# Lembar Kemajuan Skripsi

|  |  |  |
| --- | --- | --- |
| Nama | Dita Oktaria | Server HD:Users:ditaoktaria:Downloads:0227f0a.jpg |
| NIM | 105090602111001 |
| No. HP | 082332290104 |
| Email | ditaoktaria@gmail.com |
| Judul | Pengembangan Sistem *Online Classroom Presentation* berbasis Android dengan memanfaatkan Android Set-top-box |
| Pembimbing 1 | Herman Tolle, Dr. Eng, ST., MT. |
| Pembimbing 2 | Aryo Pinandito, ST., M.MT. |
| Tanggal Mulai | 29 September 2014 |

|  |  |  |
| --- | --- | --- |
| Tanggal | Pembimbing | Isi/Perbaikan |
| 20/11/2014 | 2 (email) | Bab I, II, III |
| 12/01/2015 | 2 (tatap muka) | Konsultasi Bab 1 |
| 19/01/2015 | 1 (tatap muka) | Konsultasi Bab 1 dan Player PPT |
| 26/01/2015 | 2 (email) | Konsultasi Bab 1-3 (Perbaikan Bab 1) |
| 29/03/2015 | 1 (tatap muka) | Konsultasi Bab 4 (Perbaikan Bab 1) |
|  |  |  |

PENGEMBANGAN SISTEM *ONLINE CLASSROOM PRESENTATION* BERBASIS ANDROID DENGAN MEMANFAATKAN ANDROID SET-TOP-BOX

**SKRIPSI**



Disusun Oleh :

**DITA OKTARIA**

**NIM. 105090602111001**

**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN**

**UNIVERSITAS BRAWIJAYA**

**PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER**

**PROGRAM STUDI INFORMATIKA/ILMU KOMPUTER**

**MALANG**

**2014**

**LEMBAR PERSETUJUAN**

PENGEMBANGAN SISTEM *ONLINE CLASSROOM PRESENTATION* BERBASIS ANDROID DENGAN MEMANFAATKAN ANDROID SET-TOP-BOX

**SKRIPSI**

LABORATORIUM

PEMROGRAMAN APLIKASI PERANGKAT BERGERAK

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer



Disusun Oleh:

DITA OKTARIA

NIM. 105090602111001

Skripsi ini telah disetujui oleh dosen pembimbing pada tanggal …

|  |  |
| --- | --- |
| **Dosen Pembimbing I**  **Herman Tolle, Dr. Eng, ST, MT**  **NIP. 19740823 200012 1 001** | **Dosen Pembimbing II**  **Aryo Pinandito, ST, M.MT**  **NIK. 83051916110374** |

# KATA PENGANTAR

Puji dan syukur Penulis panjatkan kehadirat Allah SWT, karena hanya dengan rahmat dan bimbingannya Penulis dapat menyelesaikan tugas akhir dengan judul “Pengembangan Sistem *Online Classroom Presentation* Berbasis Android Dengan Memanfaatkan Android Set-Top-Box” dengan baik. Tanpa rahmat dan bimbingan dari Tuhan Yang Maha Esa, maka niscaya Penulis tidak akan dapat menyelesaikan tugas akhir ini dengan baik dan tepat waktu.

Penulisan dan penyusunan laporan skripsi ini dapat terlaksana dengan baik karena adanya bantuan secara langsung maupun tidak langsung dari pihak tertentu diantaranya:

1. Bapak Herman Tolle, Dr. Eng, ST., MT. selaku dosen pembimbing I yang telah banyak memberikan ide, pemikiran, bimbingan, ilmu, dan saran dalam penyusunan tugas akhir ini
2. Bapak Aryo Pinandito, ST., M.MT. selaku dosen pembimbing II yang juga telah banyak memberikan ide, pemikiran, bimbingan, ilmu, dan saran dalam penyusunan tugas akhir ini.
3. Kedua orang tua Penulis, Drs. Sumardin dan Dra. Nurhasni yang telah memberi motivasi, kasih sayang serta dukungan moril dan materil kepada Penulis. Kakak Suceputra Sigunsa, Cengristitama, Ripsidasiona yang telah memberikan semangat dari awal sampai akhir pengerjaan tugas akhir ini. Adek Azka, Zahran, Raisa dan Amara yang secara tidak langsung telah menjadi motivasi bagi penulis agar tugas akhir ini dapat terselesaikan dengan baik. Serta Akhmad Syaiful Yamang yang telah memberi semangat, saran serta dukungan moril dan materil kepada penulis dalam pengerjaan tugas akhir ini.
4. Bu Dian Eka Ratnawati, S.Si, M.Kom selaku dosen pembimbing akademik yang telah memberikan bimbingan, ilmu, dan saran selama Penulis menuntut ilmu.
5. Semua teman-teman di PTIIK, terima kasih atas segala bantuannya selama menjadi mahasiswa.
6. Seluruh rekan kerja dan supervisor Unit TIK UB, terima kasih untuk segala pengalaman dan pelajaran selama bergabung di Unit TIK UB.
7. Semua teman-teman di PTIIK, terima kasih atas segala bantuannya selama menjadi mahasiswa.
8. Teman-teman ilmu komputer 2010 terutama Anjar Dwi, Farah Bahtera, Fayruz Al-baity, Shinta Ayu, Meitasari yang telah memberikan dukungan moril dan menjadi teman belajar mulai dari awal perkuliahan hingga akhir.
9. Semua pihak yang tidak dapat penulis sebutkan satu per satu yang terlibat baik secara langsung maupun tidak langsung demi terselesaikannya tugas akhir ini.

Semoga jasa dan amal baik mendapatkan balasan dari Allah SWT. Dengan segala kerendahan hati, penulis menyadari sepenuhnya bahwa skripsi ini masih jauh dari sempurna karena keterbatasan materi dan pengetahuan yang dimiliki penulis. Akhirnya semoga skripsi ini dapat bermanfaat dan berguna bagi pembaca terutama mahasiswa PTIIK Universitas Brawijaya.

