
Software Requirements Specification

for

Aplikasi Sistem Kepegawaian Mekari Talenta

Version 1.0 approved

Prepared by

5025211014 - Weynard

5025211008 - Razan

5025211121 - Yoyo

15 Maret 2023

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

1. Pendahuluan

1.1 Tujuan Penulisan Dokumen

<Tulis tujuan penulisan dokumen disini.>

1.2 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>

1.3 Batasan Produk

<Berikan penjelasan singkat dari perangkat lunak yang ditentukan dan tujuannya, termasuk manfaat yang relevan, tujuan, dan sasaran. Hubungkan perangkat lunak untuk tujuan perusahaan atau strategi bisnis>

1.4 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.

1.5 Refrensi

<Tulis daftar dokumen lain atau alamat web yang menjadi acuan SRS ini.>

2. Deskripsi Keseluruhan

2.1 Deskripsi Produk

<Deskripsikan produk/sistem yang akan dibuat disini.>

2.2 Fungsi Produk

<Berisi hanya rangkuman fungsi utama produk, produk harus melakukan apa atau memungkinkan pengguna melakukan apa. Hanya ringkasan tingkat tinggi (seperti bullet list) yang dibutuhkan di sini.>

2.3 Penggolongan Karakterik Pengguna

<Identifikasi berbagai golongan pengguna yang terkait dengan produk yang dikembangkan>

2.4 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>

2.5 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>

2.6 Dokumentasi Pengguna

<Daftar komponen dokumentasi pengguna (seperti user manual, on-line help, dan tutorial) yang akan disampaikan bersama dengan perangkat lunak yang akan dikirim>

Contoh Dokumentasi Program:

```
/*  
Programmer: Weynard  
Last Updated: 15 Maret 2023  
Description: Program untuk menghitung gaji karyawan pada sistem kepegawaian Mekari Talenta  
*/  
#include <iostream>
```

```
using namespace std;
```

```
int main() {  
    int gajiPokok, tunjangan, bonus, gajiTotal;  
    string namaKaryawan;  
    char jenisKelamin;
```

```
    // Input data karyawan  
    cout << "Masukkan nama karyawan: ";  
    getline(cin, namaKaryawan);  
    cout << "Masukkan jenis kelamin (L/P): ";  
    cin >> jenisKelamin;  
    cout << "Masukkan gaji pokok: ";  
    cin >> gajiPokok;  
    cout << "Masukkan tunjangan: ";  
    cin >> tunjangan;
```

```
    // Hitung bonus berdasarkan jenis kelamin  
    if (jenisKelamin == 'L') {  
        bonus = 500000;  
    } else {  
        bonus = 1000000;  
    }
```

```
    // Hitung gaji total  
    gajiTotal = gajiPokok + tunjangan + bonus;
```

```
    // Output hasil perhitungan  
    cout << endl << "Gaji " << namaKaryawan << " adalah " << gajiTotal << endl;
```

```
    return 0;  
}
```

3. Kebutuhan Antarmuka Eksternal

3.1 User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

3.2 Hardware Interface

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

3.3 Software Interface

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

3.4 Communication Interface

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

4. Functional Requirement

<Area ini menggambarkan pengorganisasian persyaratan fungsional untuk produk dengan fitur sistem, layanan utama yang disediakan oleh produk>

<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

4.1 Use Case Diagram

<Gambar use case diagramnya dari functional requirement yang didapatkan>

4.2 Nama Use Case 1

4.1.1 Deskripsi Use Case

<deskripsikan / 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

4.3 Nama Use Case 2

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

4.4 Class Diagram

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

5. Non Functional Requirements

<Uraikan dengan ringkas kebutuhan non fungsional dalam tabel sebagai berikut. Isilah Kolom Kebutuhan dengan kalimat yang jelas dan kelak dapat dites 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: “Aplikasi 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