

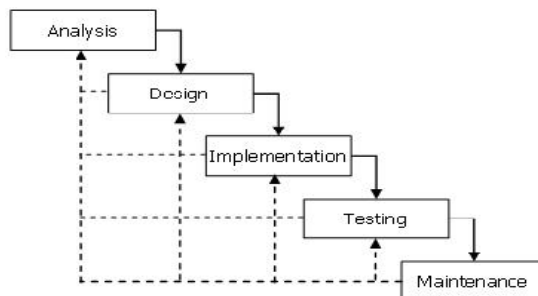
BAB III

METODE PENELITIAN

3.1 METODE PENGEMBANGAN SISTEM

Untuk pengembangan sistem penelitian ini menggunakan model SDLC (*Software Development Life Cycle*). *System Development Life Cycle* (SDLC) adalah proses pembuatan dan perubahan sistem serta model dan metodologi yang digunakan untuk mengembangkan sebuah sistem. SDLC juga merupakan pola yang diambil untuk mengembangkan sistem perangkat lunak, yang terdiri dari tahap-tahap: rencana (*planning*), analisis (*analysis*), desain (*design*), implementasi (*implementation*), uji coba (*testing*) dan pengelolaan (*maintenance*).

Model SDLC yang dipakai dalam penelitian ini adalah model *Waterfall*. *Waterfall Model* atau *Classic Life Cycle* merupakan model yang paling banyak dipakai dalam *Software Engineering* (SE). Menurut Bassil (2012) disebut *waterfall* karena tahap demi tahap yang harus dilalui menunggu selesainya tahap sebelumnya dan berjalan berurutan.



Gambar 3. 1 Metode Waterfall menurut Bassil (2012)

3.1.1 System / Information Engineering and Modeling

Permodelan ini diawali dengan mencari kebutuhan dari keseluruhan sistem yang akan diaplikasikan ke dalam bentuk *software*. Hal ini sangat penting, mengingat *software* harus dapat berinteraksi dengan elemen-elemen yang lain seperti *hardware*, *database*, dsb. Tahap ini sering disebut dengan *Project Definition*.

Pada tahap ini peneliti melakukan observasi dan wawancara langsung dengan pihak Biro Sistem Informasi (BSI) di Universitas Muhammadiyah Yogyakarta. Penjelasan lebih lengkap dari aktifitas tersebut terdapat pada bab 3.3. Pada Observasi ini terfokus pada permasalahan yang menjadi keluhan dari pihak *user*. Setelah mendapatkan beberapa data, peneliti menyimpan data tersebut sebagai bahan acuan dalam tahap analisis.

3.1.2 Software Requirements Analysis

Proses pencarian kebutuhan diintensifkan dan difokuskan pada *software*. Untuk mengetahui sifat dari *program* yang akan dibuat, maka para *software engineer* harus mengerti tentang domain informasi dari *software*, misalnya fungsi yang dibutuhkan, *user interface*, dsb. Dari 2 aktivitas tersebut (pencarian kebutuhan sistem dan *software*) harus didokumentasikan dan ditunjukkan kepada pelanggan.

Berdasarkan hasil observasi dan wawancara langsung, peneliti menganalisis data keluhan dari *user*. Apakah kebutuhan fungsi *software* untuk memenuhi kendala yang dialami oleh *user*. Sehingga didapatkan sebuah dokumen analisis

yang digunakan sebagai bahan untuk ditunjukkan kepada *user*. Penjelasan lebih lengkap dari aktifitas tersebut terdapat pada bab 3.4.

3.1.3 System Design

Proses ini digunakan untuk mengubah kebutuhan-kebutuhan diatas menjadi representasi ke dalam bentuk “*blueprint*” *software* sebelum coding dimulai. Desain harus dapat mengimplementasikan kebutuhan yang telah disebutkan pada tahap sebelumnya. Seperti 2 aktifitas sebelumnya, maka proses ini juga harus didokumentasikan sebagai konfigurasi dari *software*. Penjelasan lebih lengkap dari aktifitas ini terdapat pada bab 3.5.

Setelah peneliti mendapatkan dokumentasi dari hasil analisis, maka peneliti mengubah kebutuhan-kebutuhan fungsi *software* diatas menjadi sebuah bentuk “*blueprint*” *software*. Sehingga hasil desain akan digunakan oleh peneliti untuk membangun sebuah aplikasi.

3.1.4 Implementation

Untuk dapat dimengerti oleh mesin, dalam hal ini adalah komputer, maka desain tadi harus diubah bentuknya menjadi bentuk yang dapat dimengerti oleh mesin, yaitu ke dalam bahasa pemrograman melalui proses *coding*. Tahap ini merupakan implementasi dari tahap *design* yang secara teknis nantinya dikerjakan oleh *programmer*.

Pada tahap ini, peneliti membangun sebuah aplikasi berdasarkan desain “*blueprint*” yang telah dibuat. Pengembangan aplikasi ini dilakukan dari awal

hingga aplikasi siap dijalankan. Dari fungsi-fungsi yang dibutuhkan hingga tampilan untuk pengguna (dosen). Penjelasan lebih lengkap dari aktifitas ini terdapat pada lampiran.

3.1.5 Testing / Verification

Sesuatu yang dibuat haruslah diujicobakan. Demikian juga dengan *software*. Semua fungsi-fungsi *software* harus diujicobakan, agar *software* bebas dari *error*, dan hasilnya harus benar-benar sesuai dengan kebutuhan yang sudah didefinisikan sebelumnya.

Setelah proses pembangunan aplikasi selesai, peneliti melakukan pengujian pada tahap ini. Aplikasi diuji berdasarkan metode *black box* untuk mengetahui tingkat keberhasilan dari bagian sistem. Selain itu, peneliti juga melakukan pengujian secara langsung pada setiap sesi perkuliahan di UMY (khususnya di Jurusan Teknik Informatika). Penjelasan lebih lengkap dari aktifitas ini terdapat pada bab 4.2.

