

**PANDUAN PENGISIAN
SPESIFIKASI KEBUTUHAN DAN PERANCANGAN
PERANGKAT LUNAK
BERORIENTASI OBJEK**



Program Studi Teknik Informatika
Politeknik Negeri Batam
Jl. Ahmad Yani, Batam 29461

Daftar Isi

| | | |
|----------|--|----------|
| 1 | PENDAHULUAN | 3 |
| 2 | REFERENSI | 3 |
| 3 | DEFINISI..... | 3 |
| 4 | PENJELASAN TEMPLATE DOKUMEN | 4 |
| 4.1 | PENDAHULUAN | 5 |
| 4.1.1 | Tujuan | 5 |
| 4.1.2 | Lingkup Masalah..... | 6 |
| 4.1.3 | Definisi, Akronim dan Singkatan | 6 |
| 4.1.4 | Aturan Penamaan dan Penomoran | 6 |
| 4.1.5 | Referensi | 6 |
| 4.1.6 | Ikhtisar Dokumen..... | 6 |
| 4.2 | DESKRIPSI UMUM PERANGKAT LUNAK | 6 |
| 4.2.1 | Deskripsi Umum Sistem..... | 7 |
| 4.2.2 | Proses Bisnis Sistem | 7 |
| 4.2.3 | Karakteristik Pengguna..... | 7 |
| 4.2.4 | Batasan | 8 |
| 4.2.5 | Rancangan Lingkungan Implementasi..... | 8 |
| 4.3 | DESKRIPSI RINCI KEBUTUHAN | 8 |
| 4.3.1 | Deskripsi Fungsional | 8 |
| 4.3.2 | Deskripsi Kebutuhan Non Fungsional..... | 9 |
| 4.4 | DESKRIPSI KELAS-KELAS | 10 |
| 4.4.1 | Class Diagram | 10 |
| 4.4.2 | Class <nama class> | 10 |
| 4.4.3 | Class <nama class> dan seterusnya | 11 |
| 4.4.4 | State Machine Diagram | 11 |
| 4.5 | DESKRIPSI DATA | 11 |
| 4.5.1 | Entity-Relationship Diagram | 11 |
| 4.5.2 | Daftar Tabel | 11 |
| 4.5.3 | Struktur Tabel <nama tabel> | 11 |
| 4.5.4 | Struktur Tabel <nama tabel> dan seterusnya | 12 |
| 4.5.5 | Skema Relasi Antar Tabel..... | 12 |
| 4.6 | PERANCANGAN ANTARMUKA | 12 |
| 4.6.1 | Antarmuka <nama antarmuka> | 12 |
| 4.6.2 | Antarmuka <nama antarmuka> dan seterusnya | 12 |
| 4.7 | MATRIKS KETERUNUTAN | 13 |

1 Pendahuluan

Dokumen ini berisi penjelasan pemakaian dan penulisan dokumen Spesifikasi Kebutuhan dan Perancangan Perangkat Lunak (SKPPL) atau *Software Requirements and Design Specification (SRDS)* dengan pendekatan (ancangan) berorientasi objek. Dokumen ini selanjutnya akan menggunakan istilah SKPPL. Dokumen ini sebagian besar adalah adaptasi dari dokumen IEEE Std 830-1993.

Uraian yang dituangkan di dalam dokumen ini digunakan sebagai acuan dalam menulis SKPPL. Dokumen ini dibuat untuk membantu membuat spesifikasi dan pengembangan perancangan perangkat lunak yang akan dikembangkan dengan ancangan berorientasi objek. Pada prinsipnya, hasil analisis dan perancangan sistem perangkat lunak dengan ancangan ini diuraikan sebagai sekumpulan objek yang saling berinteraksi untuk menjalankan misi sistem. Interaksi antar objek dilakukan melalui suatu pemanggilan pesan biasa (yang berakibat dijalankannya suatu operasi/metode dari suatu objek tertentu) atau karena adanya *event* tertentu. Objek-objek tersebut terklasifikasi menjadi kelas-kelas dan terorganisasi secara statis. Analisis kebutuhan dan perancangan perangkat lunak hanya mempertimbangkan objek-objek (dan pada akhirnya kelas-kelas) yang mewakili gejala, hal, benda, atau fenomena yang ada pada ranah persoalan.

