**DPPL-01**

DESKRIPSI PERANCANGAN PERANGKAT LUNAK

LMS UDINUS

untuk:

UDINUS

Dipersiapkan oleh:

Andrian Octavianus

Program Studi Teknik Informatika/Sistem dan Teknologi Informasi

STEI - ITB

Jl. Ganesha 10, Bandung 40132

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Program Studi Teknik Informatika / Sistem dan Teknologi Informasi**  **STEI – ITB** | Nomor Dokumen | | Halaman |
| *DPPL-xx <xx:no grp>* | | *<#>/<jml #* |
| Revisi | *<nomor revisi>* | *Tgl: <isi tanggal>* |

DAFTAR PERUBAHAN

|  |  |
| --- | --- |
| Revisi | Deskripsi |
| A |  |
| B |  |
| C |  |
| D |  |
| E |  |
| F |  |
| G |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| INDEX  TGL | - | A | B | C | D | E | F | G |
| Ditulis oleh |  |  |  |  |  |  |  |  |
| Diperiksa oleh |  |  |  |  |  |  |  |  |
| Disetujui oleh |  |  |  |  |  |  |  |  |

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 dan Istilah 5

1.4 Aturan Penamaan dan Penomoran 5

1.5 Referensi 5

1.6 Ikhtisar Dokumen 5

2 Deskripsi Perancangan Global 6

2.1 Rancangan Lingkungan Implementasi 6

2.2 Deskripsi Data 6

2.2.1 Definisi Domain/Type 6

2.2.2 E-R Diagram 6

2.2.3 Daftar Tabel Basisdata 6

2.3 Deskripsi Modul 6

3 Deskripsi Perancangan Rinci 8

3.1 Deskripsi Rinci Tabel 8

3.1.1 Tabel <Nama..> 8

3.1.2 <Table 2 : t\_Detail …> 8

3.2 Deskripsi Rinci Modul 8

3.2.1 Modul <xxx> 8

3.2.1.1 Fungsi Modul 8

3.2.1.2 Spesifikasi Layar Utama 9

3.2.1.3 Spesifikasi Query (jika ada) 9

3.2.1.4 Spesifikasi Field Data Layar 9

3.2.1.5 Spesifikasi Objek-Objek pada Layar 9

3.2.1.6 Spesifikasi Proses/Algoritma 10

3.3 Matriks Kerunutan 10

Setelah Daftar Isi Boleh ada Daftar Tabel dan Daftar Gambar

# 1. Pendahuluan

## Tujuan Penulisan Dokumen

Dokumen ini merupakan kumpulan dokumentasi proyek pengembangan perangkat lunak LMS UDINUS yang dikembangkan oleh tim software house Daoen Moeda. LMS UDINUS ditujukan sebagai perangkat lunak yang mengelola materi pembelajaran dan tugas pada mata kuliah di UDINUS. Selain mengelola materi dan tugas pada mata kuliah di UDINUS juga LMS UDINUS mengelola mahasiswa dan dosen pada mata kuliah tertentu.

Dalam dokumen ini terdapat daftar kebutuhan yang telah disepakati antara pihak pengguna dan pihak pengembang dilengkapi dengan definisi dan penjelasannya. Dokumen ini juga berisi rekaman seluruh kegiatan pengembangan LMS UDINUS sejak analisis, perancangan, implementasi, hingga pengujian. Tujuan utama pembuatan dokumen ini ialah sebagai deliverables untuk UDINUS. Selain itu dokumen ini juga bertujuan sebagai acuan utama jika LMS UDINUS akan dikembangkan lebih lanjut.

## Lingkup Masalah

UDINUS, sebuah universitas ternama di Indonesia, membutuhkan adanya sistem informasi yang dapat mencakup manajemen dalam pengelolaan nilai, tugas dan materi perkuliahan yang sedang berlangsung. Solusi dari sisi teknologi informasi ini merupakan salah satu cara untuk mengoptimalkan proses perkuliahan yang ada di UDINUS. LMS UDINUS dibangun untuk mengelola materi, tugas, dan nilai pada mata kuliah yang ada di UDINUS. Pengelolaan materi membantu dosen untuk mengunggah materi yang ada pada mata kuliah tertentu dan mahasiswa dapat mengunduh materi tersebut. Pengelolaan tugas pada LMS UDINUS membantu dosen untuk memberikan tugas dan mengelola tugas yang akan diberikan kepada mahasiswa pada mata kuliah tertentu. Tugas yang terdapat pada LMS UDINUS juga dapat juga mencatat nilai mahasiswa yang mengerjakan tugas tesebut.

