**GL01**

SPESIFIKASI KEBUTUHAN PERANGKAT LUNAK

<Nama Perangkat Lunak>

untuk:

Dede Robiatul Adawiyah

Dipersiapkan oleh:

Lazuardi Firdaus

|  |  |  |
| --- | --- | --- |
| Nomor Dokumen | | Halaman |
| *GL01-SKPL* | | *<#>/<jml #* |
| Revisi | *A* | *Tgl: 30 Desember 2017* |

DAFTAR PERUBAHAN

|  |  |
| --- | --- |
| Revisi | Deskripsi |
| A |  |

Daftar Halaman Perubahan

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

Daftar Isi

1. Pendahuluan 5

1.1 Tujuan Penulisan Dokumen 5

1.2 Lingkup Masalah 5

1.3 Definisi, Istilah dan Singkatan 5

1.4 Aturan Penomoran 5

1.5 Referensi 5

1.6 Deskripsi umum Dokumen (Ikhtisar) 5

2 Deskripsi Umum Perangkat Lunak 6

2.1 Deskripsi Umum Sistem 6

2.2 Karakteristik Pengguna 6

2.3 Batasan 6

2.4 Lingkungan Operasi 6

3 Deskripsi Kebutuhan 7

3.1 Kebutuhan Antarmuka Eksternal 7

3.1.1 Antarmuka pemakai 7

3.1.2 Antarmuka Perangkat Keras 7

3.1.3 Antarmuka Perangkat Lunak 7

3.1.4 Antarmuka Komunikasi 7

3.2 Kebutuhan Fungsional 7

3.3 Kebutuhan Non Fungsional 7

3.4 Model Analisis 8

3.4.1 Diagram Use Case 8

3.4.1.1 Definisi Actor 8

3.4.1.2 Definisi Use Case 1 9

3.4.1.3 Definisi Use Case 2 9

3.4.1.4 Definisi Use Case 3 9

3.4.2 Diagram Kelas 9

3.4.2.1 Diagram Kelas 1 10

3.4.2.2 Diagram Kelas 2 10

3.4.2.3 Diagram Kelas 3 10

3.4.3 Diagram Sequence 10

4 Kerunutan (traceability) 11

4.1 Kebutuhan Fungsional vs Use Case 11

4.2 Kebutuhan Fungsional vs Diagram Kelas 11

*Setelah Daftar Isi Boleh ada Daftar Tabel dan Daftar Gambar*

# 1. Pendahuluan

## Tujuan Penulisan Dokumen

Dokumen ini berisi hasil analisa kebutuhan untuk sistem <nama sistem> yang telah dilakukan terhadap pengguna. Pada dokumen ini akan diberikan khususnya penjelasan setiap kebutuhan yang akan dipenuhi oleh sistem perangkat lunak. Dokumen ini akan menjadi acuan bagi pengembang dan pengguna tentang spesifikasi perangkat lunak yang akan dikembangkan.

## Lingkup Masalah

Aplikasi <nama sistem> akan digunakan untuk menfasilitasi uji sensory metode temporal dominance sensation terhadap makanan oleh seorang panelis. Aplikasi akan dioperasikan oleh panelis. Panelis pertama akan mulai melakukan pengunyahan terhadap makanan. Aplikasi akan menampilkan list atribut sensory di layar. Panelis akan mendeteksi sensasi dominan dari makanan selama pengunyahan dan menekan tombol pada aplikasi sesuai dengan sensasi tersebut. Aplikasi akan merekam data masukkan panelis dan akan mengeluarkan file yang berisi rekaman sensasi terhadap waktu untuk setiap panelis dan setiap pengulangan.

## Definisi, Istilah dan Singkatan

Semua definisi dan singkatan yang digunakan dalam dokumen ini dan penjelasannya. Definisi/singkatan itu biasanya terkait dengan sistem atau organisasi dari pihak pengguna/pelanggan (user/customer)

## Aturan Penomoran

Tuliskan jika anda memakai aturan penomoran

Contoh:

”Untuk dokumen pembangunan perangkat lunak ini akan menggunakan aturan penomoran sebagai berikut:

R-xx : untuk suatu kode kebutuhan (Requirement)

R-xx.yy: untuk sub-kebutuhan yang terkait dengan kebutuhan pertama”

## Referensi

Dokumentasi yang terkait dengan pekerjaan pengembangan. Misalnya dokumentasi berupa aturan pemerintah, aturan organisasi, buku panduan suatu sensor (bila terkait dengan sensor), dan lain-lain.

