**BAB III**

**ANALISIS DAN PERANCANGAN**

**3.1 Analisis**

Analisis merupakan langkah awal untuk pengembangan sebuah aplikasi, karena perancangan dan bahkan pengembangan implementasi aplikasi tidak akan berjalan dengan baik tanpa adanya analisa terhadap aplikasi yang digunakan. Analisis juga dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponen-komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi masalah-masalah, kesempatan-kesempatan, hambatan-hambatan yang terjadi serta kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan agar mendapat hasil yang maksimal.

**3.1.1 Analisis Sistem yang sedang berjalan**

Proses pendaftaran program semester pendek di Program Studi D4 Teknik Informatika dilakukan secara manual, yaitu mahasiswa datang ke BAAK lalu pegawai BAAK akan memberikan formulir pendaftaran kepada mahasiswa dan setelah mahasiswa mengisi formulir, formulir dikembalikan kepada BAAK. Dan pegawai BAAK akan melakukan rekap dari formulir tersebut dan melakukan *approve* bagi mahasiswa yang telah mendaftar program semester pendek.

**3.1.1.1 Analisis Prosedur / Flow Map yang sedang berjalan**

Berikut ini merupakan analisis sistem yang sedang berjalan dalam bentuk flowmap atau prosedur pendaftaran program semester pendek di Program Studi D4 Teknik Informatika:



Gambar 3.1 *Flowmap* Sistem yang Sedang Berjalan

**3.1.1.2 Analisis Dokumen yang digunakan**

Analisis dokumen yang digunakan dalam pendaftaran semester pendek Program Studi D4 Teknik Informatika ini adalah data *dummy* yang dibuat oleh penulis.

**3.1.2 Analisis Sistem yang akan dibangun**

Sistem ini akan mengubah proses pendaftaran semester pendek Program Studi D4 Teknik Informatika manual menjadi sebuah proses yang berbasis web. Sistem ini bisa diakses oleh semua aktor mulai dari admin, BAAK, Prodi, BAUK dan mahasiswa dengan penentuan levelisasi hak akses *user* terlebih dahulu.



Gambar 3.2 *Flowmap* Sistem yang Akan Dibuat

**3.1.2.1 Analisis Kebutuhan Aplikasi**

Sistem ini akan memerlukan atribut yang berkaitan dengan penginputan data mahasiswa dan matakuliah semester pendek. Setiap *user* memiliki hak akses yang berbeda-beda dalam pengaksesan sistem. Untuk admin, memiliki hak akses untuk mengolah semua bagian *website*. BAAK untuk melakukan *approval* data. Dan Prodi berhak untuk memberikan pengumuman matakuliah semester pendek. Mahasiswa juga bisa dengan mudah mengoperasikan sistem ini sesuai dengan batasan-batasan yang terlebih dahulu telah ditentukan.

**3.1.2.2 Analisis Kebutuhan Perangkat Lunak dan Perangkat Keras**

Sistem Informasi Semester Pendek ini adalah aplikasi untuk memudahkan admin menginputkan data mahasiswa yang akan melakukan semester pendek, membuat data tersusun rapi, dan memudahkan baak dalam memberi *approve* pada sistem semester pendek. Sistem ini juga dapat mengefisiensikan waktu dan tenaga pengguna aplikasi ini.

Sebelumnya sistem yang digunakan untuk pendaftaran ini masih manual sehingga dalam pembuatan sistem yang akan dibangun, digunakan beberapa perangkat lunak seperti:

1. Sistem Operasi :

* *Windows* 7

1. Perangkat Lunak (*software)* :

* *YII* *Framework* 2.0

1. Database :

* *XAMPP v3.2.2*
* *MySql 5.0.11*

Untuk mendukung perangkat lunak yang digunakan cukup dengan menggunakan perangkat keras dengan spesifikasi sederhana yaitu:

a. Processor : Intel Core-i5

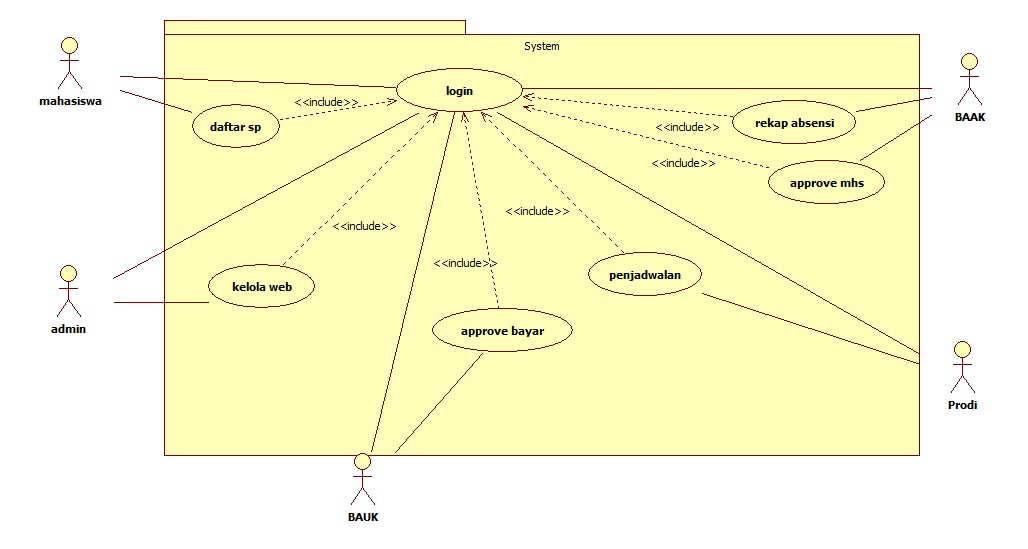
b. Harddisk : 500GB

c. VGA/Memory : DDR3 4GB

**3.2 Perancangan**

Perancangan merupakan penjelasan perancangan sistem/aplikasi/alat yang akan dibuat terdiri dari perancangan alir program (*Flow* *Map*), algoritma, data, maupun perancangan *input/output* sistem/aplikasi/alat.

**3.2.1 Use Case Diagram**

Merupakan pemodelan untuk menggambarkan kelakuan (*behavior*) sistem yang akan dibuat. Diagram ini mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat.