3.1.6 Maintenance

Pemeliharaan suatu *software* diperlukan, termasuk di dalamnya adalah pengembangan, karena *software* yang dibuat tidak selamanya hanya seperti itu. Ketika dijalankan mungkin saja masih ada *errors* kecil yang tidak ditemukan sebelumnya, atau ada penambahan fitur-fitur yang belum ada pada *software* tersebut. Pengembangan diperlukan ketika adanya perubahan dari eksternal perusahaan seperti ketika ada pergantian sistem operasi, atau perangkat lainnya.

Peneliti belum sampai pada tahap ini, sehingga tahap ini belum terlaksana. Rencana peneliti akan melakukan beberapa perbaikan tidak pada semua tahapan, namun hanya pada tahapan sebelum terjadi *error*. Sehingga peneliti tidak akan dipusingkan dengan melakukan tahapan dari awal hingga akhir kembali.

3.2 ALAT DAN BAHAN

3.2.1 Alat

Alat yang digunakan berupa perangkat keras dan perangkat lunak.

a. Perangkat Keras untuk Pengembang

Spesifikasi Laptop yang digunakan dapat dilihat pada Tabel 3.1.

Tabel 3. 1 Spesifikasi Laptop Pengembang

Sistem Operasi	Windows 10 Pro
Processor	Intel(R) Core(TM)i3-4010U CPU @ 1.70GHz
RAM	6 GB
Hard disk	500 GB

b. Perangkat Keras untuk *User* (Dosen)

Spesifikasi minimal perangkat komputer yang dapat digunakan oleh dosen untuk menjalankan aplikasi ini dapat dilihat pada Tabel 3.2.

Tabel 3. 2 Spesifikasi Minimum Perangkat Komputer *User* (Dosen)

Sistem Operasi	Windows 7
Processor	Intel(R) Pentium 4
RAM	1 GB
Hard disk	5 GB

c. Perangkat Keras untuk Server

Spesifikasi minimal perangkat komputer untuk server yang dapat digunakan untuk menjalankan aplikasi ini dapat dilihat pada Tabel 3.3.

Tabel 3. 3 Spesifikasi Minimum Perangkat Komputer *Server*

Sistem Operasi	Windows Server 2012
Processor	Intel(R) Core i3
RAM	2 GB
Hard disk	60 GB

d. Perangkat Lunak untuk Pengembang

Daftar perangkat lunak yang digunakan dapat dilihat pada Tabel 3.4.

Tabel 3. 4 Daftar Perangkat Lunak

No.	Perangkat Lunak
1	Microsoft Visual Studio 2015 Enterprise
2	SQL Server Management 2016
3	Browser Internet (Google Chrome atau Mozilla Firefox).

e. Perangkat Lunak untuk User (Dosen)

Daftar perangkat lunak yang digunakan dapat dilihat pada Tabel 3.5.

Tabel 3. 5 Perangkat Lunak Minimal untuk *User* (Dosen)

No.	Perangkat Lunak
1	Browser Internet (Google Chrome atau Mozilla Firefox).
2	NET Framework Minimal 3.5

f. Perangkat Lunak untuk Server

Daftar perangkat lunak yang digunakan dapat dilihat pada Tabel 3.6.

Tabel 3. 6 Perangkat Lunak Minimal untuk *Server*

No.	Perangkat Lunak
1	Browser Internet (Google Chrome atau Mozilla Firefox).
2	SQL Server Management 2016
3	Html 5, CSS 3 dan Javascript
4	Internet Information Services (IIS) 7

3.2.2 Bahan

Dalam Penelitian ini bahan yang digunakan adalah:

- a. Data yang diperoleh melalui studi literatur berdasarkan penelitian sebelumnya yang masih memiliki keterkaitan dengan aplikasi yang dikembangkan. Dari data yang diperoleh, maka didapatkan kebutuhan dari aplikasi pada saat pengembangan aplikasi.

- b. Database UMY (data hanya sebagian yang mendukung untuk kegiatan presensi) dari Biro Sistem Informasi (BSI) Universitas Muhammadiyah Yogyakarta.

3.3 METODE PENGUMPULAN DATA

3.3.1 Observasi

Tujuan dari observasi adalah dengan mendeskripsikan setting yang diamati, tempat kegiatan orang-orang yang berpartisipasi dalam kegiatan tersebut dan makna apa yang diamati menurut perspektif pengamat. Pengamatan dapat diklasifikasikan atas pengamatan melalui cara berperan serta dan yang tidak berperan serta.

Pada pengamatan berperan serta, pengamat melakukan dua peran sekaligus, yaitu sebagai pengamat dan sekaligus menjadi anggota penelitian dari kelompok penelitian. Pada penelitian ini peneliti mendapatkan data dengan mengamati objek yang akan diteliti secara langsung di tempat penelitian yaitu Biro Sistem Informasi (BSI) UMY. Pengumpulan data meliputi *system* pengumpulan data presensi yang digunakan di Universitas Muhammadiyah Yogyakarta.