Malang, November 2014

Penulis

# DAFTAR ISI

Lembar Kemajuan Skripsi i

KATA PENGANTAR i

DAFTAR ISI iii

DAFTAR GAMBAR iv

BAB I 1

PENDAHULUAN 1

1.1 Latar Belakang 1

1.2 Rumusan Masalah 2

1.3 Batasan Masalah 3

1.4 Tujuan 3

1.5 Manfaat 3

1.6 Sistematika Penulisan 3

BAB II 5

KAJIAN PUSTAKA DAN DASAR TEORI 5

2.1 Android Set-top-box 5

2.2 XML 6

BAB III 7

METODOLOGI PENELITIAN 7

3.1 Studi Literatur 8

3.2 Analisis Kebutuhan 8

3.3 Perancangan 8

3.4 Implementasi 9

3.5 Pengujian dan Analisis 9

3.6 Pengambilan Kesimpulan dan Saran 10

DAFTAR PUSTAKA 28

# DAFTAR GAMBAR

Gambar 2. 1 Perangkat Set-top-box 6

Gambar 2. 2 Contoh tampilan ViewerJS 8

Gambar 3. 1 Diagram alir runtutan pengerjaan skripsi 8

Gambar 4. 1 Diagram Pohon Analisis dan Perancangan 12

Gambar 4. 2 Diagram *Use Case* 16

Gambar 4. 3 *Entity Relational Diagram* 30

Gambar 4. 4 *Activity Diagram Login* 34

Gambar 4. 5 *Activity Diagram* Lupa *Password* 35

Gambar 4. 6 *Activity Diagram* Kelola Akun 36

Gambar 4. 7 *Acitvity Diagram* Unggah Materi 37

Gambar 4. 8 *Activity Diagram* Kelola Materi 39

Gambar 4. 9 *Activity Diagram* Presentasi Materi 40

Gambar 4. 10 *Activity Diagram* Kelola *Folder* 42

Gambar 4. 11 Antarmuka Halaman *Login* 43

Gambar 4. 12 Antarmuka Halaman Utama 44

Gambar 4. 14 Antarmuka Halaman Kelola Akun 45

# DAFTAR TABEL

Tabel 2. 1 Pemetaan Entitas JSON dan JAVA 7

Tabel 4. 1 Identifikasi Aktor 14

Tabel 4. 2 Spesifikasi Kebutuhan Fungsional 14

Tabel 4. 3 Skenario *Use Case Login* 16

Tabel 4. 4 Skenario *Use Case* Lupa *Password* 18

Tabel 4. 5 Skenario *Use Case* Kelola Akun 19

Tabel 4. 6 Skenario *Use Case* Unggah Materi 20

Tabel 4. 7 Skenario *Use Case* Kelola Materi 22

Tabel 4. 8 Skenario *Use Case* Presentasi Materi 24

Tabel 4. 10 Spesifikasi Kebutuhan Non-Fungsional 28

Tabel 4. 12 Strutur Tabel *Material* 32

Tabel 4. 13 Struktur Tabel *Folder* 32

Tabel 4. 14 Struktur Tabel *Course* 32

Tabel 4. 15 Struktur Tabel *Teaching* 33

Tabel 4. 16 Struktur Tabel *Lecturer\_Team* 33

# DAFTAR KODE

Kode 2. 2 Contoh Penggunaan *Query* 7

Kode 2. 3 Elemen tombol Next pada html 8

Kode 2. 4 Pemanggilan fungsi e 8

Kode 2. 5 Deklarasi fungsi e 9

Kode 2. 6 Deklarasi fungsi showPage 9

# BAB I

# PENDAHULUAN

## Latar Belakang

Perkembangan teknologi diikuti dengan perkembangan metode yang digunakan dalam menyampaikan materi dalam dunia pendidikan. Mulai dari penyampaian dengan menuliskan materi di papan tulis hingga dengan menggunakan file presentasi grafis pada PC yang ditayangkan melalui proyektor. Penggunaan presentasi grafis, misalnya Microsoft PowerPoint, di kelas dipercaya dapat membantu penyampaian materi dari penyaji presentasi kepada siswa dan meningkatkan kredibilitas penyaji presentasi. Sebanyak 73% dari 485 siswa yang diteliti menunjukkan bahwa presentasi grafis dapat membantu meningkatkan ketertarikan mereka dalam pelajaran. Namun dibutuhkan PC dan perangkat proyeksi mahal untuk mendukung presentasi grafis tersebut [AS-98].

Dalam menyampaikan materi, umumnya para pengajar bekerja sama, baik dalam menyusun materi, membuat presentasi dan tugas latihan, namun tetap mengajar secara mandiri pada kelas masing-masing. Bentuk metode pengajaran ini disebut dengan *team teaching* [KG-00]. Penggunaan metode tersebut dikarenakan setiap mata pelajaran umunya diampu oleh beberapa pengajar. Para pengajar dalam menyampaikan materinya, sering kali menggunakan laptop pribadi. Hal ini tidak efektif, karena pengajar harus selalu membawa laptop yang relatif berat untuk dapat melakukan presentasi melalui proyektor, meskipun materi yang dipresentasikan adalah materi sama yang digunakan berulang-ulang. Permasalahan dalam penyampaian materi tersebut dapat diatasi dengan memanfaatkan teknologi set-top-box.

