**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 memonitor 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

Fase awal dalam membuat suatu sistem adalah membuat suatu rencana dengan cara mengidentifikasi ruang lingkup/*scope*, persyaratan/*requirement* baik fungsional maupun non-fungsional, dan analisis risiko yang cukup untuk meminimalisir terjadinya kesalahan di kemudian hari. Hasil akhir yang dapat diperoleh adalah perkiraan visi, survei *use case*, *business case* awal, penilaian risiko awal, *scope* proyek dan lain lain (Larman, 2005).

Pada tahap awal pembuatan aplikasi diperlukan pendefinisian aktor-aktor yang terlibat untuk mengetahui kebutuhan dari aplikasi tersebut. Pada tabel IV.1 terdapat beberapa aktor yang terlibat beserta fungsinya:

Tabel IV.1 Identifikasi Aktor *Increment* Pertama

|  |  |
| --- | --- |
| 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 aktivitas pada aplikasi |

Pada Tabel IV.2 terdapat beberapa aktivitas yang dapat dilakukan oleh aktor terkait *increment* pertama:

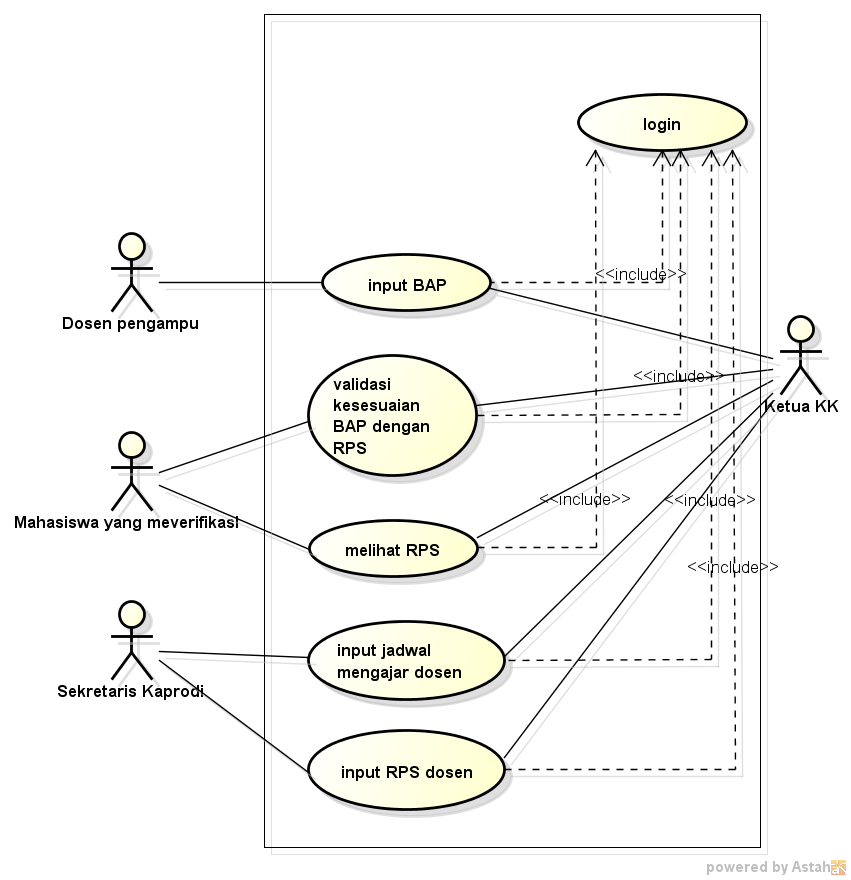
Tabel IV.2 Analisis Kebutuhan *Increment* Pertama

|  |  |  |  |
| --- | --- | --- | --- |
| Req ID | Aktor | Aksi | Kondisi setelah aksi |
| 1 | Semua Aktor | *Login* | Berhasil *login* dan masuk ke dalam aplikasi |
| 2 | Sekretaris Kaprodi | Menginputkan RPS dosen | RPS tersimpan di *database* sehingga dapat dilihat oleh mahasiswa |
| 3 | Sekretaris Kaprodi | Menginputkan jadwal mengajar dosen | Dosen dapat memilih jadwal mengajar yang telah berhasil diinputkan untuk mengisi BAP setiap pertemuan perkuliahan |
| 4 | Mahasiswa | Melihat RPS dosen | Mahasiswa memiliki RPS dosen yang digunakan untuk mevalidasi kesesuaian BAP dosen |
| 5 | Dosen | Menginputkan BAP | BAP yang berhasil diinputkan dapat divalidasi oleh mahasiswa di akhir pertemuan |
| 6 | Mahasiswa | Mevalidasi kesesuaian BAP dengan RPS | Fakultas memiliki hasil validasi kesesuaian BAP dengan RPS dosen |
| 7 | Ketua Kelompok Keahlian | memonitor aktivitas di setiap fitur aplikasi | Ketua kelompok keahlian mengetahui aktivitas yang terjadi pada aplikasi |

### IV.1.2 Tahap Elaborasi

Pada fase ini menggambarkan sistem untuk memenuhi persyaratan nonfungsional. Bisa dilaksanakan dalam dua iterasi atau lebih. Artefak yang dapat dihasilkan adalah visi yang diperjelas, *use case* awal, deskripsi awal kasus pengembangan untuk menentukan proses yang digunakan, dan lain lain (Larman, 2005).

### IV.1.2.1 *Use Case Diagram*



Gambar IV.1 *Use Case Diagram Increment* Pertama

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

Tabel IV.3 *Use Case Form 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 aktivitas 1-4 yang ada pada *normal flow* |
| *Includes* |  |

|  |  |
| --- | --- |
| ID *Use Case* | 2 |
| 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 aktivitas 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 disubmitapabila sudah benar |
| Alur alternatif |  |
| *Exceptions* | 1. Menampilkan pemberitahuan “anda belum memasukan data apapun” 2. Mengulangi aktivitas 1-2 yang ada pada alur*.* 3. Apabila *form* telah terisi tetapi tidak melakukan *submit*, maka data belum tersubmit dan belum tersimpan |
| *Includes* |  |

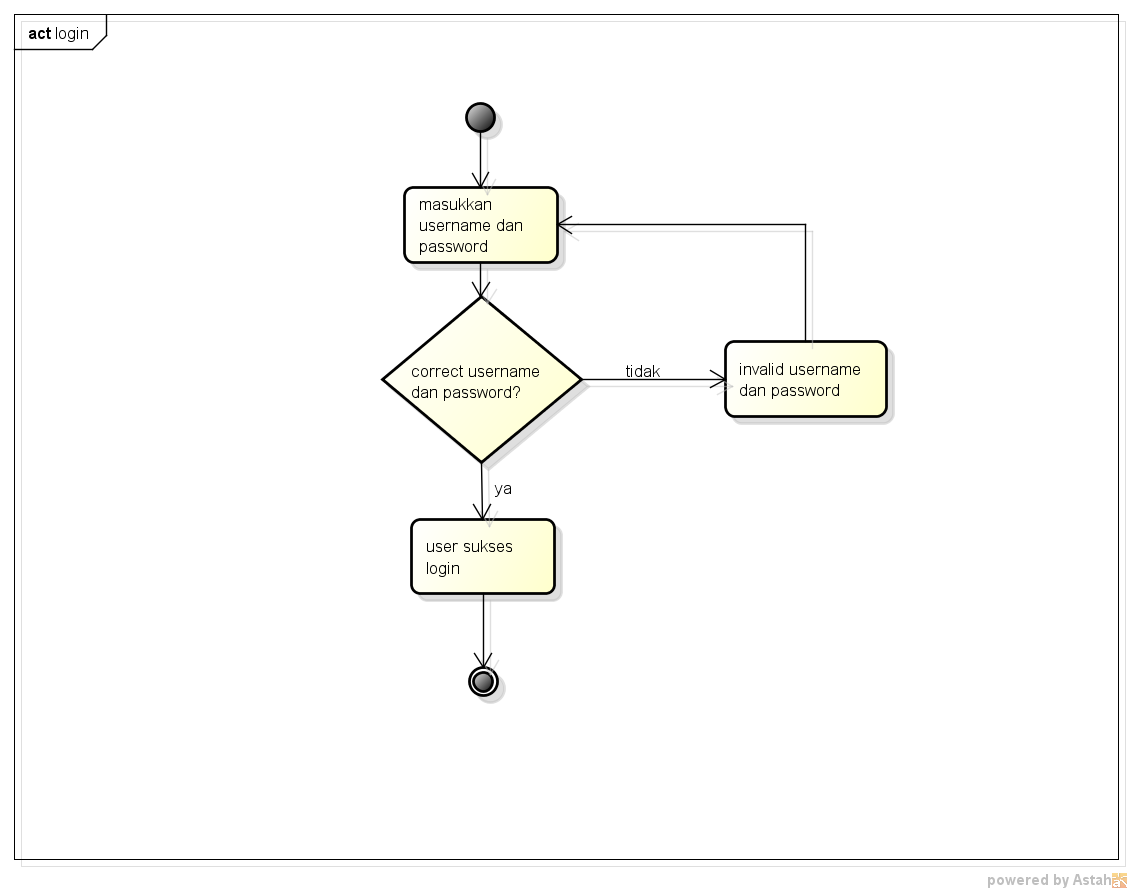
|  |  |
| --- | --- |
| ID *Use Case* | 4 |
| Nama *Use Case* | *Input* BAP |
| Aktor | Dosen Pengajar |
| Deskripsi | Aktor melakukan *input* BAP sesuai dengan pertemuan perkuliahan yang berlangsung |
| Pre-kondisi | 1. Sudah melakukan *login* 2. Sudah memilih menu *input* BAP |
| Post-kondisi | 1. 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 aktivitas 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 | 5 |
| Nama Use Case | Lihat RPS |
| Aktor | Mahasiswa yang mevalidasi |
| Deskripsi | Aktor dapat melihat RPS sebelum mevalidasi 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* |  |

