

# SPESIFIKASI KEBUTUHAN PERANGKAT LUNAK

**E-COMMERCE USAHA MIKRO, KECIL DAN MENEGAH DI KOTA SURABAYA**

# E01-EUMKMS

26 November 2012 DRAF

**Disiapkan:**

Prima Arifandi 5210100049

M. Yordanis Salam 5210100104

Putri Cahyaning B 5210100142

sebagai kelengkapan mata kuliah Rancang Bangun Perangkat Lunak di Jurusan Sistem Informasi ITS

**Abstrak:**

Dokumen ini memuat spesifikasi kebutuhan perangkat lunak E-UMKMS (E-Commerce Usaha Mikro, Kecil dan Menengah di Kota Surabaya).

# Maklumat Versi Dokumen

Seluruh versi dari dokumen ini didaftar berdasar kronologisnya. Tidak ada hubungan antara nomer dokumen dan nomor versi perangkat lunak.

|  |  |  |  |
| --- | --- | --- | --- |
| **Versi Dokumen** | **Tanggal** | **Alasan Perubahan** | **Versi**  **Perangkat Lunak** |
| DRAF | 20/11/12 | Draf | V 1.0 |
|  |  |  |  |
|  |  |  |  |

Dokumen ini dibuat oleh Tim E-UMKMS dengan pengawasan dari Jurusan Sistem Informasi ITS sebagai upaya untuk menjamin keakurasian dokumen saat akan dicetak. Penggandaan dokumen, sebaiknya dari versi yang terakhir (up to date) dan setelah mendapatkan ijin tertulis

Kepala Laboratorium e-Bisnis

Jurusan Sistem Informasi

Fakultas Teknologi Informasi

Institut Teknologi Sepuluh Nopember

Surabaya

Copyright @ 2012 Jurusan Sistem Informasi ITS

Seluruh informasinya adalah hak milik Jurusan Sistem Informasi ITS yang tidak dipulikasikan dan bersifat rahasia.

# Control Revisi Dokumen

Seluruh revisi yang telah dilakukan pada dokumen ini, dapat diikuti sebagaimana tabel berikut.

|  |  |  |  |
| --- | --- | --- | --- |
| Nomer Revisi | Tanggal | Diperiksa oleh | Keterangan singkat perbaikan |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Daftar Isi

[SPESIFIKASI KEBUTUHAN PERANGKAT LUNAK 1](#_Toc341726777)

[E01-EUMKMS 1](#_Toc341726778)

[Maklumat Versi Dokumen ii](#_Toc341726779)

[Control Revisi Dokumen iii](#_Toc341726780)

[Daftar Isi iv](#_Toc341726781)

[1. Pendahuluan 1](#_Toc341726782)

[2. Deskripsi Umum Perangkat Lunak 1](#_Toc341726783)

[2.2.1 Reader Use Case 7](#_Toc341726784)

[3. Kebutuhan Antarmuka Eksternal (Specific requirements) 11](#_Toc341726785)

[4. Kebutuhan Fungsional (Specific requirements) 14](#_Toc341726786)

[5. Kebutuhan NonFungsional Lainnya (Specific requirements) 18](#_Toc341726787)

[6. Kebutuhan Lain 22](#_Toc341726788)

[Lampiran A: 24](#_Toc341726789)

[Organisasi Use Case berdasar Organizational Goals 24](#_Toc341726790)

[Lampiran B: 25](#_Toc341726791)

[Organisasi Use Case berdasar Importance Level 25](#_Toc341726792)

[Lampiran C: 26](#_Toc341726793)

[Organisasi Use Case berdasar System Functionality 26](#_Toc341726794)

[Lampiran D: Formal Requirement Model 27](#_Toc341726795)

[Lampiran E: GUI Storyboard 28](#_Toc341726796)

[Lampiran F: Use Case Model 29](#_Toc341726797)

[Lampiran G: Traceability Model 30](#_Toc341726798)

# 1. Pendahuluan

## Tujuan

**Tujuan**: Pembuatan Spesifikasi Kebutuhan Perangkat Lunak (SKPL) ini adalah untuk memberikan gambaran kebutuhan pengguna akan perangkat lunak yang akan dikembangkan dan memberikan petunjuk untuk menghindari kesalahan dalam tahapan perancangan perangkat lunak.

## Ruang Lingkup Dokumen

Dokumen SKPL ini akan menjelaskan tentang spesifikasi kebutuhan perangkat lunak yang akan dikembangkan. Isi dari dokumen ini yaitu deskripsi umum perangkat lunak, kebutuhan antarmuka eksternal perangkat lunak kebutuhan fungsional dan non-fungsional serta kebutuhan pendukung lainnya.

## Target Audience

Dokumen ini dapat digunakan untuk internal tim pengembangnya sebagai acuan dalam mengembangkan perangkat lunak. Selain itu, dokumen ini juga dapat digunakan untuk para stakeholder terkait dan *customer (user)* agar dapat mengetahui dengan jelas mengenai spesifikasi kebutuhan perangkat lunak yang akan dikembangkan.

## Definisi, Istilah, dan Singkatan

Guna memberikan gambaran yang sama terhadap beberapa definisi, istilah dan singkatan yang digunakan di dokumen ini, perlu dijelaskan sebagaimana berikut:

IEEE : Institute of Electrical and Electronics Engineers

Adalah standart internasional untuk pengembang perangkat lunak

SKPL : Spesifikasi Kebutuhan Perangkat Lunak atau

SRS : Software Requirement Specification

Adalah dokumen hasil analisis sebuah perangkat lunak yang berisi spesifikasi kebutuhan pengguna

## Sistem Penomoran

Ada beberapa hal/bagian dalam dokumen ini yang perlu diberi nomor. Maksud penomoran ini untuk mempermudah *audience* dalam pengidentifikasian. Adapun aturan penomorannya sebagaimana tabel berikut:

|  |  |
| --- | --- |
| **Hal/Bagian** | **Aturan Penomoran** |
| Tabel/Data Store | Nomor berbentuk **TD99**, dimana **99** adalah nomor urut tabel atau data store  Contoh: TD11, TD12, TD29, TD31 dan sebagainya |
| Kebutuhan Fungsional | Nomor berbentuk **KF999.x**, dimana **999** adalah nomor urut struktur butir-butir pada kebutuhan fungsional. Sedangkan **x** adalah nomor berupa abjad dan sifatnya sebagai tambahan jika kebutuhan fungsional tersebut memiliki item turunannya.  Contoh: KF101, KF120, KF120.a, KF120.b dan sebagainya |
| Kebutuhan Non Fungsional | Nomor berbentuk **KnF99.x**, dimana **99** adalah nomor urut struktur butir-butir pada kebutuhan non fungsional. Sedangkan **x** adalah nomor berupa abjad dan sifatnya sebagai tambahan jika kebutuhan non fungsional tersebut memiliki item turunannya.  Contoh: KnF11, KnF12, KnF12.a, KnF12.b dan sebagainya |

## Referensi

Berikut adalah daftar acuan yang digunakan dalam pendokumentasian spesifikasi kebutuhan perangkat lunak ini.