3.3.2 Wawancara

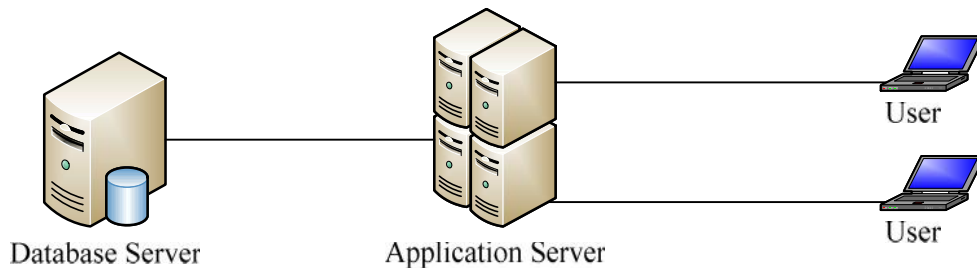
Wawancara merupakan percakapan dengan maksud tertentu. Percakapan dilakukan antara peneliti yang mengajukan pertanyaan dan yang diwawancarai memberikan jawaban atas pertanyaan itu. Proses wawancara ini dilakukan di Biro Sistem Informasi (BSI) UMY untuk mendapatkan data dengan melakukan dialog langsung dengan Kepala bagian Urusan Aplikasi (BSI) UMY Bapak Miftahul Yaum, S.Kom. yang dianggap dapat memberikan informasi yang dibutuhkan oleh

penulis. Tujuan melakukan wawancara ini adalah untuk mengetahui sistem yang dibutuhkan oleh UMY dari segi aritektur aplikasi, teknologi yang digunakan serta pemanfaatannya.

3.4 ANALISIS DATA

3.4.1 Arsitektur

Berdasarkan hasil observasi, Universitas Muhammadiyah Yogyakarta mempunyai banyak *server*. Beberapa *server* mempunyai data-data tertentu yang berbeda-beda. Setiap sistem informasi yang ada mempunyai *server* sendiri, sehingga peneliti mendapatkan sebuah analisis rancangan arsitektur seperti pada Gambar 3.2.



Gambar 3. 2 Arsitektur aplikasi

Database yang digunakan pada *database server* adalah *Microsoft SQL Server 2014* kemudian pada *application server* menggunakan *IIS* dalam mengolah *web server*. Aplikasi website yang digunakan oleh *user* adalah *web application ASP.NET MVC* yang sudah terstruktur antara *model*, *controller* dan *view*. Saat *user* mengakses aplikasi *website*, *application server* akan memuat antarmuka *website* dan melakukan pengambilan data dari *database server*. Melalui antarmuka *website* yang

telah dimuat oleh *application server* melalui *browser*, *user* dapat mengelola data baik membuat, menyimpan, menghapus dan mengedit data pada *website* dan disimpan ke *database*.

3.4.2 Analisis Pengguna

Berdasarkan hasil wawancara dengan bagian BSI (Biro Sistem Informasi) dan observasi di Universitas Muhammadiyah Yogyakarta, proses presensi mahasiswa yang diterapkan adalah sebagai berikut:

- 1) *Staf* membuat berita acara sesi perkuliahan.
- 2) Mahasiswa melakukan presensi dengan memberi tanda tangan pada lembar berita acara sesi perkuliahan.
- 3) *Staf* merekap data presensi berdasarkan lembar berita acara sesi perkuliahan pada aplikasi presensi.

Dari beberapa tahapan proses presensi yang ada, kita dapat mengurangi beberapa tahapan proses yang sebenarnya dapat dilakukan sekaligus. Sehingga dalam analisis ini peneliti memperoleh rancangan proses presensi seperti pada Gambar 3.3.



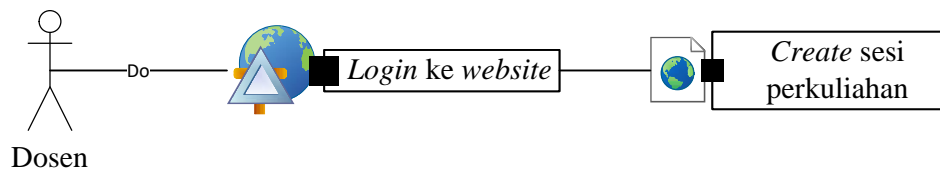
Gambar 3. 3 Analisis pengguna

Website yang dibuat dapat digunakan oleh 2 (dua) jenis pengguna yang memiliki hak akses yang berbeda, yakni:

- 1) Dosen, merupakan jenis pengguna yang dapat melakukan pengolahan data dalam *website*, Seperti menambah, menghapus, mengupdate dan serta monitoring sesi perkuliahan.
- 2) Mahasiswa, merupakan jenis pengguna yang dapat masuk ke *website* lalu melakukan proses *submit token* untuk presensi.

3.4.3 Analisis Pembuatan Sesi Perkuliahan

Pada proses presensi mahasiswa Universitas Muhammadiyah Yogyakarta saat ini, pembuatan sesi perkuliahan dilakukan oleh *staf* pada masing-masing prodi atau jurusan. Sebenarnya proses ini bisa dilakukan oleh dosen yang mengisi perkuliahan. Sehingga kita dapat mengurangi kerja staf yang dapat memperlambat proses presensi sampai data terekap. Peneliti dalam hal ini memperoleh sebuah rancangan seperti pada Gambar 3.4.



Gambar 3. 4 Analisis Pembuatan Sesi Perkuliahan

Penjelasan tentang Gambar 3.4. adalah sebagai berikut:

1. Pertama Dosen mengakses *website* presensi *online* UMY, dan login berdasarkan *email* dari kampus.
2. Dosen melakukan pembuatan sesi perkuliahan berdasarkan tahun ajaran, jurusan, matakuliah dan kelas.

3. Setelah membuat sesi perkuliahan, dosen akan masuk ke halaman *detail* sesi perkuliahan. Pada halaman *detail* sesi perkuliahan terdapat beberapa data tentang sesi perkuliahan yang telah dibuat seperti: tahun ajaran, kelas program, mata kuliah, kelas, ruangan, waktu mulai, waktu selesai, maksimal keterlambatan, isi perkuliahan, deskripsi dan *token*. Selain itu dosen juga bisa melihat data mahasiswa yang telah melakukan presensi berdasarkan *token*.
4. Kemudian dosen akan memberi kode *token* kepada mahasiswa untuk melakukan proses presensi.