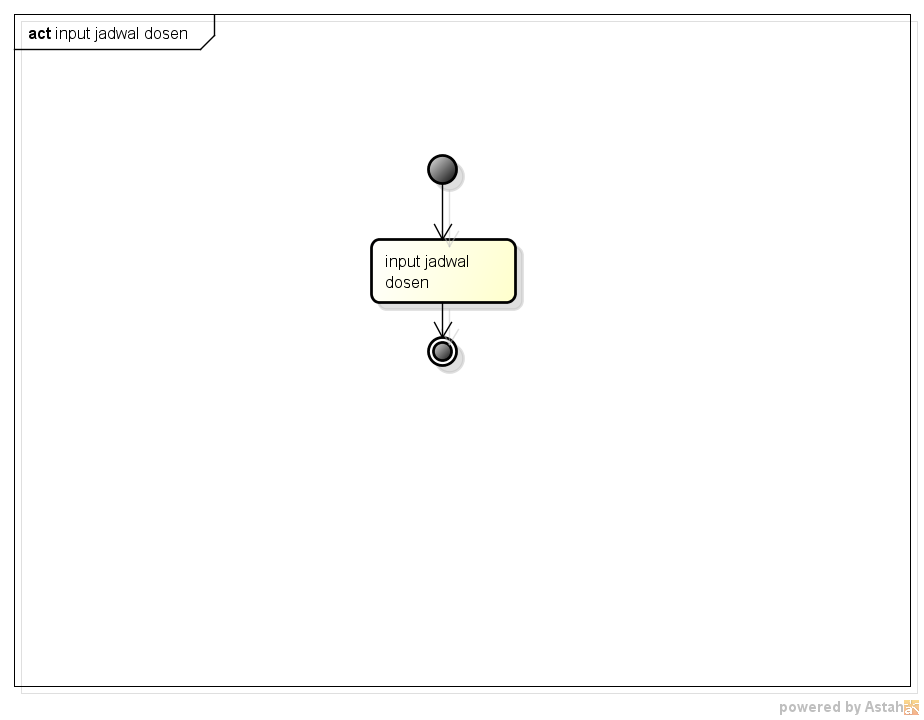
|  |  |
| --- | --- |
| ID Use Case | 6 |
| Nama Use Case | Validasi kesesuaian BAP dengan RPS dosen |
| Aktor | Mahasiswa yang mevalidasi |
| Deskripsi | Aktor melakukan validasi kesesuain BAP terhadap RPS dosen di setiap pertemuan perkuliahan |
| Pre-kondisi | 1. Sudah melakukan *login* 2. Sudah melihat RPS dosen pengampu 3. Memilih menu validasi BAP |
| Post-kondisi | 1. BAP berhasil divalidasi 2. Data validasi BAP tersimpan di *database* 3. Menampilkan pemberitahuan “Validasi Berhasil” |
| Alur | 1. Memilih menu validasi 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 aktivitas 1-3 yang ada pada alur 3. Apabila *form* telah terisi tetapi tidak melakukan *submit*, maka data belum tersubmit dan belum tersimpan |
| *Includes* |  |

### IV.1.2.2 *Activity Diagram*

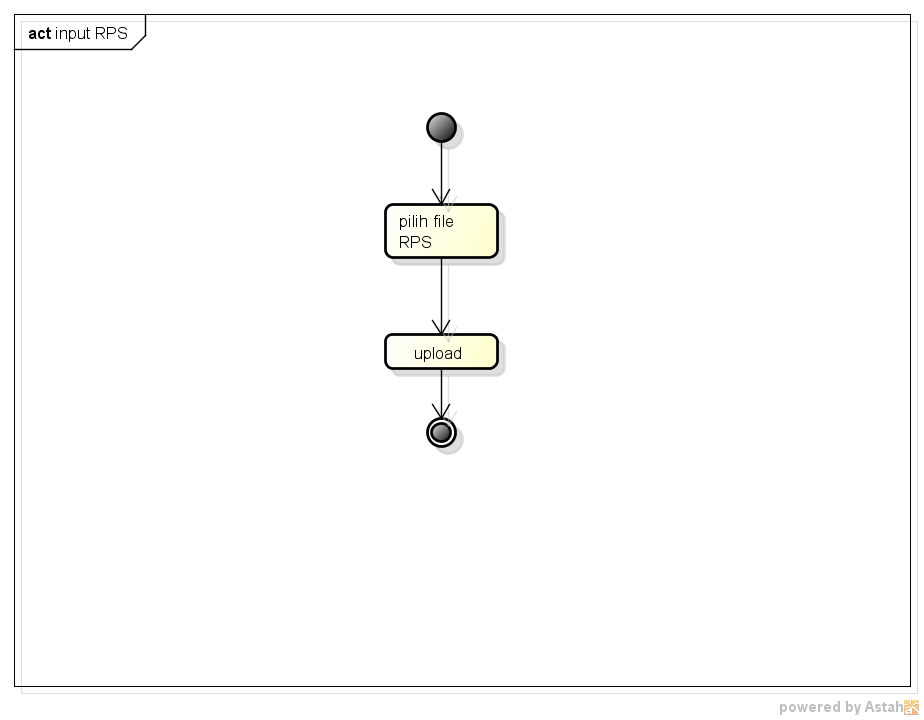
Berikut *activity diagram* yang menjelaskan alur aktivitas yang dilakukan aktor terkait *increment* pertama.



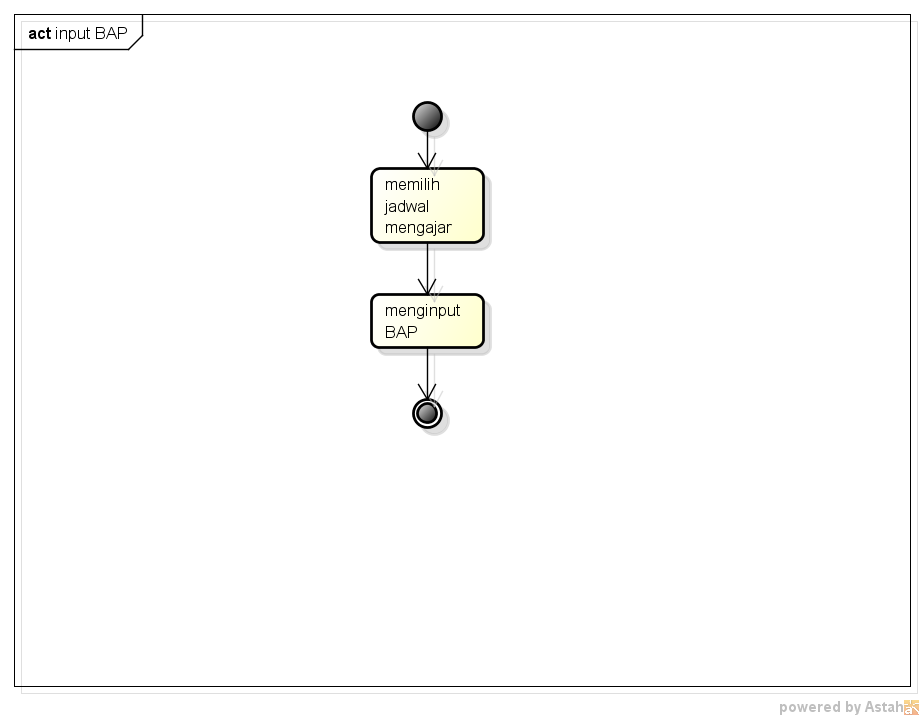
Gambar IV.2 *Activity* *Diagram Login*



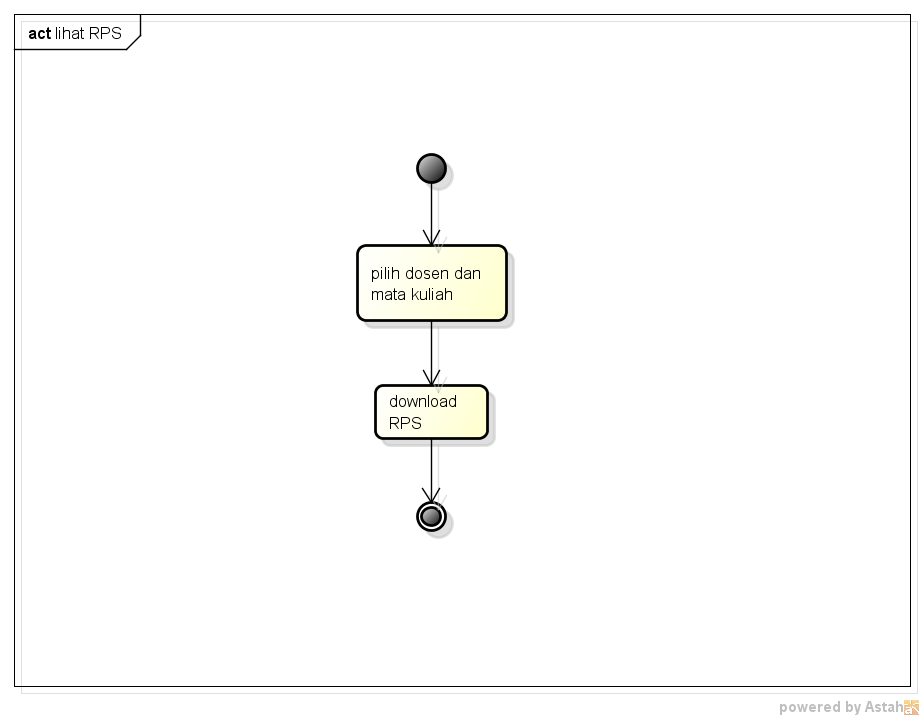
Gambar IV.3 *Activity Diagram Input* Jadwal Dosen