Set-top-box adalah sebuah perangkat yang memungkinkan sebuah pesawat televisi memiliki layanan internet dan juga memungkinkan sebuah pesawat televisi untuk menerima dan membaca siaran televisi digital. Perkembangan teknologi memungkinkan kita untuk dapat mengirimkan konten-konten multimedia melalui Internet. Pengiriman konten dapat terjadi selama perangkat pengirim dan penerima memiliki kemampuan mengenkripsi dan mendekripsi aliran multimedia. Hal inilah yang ikut memicu perubahan pasar set-top-box dari tv tunner menjadi mini PC yang mudah dikembangkan sesuai dengan kebutuhan tertentu [LK-12]. Set-top-box yang relatif lebih murah dibandingkan PC seperti laptop, umumnya digunakan untuk keperluan hiburan. Kini set-top-box generasi baru memungkinkan penggunanya untuk membuat aplikasi sendiri dan menjalankannya pada sistem operasi set-top-box yang berbasis Android, sehingga memungkinkan set-top-box untuk mengatasi permasalahan penyajian presentasi ini, yaitu dengan membuat sebuah sistem untuk mengelola file-file presentasi yang terpusat dalam satu server sekaligus sebagai media pengajar untuk menayangkan presentasi secara online melalui set-top-box. Pengajar dapat menjalankan satu file presentasi untuk kelas yang berbeda tanpa membawa laptop. Selain itu sistem yang dibangun nantinya juga diharapkan dapat mendukung metode pembelajaran *team teaching* dengan berbagi konten materi. Hal ini berarti jika salah satu anggota pengajar team teaching telah mengunggah file presentasi di sistem, maka anggota yang lain dapat langsung menayangkannya di kelas tanpa harus mengunggah kembali ke sistem. Pada penelitian ini, sistem dibuat dengan menggunakan model pengembangan *native Android application* karena sistem operasi yang digunakan oleh set-top-box merupakan sistem operasi Android.

## Rumusan Masalah

Berdasarkan pada permasalahan yang telah dijelaskan pada latar belakang, maka rumusan masalah yang dapat disusun adalah sebagai berikut :

1. Bagaimana merancang dan mengimplementasikan sistem presentasi online yang dapat mengelola file-file presentasi yang terpusat dalam satu server yang memungkinkan berbagi *file* antar pengguna.
2. Bagaimana merancang dan mengimplementasikan sistem presentasi online yang dapat menayangkan presentasi berbagai format media secara online melalui Android set-top-box.

## Batasan Masalah

Untuk menghindari kemungkinan semakin berkembangnya masalah, maka batasan masalah dalam skripsi ini antara lain adalah :

1. Sistem dalam penelitian ini dibuat untuk pengajaran mata kuliah Metode Penelitian, Perancangan Aplikasi Perangkat Bergerak dan Jaringan Multimedia dalam lingkup PTIIK.
2. Sistem dalam penelitian ini menggunakan jaringan lokal di Universitas Brawijaya.
3. File presentasi yang digunakan adalah file yang berformat PPT dan PDF.

## Tujuan

Tujuan dari penelitian skripsi ini adalah merancang dan mengimplementasikan sistem presentasi online yang dapat mengelola file-file presentasi yang terpusat dalam satu server dan menayangkan presentasi secara *online* melalui Android set-top-box.

## Manfaat

Manfaat yang dapat diperoleh dari penelitian skripsi ini antara lain :

1. Membantu pengajar dalam mengelola file-file presentasi agar lebih teratur dan bisa diakses secara *online*.
2. Membantu pengajar menayangkan presentasi tanpa perlu membawa laptop pribadi.

## Sistematika Penulisan

Sistematika penulisan ditujukan untuk memberikan gambaran dan uraian dari penulisan skripsi ini secara garis besar meliputi beberapa bab, sebagai berikut:

**BAB I PENDAHULUAN**

Pada bagian ini dipaparkan mengenai hal-hal yang melatar belakangi penelitian, rumusan, dan batasan permasalahan dalam penelitian, serta tujuan dan relevansi atau manfaat penelitian terhadap perkembangan solusi dari permasalahan, serta sistematika penulisan dalam laporan tugas akhir ini.

**BAB II KAJIAN PUSTAKA DAN DASAR TEORI**

Pada bagian ini diuraikan teori-teori penunjang dalam penulisan laporan tugas akhir ini. Pustaka yang digunakan baik dari buku, diktat, maupun materi dari internet.

**BAB III METODE PENELITIAN**

Pada bagian ini menguaraikan tentang metode dan langkah kerja yang dilakukan dalam penulisan tugas akhir yang terdiri dari studi literatur, perancangan sistem, implementasi sistem, pengujian dan analisis serta pengambilan kesimpulan dan saran.

**BAB IV ANALISIS DAN PERANCANGAN**

Pada bagian ini membahas analisis dan perancangan aplikasi sesuai dengan dasar teori dan literatur yang ada.

**BAB V IMPLEMENTASI DAN PENGUJIAN**

Pada bagian ini akan menguraikan tentang implementasi dan pengujian aplikasi sesuai dengan perancangan aplikasi yang telah dibuat.

**BAB VI PENUTUP**

Pada bagian ini berisi kesimpulan yang diperoleh dari pelaksaan pengerjaan tugas akhir. Dan juga saran untuk mengembangkan hasil analisis penelitian tugas akhir ini agar dapat dikembangkan ke penelitian berikutnya.

# BAB II

# KAJIAN PUSTAKA DAN DASAR TEORI

Pada bab ini akan diuraikan mengenai kajian pustaka terhadap penelitian sebelumnya dan teori-teori dasar pembuatan sistem diantaranya tentang Android Set-top-box, XML, PHP, MySQL, JSON.

## Android Set-top-box

Set Top Box adalah sebuah perangkat yang memungkinkan sebuah pesawat televisi menjadi pengguna internet dan juga memungkinkan sebuah pesawat televisi untuk menerima dan membaca siaran televisi digital. Perkembangan teknologi memungkinkan kita untuk dapat mengirimkan konten-konten multimedia melalui Internet. Pengiriman konten dapat terjadi selama perangkat pengirim dan penerima memiliki kemampuan mengenkripsi dan mendekripsi aliran multimedia. Hal inilah yang ikut memicu perubahan pasar set-top-box dari TV tunner menjadi mini PC yang mudah dikembangkan untuk kebutuhan tertentu [LK-12]. Set-top-box relative lebih murah dibandingkan PC maupun miniPC lainnya. Kini, set-top-box generasi baru memungkinkan penggunanya untuk membuat aplikasi sendiri dan menjalankannya pada sistem operasi set-top-box yang berbasis linux.

