**BAB IV ANALISIS DAN PERANCANGAN APLIKASI**

Pembangunan aplikasi ini menggunakan metode iterative incremental. Iterative incremental merupakan kombinasi antara model *prototyping* yang mengharuskan membuat contoh (*sample*) aplikasi sesuai dengan kebutuhan konsumen terlebih dahulu dan *incremental* dengan pengerjaan yang bertahap dan berulang-ulang sampai menghasilkan produk akhir yang sesuai (Kruchten, 2003).

Aplikasi yang dibangun merupakan aplikasi berbasis web yang digunakan untuk memonitoring kesesuaian BAP dengan RPS dosen. Pada aplikasi ini terdapat 8 aktor yaitu dosen pengajar, ketua kelompok keahlian, sekretaris kaprodi, dosen koordinator, kaprodi, dekan, wakil dekan, mahasiswa.

Terdapat beberapa fitur yang tersedia yaitu input jadwal mengajar dosen dan RPS oleh sekretaris kaprodi, mahasiswa dapat melihat RPS dosen, input BAP dosen setiap pertemuan di kelas, validasi kesesuaian BAP dengan RPS dosen oleh mahasiswa, menampilkan tabel hasil validasi kesesuaian BAP dengan RPS dosen dan forum.

## IV.1 *Increment* Pertama

Increment pertama merupakan tahap awal pembangunan aplikasi yang berisikan proses validasi kesesuaian BAP dengan RPS dosen. Pada proses validasi kesesuaian BAP dengan RPS terdapat beberapa fitur yaitu input jadwal dosen, input RPS, lihat RPS, input BAP dan validasi BAP.

### IV.1.1 Tahap Insepsi

Pada tahap awal pembuatan aplikasi diperlukan pendefinisian actor-aktor yang terlibat untuk mengetahui kebutuhan dari aplikasi tersebut. Pada tabel berikut terdapat beberapa aktor yang terlibat beserta fungsinya :

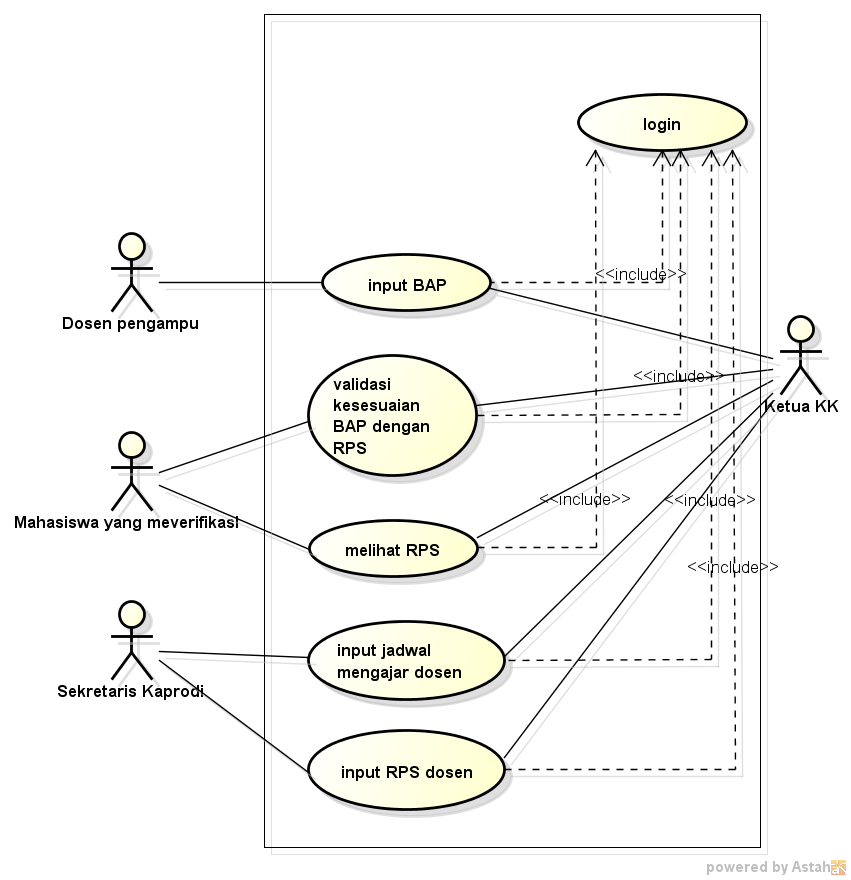
|  |  |
| --- | --- |
| Aktor | Deskripsi |
| Sekretaris Kaprodi | Bertanggung jawab menginputkan RPS dan jadwal mengajar dosen |
| Mahasiswa | Mevalidasi kesesuaian BAP dengan RPS dosen |
| Dosen Pengajar | Menginputkan BAP setiap pertemuan perkuliahan |
| Ketua Kelompok Keahlian | Bertugas memonitoring aktivitas pada aplikasi |