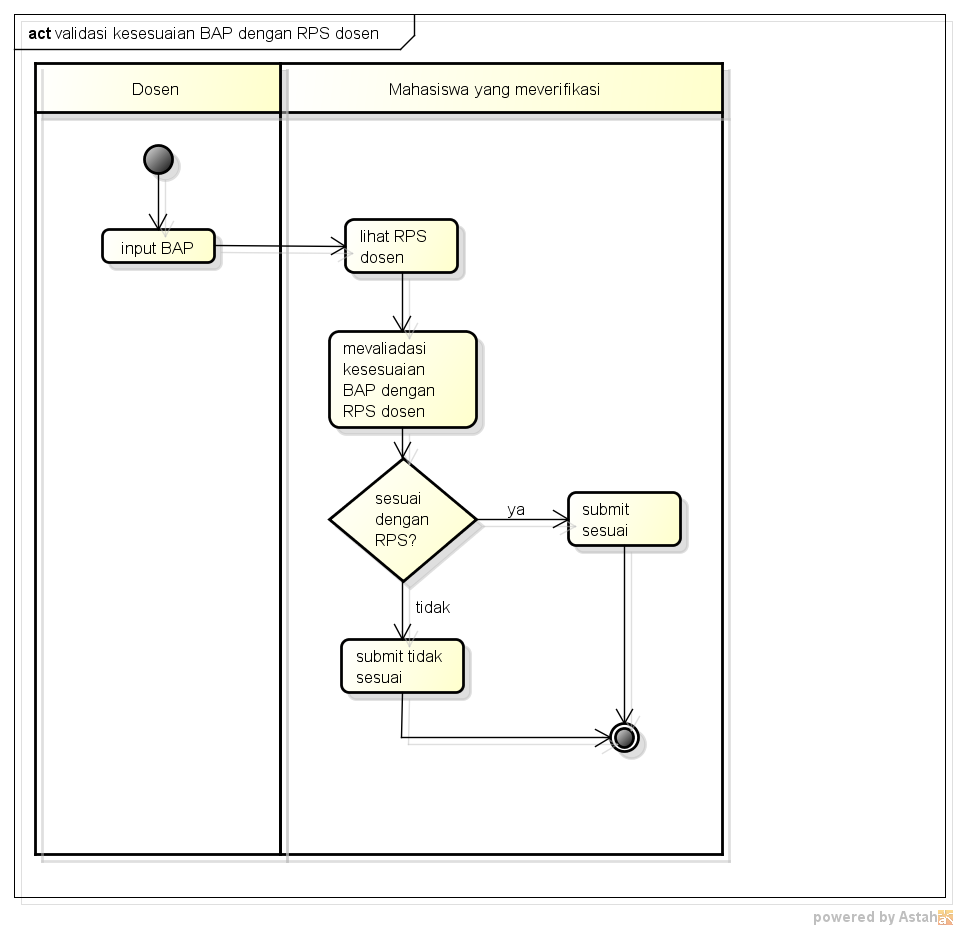
Gambar IV.4 *Activity Diagram Input* RPS



Gambar IV.5 *Activity Diagram Input* BAP



Gambar IV.6 *Activity Diagram* Lihat RPS



Gambar IV.7 Activity Diagram Lihat RPS

### IV.1.2.3 *Class Diagram*

### IV.1.2.4 *Sequence Diagram*

Berikut merupakan *sequence diagram* pada *increment* pertama:



Gambar IV.8 *Sequence Diagram Login*



Gambar IV.9 *Sequence Diagram Input* Jadwal Dosen



Gambar IV.10 *Sequence Diagram Input* RPS



Gambar IV.11 *Sequence Diagram Input* BAP



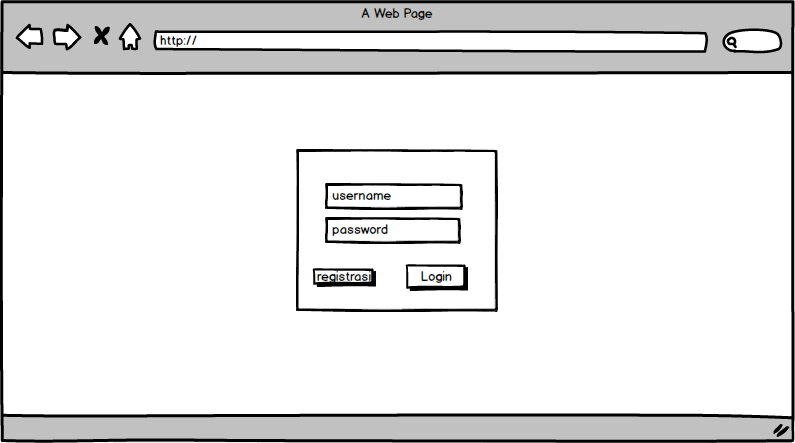
Gambar IV.12 *Sequence Diagram* Validasi Kesesuai BAP dengan RPS Dosen



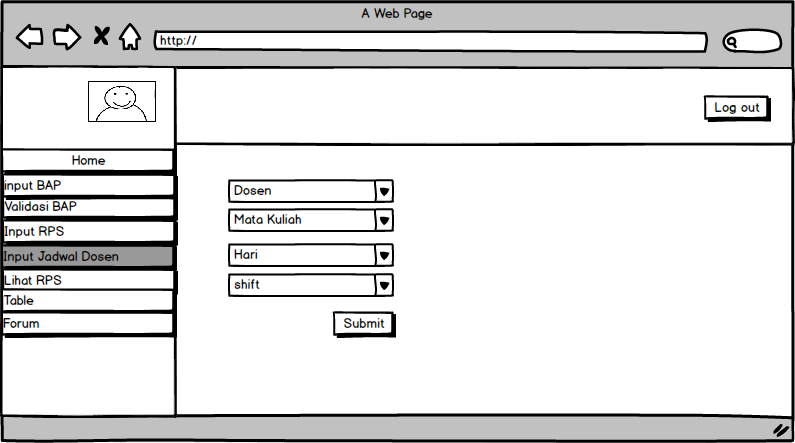
Gambar IV.13 *Sequence Diagram* Lihat RPS

### IV.1.2.5 Desain *Mockup*

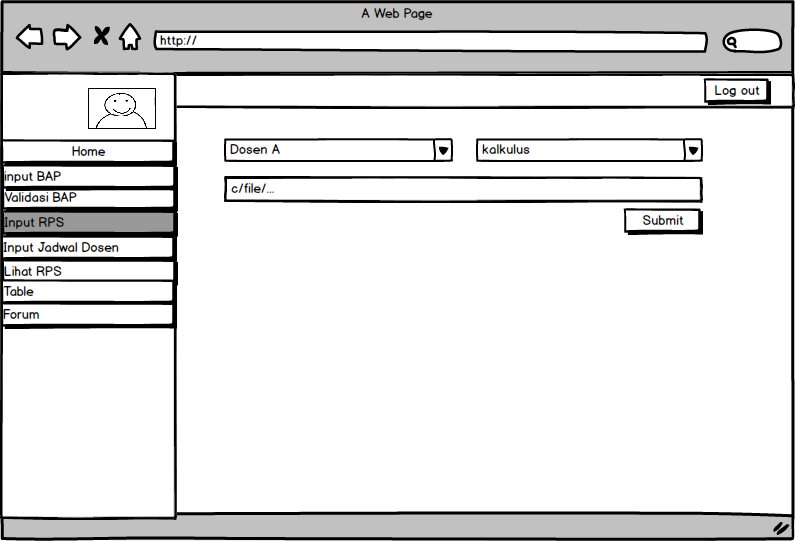
Berikut merupakan desain *mockup* pada *increment* pertama:



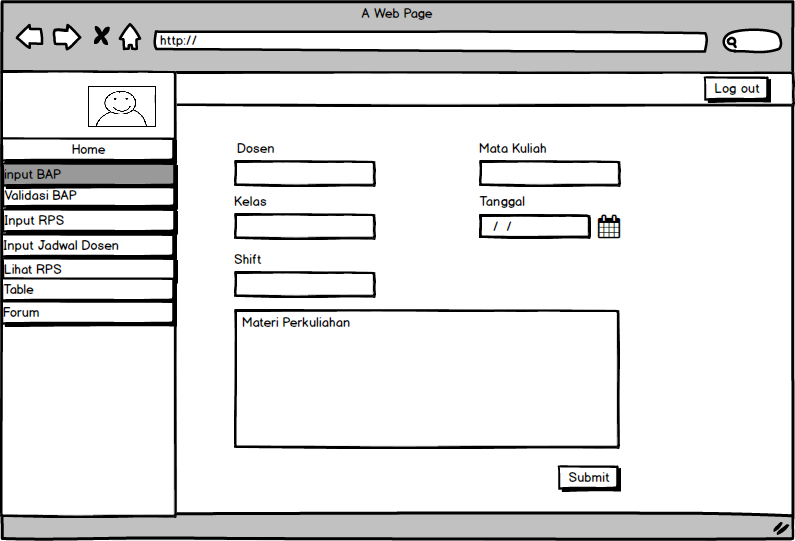
Gambar IV.14 Desain *Mockup Login*



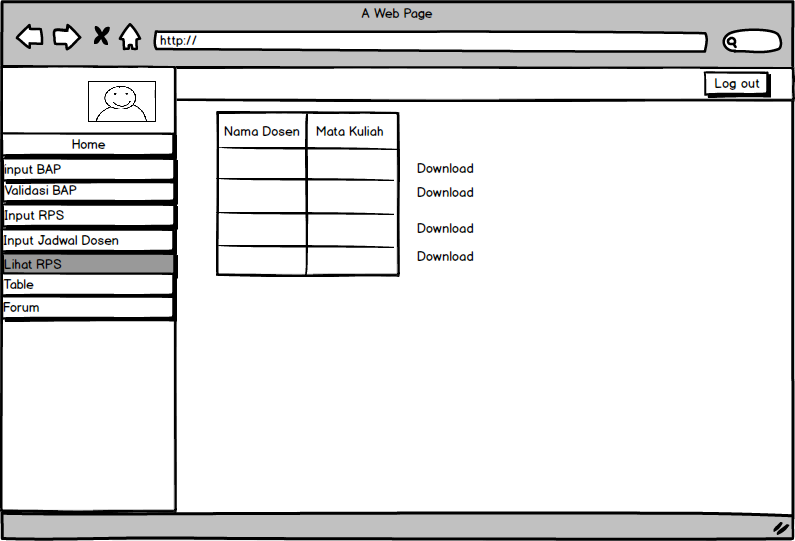
Gambar IV.15 Desain *Mockup Input* Jadwal Dosen



