System Requirement Specification

<nama proyek / kasus>

<daftar nama dan nrp mhs>

ADSI - <sebutkan kelas apa>

Dokumen *System Requirement Specification* (SRS) digunakan sebagai rujukan oleh setiap pihak yang terlobat dalam pengembangan produk SRS ini dibuat berdasarkan kebutuhan user yang diperoleh dari hasil riset ataupun survei. Dalam dokumen ini terdapat 8 bagian utama, yang terdiri sebagai berikut:

1. **Pendahuluan**

Bagian ini merupakan rangkuman dari hasil *research* oleh tim *researcher,* tersusun atas 5 sub bagian, antara lain:

1. **Latar Belakang**

Penjelasan singkat obyek permasalahan (bisa disertai sumber eksternal), dan penjelasan WHY serta PAIN dari subyek yang menjadi fokus permasalahan yang berupa ringkasan dari hasil *research* dan survei. **Tujuan**

Menjelaskan pengembangan modul atau produk yang akan dihasilkan, khususnya untuk menjawab WHY dan PAIN dari obyek permasalahan yang berupa ringkasan dari hasil *research* dan survei. Dijelaskan business value dari sistem yang akan dibuat.

1. **Ruang Lingkup**

Isian berupa ringkasan hasil *research* dan survei serta analisis batasan pengembangan aplikasi.

1. **Studi Kelayakan**

Bagian ini merupakan ringkasan dari hasil *research* dan survei yang berisikan hasil analisa kelayakan terkait modul atau aplikasi yang akan dikembangkan, apakah sudah sesuai dengan calon.

1. **Adoption Strategi**

Berisikan penjelasan singkat terkait bagaimana kedepan modul atau aplikasi yang dibangun dapat dimanfaatkan dengan baik.

1. **Definisi Aktor**

Jenis pengguna yang akan menjadi pengguna sistem. Berupa tabel yang berisikan informasi kode aktor, nama aktor, dan juga deskripsi peran di dalam sistem. Buatlah standar untuk kode (misal: menggunakan singkatan, huruf kapital dan tanda \_ untuk penghubung antar kata). Kode aktor ini akan digunakan lagi di bagian lainnya. contohnya:

| **Aktor** | | **Deskripsi** |
| --- | --- | --- |
| **Kode** | **Nama** |
| MHS | Mahasiswa | Mahasiswa yang mengakses hasil CPMK dan CPL |
| DSN | Dosen | Dosen yang memasukkan nilai asesmen |
| DSNK | Dosen koordinator MK | Dosen koordinator untuk MK untuk mengatur RPS dan soal asesmen |
| KAPRODI | Pimpinan Prodi/ Ketua Program Studi | Pimpinan Program Studi yang memasukkan CPL serta mapping CPL ke MK |
| DSNM | Dosen wali | Dosen wali seorang mahasiswa untuk  mellihat perolehan CPL dari anak walinya |
| DSNV | Dosen validator | Dosen / tim mutu yang mevalidasi RPS |

Penggunaan kode di sini ditujukan untuk konsistensi pada level dokumentasi sampai pada implementasi, harapannya semua pihak memiliki persepsi dan pengertian yang sama terkait siapa saja yang akan memiliki akses pada sistem.

1. **Kebutuhan Fungsional**

Bagian ini diisi dengan User Story (3C) yaitu

* 1. Kebutuhan Fungsional (Functional Requirement / FR). Kebutuhan Fungsional dituliskan dengan format seperti contoh di bawah ini. Tersusun atas ID, ID Use case, Fungsi, Aktor, dan Deskripsi. Gunakan format pengkodean untuk ID dari daftar fungsional (misal: menggunakan rumus F\_<Nama Modul / Aplikasi>.<2 digit numbering>, contoh : F\_OBE.01).

Keterangan: fungsi adalah kesatuan aktivitas yang dilakukan oleh seorang aktor (mencerminkan hak akses / CRUD)

Contoh :

| **ID** | **ID Use Case** | **Fungsi** | **Aktor** | **Deskripsi** |
| --- | --- | --- | --- | --- |
| F\_OBE.01 | UC\_OBE.01,  UC\_OBE.02,  UC\_OBE.03,  UC\_OBE.04 | Manajemen CPL | KAPRODI | Menyimpan data CPL tiap prodi, untuk tiap kurikulum tiap prodi bisa mempunyai daftar CPL yang berbeda. Catatan: tiap prodi bisa mempunyai banyak kurikulum (misal: kurikulum 2020, kurikulum 2024) |
| F\_OBE.02 | UC\_OBE.05,  UC\_OBE.06,  UC\_OBE.07,  UC\_OBE.08 | Manajemen performa CPL | KAPRODI | Menyimpan pengelompokkan performa CPL misal baik, sedang,rendah |

1. **Diagram UML**

Bagian ini terdiri 4 bagian utama, yaitu :

