*Software Requirements Specification*

for

Aplikasi Form Online

Oleh:

171022000045 – Jati Prasetyo (Github : jattiblee)

171022000062 – Juri Pebrianto (Github : aquarink)

171022000044 – Riky Susanto (Github : riky-cloud)

Table of Contents

1. Pendahuluan 1

1.1 Tujuan Penulisan Dokumen 1

1.2 Audien yang Dituju dan Pembaca yang Disarankan 1

1.3 Batasan Produk 1

1.4 Definisi dan Istilah 1

1.5 Refrensi 1

2. Deskripsi Keseluruhan 2

2.1 Deskripsi Produk 2

2.2 Fungsi Produk 2

2.3 Penggolongan Karakterik Pengguna 2

2.4 Lingkungan Operasi 2

2.5 Batasan Desain dan Implementasi 2

2.6 Dokumentasi Pengguna 3

3. Kebutuhan Antarmuka Eksternal 4

3.1 User Interfaces 4

3.2 Hardware Interface 4

3.3 Software Interface 4

3.4 Communication Interface 4

4. Functional Requirement 5

4.1 Use Case Diagram 5

4.2 Nama Use Case 1 5

4.3 Nama Use Case 2 5

4.4 Class Diagram 6

5. Non Functional Requirements 7

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Pendahuluan

## Tujuan Penulisan Dokumen

Tujuan dari dokumen Software Requirement Specification ini adalah untuk memberikan gambaran yang spesifik mengenai kebutuhan software dari Aplikasi Form Online. Spesifikasi tersebut termasuk dari segi perangkat lunak dan perangkat keras, untuk memberikan gambaran dan penjelasan yang dibutuhkan untuk produk termasuk kebutuhan fungsional dan non fungsional, dan kebutuhan antar muka, mulai dari antar muka pengguna hingga antar muka komunikasi.

## Audien yang Dituju dan Pembaca yang Disarankan

<Jelaskan berbagai jenis pembaca bahwa dokumen ini ditujukan untuk, seperti pengembang, manajer proyek, staf pemasaran, pengguna, penguji, dan lainnya>

## Batasan Produk

Semua hal yang tercantum di dalam dokumen ini merupakan bagian dari ruang lingkup kebutuhan pembangunan perngkat lunak yang berupa aplikasi berbasis web yang digunakan untuk membuat form secara online, menjawab pertanyaan yang telah dibuat oleh user lain, dan melihat hasil jawaban yang telah dijawab oleh user lain.

## Definisi dan Istilah

<tulis istilah dan definisikan jika ada>

* SRS : *Software Requirements Specification*, atau

Spesifikasi Kebutuhan Perangkat Lunak (SKPL)

* IEEE : *Institute of Electrical and Electronics Engineering*

Standar internasional untuk pengembangan dan perancangan produk.

## Refrensi

IEEE Std. 830-1998, IEEE Recommended Practice for Software Requirement Specifications.

# Deskripsi Keseluruhan

## Deskripsi Produk

Aplikasi Form Online ini merupakan perankat lunak yang dapat digunakan untuk membuat pertanyaan – pertanyaan agar dapat dijawab atau direspon oleh user lain secara online. Pada aplikasi ini, user yang telah membuat pertanyaan juga dapat membagikan link kepada user lain agar user lain dapat menjawab pertanyaan yang telah dibuat. Setelah user lain menjawab pertanyaan tersebut, user pemberi pertanyaan dapat melihat jawaban yang diinput oleh user lain. Aplikasi ini dapat digunakan untuk membuat Survey Online, Kuis Online, dan bisa digunakan untuk Ujian Online.

## Fungsi Produk

Produk ini memiliki beberapa fungsi dalam kebutuhan User antara lain:

* Login User
* Register User
* Buat pertanyaan
* Preview pertanyaan
* Melihat Respon (jawaban)
* Membagikan link pertanyaan
* Menjawab pertanyaan

## Penggolongan Karakterik Pengguna

Adapun pengguna dalam sistem ini adalah sebagai berikut:

* User 1 (Pembuat pertanyaan)
* User 2 (Pemberi jawaban)

Tabel berikut menggambarkan karakteristik umum pengguna yang akan mempengaruhi fungsionalitas dari produk perangkat lunak.

