



Rancangan dan Analisa Desain Data Perangkat Lunak



Yang ada pada presentasi ini :

01

Desain Data

02

Desain Arsitektural

03

Desain Antarmuka

04

Transform Mapping

05

Transaction Mapping

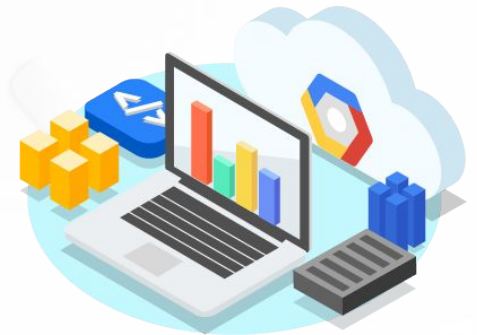




1. Desain Data

1.1 Definisi

Desain data adalah aktivitas pertama dan terpenting dari empat aktivitas desain yang dilakukan selama rekayasa perangkat lunak. Proses pemilihan struktur dalam menentukan desain yang paling efisien sesuai kebutuhan. Tujuan dari desain data adalah untuk mendapatkan struktur data yang baik sehingga diperoleh program yang lebih modular dan mengurangi kompleksitas pengembangan software.



1.2 Prinsip Desain Data

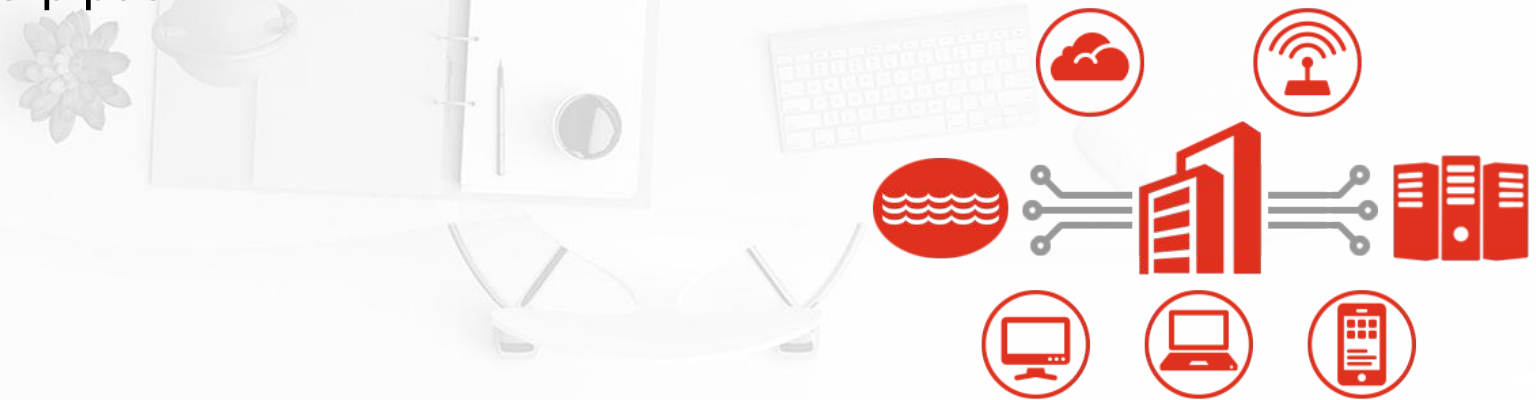
1. Prinsip analisis sistematis yang diaplikasikan pada fungsi dan perilaku harusnya juga diaplikasikan pada data.
2. Semua struktur data dan operasi yang akan dilakukan pada masing-masing struktur data harus diidentifikasi.
3. Kamus data harus dibangun dan digunakan untuk menentukan baik data maupun desain program.
4. Keputusan desain data tingkat rendah harus ditunda sampai akhir proses desain.
5. Representasi struktur data hanya boleh diketahui oleh modul-modul yang menggunakan secara langsung data yang diisikan di dalam struktur tersebut.
6. Pustaka struktur data dan operasi yang berguna yang dapat diaplikasikan pada struktur data tersebut harus dikembangkan.
7. Desain perangkat lunak dan bahasa pemrograman harus mendukung spesifikasi dan realisasi dari tipe-tipe data abstrak.