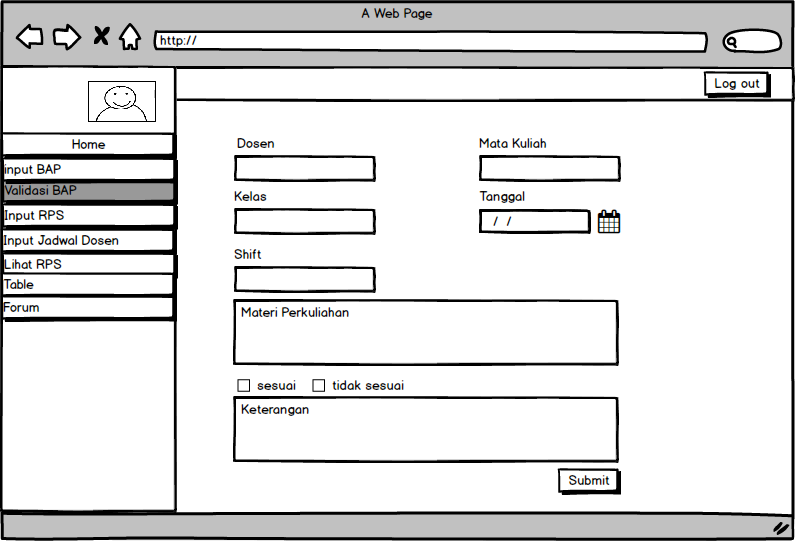
Gambar IV.16 Desain *Mockup Input* RPS



Gambar IV.17 Desain *Mockup Input* BAP



Gambar IV.18 Desain *Mockup Lihat* RPS



Gambar IV.19 Desain *Mockup* Validasi Kesesuaian BAP dengan RPS

### IV.1.3 Tahap Kontruksi

Dilakukan secara bertahap dengan cara mengisi sebuah perangkat lunak yang awalnya masih berbentuk rancangan arsitektur dengan kode produksi yang telah siap yang dihasilkan dari analisis, desain, implementasi, dan pengujian persyaratan fungsional untuk didistribusikan ke pengguna. Sama seperti fase sebelumnya, *fase* ini bisa dilaksanakan dalam dua atau lebih iterasi (Larman, 2005).

### IV.1.3.1 Implementasi *File*

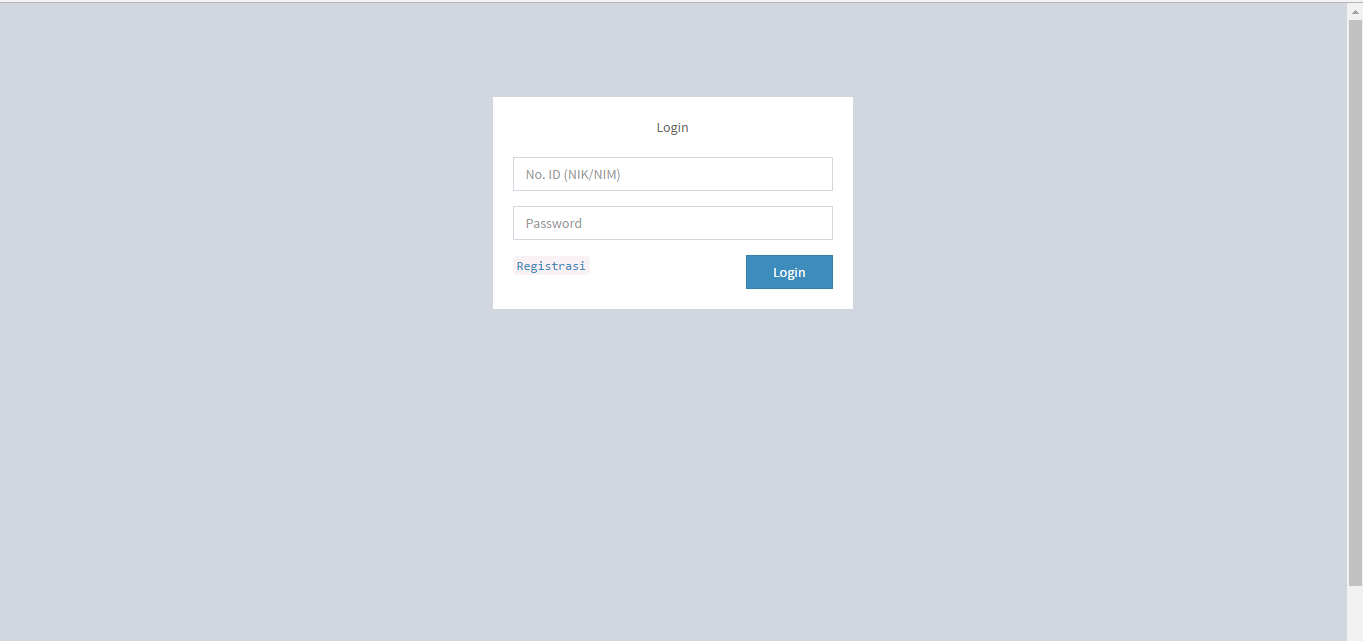
Pada tahap ini dijelaskan komponen-komponen yang digunakan untuk membangun aplikasi validasi kesesuaian BAP dengan RPS dosen. Komponen tersebut merupakan *class-class* yang terdapat di aplikasi. Berikut nama *class* dan deskripsinya:

Tabel IV.4 Implementasi *File Increment* Pertama

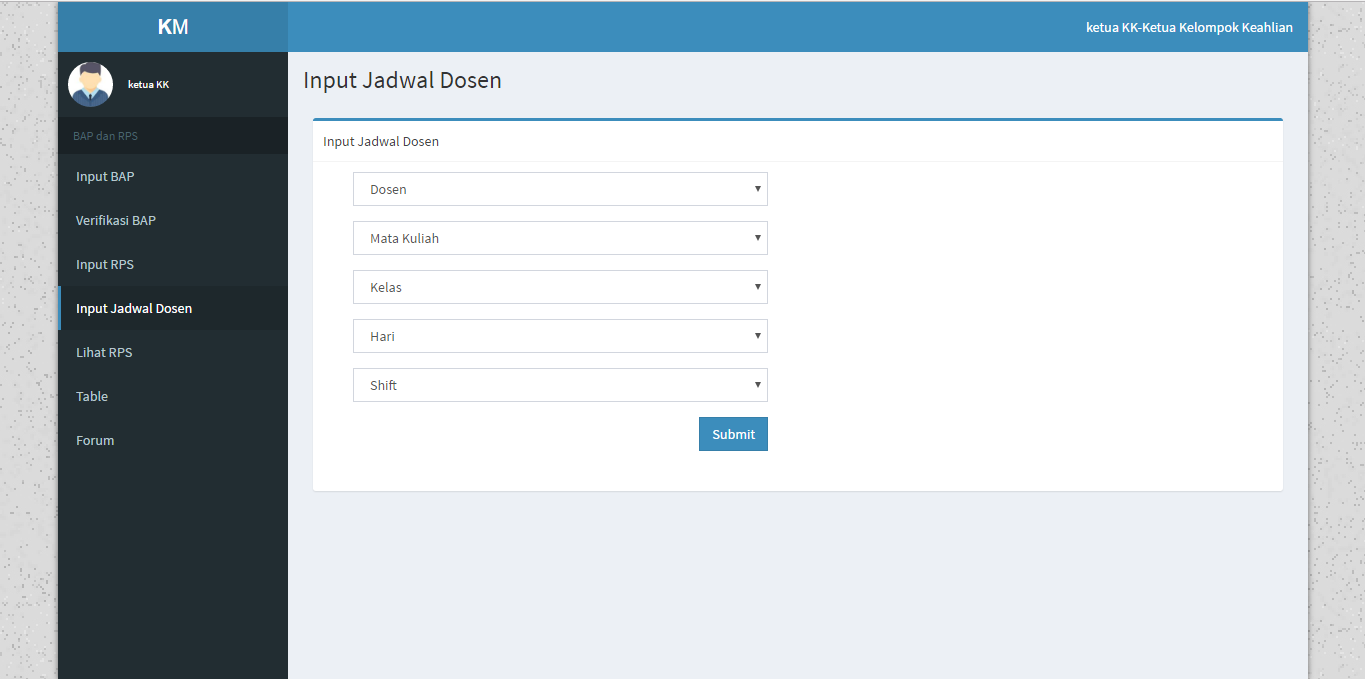
|  |  |  |  |
| --- | --- | --- | --- |
| No | MVC | Nama Class | Deskripsi |
| 1 | *Controller* | Login.php | File *controller* yang berisikan *method/*fungsi dari aktivitas *login* |
| 2 | Baprps.php | *File controller* yang berfungsi untuk menampung seluruh *method*/fungsi dari seluruh aktivitas pada aplikasi validasi kesesuaian BAP dengan RPS kecuali aktivitas *login* |
| 3 | *View* | input\_bap.php | *File view* yang berfungsi untuk menampilkan jadwal mengajar dosen |
| 4 | input\_bap2.php | *File view* yang berfungsi untuk menampilkan *form* BAP yang ditampilkan setelah memilih jadwal mengajar dosen |
| 5 | input\_jadwal.php | *File view* yang berfungsi untuk menampilkan *form input* jadwal dosen |
| 6 | input\_rps.php | *File view* yang berfungsi untuk menampilkan *form input* RPS |
| 7 | lihat\_rps.php | *File view* yang berfungsi untuk menampilkan *list* RPS dosen yang sudah terimpan di *database* |
| 8 |  | verif\_bap.php | File view yang berfungsi untuk menampilkan list BAP dosen yang sudah diinputkan dan siap untuk divalidasi |
| 9 | verif2.php | File view yang berfungsi untuk menampilkan form validasi kesesuaian BAP dengan RPS dosen |
| 10 | header.php | File view yang berfungsi untuk menampilkan bagian atas aplikasi |
| 11 | v\_login.php | File view yang berfungsi untuk menampilkan tampilan form login |
| 12 | *Model* | m\_login.php | File model yang berisikan pengelolaan data ke database seperti insert, update, select dan delete pada database khusus untuk fitur login |
| 13 | m\_inputbap.php | File model yang berisikan pengelolaan data ke database seperti insert, update, select dan delete pada database khusus untuk input BAP dosen dan forum |
| 14 | m\_inputrps.php | File model yang berisikan pengelolaan data ke database seperti insert, update, select dan delete pada database khusus untuk input RPS dosen |
| 15 | m.verif.php | File model yang berisikan pengelolaan data ke database seperti insert, update, select dan delete pada database khusus untuk validasi kesesuaian BAP dengan RPS dosen |

