsep komplementer. Kedua konsep tersebut membantu desainer dalam menciptakan suatu model desain lengkap jika desain berkembang.

3. Modularitas

Software dibagi ke dalam elemen-elemen terpisah yang dapat dipanggil, yang disebut dengan modul.

Misalkan :

$C(x)$: fungsi kompleksitas dari suatu masalah

$E(x)$: fungsi usaha/waktu yang diperlukan untuk memecahkan suatu masalah

$P1, P2$ = masalah 1, masalah 2

Jika : $C(P1) > C(P2)$ maka : $E(P1) > E(P2)$

Berdasarkan penelitian :

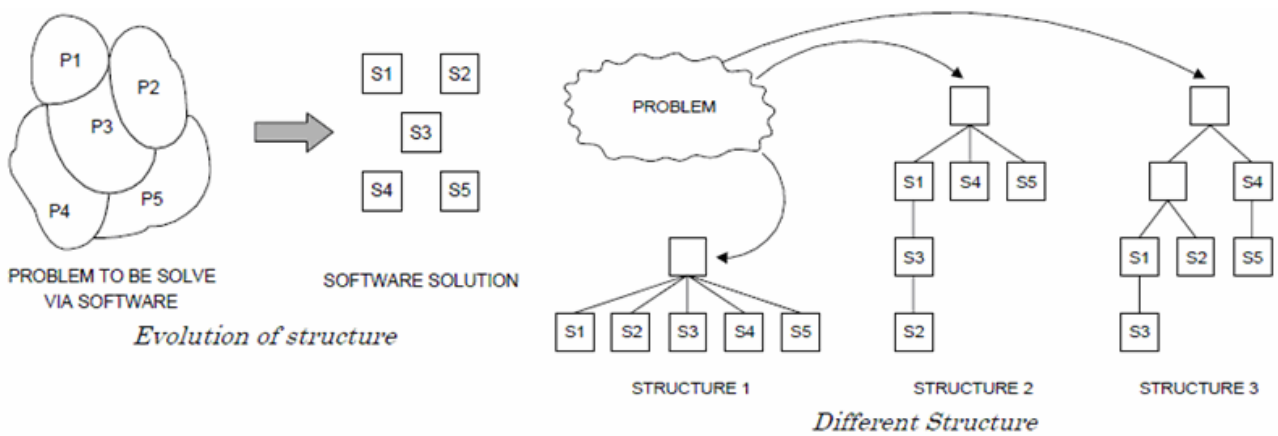
1. $C(P1 + P2) > C(P1) + C(P2)$

2. $E(P1 + P2) > E(P1) + E(P2)$

4. Arsitektur perangkat lunak

Arsitektur perangkat lunak menyinggung 2 karakteristik penting dari sebuah program komputer :

1. Hirarki struktur dari komponen-komponen prosedural (modul)
2. Struktur data



5. Partisi structural

Struktur program harus dipartisi baik secara horizontal maupun vertikal.

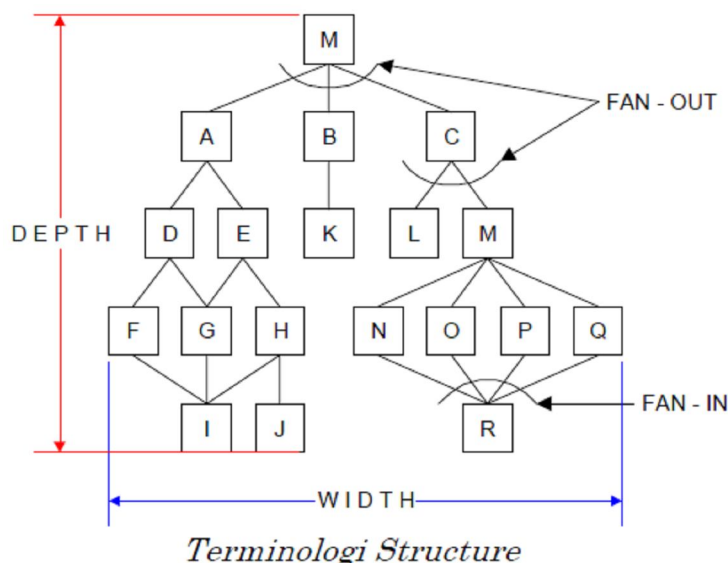
Partisi horizontal menentukan cabang-cabang terpisah dari hirarki modular untuk setiap fungsi program mayor. Keuntungannya :

- Menghasilkan perangkat lunak yang lebih mudah diuji.
- Membawa kepada perangkat lunak yang lebih mudah dipelihara.
- Menghasilkan penyebaran efek samping yang lebih sedikit.
- Menghasilkan suatu perangkat lunak yang lebih mudah untuk diperluas.

Partisi vertikal menyatakan bahwa kontrol dan kerja harus didistribusikan secara top-down dalam arsitektur program.

6. Hirarki Kontrol (Program Structure)

- Program structure menampilkan/menyajikan organisasi (seringkali organisasi hirarki) dari komponen-komponen program (modul-modul) dan mengandung arti hirarki dari kontrol program
- Notasi yang digunakan adalah diagram tree. Biasanya dinamakan structure chart



7. Prosedur perangkat lunak

Prosedur perangkat lunak berfokus pada detail-detail pemrosesan dari masing-masing modul secara individual. Prosedur harus memberikan spesifikasi yang teliti terhadap pemrosesan, mencakup urutan event, poin-poin keputusan nyata, operasi repetitif, dan organisasi struktur data.

8. Struktur data

Struktur data adalah representasi dari hubungan logis antara elemen-elemen data individual.

Contoh :

```
type G = array [1..100] of integer;
```

```
...;
```

```
Procedure S ( var T : G ; n : integer ; sum : integer );
```

```
Var
```

```
  I : integer;
```

```
begin
```

```
  sum := 0;
```

```
  for I:=1 to n do
```

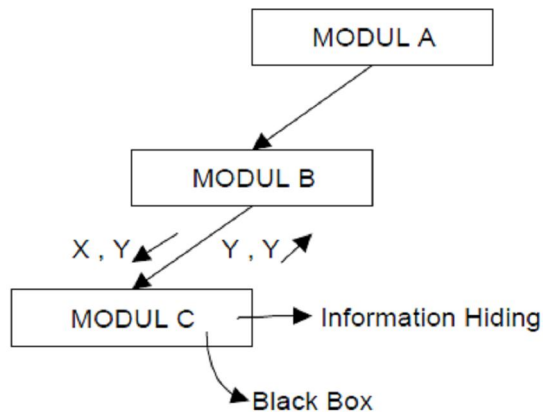
```
    sum := sum + t[i];
```

```
end;
```

9. Penyembunyian informasi

Prinsip penyembunyian informasi menyatakan bahwa modul ditandai dengan keputusan desain tersembunyi dari semua desain lain.

Contoh :



Black Box : input, output, & proses diketahui tetapi proses detail tidak diketahui.

Bagi Modul B, Modul C adalah Black Box