## Deskripsi umum Dokumen (Ikhtisar)

Tuliskan sistematika pembahasan dokumen SKPL ini.

Contoh:

”Pada bagian awal dijelaskan tentang sistem X secara umum.

Pada bagian 2, akan dijelaskan tentang deskripsi sistem, siapa saja pengguna sistem X, sistem Y, serta batasan pengembangan dan lingkup operasinya.

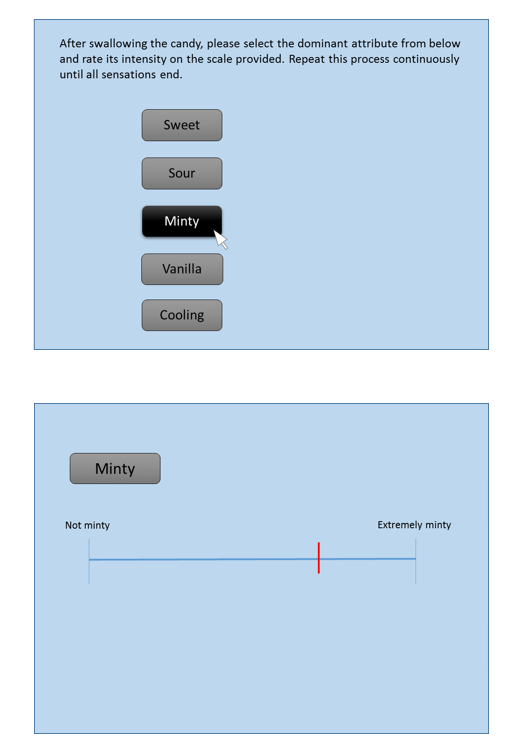
Pada bagian 3 berisi penjelasan lebih rinci tentang kebutuhan perangkat lunak, dimulai dari penjelasan antarmuka, hingga kebutuhan fungsional yang dibantu dengan pemodelan DFD. Untuk data akan diperjelas dengan diagram ER. Pada bagian ini juga akan menjelaskan kebutuhan non fungsional dan beberapa batasan dalam perancangan, dan daftar kerunutan kebutuhan perangkat lunak”