Tabel 1 Karakteristik Pengguna

|  |  |  |
| --- | --- | --- |
| Kategori Pengguna | Fasilitas | Hak akses ke Aplikasi |
| User | * Login/logout * Register * Membuat pertanyaan * Melihat respon * Membagikan *link* pertanyaan * Menjawab pertanyaan | * Hak akses ke form login * Hak akses ke form register * Hak akses ke menu membuat pertanyaan * Hak akses melihat respon User 2 |

## Lingkungan Operasi

<Jelaskan lingkungan di mana perangkat lunak akan beroperasi, termasuk platform, perangkat keras, sistem operasi dan versi, dan komponen perangkat lunak lain atau aplikasi yang berdampingan>

Aplikasi form online ini adalah aplikasi yang berjalan secara stand alone. Aplikasi ini tidak ada admin karena setiap user dapat membuat pertanyaan untuk dibagikan.

## Batasan Desain dan Implementasi

<Jelaskan setiap item atau masalah yang akan membatasi pilihan yang tersedia untuk para pengembang / developer. Ini mungkin termasuk: kebijakan perusahaan atau peraturan; keterbatasan hardware (persyaratan memori); teknologi tertentu, alat, dan database yang akan digunakan; persyaratan bahasa; protokol komunikasi; pertimbangan keamanan; atau standar pemrograman>

Batasan aplikasi Form Online dalam konteks SRS ini adalah sebagai berikut:

1. Aplikasi ini akan dibangun menggunakan PHP dan Mysql berbasis web
2. Database yang digunakan adalah MySQL versi 10.1.30-MariaDB
3. Aplikasi ini dapat digunakan oleh semua user yang terhubung koneksi internet.

Implementasi aplikasi Form Online dalam konteks SRS ini adalah sebagai berikut:

1. Aplikasi ini di bangun untuk memudahkan user dalam membuat survey online
2. Menampilkan laporan pengguna yang telah mengikuti form online

# Kebutuhan Antarmuka Eksternal

## User Interfaces

Aplikasi ini menggunakan antarmuka berbasis web. Rancangan antarmuka ini bertujuan untuk memperjelas mengenai program pada sistem ini.

* + 1. Login
    2. Register
    3. Dashboard
    4. Halaman Tambah Pertanyaan
    5. Halaman Preview Pertanyaan
    6. Halaman Lihat Respon
    7. Halaman Menjawab Pertanyaan

## Hardware Interface

|  |  |
| --- | --- |
| **Hardware** | **Function** |
| Monitor | Menampilkan tampilan berupa CMS untuk Form online |
| Mouse | Alat bantu intuk pointing menu di CMS |

## Software Interface

|  |  |
| --- | --- |
| **Jenis Software** | **Kebutuhan Hardware** |
| Sistem Operasi | Windows 7 |
| Bahasa Pemrograman | PHP, Javascript |
| Pengolah Database | PhpMyadmin |
| Framework | Codeigniter |
| Pemodelan Sistem | UML |
| Perancangan Database | Draw.io |
| Perancangan Antar Muka | Draw.io |
|  |  |

## Communication Interface

Desain antar muka yang digunakan oleh aplikasi web ini menggunakan Elephant Template. Aplikasi ini dibangun dan dioperasikan secara *live* pada server development, dalam pengembangan ini jika sudah memasuki tahap Setelah proses devopment selesai apliasi dapat di akses oleh pengguna menggunakan koneksi via browser.

# Functional Requirement

* 1. Login
     1. Description and Priority

Pada form login inputan untuk dapat masuk ke dalam aplikasi adalah username dan password yang sudah terdaftar di dalam database. Dari fungsi login ini maka sistem akan memeriksa valid tidaknya data yang dimasukkan dalam form login lalu sistem menampilkan halaman dashboard.

* + 1. Stimulus / Response Sequence

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| 1. User memasukan email dan password |  |
|  | 1. Memeriksa valid tidaknya data masukan dengan memeriksa ke table user. |
|  | 1. Masuk ke sistem sesuai dengan profile yang terdaftar |

* + 1. Functional Requirement

REQ 1 : Jika input yang dimasukan salah atau tidak terdaftar di database maka sistem akan menampilkan pesan kesalahan dan kembali ke menu form login.

* 1. Register
     1. Description and Priority

User harus melakukan register jika belum memiliki akun agar dapat mengakses aplikasi ini.