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).
- Jurusan Teknik Informatika – Institut Teknologi Bandung Panduan GL01A, Panduan Penggunaan dan Pengisian Spesifikasi Kebutuhan Perangkat Lunak.
- Jurusan Teknik Informatika – Institut Teknologi Bandung Panduan GL02A, Panduan Penggunaan dan Pengisian Deskripsi Perancangan Perangkat Lunak.
- Booch, Grady. James Rumbaugh, Ivar Jacobson. *The Unified Modeling Language User Guide*. Addison Wesley. 1999.
- Rumbaugh, James. Ivar Jacobson, Grady Booch. *The Unified Modeling Language Reference Manual*. Addison Wesley. 1999.

3 Definisi

Definisi dari istilah yang 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.

| | |
|-------|--|
| SKPPL | Spesifikasi Kebutuhan dan Perancangan Perangkat Lunak |
| SRDS | <i>Software Requirements and Design Specification (SRDS)</i> |
| UML | <i>Unified Modelling Language</i> |
| ERD | <i>Entity Relationship Diagram</i> |
| DBMS | <i>Data Base Management System</i> |

4 Penjelasan Template Dokumen

SKPPL berorientasi objek ini tidak didasarkan pada penggunaan metode tertentu melainkan menggunakan asumsi bahwa metode analisis berorientasi objek secara prinsip menggunakan notasi atau representasi *Unified Modelling Language* (UML).

Notasi dasar yang digunakan pada struktur SKPPL ini adalah notasi yang digunakan pada *use case diagram*, *class diagram*, *interaction diagram* (baik dalam bentuk *communication diagram* maupun dalam bentuk *sequence diagram*), serta bila perlu *state machine diagram*. Cara penggunaan dan definisi lebih lanjut dari UML dapat dilihat pada (Booch99) dan (Rumbaugh99).

Notasi berorientasi objek lain pada SKPPL ini, seperti notasi Coad Yourdon, Object Modelling Techniques, atau Booch, juga dapat digunakan. Penempatan semua hasil produk dengan menggunakan notasi-notasi pelengkap ini dapat dilakukan sesuai kebutuhan pembuat SKPPL. Teknik pemadanan langsung antara notasi UML dengan notasi lain tersebut juga bisa diterapkan, misalnya notasi kelas Coad Yourdon digunakan untuk menggantikan notasi kelas UML pada bagian Diagram Kelas.

SKPPL ini secara prinsip diuraikan berdasarkan *outline* seperti berikut ini.

Daftar Isi

1. Pendahuluan
 - 1.1. Tujuan
 - 1.2. Lingkup Masalah
 - 1.3. Definisi, Akronim dan Singkatan
 - 1.4. Aturan Penamaan dan Penomoran
 - 1.5. Referensi
 - 1.6. Ikhtisar Dokumen
2. Deskripsi Umum Perangkat Lunak
 - 2.1. Deskripsi Umum Sistem
 - 2.2. Fungsi Produk
 - 2.3. Karakteristik Pengguna
 - 2.4. Batasan
 - 2.5. Rancangan Lingkungan Implementasi
3. Deskripsi Rinci Kebutuhan
 - 3.1. Deskripsi Fungsional
 - 3.1.1. Use Case Diagram

- 3.1.2. Use Case <nama use case>
 - 3.1.2.1. Skenario
 - 3.1.2.2. Interaksi Objek
 - 3.1.3. Use Case <nama use case> dan seterusnya
 - 3.1.3.1. Skenario
 - 3.1.3.2. Interaksi Objek
 - 3.2. Deskripsi Non Fungsional
- 4. Deskripsi Kelas-kelas
 - 4.1. Class Diagram
 - 4.2. Class <nama class>
 - 4.3. Class <nama class> dan seterusnya
 - 4.4. State Machine Diagram
- 5. Deskripsi Data
 - 5.1. Entity-Relationship Diagram
 - 5.2. Daftar Tabel
 - 5.3. Struktur Tabel <nama tabel>
 - 5.4. Struktur Tabel <nama tabel> dan seterusnya
 - 5.5. Skema Relasi Antar Tabel
- 6. Perancangan Antarmuka
 - 6.1. Antarmuka <nama antarmuka>
 - 6.2. Antarmuka <nama antarmuka> dan seterusnya
- 7. Matriks Keterunutan

4.1 Pendahuluan

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

1. Tujuan
2. Lingkup Masalah
3. Definisi, Akronim dan Singkatan
4. Aturan Penamaan dan Penomoran
5. Referensi
6. Ikhtisar Dokumen

Definisi dari istilah yang digunakan pada dokumen ini dibuat berdasarkan hasil terjemahan dari IEEE Std 610.12-1990.

