

## Chap 5

# REKAYASA PERANGKAT LUNAK

Siti Sa'uda, M.Kom.

## SPESIFIKASI KEBUTUHAN PERANGKAT LUNAK

1. **Objektif SRS**
2. **Syarat Pembentukan SRS**
3. **2 aspek yang Harus Terlihat didalam SRS**
4. **Atribut Suatu SRS**
5. **Contoh Layout Dokumen SRS**

# Pendahuluan

- Spesifikasi kebutuhan perangkat lunak atau *Software Requirements Specification (SRS)* adalah sebuah dokumen yang berisi pernyataan lengkap dari apa yang dapat dilakukan oleh perangkat lunak, tanpa menjelaskan bagaimana hal tersebut dikerjakan oleh perangkat lunak.
- Suatu SRS harus mencantumkan tentang deskripsi dengan lingkungannya. Mencakup antarmuka untuk perangkat keras, perangkat lunak, komunikasi dan pemakai.

- **SRS bisa terdiri dari banyak dokumentasi yang saling melengkapi. Suatu SRS harus dapat :**
  1. Menguraikan definisi masalah
  2. Menguraikan masalah dengan tepat dengan cara yang tepat pula



- **Objektif SRS**

1. Persetujuan kerja dengan pelanggan
2. Daftar kebutuhan teknis yang harus dipenuhi oleh perangkat lunak

- **Syarat Pembentukan SRS**

1. Mudah diidentifikasi
2. Diuraikan dengan jelas, *simple, sederhana dan concise (Jelas, tidak ambiguous)*
3. Bisa divalidasi dan bisa dites (*test reliable, test accessible*).
4. Mampu untuk ditelusuri kembali (*traceability*)

- **Hindari hal-hal berikut saat pembentukan SRS**
  1. *Over specification (penjelasan berlebih dan berulang-ulang sehingga menjadi tidak jelas)*
  2. Tindakan *unconcistency*
  3. *Ambiguity dalam kata atau kalimat*
  4. Menuliskan “mimpi-mimpi” , yaitu hal-hal yang tidak bisa dilakukan

- Dalam Suatu SRS ada 2 aspek yang harus bisa dilihat :
  1. Fungsi
    - Menjelaskan fungsi dari perangkat lunak (digunakan untuk apa keperluan apa), sifat lunak dan datanya.
  2. Non-Fungsi
    - *a. Dependability*
      - • *reliability*
      - • *maintainability*
      - • *security*
      - • *integrity*
    - *b. Ergonomic*
    - *c. Performance*
    - *d. Constraint*



- **Atribut Suatu SRS**

1. Benar (*correct*)

Jika salah (*incorrect*), artinya spesifikasi yang ditulis adalah bukan yang diinginkan.

2. Tepat (*precise*)

Berpengaruh pada hasil perancangan dan pembuatan *software requirements design (SRD)*.

3. *Unambiguouity*

Setiap permintaan harus punya satu interpretasi, atau hanya ada satu arti dalam satu kalimat.

#### 4. Lengkap (*complete*)

Lengkap jika dilihat dari dua sudut pandang :

- • Dokumen membuat tabel isi, nomor halaman, nomor gambar, nomor tabel, dan sebagainya.
- • Tidak ada bagian yang hilang (*to be define*) yaitu tulisan yang akan didefinisikan kemudian

#### 5. Bisa diverifikasi (*verifiable*)

*Bisa diperiksa dan dicek kebenarannya. Setiap kebutuhan selalu dimulai dengan dokumen yang bisa diperiksa.*

#### 6. Konsisten

Nilai-nilai kebutuhan harus tetap sama baik dalam karakteristik maupun spesifik misalnya diminta A tetap ditulis A.