3.4.4 Analisis Presensi Mahasiswa

Proses presensi mahasiswa Universitas Muhammadiyah Yogyakarta saat ini adalah mahasiswa memberikan tanda tangan pada lembar berita acara sesi perkuliahan. Namun, data tersebut masih belum terekap pada *database*. Data akan terekap setelah *staf* pada setiap prodi atau jurusan melakukan rekap data pada aplikasi presensi. Pada kondisi ini, proses presensi masih memakan waktu yang lama. Sehingga peneliti membuat sebuah rancangan yang datanya langsung terekap pada *database* setelah mahasiswa melakukan presensi seperti pada Gambar 3.5.



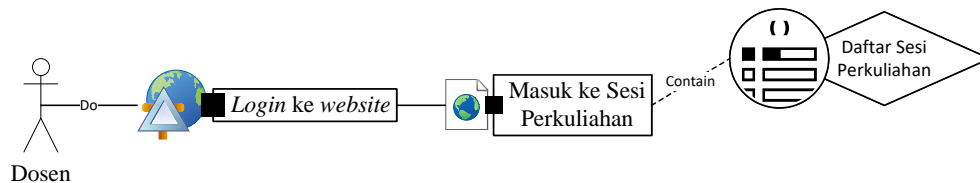
Gambar 3. 5 Analisis presensi mahasiswa

Penjelasan tentang Gambar 3.5. adalah sebagai berikut:

- 1) Pertama Mahasiswa akan memperoleh sebuah token dari dosen berdasarkan sesi yang telah dibuat.
- 2) Kemudian mahasiswa akan masuk ke aplikasi android untuk melakukan login berdasarkan email dari dari kampus.
- 3) Mahasiswa akan melakukan proses submit token untuk melakukan presensi.
- 4) Setelah berhasil, maka nama mahasiswa dan NIM akan muncul pada halaman *detail* sesi perkuliahan di *website*.

3.4.5 Analisis Pemantauan Kehadiran Mahasiswa

Berdasarkan hasil observasi pada Universitas Muhammadiyah Yogyakarta, proses pemantau kehadiran mahasiswa pada sesi perkuliahan dilakukan secara manual. Mahasiswa yang hadir akan memberi tanda tangan pada lembar berita acara sesi perkuliahan. Dosen tetap melakukan monitoring mahasiswa yang telah absen dengan yang hadir. Sehingga lembar presensi akan terkontrol keamanannya jika terdapat mahasiswa yang curang. Proses ini akan tetap diterapkan pada penelitian ini. sehingga diperoleh sebuah rancangan seperti pada Gambar 3.6.



Gambar 3. 6 Analisis pemantauan kehadiran mahasiswa

Penjelasan tentang Gambar 3.6. adalah sebagai berikut:

- 1) Pertama Dosen mengakses *website* presensi *online* UMY, dan login berdasarkan *email* dari kampus.

- 2) Dosen masuk ke *link* “lihat semua sesi”, maka *application server* akan memuat halaman *index*. Dimana halaman ini berisi daftar sesi perkuliahan yang pernah dibuat sebelumnya.
- 3) Selain itu, dosen juga bisa memantau mahasiswa yang hadir pada *link* “lihat sesi”. Didalam detail sesi perkuliahan, dosen dapat melihat mahasiswa yang hadir tepat waktu dan terlambat.

3.4.6 Analisis Kebutuhan Web

Analisis kebutuhan didapat dari studi *literature* dan diskusi bersama pengelola BSI UMY. Berikut ini merupakan analisis kebutuhan pada aplikasi presensi UMY:

- 1) Halaman Login untuk dosen yang akan mengisi sesi perkuliahan.
- 2) Halaman *Create Sesi* untuk membuat sesi perkuliahan mahasiswa.
- 3) Halaman *Detail Sesi* untuk melihat sekaligus memonitoring sesi perkuliahan.
- 4) Halaman *Edit Sesi* untuk melakukan perubahan jika terjadi kesalahan saat membuat sesi perkuliahan.
- 5) Halaman *Delete Sesi* untuk menghapus sesi perkuliahan.

3.5 RANCANGAN SISTEM

Rancangan sistem adalah suatu tahapan kegiatan yang dilakukan seseorang atau kelompok dalam merancang atau membuat sistem sebelum sistem dibuat dengan tujuan sistem yang dibangun sesuai dengan kebutuhan dalam memecahkan atau dengan kebutuhan pengguna berkaitan dengan pengolahan, pengelolaan dan perolehan informasi yang diinginkan. Rancangan sistem pada penelitian ini terdapat beberapa rancangan, yaitu rancangan sistem dan rancangan antarmuka pengguna.