## Definisi dan Istilah

Semua definisi dan singkatan yang digunakan dalam dokumen ini dan penjelasannya

## Aturan Penamaan dan Penomoran

Table 1Aturan Penamaan dan Penomoran

|  |  |  |
| --- | --- | --- |
| Penomoran | Cara Penomoran | Deskripsi Penomoran |
| Kebutuhan Functional | SRS-F-XX | XX merupakan nomor pembeda kebutuhan functional |
| Kebutuhan Non-Functional | SRS-NF-XX | XX merupakan nomor pembeda kebutuhan non-functional |
| Definisi Actor | AC-XX | XX merupakan nomor pembeda actor |
| Definisi Use Case | UC-XX | XX merupakan nomor pembeda Use Case |
| Scenario Use Case | SC-XX-YY | XX merupakan nomor urut Use Case  YY merupakan nomor pembeda Scenario Use Case |
| Antarmuka Use Case | INT-XX | XX merupakan nomor pembeda antarmuka Use Case |
| Sequence Diagram Use Case | SD-XX-YY | XX merupakan nomor urut Use Case  YY merupakan nomor urut pembeda Sequence Diagram Use Case |

## Referensi

Building Web Applications with UML Second Edition. Jim Conallen. Addison Wesley. 2002

## Ikhtisar Dokumen

LMS UDINUS merupakan perangkat lunak berbasis web. LMS UDINUS digunakan untuk mengelola data mata kuliah dan data nilai yang digunakan untuk proses perkuliahan di UDINUS. LMS UDINUS diharapakan dapat meningkatkan efisiensi perkuliahan yang ada di UDINUS. Efisiensi tersebut bertujuan untuk mengurangi jarak antara mahasiswa dan dosen dalam melakukan proses perkuliahan. Proses perkuliahan dapat dilakukan menggunakan LMS UDINUS dengan menekankan materi dan tugas yang ada pada matakuliah tersebut. Dosen dapat memberikan materi yang disimpan di server dan mahasiswa dapat mengunduh materi tersebut melalui LMS UDINUS. Selain materi juga proses pemberian tugas dapat dilakukan menggunakan LMS UDINUS. Tugas yang sebelumnya dilakukan pada proses kuliah dikelas, dengan adanya LMS UDINUS dapat dilakukan diluar kelas.

LMS UDINUS juga akan menyediakan web service yang bertujuan agar data yang ada pada LMS UDINUS dapat digunakan pada sistem yang lain. Web service dikembangkan dengan mengunakan RESTFUL API agar dapat digunakan oleh sistem lain secara online. Web service menyediakan data atau juga dapat melakukan perintah tertentu terhadap data yang ada di LMS UDINUS.

# Kebutuhan Perangkat Lunak

## Deskripsi Umum Sistem

LMS UDINUS merupakan perangkat lunak berbasis web. LMS UDINUS digunakan untuk mengelola data mata kuliah dan data nilai yang digunakan untuk proses perkuliahan di UDINUS. LMS UDINUS diharapakan dapat meningkatkan efisiensi perkuliahan yang ada di UDINUS. Efisiensi tersebut bertujuan untuk mengurangi jarak antara mahasiswa dan dosen dalam melakukan proses perkuliahan. Proses perkuliahan dapat dilakukan menggunakan LMS UDINUS dengan menekankan materi dan tugas yang ada pada matakuliah tersebut. Dosen dapat memberikan materi yang disimpan di server dan mahasiswa dapat mengunduh materi tersebut melalui LMS UDINUS. Selain materi juga proses pemberian tugas dapat dilakukan menggunakan LMS UDINUS. Tugas yang sebelumnya dilakukan pada proses kuliah dikelas, dengan adanya LMS UDINUS dapat dilakukan diluar kelas.