Gambar 3.3 *Use Case Diagram* Sistem Informasi Semester Pendek

**3.2.1.1 Definisi Aktor**

Aktor, menetukan peran yang dimainkan oleh user atau sistem lain yang berinteraksi dengan subjek. Aktor adalah segala sesuatu yang berinteraksi dengan sistem aplikasi komputer, seperti orang, benda atau lainnya. Tugas aktor memberikan informasi kepada sistem dan dapat memerintahkan sistem untuk melakukan tugas.[6] Adapun deskripsi dari aktor yang terlibat dalam Sistem Informasi Pendaftaran Semester Pendek Prodi D-IV Teknik Informatika ini adalah sebagai berikut:

|  |  |  |
| --- | --- | --- |
| **No** | **Aktor** | **Deskripsi** |
| 1. | Admin | Merupakan pihak yang dapat mengakses seluruh sistem |
| 2. | Mahasiswa | Merupakan pihak yang dapat masuk ke sistem namun hanya dapat masuk ke menu pendaftaran sp |
| 3 | BAAK | Merupakan pihak yang dapat masuk ke sistem dan *approve* data sp mahasiswa yang akan sp |
| 4 | Prodi | Merupakan pihak yang dapat masuk ke sistem dan melakukan penjadwalan serta pengumuman |
| 5 | BAUK | Merupakan pihak yang dapat melakukan *approve* pembayaran |

Tabel 3.1 Definisi Aktor

**3.2.1.2 Definisi *Use Case Diagram***

*Use case* yang ada dalam diagram didefinisikan pada tabel di bawah ini:

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **No Definisi** | **Use Case** | **Keterangan** |
| 1. | UC-01 | Login | Merupakan proses semua yang ada di sisem |
| 2. | UC-02 | Kelola *Web* | Merupakan proses mengolah web |
| 3. | UC-03 | Daftar sp | Merupakan proses pendaftaran sp yang dilakukan oleh *user*/mahasiswa |
| 4. | UC-04 | *Approve* mhs | Merupakan proses *approve* data mahasiswa yang dilakukan oleh baak. |
| 5. | UC-05 | Penjadwalan | Merupakan proses penjadwalan matakuliah dan dosen semester pendek oleh prodi. |
| 6 | UC-06 | *Approve* bayar | Merupakan proses *approve* bukti pembayaran sp mahasiswa |
| 7 | UC-07 | Rekap Absensi | Merupakan proses pembuatan rekap absensi semester pendek |

Tabel 3.2 *Definisi* *Use Case*

**3.2.1.3 Skenario Use Case Diagram**

Skenario untuk masing-masing *use case* dari sistem informasi pendaftaran semester pendek ini adalah:

1. **Skenario Use Case Login**

Skenario dalam *use case* input data pada tabel di bawah ini:

|  |  |
| --- | --- |
| Identifikasi | |
| Nomor | UC-01 |
| Nama | Login |
| Tujuan | Melakukan login untuk masuk halaman utama |
| Deskripsi | Login dengan memasukkan *user*name dan pssword |
| Aktor | Admin,BAAK,Prodi,Mahasiswa, dan BAUK |
| Skenario | |
| Kondisi Awal | Menampilkan form login |
| Aksi Aktor | Reaksi Sistem |
| 1. Aktor meminta form login pada sistem | 1. Menampilkan form login |
| 1. Memasukkan *user*name dan pssword | 1. Melakukan validasi, jika salah memasukkan password dan *user*name maka akan muncul pesan “*username is not correct”* |
|  | 1. Jika salah maka akan kembali ke form login. Dan jika benar maka akan masuk ke form input data |
| Kondisi Akhir | Menampilkan form halaman utama |

Tabel 3.3 *Skenario* *Use Case* *Login*

1. **Skenario Use Case Kelola Web**

|  |  |
| --- | --- |
| Identifikasi | |
| Nomor | UC-02 |
| Nama | Kelola Web |
| Tujuan | Pengelolaan web dan Penetapan hak akses *user* |
| Deskripsi | Aktivitas penetepan hak akses *user* dan pengelolaan web |
| Aktor | Admin |
| Skenario | |
| Kondisi Awal | Menampilkan form penetapan hak akses *user* dan semua menu web |
| Aksi Aktor | Reaksi Sistem |
| 1. Admin login | 1. Admin menginputkan data penetapan hak akses *user* dan mengelola web |
|  | 1. Sistem akan menyimpan hak akses di database |
| Kondisi Akhir | Menampilkan hak akses *user,* perubahan pada web |

Tabel 3.4 *Skenario* *Use Case* Kelola Web

1. **Skenario Use Case Daftar SP**

|  |  |
| --- | --- |
| Identifikasi | |
| Nomor | UC-03 |
| Nama | Daftar SP |
| Tujuan | Pengisian form SP oleh mahasiswa |
| Deskripsi | Aktivitas daftar SP |
| Aktor | Mahasiswa |
| Skenario | |
| Kondisi Awal | Menampilkan form sp |
| Aksi Aktor | Reaksi Sistem |
| 1. Mahasiswa mengisi form | 1. Sistem menampilkan form pendaftaran semester pendek |
| 1. Mahasiswa menyimpan data matakuliah yang telah dipilih | 1. *Submit* data dan laporan |
| Kondisi Akhir | Form akan disimpan di database |

Tabel 3.5 Skenario *Use Case* Daftar SP

1. **Skenario Use Case Approve Mhs**

|  |  |
| --- | --- |
| Identifikasi | |
| Nomor | UC-04 |
| Nama | *Approve* SP mhs |
| Tujuan | Menginputkan data sp dari mahasiswa |
| Deskripsi | Aktivitas BAAK untuk *approve* sp mahasiswa |
| Aktor | BAAK |
| Skenario | |
| Kondisi Awal | Menampilkan form approve |
| Aksi Aktor | Reaksi Sistem |
| 1. BAAK memilih mahasiswa yang telah mendaftar SP dan telah di-*approve* pembayarannya oleh BAUK | 1. Sistem menampilkan daftar mahasiswa yang telah di *approve* |
| 1. BAAK melakukan *input* data mahasiswa | 1. sistem akan menyimpan data kedalam database |
| Kondisi Akhir | Menampilkan data yang telah dimasukkan |