# Deskripsi Umum Perangkat Lunak

## Deskripsi Umum Sistem

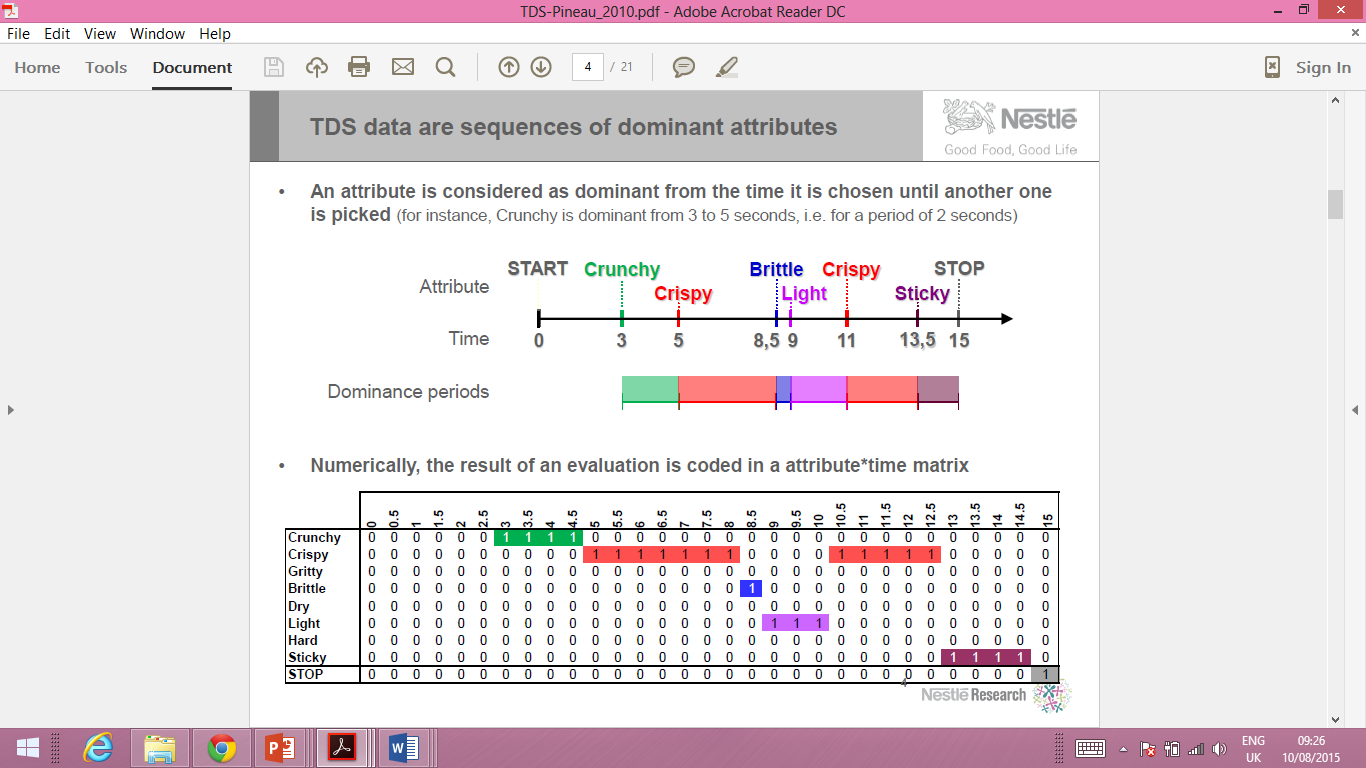
Tuliskan overview P/L, dalam bentuk gambar dan narasi yang dapat memberikan gambaran tentang aplikasi dan konteksnya, yaitu hubungannya dengan dunia luar. Pada bagian ini dapat dimasukkan juga gambar yang memberikan bentuk dari system yang dibuat, ataupun dapat berupa diagram konteks (seperti pada sub bab 3.2.1.)

Aplikasi <nama sistem> akan digunakan untuk menfasilitasi uji sensory metode temporal dominance sensation terhadap makanan oleh seorang panelis. Aplikasi akan dioperasikan oleh panelis. Panelis pertama akan mulai melakukan pengunyahan terhadap makanan. Aplikasi akan menampilkan list atribut sensory di layar.



Gambar 1. Contoh antarmuka program saat pengujian (sumber: https://deakincass.wordpress.com/2014/02/18/introducing-temporal-dominance-of-sensation/)

Panelis akan mendeteksi sensasi dominan dari makanan selama pengunyahan dan menekan tombol pada aplikasi sesuai dengan sensasi tersebut. Aplikasi akan merekam data masukkan panelis dan akan mengeluarkan file yang berisi rekaman sensasi terhadap waktu untuk setiap panelis dan setiap pengulangan.



Gambar 2. Visualisasi data yang akan dihasilkan (sumber: Pineau et al (2010)

## Karakteristik Pengguna

Minimal sebuah tabel dengan Kolom : Pengguna, Pekerjaan, Hak Akses. Kolom Hak Akses dihubungkan dengan Fungsi utama yang muncul pada Fungsi Produk

| **Kategori Pengguna** | **Tugas** | **Hak Akses ke aplikasi** |
| --- | --- | --- |
| Panelis | Melakukan uji |  |
|  |  |  |

## Batasan

* Harus bisa berjalan diatas sistem operasi Windows
* data yang dikeluarkan harus dengan mudah dimasukkan ke aplikasi Microsoft Excel

## Lingkungan Operasi

Berjalan di komputer klien yang menggunakan sistem operasi Windows

# Deskripsi Kebutuhan

## Kebutuhan Antarmuka Eksternal

Data yang dikeluarkan dapat dengan mudah dimasukkan ke microsoft Excel

### Antarmuka pemakai

Pemakai mengoperasikkan perangkat lunak menggunakan keyboard, mouse, dan monitor.

### Antarmuka Perangkat Lunak

Perangkat lunak menggunakan API Windows.