LMS UDINUS juga akan menyediakan web service yang bertujuan agar data yang ada pada LMS UDINUS dapat digunakan pada sistem yang lain. Web service dikembangkan dengan mengunakan RESTFUL API agar dapat digunakan oleh sistem lain secara online. Web service menyediakan data atau juga dapat melakukan perintah tertentu terhadap data yang ada di LMS UDINUS. Implementasi dari LMS UDINUS menggunakan laravel sebagai framework web.

|  |  |
| --- | --- |
| ID | Deskripsi |
| SRS-F-01 | Sistem dapat membuat mata kuliah |
| SRS-F-02 | Sistem dapat menambah mata kuliah pada semester baru berdasarkan mata kuliah sebelumnya |
| SRS-F-03 | Sistem dapat menambah user baru |
| SRS-F-04 | Sistem dapat mendaftarkan user sebagai dosen atau mahasiswa pada mata kuliah tertentu |
| SRS-F-05 | Sistem dapat menambahkan user berdasarkan file dengan format excel |
| SRS-F-06 | Sistem dapat mengubah setting dari mata kuliah |
| SRS-F-07 | Sistem dapat menambah materi pada mata kuliah |
| SRS-F-08 | Sistem dapat menambah tugas pada mata kuliah |
| SRS-F-09 | Sistem dapat menambah assignment pada mata kuliah |
| SRS-F-10 | Sistem dapat menambah diskusi kelompok pada mata kuliah |
| SRS-F-11 | Sistem dapat mengelompokkan mahasiswa pada mata kuliah |
| SRS-F-12 | Sistem dapat memberikan nilai terhadap tugas dan assignment pada mata kuliah |
| SRS-F-13 | Sistem dapat digunakan untuk mengumpulkan tugas |
| SRS-F-14 | Sistem dapat digunakan untuk melakukan diskusi kelompok |
| SRS-F-15 | Sistem dapat digunakan untuk mengerjakan assignment |

## Fitur Utama Sistem

### Kebutuhan Functional

Table 2 Kebutuhan Functional

|  |  |
| --- | --- |
| ID | Deskripsi |
| SRS-F-01 | Sistem dapat membuat mata kuliah |
| SRS-F-02 | Sistem dapat menambah mata kuliah pada semester baru berdasarkan mata kuliah sebelumnya |
| SRS-F-03 | Sistem dapat menambah user baru |
| SRS-F-04 | Sistem dapat mendaftarkan user sebagai dosen atau mahasiswa pada mata kuliah tertentu |
| SRS-F-05 | Sistem dapat menambahkan user berdasarkan file dengan format excel |
| SRS-F-06 | Sistem dapat mengubah setting dari mata kuliah |
| SRS-F-07 | Sistem dapat menambah materi pada mata kuliah |
| SRS-F-08 | Sistem dapat menambah tugas pada mata kuliah |
| SRS-F-09 | Sistem dapat menambah assignment pada mata kuliah |
| SRS-F-10 | Sistem dapat menambah diskusi kelompok pada mata kuliah |
| SRS-F-11 | Sistem dapat mengelompokkan mahasiswa pada mata kuliah |
| SRS-F-12 | Sistem dapat memberikan nilai terhadap tugas dan assignment pada mata kuliah |
| SRS-F-13 | Sistem dapat digunakan untuk mengumpulkan tugas |
| SRS-F-14 | Sistem dapat digunakan untuk melakukan diskusi kelompok |
| SRS-F-15 | Sistem dapat digunakan untuk mengerjakan assignment |

