

Panduan Penulisan versi 1.1

Spesifikasi Kebutuhan Perangkat Lunak (*Software Requirements Specification, SRS*)

Pengantar

SRS adalah spesifikasi suatu produk perangkat lunak dengan kemampuan tertentu yang beroperasi pada lingkungan tertentu. SRS berisi daftar kemampuan (kebutuhan fungsional) yang harus disediakan oleh perangkat lunak tersebut, beserta prasyarat dan kekangan operasinya (kebutuhan non-fungsional).

Perangkat lunak bisa merupakan bagian dari sistem yang lebih besar. SRS perangkat lunak seperti ini juga mencakup kebutuhan antarmuka dengan bagian-bagian sistem yang lain. SRS harus mematuhi kebutuhan sistem yang lebih besar.

Dokumen ini mengadopsi IEEE Standard 830-1998 dan IEEE 29418-2011.

Pihak-pihak yang terlibat

Tiga pihak yang terlibat dalam pengembangan perangkat lunak adalah

- Peminta (pelanggan, klien) : pihak yang membiayai pengembangan dan biasanya menentukan kebutuhan perangkat lunak.
- Penyedia: pihak yang mengembangkan perangkat lunak bagi pihak peminta.
- Pengguna: pihak yang langsung mengoperasikan perangkat lunak.

SRS ditulis oleh pihak peminta dan pihak penyedia secara bersama-sama karena

- Peminta biasanya tidak mengetahui proses pengembangan perangkat lunak.
- Penyedia biasanya tidak mengetahui permasalahan dan sistem yang diinginkan peminta.

SRS menjadi kesepakatan (kontrak) antara pihak peminta dan penyedia mengenai kemampuan perangkat lunak yang akan dikembangkan. Dalam penulisannya, pihak pengguna sering dilibatkan untuk mendapatkan informasi rinci mengenai fungsionalitas atau fitur yang dibutuhkan saat operasionalnya.

Sistematika SRS

Sistematika SRS adalah sebagai berikut.

Halaman judul

Riwayat versi

Tuliskan riwayat versi dan tanggalnya

Daftar isi

1. Pengantar

1.1. Kegunaan

Sebutkan kegunaan SRS ini

Sebutkan pembaca sasaran SRS ini

1.2. Lingkup

Sebutkan nama produk yang akan dikembangkan

Sebutkan tujuan, kegunaan dan manfaat produk

1.3. Definisi

Buat daftar istilah dan definisinya yang digunakan dalam SRS ini

Buat daftar singkatan, kepanjangan dan definisinya

1.4. Referensi

Sebutkan daftar referensi/rujukan dalam SRS (sebagaimana daftar pustaka)

1.5. Sistematika

Berikan gambaran isi SRS ini

2. Gambaran Umum

2.1. Perspektif produk

Jelaskan posisi produk terhadap produk atau sistem lain yang lebih besar.

Gambarkan diagram blok yang menyatakan posisi dan hubungannya dengan produk lain.

Jika produk berdiri sendiri (stand-alone), sebutkan bahwa produk berdiri sendiri.

2.2. Fungsi produk

Jelaskan fungsi produk secara umum.

2.3. Karakteristik pengguna

Sebutkan pengguna produk beserta karakteristiknya

2.4. Kekangan

Sebutkan hal-hal yang dapat membatasi/mempengaruhi pengembangan produk, misal: kebijakan dan regulasi, keterbatasan perangkat keras, pertimbangan kehandalan dan keamanan.

2.5. Asumsi dan ketergantungan

Sebutkan asumsi-asumsi yang dipakai dalam pembuatan SRS.

3. Fungsionalitas sistem

Gambarkan diagram usecase

3.1. Fungsionalitas 1 (Sebutkan nama fungsionalitas sesuai usecase)

Deskripsi

Sebutkan kegunaan fungsionalitas

Urutan langkah

Sebutkan urutan langkah penggunaan fungsionalitas

Kebutuhan sistem

R01.1 Kebutuhan sistem 1 (Tuliskan kebutuhan sistem)

R01.2 Kebutuhan sistem 2 (Tuliskan kebutuhan sistem)

R... ...

3.2. Fungsionalitas 2 (Sebutkan nama fungsionalitas)

3.3. ...

4. Kebutuhan antarmuka eksternal

4.1. Antarmuka pengguna

Sebutkan antarmuka pengguna yang diharapkan

4.2. Antarmuka perangkat keras

Sebutkan perangkat keras yang diperlukan dalam operasi

4.3. Antarmuka perangkat lunak	<i>Sebutkan perangkat lunak yang diperlukan dalam operasi</i>
4.4. Antarmuka komunikasi	<i>Sebutkan protokol komunikasi yang digunakan</i>
5. Kebutuhan lain	<i>Tuliskan kebutuhan-kebutuhan lain, misal: kinerja sistem (cacah pengguna simultan, waktu pengolahan transaksi, cacah transaksi per menit, dsb.), keamanan (enkripsi, kendali akses, dsb.)</i>
Lampiran	

SRS tidak berisi rancangan perangkat lunak atau kebutuhan proyek pengembangan.

Level kebutuhan

Kebutuhan dalam SRS dituliskan dari sisi sistem, karena kebutuhan menyatakan kemampuan perangkat lunak. Penulisan kebutuhan sebaiknya menggunakan kata kunci yang mengindikasikan level signifikansi kebutuhan tersebut.