Data yang dihasilkan harus dapat dengan mudah dibaca oleh Microsoft Excel

## Kebutuhan Fungsional

Diawali dengan membuat daftar kebutuhan fungsional P/L, lengkap dengan ID (Identitas) atau kode dan penjelasan jika perlu. Bisa dibuat dalam bentuk tabel.

| **ID** | **Kebutuhan** | **Penjelasan** |
| --- | --- | --- |
| FR-01 | Nama Panelis dan Nomor Ulangan | Perangkat lunak akan meminta dan menyimpan nama panelis dan nomor ulangan uji. Nama panelis dan nomor ulangan akan disimpan pada data pengeluarana |
| FR-02 | Tombol Start | Perangkat lunak memiliki tombol start untuk mulai melakukan uji dan mencatat data yang dimasukkan |
| FR-03 | List Atribut Sensory | Perangkat lunak akan menampilkan list atribut sensory yang bisa dipilih selama keberlangsungan uji |
| FR-04 | Edit List Atribut Sensory | Pemakai bisa menambah dan megurangi list atribut sensory yang akan digunakan dalam uji. Penambahan dan pengurangan ini dilakukan di luar uji. Maksimum atribut yang bisa ditambahkan adalah 15. |
| FR-05 | Tombol Stop | Perangkat lunak memiliki tombol stop untuk meghentikan uji. |
| FR-06 | Data Record | Perangkat lunak akan menyimpan sebuah data rekaman uji per panelis per ulangan ke dalam suatu folder. Data yang disimpan berupa matrix atribut\*waktu sesuai **gambar 2** dengan interval waktu 0.5 detik. Data rekaman dapat dengan mudah dimasukkan ke dalam aplikasi Microsoft Excel untuk diolah lebih lanjut. Data juga menyimpan nama panelis dan nomor ulangan. |
|  | Stopwatch? |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## Kebutuhan Non Fungsional

Uraikan dengan ringkas kebutuhan non fungsional dalam tabel sebagai berikut. Isilah Kolom Kebutuhan dengan kalimat yang jelas dan kelak dapat ditest untuk dipenuhi.ID adalah nomor kebutuhan yang harus ditelusuri pada saat test. Tuliskan N/A bila Not Applicable..

| **ID** | **Parameter** | **Kebutuhan** |
| --- | --- | --- |
| NR-01 | Portability | Perangkat lunak bisa dengan mudah diedarkan ke beberapa komputer dengan mudah |
| NR-02 | Easy To Use | Perangkat lunak harus bisa digunakan oleh pengguna yang hanya menerima sedikit pelatihan |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Catatan :

*Availability : ketersediaan aplikasi, misalnya harus terus menerus beroperasi 7 hari perminggu, 24 jam per haritanpa gagal*

*Reliability : keandalan, misalnya tidak pernah boleh gagal(atau kegagalan yang ditolerir adalah …%) sehingga harus dipikirkan fault tolerant architecture. Biasanya hanya perlu untuk Critical Application yang jika gagal akan berakibat fatal.*

*Ergonomy : kenyamanan pakai bagi pengguna*

*Portability : kemudahan untuk dibawa dan dioperasikan ke mesin/sistem operasi/platform yang lain*

*Memory : jika perhitungan kapasitas memori internal kritis (misalnya untuk SW yang harus dijadikan CHIPS dan ukurannya harus kecil*

*Response time : Batasan waktu yang harus dipenuhi. Sangat penting untuk aplikasi Real Time. Contoh: “Aaplikasi harus mampu menampilkan hasil dalam 4 detik”, atau “ATM harus menarik kembali kartu yang tidak diambil dalam waktu 3 menit”*

*Safety: yang menyangkut keselamatan manusia, misalnya untuk SW yang dipakai pada sistem kontrol di pabrik*

*Security : aspek keamanan yang harus dipenuhi.*

## Model Analisis

### Diagram Use Case

Bagian ini diisi dengan diagram use case keseluruhan.