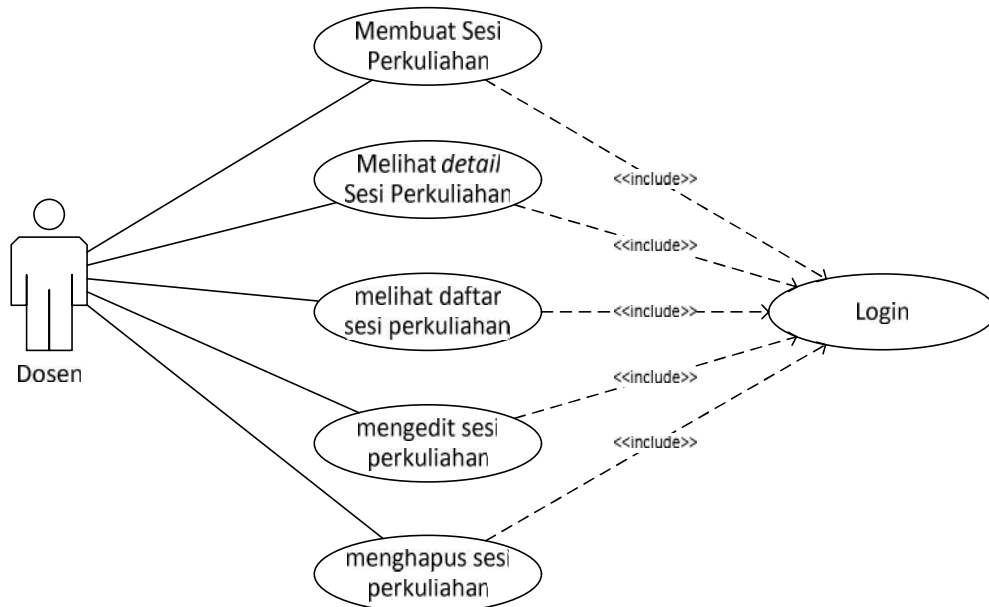
3.5.1 Rancangan Sistem

Dalam pembuatan aplikasi *web* dilakukan perancangan sistem menggunakan UML. Metode UML yang digunakan dalam perancangan aplikasi web ini antara lain *Use Case Diagram*, *Activity Diagram* dan *Sequence Diagram*. Perancangan tersebut dijelaskan sebagai berikut:

a. *Use Case Diagram*

Use Case Diagram merupakan teknik pemodelan untuk menjelaskan setiap aktifitas yang mungkin terjadi dengan menggunakan sudut pandang dari aktor sebagai pengguna sistem dan sudut pandang administrator.

Gambaran *Use Case Diagram* yang digunakan pada website dapat dilihat pada Gambar 3.7.

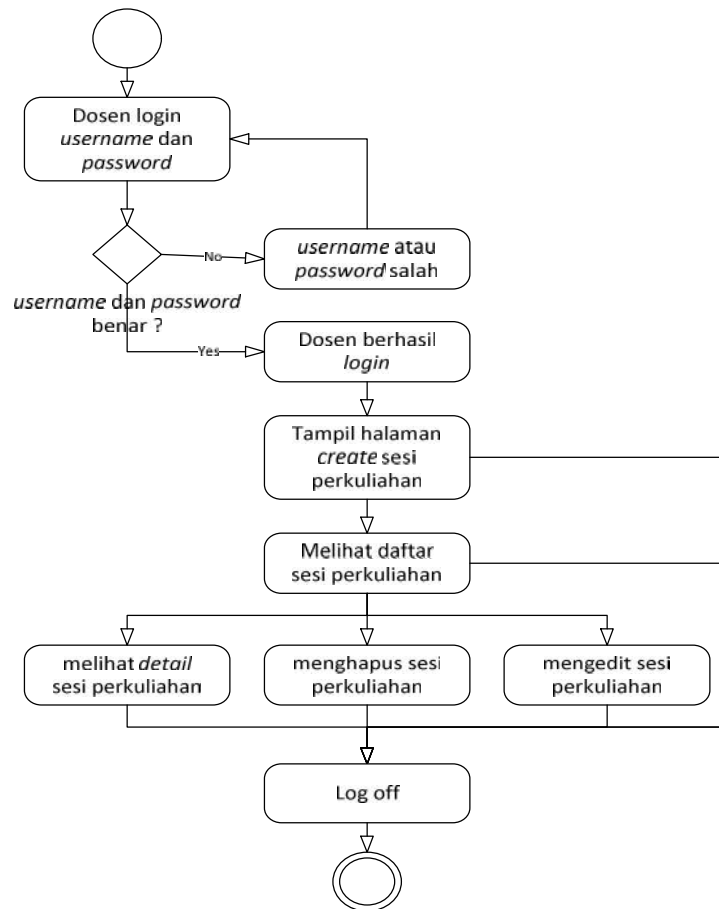


Gambar 3.7 *Use case diagram*

Gambar 3.7 menunjukkan bagaimana hubungan antara *actor* dan *use case*. Pada Gambar 3.7 menjelaskan bahwa pada aplikasi hanya Dosen yang diberi hak untuk akses. Dosen dapat mengolah data di semua fitur setelah melakukan proses *login*. Kemudian setelah *login* Dosen dapat membuka, melihat, mengedit dan menghapus sesi perkuliahan.

b. Activity Diagram

Berdasarkan *use case* yang telah dibuat maka dapat diperoleh *activity diagram* berdasarkan actor yang terlibat dalam *use case diagram*. Untuk lebih jelasnya dapat dilihat pada Gambar 3.8.