Pada tabel berikut terdapat beberapa aktivitas yang dapat dilakukan oleh aktor terkait increment pertama :

|  |  |  |  |
| --- | --- | --- | --- |
| Req ID | Aktor | Aksi yang dapat dilakukan | Yang terjadi setelah aksi dilakukan |
| 1 | Sekretaris Kaprodi | Menginputkan RPS dosen | RPS tersimpan di database sehingga dapat dilihat oleh mahasiswa |
| 2 | Sekretaris Kaprodi | Menginputkan jadwal mengajar dosen | Dosen dapat memilih jadwal mengajar yang telah berhasil diinputkan untuk mengisi BAP setiap pertemuan perkuliahan |
| 3 | Mahasiswa | Melihat RPS dosen | Mahasiswa memiliki RPS dosen yang digunakan untuk mevalidasi kesesuaian BAP dosen |
| 4 | Dosen | Menginputkan BAP | BAP yang berhasil diinputkan dapat divalidasi oleh mahasiswa di akhir pertemuan |
| 5 | Mahasiswa | Mevalidasi kesesuaian BAP dengan RPS | Fakultas memiliki hasil validasi kesesuaian BAP dengan RPS dosen |

### IV.1.2 Tahap Elaborasi

### IV.1.2.1 Use Case Diagram



Berikut use case description yang menjelaskan detail use case diagram pada increment pertama

|  |  |
| --- | --- |
| ID Use Case | 1 |
| Nama Use Case | Login |
| Aktor | 1. Dosen pengajar 2. *Knowledge Worker* (Ketua KK ) 3. Kaprodi 4. Sekretaris Kaprodi 5. Dosen Koordinator 6. Dekan 7. Wakil Dekan 8. Mahasiswa yang mevalidasi |
| Deskripsi | Melakukan login dengan cara memasukan *username* dan *password* agar dapat masuk kedalam menu utama |
| Pre-kondisi | Sudah memiliki *user name* dan *password* yang sudah tersimpan dalam *database* |
| Post-kondisi | 1. *User* berhasil *login* 2. Transaksi tercatat |
| Alur | 1. Menampilkan *form login* 2. *User* memasukan *username* dan *password* 3. Sistem melakukan validasi *user* dan *pasword* 4. Jika *username* dan *password* sesuai, maka akan muncul pemberitahuan *login* berhasil |
| Alur alternatif |  |
| Exceptions | 1. Apabila *login* gagal, maka akan muncul pemberitahuan *login* gagal 2. Mengulangi aktifitas 1-4 yang ada pada normal flow |
| Includes |  |

|  |  |
| --- | --- |
| ID Use Case | 2 |
| Nama Use Case | Input BAP |
| Aktor | Dosen Pengajar |
| Deskripsi | Aktor melakukan *input* BAP sesuai dengan pertemuan perkuliahan yang berlangsung |
| Pre-kondisi | 1. Sudah melakukan *login* 2. Sudah memilih menu *input* BAP |
| Post-kondisi | 1. BAP berhasil diinputkan 2. Data BAP tersimpan di database 3. Menampilkan pemberitahuan “BAP sudah berhasil diinputkan |
| Alur | 1. User menginputkan data   BAP kedalam form yang tersedia   1. Form BAP yang telah diinputkan, dapat disubmit apabila sudah benar |
| Alur alternatif |  |
| Exceptions | 1. Menampilkan pemberitahuan “anda belum memasukan data apapun” 2. Mengulangi aktifitas 1-2 yang ada pada alur*.* 3. Apabila *form* telah terisi tetapi tidak melakukan *submit*, maka data belum tersubmit dan belum tersimpan |
| Includes |  |