1. **Use Case**

Bisa ditampilkan dalam bentuk diagram use case atau dengan menggunakan tabel seperti contoh agar mudah dilihat. Bagian ini digunakan untuk pemetaan aktor dan fungsional sistem. Digunakan untuk mempermudah programmer dan juga UI designer memahami irisan-irisan bagian dari sistem yang akan dibangun, sehingga proses development menjadi efisien. Terdiri atas diagram use case dan penjelasan singkat. Buat format pengkodean untuk ID dari tiap usecase (misal: menggunakan rumus UC\_<Nama Modul / Aplikasi>.<2 digit numbering>, contoh : UC\_OBE.01)

Contoh :

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID Usecase** | **Usecase** | **Actors** | | | | | |
| **KAPRODI** | **DSNK** | **DSNV** | **DSN** | **DSNW** | **MHS** |
| UC\_OBE.01 | Tambah CPL | v |  |  |  |  |  |
| UC\_OBE.02 | Ubah CPL | v |  |  |  |  |  |
| UC\_OBE.03 | Hapus CPL | v |  |  |  |  |  |
| UC\_OBE.04 | Lihat CPL |  | v |  | v |  | v |
| UC\_OBE.05 | Tambah Performa CPL | v |  |  |  |  |  |
| UC\_OBE.06 | Ubah Performa CPL | v |  |  |  |  |  |
| UC\_OBE.07 | Hapus Performa CPL | v |  |  |  |  |  |

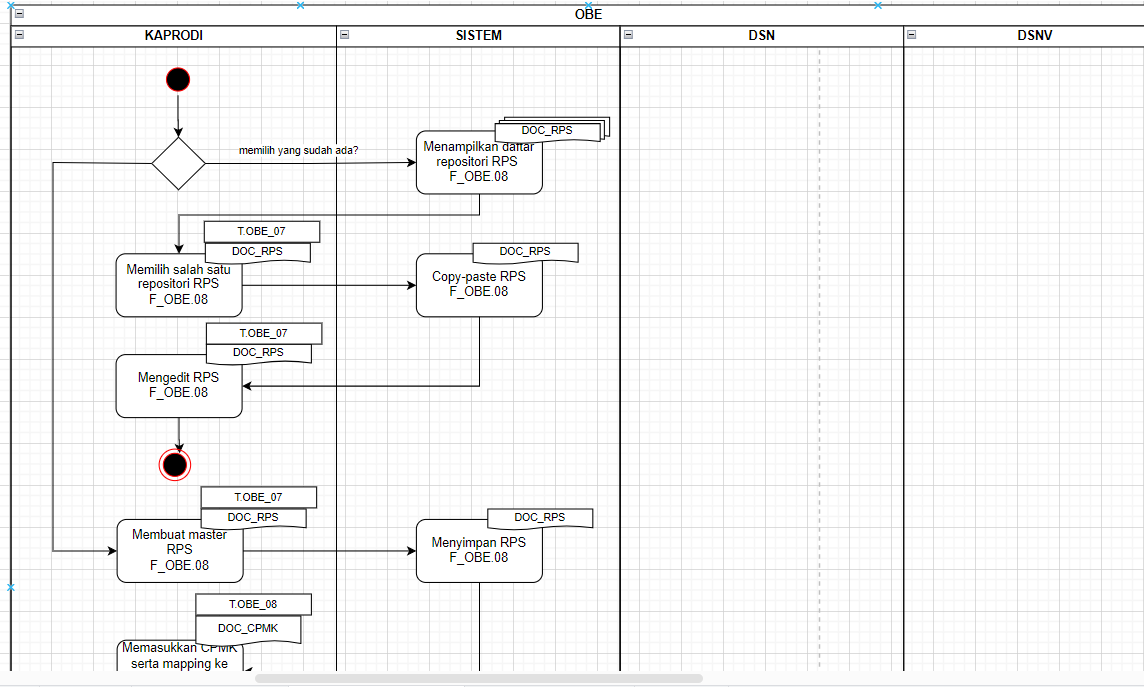
Setelah diagram use case, cantumkan use case form/ UC Specification.

1. **Activity Diagram**

Activity diagram yang menggambarkan interaksi antara actor dengan sistem dalam melakukan pengolahan sumber daya dokumen. Posisi horizontal dari atas ke bawah menunjukkan runutan pelaksanaan proses berdasar urutan waktu pengerjaan.

Dalam prakteknya kebutuhan pada *activity diagram* ini dapat dipecah menjadi lebih dari 1 bagian, misalnya : *Activity Master*, dan *Activity Proses.*

Gunakan format pengkodean untuk ID dari activity diagram (misal menggunakan rumus A\_<Nama Modul / Aplikasi>.<2 digit numbering>, contoh: A\_OBE.01).



