**SKPL- PKKPT**

SPESIFIKASI KEBUTUHAN PERANGKAT LUNAK

PKKPT

( Pendeteksi Ketinggian Kendaraan Pada Tol )

Dipersiapkan oleh:

Kelompok 7

Arsy Novita Syahada J3D117090

Muhammad Zaidab J3D117104

Muhammad Farrell Pahlevi J3D117108

Program Studi Teknik Komputer

Sekolah Vokasi

Institut Pertanian Bogor

2019

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Program Studi Teknik Komputer**  **Sekolah Vokasi**  Institut Pertanian Bogor | Nomor Dokumen | | Halaman |
| *SKPL – PKKPT* | | *1/12* |
| *Revisi* | *1* | *Tgl: 18/09/19* |

DAFTAR PERUBAHAN

|  |  |
| --- | --- |
| Revisi | Deskripsi |
| A | Penambahan *flow chart* pada bab dua mengenai deskripsi global prangkat lunak |
| B |  |
| C |  |
| D |  |
| E |  |
| F |  |
| G |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| INDEX  TGL | - | A | B | C | D | E | F | G |
| Ditulis oleh |  |  |  |  |  |  |  |  |
| Diperiksa oleh |  |  |  |  |  |  |  |  |
| Disetujui oleh |  |  |  |  |  |  |  |  |

Daftar Halaman Perubahan

|  |  |  |  |
| --- | --- | --- | --- |
| Halaman | Revisi | Halaman | Revisi |
|  |  |  |  |

Daftar Isi

1. Pendahuluan 5

1.1 Tujuan 5

1.2 Lingkup Masalah 5

1.3 Definisi , Akronim, dan singkatan 5

1.4 Referensi 6

1.5 Deskripsi Umum Dokumen 6

2 Deskripsi Global Perangkat Lunak 7

2.1 Perspektif Produk 7

2.2 Fungsi Produk 7

2.3 Karakteristik Pengguna 8

2.4 Batasan-batasan 8

2.5 Asumsi dan Kebergantungan 8

3 Deskripsi Rinci Kebutuhan Perangkat Lunak 9

3.1 Kebutuhan Antarmuka Eksternal 9

3.1.1 Antarmuka Pemakai 9

3.1.2 Antarmuka Perangkat Keras 10

3.1.3 Antarmuka Perangkat Lunak 10

3.1.4 Antarmuka Komunikasi 10

3.2 Kebutuhan Fungsional 11

3.2.1 Aliran Informasi 11

3.2.1.1 DFD Level 1 11

3.2.1.2 DFD Level 2 dst 11

3.2.2 Deskripsi Proses 11

3.2.2.1 Proses 1 11

3.2.2.2 Proses 2 dst 11

3.3 Deskripsi Data 11

3.3.1 Data 1 12

3.3.2 Data 2 dst 12

3.4 Deskripsi Kebutuhan Non Fungsional 12

3.4.1 Performansi 12

3.4.2 Batasan Memori 12

3.4.3 Modus Operasi 13

3.4.4 Kebutuhan Adaptasi Lokasi 13

3.5 Atribut Kualitas Perangkat Lunak 13

3.5.1 Kehandalan *(reliability)* 13

*3.5.2* Ketersediaan *(Availability)* 13

3.5.3 Keamanan (*Security*) 13

*3.5.4* *Maintainability* 13

*3.5.5* *Portability* 14

3.6 Batasan Perancangan 14

4 Matriks Kerunutan 14

5 Informasi Tambahan 14

5.1 Daftar Isi dan Indeks 14

5.2 Lampiran 14

Setelah Daftar Isi Boleh ada Daftar Tabel dan Daftar Gambar

# 1. Pendahuluan

Dokumen ini berisi Spesifikasi Kebutuhan Perangkat Lunak (SKPL) atau Software *Requirement Specification* (SRS) untuk PKKPT ( Pendeteksi Ketinggian Kendaraan Pada TOL ). Untuk penamaan dokumen ini selanjutnya akan menggunakan istilah SKPL.