4.1.1 Tujuan

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

4.1.3 Definisi, Akronim dan Singkatan

Harus memberikan penjelasan terhadap semua definisi, akronim dan singkat yang digunakan agar dapat menginterpretasikan SKPPL 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 Aturan Penamaan dan Penomoran

Bagian ini berisi aturan penomoran dan penamaan yang dipakai dalam dokumen ini jika ada (misalnya aturan penomoran kebutuhan, penomoran use case, penamaan file, dsb).

4.1.5 Referensi

Bagian ini harus memberikan:

- Daftar lengkap dari dokumen (baik itu berupa buku, panduan, atau spesifikasi/deskripsi lain) yang dirujuk pada dokumen SKPPL 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

4.1.6 Ikhtisar Dokumen

Bagian ini adalah ikhtisar dari dokumen SKPPL. Tuliskan sistematika pembahasan dokumen SKPPL ini. Pada bagian ini, dijelaskan pula tentang proses transformasi dari model UML ke dalam bentuk rancangan. dan penempatan notasi-notasi lain di luar notasi UML bila ada sesuai dengan metode analisis perangkat lunak yang digunakan.

4.2 Deskripsi Umum 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 lingkungan operasi perangkat lunak.

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

Bagian ini terdiri dari:

1. Deskripsi Umum Sistem
2. Fungsi Produk
3. Karakteristik Pengguna
4. Batasan
5. Rancangan Lingkungan Implementasi

4.2.1 Deskripsi Umum Sistem

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 SKPPL 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

4.2.2 Proses Bisnis Sistem

Bagian ini mengutarakan proses bisnis sistem yang utama. Nantinya proses bisnis sistem ini akan selaras dengan *use case* yang ada pada *use case diagram*. Untuk menggambarkan proses bisnis sistem, digunakan *Activity Diagram*.

Sebagai contoh SKPPL untuk perangkat lunak apotek, bagian ini digunakan untuk menjelaskan secara umum tentang pengelolaan obat, penerimaan resep, pendaftaran pemasok, serta activity diagram untuk masing-masing proses 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). Minimal seperti tabel berikut.

| Pengguna | Tanggung Jawab | Hak Akses ke aplikasi | Kemampuan yang harus dimiliki |
|----------|----------------|-----------------------|-------------------------------|
| | | | |
| | | | |
| | | | |

4.2.4 Batasan

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

1. Kebijakan 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*

4.2.5 Rancangan Lingkungan Implementasi

Bagian ini berisi Sistem Operasi, DBMS, Development Tools, Filling System, Bahasa Pemrograman yang dipakai.

4.3 Deskripsi Rinci Kebutuhan

Bagian SKPPL 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 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

Prinsip-prinsip yang harus digunakan:

1. Semua kebutuhan rinci harus dinyatakan sesuai dengan karakteristik kebutuhan yang baik.
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 diidentifikasi secara unik.
4. Organisasi pernyataan kebutuhan harus sedemikian yang memaksimalkan kemudahan pembacaan (*readability*).

4.3.1 Deskripsi Fungsional

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

4.3.1.1 Use Case Diagram

Untuk analisis yang dilakukan dengan orientasi objek, khususnya jika menggunakan tools UML,

maka kebutuhan ini dapat dinyatakan dengan *Use Case Diagram*. Untuk satu sistem secara keseluruhan digambarkan sebuah diagram yang mewakili kebutuhan fungsional sistem secara keseluruhan.

4.3.1.2 Use Case <nama use case>

Setiap *use case* mewakili satu kebutuhan fungsional sistem. Bagian ini menjelaskan lebih lanjut tentang sebuah *use case*. Beri judul yang sesuai dengan nama *use case* yang dijelaskan. Setiap *use case* umumnya melibatkan setidaknya satu aktor atau lebih. Untuk satu *use case* harus dideskripsikan skenario dan dinamika objek yang mungkin terjadi pada saat kebutuhan itu diakses oleh pengguna.

4.3.1.2.1 Skenario

Bagian ini berisi skenario untuk satu *use case*. Di dalam skenario ini harus tercakup: siapa aktornya, tujuan use-case, ringkasan cerita, kondisi awal (*pre-condition*), kondisi akhir (*post-condition*), urutan aksi normal (*normal course of event* atau *success scenario*), urutan aksi alternatif (alternatif kejadian – *alternative course of event*), aksi aktor, dan reaksi sistem. Skenario dapat dituliskan secara bebas, maupun dengan terstruktur seperti contoh berikut:

| Identifikasi | |
|---------------------|---------------|
| Nomor | |
| Nama | |
| Tujuan | |
| Deskripsi | |
| Aktor | |
| Skenario Utama | |
| Kondisi awal | |
| Aksi Aktor | Reaksi Sistem |
| | |
| | |
| | |
| Skenario Alternatif | |
| Aksi Aktor | Reaksi Sistem |
| | |
| Kondisi akhir | |

4.3.1.2.2 Interaksi Objek

Bagian ini dapat digambarkan dengan berisi *Sequence Diagram* atau *Communication Diagram* yang menggambarkan dinamika dari objek-objek yang teridentifikasi dari suatu skenario *use case* yang telah diuraikan sebelumnya

4.3.1.3 Use Case <nama use case> dan seterusnya

Cantumkan dan beri penjelasan use case kedua dan seterusnya. Beri judul yang sesuai dengan nama *Use Case* yang dijelaskan.