* + 1. Stimulus / Response Sequence

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| 1. User mendaftarkan username |  |
|  | 1. Sistem memeriksa apakah username sudah terdaftar atau belum |
| 1. Jika username sudah ada, maka kembali ke form register. Jika belum ada maka user masuk ke form login |  |

* + 1. Functional Requirement

REQ 1 : Jika input username sudah ada di database maka sistem akan menampilkan pesan bahwa username tersebut sudah ada dan akan dikembalikan ke halaman register untuk diinput username yang lainnya.

* 1. Halaman Buat Pertanyaan
     1. Description and Priority

User dapat membuat pertanyaan dengan input judul pertanyaan terlebih dahulu, lalu klik tombol tambah pertanyaan dan kemudian dapat membuat bermacam – macam pertanyaan.

* + 1. Stimulus / Response Sequence

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| 1. User input judul pertanyaan |  |
| 1. User Klik tombol Tambah Pertanyaan |  |
|  | 1. Aplikasi akan menampilkan window pop up untuk user agar dapat input pertanyaan dan pilihan – pilihannya. |
| 1. User klik Simpan Pertanyaan |  |
|  | 1. Aplikasi akan menyimpan pertanyaan ke database. |

* + 1. Functional Requirement

REQ 1 : Jika User tidak input judul pertanyaan, maka user tidak dapat klik tombol tambah pertanyaan.

REQ 2 : Jika User input pilihan hanya 1, maka user tidak dapat melanjutkan untuk menambahkan jumlah pertanyaan.

* 1. Halaman preview pertanyaan
     1. Description and Priority

Setelah user selesai membuat pertanyaan, user dapat melihat pertanyaan yang telah dibuat, dan pada halaman preview pertanyaan ini, user juga dapat mendapatkan *link* untuk dibagikan kepada user lain agar pertanyaan yang dapat dibuat bisa dijawab oleh user lain.

* + 1. Stimulus / Response Sequence

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| 1. User klik tombol Preview Pertanyaan |  |
|  | 1. Aplikasi akan mengarahkan ke halaman preview pertanyaan dan men-generate *link* |

* 1. Halaman lihat respon
     1. Description dan Priority

User dapat melihat respon jawaban dari user lain yang telah selesai menjawab pertanyaan pada halaman ini. User dapat melihat respon secara individual dan Summary. Individual adalah jawaban user secara satu persatu, sedangkan summary adalah jawaban user yang telah dirangkum dan disediakan perhitungannya.

* + 1. Stimulus / Response Sequence

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| 1. User klik tombol lihat response |  |
|  | 1. Aplikasi akan mengarahkan ke halaman Response dan menampilkan data hasil jawaban yang telah dijawab oleh user lain. |

* 1. Halaman menjawab pertanyaan
     1. Description and Priority

Selain dapat membuat pertanyaan, user juga dapat menjawab pertanyaan yang telah dibuat oleh user lain. Dengan catatan, user yang akan menjawab pertanyaan harus memiliki *link* pertanyaan tersebut.

* + 1. Stimulus / Response Sequence

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| 1. User klik *link* yang telah dibagikan oleh user lain |  |
|  | 1. Aplikasi akan mengarahkan ke halaman pertanyaan tersebut |

<Tulis Kebutuhan Fungsional / Functional Requirement disini>

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

| **ID** | **Kebutuhan Fungsional** | **Penjelasan** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## E:\Downloads\Untitled Diagram.jpgUse Case Diagram

## Nama Use Case 1

4.1.1 Deskripsi Use Case

<desripsikan / jabarkan mengenai use case ini >

4.1.2 Stimulus and Respon

<menyediakan daftar aksi yang dilakukan oleh user dan respon dari sistem.>

|  |  |
| --- | --- |
| Action by user | Response from system |
| 1 |  |
|  | 2 |
| 3 |  |
|  | 4 .. |

4.1.4 *Activity Diagram*

## Nama Use Case 2

<Sama seperti di atas, dan seterusnya sesuai jumlah use case yang didapatkan>

## Class Diagram

<*identifikasi kelas yang terkait dan hubungannya pada sistem yang dikembangkan*>

# Non Functional Requirements

<*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** |
| --- | --- | --- |
|  | Availability |  |
|  | Reliability |  |
|  | Ergonomy |  |
|  | Portability |  |
|  | Memory |  |
|  | Response time |  |
|  | Safety | N/A |
|  | Security |  |
|  |  |  |
|  | Others 1: Bahasa komunikasi | Misalnya : semua tanya jawab harus dalam bahasa Indonesia |
|  |  | Setiap layar harus mengandung logo PT Pos Indonesia |
|  |  |  |

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*