* IEEE Std. 1233, 1998 Edition IEEE Guide for Developing System Requirements Specifications
* IEEE Std 830-1998 (Revision of IEEE Std 830-1993), “IEEE Recommended Practice for Software Requirements Specifications”
* James L. Goldman, George Abraham, and Il­Yeol Song (2007), “Generating  Software  Requirements  Specification  (IEEE­ Std. 830­1998) document  with  Use Cases.
* Goldman, J. And Song, I. (2005), “Organizing and Managing Use Case,” in  the  First  International  Workshop  on  Best  Practices  of  UML,  Oct. 26­28, 2005, Klagenfurt, Austria, (in Perspectives in Conceptual Modeling,  LNCS Vol. 3770, Editor:  J. Akoka, etc.  Springer Verlag,  2005),  pp. 43­52.
* Holil, Achmad. Template: Spesifikasi Kebutuhan Perangkat Lunak, Jurusan Sistem Informasi ITS, 2006.

## Ikhtisar Dokumen

Dokumen SKPL ini dibagi menjadi tiga bagian utama. Bagian pertama berisi penjelasan tentang dokumen SKPL yang mencakup tujuan pembuatan dokumen ini, lingkup masalah yang diselesaikan oleh perangkat lunak yang dikembangkan, definisi, referensi dan ikhtisar dokumen.

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.

# 2. Deskripsi Umum Perangkat Lunak

E-UMKMS (E-Commerce Usaha Mikro, Kecil dan Menengah di Kota Surabaya) adalah perangkat lunak yang berupa website e-commerce. E-UMKMS ditujukan untuk para pelaku bisnis UMKM (Usaha Mikro, Kecil dan Menengah) di Surabaya. Dengan menggunakan E-UMKMS, UMKM di kota Surabaya dapat memasarkan produknya secara online. Fitur-fitur yang akan disajikan dalam website ini antara lain :

1. Menu Home
2. Menu Profil UMKM
3. Menu Produk
4. Fitur Add to Cart
5. Menu About Us
6. Menu Contact Us

Tujuan dibuatnya website ini yaitu mewadahi para pelaku bisnis UMKM di Surabaya untuk mepromosikan dan memasarkan produknya. Selain itu, website ini juga bertujuan agar masyarakat dapat lebih mengenal UMKM di Surabaya dan dapat mendukung pertumbuhan UMKM dengan membeli produk hasil karya UMKM.

## Perspektif Perangkat Lunak

Perangkat lunak dalam SKPL ini masih dalam tahap pembuatan, dimana sistem yang digunakan diadopsi dari sistem online shop.

* Berikut adalah contoh-contoh Deployment Diagram:
* Berikut adalah contoh-contoh Use Case Diagram:

## Fungsi-fungsi Perangkat Lunak

Berikut adalah penjelasan untuk masing-masing fungsi perangkat lunak E-UMKMS :

**Karakteristik dan Klasifikasi Pengguna**

<Memperkenalkan variasi klasifikasi pengguna yang akan mempergunakan pelarangkat lunak ini. Klasifikasi pengguna bisa dibedakan berdasarkan banyak pengguna, kumpulan pengguna fungsi perangkat lunak, keahlian teknis, keamanan atau pembagian hak akses. Terangkan karakteristik dan kebutuhan yang berhubungan dengan setiap klasifikasi pengguna. Karakteristik umum dari pengguna produk meliputi tingkat pendidikan, pengalaman, dan keahlian teknis yang dimiliki pengguna. Membedakan klasifikasi pengguna yang sangat penting untuk perangkat lunak ini, dengan klasifikasi pengguna yang dianggap kurang penting untuk dipuaskan.

**Pengguna produk tersebut dapat diambilkan dari aktor-aktor yang ada dalam Use Case Model**. Untuk membantu mendapatkan latar belakang aktor yang meliputi tingkat pendidikan, pengalaman, dan keahlian teknis yang dimilikinya dapat digunakan Use Case Diagram yang menggambarkan hubungan antara aktor dan usecase.>

**Lingkungan Operasi**

Perangkat lunak ini akan beroperasi apabila lingkungan operasinya mempunyai minimum requirement dengan spesifikasi hardware dan software berikut.

* Spesifikasi Hardware
* Processor Pentium IV
* RAM 250 MB
* Minimum Space Required 100 MB
* Display 16 bit color
* Spesifikasi Software
* Operating Environment Windows XP
* Platform .Net Framework & IIS Visual Studio 2008
* Database SQL Server 2005
* For User
* Internet Browser
* Internet Connection

<Gambaran lingkungan dimana perangkat lunak ini akan beroperasi, termasuk platform perangkat keras, versi dan sistem operasi, dan berbagai software atau aplikasi lain yang diperlukan untuk mendampinginya>

**Batasan Desain dan Implementasi**

<Gambaran beberapa item atau isu yang dapat membatasi pengembangan perangkat lunak. Hal ini termasuk:

1. Kebijakan aturan (*Regulatory policies*);
2. Keterbatasan Perangkat Keras (*Hardware limitations; e.g., signal timing requirements*);
3. Interface untuk aplikasi lain (*Interfaces to other applications*);
4. Pengoperasian Paralel (*Parallel operation*);
5. Fungsi Audit (*Audit functions*);
6. Fungsi Kontrol (*Control functions*);
7. *Higher-order language requirements*;
8. *Signal handshake protocols (e.g., XON-XOFF, ACK-NACK)*;
9. Persyaratan Keandalan (*Reliability requirements*);
10. Kekritisan dari aplikasi (*Criticality of the application*);
11. Pertimbangan Keselamatan dan Keamanan (*Safety and security considerations*).
12. konvensi desain atau standart pemrograman (contohnya jika organisasi customer/pengguna akan bertanggung jawab dalam pemelihara perangkat lunak yang telah diberikan)

Batasan (constraint) kadang disebut juga sebagai kebutuhan non fungsional yang harus dipenuhi oleh produk. >

**Dokumentasi Bagi Pengguna**

<Daftar komponen-komponen dokumentasi yang diperuntukkan kepada pengguna (seperti: user manual, bantuan on-line, dan tutorials) dan yang akan diberikan bersama-sama perangkat lunaknya>

**Asumsi dan Ketergantungan**

<Daftar beberapa asumsi yang akan mempengaruhi beberapa kebutuhan dalam SKPL ini. Termasuk di dalamnya third-party atau komponen komersil dalam perencanaan yang digunakan, isu-isu atau batasan tentang pengembangan atau lingkungan operasi. Pekerjaan pembuatan SKPL ini akan terpengaruh jika asumsi-asumsinya tidak benar, tidak *shared*, atau berubah. Tunjukkan juga ketergantungan terhadap faktor eksternal, seperti komponen-komponen perangkat lunak yang dimaksudkan untuk digunakan pada proyek lain, kecuali kalau telah disiapkan pada dokumen lain (misalnya dalam dokumen perencanaan proyek). Sebagai contoh, asumsi bahwa sistem operasi tertentu akan tersedia pada perangkat keras yang ditunjuk untuk digunakan oleh produk perangkat lunak. Jika pada kenyataannya sistem operasi tidak tersedia, maka SKPL harus disesuaikan.