Level	Kata kunci	Penjelasan
1	HARUS (MUST, SHALL, REQUIRED)	Pernyataan kebutuhan harus dipenuhi.
	HARUS TIDAK (MUST NOT, SHALL NOT)	Pernyataan kebutuhan harus tidak ada/terjadi.
2	SEBAIKNYA (SHOULD, RECOMMENDED)	Pernyataan kebutuhan sebaiknya dipenuhi. Pernyataan kebutuhan tidak dipenuhi jika ada alasan kuat untuk mengabaikannya.
	SEBAIKNYA TIDAK (SHOULD NOT, NOT RECOMMENDED)	Pernyataan kebutuhan sebaiknya tidak dipenuhi. Pernyataan kebutuhan dipenuhi jika ada alasan kuat untuk memenuhinya.
3	BOLEH (MAY, OPTIONAL)	Pernyataan kebutuhan boleh dipenuhi atau tidak.

Contoh sistem pemesanan tiket pesawat:

- Sistem HARUS bisa menampilkan daftar pesawat pada rute dan tanggal yang diminta.
- Sistem SEBAIKNYA bisa memberikan pilihan pengurutan daftar pesawat berdasar jam keberangkatannya atau harga tiketnya.
- Daftar pesawat BOLEH menampilkan cacah sisa kursi yang masih tersedia pada tiap pesawat.

Karakteristik SRS yang baik

SRS harus mempunyai karakteristik berikut.

1. Benar
2. Tidak ambigu
3. Lengkap
4. Konsisten

5. Mempunyai indikasi signifikansi atau stabilitas
6. Dapat diverifikasi (*verifiable*)
7. Dapat dimodifikasi (*modifiable*)
8. Dapat dilacak (*traceable*)

Penjelasan masing-masing karakteristik adalah sebagai berikut.

Benar

Semua kebutuhan harus sesuai dengan

- Kebutuhan pelanggan atau pengguna.
- Spesifikasi sistem yang lebih besar atau standar yang berlaku.

Tidak ambigu

Setiap kebutuhan hanya mempunyai tafsiran tunggal. Istilah-istilah khusus atau yang berpotensi mempunyai tafsiran ganda harus didefinisikan dalam daftar istilah (Bagian 1.3).

Lengkap

SRS harus mencakup hal-hal berikut:

- Semua kebutuhan yang penting, baik fungsional ataupun non-fungsional.
- Definisi tanggapan perangkat lunak terhadap semua kemungkinan jenis data masukan, baik yang valid maupun invalid, dalam semua kemungkinan situasi.
- Label dan perujukan ke semua gambar, tabel dan diagram dalam SRS.
- Definisi istilah dan satuan ukuran.

Konsisten

SRS tidak boleh mengandung konflik atau pertentangan antar kebutuhan, baik dalam hal masukan, keluaran, serta proses dan urutan proses. Penggunaan istilah juga harus konsisten.

Mempunyai indikasi signifikansi atau stabilitas

Setiap kebutuhan harus mempunyai indikasi signifikansi atau stabilitas. Signifikansi menandakan seberapa penting kebutuhan tersebut dibandingkan kebutuhan-kebutuhan yang lain. Tingkat signifikansi dapat ditandai dengan kata kunci pada RFC 2119. Alternatif lain adalah

- Esensial : penting dan harus dipenuhi
- Kondisional : penting, tapi tak harus dipenuhi
- Opsional : tak penting, tak harus dipenuhi

Stabilitas menandakan seberapa besar kemungkinan kebutuhan tersebut dapat berubah.

Dapat diverifikasi

Setiap kebutuhan harus dapat diverifikasi, yakni dapat dicek apakah kebutuhan tersebut sudah disediakan oleh perangkat lunak. Kebutuhan yang ambigu tidak dapat diverifikasi. Pernyataan seperti “bekerja dengan baik” atau “antarmuka pengguna yang bagus” harus dihindari tidak bisa diverifikasi.

Dapat dimodifikasi

Struktur SRS harus mampu mengakomodasi perubahan kebutuhan secara mudah, lengkap, dan konsisten. Hal ini memerlukan SRS yang

- Mudah diorganisasi, antara lain: mempunyai daftar isi, indeks dan perujukan eksplisit antar bagian SRS.
- Tidak redundan, yakni satu kebutuhan dituliskan satu kali saja.
- Tiap kebutuhan ditulis terpisah; tidak digabungkan dengan kebutuhan lain.

Dapat dilacak

Asal setiap kebutuhan harus jelas. SRS harus memudahkan dalam pengembangan mendatang atau perbaikan dokumentasi. Pelacakan bisa berupa

- Pelacakan balik (*backward traceability*, ke sumber atau pengembangan sebelumnya).
Tiap kebutuhan harus merujuk ke sumber asal atau alasannya.
- Pelacakan maju (*forward traceability*, ke semua dokumen merujuk ke SRS ini).
Tiap kebutuhan harus mempunyai nama unik atau nomor referensi.

Contoh SRS

Lihat dokumen contoh SRS sistem informasi perpustakaan SIPUT.

Referensi

IEEE Std 830-1998, *IEEE Recommended Practice for Software Requirements Specifications*.

ISO/IEC/IEEE 29418-2011, *System and software engineering – Life cycle processes – Requirements engineering*.

IETF RFC 2119-1997, *Keywords for Use in RFCs to Indicate Requirement Levels*.