### IV.1.3.2 Tampilan Program

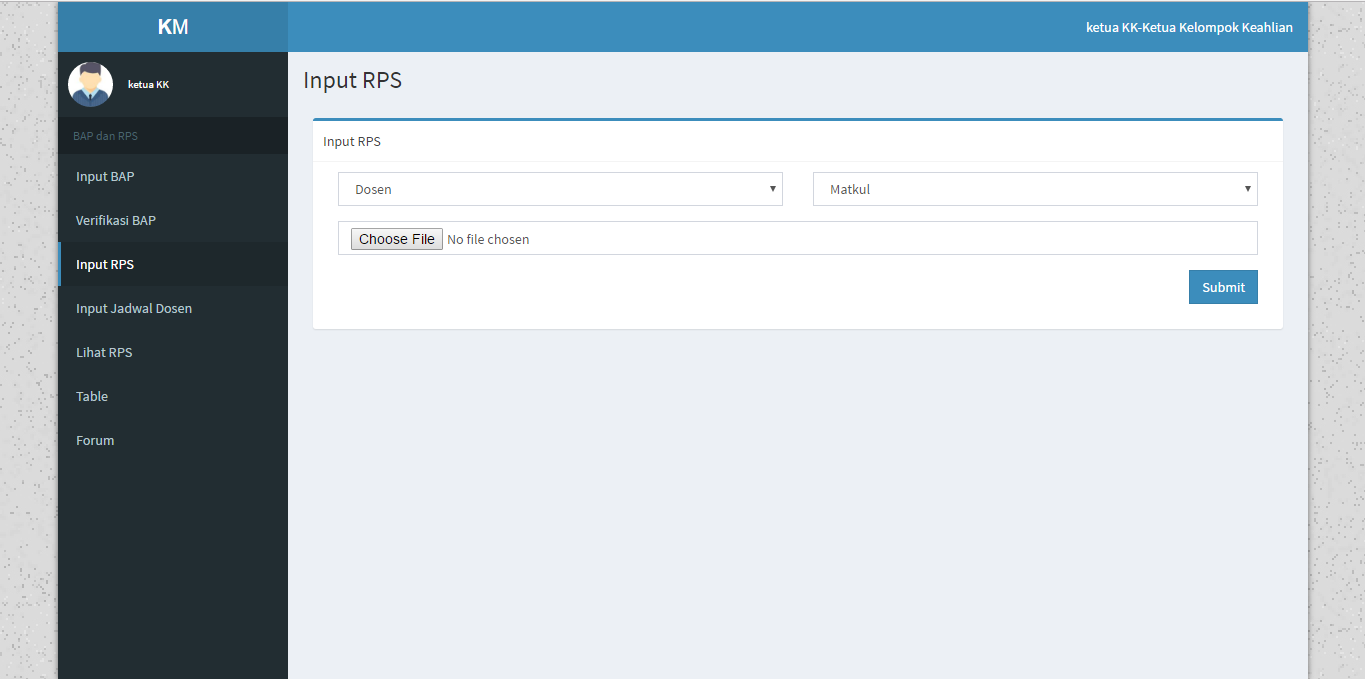
Berikut merupakan tampilan aplikasi terkait *increment* pertama:



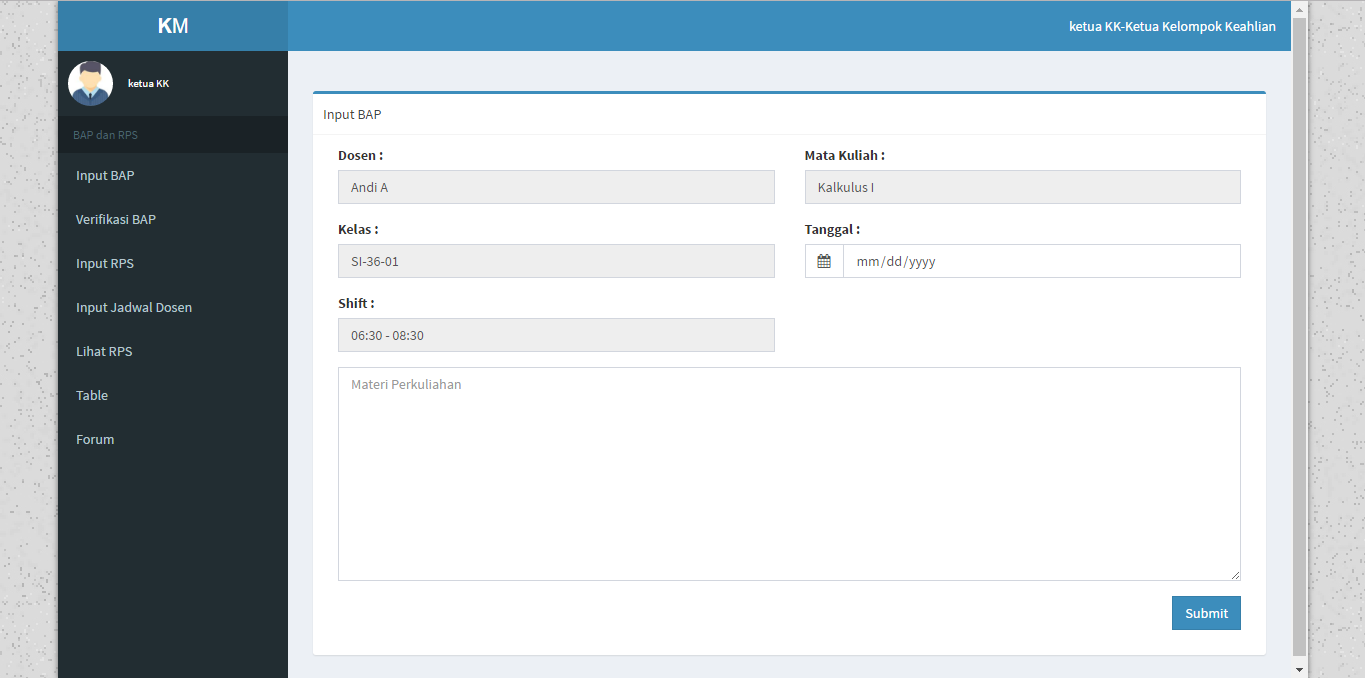
Gambar IV.20 Tampilan Halaman *Login*



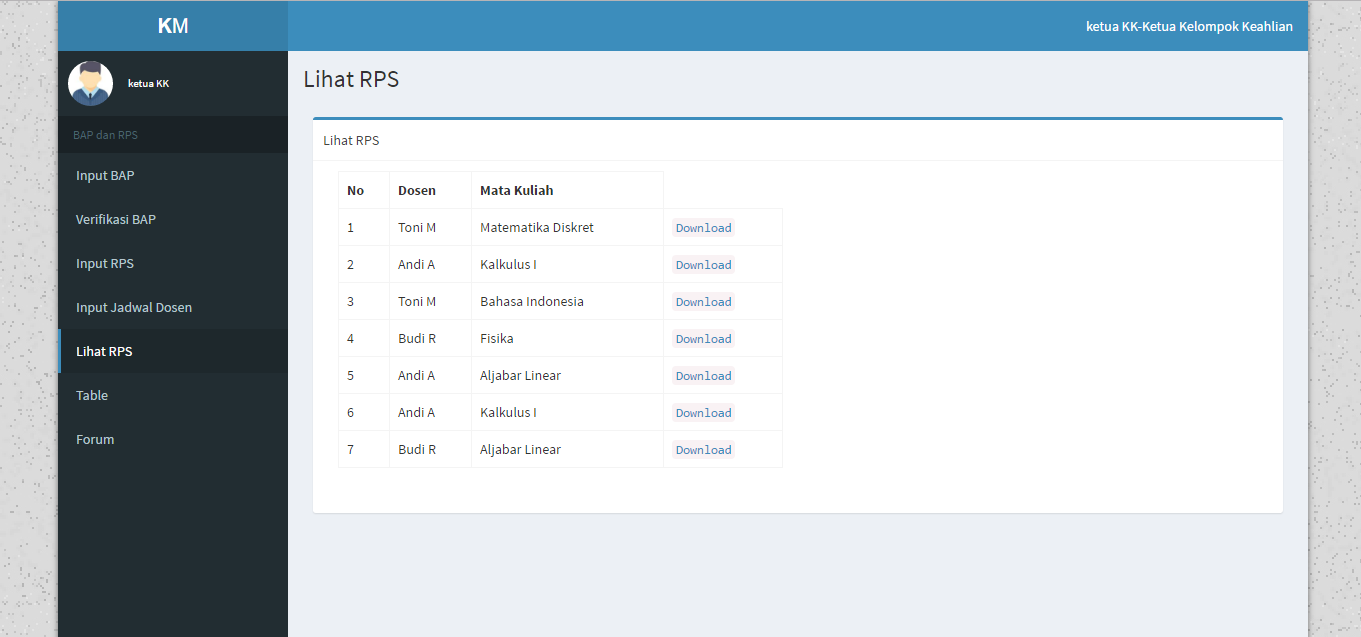
Gambar IV.21 Tampilan Halaman *Input* Jadwal Dosen