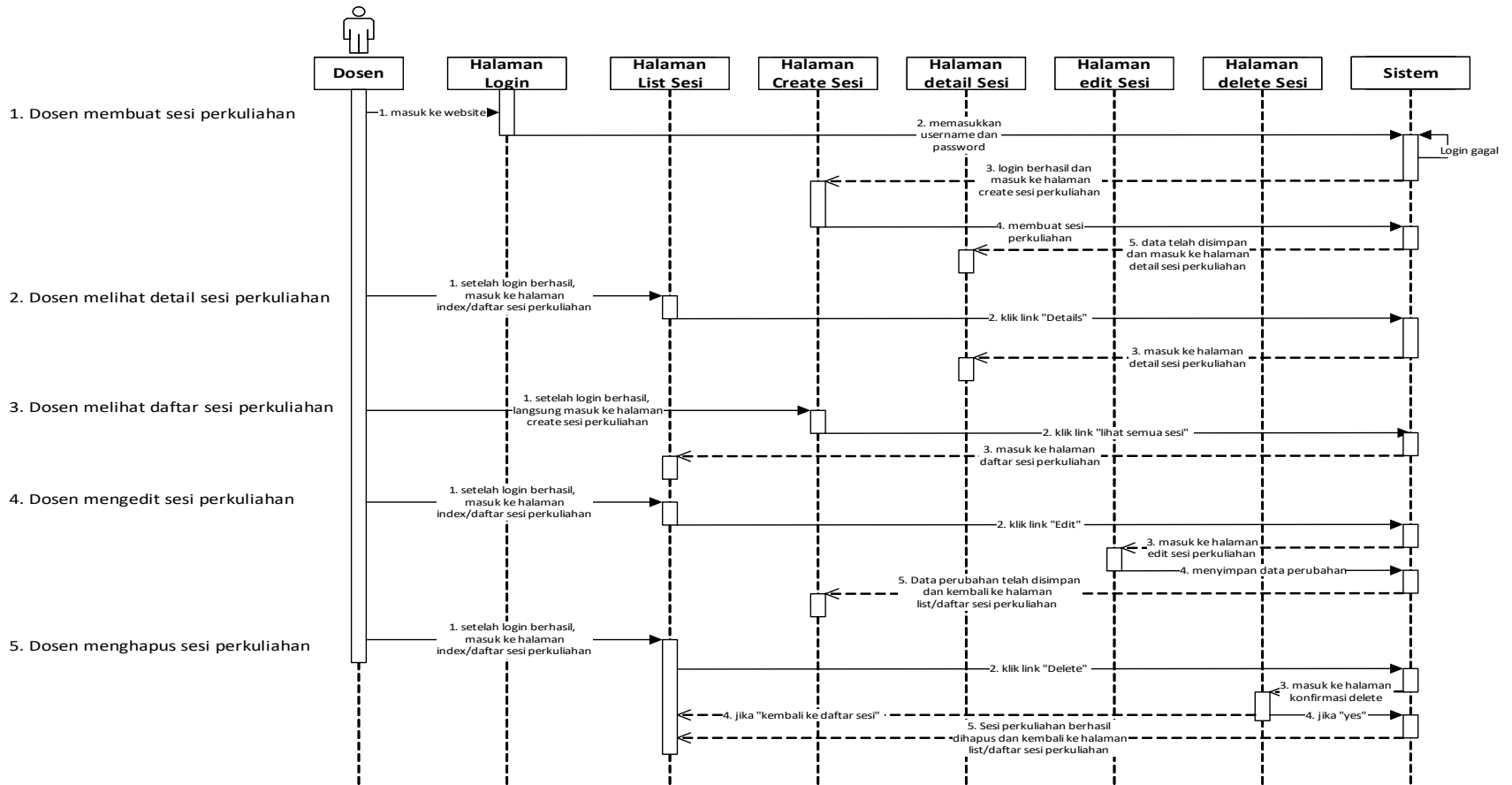
Gambar 3. 8 Activity diagram

Berikut penjelasan tentang langkah-langkah pada Gambar 3.8:

- 1) Terdapat beberapa proses ketika Dosen masuk pada *website* presensi *online*. Dosen diharuskan untuk melakukan *login* terlebih dahulu.
- 2) Jika *username* dan *password* benar maka akan masuk ke halaman *create* sesi perkuliahan, jika salah maka sistem akan meminta untuk memasukkan *username* dan *password* kembali
- 3) Setelah masuk ke halaman *create* sesi perkuliahan, Dosen membuat sesi perkuliahan berdasarkan mata kuliah yang diampu.
- 4) Setelah berhasil maka sistem akan mengarahkan ke halaman *detail* sesi perkuliahan.
- 5) Dalam kondisi ini Dosen juga bisa masuk ke beberapa halaman seperti: daftar sesi perkuliahan, *detail* sesi perkuliahan, *edit* sesi perkuliahan dan *delete* sesi perkuliahan.

c. *Sequence Diagram*

Berdasarkan *use case* yang telah dibuat maka dapat diperoleh juga *sequence diagram* yang menggambarkan kelakuan objek pada *use case* dengan mendiskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. *Sequence diagram* pada aplikasi ini dapat dilihat pada Gambar 3.9.



Gambar 3.9 Sequence diagram

Pada Gambar 3.9 diatas terdapat beberapa proses yang dijelaskan, antara lain:

- 1) Dosen membuat sesi perkuliahan, dalam proses ini dosen diharuskan untuk melakukan *login* terlebih dahulu. Setelah proses *login* berhasil maka sistem akan mengarahkan ke halaman *create* sesi perkuliahan. Dosen mengisi *form* berdasarkan mata kuliah yang diampu dan setelah dibuat sistem akan mengarahkan ke halaman *detail* sesi perkuliahan. Sehingga sampai disini dosen akan mendapatkan beberapa data berdasarkan data yang telah dimasukkan, data tersebut diataranya adalah *token* yang nantinya digunakan oleh mahasiswa untuk melakukan presensi.
- 2) Dosen melihat detail sesi perkuliahan, dalam proses ini dosen diharuskan untuk melakukan *login* terlebih dahulu. Masuk ke halaman daftar sesi perkuliahan, setelah itu klik link *detail* pada sesi yang akan dilihat. Setelah itu sistem akan mengarahkan pada halaman *detail* berdasarkan sesi yang dilihat.
- 3) Dosen melihat daftar sesi perkuliahan, dalam proses ini dosen diharuskan untuk melakukan *login* terlebih dahulu. Klik link “lihat daftar sesi” pada saat berada pada halaman *create* sesi perkuliahan
- 4) Dosen mengedit sesi perkuliahan, dalam proses ini dosen diharuskan untuk melakukan *login* terlebih dahulu. Masuk ke halaman daftar sesi perkuliahan, setelah itu klik link *edit* pada sesi yang akan diedit. Setelah itu sistem akan mengarahkan pada halaman *detail* berdasarkan sesi yang dilihat. Dosen melakukan perubahan data pada *form*. Setelah selesai klik *save*, maka sistem akan kembali pada halaman daftar sesi perkuliahan.