Tabel 3.6 Skenario *Use Case* *Approve* SP Mahasiswa

1. **Skenario Use Case Penjadwalan**

|  |  |
| --- | --- |
| Identifikasi | |
| Nomor | UC-05 |
| Nama | Penjadwalan |
| Tujuan | Memberikan laporan penjadwalan matakuliah dan daftar dosen semester pendek |
| Deskripsi | Aktivitas prodi memberikan laporan penjadwalan semester pendek |
| Aktor | Prodi |
| Skenario | |
| Kondisi Awal | Menampilkan form approve yang berisi data matakuliah dan nama dosen |
| Aksi Aktor | Reaksi Sistem |
| * + - 1. Prodi menginput nama dosen dan nama matakuliah yang akan di selenggarakan | * + - 1. Sistem akan melakukan *generate* laporan |
| Kondisi Akhir | Menampilkan laporan |

Tabel 3.7 Skenario *Use Case* *Approve* Penjadwalan

1. **Skenario Use Case Approve Bayar**

|  |  |
| --- | --- |
| Identifikasi | |
| Nomor | UC-06 |
| Nama | *Approve* Bayar |
| Tujuan | Melakukan *approve* pembayaran semester pendek yang telah dilakukan mahasiswa |
| Deskripsi | BAUK mengubah status *approve* mahasiswa yang telah melakukan pembayaran |
| Aktor | BAUK |
| Skenario | |
| Kondisi Awal | Menampilkan form *approve* yang berisi data mahasiswa yang telah dan yang belum membayar |
| Aksi Aktor | Reaksi Sistem |
| * + - 1. BAUK mengubah status *approve* mahasiswa | * + - 1. Sistem akan melakukan perubahan status dan menyimpan data |
| Kondisi Akhir | Perubahan status *approve* |

Tabel 3.8 Skenario *Use Case* *Approve* Bayar

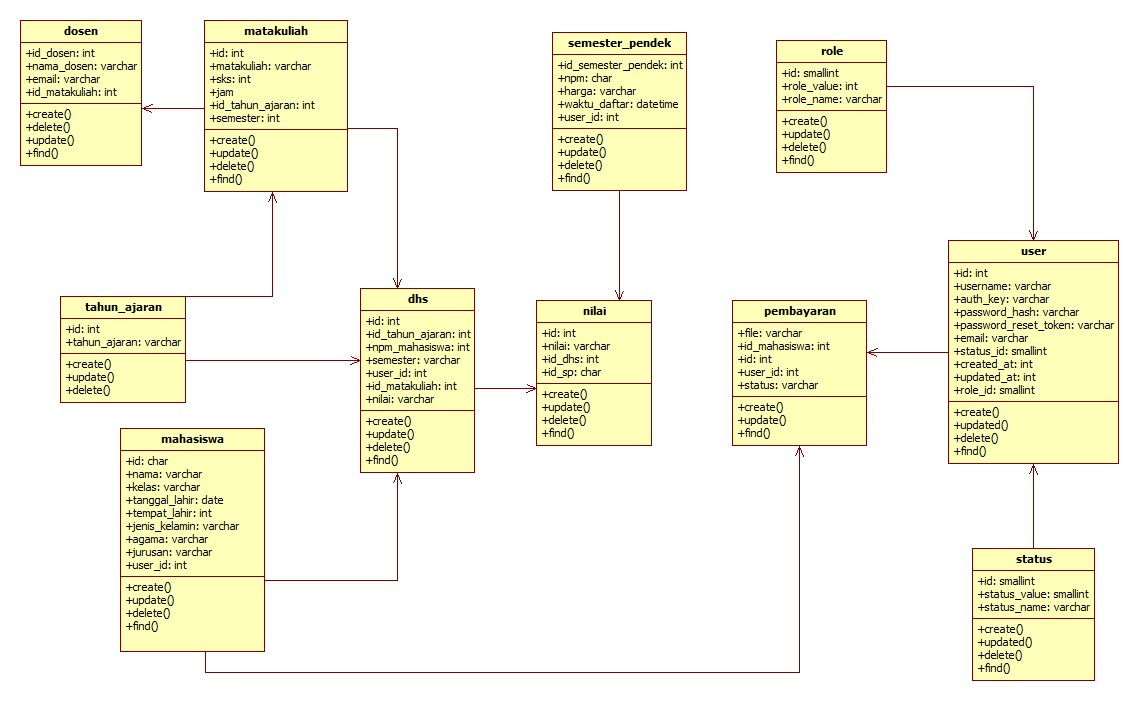
1. **Skenario Use Case Rekap Absensi**

|  |  |
| --- | --- |
| Identifikasi | |
| Nomor | UC-07 |
| Nama | Rekap Absensi |
| Tujuan | Melakukan pembuatan rekap absensi |
| Deskripsi | BAAK akan melakukan rekap absensi semester pendek yang diselenggarakan |
| Aktor | BAAK |
| Skenario | |
| Kondisi Awal | Menampilkan form rekap absensi yang berisi data mahasiswa yang telah mengikuti semester pendek |
| Aksi Aktor | Reaksi Sistem |
| * + - 1. BAAK akan memilih mahasiswa yang mengikuti semester pendek berdasarkan matakuliah yang diikuti | * + - 1. Sistem akan melakukan *generate* laporan rekap absensi |
| Kondisi Akhir | Menampilkan laporan rekap absensi |