Android set-top-box dilengkapi dengan 10/100M Ethernet dan 802.11 b/g/n Wi-Fi yang memungkinkan perangkat ini dapat mengakses konten-konten dari internet maupun melakukan pengiriman konten dengan perangkat lainnya melalui internet. Selain itu, setiap Android set-top-box memiliki port seperti hdmi atau AV yg berguna untuk menampilkan gambar atau video ke layar monitor.



Gambar 2.1 Perangkat Set-top-box

Sumber : [AND-14]

## PHP

PHP (*Hypertext Preprocessor*) adalah bahasa skrip yang berjalan pada sisi server. PHP bersifat open source dan dapat dijalankan di semua tipe sistem operasi. Bahasa pemrograman Java tidak dapat memanggil *database* MySQL untuk mendapatkan data. Sedangkan PHP memiliki dukungan untuk instansiasi objek java dan menggunakannya sebagai objek PHP [PHP-15]. Sehingga dibutuhkan skrip PHP yang akan memanggil MySQL kemudian mengubah data dari MySQL menjadi ke dalam bentuk JSON. Kemudian PHP akan memberikan data dalam bentuk JSON tersebut kepada Java. Penggunaan skrip php dimulai oleh tag <?php dan diakhiri oleh tag ?>. Berikut adalah contoh penggunaan *script* php yang digunakan dalam penelitian ini untuk melakukan koneksi ke database mysql :

|  |
| --- |
| <?php  $con = mysqli\_connect(“ditaoktaria.com”,”username”,”password”,”skripsi\_db”);  $sql=”CREATE TABLE table1(Username CHAR(30), Password CHAR(30), Role CHAR(30))”;  if (mysqli\_query($con, $sql))  {  echo “Table have been created successfully”;  }  ?> |

Kode 2. 1 Contoh Penggunaan *Script* PHP

## MySQL

MySQL adalah Relational Database Management System (RDBMS) yang bersifat *open source*. MySQL bersifat cepat, dapat diandalkan, dan mudah digunakan [MSQ-15]. MySQL pada penelitian ini berjalan pada sisi server dan digunakan bersama skrip PHP.

Berikut contoh query sederhana yang digunakan dalam penelitian ini :

|  |
| --- |
| select material\_name from material, lecturer where material.id\_lecturer=lecturer.id\_lecturer; |

Kode 2. 2 Contoh Penggunaan *Query*

## JSON

Data dari MySQL tidak bisa langsung dibaca oleh bahasa Java. Data dari MySQL akan diubah ke dalam bentuk JSON oleh PHP kemudian dikirimkan ke dalam fungsi java untuk dibaca dan diformat kembali ke dalam bentuk tampilan yang bisa dibaca oleh pengguna. Tabel 2.1 Menunjukkan pemetaan antara entitas JSON dan Java [TP-15] :

Tabel 2. 1 Pemetaan Entitas JSON dan Java

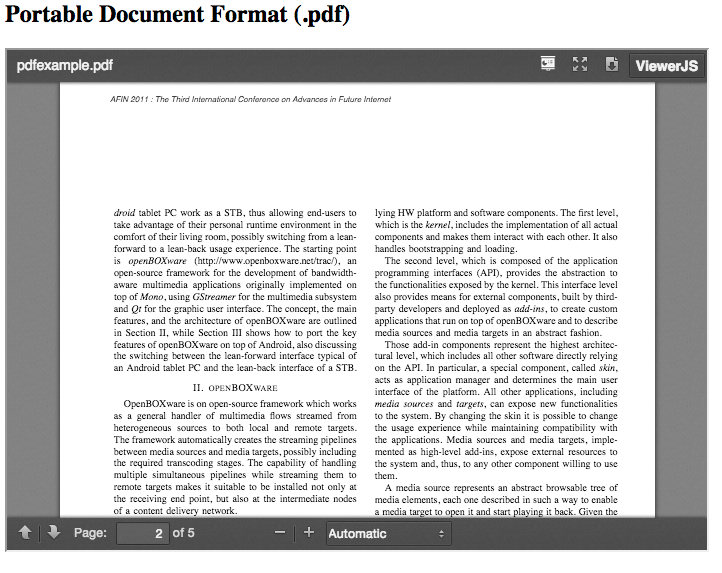
|  |  |
| --- | --- |
| JSON | Java |
| string | java.lang.String |
| number | java.lang.Number |
| truefalse | java.lang.Boolean |
| null | null |
| array | java.util.List |
| object | java.util.Map |

## VideoView

VideoView dapat memutar video yang memiliki format 3GP, MP4, MKV, WEBM.

## ViewerJS

VieverJS adalah sebuah *opensource library* yang digunakan untuk menjalankan *file* presentasi dalam sebuah halaman web. Format *file* yang dapat dijalankan dengan ViewerJS adalah ODT, ODP, ODS, dan PDF [VJS-15]. Tampilan *file* PDF dengan menggunakan ViewerJS ditunjukkan pada Gambar 2.2.