**Beberapa asumsi biasanya dinyatakan dalam prasyarat (*preconditions*) dari usecase, terutama yang berkaitan dengan eksternal sistem** (aktor atau sistem lain). Atau yang terkait dengan saling ketergantungan dari usecases.>

**Klasifikasi Kebutuhan**

< Pada sub bagian ini SKPL harus dapat mengidentifikasi kebutuhan yang mungkin tidak diimplementasikan pada sistem yang dibuat saat ini atau ditunda pada versi selanjutnya. Serta untuk membedakan kebutuhan fungsional dan non fungsional. Untuk mengidentifikasi hal tersebut dapat digunakan metode yang ditawarkan Goldman  and Song dalam jurnalnya yang berjudul “Organizing and Managing Use Cases”, yaitu membuat tabel daftar usecase yang dilengkapi tiga kolom yang memuat informasi kategori dari masing-masing usecase tersebut.

Kolom kategori pertama menginformasikan kategori usecase berdasarkan *Organizational Goals*, kolom akan berisi nilai "C/A/R" yang menunjukkan apakah use case dianggap "Core", "Administrative", atau "Routine." (**Lihat Lampiran A**). Kolom kategori kedua menginformasikan kategori usecase berdasarkan *Importance Level*, kolom akan berisi nilai "P/S/O" yang menunjukkan apakah use case dianggap "Primary," "Secondary" atau "Optional." (**Lihat Lampiran B**). Kolom kategori kedua menginformasikan kategori usecase berdasarkan *System Functionality*, kolom akan berisi nilai angka “1..7” yang menunjukkan tipe fungsi generik dari usecase (Data entry/maintenance (1), Transaction Recording (2), Complex Calculation (3), Transformation (4), Communication (5), Device Control (6), System administration(7) ). (**Lihat Lampiran C**). Contoh tabel tersebut dapat dilihat pada **Tabel TD01**.

Tabel TD01.

Contoh Tabel Klasifikasi Kebutuhan berdasar Organizational Goals, Importance Level, dan Functionality System

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Use Case ID** | **Brief Use Case Description** | **C/A/R** | **P/S/O** | **G.F.T** |  |
| 1 | UC001 | Add subscriber information: a new subscriber’s information is added to the database. | C | P | 1 |  |
| 2 | UC002 | Add new subscription: a new subscription is  created in the system. | C | P | 2 |  |
| 3 | UC003 | Renew existing subscription: an existing  subscription is  renewed for additional time. | C | P | 2 |  |
| 4 | UC004 | Print fulfillment mailing labels: labels to be  attached to magazine issues for mailing. | C | P | 5 |  |
| 5 | UC005 | Log in: user identifies self with login ID and  password. | C | S | 7 |  |
| 6 | UC006 | Add advertiser: a new advertiser’s information  is added to the database. | R | P | 1 |  |
| 7 | UC007 | Add advertisement: an advertisement is  received from an advertiser. | C | P | 1 |  |
| 8 | UC008 | Schedule advertisement (process insertion  order): an advertisement is scheduled to  appear in  one or more magazine issues. | C | P | 2 |  |
| 9 | UC009 | Close issue: total the advertising revenue from scheduled ads in this issue and prohibit further advertisement insertions. | R | S | 4 | Bisa ditunda |
| 10 | UC010 | Back up certain data. | A | O | 7 |  |
| 11 | UC011 | Receive payment: payments are recorded for  subscriptions or for advertisements. | C | P | 2 |  |
| 12 | UC012 | Cancel subscription. | C | S | 2 |  |
| 13 | UC013 | Maintain issue publication schedule: record the dates on which tasks such as closing an issue and printing mailing labels must be completed  for each magazine issue. | A | S | 1 |  |
| 14 | UC014 | Print circulation report: print a summary report of descriptive statistics | R | S | 5 |  |
| 15 | UC015 | The system must be n% reliable: This is an  example  of a non ­functional  requirement  expressed  as  a goal ­oriented use case. | n/a | P | n/a | non fungsional |
| 16 | UC016 | Initialize the system: configure default settings and start the system application. | A | P | 7 |  |
| 17 | UC017 | Print loyalty report: print a report showing  customers ranked by longevity and number of magazines | R | O | 5 | Bisa ditunda |

**Kebutuhan Non Fungsional**

.........

# 3. Kebutuhan Antarmuka Eksternal (Specific requirements)

< Interface eksternal yang dijelaskan pada bagian ini lebih detail dibandingkan deskripsi interface yang ada pada sub bagian Perspektif Perangkat Lunak. Pada bagian ini skenario utama dan skenario alternatif usecase akan digambarkan dalam bentuk deskripsi naratif. Penggambarannya biasanya dalam format request and response, di mana tindakan aktor diikuti oleh satu atau lebih respon sistem, diikuti oleh tindakan aktor lanjutan dan tanggapan sistem, sampai selesainya use case.

Deskripsi naratif ini dapat diambilkan dari use case descriptions (narratives), dan **dapat dilengkapi dengan use case diagram** yang menggambarkan interaksi semua aktor dengan semua use case yang ada dalam ruang lingkup system.