Tabel 3.9 Skenario *Use Case* Rekap Absensi

**3.2.2 *Class Diagram***

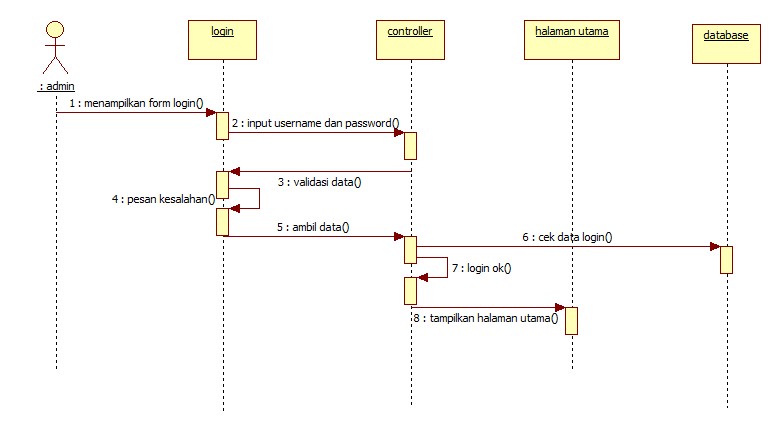
*Class* *Diagram* menggambarkan struktur statis dari kelas dalam sistem anda dan menggambarkan atribut, operasi dan hubungan antara kelas. *Class* diagram membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. Selama tahap desain, class diagram berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat. *Class* memiliki tiga area pokok yaitu: nama (dan stereotype), atribut, metoda.*Class* adalah sebuah spesifikasi yang jika diinstalasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode/fungsi). Berikut ini adalah *class* diagram Sistem Informasi Semester Pendek Program Studi D-IV Teknik Informatika.

****

Gambar 3.4 *Class Diagram* Sistem Informasi Semester Pendek

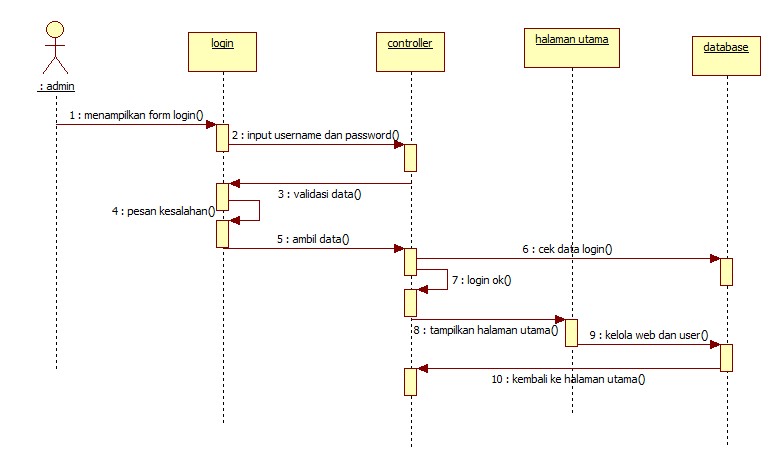
**3.2.3 Sequence Diagram**

Diagram sekuen menggambarkan perilaku objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Banyaknya diagram sekuen yang harus digambar adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

* + - 1.  ***Sequence* *Diagram* *Login Admin***

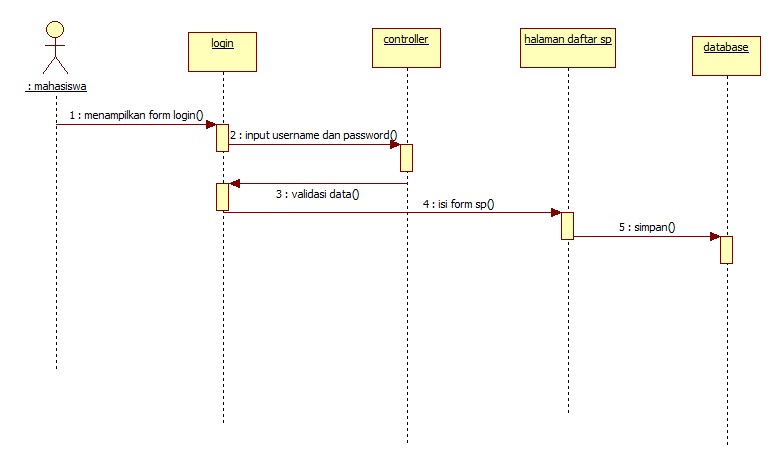
Gambar 3.5 *Sequence Diagram* *Login Admin*

Pada proses ini admin melakukan login ke dalam sistem dengan memasukkan *user*name dan *password* yang benar. Setelah masuk ke sistem, *admin* dapat mengakses seluruh sistem. Sedangkan untuk *user* biasa, setelah masuk ke sistem *user* hanya bisa mengakses beberapa menu saja.

* + - 1. ** *Sequence Diagram* Kelola Data *Web***

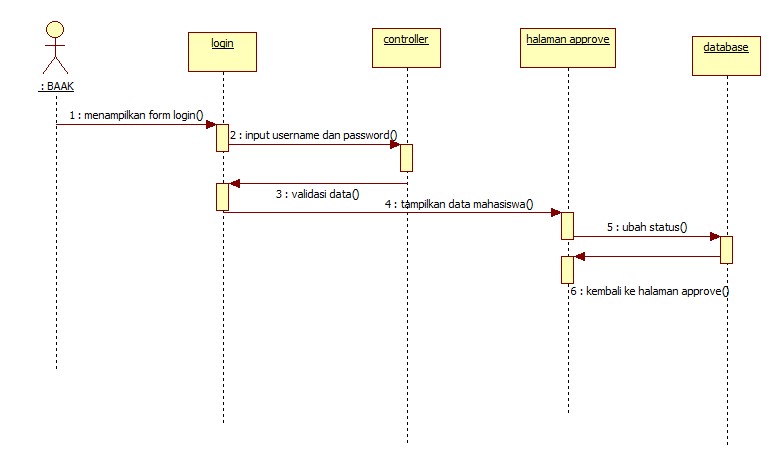
Gambar 3.6 *Sequence Diagram* Kelola Data *Web*

Pada proses ini admin telah masuk ke dalam sistem. Setelah masuk ke sistem, *admin* akan melakukan kelola web yang termasuk juga pengelolaan hak akses user dan juga pengelolaan menu pada web.

