

SPEKIFIKASI KEBUTUHAN PERANGKAT LUNAK (SKPL) BERORIENTASI PROSES

1. Pendahuluan

Dokumen ini berisi penjelasan pemakaian dan penulisan dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) atau *Software Requirement Specification (SRS)* dengan pendekatan (ancangan) berorientasi proses. Dokumen ini selanjutnya akan menggunakan istilah SKPL. Dokumen ini sebagian besar adalah adaptasi dari dokumen IEEE Std 830-1993.

Uraian yang dituangkan di dalam dokumen ini digunakan sebagai acuan dalam menulis SKPL. Dokumen ini dibuat untuk membantu membuat spesifikasi perangkat lunak yang akan dikembangkan dengan ancangan berorientasi proses. Pada prinsipnya, hasil analisis sistem perangkat lunak dengan ancangan ini diuraikan sebagai sekumpulan proses yang terorganisasi secara hirarkis. Proses-proses tersebut saling berkomunikasi melalui suatu jalur aliran data.

2. Referensi

- IEEE Std 830-1993, IEEE Recommended Practice for Software Requirement Specifications.
- IEEE Std 610.12-1990 IEEE Standard Glossary of Software Engineering Terminology (ANSI).

3. Definisi, Singkatan, dan Akronim Definisi dari istilah yang akan digunakan pada dokumen ini dibuat berdasarkan hasil terjemahan dari IEEE Std 610.12-1990.

1. Pelanggan Adalah orang atau organisasi yang membayar produk, dan biasanya (tidak harus) ia yang akan memutuskan kebutuhannya.
2. Pengembang Adalah orang yang menghasilkan produk untuk pelanggan.
3. Pengguna Adalah orang yang akan langsung menjalankan atau menggunakan produk. Pengguna dan pelanggan umumnya adalah orang yang sama.

SKPL Spesifikasi Kebutuhan Perangkat Lunak

SRS *Software Requirement Specification*

DFD *Data Flow Diagram*

ERD *Entity Relationship Diagram*

STD *State Transition Diagram*

DBMS *Data Base Management System*

4. Bagian-bagian SKPL

SKPL berorientasi proses ini tidak didasarkan pada penggunaan metode tertentu melainkan menggunakan asumsi bahwa metode analisis berorientasi proses secara prinsip menggunakan notasi atau representasi konvensional umum seperti Data Flow Diagram (DFD) sebagai dasar Entity Relationship Diagram (ERD). Notasi pelengkap lain seperti pseudo-code, flow chart, flow map, matriks-matriks (Proses-Data, Proses- Organisasi, Organisasi-Data), state transition diagram dapat digunakan pula. Penempatan semua hasil

produk dengan menggunakan notasi-notasi pelengkap ini dapat dilakukan sesuai kebutuhan pembuat SKPL (melalui proses *tailoring*). SKPL ini secara prinsip diuraikan berdasarkan *outline* seperti berikut ini.

Daftar Isi

1. Pendahuluan

- 1.1. Tujuan Penulisan Dokumen
- 1.2. Lingkup Masalah
- 1.3. Definisi, Akronim dan Singkatan
- 1.4. Referensi
- 1.5. Deskripsi Umum Dokumen

2. Deskripsi Global Perangkat Lunak

- 2.1. Perspektif Produk
- 2.2. Fungsi Produk
- 2.3. Karakteristik Pengguna
- 2.4. Batasan-batasan
- 2.5. Asumsi dan Kebergantungan

3. Deskripsi Rinci Kebutuhan

- 3.1. Kebutuhan antarmuka eksternal
 - 3.1.1. Antarmuka pemakai
 - 3.1.2. Antarmuka perangkat keras
 - 3.1.3. Antarmuka perangkat lunak
 - 3.1.4. Antarmuka komunikasi
- 3.2. Deskripsi Fungsional
 - 3.2.1. Aliran informasi
 - 3.2.1.1.DFD Level 1
 - 3.2.1.2.DFD Level 2 dan seterusnya
 - 3.2.2. Deskripsi proses
 - 3.2.2.1.Proses 1
 - 3.2.2.2.Proses 2 dan seterusnya
- 3.3. Deskripsi Data
 - 3.3.1.1.Data 1
 - 3.3.1.2.Data 2 dan seterusnya
- 3.4. Deskripsi Kebutuhan Non Fungsional
- 3.5. Atribut Kualitas Perangkat Lunak
- 3.6. Batasan Perancangan