Contoh pendeteksian antarmuka eksternal dari usecase “Process Sale Transaction and Payment”

|  |  |
| --- | --- |
| Actor Action | System Response |
| Cashier scans barcode on product box | System displays item description and current price on point of sale terminal |

Dari deskripsi usecase tersebut untuk keperluan SKPL dapat dituliskan sebagai berikut:

1. **Name of item**: “Cashier scans barcode on product box.”
2. **Description of purpose**: “System displays item description and current price on point of sale terminal.”
3. **Source of input**: Cashier (actor name)
4. **Valid range, accuracy, and/or tolerance**: as stated in preconditions
5. **Units of measure**: as stated in use case summary, or in the scenario narrative
6. **Timing**: shown by sequence of steps in the use case scenario narrative.
7. **Relationship to other inputs/outputs**: The most relevant related inputs/ outputs will be those that are also involved in the interactions within the same use case. Others may be separately noted.
8. **Screen formats/organization**: If required, these should be noted as system responses in the use case scenario narratives where appropriate. For example, a requirement for a credit card entry form could be described by this scenario:

|  |  |
| --- | --- |
| Actor Action | System Response |
| Cashier indicates credit card payment is desired | System displays empty credit card entry form in window |
| Cashier swipes credit card | System processes credit card |

1. Window formats/organization: If required, these should also be noted as  system responses where appropriate.
2. Data formats: These may be noted either in actor actions or in system  responses, depending on where the requirement applies. For example:  “System displays  customer’s  zip  code  left  justified;  hyphen  to  appear  after  first  five  digits  if nine digit zip code on file.”
3. Command formats:
4. End messages:  These  should  appear  as  the  last  system  response  in  the  scenario
5. narrative, and/or as described in the use case post­conditions.

>

**Antarmuka Pengguna**

<Uraikan karakteristik logik dari setiap antarmuka antara produk perangkat lunak dan penggunanya. Bisa berupa contoh gambar *screen*, beberapa standart GUI atau arahan bentuk yang harus diikuti, batasan *screen layout*, standart buttons dan function (misal help) yang akan kelihatan pada setiap *screen*, *keyboard shortcuts*, standart tampilan *error message*, dan yang lainnya. Tentukan komponen perangkat lunak yang diperlukan untuk antarmuka pengguna. Detail dari desain antarmuka pengguna ada pada dokumen terpisah yaitu spesifikasi antarmuka pengguna.

Yang dibahas pada disini antara lain:

1. Karakteristik logis dari setiap interface yang menjembatani antara user dengan produk perangkat lunak. Termasuk karakteristik konfigurasi (misal, format layar yang diperlukan, layout halaman atau jendela, isi laporan atau menu, atau ketersediaan tombol fungsi yang dapat diprogram) yang diperlukan untuk memenuhi kebutuhan dari produk perangkat lunak.
2. Semua aspek untuk mengoptimalkan interface dengan siapa saja yang akan menggunakan sistem. Ini mungkin hanya terdiri dari daftar apa yang boleh dan tidak boleh ditampilkan kepada pengguna. Salah satu contoh adalah kebutuhan akan pilihan akan ukuran pesan kesalahan (panjang atau pendek). Seperti yang lain, kebutuhan tersebut harus diverifikasi, misalnya, pernyataan kebutuhan "juru ketik level ke-4 dapat melakukan X fungsi dalam Z menit setelah 1 jam pelatihan" akan lebih baik daripada "juru ketik dapat melakukan fungsi X." Hal ini juga dapat ditentukan di bagian SKPL mengenai *Software System Attributes*  di sub bagian *Ease of use* (Kemudahan Penggunaan). >

**Antarmuka Perangkat Keras**

<Gambarkan karakteristik logik dan fisik dari setiap antarmuka antara produk perangkat lunak dan komponen perangkat keras dari sistem. Boleh berupa tipe peralatan pendukung, data alamiah dan kontrol interaksi antara perangkat lunak dan perangkat keras, dan protokol komunikasi yang digunakan.

Yang dijelaskan disini meliputi karakteristik logis dari setiap interface produk perangkat lunak dengan perangkat keras dari sistem. Ini termasuk karakteristik konfigurasi (jumlah port, instruction set, dll). Selain itu juga mencakup hal-hal seperti apa saja perangkat harus didukung, bagaimana caranya, dan menggunakan protokol apa. Sebagai contoh, dukungan terminal mungkin dapat ditetapkan untuk menampilkan full screen dari pada menampilkan baris demi baris.>

**Antarmuka Perangkat Lunak**

<Jelaskan koneksi antara perangkat lunak ini dengan komponen perangkat lunak tertentu lainnya (nama dan versi), termasuk database, sistem operasi, tools, libraries, dan komponen komersial yang terintegrasi. Tunjukkan item-item data atau pesan yang datang kepada sistem dan hasilnya dan gambaran dari penggunaan setiap hasil tersebut. Gambaran kebutuhan servis dan komunikasi. Menunjuk pada dokumen yang menguraikan detail pemrograman aplikasi interface protocol. Identifikasi data yang akan dibagi antar komponen perangkat lunak. Jika mekanisme pembagian data harus terimplementasi dengan cara yang khusus (contoh, penggunaan lingkungan data global si sistem operasi multitasking), terutama batasan implementasinya>

**Antarmuka Komunikasi**

<Uraikan asosiasi kebutuhan dengan beberapa fungsi komunikasi yang dibutuhkan oleh perangkat lunak ini, termasuk e-mail, web browser, protokol komunikasi network server, forms elektronik, dan lain sebagainya. Identifikasi beberapa hal yang berhubungan dengan format message. Identifikasi bebrapa standart komunikasi yang akan digunakan, seperti FTP atau HTTP. Menetapkan keamanan komunikasi atau isu tentang encrypsi, kecepatan transfer data, dan mekanisme sinkronisasi>

# 4. Kebutuhan Fungsional (Specific requirements)

\* “Functional Requirements” that should be further split into “Input Requirements” (related to user inputs, commands, etc.), “Output Requirements” (mostly related to the GUI), “Input/Output Requirements” (if they cannot be separated), and “Processing Requirements” 🡺 Organizing Use Cases by System Functionality

- Use Case Diagrams have to be included in most sections, specifically in the “Functional Requirements” section. Several Use Case Diagrams have to be presented, including specific scenarios, how the system will respond to certain user/operator requests or commands, or network behavior.”

<Setiap kebutuhan fungsional dari produk perangkat lunak akan digambarkan secara menyeluruh dan akan ditampilkan sebagai item kedua setelah judul/nama use case dengan format deskripsi use case.

Sebelumnya diawali dengan terlebih dahulu mengklasifikasikan kebutuhan fungsional (functional usecase) dalam tiga dimensi:

1. Organizing Use Cases by Organizational Goals
2. Organizing Use Cases by Importance Level
3. Organizing Use Cases by System Functionality

Setelah itu akan diberikan penjelasan usecase dalam bentuk Usecase Description/Narrative. Contoh Format Usecase Narrative dapat dilihat pada **Tabel 2**. Dibawah ini merupakan contoh penggambaran kebutuhan nya, dengan menggunakan contoh usecase yang ada pada **Tabel 1**.>

Tabel TD02.