### Kebutuhan Non-Functional

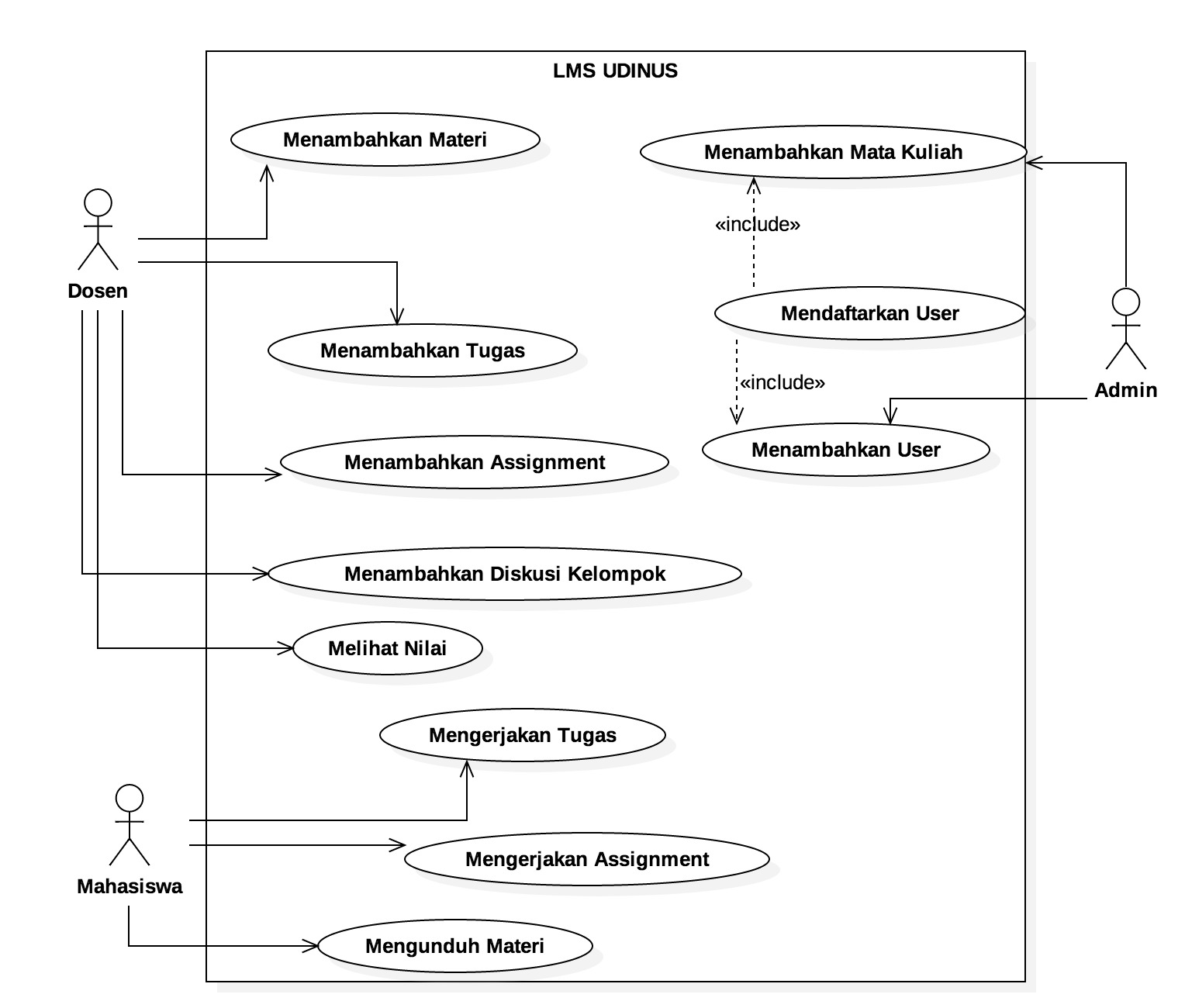
Table 3 Kebutuhan Non-Functional

|  |  |
| --- | --- |
| ID | Deskripsi |
| SRS-NF-01 | Sistem dilengkapi dengan login pengguna untuk dapat membedakan role pengguna yang sedang login |
| SRS-NF-02 | Sistem melakukan konfirmasi terhadap data yang akan dihapus |
| SRS-NF-03 | Judul pada halaman web relevan dengan konten yang ditampilkan |
| SRS-NF-04 | Sistem merahasiakan password yang disimpan di server |
| SRS-NF-05 | Sistem menyediakan informasi posisi user saat ini terhadap halaman utama |

## Model Use Case

Gambarkan struktur modul di sini:

### Diagram Use Case



Gambar 1 Use Case Diagram

### Definisi Actor

Table 4 Definisi Actor

|  |  |  |
| --- | --- | --- |
| ID | Actor | Definisi |
| AC-01 | Dosen | Dosen dapat menambahkan materi, tugas, quiz, dan assignment pada mata kuliah |
| AC-02 | Mahasiswa | Mahasiswa dapat mengunduh materi. Mengerjakan tugas, quiz dan assignment pada mata kuliah. |
| AC-03 | Admin | Admin dapat menambahkan user dan mendaftarkan kedalam kelompok mata kuliah |