4. Matriks Keterunutan

Lampiran

4.1 Pendahuluan

Pendahuluan dari SKPL harus memberikan gambaran umum dari seluruh dokumen SKPL (bukan sistem perangkat lunak yang hendak dibangun). Pendahuluan SKPL harus berisi bagian-bagian berikut:

- Tujuan
- Lingkup Masalah
- Definisi, Akronim dan Singkatan
- Referensi
- Deskripsi Umum Dokumen

4.1.1 Tujuan

Bagian ini harus menunjukkan tujuan pembuatan SKPL secara umum. Uraikan pula pengguna dari dokumen SKPL ini dan dengan tujuan apa para pengguna tersebut menggunakan SKPL ini.

4.1.2 Lingkup Masalah

Bagian ini harus:

- Mengidentifikasi produk perangkat lunak yang dispesifikasi pada dokumen ini berdasarkan nama. Contoh, “MySoft Professional versi 2.3 for Windows”.
- Menjelaskan apa yang akan dilakukan dan tidak dilakukan (bila perlu) oleh perangkat lunak yang dispesifikasikan pada dokumen ini.
- Menjelaskan penerapan perangkat lunak yang dispesifikasi pada

dokumen ini beserta manfaat, tujuan dan sasaran dari pembuatan perangkat lunak tersebut.

- Merujuk pada identifikasi spesifikasi yang ada di dokumen-dokumen pendahulu SKPL ini (misalnya kontrak atau spesifikasi sistem) dan apa yang diutarakan pada bagian ini (serta bagian-bagian lainnya) harus konsisten dengan dokumen-dokumen tersebut

4.1.3 Definisi, akronim dan singkatan

Harus memberikan penjelasan terhadap semua definisi, akronim dan singkat yang digunakan agar dapat menginterpretasikan SKPL dengan benar dan satu arti. Informasi ini dapat dibuat pada lampiran atau dokumen terpisah. Pada kasus ini, bagian ini diisi dengan rujukan ke lampiran atau dokumen yang dimaksud.

4.1.4 Referensi

Bagian ini harus memberikan:

- Daftar lengkap dari dokumen (baik itu berupa buku, panduan, atau spesifikasi/deskripsi lain) yang dirujuk pada dokumen SKPL ini
- Identifikasi dari setiap dokumen berdasarkan judul, nomor dokumen (bila ada), tanggal dan organisasi penerbit
- Bila perlu, sebutkan sumber-sumber atau organisasi yang dapat memberikan referensi yang dituliskan tersebut

4.1.5 Deskripsi Umum Dokumen

Bagian ini adalah ikhtisar dari dokumen SKPL. Tuliskan sistematika pembahasan dokumen SKPL ini. Pada bagian ini, dijelaskan pula tentang penempatan notasi-notasi lain di luar DFD

dan ERD bila ada sesuai dengan metode analisis perangkat lunak yang digunakan. Pada sejumlah metode analisis terstruktur (berorientasi proses), *state transition diagram* (STD) bisa jadi merupakan salah satu hasil produknya. Khusus untuk STD, dapat ditempatkan sebagai salah satu sub bab di bagian 3. Deskripsi Rinci Kebutuhan

4.2 Deskripsi Global Perangkat Lunak

Bagian ini merupakan penjelasan tentang perangkat lunak secara umum. Dijelaskan melalui perspektif perangkat lunak relatif terhadap konteksnya, fungsi dasar perangkat lunak, karakteristik pengguna yang diarah, batasan-batasan yang mempengaruhi perangkat lunak secara umum, serta asumsi dasar yang digunakan dan kebergantungan perangkat lunak pada fenomena lain di luar perangkat lunak.