* + - 1. ***Sequence Diagram* Mahasiswa Daftar SP**

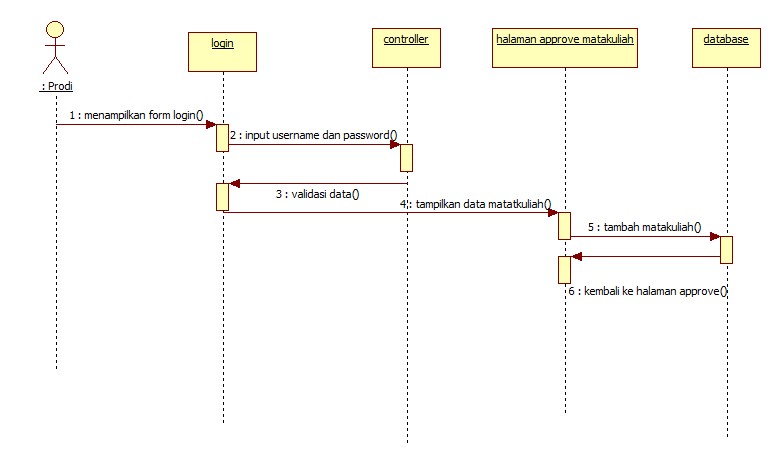
Gambar 3.7 *Sequence Diagram* Mahasiswa Daftar SP

Pada menu ini, mahasiswa mengisi *form* untuk pendaftaran Semester Pendek dengan mengisi data sesuai *field* yang tersedia pada *form* tersebut.

* + - 1. ***Sequence Diagram Approve* SP**

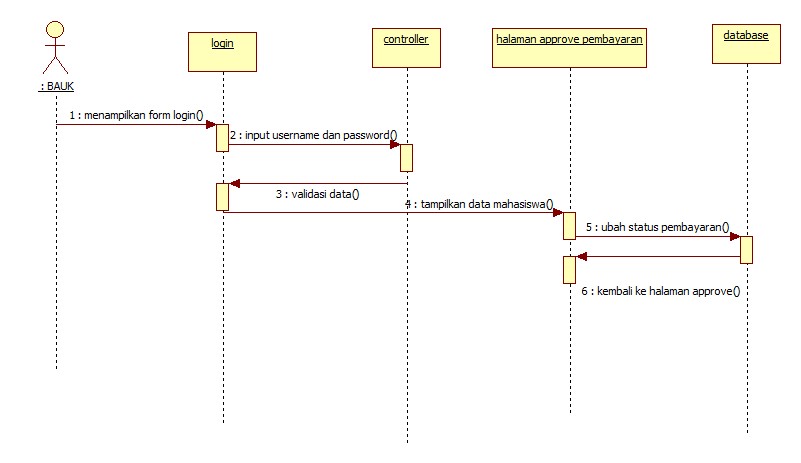
Gambar 3.8 *Sequence Diagram* *Approve* SP

Pada menu *approve* sp ini , BAAK login untuk melakukan *approve* SP mahasiswa yang telah mendaftar semester pendek sebelumnya.

* + - 1. ***Sequence Diagram Approve* Matakuliah SP**

Gambar 3.9 *Sequence Diagram Approve* MatakuliahSP

Proses ini dilakukan oleh prodi yang telah *login* ke sistem untuk melakukan *approve* matakuliah yang akan segera di SP.

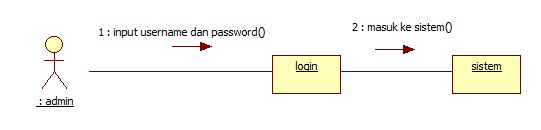
* + - 1. *** Sequence Diagram Approve* Bayar**

Gambar 3.10 *Sequence Diagram Approve* Bayar

Proses ini dilakukan oleh bauk yang telah *login* ke sistem untuk melakukan *approve* pembayaran semester pendek. Dalam hal ini, bauk akan merubah status mahasiswa yang telah sesuai dengan ketentuan pembayaran semester pendek yang berlaku, dengan merubah status yang tadinya “Belum di Approve”, menjadi “Telah di Approve”.

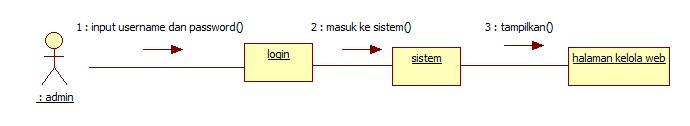
**3.2.4 Collaboration Diagram**

Diagram kolaborasi mengelompokkan *message* pada kumpulan diagram sekuen menjadi sebuah diagram. *Collaboration diagram* juga menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message*. Setiap *message* memiliki *sequence number*.

**3.2.4.1 *Collaboration Digram* Login**

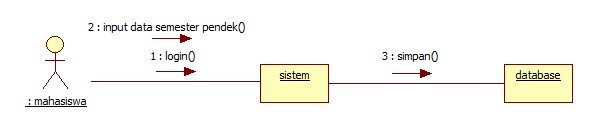
Gambar 3.11 *Collaboration Diagram* *Login*

Admin akan *login* terlebih dahulu sebelum masuk ke sistem. Setelah masuk ke sistem akan ada tampilan menu utama.

**3.2.4.2 *Collaboration Diagram* Kelola *Web***

Gambar 3.12 *Collaboration* *Diagram* Kelola *User*

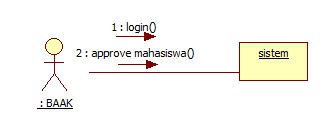
Pada proses ini admin telah masuk ke sistem dan akan masuk pada halaman kelola web dan kelola user *user* serta mengatur hak akses *user*.

**3.2.4.3 Collaboration *Diagram* Daftar SP**

Gambar 3.13 *Collaboration* *Diagram* DaftarSP

Pada proses ini mahasiswa *login* ke sistem untuk masuk ke *form* *input* data pendaftaran semester pendek dan akan disimpan di *database*.

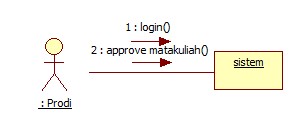
**3.2.4.4 *Collaboration* *Diagram* *Approve* SP**



Gambar 3.14 *Collaboration Diagram Approve* SP

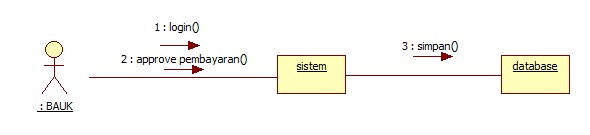
Pada proses ini BAAK *login* ke sistem dan melakukan *approve* mahasiswa yang akan telah mendaftar Semester Pendek dan akan disimpan di *database*.