4.3.2 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 menjadi sub-bab untuk per kebutuhan.

| SKPPL-Id | Parameter | Kebutuhan |
|-----------|-------------------|---|
| SKPPL-N01 | Availability | Aplikasi ini harus dapat beroperasi terus menerus selama 7 hari per minggu, 24 jam per hari tanpa berhenti, karena aplikasi ini akan bersifat <i>web-based</i> dan akan diakses oleh mahasiswa yang membutuhkan dari berbagai tempat pada waktu yang berbeda-beda. |
| SKPPL-N02 | Reliability | Aplikasi ini harus dibangun dengan kehandalan yang setinggi mungkin meskipun tidak perlu setinggi kehandalan sebuah <i>critical application</i> . Kegagalan yang dapat ditoleransi kurang lebih 10%. Dengan kahandalan yang tinggi diharapkan aplikasi ini dapat digunakan dengan baik pada saat dibutuhkan. Kehandalan yang dimiliki oleh aplikasi ini juga akan sangat bergantung pada beberapa hal eksternal, seperti kehandalan jaringan telekomunikasi yang digunakan untuk akses internet, kehandalan 10sistem daya listrik yang digunakan, dll. |
| SKPPL-N03 | Ergonomy | Aplikasi ini harus memiliki nilai ergonomi/ kenyamanan dipakai yang tinggi bagi user. Aplikasi akan dibangun dengan antarmuka user yang mudah dimengerti, indah dilihat, konsisten, mudah dioperasikan dan tidak membingungkan. |
| SKPPL-N04 | Portability | |
| | Memory | |
| SKPPL-N05 | Response time | |
| | Safety | |
| SKPPL-N06 | Security | |
| SKPPL-N07 | Bahasa komunikasi | |
| SKPPL-N08 | Lain-lain | |

4.4 Deskripsi Kelas-Kelas

Bagian ini mendeskripsikan kelas-kelas yang disarikan dari berbagai objek yang teridentifikasi. Deskripsi kelas ini terdiri atas diagram kelas yang menggambarkan struktur statis antar kelas dan spesifikasi setiap kelasnya.

4.4.1 Class Diagram

Bagian ini berisi diagram kelas beserta struktur yang lengkap. Harus disertakan juga deskripsi atau penjelasan tentang struktur yang dihasilkan (*generalisasi-spesialisasi*, *agregasi* atau *whole-part*, asosiasi, dan lainnya sesuai dengan metode analisis berorientasi objek yang digunakan).

4.4.2 Class <nama class>

Bagian ini menjelaskan secara rinci tentang sebuah kelas yang teridentifikasi. Beri judul yang sesuai dengan Nama Kelas. Spesifikasi kelas meliputi atribut dan operasi.

Spesifikasi atribut memuat spesifikasi atribut-atribut dasar dari kelas yang teridentifikasi. Yang disebut dengan atribut dasar adalah atribut yang memang perlu ada dan mewakili suatu fenomena/gejala alamiah dari sistem yang dimodelkan.

Bagian yang memuat spesifikasi operasi-operasi (disebut juga metode) dasar dari kelas yang teridentifikasi. Yang disebut dengan operasi dasar adalah operasi yang memang perlu ada dan mewakili suatu tugas alamiah dari objek yang dimodelkan. Uraikan deskripsi operasi dengan menggunakan narasi dengan bahasa alami atau dengan *pseudo-code*.

Nama Kelas:

| <i>Nama Atribut</i> | <i>Visibility (private, public)</i> | <i>Tipe</i> |
|----------------------------------|-------------------------------------|---|
| <i>Diisi dengan nama atribut</i> | | <i>Tuliskan tipenya sesuai dengan yang dikenal pada bahasa pemrograman yang digunakan</i> |
| | | |
| | | |

| <i>Nama Operasi</i> | <i>Visibility (private, public)</i> | <i>Keterangan</i> |
|----------------------------------|-------------------------------------|--|
| <i>Diisi dengan nama operasi</i> | | <i>Tuliskan deskripsi operasi atau menggunakan pseudo-code</i> |
| | | |
| | | |

4.4.3 Class <nama class> dan seterusnya

Cantumkan dan beri penjelasan kelas kedua dan seterusnya. Beri judul yang sesuai dengan Nama Kelas yang dijelaskan.

