

# MODUL

## KEBUTUHAN SISTEM PERANGKAT LUNAK

### A. Tujuan Pembelajaran

1. Peserta didik dapat menjelaskan definisi kebutuhan perangkat lunak
2. Peserta didik dapat menjelaskan kategori kebutuhan perangkat lunak
3. Peserta didik dapat menerapkan tahapan kebutuhan perangkat lunak

### B. Materi Pembelajaran

#### 1. Definisi Kebutuhan Perangkat Lunak

Menurut kamus Webster seperti dikutip oleh Davis (DAV93), kebutuhan adalah sesuatu yang disyaratkan, sesuatu yang diinginkan atau diperlukan. Sedangkan menurut IEEE, kebutuhan adalah:

- Kondisi atau kemampuan yang diperlukan pemakai untuk menyelesaikan suatu persoalan atau untuk mencapai tujuan.
- Kondisi atau kemampuan yang harus dimiliki atau dimiliki oleh sistem atau komponen sistem untuk memenuhi kontrak, standar, spesifikasi, atau dokumen formal lainnya.

Kebutuhan perangkat lunak adalah kondisi, kriteria, syarat atau kemampuan yang harus dimiliki oleh perangkat lunak untuk memenuhi apa yang disyaratkan atau diinginkan pemakai. Sedangkan analisis kebutuhan (*requirements analysis*) merupakan langkah awal untuk menentukan gambaran perangkat yang akan dihasilkan ketika pengembang melaksanakan sebuah proyek pembuatan perangkat lunak. Perangkat lunak yang baik dan sesuai dengan kebutuhan pengguna sangat tergantung pada keberhasilan dalam melakukan analisis kebutuhan. Untuk proyek-proyek perangkat lunak yang besar, analisis kebutuhan dilaksanakan setelah aktivitas sistem *information engineering* dan *software project planning*.

Analisis kebutuhan perangkat lunak dapat diartikan sebagai:

- Proses mempelajari kebutuhan pemakai untuk mendapatkan definisi kebutuhan sistem atau perangkat lunak.
- Proses untuk menetapkan fungsi dan unjuk kerja perangkat lunak, menyatakan antarmuka perangkat lunak dengan elemen-elemen sistem lain dan menentukan kendala yang harus dihadapi oleh perangkat lunak.

Analisa kebutuhan yang baik belum tentu menghasilkan perangkat lunak yang baik, tetapi analisa kebutuhan yang tidak tepat menghasilkan perangkat yang tidak berguna. Mengetahui adanya kesalahan pada analisis kebutuhan pada tahap awal memang jauh lebih baik, tetapi kesalahan analisis kebutuhan yang diketahui ketika sudah memasuki penulisan kode atau pengujian, bahkan hampir masuk dalam tahap penyelesaian merupakan malapetaka besar bagi pembuat perangkat lunak. Biaya dan waktu yang diperlukan akan menjadi sia-sia.

Analisa kebutuhan adalah suatu proses untuk mendapatkan informasi, model, spesifikasi tentang perangkat lunak yang diinginkan klien/pengguna. Kedua belah

pihak, yaitu klien dan pembuat perangkat lunak terlibat aktif dalam tahap ini. Informasi dari klien yang akan menjadi acuan untuk melakukan desain perangkat lunak.

Analisis kebutuhan merupakan satu diantara banyak aktivitas kritis pada proses rekayasa kebutuhan perangkat lunak untuk memahami ranah permasalahan dari sistem yang berjalan dan ranah solusi dari sistem yang akan dibuat (Yen et al, 1998). Ada tiga faktor yang harus dipenuhi ketika melakukan analisis kebutuhan ini, yaitu lengkap, detail, dan benar. Lengkap artinya semua yang diharapkan oleh klien telah didapatkan oleh pihak yang melakukan analisis. Detail maksudnya adalah berhasil mengumpulkan informasi yang terperinci. Semua data dari analisis kebutuhan ini haruslah benar, sesuai apa yang dimaksud oleh klien, bukan benar menurut apa yang dipikirkan oleh pihak analisis.

Tujuan analisis kebutuhan diantaranya:

- Memahami masalah yang akan dibuat perangkat keduanya secara menyeluruh (komprehensif).
- Mendefinisikan apa yang harus dikerjakan oleh perangkat lunak untuk memenuhi keinginan pemakai.

Ada tiga tujuan utama dari proses analisis kebutuhan yang dapat diformulasikan sebagai berikut:

- Mengelola hasil elistasi kebutuhan untuk menghasilkan dokumen spesifikasi kebutuhan yang isi keseluruhannya sesuai dengan apa yang dingiinkan pengguna (Liu and Yen, 1996).
- Mengembangkan persyaratan kualitas yang memadai dan rinci, dimana para manajer dapat membuat pekerjaan proyek yang realistis dan staf teknis dapat melanjutkan dengan perancangan, implementasi dan pengujian (Wieggers, 2003).
- Membangun pemahaman tentang karakteristik ranah permasalahan dan sekumpulan kebutuhan untuk menemukan solusi.