**3.2.4.5 *Collaboration Diagram Approve* Matakuliah**

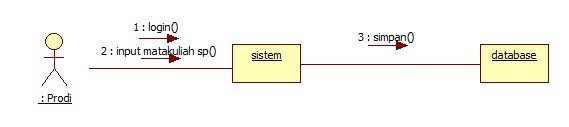
****

Gambar 3.15 *Collaboration Approve* Matakuliah

Pada proses ini prodi *login* ke sistem dan melakukan *approve* terhadap matakuliah yang akan di Semester Pendek yang telah di setujui oleh BAAK dan akan disimpan di *database*.

**3.2.4.6 *Collaboration Diagram Approve* Bayar**

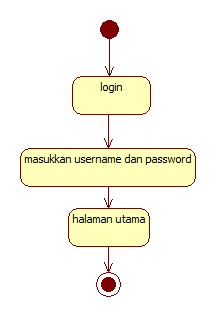
Gambar 3.16 *Collaboration Diagram Approve* Bayar

**3.2.4.7 *Collaboration Diagram* Penjadwalan**

Gambar 3.17 *Collaboration Diagram* Penjadwalan

**3.2.5 *Statechart Diagram***

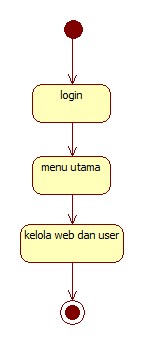
Diagram status digunakan untuk menyatakan kondisi (status) sebuah objek pada saat sistem informasi berjalan. Diagram ini diadopsi dari penggambaran kondisi mesin status (*state machine*) yang menggambarkan status apa saja yang dialami oleh mesin.

**3.2.5.1 *Statechart Diagram* *Login***

Gambar 3.18 *Statechart Diagram Login*

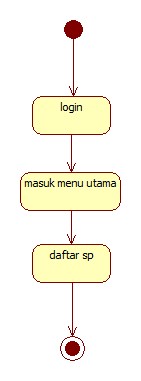
State diagram ini menjelaskan bahwa admin harus *login* terlebih dahulu dengan memasukkan *username* dan *password* yang benar. Setelah *login* dan masuk ke sistem *admin* akan menemukan tampilan menu utama.

**3.2.5.2 Statechart *Diagram* Kelola *Web* dan *User***



Gambar 3.19 *Statechart* *Diagram* Kelola *User*

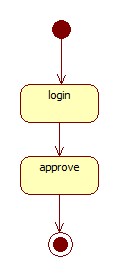
Proses Statechart Diagram Kelola Data User ini adalah admin *login* dan sistem menampilkan menu utama, *admin* akan mengatur hak akses *user* serta melakukan pengelolaan website.

**3.2.5.3 *Statechart Diagram* Daftar SP**

Gambar 3.20 *Statechart* *Diagram* Daftar SP

Proses dari Statechart Diagram Daftar SP ini adalah mahasiswa *login* ke dalam menu utama aplikasi, sistem akan menampilkan *form* data yang akan diisi untuk semester pendek dan akan disimpan ke *database*.

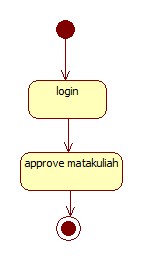
**3.2.5.4 *Statechart Diagram Approve* SP**



Gambar 3.21 *Statechart* *Diagram* *Approve* SP

Proses Statechart Diagram Approve SP ini BAAK akan melakukan *approve* mahasiswa yang akan melakukan semester pendek sesuai dengan aturan SP dan akan disimpan di *database*.

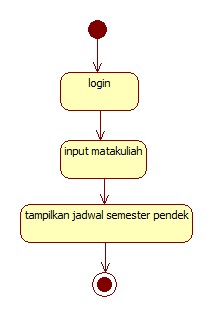
**3.2.5.5 *Statechart Diagram Approve* Matakuliah**



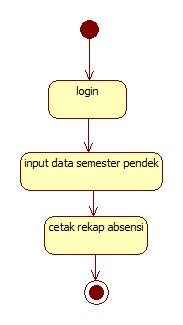
Gambar 3.22 *Statechart Diagram Approve* Matakuliah SP

Proses statechart Diagram *Approve* Matakuliah ini adalah :

1. Prodi login dengan menggunakan *username* dan *password*.
2. Prodi lalu masuk ke halaman utama dari *website* dan akan memilih menu *approve* matakuliah.

**3.2.5.6 *Statechart Diagram* Penjadwalan**

Gambar 3.23 *Statechart Diagram* Penjadwalan

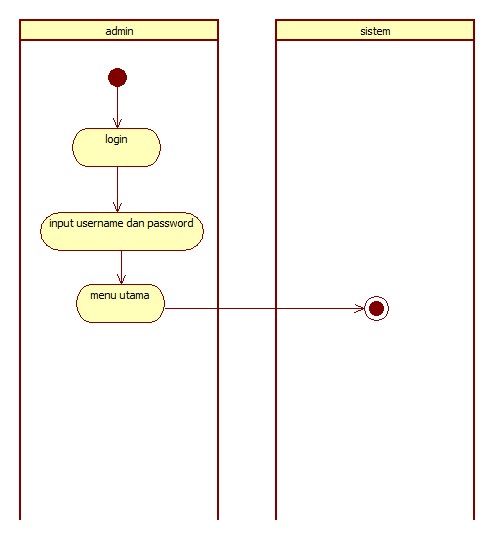
**3.2.5.7 *Statechart Diagram* Rekap Absensi**