1. **Dokumen**

Dokumen (sekumpulan data) yang nantinya akan mengalir untuk diproses oleh sistem. Bisa berupa dokumen utama ataupun dokumen pendukung yang bergerak selama pemrosesan. Bedakan cara pengkodean dengan kode actor ggar pengkodean tidak tertukar, misal pengkodean menggunakan huruf kapital dan diawali dengan tulisan *“DOC\_”*. Contoh: DOC\_CPL, DOC\_MHS, dan lain-lain

Contoh:

1. ***DOC\_CPL***

| **Field** | **Nama** | **Validasi** | **Deskripsi** |
| --- | --- | --- | --- |
| id\_CPL | Id CPL | ID Unik |  |
| kode\_CPL | Kode CPL | ID Unik | Pengkodean bisa dikonfigurasi sesuai dengan aturan pengkodean CPL Univ masing-masing |
| deskripsi\_CPL\_Ind | Deskripsi CPL bahasa |  | Deskripsi dari CPL dalam bahasa indonesia |
| deskripsi\_CPL\_Ing | Deskripsi CPL inggris |  | Deskripsi dari CPL dalam bahasa inggris |
| id\_kurikulum | id\_kurikulum |  | Kurikulum dimana CPL ini berlaku |

1. ***Class diagram***

Bagian ini diisi dengan class diagram.

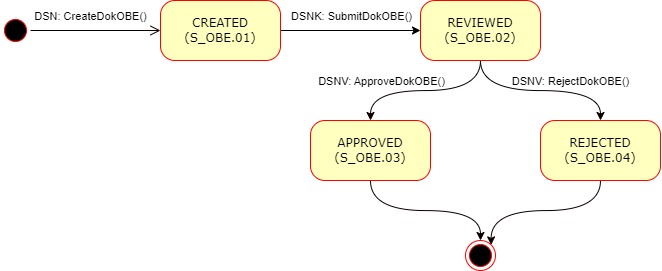
1. **Sequence Diagram**

Sequence diagram yang menggambarkan  interaksi antar objek dan menunjukkan (memberi tanda atau petunjuk) komunikasi diantara objek-objek tersebut.

Gunakan format pengkodean untuk ID dari daftar sequence diagram (misal menggunakan rumus SEQ\_<Nama Modul / Aplikasi>.<2 digit numbering>, contoh: SEQ\_OBE.01).

1. **State Diagram**

State diagram disesuaikan dengan jumlah dokumen sumber yang didefinisikan. Gunakan format pengkodean untuk ID dari daftar fungsional (misal menggunakan rumus S\_<Nama Modul / Aplikasi>.<2 digit numbering>, contoh: S\_OBE.01).



1. **Pemetaan Akses**

Tabel akses setiap *field* dari dokumen sumber untuk masing-masing aktor. Disusun dengan menggunakan tabel untuk setiap Dokumen yang didefinisikan di Bagian 3. Menggunakan kolom Field, Function (Create, Read, Update, Delete), dan juga daftar aktor.

Contoh :

1. ***DOC\_CPL***

## ****Resource D****OC\_CPL

| **Field** | **Function** | **KAPRODI** |
| --- | --- | --- |
| id\_CPL |  |  |
| kode\_CPL |  |  |
| deskripsi\_CPL\_Ind | create | V |
| read | V |
| update | V |
|  |  |  |
| deskripsi\_CPL\_Ing | create | V |
| read | V |
| update | V |
|  |  |  |
| id\_kurikulum | create | V |
| read | V |
| update | V |

1. **Design Antar-muka**

Gambaran antar-muka menggunakan sketsa dasar(*wireframe*), untuk menggambarkan kebutuhan antarmuka dan hubungan nya dengan masing-masing aktor dalam sistem.

Gunakan format pengkodean untuk ID dari daftar desain antar-muka (misal menggunakan rumus T\_<Nama Modul / Aplikasi>.<2 digit numbering>, contoh: T\_OBE.01).

1. **Acceptance Criteria**

Bagian ini merupakan bagian dari SRS nantinya yang akan dijadikan rujukan oleh tim *Quailty Assurance* (QA), dalam melakukan pengujian secara detail berdasarkan kasus-kasus uji yang akan disiapkan.

Daftar persyaratan dasar produk aplikasi dapat dinyatakan berjalan dengan baik. Gunakan format untuk kode uji, misal menggunakan rumus kode sebagai berikut: UJI\_<Nama Modul/App>.<Penomoran 2 digit>, contoh: UJI\_OBE.01.

Contoh :

| **Kode Uji** | **ID Use Case** | **Role** | **Menu/Sub Menu/Fitur** | **Deskripsi** | **Hasil Yang Diharapkan** |
| --- | --- | --- | --- | --- | --- |
| UJI\_OBE.01 | UC\_OBE.01 | KAPRODI | Tambah CPL | Pembuatan CPL dari prodi untuk kurikulum yang aktif. | CPL berhasil tersimpan dalam sistem |
| UJI\_OBE.02 | UC\_OBE.02 | KAPRODI | Ubah CPL | Perubahan terhadap deskripsi CPL yang telah dibuat sebelumnya. | Perubahan CPL berhasil tersimpan dalam sistem |
| UJI\_OBE.03 | UC\_OBE.03 | KAPRODI | Hapus CPL | Penghapusan terhadap CPL yang telah dibuat sebelumnya. | Penghapusan CPL berhasil dilakukan. Jika CPL sudah dihubungkan dengan indikator CPL dan MK maka tidak dapat dihapus. |
| UJI\_OBE.04 | UC\_OBE.04 | KAPRODI | Lihat CPL | Aktor melihat data CPL dari suatu kurikulum tertentu, default: kurikulum aktif | Tampilan daftar CPL |