2. Desain Arsitektur

2.1 Definisi

Desain arsitektur adalah untuk mengembangkan struktur program modular dan merepresentasikan hubungan kontrol antar modul. Metode desain yang disajikan pada bagian ini mendorong rekayasa perangkat lunak untuk berkonsentrasi pada desain arsitektur sebelum mencemaskan masalah perpipaian.



2.2 Fungsi Desain Arsitektur

1. Analisis efektifitas dari desain untuk memenuhi kebutuhan yang diminta. Berguna untuk komunikasikan gambaran sistem dengan *stakeholder* yang berkaitan dengan pembangunan sistem.
2. Mempertimbangkan alternatif lain ketika ada perubahan desain terjadi. Arsitektur memberikan gambaran bagaimana sistem itu saling terkait dan bekerja.
3. Mengurangi risiko yang berkaitan dng pembangunan *software* atau *coding*. Arsitektur memiliki pengaruh besar dalam rekayasa perangkat lunak karena menjelaskan hasil keputusan dalam bentuk desain.



3. Desain Antarmuka

3.1 Definisi

Desain antarmuka memberikan suatu gambaran mengenai struktur program kepada perekraya perangkat lunak. Fokus desain antarmuka adalah sebagai berikut :

1. *Desain interface* antar modul
2. *Desain interface* antara perangkat lunak dan entitas eksternal (produser & konsumen)
3. *Desain interface* manusia dengan komputer



3.2 Fungsi Desain Arsitektur

1. Analisis efektifitas dari desain untuk memenuhi kebutuhan yang diminta. Berguna untuk komunikasikan gambaran sistem dengan *stakeholder* yang berkaitan dengan pembangunan sistem.
2. Mempertimbangkan alternatif lain ketika ada perubahan desain terjadi. Arsitektur memberikan gambaran bagaimana sistem itu saling terkait dan bekerja.
3. Mengurangi risiko yang berkaitan dng pembangunan *software* atau *coding*. Arsitektur memiliki pengaruh besar dalam rekayasa perangkat lunak karena menjelaskan hasil keputusan dalam bentuk desain.



4. Transform Mapping

4.1 Definisi

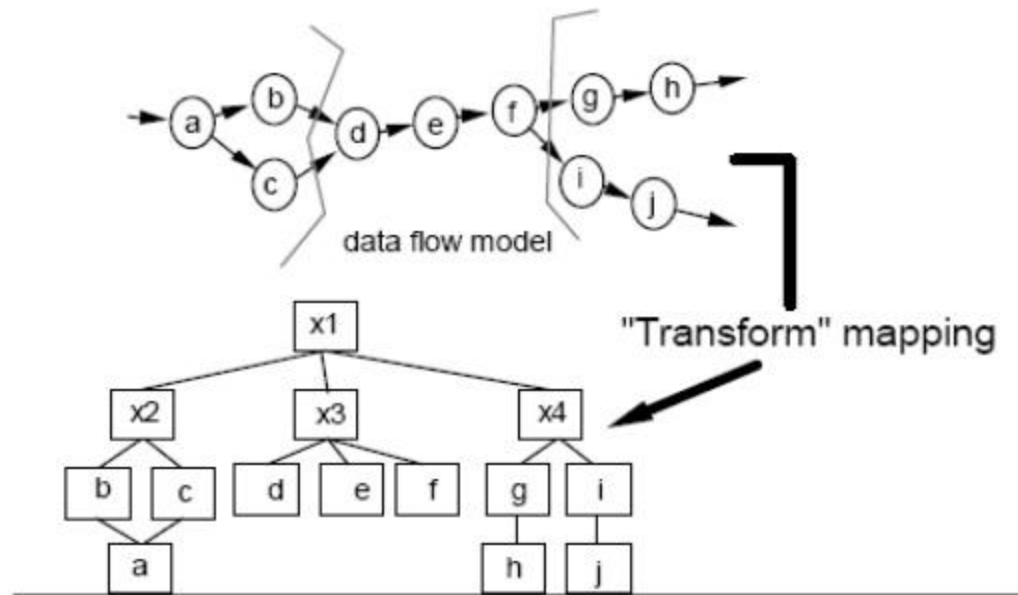
Transform mapping adalah seperangkat langkah desain yang dapat mengubah *data flow diagram* dengan perubahan karakteristik yang nantinya akan dipetakan pada gaya arsitektur tertentu.