Gambar 3.24 *Statechart Diagram* Rekap Absensi

**3.2.6 *Activity Diagram***

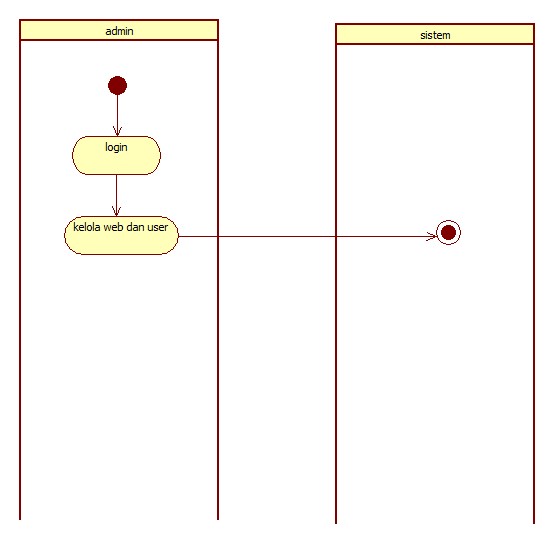
*Activity* *diagram* menggambarkan *workflow* suatu sistem atau proses bisnis. Diagram ini menunjukkan suatu alur kegiatan secara berurutan. Digram ini digunakan untuk mendeskripsikan kegiatan-kegiatan dalam sebuah operasi meskipun juga dapat digunakan untuk mendeskripsikan alur kegiatan yang lainnya seperti *use case* atau suatu interaksi.

**3.2.6.1 *Activity Diagram Login Admin***



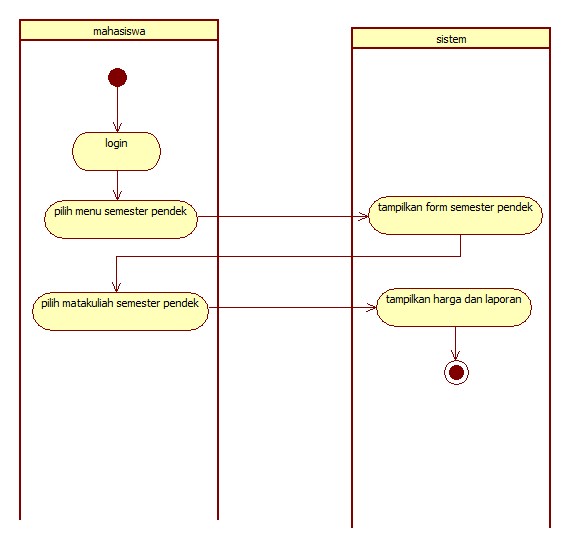
Gambar 3.25 *Activity Diagram Login* *Admin*

Pada proses ini, admin akan masuk ke sistem dengan menginputkan *username* dan *password* yang benar, setelah masuk ke sistem, tampilan awal yang akan dilihat adalah menu utama.

**3.2.6.2 *Activity Diagram* Kelola *Web* dan *User***

Gambar 3.26 *Activity* *Diagram* Kelola *Web* dan *User*

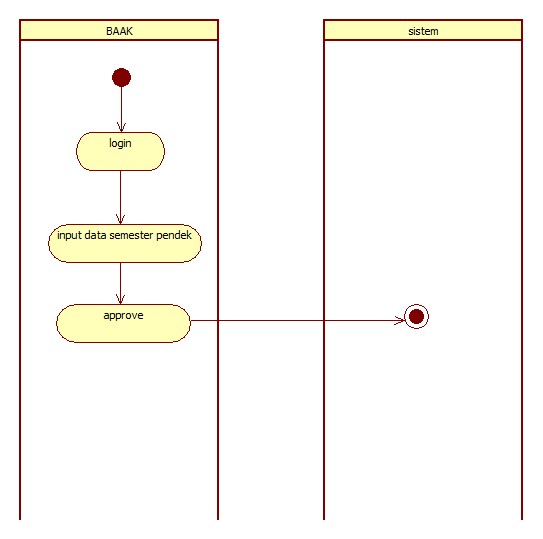
Pada proses ini admin akan melakukan kelola *user* dengan mengatur hak akses *user* yang akan mengatur hak akses dalam menu-menu tertentu pada aplikasi serta pengelolaan web semester pendek.

**3.2.6.3 *Activity* *Diagram* Daftar SP**

Gambar 3.27 *Activity Diagram* Daftar SP

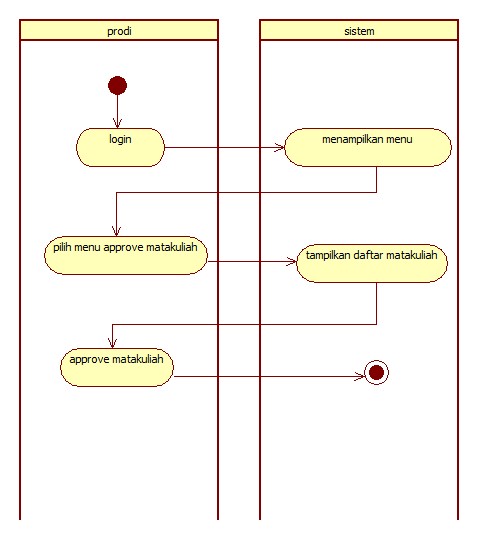
Pada proses ini mahasiswa *login* setelah melakukan *sign up* akan *login* lalu input data ke *form* semester pendek untuk melakukan daftar Semester Pendek dan di simpan di *database* dan menunggu *approve* dari BAAK.

**3.2.6.4 *Activity Diagram* Approve SP**



Gambar 3.28 *Activity Diagram Approve* SP

Pada proses ini BAAK memberikan *approve* pada pendaftaran semester pendek yang didaftar oleh mahasiswa dan akan tersimpan di *database*.