Bagian ini tidak memberikan kebutuhan rinci, hanya latar belakang dari kebutuhan tersebut. Bagian ini terdiri dari

- Perspektif produk
- Fungsi Produk
- Karakteristik Pengguna
- Batasan-batasan
- Asumsi dan Ketergantungan

4.2.1 Perspektif Produk

Bagian ini menjelaskan posisi perangkat lunak relatif terhadap konteks sistem lain yang melingkupinya. Jika produk tidak bergantung pada sistem atau produk lain, maka harus juga dinyatakan di sini. Jika SKPL mendefinisikan perangkat lunak sebagai sebuah komponen dari suatu sistem yang lebih besar yang

melingkupinya, maka bagian ini harus menghubungkan kebutuhan dari sistem yang lebih besar ini dengan fungsionalitas dari perangkat lunak yang dispesifikasikan dan harus mengindentifikasikan bagaimana antarmuka antara keduanya.

Untuk mempermudah, sebuah diagram blok dapat digunakan untuk menjelaskan disertai dengan narasinya. Diagram blok sebaiknya dapat menunjukkan:

- komponen-komponen utama dari sistem yang lebih besar yang melingkupi perangkat lunak yang dispesifikasikan
- interkoneksi antara perangkat lunak yang dispesifikasikan dengan komponen/sistem lain yang melingkupinya
- antarmuka eksternal dari perangkat lunak yang dispesifikasikan tersebut.

4.2.2 Fungsi Produk

Bagian ini mengutarakan fungsi-fungsi dasar sistem yang utama. Pada akhirnya fungsi-fungsi dasar sistem ini berimpitan dengan *bubble* pada *level 1* DFD. Namun sebagai panduan kasar, yang disebut sebagai fungsi dasar sistem adalah jawaban atas masalah yang hendak diselesaikan melalui pembuatan perangkat lunak ini. Kadang-kadang kesimpulan dari fungsi yang diperlukan untuk bagian ini dapat diambil secara langsung dari bagian spesifikasi yang lebih tinggi (jika ada) yang akan mengalokasikan fungsi khusus dari produk perangkat lunak. Perhatikan bahwa untuk alasan kejelasan:

- Fungsi harus diorganisasikan dengan suatu cara sehingga daftar fungsi dapat dimengerti oleh pengguna atau orang lain yang membaca dokumen pertama kali
- Metode tekstual dan grafik dapat digunakan untuk menunjukkan fungsi yang berbeda dan keterhubungannya. Diagram ini tidak ditujukan untuk menunjukkan perancangan produk

tetapi juga menunjukkan hubungan logik antar fungsi.

Sebagai contoh SKPL untuk perangkat lunak apotek, bagian ini digunakan untuk menjelaskan secara umum tentang pengelolaan obat, penerimaan resep, pendaftaran pemasok tanpa menyebutkan kebutuhan rinci dari masing-masing fungsi tersebut.

4.2.3 Karakteristik Pengguna

Karakteristik pengguna menggambarkan siapa saja pengguna dari perangkat lunak yang dispesifikasikan dan apa saja haknya terhadap perangkat lunak tersebut. Pengguna penting disebutkan karena pada akhirnya perangkat lunak yang dibangun harus mampu menjawab tantangan kebutuhan dari pengguna yang spesifik pula.

Pengungkapan karakteristik pengguna dapat dilakukan dengan menyatakannya pada sebuah tabel dengan kolom-kolom: Pengguna, Tanggung Jawab (tanggung jawabnya relatif yang berkaitan dengan perangkat lunak ini), Hak Akses (hak akses ini dihubungkan pula ke fungsi dasar sistem yang tertulis pada bagian Fungsi Produk), Tingkat Pendidikan, Tingkat Keterampilan (yang dibutuhkan), Pengalaman (yang dibutuhkan), Jenis Pelatihan (yaitu pelatihan yang dibutuhkan agar pengguna ini dapat melakukan tanggung jawabnya, sifatnya opsional hanya diisi jika dibutuhkan).