Gambar 2. 2 Contoh tampilan ViewerJS

ViewerJS dibuat dengan menggunakan bahasa HTML dan Javascript. Seperti yang ditunjukkan pada Gambar 2.2, ViewerJS terdiri dari tombol panah ke atas yang berfungsi untuk membuka halaman sebelumnya dan tombol panah ke bawah yang berfungsi untuk membuka halaman selanjutnya.

|  |
| --- |
| <button id = "next" class = "toolbarButton pageDown" title = "Next Page"></button> |

Kode 2. 3 Elemen tombol Next pada html

File index.html adalah file yang pertama kali akan dijalankan ketika library ViewerJS digunakan. Kode 2.3 menunjukkan elemen tombol Next pada file index.html yang dimiliki oleh ViewerJS. Elemen tombol *next* memiliki id “next”, id ini digunakan untuk operasi javascript pada file viewer.js. Kode 2.4 menunjukkan pemanggilan fungsi e yang memiliki nilai parameter next dan b.showNextPage pada *file* viewer.js.

|  |
| --- |
| e("next", b.showNextPage) |

Kode 2. 4 Pemanggilan fungsi e

Ditunjukkan oleh Kode 2.5 bahwa next=a dan b.showNextPage=b. Sehingga variabel c dapat dideklarasikan sebagai elemen next. Ketika elemen next ditekan, maka fungsi b.showNextPage akan dijalankan.

|  |
| --- |
| function e(a,  b) {  var c = document.getElementById(a);  c.addEventListener("click", function() {  b();  c.blur()  })  } |

Kode 2. 5 Deklarasi fungsi e

Operasi fungsi showNextPage ditunjukkan oleh Kode 2.6. Fungsi showNextPage berisi pemanggilan fungsi showPage dengan parameter n+1, dimana atas deklarasi fungsi showPage n didapat dari elemen pageNumber, maka fungsi showNextPage akan menampilkan halaman dengan nomor halaman lebih besar satu nomor dari nomor halaman yang sedang ditampilkan saat ini. Sedangkan fungsi untuk membuka halaman sebelumnya ditunjukan oleh fungsi showPreviousPage yang akan memanggil bernilai n-1, sehingga akan ditampilkan halaman dengan nomor halaman satu nomor lebih kecil dari nomor halaman yang sedang ditampilkan sekarang.

|  |
| --- |
| this.showPage = function(a) {  0 >= a ? a = 1 : a > r.length && (a = r.length);  c.showPage(a);  n = a;  document.getElementById("pageNumber").value = n  };  this.showNextPage = function() {  b.showPage(n + 1)  };  this.showPreviousPage = function() {  b.showPage(n - 1)  }; |

Kode 2. 6 Deklarasi fungsi showPage

# BAB III

# METODOLOGI PENELITIAN

Pada bab ini dijelaskan langkah - langkah yang akan dilakukan dalam pengerjaan skripsi, yaitu deskripsi aplikasi, studi literatur, perancangan, implementasi, analisis dan pengujian dari sistem yang akan dibuat. Kesimpulan dan saran disertakan sebagai catatan atas aplikasi dan kemungkinan arah pengembangan sistem selanjutnya. Berikut diagram alir runtutan pengerjaan skripsi ini:



Gambar 3. 1 Diagram alir runtutan pengerjaan skripsi

## Studi Literatur

Studi literatur menjelaskan dasar teori yang digunakan untuk menunjang penulisan skripsi. Berikut dasar teori yang digunakan dalam studi literatur:

1. PengenalanAndroid set-top-box
2. PHP
3. MySQL
4. JSON

## Analisis Kebutuhan

Analisis kebutuhan bertujuan untuk mendapatkan semua kebutuhan yang diperlukan oleh sistem yang akan dibangun. Analisis kebutuhan dimulai dengan mempelajari sistem yang telah ada, kemudian menganalisis permasalahan, kelemahan dan kebutuhan dari sistem tersebut. Hingga hasilnya dapat digunakan sebagai bahan analisis untuk mendapatkan solusi sistem baru yang memiliki keunggulan dari sistem yang sudah ada

Analisis kebutuhan sistem digambarkan menggunakan bahasa pemodelan UML. Dalam pemodelan UML, tahap analisis kebutuhan direpresentasikan dengan menggunakan *Use Case diagram* untuk mendapatkan konsep yang jelas mengenai sistem yang akan dikembangkan. Use Case Diagram merepresentasikan interaksi antara pengguna dengan sistem [AN-09]. Terdapat empat langkah dalam tahap analisis kebutuhan ini, yaitu melakukan penjabaran tentang gambaran umum sistem, melakukan proses identifikasi aktor yang terlibat dalam sistem, membuat daftar kebutuhan pengguna dengan menganalisis kebutuhan fungsional dan kebutuhan non fungsional.

## Perancangan

Proses perancangan dilakukan setelah semua kebutuhan sistem didapatkan melalui tahap analisis kebutuhan. Tahap pertama dalam perancangan sistem adalah perancangan umum sistem, kemudian tahap selanjutnya adalah proses perancangan perangkat lunak. Tahap perancangan umum sistem menjelaskan mengenai gambar proses kerja sistem secara umum. Sedangkan proses perancangan perangkat lunak terdiri dari:

* 1. Perancangan arsitektur sistem

Perancangan arsitektur sistem dilakukan dengan merepresentasikan relasi dari masing-masing komponen penyusun sistem.

* 1. Pemodelan *activity diagram*

Pemodelan activity diagram digunakan untuk merancang alur aktivitas yang dilakukan antara pengguna dengan sistem.

* 1. Perancangan basis data

Model representasi dari perancangan basis data menggunakan *Entity Relationship Diagram* (ERD) yang merepresentasikan entitas dan atribut beserta relasinya.

* 1. Perancangan antarmuka

Perancangan antarmuka dilakukan dengan membuat *layout* yang terdiri dari komponen-komponen *layout* seperti *textbox*, *button* dan *label*.

## Implementasi

Implementasi sistem mengacu pada perancangan sistem. Implementasi sistem pada sistem ini menggunakan model *native* dengan menggunakan bahasa pemrograman Java Android. Sedangkan pengelolaan data sistem yang dibangun menggunakan Database Management System MySQL. Pada tahap akhir dilakukan implementasi simulasi pada *hardware* secara langsung dari Eclipse menggunakan Android Develepmoent Tools (ADT).