## Tujuan

Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) merupakan dokumen spesifikasi kebutuhan perangkat lunak yang akan dikembangkan. Dokumen ini akan digunakan oleh pengembang perangkat lunak sebagai acuan teknis pengembangan perangkat lunak pada tahap selanjutnya.

## Lingkup Masalah

Pendeteksi ketinggian kendaraan pada jalan tol merupakan sebuah perangkau lunak yang dapat digunakan untuk mendeteksi ketinggian kendaraan yang ada pada tol dan situs web ini juga dapat memperkirakan anggaran tarif yang akan dikeluarkan selama perjalanan. Ketinggian pada kendaraan merupakan yang yang penting dalam penentuan golongan dan tarif tol, kendaraan yang memiliki ketinggian yang berbeda maka kan memiliki tarif yang berbeda pula, untuk menjalankan situs web ini maka *user* dapat melakukan penge­*check*an info tarif tol sebelum melakukan perjanan. Untuk medapatkan info tarif tol *user* harus mengklik tombola tau *button* ­*generate* tinggi untuk mengetahui tinggi kendaraan beserta tipe golongan kendaraan tersebut, setelah mendapatkan informasi mengenai tinggi kendaraan maka *user* dapat beralih pada langkah berikutnya yaitu *user* memilih pintu masuk gerbang tol dan pintu keluar gerbang tol, setelah melakukan langkah tersebut *user* dapat mengklik *button* cek hasil untuk mengetahui tarif kendaraan tersebut. *User* juga dapat melakukan pengecekkan laporan yang akan berisi data secara keseluruhan yaitu, jenis golongan kendaraan, gerbang masuk dan keluar tol serta tarif tol. Penerapan perangkat lunak Sistem Pendeteksi Tinggi Kendaraan Pada TOL memiliki beberapa manfaat dan tujuan.

## Definisi , Akronim, dan singkatan

* SKPL adalah Spesifikasi Kebutuhan Perangkat Lunak, atau dalam bahasa Inggrisnya sering juga disebut sebagai *Software Requirements Spesification* (SRS) dan merupakan spesifikasi dari perangkat lunak yang akan dikembangkan.
* SKPL-VMS.K-xxxx adalah kode yang digunakan untuk mempresentasikan kebutuhan (*requirement*) pada VMS, dengan VMS merupakan kode perangkat lunak, VMS.K adalah kode fase, dan xxxx adalah digit atau nomor kebutuhan (*requirement*).
* DFD adalah *Data Flow Diagram*, diagram dan notasi yang digunakan untuk menunjukkan aliran data pada perangkat lunak.
* ERD adalah *Entity Relationship Diagram*, diagram dan notasi yang digunakan untuk merepresentasikan struktur data statis pada perangkat lunak.

## 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

## Deskripsi Umum Dokumen

Dokumen SKPL ini dibagi menjadi tiga bagian utama. Bagian utama berisi penjelasan tentang dokumen SKPL yang akan mencakup tujuan pembuatan dokumen ini, lingkup masalah yang diselesaikan oleh perangkat lunak yang dikembangkan, definisi, referensi dan deskripsi umum. Bagian kedua berisi penjelasan secara umum mengenai perangkat lunak yang akan dikembangkan meliputi fungsi dari perangkat lunak, karakteristik pengguna, batasan, dan asumsi yang diambil dalam pengembangan perangkat lunak. Bagian ketiga berisi uraian kebutuhan perangkat lunak secara lebih rinci.

# Deskripsi Global Perangkat Lunak

Pendeteksi Ketinggian Kendaraan Pada Tol merupakan perangkat lunak yang dapat mendeteksi ketinggian kendaraan yang akan memasuki salah satu gerbang tol dan dapat memberitahukan informasi mengenai tipe golongan kendaraan serta tarif yang akan dikeluarkan selama perjalanan.