4.2.4 Batasan-batasan

Bagian SPKL ini berisi deskripsi umum dari item lain yang akan membatasi pilihan atau keputusan pada spesifikasi. Hal-hal tersebut antara lain:

- Kebijakan umum organisasi/lingkungan
- Keterbatasan karena perangkat keras, contohnya kebutuhan *signal timing*

- Standar antarmuka ke aplikasi atau sistem lain
- Tuntutan pengoperasian secara paralel atau *multi platform*

4.2.5 Asumsi dan Kebergantungan

Bagian ini mengungkapkan setiap factor yang mempengaruhi kebutuhan yang dinyatakan pada SKPL. Faktor-faktor ini bukan merupakan pembatasan atas keputusan yang diambil untuk perancangan perangkat lunak, melainkan hal-hal di luar cakupan perangkat lunak yang dispesifikasikan, yang bila diubah dapat berakibat pada atau mengubah kebutuhan yang tertulis di SKPL. Sebagai contoh asumsi bahwa suatu sistem operasi akan tersedia pada suatu platform perangkat keras dari produk perangkat lunak. Jika sistem operasi tidak ada maka SKPL harus diubah karena hal tersebut.

Di bagian ini dapat pula diungkapkan prioritas pengembangan dari sejumlah fungsi dasar sistem yang telah diuraikan sebelumnya. Identifikasikan pula kebutuhan yang ditunda pengembangannya sampai versi-versi lanjut.

4.3 Deskripsi Rinci Kebutuhan

Bagian SKPL ini harus berisi semua kebutuhan perangkat lunak hingga pada tingkat rinci yang memungkinkan pengembang untuk merancang sistem perangkat lunak untuk memenuhi kebutuhan-kebutuhan itu dan juga bagi penguji untuk menguji sistem terhadap kebutuhan. Pada bagian ini, setiap pernyataan kebutuhan harus dapat diterima oleh pengguna, operator atau sistem eksternal lain. Kebutuhan ini harus melibatkan paling tidak:

- deskripsi dari setiap masukan ke sistem (stimulus)
- deskripsi dari setiap keluaran dari sistem (respon)
- deskripsi dari semua fungsi yang dilakukan oleh sistem untuk

menanggapi masukan dan mendukung keluaran dari sistem dan semua fungsi dilakukan oleh sistem sebagai respon terhadap masukan/keluaran

Karena bagian ini merupakan bagian yang paling besar dan bagian penting dari SKPL, maka prinsip-prinsip yang digunakan:

- Semua kebutuhan rinci harus dinyatakan sesuai dengan karakteristik kebutuhan yang baik (lihat GL01)
- Semua kebutuhan khusus harus sedapat mungkin diacusilangkan dengan dokumen sebelumnya yang berhubungan (dengan kata lain sesuai dengan dokumen yang diacu)
- Semua kebutuhan harus dapat diidentifikasi secara unik.
- Organisasi pernyataan kebutuhan harus sedemikian yang memaksimalkan kemudahan pembacaan (*readability*).

4.3.1 Kebutuhan antarmuka eksternal

Antarmuka eksternal merincikan deskripsi masukan dan keluaran perangkat lunak yang dispesifikasikan. Ada berbagai macam antarmuka eksternal, masing-masing bila perlu dapat diuraikan dengan cara yang berbeda. Pengungkapan isi dan format dari setiap antarmuka eksternal dapat berbentuk:

1. Nama item
2. Deskripsi penggunaan
3. Sumber masukan atau tujuan keluaran
4. Jangkauan yang diterima, kebenaran atau toleransi.
5. Unit pengukuran
6. Pewaktuan (*timing*)
7. Keterhubungan dengan masukan/keluaran lain
8. Format/organisasi layar
9. Format/organisasi window

10. Format data
11. Format perintah
12. Pesan-pesan akhir

Secara lebih rinci antarmuka eksternal dikelompokkan menjadi antarmuka pemakai, antarmuka perangkat keras, antarmuka perangkat lunak, dan antarmuka komunikasi.

Dengan menggunakan kaskas DFD, maka pada bagian ini dicantumkan diagram konteks beserta penjelasan tentang aliran data yang masuk dan keluar dari sistem.