Contoh Tabel Core Requirement / Use Cases

|  |  |  |  |
| --- | --- | --- | --- |
| **Core Requirement/Use Cases** | | | |
| Importance Level | System Functionality | Use Case Id | Brief Use Case Description |
| Primary | Data Entry/ Maintenance | UC001 | Renew existing subscription |
|  |  | UC007 | Add advertisement |
|  | Transaction Recording | UC002 | Add new subscription |
|  |  | UC003 | Renew existing subscription |
|  |  | UC008 | Schedule advertisement |
|  |  | UC011 | Receive Payment |
|  | Complex Calculation |  |  |
|  | Transformation |  |  |
|  | Communication | UC004 | Print fulfillment labels |
|  | Device Control |  |  |
|  | System Administration |  |  |
| Secondary | Data Entry/ Maintenance |  |  |
|  | Transaction Recording | UC012 | Cancel subscription |
|  | Complex Calculation |  |  |
|  | Transformation |  |  |
|  | Communication |  |  |
|  | Device Control |  |  |
|  | System Administration |  |  |
| Optional | Data Entry/ Maintenance |  |  |
|  | Transaction Recording |  |  |
|  | Complex Calculation |  |  |
|  | Transformation |  |  |
|  | Communication |  |  |
|  | Device Control |  |  |
|  | System Administration |  |  |

Tabel TD03.

Contoh Tabel Administrative Requirement / Use Cases

|  |  |  |  |
| --- | --- | --- | --- |
| **Administrative Requirement/Use Cases** | | | |
| Importance Level | System Functionality | Use Case Id | Brief Use Case Description |
| Primary | Data Entry/ Maintenance |  |  |
|  | Transaction Recording |  |  |
|  | Complex Calculation |  |  |
|  | Transformation |  |  |
|  | Communication |  |  |
|  | Device Control |  |  |
|  | System Administration | UC016 | Initialize the system |
| Secondary | Data Entry/ Maintenance | UC013 | Maintain issue publication sch |
|  | Transaction Recording |  |  |
|  | Complex Calculation |  |  |
|  | Transformation |  |  |
|  | Communication |  |  |
|  | Device Control |  |  |
|  | System Administration | UC005 | Log in |
| Optional | Data Entry/ Maintenance |  |  |
|  | Transaction Recording |  |  |
|  | Complex Calculation |  |  |
|  | Transformation |  |  |
|  | Communication |  |  |
|  | Device Control |  |  |
|  | System Administration | UC010 | Back up certain data |

Tabel TD04.

Contoh Tabel Routine Requirement / Use Cases

|  |  |  |  |
| --- | --- | --- | --- |
| **Routine Requirement/Use Cases** | | | |
| Importance Level | System Functionality | Use Case Id | Brief Use Case Description |
| Primary | Data Entry/ Maintenance | UC006 | Add advertiser |
|  | Transaction Recording |  |  |
|  | Complex Calculation |  |  |
|  | Transformation |  |  |
|  | Communication |  |  |
|  | Device Control |  |  |
|  | System Administration |  |  |
| Secondary | Data Entry/ Maintenance |  |  |
|  | Transaction Recording |  |  |
|  | Complex Calculation |  |  |
|  | Transformation | UC009 | Close issue |
|  | Communication | UC014 | Print circulation report |
|  | Device Control |  |  |
|  | System Administration |  |  |
| Optional | Data Entry/ Maintenance |  |  |
|  | Transaction Recording |  |  |
|  | Complex Calculation |  |  |
|  | Transformation |  |  |
|  | Communication | UC017 | Print loyalty report |
|  | Device Control |  |  |
|  | System Administration |  |  |

Tabel TD05.

Contoh Tabel Non-Functional Requirement / Use Cases

|  |  |  |  |
| --- | --- | --- | --- |
| **Non-Functional Requirement / Use Cases** | | | |
| Importance Level |  | Use Case Id | Brief Use Case Description |
| Primary |  | UC015 | The system must be n%  reliable |
| Secondary |  |  |  |
| Optional |  |  |  |

<Berikutnya, setiap kebutuhan atau usecase dibuat Usecase Narrative-nya dengan format seperti pada Tabel TD06.xx.>

Tabel TD06.xx

Format Use Case Narrative

|  |  |  |
| --- | --- | --- |
| **Use Case Name**:  ......................... | **Use Case Id**:  ................... | **Importance Level:**  ............... |
| Primary Actor:  ......................... | Use Case Type:  ...................... | |
| Stakeholders and Interest:  ......................... | | |
| Brief Description:  ......................... | | |
| Pre-Conditions:  ......................... | | |
| Trigger:  ......................... | | |
| Relationship:  ......................... | | |
| Normal Flow of Event: (Basic Course)  ........................  ........................ | | |
| Post-Conditions:  ....................... | | |
| Alternate Flow: (Alternate Course)  ......................  ...................... | | |

< Petunjuk pengisian dan pembuatan Use Case Narrative:

1. Use Case Name:

Nama Use Case

1. Use Case Id:

Nomor Identitas Use Case

1. Importance Level:

*<one of: summary, user-goal, subfunction> atau*

*( Sesuai aturan Organizing Use Cases by Organizational Goals (Core, Administrative, Routine) dan Organizing Use Cases by Importance Level (Primary, Secondary, Optional). Contoh: Core - Primary )*

1. Primary Actor:

*<a role name for the primary actor, or description>*

1. Use Case Type:

*(Sesuai aturan Organizing Use Cases by System Functionality (Data entry/ maintenance(1), Transaction Recording(2), Complex Calculation (3), Transformation(4), Communication (5), Device Control(6), System administration(7) )*)

1. Stakeholders and Interest:

*<list of stakeholders and key interests in the use case>*

1. Brief Description:

*(Deskripsi singkat dari use case)*

1. Pre-Conditions:

*<what we expect is already the state of the world>*

1. Trigger:

*<what starts the use case, may be time event>*

1. Relationship:

*<list association with other usecases>*

1. Normal Flow of Event (Basic Course):

*(sesuai aturan penulisan dalam metode Use Case Driven Object)*

1. Tulis deskripsi Basic Course use cases dengan kalimat aktif (*active voice*) dalam satu paragraf. Proses Use cases ketika berjalan dalam kondisi normal (*sunny day*).
2. Tulis deskripsi use cases dengan suatu pola aliran kejadian/respon (*event/response*) yang menggambarkan dialog kedua sisi antara pengguna dan sistem.
3. Tulis deskripsi use case dengan struktur kalimat Subyek-Predikat- Obyek-Keterangan Tempat (*Noun-Verb-Noun-Noun*). Keterangan tempat boleh juga ditulis di awak kalimat.
4. Sebutkan aktor (subyek) yang menggunakan use case, apa yang dikerjakan (predikat), serta obyek dan lokasinya.
5. Menggunakan referensi obyek yang ada dalam Domain Model (*High Level Class Diagram*).
6. Post Conditions:

*<the state of the world if goal succeeds>*

1. Alternate Flow (Alternate Course):

*(sesuai aturan penulisan dalam metode Use Case Driven Object)*

1. Tulis deskripsi Alternate Course use cases dengan kalimat aktif (*active voice*). Proses Use cases ketika berjalan dalam kondisi tidak normal (*rainy day*).
2. Tuliskan semua kondisi tidak normal yang mungkin dilakukan pengguna, beserta respon sistem.