## Perspektif Produk

Pendeteksi Ketinggian Kendaraan Pada Tol, memiliki komponen utama dari Pendeteksi Ketinggian Kendaraan Pada Tol adalah mekanisme untuk menentukan tinggi kendaraan yang masuk pada gerbang tol. Pendeteksi Ketinggian kendaraan Pada Tol, memiliki alur kerja umum untuk menentukan tinggi kendaraan tersebut maka kendaraan harus melewati sebuah sensor yang dipasangkan pada ketinggian gerbang tol dalam hal ini keberadaan sensor akan diilustrasikan sebagai *button generate* yang kemudian nilai dari *generate* tersebut akan digunakan sebagai nilai tinggi kendaraan dan tipe golongan kendaraan tersebut. Setelah nilai ketinggian kendaraan didapatkan dari hasil *generate* maka *user* dapat melanjutkan kelangkah berikutnya yaitu pemilihan gerbang tol, pada tahap ini *user* diharuskan memilih gerbang tol yang telah disediakan, gerbang tol tersebut berupa pilihan untuk gerbang masuk dan gerbang keluar kendaraan pada tol. Proses perhitungan tarif tol akan dilakukan setelah dua tahapan sebelumnya dijalankan, proses perhitungan tersebut memerlukan data dari *generate* dan pemilihan gerbang tol. *User* hanya dapat melihat data kendaraan berupa golongan kendaraan, gerbang masuk dan keluar tol serta tarif yang dikenakan, untuk melihat data secara keseluruhan hanya dapat dilakukan oleh *admin* tang telah melakukan *login* sebelumnya.

## Fungsi Produk

Adapun fungsi-fungsi yang dimiliki oleh perangkat lunak ini, adalah :

* Melakukan *generate*. *Generate* yang dilakukan oleh *user* guna mendapatkan nilai ketinggian kendaraan dan tipe golongan kendaraan. [SKPL-PKKPT.K-0001].
* Memilih gerbang tol. Pemilihan gerbang tol dilakukan oleh *user* berupa gerbang masuk tol dan gerbang keluar tol. [SKPL-PKKPT.K-0002].
* Perhitungan matematis. Perhitungan matematis dilakukan oleh sistem untuk mendapatkan tarif yang akan dibayarkan *user* selama melakukan perjalanan. [SKPL-PKKPT.K-0003].
* Verifikasi *username* dan *password*. Perangkat lunak akan melakukan pengecekan apakah data *username* dan *password* yang dimasukkan *valid* atau *invalid*. Data *username* dan *password* hanya dimiliki oleh admin saja. [SKPL-PKKPT.K-0004].
* Melihat laporan data. Melihat laporan data secara keseluruhan hanya dapat dilakukan oleh *admin*. [SKPL-PKKPT.K-0005].

## Karakteristik Pengguna

Pengguna perangkat lunak ini adalah *customer* yang hendak melakukan perjalanan pada jalan tol, dimana *customer* dapat mengetahui tarif yang akan dikeluarkan selama perjalanan. Pengguna lainnya adalah *Administrator* yang memiliki hak otorisasi lebih dibandingkan *customer* karena dapat melihat data secara keseluruhan yang lebih spesifik.

Table 1 Kategori Pengguna PKKPT

|  |  |  |
| --- | --- | --- |
| Kategori Pengguna | Tugas | Hak akses ke Aplikasi |
| Umum | Melakukan *generate* kemudian memilih gerbang tol. | [SKPL-PKKPT.K-0001] [SKPL-PKKPT.K-0002]  [SKPL-PKKPT.K-0003] |
| *Administrator* | Melakukan *login* data dan melihat data laporan secara keseluruhan. | [SKPL-PKKPT.K-0004] [SKPL-PKKPT.K-0005] |

## Batasan-batasan

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

1. Kebijaksanaan umum organisasi/lingkungan

2. Keterbatasan karena perangkat keras, contohnya kebutuhan *signal timing*

3. Standar antarmuka ke aplikasi atau sistem lain

4. Tuntutan pengoperasian secara paralel atau *multi platform*

## Asumsi dan Kebergantungan

Bagian ini mengungkapkan setiap faktor 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.