4.3.1.1 Antarmukapemakai

Bagian ini berisi hal-hal berikut:

- Karakteristik logis dari setiap antarmuka antara produk perangkat lunak dan penggunaannya. Hal ini akan melibatkan karakteristik konfigurasi (misalnya standar format layar, tataletak window, isi laporan/menu –bukan tata letak tiap layar/windownya sendiri- atau ketersediaan kunci khusus atau jenis mouse) untuk memenuhi kebutuhan sistem.
- Semua aspek optimisasi antarmuka dengan orang yang akan menggunakan sistem. Bagian ini mungkin hanya berisi daftar yang harus dan tidak boleh dilakukan oleh sistem dari sudut pandang pengguna. Misalnya kebutuhan untuk pemilihan pesan yang singkat atau panjang. Seperti kebutuhan lain, kebutuhan ini harus dapat di verifikasi. Misalnya kalimat “seorang pegawai berpengalaman dapat melakukan X dalam Z menit setelah 1 jam training” akan lebih baik daripada hanya mendefinisikan “Seorang pegawai berpengalaman dapat melakukan X”.

4.3.1.2 Antarmukaperangkatkeras

Bagian ini menjelaskan karakteristik logis dari setiap antarmuka antara produk perangkat lunak dan komponen perangkat keras dari

sistem. Bagian ini akan melibatkan karakteristik konfigurasi (jumlah port, jumlah instruksi, dll). Antarmuka ini juga melibatkan hal-hal seperti perangkat pendukung, dan bagaimana peralatan tersebut menjadi pendukung, dan protokol. Bagian ini hanya diisi jika sistem perangkat lunak yang dispesifikasikan membutuhkan perangkat keras khusus, contoh: *VideoGrabber Card*, *FM Tuner*, *Sound Card*, dan lain-lain.

4.3.1.3 Antarmukaperangkatlunak

Bagian ini menspesifikasikan penggunaan produk perangkat lunak lain (misalnya sistem manajemen basis data, sistem operasi atau paket matematik) dan antarmuka dengan sistem aplikasi lain (sebagai contoh hubungan antara sistem *account*

receivable dan sistem *General Ledger*). Bagian ini hanya diisi jika perangkat lunak yang dispesifikasikan memakai antarmuka (berupa perangkat lunak lain atau mekanisme khusus), misalnya API Windows. Jadi jika perangkat lunak direncanakan hanya berjalan di atas Windows saja tanpa menggunakan layanan Windows misalnya, tidak perlu dituliskan.

Untuk setiap perangkat lunak yang dibutuhkan atau terkait, harus disertai dengan:

- Nama
- Mnemonic
- Nomor spesifikasi
- Nomor V ersi
- Sumber

Untuk setiap antarmuka, harus disertai dengan hal-hal berikut:

- Tujuan menghubungkan perangkat lunak tersebut dengan perangkat lunak yang dispesifikasikan.
- Definisi dari antarmuka dalam bentuk isi pesan dan formatnya. Jika antarmuka yang sudah terdokumentasi dengan baik, maka tidak perlu diuraikan ulang tetapi cukup mengacu ke dokumen tersebut.

4.3.1.4 Antarmuka komunikasi

Bagian ini harus menspesifikasikan berbagai antarmuka untuk komunikasi, seperti protokol jaringan lokal. Bagian ini hanya diisi jika perangkat lunak yang dispesifikasikan beroperasi dengan memanfaatkan antarmuka tersebut. Contoh: RS232, TCP/IP, WinSock. Jadi, jika perangkat lunak yang dispesifikasi hanya sekedar dijalankan di atas Unix tanpa menggunakan protokol TCP atau IP, maka TCP/IP tidak perlu disebutkan.

4.3.2 Kebutuhan Fungsional

Kebutuhan fungsional harus mendefinisikan aksi dasar yang harus diambil oleh perangkat lunak untuk menerima dan memproses masukan dan menghasilkan keluaran.

Dapat dilakukan juga pembagian kebutuhan fungsional menjadi sub fungsional atau sub proses. Hal ini tidak berarti bahwa rancangan perangkat lunak akan dibagi dengan cara yang sama.

4.3.2.1 Aliran informasi

Bagian ini mencantumkan dan menguraikan DFD level demi level.