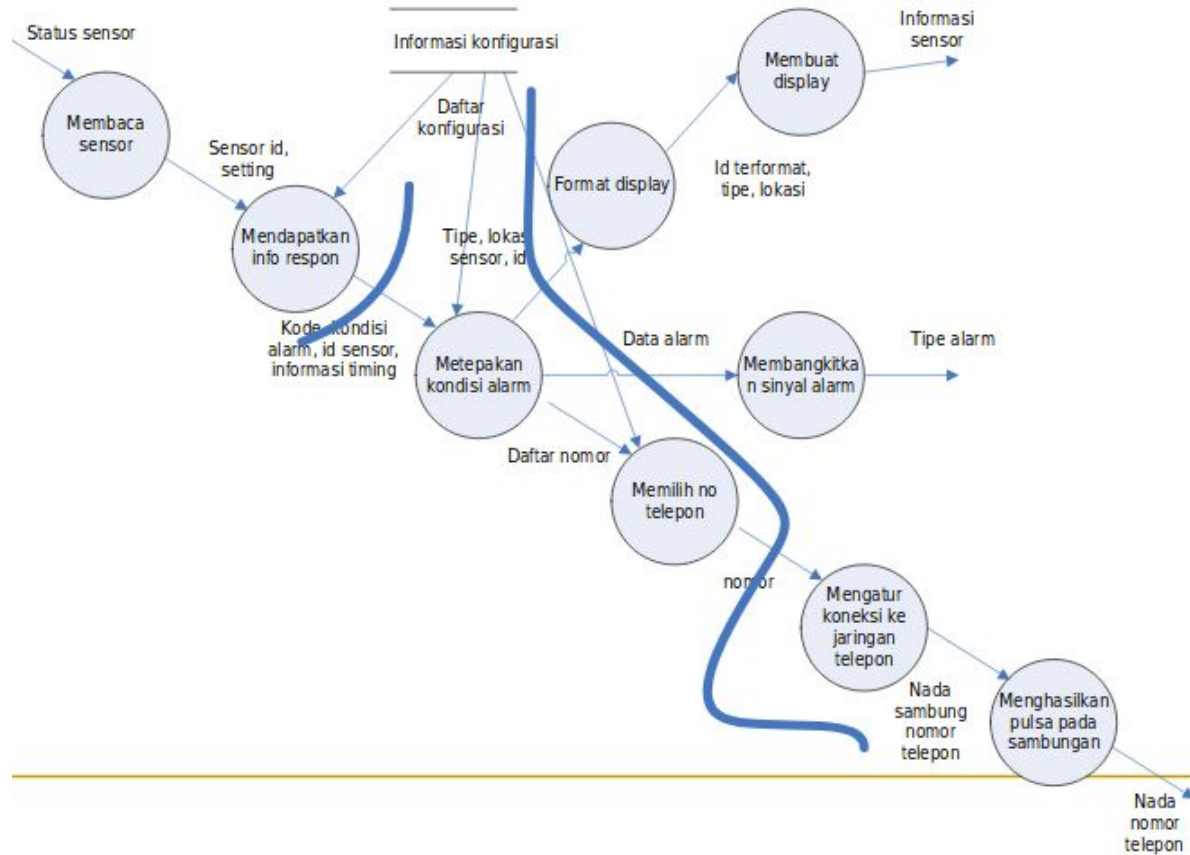
4.2 Langkah-langkah transform mapping

1. Tinjau sisi fundamental dari suatu model sistem
2. Tinjau dan perbaiki data flow diagram perangkat lunak tersebut
3. Menentukan apakah DFD memiliki karakteristik *transform flow* atau *transaction flow*
4. Isolasi pusat transformasi dengan menetapkan batas aliran masuk dan keluar
5. Lakukan "*factoring* tingkat pertama."
6. Lakukan "*factoring* tingkat kedua." Pemfaktoran tingkat kedua dilakukan dengan memetakan transformasi individu DFD menjadi modul yang sesuai dalam arsitektur
7. Perbaiki arsitektur iterasi pertama menggunakan desain heuristik untuk meningkatkan kualitas perangkat lunak.

Contoh Transform Mapping



Contoh Kasus Transform Flow





5. Transaction Mapping

5.1 Fungsi

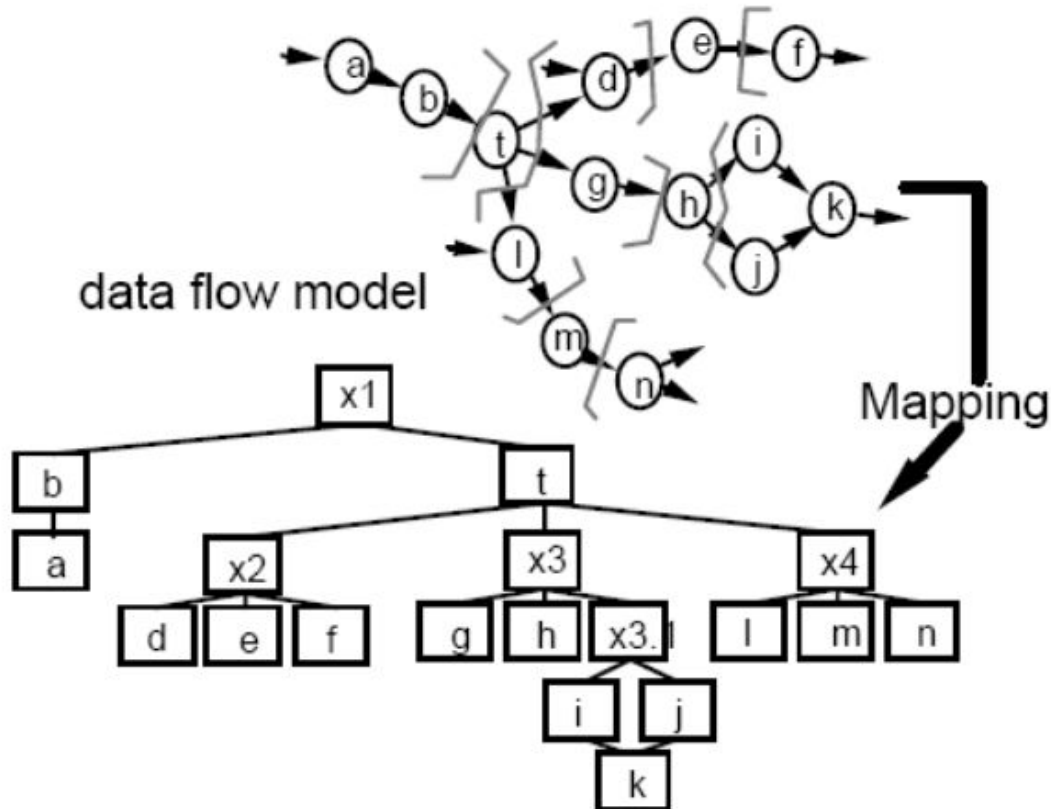
Diterapkan jika sebuah item informasi menyebabkan percabangan, yang disebut transaksi, yang memicu aliran data lain sepanjang salah satu dari beberapa jalur. DFD dipetakan menjadi sebuah struktur yang mengalokasikan kontrol menjadi sebuah sub struktur yang mendapatkan dan mengevaluasi sebuah transaksi.



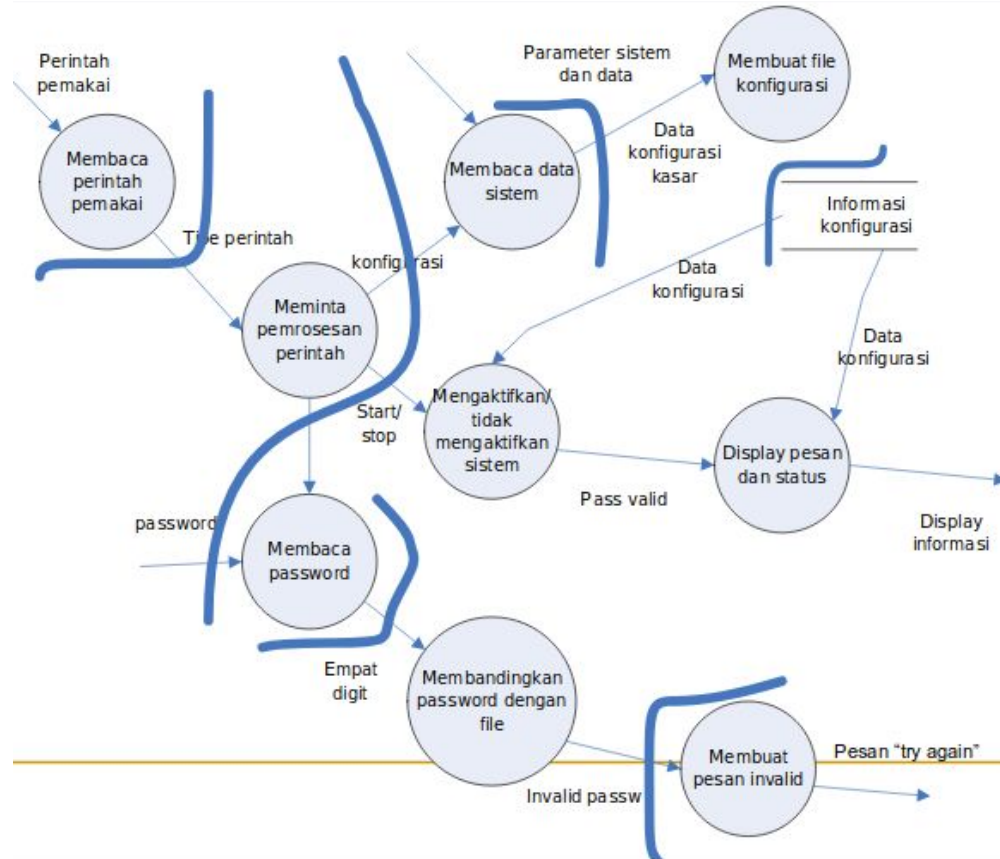
5.2 Langkah-langkah *transaction mapping*

1. Kaji ulang model sistem dasarnya
2. Kaji ulang dan perhalus DFD
3. Tentukan apakah DFD mempunyai aliran *transformation* atau *transaction*
4. Tentukan pusat transaksi dan jenis aliran disepanjang setiap jalur aksi
5. Petakan DFD ke dalam struktur program sesuai dengan proses transaksi
6. Faktorisasi dan perhalus struktur transaksi dan juga struktur di setiap jalur aksi
7. Perhalus struktur program yang diperoleh dari iterasi pertama

Contoh Transaction Mapping



Contoh Kasus Transaction Flow





Referensi

<http://41814120237.blog.mercubuana.ac.id/wp-content/uploads/sites/98/2015/06/Materi-RPL-TM12-Desain-Data-dan-Arsitektur.pdf>

<http://41813320013.blog.mercubuana.ac.id/wp-content/uploads/sites/117/2015/06/Desain-Data-dan-Arsitektur-Perangkat-Lunak.pdf>

<http://www.1000sourcecodes.com/2012/05/software-engineering-transform-mapping.html>

<https://studylibid.com/doc/266126/apl-124-rekayasa-perangkat-lunak-i--software-engineering->

<https://www.slideshare.net/fitrayagami/rpl-9perancangan-arsitektural>

The background image shows a clean, white desk with various items: a large teal circle in the center, a teal desk lamp on the left, a small green succulent, a black monitor on the right, a white keyboard, and a white mouse. The teal circle contains the text.

Terima kasih!

presentasi ini bisa
didapat melalui tautan
berikut : s.id/4qRqC