## Pengujian dan Analisis

Pengujian yang dilakukan pada penilitian ini adalah pengujian validasi dan kompatibilitas. Pengujian validasi menggunakan metode black-box testing yang dilakukan untuk mengetahui kesesuaian sistem yang dibuat telah sesuai dengan spesifikasi kebutuhan yang melandasinya. Sedangkan pengujian kompatibilitas digunakan untuk mengetahui tingkat keberhasilan aplikasi untuk diterapkan dalam versi-versi berbeda dalam sistem operasi Android. Setelah tahap pengujian selesai, maka dilakukan analisis terhadap hasil pengujian sehingga mendapatkan kesimpulan dari sistem yang telah dibuat.

## Pengambilan Kesimpulan dan Saran

Tahap terakhir dari penulisan adalah pengambilan kesimpulan dan saran.. Kesimpulan diambil dari hasil pengujian dan analisis terhadap sistem yang dibangun untuk mengetahui sistem yang dibuat telah sesuai dengan spesifikasi kebutuhan dan telah memenuhi manfaat dan tujuan dilakukannya penelitian. Pengambilan saran dilakukan dengan tujuan untuk memperbaiki kesalahan atau kekurangan yang terjadi pada penelitian untuk memberikan pertimbangan atas pengembangan sistem lebih lanjut.

# BAB IV

# ANALISIS DAN PERANCANGAN

Bab ini membahas mengenai analisis dan perancangan aplikasi. Berikut diagram tahapan analisis dan perancangan aplikasi pada penelitian ini :



Gambar 4. 1 Diagram Pohon Analisis dan Perancangan

## Analisis Kebutuhan

Tahap analisis kebutuhan diawali dengan menjabarkan gambaran umum aplikasi, melakukan identifikasi aktor untuk menentukan aktor-aktor yang terlibat dalam penggunaan aplikasi, penjabaran kebutuhan fungsional yang akan direpresentasikan ke dalam diagram *use case* serta yang terakhir adalah penjabaran kebutuhan non fungsional. Analisis kebutuhan ini dilakukan untuk mendapatkan kebutuhan yang harus disediakan oleh sistem agar dapat memenuhi kebutuhan pengguna.

### Gambaran Umum Aplikasi

Penjabaran gambaran umum aplikasimeliputi penjelasan tentang deskripsi umum aplikasi dan lingkungan aplikasi. Berikut penjabarannya :

##### Deskripsi Aplikasi

Sistem yang dibangun pada penelitian ini adalah sebuah sistem presentasi *online* yang dapat mengelola *file-file* presentasi yang terpusat dalam satu server dan menayangkan presentasi secara *online* melalui set-top-box. Sistem ini digunakan sebagai alat bantu pengajar untuk melakukan presentasi di kelas. Pengguna diharapkan dapat mengelola *file-file* presentasi agar tersimpan lebih teratur, yaitu dapat mengelompokkan *file-file* berdasarkan mata kuliah dan saling berbagi *file* dengan anggota *team teaching*. Pengguna juga diharapkan dapat mengakses *file-file* secara *online*, sehingga pengguna dapat mengunggah satu *file* yang dapat digunakan berulang-ulang dan di kelas yang berbeda. Selain itu pengguna juga diharapkan dapat menayangkan presentasi di kelas tanpa perlu membawa laptop pribadi.

Sistem ini dibuat dalam lingkup PTIIK dimana pengguna dalam hal ini dosen tidak perlu mendaftarkan diri untuk dapat menggunakan sistem. Selama pengguna terdaftar sebagai dosen di PTIIK Universitas Brawijaya dan telah memiliki *e-mail* UB, maka pengguna bisa melakukan *sign in* ke dalam sistem. Untuk melakukan *sign in* pertama kali, *username* dan *password* yang digunakan adalah *e-mail* UB pengguna, lalu pengguna akan diminta mengganti *password* dengan kata kunci yang lebih rahasia. *Password* baru ini akan menggantikan *password* yang lama untuk melakukan *login* di waktu lainnnya.

##### Lingkungan Aplikasi

Lingkungan utama tempat aplikasi ini adalah perangkat Android set-top-box yang dipasang di kelas sebagai media untuk menayangkan presentasi. Namun aplikasi ini diharapkan juga dapat berjalan pada *smartphone* Android dengan tujuan agar pengguna dapat mengunggah *file* presentasi secara online maupun mengecek presentasi dengan menjalankan *file* presentasi melalui *smartphone*.

Aplikasi ini membutuhkan koneksi jaringan lokal Brawijaya dan jaringan publik. Jaringan lokal digunakan untuk mengunggah file ke server maupun maupun membuka *file* dari server dengan cepat. Jaringan publik dibutuhkan ketika pengguna berada di luar Brawijaya.

### Identifikasi Aktor

Tahap ini adalah tahap untuk melakukan identifikasi terhadap aktor-aktor yang berinteraksi dengan aplikasi. Tabel 4.1 menjabarkan aktor beserta keterlibatannya dengan aplikasi.