4.3.2.1.1 DFD1

Cantumkan dan beri penjelasan DFD pada level satu. Sebaiknya beri judul yang sesuai. Lengkapi diagram DFD Level 1 dengan tabel yang kolom-kolomnya adalah: Nomor Proses, Nama Proses, Masukan, Keluaran, dan Keterangan (hanya diisi bila perlu, antara

lain untuk menjelaskan apakah masukan atau keluaran berupa data atau kontrol). Lengkapi pula dengan kamus data untuk semua aliran data yang tertulis.

4.3.2.1.2 DFD2 dan seterusnya

Cantumkan dan beri penjelasan DFD pada level dua (dan seterusnya). Sebaiknya beri judul yang sesuai dengan Nama Proses di level atasnya. Lengkapi diagram DFD Level 2 (dan seterusnya) dengan tabel yang kolom-kolomnya adalah: Nomor Proses, Nama Proses, Masukan, Keluaran, dan Keterangan (hanya diisi bila perlu, antara lain untuk menjelaskan apakah masukan atau keluaran berupa data atau kontrol). Lengkapi pula dengan kamus data untuk semua aliran data yang tertulis.

4.3.2.2 Deskripsi proses

Deskripsi Proses hanya dituliskan untuk proses yang sudah tidak dapat didekomposisi lebih jauh lagi.

4.3.2.2.1 Proses1

Uraikan deskripsi proses dengan menggunakan narasi dengan bahasa alami atau dengan pseudo-code. Deskripsi proses harus memberikan gambaran kebutuhan fungsional dengan jelas yang mencakup:

- Validasi terhadap masukan
- Urutan pasti dari operasi
- Tanggapan atas situasi abnormal termasuk *overflow*, fasilitas untuk komunikasi atau penanganan kesalahan (*error handling*) dan pemulihan (*recovery*).
- Efek dari keberadaan dan nilai parameter
- Hubungan antara keluaran ke masukan, termasuk urutan

masukan/keluaran, atau formula untuk konversi masukan ke keluaran.

Sebaiknya beri judul yang sesuai dengan Nama Proses yang diuraikan.

4.3.2.2 Proses2 dan seterusnya

Sama seperti pada uraian Proses 1. Beri judul yang sesuai dengan Nama Proses yang diuraikan.

4.3.3 Deskripsi Data

Kebutuhan ini harus menspesifikasi kebutuhan logis untuk setiap informasi yang diletakkan ke basisdata. Nyatakanlah kebutuhan data ini dengan Entity Relationship Diagram dan lengkapi dengan skema relasi. Bila perlu jelaskan pula:

- Batasan integritas
- Frekuensi pemakaian
- Retensi (kelangsungan) data

Harap diperhatikan bahwa semua *storage* yang ada pada DFD harus memiliki representasi data yang sesuai di sini. Ada kalanya representasi data tersebut tidak dapat terhubung langsung di ERD. Pada kasus ini, representasi data tetap diuraikan tetapi secara terpisah dari ERD.

4.3.3.1 Data1

Uraikan satu per satu *entity*, *relationship* atau representasi data lain, yang pada

akhirnya nanti akan menjadi tabel atau suatu data persisten secara detil. Beri judul

yang sesuai dengan data yang diuraikan.

Kamus data dapat dinyatakan dengan tabel yang memiliki kolom-kolom:

- Nama sub-data pembentuk
- Representasi, misalnya: teks, karakter, numerik.
- Unit/format, misalnya: kg, meter, orang.
- Presisi, misalnya 2 desimal
- Range, misalnya 1-100, A..F
- Nilai tetap (*default*)
- Boleh kosong/tidak

4.3.3.2 Data2danseterusnya

Sama seperti Data 1. Beri judul yang sesuai dengan nama data yang diuraikan.

4.3.4 Deskripsi Kebutuhan Non Fungsional

Bagian ini menspesifikasikan ukuran kuantitatif yang harus dipenuhi oleh perangkat lunak. Uraian minimal pada bagian ini berisi sebuah tabel, dengan kolom: Kriteria Kebutuhan, Tuntutan kebutuhan. Kebutuhan tersebut antara lain: Performansi, Batasan Memori, Modus Operasi, Adaptasi Situs atau Ergonomi. Bila diperlukan uraian khusus, dapat dilakukan dengan membagi sub-bab seperti di bawah ini.