4.4.4 State Machine Diagram

Bagian ini hanya diisi jika ada kelas yang kompleks. Perubahan status kelas tersebut harus digambarkan dalam bentuk *state machine diagram*. Boleh dibuat subbab per kelas.

4.5 Deskripsi Data

Berisi deskripsi tabel-tabel data jika aplikasi berbasis data. Awali dengan entity-relationship diagram kemudian dilanjutkan dengan daftar tabel dan deskripsi isinya.

4.5.1 Entity-Relationship Diagram

Bagian ini berisi diagram entity-relationship beserta struktur yang lengkap. Untuk satu sistem secara keseluruhan digambarkan sebuah diagram yang mewakili deksripsi data untuk sistem secara keseluruhan.

4.5.2 Daftar Tabel

Berisi daftar dari tabel yang akan digunakan sebagai media penyimpanan data (*Data Storage*) untuk basis data yang dipergunakan oleh perangkat lunak.

4.5.3 Struktur Tabel <nama tabel>

Untuk setiap tabel, harus mengandung nama tabel, deskripsi isi, primary key, dan constraint integrity dengan tabel lain (jika ada). Beri judul sesuai tabel yang dijelaskan.

Nama tabel: ...

Primary key: ...

| Field | Tipe data | Ukuran | Key | Keterangan |
|-----------|------------|-----------|---------------------------------|--------------------|
| <i>id</i> | <i>int</i> | <i>50</i> | <i>Primary key, foreign key</i> | <i>id pengguna</i> |
| | | | | |

4.5.4 Struktur Tabel <nama tabel> dan seterusnya

Cantumkan dan beri penjelasan tabel kedua dan seterusnya. Beri judul yang sesuai dengan Nama tabel yang dijelaskan.

4.5.5 Skema Relasi Antar Tabel

Berisi skema rancangan hubungan fisik antar tabel pada basis data yang digunakan oleh perangkat lunak.

4.6 Perancangan Antarmuka

Bagian ini diisi dengan versi awal prototipe antarmuka. Selanjutnya, untuk setiap antarmuka/layar, tuliskan spesifikasi detilnya.

4.6.1 Antarmuka <nama antarmuka>

Dibuat satu sub bab untuk setiap antarmuka. Gambarkan rancangan antarmuka. Beri judul sesuai dengan nama antarmuka yang dijelaskan.

Sebutkan identitas antarmuka dan deskripsinya. Uraian minimal dapat dinyatakan dengan sebuah tabel dengan kolom:

1. Nama atau id objek
2. Jenis, misalnya: button, textbox.
3. Keterangan, diisi dengan penjelasan reaksi sistem, misalnya membuka layar apa, link kemana. Jika menyangkut suatu kode yang cukup rumit, rujuklah algoritma yang telah diuraikan di atas..

Contohnya seperti berikut:

| Id Objek | Jenis | Nama | Keterangan |
|----------------|----------------|---|---|
| | | <i>Diisi dengan string yg tampil pd layar</i> | <i>Diisi dengan penjelasan reaksi sistem, misalnya membuka layar apa, link kemana. Jika menyangkut suatu kode yang cukup rumit, acu algoritma yang telah diuraikan di atas.</i> |
| <i>Button1</i> | <i>Button</i> | <i>OK</i> | <i>Jika diklik, akan mengaktifkan Proses AlgoXXX.</i> |
| <i>RTF1</i> | <i>RTF Box</i> | | <i>Isi Teks yang disimpan pada File xxx</i> |

4.6.2 Antarmuka <nama antarmuka> dan seterusnya

Cantumkan dan beri penjelasan antarmuka kedua dan seterusnya. Beri judul yang sesuai dengan Nama antarmuka yang dijelaskan.

4.7 Matriks Keterunutan

Bagian ini menunjukkan hubungan dari kebutuhan perangkat lunak dan deskripsi perancangan. Matriks ini dapat digambarkan dalam bentuk tabel dengan kolom-kolom nama kebutuhan fungsional, nama use case, nama class yang terlibat, nama tabel yang digunakan, dan nama antarmuka. Contohnya seperti tabel berikut:

| Id kebutuhan fungsional | Id use case | Nama class | Nama tabel | Nama antarmuka |
|--------------------------------|--------------------|-------------------|-------------------|-----------------------|
| | | | | |
| | | | | |