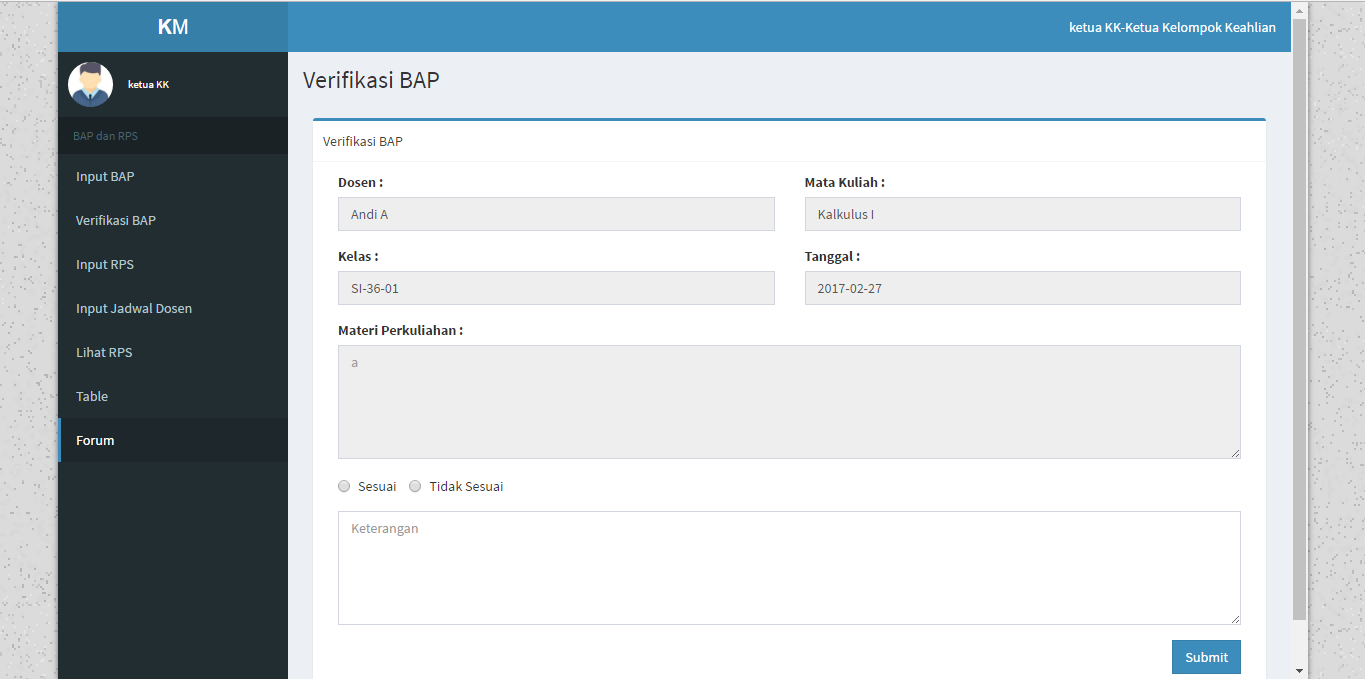
Gambar IV.22 Tampilan Halaman *Input* RPS



Gambar IV.23 Tampilan Halaman pada *Input* BAP



Gambar IV.24Tampilan Halaman Lihat RPS



Gambar IV.25 Tampilan Halaman pada Validasi Kesesuaian BAP dengan RPS

### IV.1.4 Tahap Transisi

Difokuskan untuk mengenalkan produk yang sudah dihasilkan ke pengguna yang bersangkutan, menyelesaikan pengujian versi beta, pelatihan ke *user* dan pengujian *user acceptance*. Hasil artefak dari fase ini adalah dokumentasi buku manual untuk pengguna, aplikasi/sistem yang sudah kompleks (Larman, 2005).

### IV.1.4.1 Pengujian

Berikut merupakan hasil dari pengujian yang dilakukan oleh *developer* dari aplikasi pada *increment* pertama :

Tabel IV.5 Daftar Hasil Pengujian *Increment* Pertama

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Kode Pengujian | Prosedur Pengujian | Hasil yang Diharapkan | Hasil |
| 1 | Req 1-1 | *Login* dengan *username* dan *password* yang sesuai | Berhasil *login* dan masuk ke halaman berdasarkan otoritas masing-masing *user* |  |
| 2 | Req 2-1 | Membuka menu *input* RPS dosen | Menampilkan *form input* RPS dosen |  |
| 3 | Req 2-2 | Mengisi semua *field* yang tersedia dan submit | Menampilkan notifikasi sukses *input* RPS dosen |  |
| 4 | Req 3-1 | Membuka menu *input* jadwal dosen | Menampilkan *form input* jadwal dosen yang harus diisi |  |
| 5 | Req 3-2 | Mengisi semua *field* yang tersedia dan *submit* | Menampilkan notifikasi sukses *input* jadwal dosen |  |
| 6 | Req 4-1 | Membuka menu lihat RPS dosen | Menampilkan *list* RPS dosen |  |
| 7 | Req 4-2 | Mendownload RPS dosen | RPS sukses didownload |  |
| 8 | Req 5-1 | Membuka menu *input* BAP | Menampilkan jadwal mengajar dosen |  |
| 9 | Req 5-2 | Memilih jadwal mengajar untuk *input* BAP | Menampilkan detail jadwal dosen sesuai dengan yang dipilih |  |
| 10 | Req 5-3 | Mengisikan materi perkuliahan dan *submit* | Menampilkan notifikasi berhasil *input* BAP |  |
| 11 | Req 6-1 | Membuka menu validasi BAP | Menampilkan daftar BAP dosen yang telah diinputkan |  |
| 12 | Req 6-2 | Memilih BAP dosen untuk divalidasi | Menampilkan detail BAP dosen sesuai dengan yang dipilih |  |
| 13 | Req 6-3 | Mengisikan *form* yang tersedia dan *submit* | Memberikan notifikasi berhasil mevalidasi kesesuaian BAP dengan RPS dosen |  |
| 14 | Req 7-1 | Membuka seluruh menu yang tesedia di aplikasi | Menampilkan seluruh tampilan dari masing-masing menu yang dibuka |  |

## IV.2 *Increment* Kedua

Fitur yang akan dibuat pada *increment* kedua yaitu fitur memonitor dan forum. Pada *i*ncrement kedua, terdapat dua menu yaitu menu tabel yang digunakan untuk melakukan aktivitas memonitor dan menu *forum* yang berfungsi untuk membantu terjadinya *knowledge sharing* dimana aktor dapat membuat *thread* dan saling mengomentari *thread* tersebut.

### IV.2.1 Tahap Insepsi

Fase awal dalam membuat suatu sistem adalah membuat suatu rencana dengan cara mengidentifikasi ruang lingkup/*scope*, persyaratan/*requirement* baik fungsional maupun non-fungsional, dan analisis risiko yang cukup untuk meminimalisir terjadinya kesalahan di kemudian hari. Hasil akhir yang dapat diperoleh adalah perkiraan visi, survei *use case*, *business case* awal, penilaian risiko awal, *scope* proyek dan lain lain (Larman, 2005).

Tahap insepsi pada *increment* kedua, terdapat beberapa aktor yang terlibat. Pada Tabel IV.6 akan dijelaskan aktor dan fungsinya sebagai berikut:

Tabel IV.6 Identifikasi Aktor *Increment* Kedua

|  |  |
| --- | --- |
| Aktor | Deskripsi |
| Dekan, Wakil Dekan, Kaprodi, Dosen Koordinator | Bertugas untuk melakukan *monitoring* pada proses pengajaran yang dilakukan oleh dosen dan memiliki hak untuk memberikan *feedback* kepada dosen secara langsung maupun melalui *forum* |
| Dosen Pengajar | Memiliki hak untuk mendapatkan *feedback* dari entitas lainnya untuk proses pengajaran yang lebih baik |
| Ketua Kelompok Keahlian | Bertugas memonitor seluruh aktivitas pada aplikasi |
| Sekretaris Kaprodi | Memiliki hak akses kedalam *forum* |

Pada Tabel IV.7 terdapat beberapa aktivitas yang dapat dilakukan oleh aktor terkait *increment* kedua:

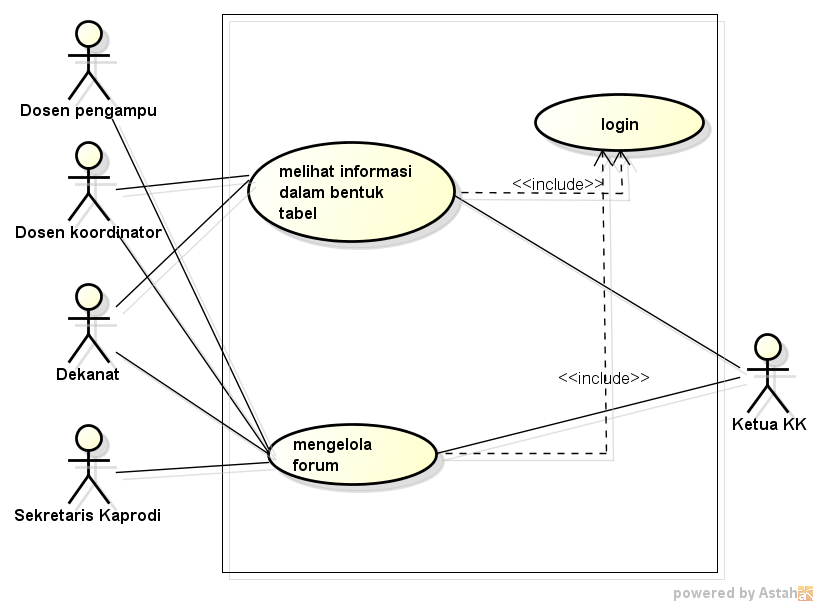
Tabel IV.7 Analisis Kebutuhan *Increment* Kedua

|  |  |  |  |
| --- | --- | --- | --- |
| Req ID | Aktor | Aksi | Kondisi setelah aksi |
| 1 | Dekanat, Dosen Koordinator | Melihat hasil dari proses validasi kesesuaian BAP dengan RPS dosen dalam bentuk tabel | Dapat memberikan *feedback* kepada dosen pengajar |
| 2 | Dekanat, Dosen Koordinator, Dosen Pengajar, Sekretaris Kaprodi | Melakukan aktivitas *posting* *thread* dan *comment* di *forum* | Terjadinya *knowledge* *sharing* pada *forum* |
| 3 | Ketua Kelompok Keahlian | Memonitor aktivitas di setiap fitur aplikasi | Ketua kelompok keahlian mengetahui aktivitas yang terjadi pada aplikasi |

### IV.2.2 Tahap Elaborasi

Pada fase ini menggambarkan sistem untuk memenuhi persyaratan nonfungsional. Bisa dilaksanakan dalam dua iterasi atau lebih. Artefak yang dapat dihasilkan adalah visi yang diperjelas, *use case* awal, deskripsi awal kasus pengembangan untuk menentukan proses yang digunakan, dan lain lain (Larman, 2005).