4.3.4.1 Performansi

Bagian ini harus menspesifikasikan baik kebutuhan numerik statik/dinamik yang terletak pada interaksi perangkat lunak atau

pada interaksi manusia dengan perangkat lunak secara keseluruhan. Kebutuhan numerik statis mungkin melibatkan:

- Jumlah terminal yang didukung
- Jumlah pengguna simultan yang didukung
- Jumlah dan tipe informasi yang ditangani

Kebutuhan numerik statik sering diidentifikasi pada bagian terpisah yang disebut kapasitas. Kebutuhan numerik dinamik mungkin dapat melibatkan, sebagai contoh, jumlah transaksi dan tugas dan jumlah data yang akan diproses selama jangka waktu tertentu, baik kondisi normal atau kondisi beban puncak.

Semua kebutuhan ini harus dinyatakan dalam istilah yang dapat diukur. Contohnya, kalimat “95 % transaksi harus diproses dalam 1 detik”, akan lebih baik daripada kalimat “operator mungkin tidak harus menunggu transaksi akan selesai.

4.3.4.2 Batasan Memori

Bagian ini menspesifikasikan setiap karakteristik dan batasan memori primer dan sekunder. Upayakan seakurat mungkin. Bila tidak maka ungkapkan batas maksimum atau minimumnya.

4.3.4.3 Modus Operasi

Bagian ini merincikan sejumlah modus operasi perangkat lunak bila ada. Rincian ini menentukan operasi normal dan operasi khusus yang dibutuhkan oleh pengguna seperti misalnya:

- Berbagai variasi modus operasi dalam organisasi pengguna, misalnya operasi yang bersifat *user-initiated* (inisiatif dari pengguna)
- Periode operasi interaktif dan periode operasi *offline*

- Fungsi pendukung untuk pemrosesan data
- Operasi *backup* dan *recovery*

4.3.4.4 Kebutuhan adaptasi okasi

Bagian ini dapat berisi:

- Pendefinisian kebutuhan untuk setiap data atau urutan inialisasi yang tergantung pada lokasi, misi atau modus operasi (misalnya batas keselamatan).
- Menspesifikasikan modifikasi yang perlu diterapkan pada lokasi atau hal lain yang berhubungan dengan misi untuk mengadaptasi perangkat lunak terhadap suatu instalasi tertentu.

4.3.5 Atribut Kualitas Perangkat Lunak

Ada sejumlah atribut kualitas perangkat lunak yang dapat ditampilkan sebagai kebutuhan. Atribut yang diinginkan harus dispesifikasikan sedemikian sehingga hasilnya dapat diverifikasi. Uraian minimum pada bagian ini berisi sebuah tabel dengan kolom: Kriteria Kualitas, Tuntutan Kualitas. Butir kualitas yang dapat dipertimbangkan antara lain: keandalan (*reliability*), ketersediaan (*availability*), keamanan (*security*), keremawatan (*maintainability*), kepemindahan (*portability*). Bila diperlukan uraian khusus, dapat dilakukan dengan menguraikannya menjadi sub-bab tersendiri.

4.3.5.1 Keandalan

Bagian ini berisi spesifikasi factor-faktor yang diperlukan untuk mencapai keandalan sistem pada saat diserahkan.

4.3.5.2 Ketersediaan.

Bagian ini berisi spesifikasi factor-faktor yang diperlukan untuk menjamin tingkat ketersediaan seluruh sistem saat sistem

beroperasi, seperti *checkpoint*, *recovery* dan *restart*.