>

# 5. Kebutuhan NonFungsional Lainnya (Specific requirements)

**Kebutuhan Kinerja**

< Pada sub bagian ini akan diberikan spesifikasi dari *static and the dynamic numerical requirements* pada perangkat lunak atau interaksi manusia dengan perangkat lunak secara keseluruhan. Kebutuhan numerik statis yang termasuk antara lain:

1. Jumlah terminal yang harus didukung
2. Jumlah pengguna simultan yang harus didukung
3. Jumlah dan jenis informasi yang akan ditangani

Kebutuhan numerik statis kadang diidentifikasi pada sub bagian yang terpisah dengan judul *Capacity.*

Kebutuhan numerik dinamis antara lain jumlah transaksi, jumlah kerja (task) dan jumlah data yang akan diproses dalam jangka waktu tertentu untuk kondisi beban kerja normal dan puncak.

Semua kebutuhan ini harus dinyatakan dalam istilah yang terukur.

Sebagai contoh:

95% dari transaksi akan diproses dalam waktu kurang dari 1 detik.

bukan,

Seorang operator tidak harus menunggu transaksi berjalan sampai selesai.

Sebenarnya kebutuhan ini dapat juga didokumentasikan menggunakan use case yang ada dalam UML, dengan sedikit menambahkan informasi yang mengakomodasi “goal-oriented” use cases.

Dalam usecase standar, informasi tentang kebutuhan tersebut harus ditangkap secara eksplisit dalam proses pengembangan use case. Karena “goal-oriented” use cases bukan merupakan reprentasi formal model use case di UML. >

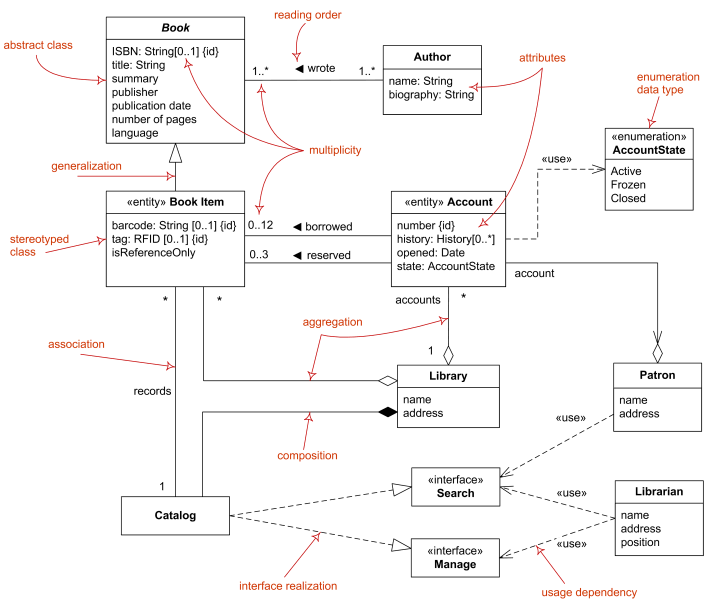
**Kebutuhan Database Logik**

< Bagian ini memberikan spesifikasi kebutuhan logik untuk informasi yang akan ditempatkan ke dalam database. Yang termasuk spesifikasi tersebut antara lain:

1. Jenis-jenis informasi yang digunakan oleh berbagai fungsi;
2. Frekuensi penggunaan;
3. Kemampuan mengakses;
4. Entitas data dan hubungannya;
5. Integrity constraints,
6. Data retention requirements

Karena usecase tidak secara spesifik memberikan spesifikasi database logis ketika digunakan untuk mendefinisikan fungsi sistem, maka akan **digunakan class diagram (atau minimal domain model)** untuk melengkapi aspek kebutuhan database.

Contoh Domain Diagram (High Level Class Diagram):



Gambar 11 (Domain diagram overview - classes, interfaces, associations, usage, realization, multiplicity)

>

**Batasan Disain**

< Bagian ini membahas batasan-batasan eksternal (external constraints) yang akan diberlakukan pada saat disain dan implementasi sistem. Batasan-batasan tersebut harus dinyatakan sebagai suatu kebutuhan. Batasan-batasan tersebut meliputi kepatuhan terhadap standar dan perundangan, keterbatasan perangkat keras, dan kebutuhan lainnya yang bukan merupakan kebutuhan non fungsional dan tidak termasuk dalam kebutuhan kinerja. Yaitu kebutuhan yang berkaitan dengan keandalan (reability), ketersediaan (availability), keamanan (security), kemudahan merawat (maintainability), dll.

Semua kebutuhan tersebut merupakan aspek-aspek non-functional goal-oriented dari sistem, yang harus didokumentasikan secara eksplisit karena tidak dibahas dalam use case.

Pada pembahasan tentang Kepatuhan pada standar (Standards Compliance) harus menyebutkan kebutuhan dari standar dan peraturan perundangan yang ada. Yang mungkin termasuk dalam hal itu antara lain:

a)Format Laporan (*Report format*)*;*

b) Penamaan Data (*Data naming*)*;*

c) Prosedur Akuntansi (*Accounting procedures*)*;*

d) *Audit tracing.*

Sebagai contoh, untuk menentukan kebutuhan dari perangkat lunak yang akan digunakan untuk melacak aktifitas pengolahan. Pelacakan tersebut dibutuhkan untuk mendapatkan peraturan atau standar keuangan minimal yang harus dipenuhi oleh aplikasi.

Contoh kebutuhan *Audit Tracing* adalah Kebutuhan yang menyatakan bahwa semua perubahan data dalam database penggajian harus disimpan dalam sebuah file log (*trace file*), yang disimpan adalah nilai sebelum dan sesudah perubahan.>

**Atribut Kualitas Perangkat Lunak**

< Ada sejumlah atribut perangkat lunak yang dapat dinyatakan sebagai kebutuhan. Untuk itu harus ditentukan atribut-atribut mana yang akan digunakan, sehingga keberadaannya dapat diverifikasi secara obyektif.

Contoh-contoh atribut itu antara lain:

*1 Reliability*(Keandalan)

Menentukan faktor-faktor yang dibutuhkan untuk menentukan keandalan yang dibutuhkan dari sistem perangkat lunak pada saat pengiriman.