Tabel 4. 1 Identifikasi Aktor

|  |  |
| --- | --- |
| **Aktor** | **Deskripsi** |
| Pengguna | Pengguna merupakan pengajar yang telah melakukan registrasi. Pengguna dapat menggunakan aplikasi untuk menayangkan materi presentasi di kelas melalui set-top-box serta mengunggah *file*  ke sistem. Pengguna juga dapat berbagi konten materi dengan pengguna lainnya untuk membantu kegiatan *team teaching*. Selain itu pengguna juga dapat mengelompokkan materi ke dalam mata kuliah atau kelas. |

### Analisis Kebutuhan Fungsional

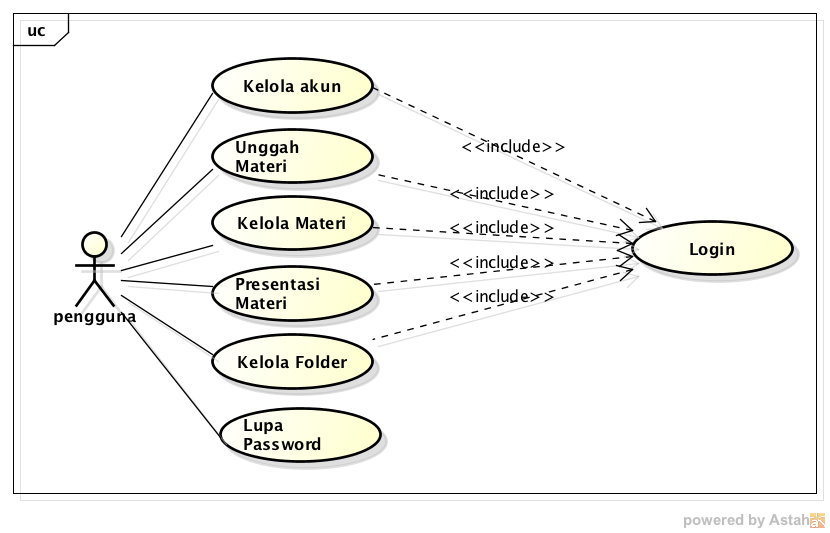
Kebutuhan fungsional dijabarkan dalam bentuk daftar dengan menggunakan nomor SRS (*Software Requirement Spesification)* yang ditunjukkan pada tabel 4.2.

Tabel 4. 2 Spesifikasi Kebutuhan Fungsional

|  |  |  |
| --- | --- | --- |
| Nomor *SRS* | Kebutuhan | *Use Case* |
| SRS\_001\_01 | Aplikasi harus dapat menyediakan fasilitas *login* bagi pengguna. Jika pengguna tidak melakukan login, maka pengguna tidak dapat menggunakan fasilitas apapun kecuali fasilitas lupa *password*. | *Login* |
| SRS\_001\_02 | Aplikasi harus dapat menyediakan fasilitas untuk membuat *password* yang baru ketika pengguna tidak dapat *login* karena tidak bisa memasukkan *password* yang benar. | Lupa *Password* |
| SRS\_001\_03 | Aplikasi harus dapat menyediakan fasilitas mengelola data akun pengguna yaitu melihat dan mengubah data pengguna serta mengubah password akun pengguna. | Kelola Akun |
| SRS\_002\_01 | Aplikasi harus dapat menyediakan fasilitas unggah materi bagi pengguna sehingga pengguna dapat menambahkan materi ke dalam sistem. | Unggah Materi |
| SRS\_002\_02 | Aplikasi harus dapat menyediakan fasilitas untuk mengelola materi, sehingga pengguna dapat menghapus materi, mengganti nama materi dan memindahkan materi dalam folder. | Kelola Materi |
| SRS\_002\_03 | Aplikasi harus dapat menyediakan fasilitas untuk mempresentasikan materi dalam berbagai format media. | Presentasi Materi |
| SRS\_002\_04 | Aplikasi harus dapat menyediakan fasilitas untuk mengelola folder, sehingga pengguna dapat menambah, menghapus, atau mengubah nama *folder*. | Kelola *Folder* |

##### Diagram *Use Case*

Diagram *use case*  merepresentasikan interaksi antara pengguna dengan system. Diagram *use case* terdiri aktor yang merepresentasikan pengguna, use case yang merupakan perilaku system, dan relasi yang menghubungkan aktor dengan *use case.* Diagram *use case* pengguna ditunjukkan pada Gambar 4.2.



Gambar 4. 2 Diagram *Use Case*

##### Skenario *Use Case*

Diagram *use case*  secara lebih mendetail dijabarkan melalui skenario *use case*. Di dalam skenario *use case* diuraikan nama *use case*, tujuan dari *use case,*  gambaran mengenai *use case*,aktor yang memiliki interaksi dengan *use case*, kondisi awal yang harus dipenuhi sebelum *use case* tersebut berjalan, alur utama, alur alternatif dan konidisi akhir yang diharapkan setelah berjalannya *use case*.

###### Skenario *Use Case* *Login*

Kebutuhan fungsional yang harus disediakan oleh sistem adalah pengguna harus dapat login yang direpresentasikan dengan *use case login*. Skenario *use case login* ditunjukkan pada Table 4.3.

Tabel 4. 3 Skenario *Use Case Login*

|  |  |
| --- | --- |
| **Nama** | *Login* |
| **Kode SRS** | SRS\_001\_01 |
| **Tujuan** | Melakukan *Login* |
| **Deskripsi**  ***(Brief Description)*** | *Use case* ini memungkinkan pengguna untuk melakukan *login* untuk mendapatkan hak akses dalam menggunakan sistem. |
| **Aktor** | Pengguna |
| **Kondisi Awal *(Pre-Conditions)*** | Aktor harus menjalankan aplikasi terlebih dahulu. |
| ***Flow of Events*** | |
| **Alur Utama** ***(Basic Flow)*** | |
| *Use case* ini dimulai ketika aktor ingin melakukan *login*.   1. Sistem akan menampilkan halaman *login* yang berisi *form* masukan untuk memasukkan *username* dan *password*. 2. Sistem mengizinkan aktor untuk memasukkan *username* dan *password*. 3. Aktor memasukkan *username* dan *password*. 4. Aktor menekan tombol *login*. 5. Sistem mengecek apakah *username* dan *password* yang dimasukkan aktor sesuai dengan yang tersimpan di dalam *database* sistem. 6. Jika *username* dan *password* yang dimasukkan aktor sesuai dengan yang tersimpan di dalam *database* sistem, maka sistem akan menampilkan halaman utama. | |
| **Alur Alternatif (*Alternative Flows)*** | |
| ***Username* dan *password* tidak sesuai** | 1. *Username* dan *password* yang dimasukkan aktor tidak sesuai dengan yang tersimpan di dalam *database* sistem. 2. Sistem akan menampilkan pesan informasi bahwa *password* yang dimasukkan tidak sesuai. |
| **Kondisi Akhir**  ***(Post-Conditions)*** | Jika *use case* berhasil, aktor dapat masuk ke halaman utama dan melakukan aktivitas yang telah disediakan oleh sistem. |