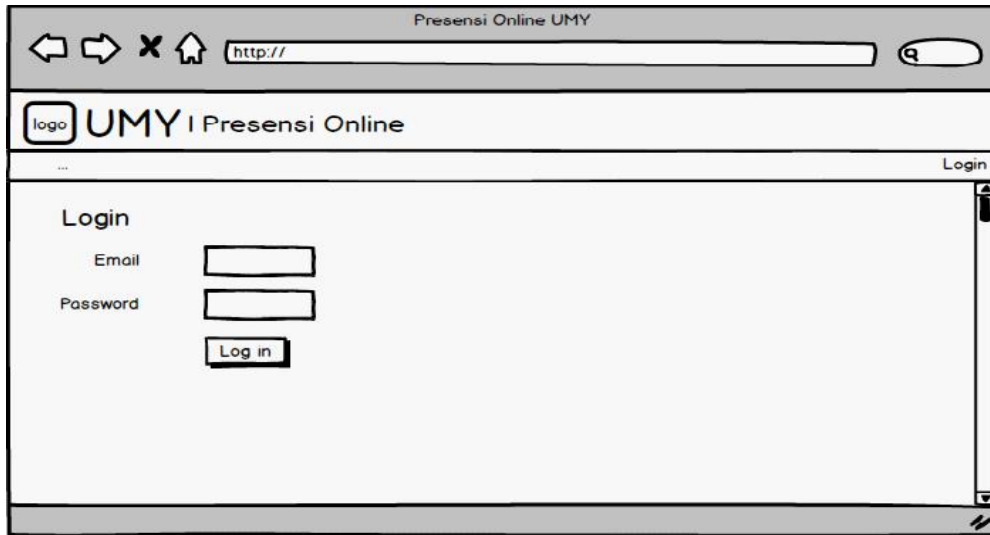
- 5) Dosen menghapus sesi perkuliahan, dalam proses ini dosen diharuskan untuk melakukan *login* terlebih dahulu. Masuk ke halaman daftar sesi perkuliahan, setelah itu klik link *delete* pada sesi yang akan dihapus. Setelah itu sistem akan mengarahkan pada halaman *delete* berdasarkan sesi yang dilihat. Konfirmasi akan muncul, apakah dosen yakin untuk menghapus sesi. Setelah di klik “Yes”, maka sistem akan menghapus data sesi perkuliahan yang dihapus dan mengembalikan ke halaman daftar sesi perkuliahan.

3.5.2 Rancangan Antarmuka

Rancangan antarmuka atau tampilan yang akan dirancang memberikan gambaran dari setiap bagian dalam *website*. Rancangan antarmuka ini menjelaskan keterkaitan setiap halaman menu dan penjelasan cara kerja dari setiap menu dari *website*. Rancangan antarmuka ini menjadi dasar untuk membuat tampilan pada *website* yang akan dibuat. Rancangan antarmuka ini hanya digunakan untuk dosen.

a. Rancangan Antarmuka Halaman *Login* Dosen

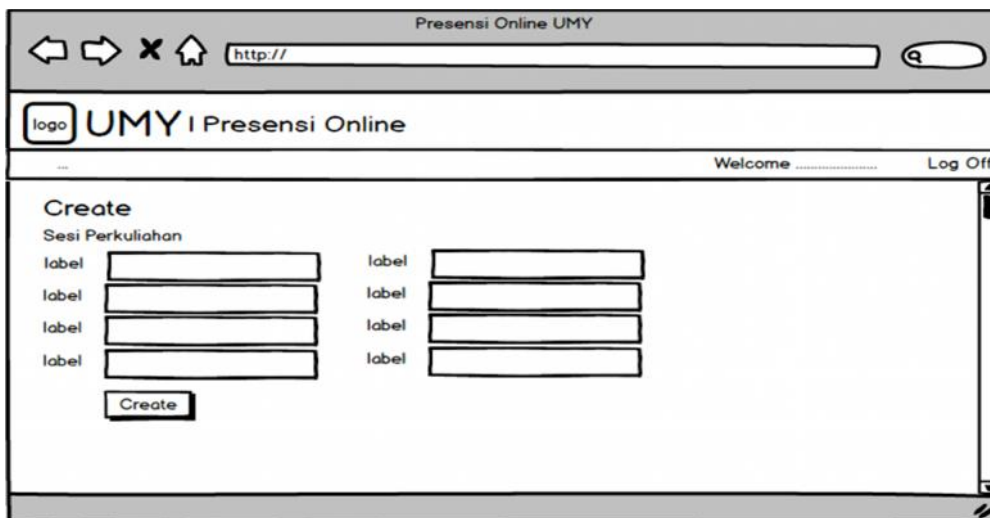
Rancangan antar muka ini adalah rancangan yang pertama kali muncul saat *website* pertama kali diakses. Dosen melakukan *login* sebelum masuk pada halaman sesi perkuliahan. Gambaran rancangan antarmuka halaman ini dapat dilihat pada Gambar 3.10.



Gambar 3. 10 Rancangan antarmuka halaman *login* dosen.

b. Rancangan Antarmuka Halaman *Create Sesi Perkuliahan*

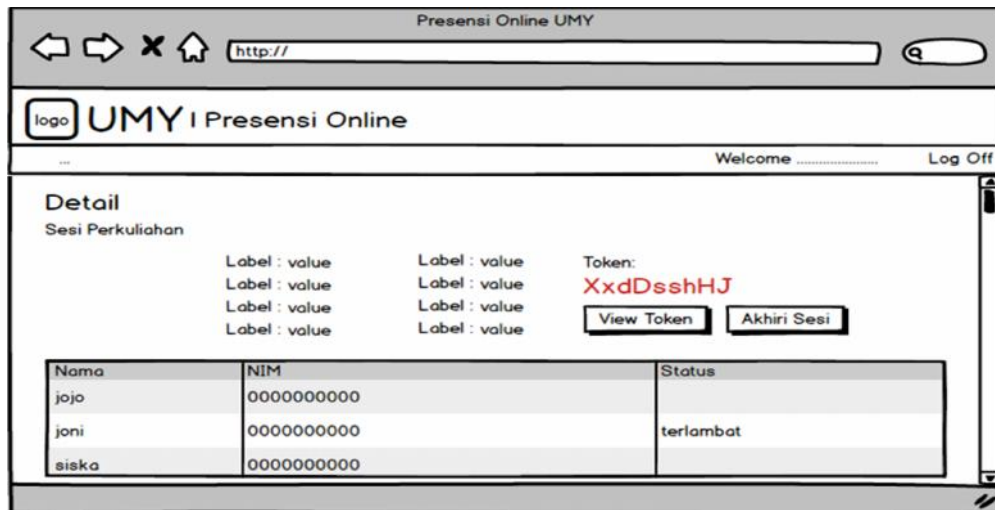
Rancangan antarmuka ini merupakan Halaman yang berisi *form* untuk membuat sesi perkuliahan untuk mahasiswa. Sehingga dosen akan membuat sesi baru ketika pergantian sesi perkuliahan. Gambaran rancangan antarmuka halaman ini dapat dilihat pada Gambar 3.11.