*2 Availability*(Ketersediaan)

Menentukan faktor-faktor yang dibutuhkan untuk menjamin tingkat ketersediaan yang ditetapkan untuk seluruh sistem, seperti: *checkpoint, recovery, and restart.*

*3 Security*(Keamanan)

Menentukan faktor-faktor untuk melindungi perangkat lunak dari *accidental or malicious access, use, modification, destruction, or disclosure.*

Kebutuhan khusus untuk keperluan ini antara lain:

1. *Utilize certain cryptographical techniques;*
2. *Keep specific log or history data sets;*
3. *Assign certain functions to different modules;*
4. *Restrict communications between some areas of the program;*
5. *Check data integrity for critical variables.*

*4 Maintainability* (Kemudahan Merawat)

Menentukan atribut perangkat lunak yang berkaitan dengan kemudahan pemeliharaan perangkat lunak itu sendiri. Yang mungkin termasuk disini antara lain: *certain modularity, interfaces, complexity, etc.*

Kebutuhan-kebutuhan tersebut sebaiknya tidak dimunculkan hanya karena bertujuan supaya dianggap memenuhi *good design practices.*

*5 Portability* (Portabilitas)

Menentukan atribut dari perangkat lunak yang berhubungan dengan kemudahan memasang perangkat lunak pada komputer dan/atau sistem operasi lain. Yang termasuk dalam hal ini antara lain:

1. *Percentage of components with host-dependent code;*
2. *Percentage of code that is host dependent;*
3. *Use of a proven portable language;*
4. *Use of a particular compiler or language subset;*
5. *Use of a particular operating system.*

>

*Once the relevant characteristics are selected, a subsection should be written for each, explaining the rationale for including this characteristic and how it will be tested and measured. A chart like this might be used to identify the key characteristics (rating them High or Medium), then identifying which are preferred when trading off design or implementation decisions (with the ID of the preferred one indicated in the chart to the right).*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Characteristic** | **H/M/L** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** |
| 1 | Correctness |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | Efficiency |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | Flexibility |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | Integrity/Security |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 | Interoperability |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 | Maintainability |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 | Portability |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 8 | Reliability |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 9 | Reusability |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 10 | Testability |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 11 | Usability |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 12 | Availability |  |  |  |  |  |  |  |  |  |  |  |  |  |

*Definitions of the quality characteristics not defined in the paragraphs above follow.*

*• Correctness - extent to which program satisfies specifications, fulfills user’s mission objectives*

*• Efficiency - amount of computing resources and code required to perform function*

*• Flexibility - effort needed to modify operational program*

*• Interoperability - effort needed to couple one system with another*

*• Reliability - extent to which program performs with required precision*

*• Reusability - extent to which it can be reused in another application*

*• Testability - effort needed to test to ensure performs as intended*

*• Usability - effort required to learn, operate, prepare input, and interpret output*

**Aturan Penggunaan**

<Daftar beberapa prinsip pengoperasian perangkat lunak, seperti fungsi-fungsi yang dapat dilakukan seseorang pada situasi tertentu. Ingat, bukan untuk kebutuhan fungsional, tetapi yang menyatakan beberapa kebutuhan fungsional tertentu sebagai sebuah aturan>

# 6. Kebutuhan Lain

<Tentukan beberapa kebutuhan lain yang tidak tercover pada SKPL ini. Mungkin bisa termasuk kebutuhan database, kebutuhan menginter-nasionalisasikan, kebutuhan legal/hukum, penggunaan kembali pada sebuah proyek, dan sebagainya. Ditambah beberapa bagian yang relevan untuk SKPL ini>

**< Appendixes** (Lampiran)

Lampiran tidak selalu dianggap sebagai bagian dari SKPL yang harus ada dan tidak selalu diperlukan. Yang seharusnya ditampilkan pada bagian ini antara lain:

1. Contoh Format input/output, deskripsi studi analisis biaya, atau hasil survei pengguna;
2. Latar belakang atau Informasi pendukung yang dapat membantu para pembaca memahami SKPL;
3. Penjelasan mengenai masalah yang harus diselesaikan oleh perangkat lunak;
4. Instruksi khusus mengenai kemasan seperti kode dan media untuk memenuhi keamanan, ekspor, pemuatan awal, atau persyaratan lainnya.

Ketika lampiran yang disertakan, SKPL secara eksplisit harus menyatakan apakah lampiran merupakan bagian dari kebutuhan yang harus dipenuhi.>

***DiHAPUS SAJA***

# Lampiran A:

# Organisasi Use Case berdasar Organizational Goals

Song mengusulkan use case diklasifikasikan berdasarkan bidang Organizatioal Goals yang didukung use case tersebut apakah use case tersebut termasuk **core use cases, administrative use cases, atau routine use cases.**

**Core use cases** menjelaskan fungsionalitas baru atau khas yang dispesifikasikan untuk sistem. Setiap core use case menyediakan fungsionalitas yang nyata bagi pengguna sistem. Core use digunakan untuk menjawab pertanyaan “Apa yang sistem lakukan untuk kita?” Core use cases menunjukkan fungsi utama dari sistem.

**Administrative use cases** menjelaskan operasi-operasi yang diperlukan untuk mengintegrasikan seluruh operasi sistem. Yang termasuk administrative use cases antara lain “Back up the system,” “Shut down the system,” “Synchronize the remote databases,” dll.

**Routine use cases** menjelaskan operasi-operasi yang harus pengguna lakukan secara berulang untuk kepentingan fungsionalitas yang disediakan oleh sistem. Routine use cases tidak berdiri sendiri dalam menyediakan fungsionalitas yang merupakan tujuan dari sistem. Contoh routine use cases antara lain “Log in,” “Print usage log report,” “Change pass-word,” dll. Perbedaan Routine use cases dengan administrative use cases: routine use cases lebih sering digunakan ketika menjalankan sistem dalam kondisi normal; sedangkan administrative use cases lebih sering digunakan administrator administrators untuk melakukan prosedur operasi sistem atau untuk maintenance sistem.

Metode sederhana untuk mengklasifikasikan use case apakah termasuk core, administrative, atau routine, adalah dengan menjawab pertanyaan-pertanyaan berikut:

1. Apakah sistem dibuat untuk menyediakan fungsi yang dideskripsikan dalam use case ini? *(Was the system created in order to provide the functionality described in this use case?)*

Jika jawabannya “**Ya**”, maka use case **termasuk kategori Core use case**.

1. Apakah use case ini mendeskripsikan suatu operasi yang tanpanya seluruh sistem tidak akan beroperasi dengan baik? *(Does this use case describe an operation without which the entire system would not operate properly over time?)*

Jika jawabannya “**Ya**”, maka use case ini diperlukan oleh sistem untuk bisa beroperasi dengan tepat, sehingga use case tersebut **masuk kategori Administrative use case**.