### IV.2.2.1 *Use Case Diagram*

 Gambar IV.26 *Use Case Diagram* *Increment* Kedua

Berikut *use case description* yang menjelaskan detail *use case diagram* pada *increment* kedua:

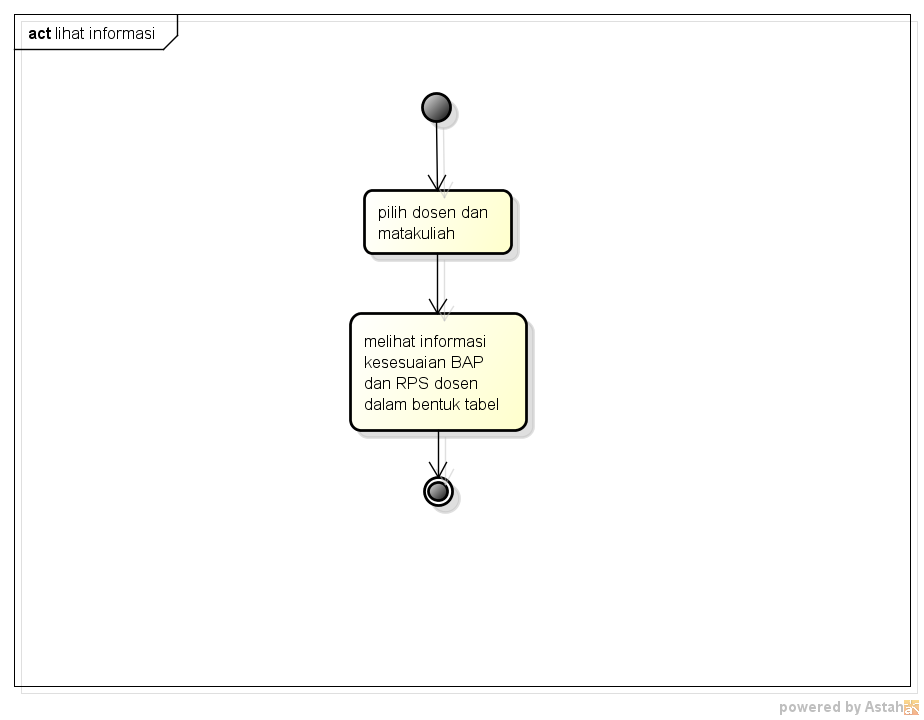
Tabel IV.8 *Use Case Form Increment* Kedua

|  |  |
| --- | --- |
| ID *Use Case* | 6 |
| Nama *Use Case* | Melihat informasi dalam bentuk tabel |
| Aktor | 1. Ketua KK 2. Dekan 3. Wakil Dekan 4. Kaprodi 5. Dosen Koordinator |
| Deskripsi | Aktordapat melihat keseluruhan informasi mengenai proses verifikasi kesesuaian BAP dengan RPS dosen |
| Pre-kondisi | 1. Sudah *Login.* 2. Sudah ada data yang tercatat pada aplikasi. |
| Post-kondisi | Aktor sudah melihat informasi mengenai validasi kesesuaian BAP dan RPS setiap dosen. |
| Alur | 1. Memilih menu lihat data 2. Memilih dosen dan matakuliah yang diampu |
| Alur alternatif |  |
| *Exceptions* |  |
| *Includes* |  |

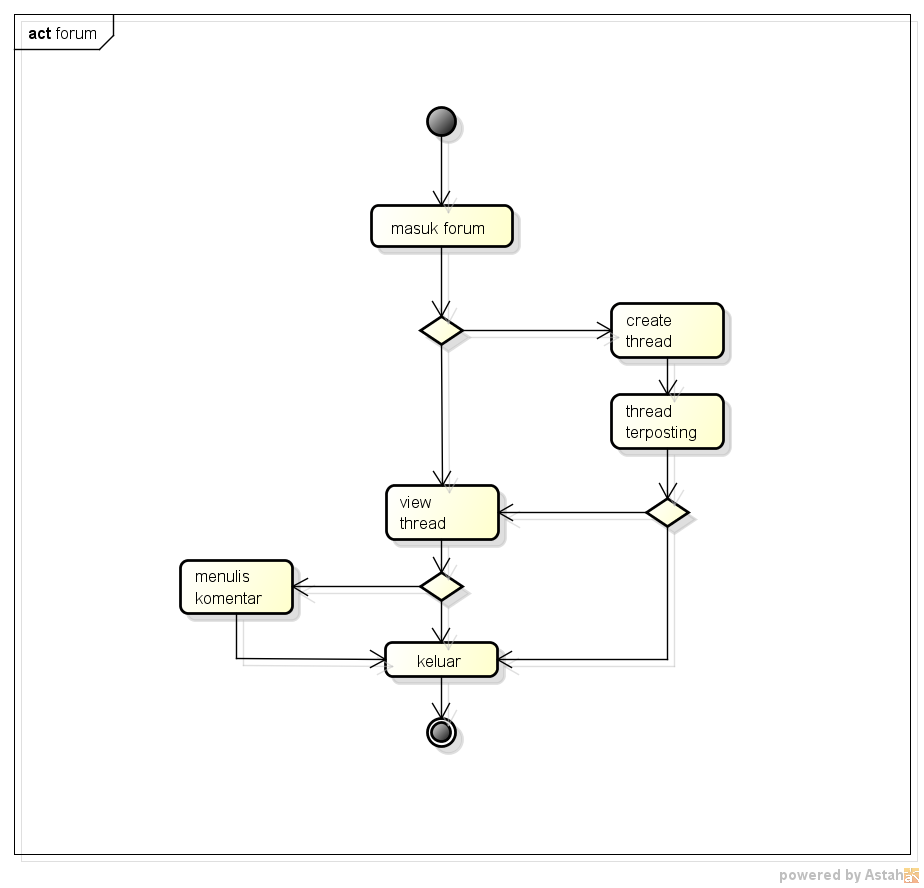
|  |  |
| --- | --- |
| ID *Use Case* | 7 |
| Nama *Use Case* | Mengelola *forum* |
| Aktor | 1. Dosen pengajar 2. Ketua Kelompok Keahlian 3. Dosen Koordinator 4. Dekanat 5. Sekretaris kaprodi |
| Deskripsi | Aktor dapat melakukan interaksi didalam forum sebagai sarana *sharing knowledge.* |
| Pre-kondisi | 1. Sudah *Login.* 2. Sudah memilih menu *forum* |
| Post-kondisi | 1. Aktor telah melakukan interaksi melalui *forum* 2. Aktivitas di forum tersimpan di *database* |
| Alur | 1. Memilih menu *forum*. 2. Melakukan interaksi seperti *create* *thread* dan *comment* di dalam *forum*. |
| Alur alternatif |  |
| *Exceptions* |  |
| *Includes* |  |

### IV.2.2.2 *Activity* *Diagram*

Berikut *activity* *diagram* yang menjelaskan alur aktivitas yang dilakukan aktor terkait *increment* kedua:



Gambar IV.27 *Activity Diagram* Melihat Informasi dalam Bentuk Tabel



Gambar IV.28 *Activity Diagram* Mengelola *Forum*

### IV.1.2.3 *Class Diagram*

### IV.1.2.4 *Sequence Diagram*

Berikut merupakan *sequence* *diagram* pada *increment* kedua:



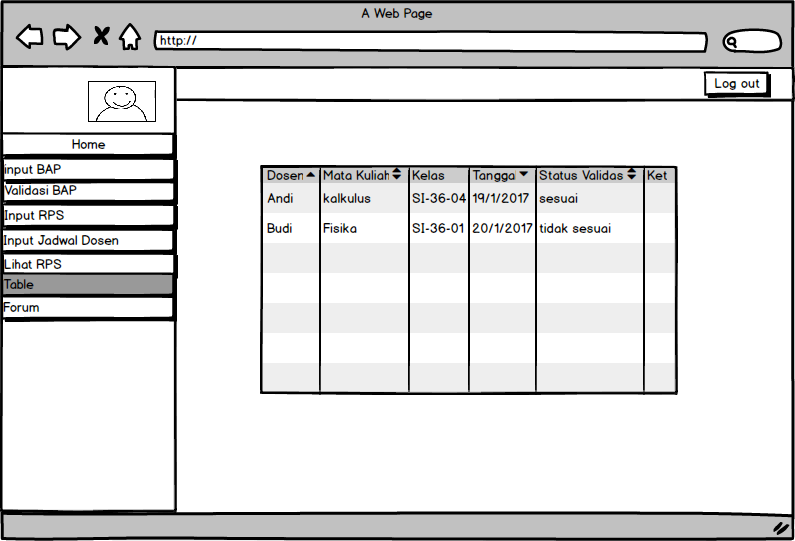
Gambar IV.29 *Sequence* *Diagram* Melihat Informasi dalam Bentuk Tabel



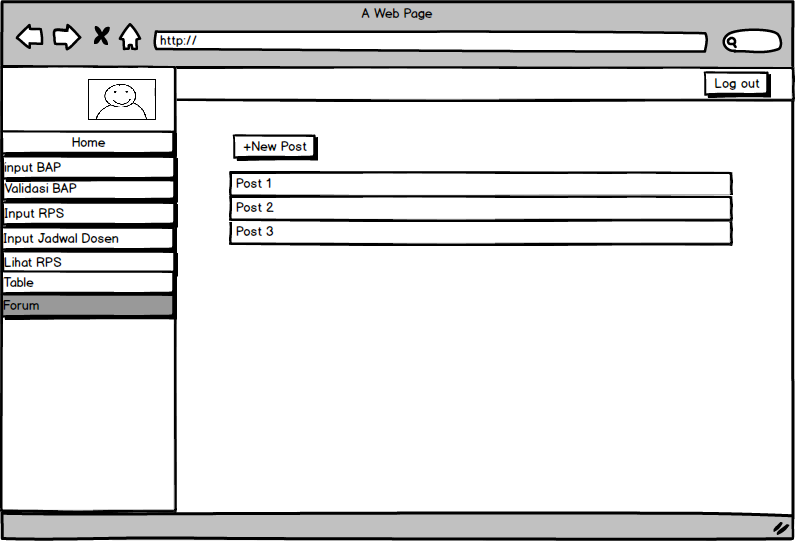
Gambar IV.30 *Sequence* *Diagram* Mengelola *Forum*

### IV.1.2.5 Desain *Mockup*

Berikut merupakan desain *mockup* pada *increment* kedua:



Gambar IV.31 Desain *Mockup* Melihat Informasi dalam Bentuk Tabel



Gambar IV.32 Desain *Mockup* Mengelola *Forum*

### IV.2.3 Tahap Kontruksi

Dilakukan secara bertahap dengan cara mengisi sebuah perangkat lunak yang awalnya masih berbentuk rancangan arsitektur dengan kode produksi yang telah siap yang dihasilkan dari analisis, desain, implementasi, dan pengujian persyaratan fungsional untuk didistribusikan ke pengguna. Sama seperti fase sebelumnya, fase ini bisa dilaksanakan dalam dua atau lebih iterasi (Larman, 2005).