Gambar 3. 11 Rancangan antarmuka halaman *create* sesi perkuliahan

c. Rancangan Antarmuka Halaman *Detail* Sesi Perkuliahan

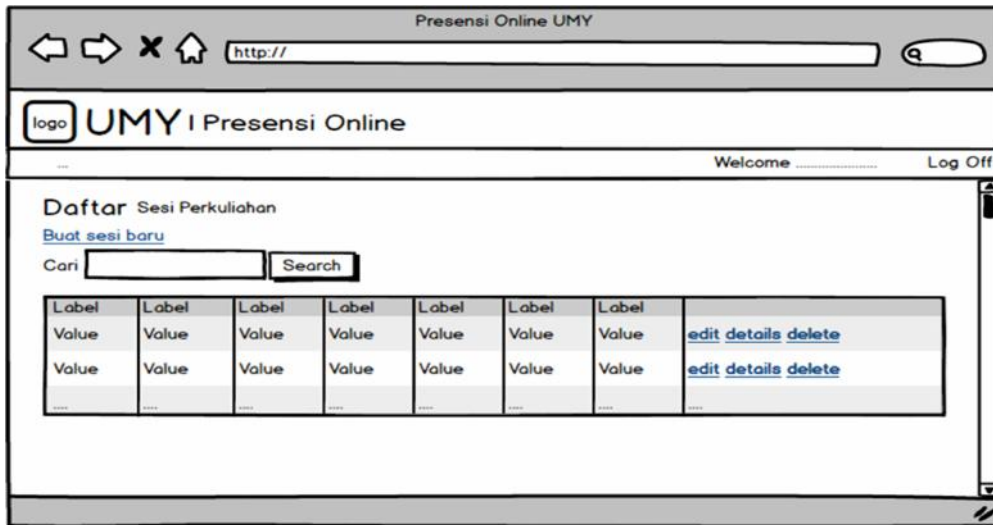
Rancangan antarmuka ini merupakan Halaman detail sesi perkuliahan yang telah dibuat. Halaman ini menampilkan beberapa data dari sesi perkuliahan salah satunya adalah *token*. Token akan digunakan oleh mahasiswa untuk melakukan presensi. Setelah mahasiswa melakukan presensi, maka data mahasiswa akan muncul di table bagian bawah. Gambaran rancangan antarmuka halaman ini dapat dilihat pada Gambar 3.12.



Gambar 3. 12 Rancangan antarmuka halaman *detail* sesi perkuliahan

d. Rancangan Antarmuka Halaman *List* Sesi Perkuliahan

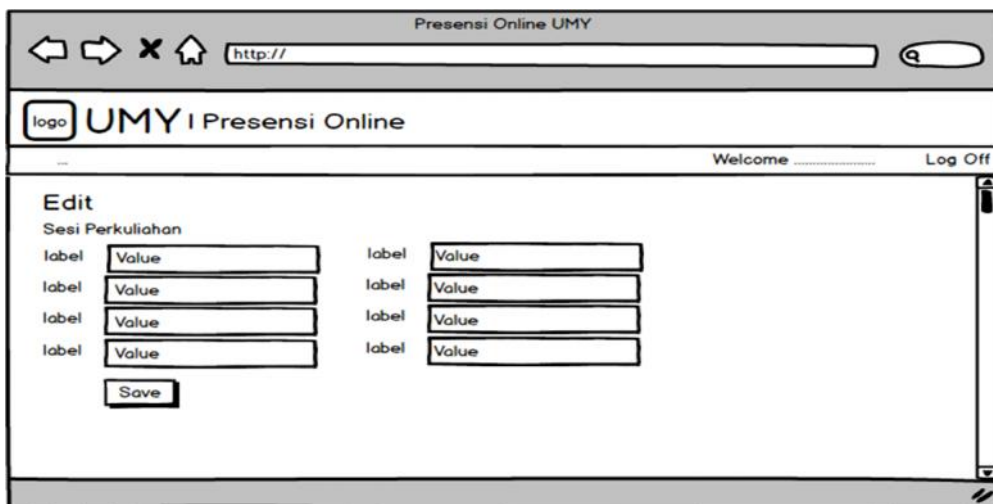
Rancangan antarmuka ini merupakan Halaman untuk melihat daftar sesi perkuliahan yang telah dibuat oleh dosen. Dosen dapat melakukan edit, delete dan melihat detail sesi perkuliahan. Gambaran rancangan antarmuka halaman ini dapat dilihat pada Gambar 3.13.



Gambar 3. 13 Rancangan antarmuka halaman *list* sesi perkuliahan

e. Rancangan Antarmuka Halaman *Edit* Sesi Perkuliahan

Rancangan antarmuka ini merupakan Halaman yang berisi *form* untuk mengedit sesi perkuliahan yang telah dibuat. Sehingga dosen akan mengedit sesi perkuliahan ketika terjadi kesalahan pada sesi perkuliahan. Gambaran rancangan antarmuka halaman ini dapat dilihat pada Gambar 3.14.



Gambar 3. 14 Rancangan antarmuka halaman *edit* sesi perkuliahan