|  |  |
| --- | --- |
| ID Use Case | 3 |
| Nama Use Case | Input RPS |
| Aktor | Sekretaris Kaprodi |
| Deskripsi | Aktor melakukan *input* RPS sesuai dengan rencana perkuliahan yang telah dirancang |
| Pre-kondisi | 1. Sudah melakukan *login* 2. Sudah memilih menu *input* RPS |
| Post-kondisi | 1. RPS berhasil diinputkan 2. Data RPS tersimpan di database 3. Menampilkan pemberitahuan “RPS sudah berhasil |
| Alur | 1. Menginputkan data RPS kedalam form yang tersedia 2. Form RPS yang telah diinputkan, dapat disubmit apabila sudah benar |
| Alur alternatif |  |
| Exceptions | 1. Menampilkan pemberitahuan “anda belum memasukan data apapun” 2. Mengulangi aktifitas 1-2 yang ada pada alur*.* 3. Apabila *form* telah terisi tetapi tidak melakukan *submit*, maka data belum tersubmit dan belum tersimpan |
| Includes |  |

|  |  |
| --- | --- |
| ID Use Case | 4 |
| Nama Use Case | Lihat RPS |
| Aktor | Mahasiswa yang mevalidasi |
| Deskripsi | Aktor dapat melihat RPS sebelum meverifikasi kesesuaian BAP dengan RPS |
| Pre-kondisi | 1. Sudah melakukan *login* 2. Memilih menu lihat RPS 3. Data RPS sudah diinputkan kedalam sistem |
| Post-kondisi | Aktor berhasil melihat RPS dosen pengampu |
| Alur | 1. Membuka program dan mengklik menu lihat data RPS 2. Memilih dosen dan mata kuliah yang diampu |
| Alur alternatif |  |
| Exceptions |  |
| Includes |  |

|  |  |
| --- | --- |
| ID Use Case | 5 |
| Nama Use Case | Validasi kesesuaian BAP dengan RPS dosen |
| Aktor | Mahasiswa yang mevalidasi |
| Deskripsi | Aktor melakukan validasi kesesuain BAP terhadap RPS dosen di setiap pertemuan perkuliahan |
| Pre-kondisi | 1. Sudah melakukan *login* 2. Sudah melihat RPS dosen pengampu 3. Memilih menu verifikasi BAP |
| Post-kondisi | 1. BAP berhasil divalidasi 2. Data validasi BAP tersimpan di database 3. Menampilkan pemberitahuan “Validasi Berhasi” |
| Alur | 1. Memilih menu verifikasi kesesuaian BAP dengan RPS 2. Jika BAP sesuai dengan RPS maka user mencentang tombol sesuai, jika tidak sesuai maka user mencentang tombol tidak sesuai 3. Jika user sudah mengisi dengan benar, proses validasi dapat di submit |
| Alur alternatif |  |
| Exceptions | 1. Menampilkan pemberitahuan “anda belum mesukan data dengan lengkap” 2. Mengulangi aktifitas 1-3 yang ada pada alur 3. Apabila *form* telah terisi tetapi tidak melakukan *submit*, maka data belum tersubmit dan belum tersimpan |
| Includes |  |

|  |  |
| --- | --- |
| ID Use Case | 8 |
| Nama Use Case | Input jadwal dosen |
| Aktor | Sekretaris Kaprodi |
| Deskripsi | Aktor melakukan *input* jadwal mengajar dosen |
| Pre-kondisi | 1. Sudah melakukan *login* 2. Sudah memilih menu *input* jadwal dosen |
| Post-kondisi | 1. Jadwal mengajar dosen berhasil diinputkan 2. Data jadwal mengajar dosen tersimpan di database 3. Menampilkan pemberitahuan “jadwal dosen sudah berhasil diinputkan” |
| Alur | 1. Menginputkan data diri dosen kedalam form yang tersedia 2. Form jadwal dosen yang telah diinputkan, dapat disubmit apabila sudah benar |
| Alur alternatif |  |
| Exceptions | 1. Menampilkan pemberitahuan “anda belum memasukan data apapun” 2. Mengulangi aktifitas 1-2 yang ada pada alur*.* 3. Apabila *form* telah terisi tetapi tidak melakukan *submit*, maka data belum tersubmit dan belum tersimpan |
| Includes |  |