## 2. Kategori Kebutuhan Perangkat Lunak

Secara kategoris, ada tiga buah jenis kebutuhan perangkat lunak (IEEE93):

### a. Kebutuhan Fungsional (*Functional Requirement*)

Disebut juga kebutuhan operasional, yaitu kebutuhan yang berkaitan dengan fungsi atau proses transformasi yang harus mampu dikerjakan oleh perangkat lunak. Sebagai contoh:

- Perangkat lunak harus dapat menyimpan semua rincian data pesanan pelanggan.
- Perangkat lunak harus mampu mencetak laporan penjualan sesuai periode yang diinputkan.
- Perangkat lunak harus mampu menyajikan informasi jalur pengiriman terpendek.

### b. Kebutuhan Antarmuka (*Interface Requirement*)

Kebutuhan antarmuka yang menghubungkan perangkat lunak dengan elemen perangkat keras, perangkat lunak, atau basis data. Sebagai contoh:

- Akses ke basis data menggunakan ODBC (*Open Data Base Connectivity*).
  - Perangkat untuk memasukkan data menggunakan *keyboard*, *mouse*, dan *scanner*.
- c. Kebutuhan Unjuk Kerja (*Performance Requirement*)
- Kebutuhan yang menetapkan karakteristik unjuk kerja yang harus dimiliki oleh perangkat lunak, seperti kecepatan, ketepatan, atau frekuensi. Sebagai contoh:
- Waktu tanggap penyajian informasi maksimal selama satu menit.
  - Perangkat lunak harus mampu mengolah data sampai 1 juta *record* untuk setiap transaksi.
  - Perangkat lunak harus dapat digunakan secara *multi user* sesuai otoritas yang diberikan kepada masing-masing pemakai.

### 3. Tahapan Kebutuhan Perangkat Lunak

Tahap kebutuhan perangkat lunak dimulai dengan (DAV93):

- a. Adanya masalah yang membutuhkan penyelesaian:
- Orientasi aplikasi, misalnya *inventory*
  - Orientasi bisnis, misalnya produk baru, peralihan pendapatan
  - Orientasi peningkatan produk, misalnya pemeliharaan
- b. Munculnya ide untuk membuat sebuah perangkat lunak baru.

Tahap kebutuhan berakhir apabila deskripsi lengkap dari perilaku eksternal perangkat lunak yang akan dibangun sudah didapat, termasuk dokumentasi seluruh antarmuka perangkat lunak dengan lingkungannya (perangkat keras, perangkat lunak lain, pemakai) yang dicatat dalam spesifikasi kebutuhan perangkat lunak.

Secara teknis pelaksanaan pekerjaan analisis kebutuhan perangkat lunak pada dasarnya terdiri dari urutan aktivitas:

- 1) Mempelajari dan memahami persoalan/masalah
- 2) Mengidentifikasi kebutuhan pemakai
- 3) Mendefinisikan kebutuhan perangkat lunak
- 4) Membuat dokumen spesifikasi kebutuhan
- 5) Mengkaji ulang (*review*) kebutuhan

#### **Mempelajari dan Memahami Masalah**

Pada tahap ini, masalah yang akan dibuat perangkat lunaknya dipelajari sehingga dapat ditentukan:

- Siapa pemakai yang akan menggunakan perangkat lunak
- Dimana perangkat lunak akan digunakan
- Pekerjaan apa dari pemakai yang akan dibantu oleh perangkat lunak
- Dari dan sampai mana cakupan pekerjaan tersebut, dan bagaimana mekanisme pelaksanaannya
- Apa yang menjadi kendala atau keterbatasannya dilihat dari sisi teknologi yang akan digunakan atau dari sisi hukum dan standar

Cara yang digunakan untuk dapat memahami masalah biasanya adalah:

- Wawancara dengan pemakai
- Observasi atau pengamatan lapangan
- Kuesioner
- Mempelajari referensi atau dokumen-dokumen yang digunakan, seperti dokumen hasil analisis dan perancangan sistem

Hasil pemahaman masalah tersebut selanjutnya digambarkan dalam bentuk model-model tertentu sesuai dengan jenis masalahnya. Sebagai contoh, untuk masalah bisnis dapat menggunakan *flowmap* atau *business use case*, sementara untuk masalah matematika dapat menggunakan, misalnya, graf.