# Deskripsi Rinci Kebutuhan Perangkat Lunak

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, opoerator atau sistem eksternal lain. Kebutuhan ini harus melibatkan paling tidak:

1. deskripsi dari setiap masukan ke sistem (stimulus)

2. deskripsi dari setiap keluaran dari sistem (respon)

3. 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:

1. Semua kebutuhan rinci harus dinyatakan sesuai dengan karakteristik kebutuhan yang baik (lihat GL01)

2. Semua kebutuhan khusus harus sedapat mungkin diacusilangkan dengan dokumen sebelumnya yang berhubungan (dengan kata lain sesuai dengan dokumen yang diacu)

3. Semua kebutuhan harus dapat diidentifikasikan secara unik.

4. Organisasi pernyataan kebutuhan harus sedemikian yang memaksimalkan kemudahan pembacaan (*readability*).

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

### Antarmuka Pemakai

Bagian ini berisi hal-hal berikut:

1. Karakteristik logis dari setiap antarmuka antara produk perangkat lunak dan penggunanya. 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.
2. 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 diverifikasi. 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”.

### Antarmuka Perangkat Keras

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.

### Antarmuka Perangkat Lunak

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:

1. Nama
2. Mnemonic
3. Nomor spesifikasi
4. Nomor Versi
5. Sumber

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

1. Tujuan menghubungkan perangkat lunak tersebut dengan perangkat lunak yang dispesifikasikan.
2. 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.

### 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.

## 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.

### Aliran Informasi

Bagian ini mencantumkan dan menguraikan DFD level demi level.

#### DFD Level 1

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.

#### DFD Level 2 dst

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.

### Deskripsi Proses

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

#### Proses 1

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

1. Validasi terhadap masukan
2. Urutan pasti dari operasi
3. Tanggapan atas situasi abnormal termasuk *overflow*, fasilitas untuk komunikasi atau penanganan kesalahan (*error handling*) dan pemulihan (*recovery*).
4. Efek dari keberadaan dan nilai parameter
5. 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.

#### Proses 2 dst

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

## 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:

1. Batasan integritas

2. Frekuensi pemakaian

3. 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 terhubungkan langsung di ERD. Pada kasus ini, representasi data tetap diuraikan tetapi secara terpisah dari ERD.

### Data 1

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:

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

### Data 2 dst

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

## 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.

### 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:

1. Jumlah terminal yang didukung

2. Jumlah pengguna simultan yang didukung

3. Jumlah dan tipe informasi yang ditangani

Kebutu han numerik statik sering diidentifikasi pada bagian terpisah ayng disebut kapasitas. Kebutuhan numerik dinamik mungkin dapat melibatkan, sebagai contoh, jumlah transaksi dan tugas dan jumlah hdata 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.

### Batasan Memori

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

### 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:

1. Berbagai variasi modus operasi dalam organisasi pengguna, misalnya operasi yang bersifat *user-initiated* (inisiatif dari pengguna)
2. Periode operasi interaktif dan periode operasi *offline*
3. Fungsi pendukung untuk pemrosesan data
4. Operasi *backup* dan *recovery*

### Kebutuhan Adaptasi Lokasi

Bagian ini dapat berisi:

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

## 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.

### Kehandalan *(reliability)*

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

### Ketersediaan *(Availability)*

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

*checkpoint, recovery,* dan *restart*.

### Keamanan (*Security*)

Bagian ini berisi faktor untuk memproteksi perangkat lunak dari akses, penggunaan, pengubahan, 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

### *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.

### *Portability*

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

1. Persentase komponen yang berisi kode yang bergantung pada *host*

2. Persentase kode yang bergantung pada *host*

3. Penggunaan bahasa yang kepemindahannya terbukti

4. Penggunaan suatu kompilator tertentu atau subset bahasa tertentu

5. Penggunan suatu sistem operasi tertentu

## 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:

1. Format laporan
2. Penamaan data
3. Prosedur akunting
4. 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, kakas pengembangan, bahasa pemrograman, kompilator) bila memang merupakan tuntutan yang ditentukan oleh pelanggan

# Matriks Kerunutan

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.

# Informasi Tambahan

Dukungan informasi yang membuat SKPL mudah digunakan, antara lain:

1. Daftar isi
2. Index
3. Lampiran

## Daftar Isi dan Indeks

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

## 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 dipecahkanoleh 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.