4.3.5.3 Keamanan

Bagian ini berisi faktor untuk memproteksi perangkat lunak dari akses, penggunaan, perubahan, penghancuran atau pengungkapan (*disclosure*) yang tidak disengaja atau yang merusak. Kebutuhan yang spesifik termasuk hal-hal berikut:

1. Penggunaan teknik kriptografi
2. Penyimpanan data log/history
3. Pemberian suatu fungsi ke modul-modul yang berbeda
4. Pembatasan komunikasi terhadap suatu area tertentu dalam program
5. Pemeriksaan integritas data untuk peubah-peubah kritis

4.3.5.4 Pemeliharaan (*Maintainability*)

Bagian ini menentukan atribut perangkat lunak yang berhubungan dengan kemudahan perawatan dari perangkat lunak tersebut. Atribut tersebut dapat berupa kebutuhan akan tingkat modularitas, antarmuka, kompleksitas, dan lain-lain. Penulisan atribut keremawatan tidak dilakukan hanya atas dasar pemikiran atas praktik perancangan yang baik saja, tetapi harus didasari pada tuntutan kondisi sistem.

4.3.5.5 Kepemindahan (*Portability*)

Atribut dari perangkat lunak yang berhubungan dengan kemudahan pemindahan perangkat lunak ke mesin dan/atau sistem operasi lain. Atribut ini berbentuk antara lain:

- Persentase komponen yang berisi kode yang bergantung pada *host*
- Persentase kode yang bergantung pada *host*
- Penggunaan bahasa yang kepemindahannya terbukti

- Penggunaan suatu kompilator tertentu atau subset bahasa tertentu
- Penggunaan suatu sistem operasi tertentu

4.3.6 Batasan Perancangan

Bagian ini menspesifikasikan batasan atas keputusan-keputusan perancangan yang dituntut oleh standar lain, keterbatasan perangkat keras, dan lain-lain. Standar atau aturan yang ada dapat menurunkan spesifikasi kebutuhan khusus antara lain:

- Format laporan
- Penamaan data
- Prosedur akunting
- Penelusuran audit

Sebagai contoh, bagian ini dapat menentukan kebutuhan perangkat lunak keuangan untuk menelusuri aktivitas pemrosesan.

Penelusuran ini diperlukan agar suatu aplikasi sesuai dengan peraturan atau standar keuangan. Kebutuhan penelusuran audit, sebagai contoh, menyatakan bahwa semua perubahan harus dicatat pada suatu file khusus untuk penelusuran dengan isi sebelum dan sesudah dilakukan.

Contoh lain adalah menyatakan lingkungan implementasi (seperti sistem operasi, DBMS, kaskas pengembangan, bahasa pemrograman, kompilator) bila memang merupakan tuntutan yang ditentukan oleh pelanggan

4.4 Matriks Keterunutan

Bagian ini berisi daftar seluruh kebutuhan beserta identifikasinya serta cara verifikasi yang direncanakan, yaitu: Inspeksi, Analisis, Demonstrasi. Inspeksi dilakukan dengan mengamati produk yang dihasilkan (biasanya kode program) yang dibandingkan dengan standar atau spesifikasi yang ada. Analisis dilakukan dengan menerapkan pengukuran matematis/kuantitatif terhadap hasil yang

didapat dari penerapan produk. Demonstrasi dilakukan dengan mengamati perilaku produk akhir, yaitu melihat kesesuaian antara masukan dan keluaran.

4.5 Informasi tambahan

Dukungan informasi yang membuat SKPL mudah digunakan, antara lain:

1. Daftar isi
2. Index
3. Lampiran

4.5.1 Daftar isi dan Index

Daftar isi dan index adalah cukup penting dan harus mengikuti standard yang ada.

4.5.2 Lampiran-lampiran

Lampiran tidak selalu menjadi bagian dari spesifikasi kebutuhan aktual dan tidak harus selalu ada. Lampiran dapat berisi:

1. Contoh format masukan/keluaran, deskripsi analisa biaya, hasil survey
2. Dukungan informasi yang membantu SKPL.
3. Deskripsi dari masalah yang dipecahkan oleh perangkat lunak.
4. Instruksi khusus, dan media yang cocok untuk pengamatan, dan kebutuhan lain.
5. Flow Map atau prosedur manual yang merupakan lingkungan tempat perangkat lunak yang dispesifikasikan akan dijalankan.
6. Lampiran lain yang dianggap perlu dan berhubungan dengan spesifikasi perangkat lunak

Jika disertakan lampiran, SKPL harus secara eksplisit menegaskan

apakah lampiran ini adalah bagian dari kebutuhan.