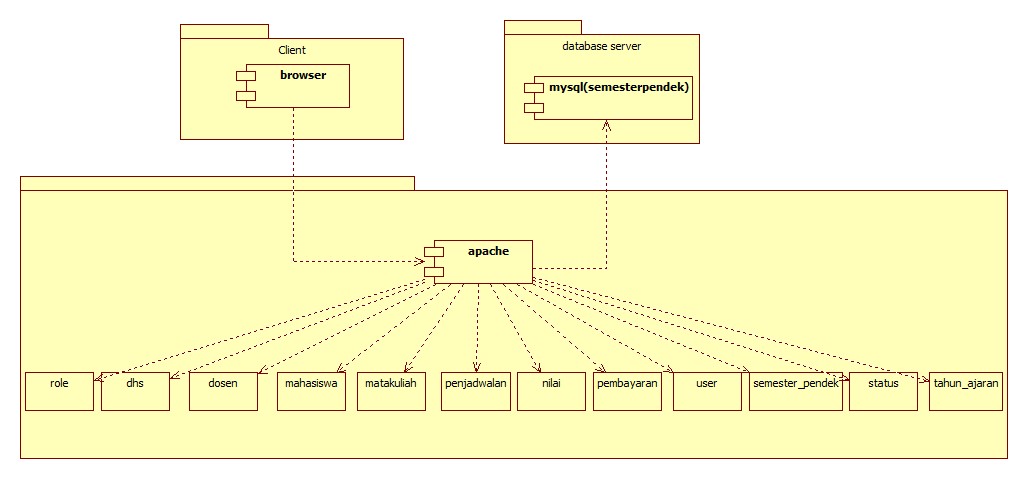
**3.2.6.4 *Activity Diagram* *Approve* Matakuliah**

Gambar 3.29 *Activity Diagram* *Approve* Matakuliah

Pada proses ini prodi akan login untuk melakukan *approve* terhadap matakuliah yang sudah di *approve* BAAK untuk pelaksanaan semester pendek.

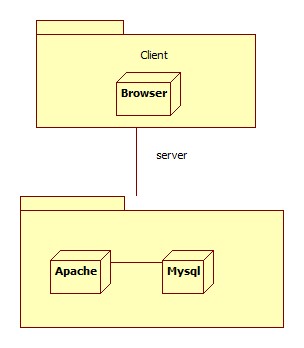
**3.2.7 *Component Diagram***

Komponen diagram digunakan untuk menunjukkkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem.



Gambar 3.*30 Component Diagram* Sistem Informasi Semester Pendek

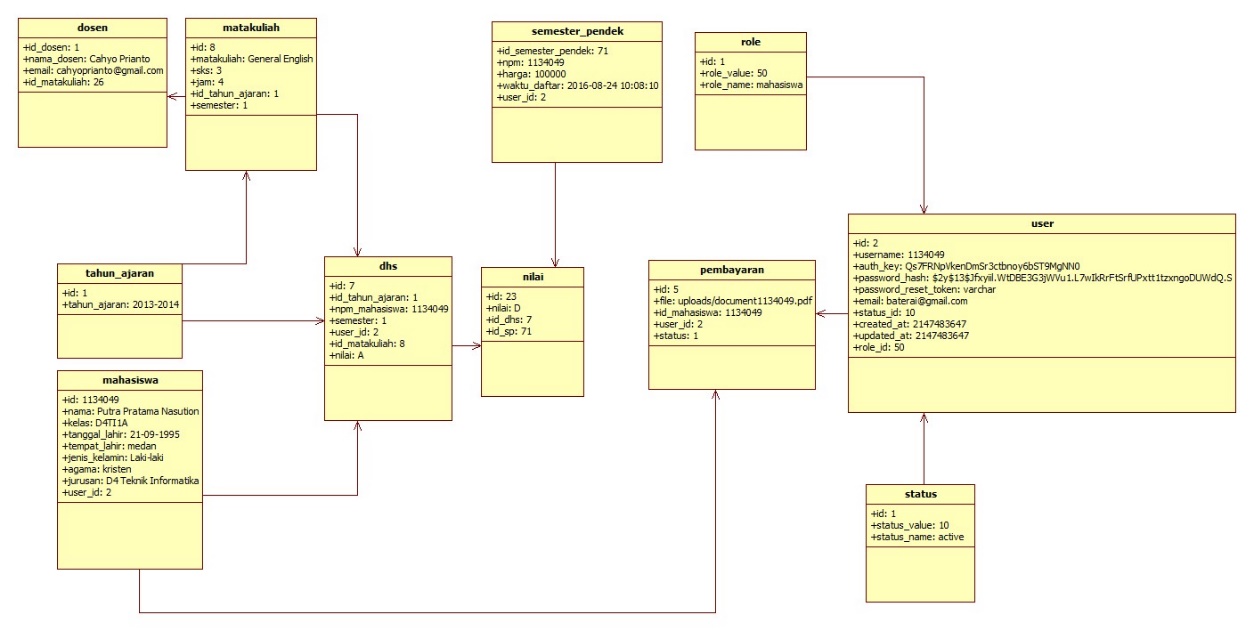
**3.2.8 *Deployment Diagram***

Deployment diagram menujukkan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram deployment juga dapat digunakan untuk memodelkan hal-hal seperti sistem tambahan (*embedded system*) yang menggambarkan rancangan *device*, *node*, dan *hardware*, sistem *client/server*, sistem terdistribusi murni, dan rekayasa ulang aplikasi.

Gambar 3.31 *Deployment Diagram* Sistem Informasi Semester Pendek

**3.2.9 Object Diagram**

Objek diagram adalah diagram yang memberikan gambaran struktur model sebuah sistem, dalam kurun waktu tertentu. Diagram objek berasal dari diagram kelas sehingga diagram objek tergantung pada diagram kelas. Objek diagram kadang-kadang disebut sebagai diagram *instance* dan sangat mirip dengan diagram kelas. Perbedaannya adalah bahwa diagram kelas merupakan model abstrak yang terdiri dari kelas dan hubungan mereka sedangkan diagram objek merupakan contoh pada saat tertentu yang konkret.



Gambar 3.32 *Object Diagram* Sistem Informasi Semester Pendek

* + 1. **Struktur Menu**

**ADMIN**

**MAIN KELOLA *ROLES* STATUS**

***USER***

**Mahasiswa**

**MAIN DAFTAR SEMESTER PENDEK PROFILE DHS**

Gambar 3.33 Struktur Menu

* + 1. **Perancangan Antarmuka**

**3.2.11.1 Halaman Login**

Gambar 3.34 Halaman *Login*

**3.2.11.2 Halaman *Form Input Data***

Create Mahasiswa

*User*\_ID

Nama\_Mhs

Kelas

Tanggal\_Lahir

Tahun\_Angkatan

Create

Gambar 3.36 *Form Input Data*