### Definisi Use Case

Table 5 Definisi Use Case

|  |  |  |
| --- | --- | --- |
| ID | Use Case | Deskripsi |
| UC-01 | Menambahkan Mata Kuliah | Sistem menambahkan mata kuliah baru yang dimasukkan oleh admin ke database. Selain menambahkan mata kuliah baru, sistem dapat mengubah, dan menghapus mata kuliah yang ada pada database. |
| UC-02 | Menambahkan User | Sistem menambahkan user baru yang dimasukkan oleh admin ke database. Selain menambahkan user baru, sistem dapat mengubah, dan menghapus user yang sudah ada pada database. |
| UC-03 | Mendaftarkan User | Sistem menambahkan user kedalam kelompok mata kuliah tertentu dan mendefinisikan role yang digunakan oleh user tersebut. |
| UC-04 | Menambahkan Materi | Sistem menambahkan materi baru yang dimasukkan oleh dosen ke database. Selain menambahkan materi baru, sistem dapat mengubah, dan menghapus materi yang sudah ada pada database. |
| UC-05 | Menambahkan Tugas | Sistem menambahkan tugas baru yang dimasukkan oleh dosen ke database. Selain menambahkan tugas baru, sistem dapat mengubah dan menghapus tugas yang sudah ada pada database. |
| UC-06 | Menambahkan Assignment | Sistem menambahkan assignment baru yang dimasukkan oleh dosen ke database. Selain menambahkan assignment baru, sistem dapat mengubah dan menghapus assignment yang sudah ada pada database. |
| UC-07 | Menambahkan Diskusi Kelompok | Sistem menambahkan diskusi kelompok yang dimasukkan oleh dosen ke database. Selain menambahkan diskusi kelompok baru, sistem dapat mengubah dan menghapus diskusi kelompok yang sudah ada pada database. |
| UC-08 | Melihat Nilai | Sistem menampilkan nilai yang sudah masuk kedalam database. |
| UC-09 | Mengerjakan Tugas | Sistem menampilkan tugas yang sudah diunggah oleh d osen. Sistem dapan mengunggah file hasil pekerjaan mahasiswa kedalam database. |
| UC-10 | Mengerjakan Assignment | Sistem menampilkan assignment yang sudah diunggah oleh dosen. Sistem dapat menerima file assignment hasil pekerjaan mahasiswa kedalam database. |
| UC-11 | Mengunduh Materi | Sistem menampilkan materi yang sudah diunggah oleh dosen. Mahasiswa dapat mengunduh materi yang sudah diunggah oleh dosen. |

### Skenario Use Case

#### Use Case Menambahkan Mata Kuliah

Use Case : UC-1

Nama Use Case : Menambahkan Mata Kuliah

Actor : Admin

Deskripsi : Use case digunakan untuk menambahkan, mengubah, dan menghapus mata kuliah pada database.

Pre-Kondisi : Admin sudah login kedalam sistem

##### Skenario Normal

Skenario Id : SC-01-01

Deskripsi : Skenario normal, menambahkan mata kuliah kedalam database dan menampilkan daftar mata kuliah.

Skenario :

Table 6 SC-01-01

|  |  |
| --- | --- |
| Aksi Actor | Reaksi Sistem |
| 1. Menekan tombol tambah mata kuliah |  |
|  | 2. Menampilkan formulir tambah mata kuliah |
| 3. Memasukkan data mata kuliah |  |
| 4. Menekan tombol tambah |  |
|  | 5. Validasi isi formulir |
|  | 6. Menyimpan data mata kuliah |
|  | 7. Memberikan notifikasi bahwa penyimpanan berhasil |
| 8. Melihat daftar mata kuliah. |  |

##### Skenario Alternatif 1

Skenario Id : SC-01-02

Deskripsi : Skenario alternatif 1, menambahkan mata kuliah kedalam database namun masukan mata kuliah tidak valid

Skenario :

Table 7 SC-01-02

|  |  |
| --- | --- |
| Aksi Actor | Reaksi Sistem |
| 1. Menekan tombol tambah mata kuliah |  |
|  | 2. Menampilkan formulir tambah mata kuliah |
| 3. Memasukkan data mata kuliah |  |
| 4. Menekan tombol tambah |  |
|  | 5. Validasi isi formulir |
|  | 6. Menampilkan formulir tambah mata kuliah dengan notifikasi kesalahan masukan pada formulir |

##### Skenario Alternatif 2

Skenario Id : SC-01-03

Deskripsi : Skenario alternatif 2, edit mata kuliah yang sudah ada pada database dan berhasil update

Skenario :

Table 8 SC-01-03

|  |  |
| --- | --- |
| Aksi Actor | Reaksi Sistem |
| 1. Menekan tombol update mata kuliah |  |
|  | 2. Menampilkan formulir edit mata kuliah dengan nilai data mata kuliah sebelumnya |
| 3. Mengubah data mata kuliah |  |
| 4. Menekan tombol selesai |  |
|  | 5. Validasi isi formulir |
|  | 6. Memberikan notifikasi bahwa update berhasil |
| 7. Melihat daftar mata kuliah |  |

##### Skenario Alternatif 3

Skenario Id : SC-01-04

Deskripsi : Skenario alternatif 3, edit mata kuliah yang sudah ada pada database namun data masukan tidak valid

Skenario :