### **Mengidentifikasi Kebutuhan Pemakai**

Sebenarnya, tahap identifikasi kebutuhan pemakai (*user requirement*) ini pada prakteknya dilaksanakan bersamaan dengan pemahaman masalah. Cara yang digunakan pun relatif sama. Hanya saja, substansi yang ditanyakan biasanya adalah:

- Data atau informasi apa yang akan diproses
- Fungsi apa yang diinginkan
- Kelakuan sistem apa yang diharapkan
- Antarmuka apa yang tersedia (*user interface*, *hardware interface*, *software interface*, dan *communications interface*)

Untuk dapat menangkap kebutuhan pemakai dengan baik, utamanya kesamaan persepsi, dibutuhkan:

- Komunikasi dan *brainstorming* yang insentif
- *Prototype* perangkat lunak, atau *screen snapshot*
- Data atau dokumen yang lengkap

### **Mendefinisikan Kebutuhan Perangkat Lunak**

Saat mengidentifikasi kebutuhan pemakai, informasi yang diperoleh belum terstruktur. Pemakai akan mengungkapkan apa yang dibutuhkannya dengan bahasa sehari-hari yang biasa digunakan pemakai. Sebagai contoh, ungkapan kebutuhan pemakai di bagian akuntansi, misalnya:

- Saya ingin data yang dimasukkan oleh bagian penjualan bisa langsung dijurnal.
- Informasi neraca bisa saya lihat kapan saja.

Pada tahap ini, kebutuhan pemakai yang belum terstruktur tersebut dianalisis, diklasifikasikan, dan diterjemahkan menjadi kebutuhan fungsional, antarmuka, dan unjuk kerja perangkat lunak. Sebagai gambaran, kebutuhan “data yang dimasukkan oleh bagian penjualan dapat langsung dijurnal” setelah dianalisis, diklasifikasikan, dan diterjemahkan, mungkin memberikan hasil:

1) Kebutuhan fungsional:

- *entry* dan rekam data transaksi penjualan.
- *retrieve* nilai transaksi penjualan untuk periode tertentu (sesuai periode yang diinputkan melalui *keyboard*).

- rekam nilai akumulasi transaksi penjualan periode tertentu ke jurnal umum berikut *account* pasangannya (kas).
- 2) Kebutuhan antarmuka:
    - antarmuka pemakai untuk merekam data penjualan.
    - antarmuka pemakai untuk menyajikan dan menjurnal informasi nilai transaksi penjualan periode tertentu.
    - jaringan lokal untuk menghubungkan perangkat lunak aplikasi di bagian penjualan dengan perangkat lunak di bagian akuntansi.
  - 3) Kebutuhan unjuk kerja
    - ada otoritas pemakaian perangkat lunak dan akses data.
    - proses jurnal hanya dapat dilakukan sekali setelah data transaksi penjualan direkam.

Selanjutnya, kebutuhan tersebut diubah menjadi model atau gambar tertentu dengan memanfaatkan teknik analisis dan alat bantu tertentu. Sebagai gambaran, kebutuhan fungsional dapat dimodelkan dengan menggunakan:

- Data Flow Diagram (DFD), kamus data, dan spesifikasi proses jika menggunakan teknik terstruktur.
- Diagram Use Case dan skenario sistem jika menggunakan pendekatan objek.

### **Membuat Dokumen Spesifikasi Kebutuhan**

Semua kebutuhan yang telah didefinisikan selanjutnya dibuatkan dokumentasinya, yaitu Spesifikasi Kebutuhan Perangkat Lunak (SKPL) atau *Software Requirements Specification* (SRS). SKPL yang dibuat harus dapat menyatakan secara lengkap apa yang dapat dilakukan oleh perangkat lunak, termasuk deskripsi lengkap dari semua antarmuka yang digunakan. SKPL bisa terdiri dari banyak dokumentasi yang saling melengkapi.

### **Mengkaji Ulang (Review) Kebutuhan**

Proses untuk memeriksa (validasi) SKPL apakah sudah konsisten, lengkap, dan sesuai dengan apa yang diinginkan pemakai. Proses ini mungkin dilakukan lebih dari satu kali. Dan sering kali muncul kebutuhan-kebutuhan baru dari pemakai. Untuk itu, diperlukan negosiasi antar pihak pengembang dengan pemakai sesuai prinsip "*win-win solution*" sampai kebutuhan tersebut dapat disepakati kedua belah pihak.

### **C. Tugas Pembelajaran**

- 1) Sebutkan dan jelaskan 3 faktor yang harus dipenuhi ketika melakukan analisis kebutuhan!
- 2) Mengapa dalam tahapan analisis kebutuhan perangkat lunak terdapat tahapab mempelajari dan memahami persoalan/masalah?
- 3) Apa yang harus dilakukan dalam menyamakan persepsi untuk menangkap kebutuhan pemakai dengan baik?
- 4) Jelaskan model gambaran yang digunakan dari kebutuhan fungsional!

#### **D. Referensi**

- 1) Juita, Rani. (2015). *Analisis Kebutuhan Perangkat Lunak*. Jakarta: Universitas Mercu Buana Jakarta, Fakultas Ilmu Komputer Jurusan Sistem Informasi.
- 2) Liana, Linda. (2015). *Analisis Kebutuhan Perangkat Lunak*. Jakarta: Universitas Mercu Buana Jakarta, Program Studi Sistem Informasi.
- 3) Wardani, Ratna. (2012). *Rekayasa Perangkat Lunak*. Yogyakarta: Universitas Negeri Yogyakarta, Jurusan Pendidikan Teknik Elektronika.