### IV.2.3.1 Implementasi *File*

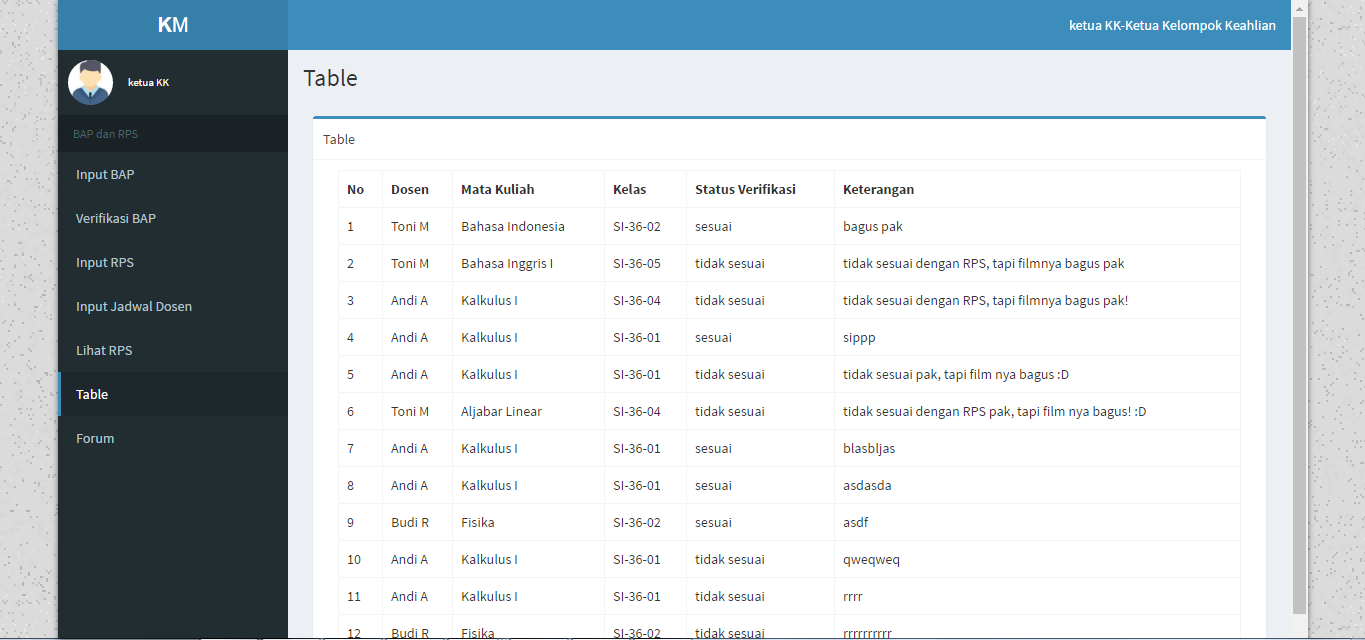
Pada tahap ini dijelaskan komponen-komponen yang digunakan untuk membangun aplikasi validasi kesesuaian BAP dengan RPS dosen. Komponen tersebut merupakan class-*class* yang terdapat di aplikasi. Berikut nama *class* dan deskripsinya:

Tabel IV.9 Implementasi *File* *Increment* Kedua

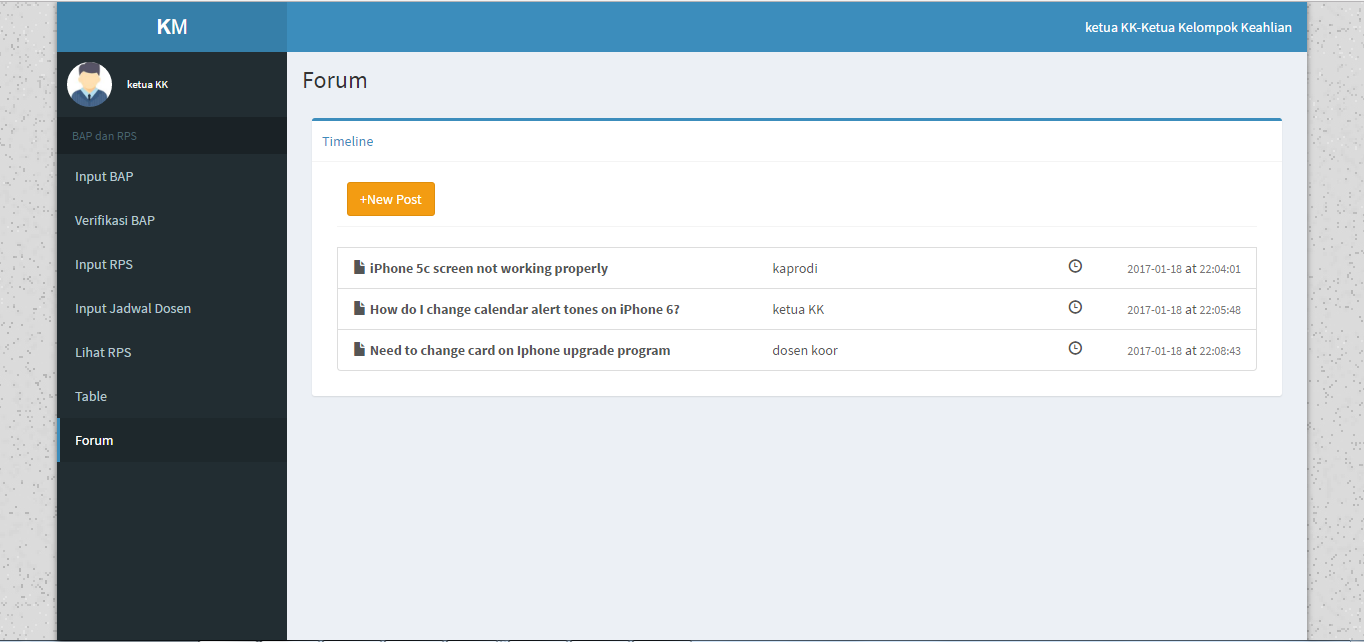
|  |  |  |  |
| --- | --- | --- | --- |
| No | MVC | Nama Class | Deskripsi |
| 1 | *Controller* | Baprps.php | *File* *controller* yang berfungsi untuk menampung seluruh *method*/fungsi dari seluruh aktivitas pada aplikasi validasi kesesuaian BAP dengan RPS kecuali aktivitas *login* |
| 2 | *View* | forum.php | *File* *view* yang berfungsi untuk menampilkan halaman utama *forum* |
| 3 | comment.php | *File* *view* yang berfungsi untuk menampilkan detail *thread* dan kolom *comment* |
| 4 | newpost.php | *File* *view* yang berfungsi untuk menampilkan *form* *add* *new* *post* |
| 5 | tabel.php | *File* *view* yang berfungsi untuk menampilkan hasil validasi kesesuaian BAP dengan RPS dosen |
| 6 | Model | m\_inputbap.php | *File* *model* yang berisikan pengelolaan data ke *database* seperti *insert*, *update*, *select* dan *delete* pada *database* khusus untuk *input* BAP dosen dan *forum* |
| 7 | m\_inputrps.php | *File* *model* yang berisikan pengelolaan data ke *database* seperti *insert*, *update*, *select* dan *delete* pada *database* khusus untuk *input* RPS dosen |
| 8 | m.verif.php | *File* *model* yang berisikan pengelolaan data ke *database* seperti *insert*, *update*, *select* dan *delete* pada *database* khusus untuk validasi kesesuaian BAP dengan RPS dosen |

### IV.2.3.2 Tampilan Program

Berikut merupakan tampilan aplikasi terkait *increment* kedua:



Gambar IV.33 Tampilan Halaman Tabel



Gambar IV.34 Tampilan Halaman *Forum*

### IV.1.4 Tahap Transisi

### IV.1.4.1 Pengujian

Berikut merupakan hasil dari pengujian yang dilakukan oleh *developer* dari aplikasi pada *increment* kedua :

Tabel IV.10 Daftar Hasil Pengujian *Increment* Kedua

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Kode Pengujian | Prosedur Pengujian | Hasil yang Diharapkan | Hasil |
| 1 | Req 1-1 | Membuka menu Tabel | Menampilkan tabel hasil validasi kesesuaian BAP dengan RPS |  |
| 2 | Req 2-1 | Membuka menu *forum* | Menampilkan halaman utama *forum* |  |
| 3 | Req 2-2 | Menambahkan *thread* baru | *Thread* baru muncul di halaman utama *forum* |  |
| 4 | Req 2-3 | Menambahkan *comment* | *Comment* yang ditambahkan tampil di *thread* yang bersangkutan |  |
| 14 | Req 3-1 | Membuka seluruh menu yang tesedia di aplikasi | Menampilkan seluruh tampilan dari masing-masing menu yang dibuka |  |