1. Apakah use case ini mendeskripsikan fungsionalitas yang tanpanya pengguna tidak dapat memanfaatkan fitur dan interface sistem dengan benar? *(Does this use case describe functionality without which user would not be able to properly utilize the system’s features and interfaces?)*

Jika jawabannya “**Ya**”, maka use case ini **masuk kategori Routine use case**.

1. Jika jawaban untuk ketiga pertanyaan itu adalah “**Tidak**”, mungkin **use case tersebut tidak diperlukan**.

# Lampiran B:

# Organisasi Use Case berdasar Importance Level

Use cases dikategorikan berdasar tingkat kepentingannya yaitu use cases **primary**, **secondary**, atau **optional**.

**Perbedaan antara use cases primary dan secondary** adalah use case primary menyediakan fungsionalitas yang penting, dan sering dilakukan. **Use case Secondary** fungsionalitasnya kurang penting dan lebih jarang dilakukan. **Sedangkan use case Optional** fungsionalitasnya dibutuhkan tetapi tidak penting.

Metode untuk menentukan apakah use case merupakan primary, secondary, atau optional adalah dengan menjawab dua pertanyaan berikut:

1. Dapatkah sistem mulai dioperasikan jika use case ini belum diimplementasikan? *(Can system operation begin if this use case has not been implemented?)*

Jika jawabannya “Tidak”, maka use case ini merupakan kategori Primary use case.

1. Dapatkah sistem substansial memberikan nilai yang dimaksudkan kepada pengguna jika use case ini belum diimplementasikan? *(Can the system sub-stantially deliver its intended value to the users if this use case has not been implemented?)*

Jika jawabannya “Tidak”, maka use case ini merupakan kategori Secondary use case.

1. Jika jawabannya “Ya”, maka use case merupakan kategori Optional use case.

# Lampiran C:

# Organisasi Use Case berdasar System Functionality

1. **Data entry / maintenance.** Tujuan utama dari use cases kategori ini adalah untuk memanipulasi data dalam database. Contoh dari use case kategori ini adalah *Adding a new customer record, change customer data,* atau *up-dating an inventory*.
2. **Transaction processing.** Dalam suatu transaksi, pada saat transaksi diproses dan dicatat. Seringkali, transaksi tersebut harus diulang (*rolled back*). Banyak fungsionalitas sistem informasi bisnis yang melibatkan pengolahan dan merekam transaksi bisnis, seperti : *sales, refunds, hiring, registrations,* dan *reservations*.
3. **Complex Calculation.** Use cases kategori ini seringkali termasuk dalam sub bagian dari use case dasar, tetapi adakalanya juga menjadi use case primary. Sebagai contoh: sistem Payroll mengkin memiliki use case untuk menghitung jumlah pemotongan pajak yang beragam. Tax computations merupakan salah satu contoh use cases kategori ini.
4. **Transformation.** Sebuah transformasi, atau konversi, operasi dapat menyebabkan perubahan besar di banyak tabel dalam database atau file dalam sistem file. Ini dapat didasarkan pada aturan bisnis, perhitungan, atau perubahan lingkungan eksternal. Contoh yang mungkin termasuk dalam kategori ini adalah proses penutupan masa pajak atau merger yang terkait dengan perubahan konten dan struktur informasi. Use case kategori ini seringkali berkaitan dengan fungsi-fungsi untuk mempersiapkan kompatibilitas data dengan sistem eksternal.
5. **Communication.** Use cases yang tujuan utamanya untuk memindahkan informasi ke/dari luar sistem masuk dalam kategori ini. *Electronic mail* adalah contoh yang jelas, tetapi transfer data ke/dari sistem lain juga termasuk dalam kategori ini. Misalnya, menyiapkan dan mengirim pesan informasi order *eCommerce* ke dalam sistem rantai pasokan (*supply chain*) juga akan menjadi use case kategori ini. Laporan juga termasuk dalam hal komunikasi. Laporan adalah komunikasi satu arah dari sistem ke pengguna.
6. **Device control.** Ini adalah use cases yang digunakan terutama untuk menentukan kontrol dari perangkat fisik seperti lengan robot, bagian konveyor, dll.
7. **System administration.** Use cases kategori ini merupakan fungsi-fungsi yang diperlukan supaya sistem selalu bisa beroperasi dengan baik, tetapi use cases kategori ini tidak berkontribusi langsung pada tujuan bisnis yang akan dibuat. Yang termasuk dalam kategori ini meliputi *starting the system, backing up data, logging in,* dan *generating usage reports*.

# Lampiran D: Formal Requirement Model

<Daftar nomer-nomer kebutuhan yang dapat ditunjukkan pada SKPL ini, sehingga bisa di telusuri lagi 🡪 **Formal Requirement Model**> Bagaimana dengan Business Process Model?

# Lampiran E: GUI Storyboard

<Bisa.berupa model analisis yang relevan: **Deployment Diagram, Use Case Diagram, dan high level Class Diagram sudah ada. Butuh tambahan apa lagi??? GUI storyboard** >

# Lampiran F: Use Case Model

<Daftar nomer-nomer kebutuhan yang dapat ditunjukkan pada SKPL ini, sehingga bisa di telusuri lagi 🡪 **Use Case Model**>

# Lampiran G: Traceability Model

<Daftar nomer-nomer kebutuhan yang dapat ditunjukkan pada SKPL ini, sehingga bisa di telusuri lagi 🡪 **Traceabilty Model**>

USE CASE TEMPLATE

use case template from renowned guru Alistair Cockburn’s book Writing Effective Use Cases (New York: Addison-Wesley, 2000), p. 119.

USE CASE 24: FULLY DRESSED USE CASE TEMPLATE <NAME>

<the name should be the goal as a short active verb phrase>

**Context of use:**

<a longer statement of the goal, if needed, its normal occurrence conditions>

**Scope:**

<design scope, what system is being considered black-box under design>

**Level:**

<one of: summary, user-goal, subfunction>

**Primary Actor:**

<a role name for the primary actor, or description>

**Stakeholders & Interests:**

<list of stakeholders and key interests in the use case>

**Precondition:**

<what we expect is already the state of the world>

**Minimal Guarantees:**

<how the interests are protected under all exits>

**Success Guarantees:**

<the state of the world if goal succeeds>

**Trigger:**

<what starts the use case, may be time event>

**Main Success Scenario:**

<put here the steps of the scenario from trigger to goal delivery and any cleanup after>

<step #>

<action description>

**Extensions:**

<put here there [sic] extensions, one at a time, each referring to the step of the main scenario>

<step altered> <condition>: <action or sub use case>

<step altered> <condition>: <action or sub use case>

**Technology & Data Variations List:**

<put here the variations that will cause eventual bifurcation in the scenario>

<step or variation #> <list of variations>

<step or variation #> <list of variations>

**Related Information:**

<whatever your project needs for additional information>