Table 9 SC-01-04

|  |  |
| --- | --- |
| Aksi Actor | Reaksi Sistem |
| 1. Menekan tombol update mata kuliah |  |
|  | 2. Menampilkan formulir edit mata kuliah dengan nilai data mata kuliah sebelumnya |
| 3. Mengubah data mata kuliah |  |
| 4. Menekan tombol selesai |  |
|  | 5. Validasi isi formulir |
|  | 6. Menampilkan formulir edit mata kuliah dengan notifikasi kesalahan masukan pada formulir |

# Deskripsi Perancangan Rinci

## Deskripsi Rinci Tabel

Setiap tabel pada rancangan global, dirinci satu per satu

### Tabel <Nama..>

Identifikasi/Nama : t\_master

Deskripsi Isi : ……..

Jenis : tabel datainduk/referensi/transaksi

Volume : ……….

Laju : ………….

Primary Key : ………….

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Id Field** | **Deskripsi** | **Tipe & length** | **Boleh**  **NULL** | **Default** | **Keterangan** |
| Id\_master |  | CHAR(8) | NO |  |  |
| Id\_ref |  |  |  |  | Refer ke t\_ref |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Catatan : kolom “Boleh NULL” berisi “NO” artinya tidak boleh kosong, berisi “YES” artinya boleh NULL

### <Table 2 : t\_Detail …>

Buat seperti di atas

## Deskripsi Rinci Modul

Setiap modul pada rancangan global, dirinci satu per satu

### Modul <xxx>

#### Fungsi Modul

|  |  |  |  |
| --- | --- | --- | --- |
| No | Fungsi | Jenis | Tabel Terkait |
|  |  | *Form Entry columnar/Tabular/Master-Detail*  *Report Columnar/tabular/Master-Detail*  *Form berisi dialog/button saja*  *Proses tanpa layar* |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

#### Spesifikasi Layar Utama

*Gambarkan layar dan percabangan ke layar lain function key/pilihan yang dilakukan)*

*Jika layar mengandung filed dan label, gambarkanlah pada posisi nya, supaya siap dikoding. Jika ada zoning/frame, gambarkan pula an jelaskan pada spesifikasi Objek pada layar*

Objek

OK

#### Spesifikasi Query (jika ada)

Tuliskan ekspresi Query jika layar didasari dari suatu Query dan bukan langsung dari Tabel :

**<nama Query > : Ekspresi**

#### Spesifikasi Field Data Layar

Catatan :

- Hanya ada isinya jika layar “dibelakang”nya memproses basisdata

- kolom validasi hanya diisi jika layar entry/Update (field ybs bukan hanya tampilan)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Label** | **Field** | **Tabel/**  **Query** | **Validasi** | **Keterangan** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

#### Spesifikasi Objek-Objek pada Layar

| **Id\_Objek** | **Jenis** | **Keterangan** |
| --- | --- | --- |
|  |  | Diisi dengan penjelasan reaksi sistem, misalnya membuka layar apa, link kemana. Jika menyangkut suatu kode yang cukup rumit, beri nama Prosedur terkait, dan uraiakn algoritmanya di bagian Spesifikasi Proses |
| *Button1* | Button | Jika diklik, akan mengaktifkan Proses AlgoXXX. |
| *RTF1* | *RTF Box* | Isi Teks yang disimpan pada File xxx |
| *DB1* | *Data control* | Diasosiasikan ke QueryXYZ yang dijelaskan pada bagian Query |

Jika objek dikaitkan ke File lain (misalnya file gambar, file teks), berikan nama file terkait dan deskripsi ringkas dalam kolom keterangan

#### Spesifikasi Proses/Algoritma

*Hanya diisi untuk proses/algoritma yang rumit*

**<Id. Proses > : ……**

**Objek terkait :**

**Event :**

**Initial State (IS)**:

Tabel … sudah dibuat strukturnya tetapi mungkin masih kosong

**Final State (FS)**:

Menghasilkan tabel TRANSAKSI yang telah ditambahkan satu rekord

**Spesifikasi Proses/algoritma**:

1. Cek apakah data yang dimasukkan ada
2. IF belum ada, tambahkan 1 record pada tabel t\_petugas

ELSE

Jika sudah ada, tampilkan pesan kesalahan sesuai layar XXXX (nomro layar)

Catatan : jika satu layar mengandung banyak button, dan untuk satu button ditulis sebuah prosedur, maka akan ada beberapa algoritma

## Matriks Kerunutan

| **ID Kebutuhan** | **Modul** | **Keterangan** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# 