Berikan gambar use case, beserta aktornya, juga penjelasan secukupnya untuk diagram use case ini.

|  |
| --- |
| Gambar 3 Contoh Use Case |

#### Definisi Actor

*Bagian ini diisi dengan daftar actor dan deskripsi role untuk actor tersebut. Deskripsi role harus menjelaskan wewenang pada role tersebut dalam perangkat lunak. Bisa dibuat dalam bentuk tabel berikut*:

|  |  |  |
| --- | --- | --- |
| *No* | *Actor* | *Deskripsi* |
| *1* | *Guest* | *Actor dengan role ini mempunyai wewenang untuk melakukan registrasi serta melihat informasi-informasi yang sifatnya umum seperti profil perusahaan, ….* |

#### Definisi Use Case 1

Bagian ini diisi dengan skenario (*flow of event*) untuk beberapa use case 1, yang menggambarkan urutan interaksi actor dengan use case tersebut, dari awal sampai akhir.

*Contoh:*

*Nama Use Case: Melihat daftar produk*

*:*

|  |  |
| --- | --- |
| ***Aksi Actor*** | ***Reaksi Sistem*** |
| ***Skenario Normal*** |  |
| *1. Memilih menu Daftar Produk* |  |
|  | *2. Menampilkan daftar produk dari basisdata ke*  *layar* |
| *3. Menekan tombol navigasi (next, prev)* |  |
|  | *4. Me-refresh tampilan daftar produk* |
| ***Skenario Alternatif*** |  |
| *1. Memilih menu Daftar Produk* |  |
|  | *2. Menampilkan pesan ‘Tidak ada produk’* |

#### Definisi Use Case 2

*Isikan sesuai dengan 3.4.1.2*

#### Definisi Use Case 3

*Isikan sesuai dengan 3.4.1.2, dan seterusnyahingga seluruh use case dijelaskan.*

### Diagram Kelas

*Bagian ini diisi dengan daftar kelas analisis sebagai pendukung model use case. Setiap usecase seharusnya memiliki skenario yang implementasinya dapat ditemukan pada suatu kelas (bagian metode/operasi).*

*Gmbarkan diagram kelas dan keterhubungannya pada bagian ini.*

|  |
| --- |
| *http://www.agiledata.org/images/oo101ClassDiagram.gif*  Gambar 4 Contoh Diagram Kelas |

#### Diagram Kelas 1

Definisi atribut, dan metode

Jelaskan jika ada hubungan dengan kelas lain (apa hubungannya). Jelaskan dalam bentuk tabel seperti pada contoh

Contoh (diambil dari gambar 2):

Atribut:

NIM : nomor mahasiswa

Nama: Nama mahasiswa

Alamat: alamat mahasiswa

Operasi:

IsEligible: memeriksa apakah mahasiswa berhak...

GetSeminarsTaken: menampilkan daftar seminar yang sudah diambil

Dst.

Kelas yang terhubung:

|  |  |
| --- | --- |
| Nama Kelas | Deskripsi |
| SeminarEnrollment | Mahasiswa terdaftar pada suatu kelas SeminarEnrollment |
| Seminar | ... |
| Address | ... |

#### Diagram Kelas 2

Definisi atribut, dan metode, seperti pada 3.4.2.1

#### Diagram Kelas 3

Definisi atribut, dan metode, seperti pada 3.4.2.1 dan seterusnya sejumlah kelas yang didefinisikan.

### Diagram Sequence

*Diagram Sekuens digunakan untuk menunjukkan bagaimana antar proses berinteraksi, pada diagram ini urutan juga akan tergambarkan pada diagram ini. Diagram ini sering digunakan untuk menjelaskan skenario dari suatu use case. Pada diagram ini, suatu skenario dapat menggambarkan objek-objek yang saling berinteraksi.*

# Kerunutan (traceability)

Diisi dengan tabel yang berisi traceability dari hasil analisis. Gunanya untuk menilai apakah hasil analisis “runut” dan lojik. Untuik sementara, baru didefinisikan Data-store versus E-R.

## Kebutuhan Fungsional vs Use Case

Mapping kebutuhan fungsional dengan use case terkait

|  |  |
| --- | --- |
| **ID Kebutuhan Fungsional** | **ID Use Case Terkait** |
|  |  |
|  |  |
|  |  |

## Kebutuhan Fungsional vs Diagram Kelas

Mapping kebutuhan fungsional dengan kelas terkait

|  |  |
| --- | --- |
| **Id kebutuhan fungsional** | **ID Kelas (Nama kelas)** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |