Kebutuhan Pemodelan

Rekayasa Perangkat Lunak
Chapter 17

1. Pendahuluan

Suatu kebutuhan yang kompleks mungkin diperlukan untuk memperlihatkan bagaimana objek-objek data ditransformasi saat mereka bergerak didalam sistem.

- **Strategi-strategi Model Kebutuhan**

Bentuk dari model kebutuhan adalah :

1. Analis terstruktur

Model ini mempertimbangkan data dan proses-proses yang melakukan transformasi terhadap data tersebut sebagai entitas-entitas yang saling terpisah antara satu dengan yang lainnya

2. Analisis berorientasi objek

Model ini fokus pada teknik-teknik pendefinisian kelas-kelas dan cara bagaimana mereka saling berkolaborasi satu dengan yang lainnya untuk memenuhi kebutuhan-kebutuhan para pengguna sistem/PL yang akan dikembangkan.

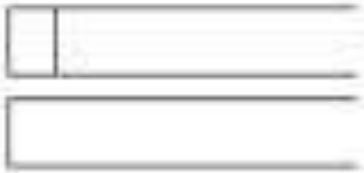
2. Pemodelan Berorientasi Aliran

Pada pemodelan ini menggunakan DFD (Data Flow Diagram) yang memperlihatkan gambaran tentang masukan-proses-keluaran dari suatu sistem/PL yaitu objek-objek data mengalir kedalam PL, kemudian objek-objek data itu akan ditransformasi oleh elemen-elemen pemrosesan, dan objek-objek data hasilnya akan mengalir keluar dari sistem/PL.

Catatan….

Beberapa orang mengatakan bahwa DFD adalah peninggalan lama dan semestinya kita tidak menggunakannya lagi pada praktik-praktik RPL , pandangan ini mengabaikan penggunaan dfd yang masih cukup bermanfaat untuk membuat representasi-representasi pada peringkat analisis.

Simbol-simbol dalam DFD

Simbol	Keterangan
	*External Entity*, merupakan kesatuan di lingkungan luar sistem yang bisa berupa orang, organisasi atau sistem lain
	*Process*, merupakan proses seperti perhitungan aritmatik penulisan suatu formula atau pembuatan laporan
	*Data Store* (Simpan Data), dapat berupa suatu file atau database pada sistem komputer atau catatan manual
	*Data Flow* (arus data), arus data ini mengalir diantara proses, simpan data dan kesatuan luar

Tabel simbol DFD

2.1 Membuat suatu model Aliran Data

- **Dfd pada dasarnya digambarkan dalam bentuk hierarki, yaitu :**

- DFD yang pertama (sering dinamakan DFD level 0 atau diagram konteks) menggambarkan sistem secara keseluruhan
- DFD berikutnya merupakan penghalusan dari diagram konteks, memberikan gambaran yang semakin rinci dari diagram konteks, dan hal ini akan berlanjut ke peringkat-peringkat selanjutnya.

- **Pada penghalusan DFD dilakukan suatu dekomposisi fungsional implisit dari sistem/PL yang akan dikembangkan.**

  Hasilnya : penghalusan data yang terkait saat data tersebut mengalir melewati proses-proses yang ada dalam sistem/PL.

  penghalusan ini berlanjut sampai masing-masing gelembung melakukan sebuah fungsi sederhana hingga proses yang diperlihatkan melakukan fungsi-fungsi yang dapat dengan mudah diimplementasikan sebagai sebuah komponen program

2.2 Membuat Model Aliran Kendali

Untuk memilih kandidat yang potensial, disarankan :

- Mendaftarkan semua yang akan dibaca oleh sistem/PL
- Daftarkan semua kondisi yang mungkin menghentikan/menunda jalannya operasi-operasi dalam sistem/PL
- Daftarkan semua tombol pengendalian yang dipicu oleh aksi operator
- Daftar semua kondisi data
- Perhatikan pemisahan berdasarkan kata benda dan kata kerja
- Deskripsikan perilaku sistem/PL
- Fokus pada pengabaian-pengabaikan yang mungkin terjadi.

Dalam hal ini aplikasi selain memerlukan pemodelan aliran dsts juga memerlukan pemodelan aliran kendali

2.3 Spesifikasi Kendali

Spesifikasi kendali atau CSPEC (*Control Spesification*) pada dasarkan mempresentasikan perilaku sistem/PL (pada peringkat darimana ia telah dirujuk), dengan cara :

1. Memuat diagram state yang merupakan spesifikasi berurutan dari suatu perilaku
2. Memiliki tabel aktivasi program

2.4 Spesifikasi Proses

Spesifikasi proses (*process specification)* pada dasarnya dapat digunakan untuk mendeskripsikan semua proses model aliran yang muncul pada peringkat akhir penghalusan.

Isi dari spesifikasi proses adalah :

- Teks berbentuk narasi
- Deskripsi-deskripsi algoritma prosesdalam bentuk Bahasa perancangan program (yang berisi Persamaan-persamaan matematika, Tabel-table atau diagram aktifitas UML)

3. Membuat Model Perilaku

Pada dasarkan menggambarkan bagaimana PL akan berperilaku dalam menanggapi event-event yang datang dari arah luar atau bagaimana PL akan berperilaku terhadap rangsangan-rangsangan yang muncul dari arah luar.

Langkah-langkah membuat model perilaku adalah :

1. Mengidentifikasi event-event menggunakan use case
2. Repsentasi keadaan

3.1 Mengidentifikasi event-event menggunakan *usecase*

Suatu event terjadi setiap saat sistem dan actor-aktornya saling bertukar informasi.
Sebuah actor seharusnya diidentifikasi untuk masing-masing event, informasi yang diubah dilkukan pencatatan, dan tiap-tiap kondisi serta batasan yang ada harus didaftarkan.

3.2 Representasi Keadaan (state)

2 karakteristik keadaan yang harus dipertimbangkan :
1. Keadaan masing-masing kelas saat kelas-kelas itu melaksanakan fungsinya.
2. Keadaan sistem saat sistem itu diobservasi dari luar saat sistem itu melaksanakan fungsinya.

Komponen-komponen diagram state adalah :

1. Diagram state untuk kelas analisis
   Dalam hal ini mereprentasikan keadaan-keadaan(state) aktif u tuk masing-masing kelas dan event-event(pemicu) yang menyebabkan perubahan-perubahan diantara state aktif
2. Diagram urutan aksi-aksi (sequence diagram)
   Urutan aksi dalam UML bermanfaat untuk memperlihatkan bagaimana event-event yang terjadi bisa mengakibatkan transisi dari suatu objek ke objek lainnya.

Questions & Discussion