## 7. *Understandable*

Dapat dimengerti oleh pemrograman, analisis sistem atau *sistem engineer*

## 8. Bisa dimodifikasi (*modifiedable*)

*Bisa diubah-ubah dan pengubahannya sangat sederhana tetapi tetap konsisten dan lengkap.*

## 9. Dapat ditelusuri (*traceable*)

Jika ditelusuri, harus tahu mana bagian yang diubah

## 10. Harus dapat dibedakan bagian *what* (*bagian spesifikasi*) dan *how* (*bagian yang menjelaskan bagaimana menjelaskan what tadi*)

11. Dapat mencakup dan melingkupi seluruh sistem
12. Dapat melingkupi semua lingkungan operasional, misalnya interaksi fisik dan operasional.
13. Bisa menggambarkan sistem seperti yang dilihat oleh pemakai.
14. Harus toleran (bisa menerima) terhadap ketidaklengkapan, ketidakpastian (*ambiguous*) dan ketidak konsistenan.
15. Harus bisa dilokalisasi dengan sebuah *coupling*, yaitu hubungan ketergantungan antara dua model yang tidak terlalu erat.



- Ada 9 macam orang yang terlibat dalam pembuatan SRS :
  1. Pemakai (*user*)  
Yang mengoperasikan / menggunakan produk final dari perangkat lunak yang dibuat.
  2. Client  
Orang atau perusahaan yang mau membuat sistem (yang menentukan).
  3. Sistem analyst (*sistem engineer*)  
Yang biasa melakukan kontak teknik pertama dengan client. Bertugas menganalisis persoalan, menerima *requirement* dan *menulis requirement*.

#### 4. Software engineer

Yang bekerja setelah kebutuhan perangkat lunak dibuat (bekerja sama dengan sistem engineer berdasarkan SRS)

#### 5. Programmer

Menerima spesifikasi perancangan perangkat lunak, membuat kode dalam bentuk modul, menguji dan memeriksa (tes) modul.

#### 6. Test integration group

Kumpulan orang yang melakukan tes dan mengintegrasikan modul.

## 7. Maintenance group

Memantau dan merawat performansi sistem perangkat lunak yang dibuat selama pelaksanaan dan pada saat modifikasi muncul (80% dari pekerjaan).

## 8. Technical Support

Orang-orang yang mengelola (*manage*) pengembang perangkat lunak, termasuk konsultan atau orang yang mempunyai kepandaian lebih tinggi.

## 9. Staff dan Clerical Work

Bertugas mengetik, memasukkan data dan membuat dokumen.

- Keberhasilan pengembangan perangkat lunak bisa dilihat dari 10 aspek atau titik pandang, yaitu :
  1. Ketelitian dari pembuatnya
  2. Kualitas dari spesifikasi perangkat lunak yang dihasilkan (Baik, jika ada sedikit kesalahan).
  3. Integritas
  4. Ketelitian
  5. Proses Pembuatan yang mantap
  6. Mudah dikembangkan
  7. Jumlah versi yang tidak banyak



8. Ketelitian dari model pengembangan yang digunakan untuk meramal atribut perangkat lunak
9. Efektivitas rencana tes dan integrasi
10. Tingkat persiapan untuk sistem perawatan (mempersiapkan pencarian *bugs*)

# Contoh Layout Dokumen SRS

## 1. PENDAHULUAN

- 1.1. Tujuan
- 1.2. Ruang Lingkup
- 1.3. Definisi
- 1.4. Referensi
- 1.5. Sistematika

## 2. DESKRIPSI UMUM

- 2.1. Perspektif
- 2.2. Kegunaan
- 2.3. Karakteristik Pengguna
- 2.4. Batasan-batasan
- 2.5. Asumsi dan Ketergantungan

### 3. SPESISIKASI KEBUTUHAN

#### – 3.1. Kebutuhan Fungsional

- 3.1.1. Pendahuluan
- 3.1.2. Input
- 3.1.3. Proses
- 3.1.4. Output

#### – 3.2. Kebutuhan Antarmuka Eksternal

- 3.2.1. Antarmuka Pengguna
- 3.2.2. Antarmuka Perangkat Keras
- 3.2.3. Antarmuka Perangkat Lunak
- 3.2.4. Antarmuka Komunikasi

- 3.3. Kebutuhan Performansi
- 3.4. Kendala Desain
  - 3.4.1. Standard Compliance
  - 3.4.2. Perangkat Keras
- 3.5. Atribut
  - 3.5.1. Keamanan Sistem
  - 3.5.2. Pemeliharaan
- 3.6. Kebutuhan Lain
  - 3.6.1. Database
  - 3.6.2. Pengoperasian
  - 3.6.3. Penyesuaian Tempat



**Thank You !**