###### Skenario *Use Case* Lupa *Password*

Kebutuhan fungsional selanjutnya yang harus disediakan oleh sistem adalah kebutuhan *reset password* untuk membuat *password* kembali ketika pengguna tidak dapat melakukan *login* karena tidak dapat memasukkan *password* yang cocok. Skenario *use case* lupa *password* ditunjukkan oleh Tabel 4.4

Tabel 4. 4 Skenario *Use Case* Lupa *Password*

|  |  |
| --- | --- |
| **Nama** | Lupa *Password* |
| **Kode SRS** | SRS\_001\_02 |
| **Tujuan** | Melakukan *reset password* |
| **Deskripsi**  ***(Brief Description)*** | *Use case* ini memungkinkan pengguna untuk membuat *password* yang baru ketika pengguna tidak dapat melakukan *login* karena tidak bisa memasukkan *password* yang benar. |
| **Aktor** | Pengguna |
| **Kondisi Awal *(Pre-Conditions)*** | Aktor harus menjalankan aplikasi terlebih dahulu. |
| ***Flow of Events*** | |
| **Alur Utama** ***(Basic Flow)*** | |
| *Use case* ini dimulai ketika aktor ingin melakukan *reset password*.   1. Aktor memilih menu *forgot* *password*. 2. Sistem akan menampilkan form masukan dan mengizinkan aktor untuk memasukkan *e-mail* yang sebelumnya telah didaftarkan untuk menggunakan aplikasi. 3. Aktor memasukkan *e-mail dan* menekan tombol submit. 4. Sistem akan mengirimkan pesan ke *e-mail* aktor berupa *link* halaman untuk melakukan *reset password*. 5. Aktor menekan *link* untuk membuka halaman *reset password*. 6. Sistem menampilkan halaman *reset password* yang berisi *form* masukan untuk memasukkan *password* yang baru. 7. Aktor memasukkan password yang baru. 8. Aktor menekan tombol *reset password*. 9. Sistem menyimpan perubahan password. 10. Kembali ke alur utama. | |
| **Kondisi Akhir**  ***(Post-Conditions)*** | Jika *use case* berhasil, aktor dapat masuk ke halaman utama dan melakukan aktivitas yang telah disediakan oleh sistem. |

###### Skenario *Use Case* Kelola Akun

Kebutuhan fungsional selanjutnya yang harus disediakan oleh sistem adalah kebutuhan pengguna untuk mengelola akun yang direpresentasikan dengan *use case* kelola akun. Skenario *use case* kelola akun ditunjukkan pada Table 4.5.

Tabel 4. 5 Skenario *Use Case* Kelola Akun

|  |  |
| --- | --- |
| **Nama** | Kelola Akun |
| **Kode SRS** | SRS\_001\_03 |
| **Tujuan** | Mengelola akun |
| **Deskripsi *(Brief Description)*** | *Use case* ini memungkinkan akor untuk dapat melihat dan mengubah data profil akunnya sendiri, yaitu seperti nama lengkap, *username*, *password*, *e-mail.* |
| **Aktor** | Pengguna |
| **Kondisi Awal *(Pre-Conditions)*** | Aktor harus melakukan *login* terlebih dahulu kemudian masuk ke halaman profil. |
| ***Flow of Events*** | |
| **Alur Utama** ***(Basic Flow)*** | |
| *Use case* ini dimulai ketika aktor ingin melihat dan mengubah data profil akun.   1. Sistem akan menampilkan data akun aktor. 2. Jika aktor memilih menu *edit*, maka sistem akan menampilkan *form* masukanyang telah berisi data profil akun aktor. 3. Sistem mengizinkan aktor untuk melakukan perubahan terhadap data aktor. 4. Aktor melakukan perubahan. 5. Setelah aktor memilih menu simpan, maka sistem akan melakukan memperbarui data yang lama dengan data yang baru. | |
| **Kondisi Akhir *(Post-Conditions)*** | Data profil akun yang ditampilkan sesuai dengan yang tersimpan saat perubahan terakhir dilakukan. |

###### Skenario *Use Case* Unggah Materi

Kebutuhan fungsional selanjutnya yang harus disediakan oleh sistem adalah pengguna harus dapat mengunggah materi yang direpresentasikan dengan *use case* unggah materi. Skenario *use case* unggah materi ditunjukkan pada Table 4.6.

Tabel 4. 6 Skenario *Use Case* Unggah Materi

|  |  |
| --- | --- |
| **Nama** | Unggah Materi |
| **Kode SRS** | SRS\_002\_001 |
| **Tujuan** | Mengunggah Materi |
| **Deskripsi *(Brief Description)*** | Use case ini memungkinkan pengguna untuk mengunggah materi. |
| **Aktor** | Pengguna |
| **Kondisi Awal**  ***(Pre-Conditions)*** | Aktor harus melakukan *login* terlebih dahulu dan masuk pada menu *materials* kemudian masuk ke salah satu *folder* mata kuliah. |
| ***Flow of Events*** | |
| **Alur Utama** ***(Basic Flow)*** | |
| *Use case* ini dimulai ketika aktor ingin mengunggah materi.   1. Sistem menampilkan menu *upload*. 2. Sistem akan menampilkan *file manager* dari perangkat. 3. Sistem mengizinkan aktor untuk memilih *file* yang akan diunggah. 4. Aktor memilih file yang akan diupload dan menekan tombol ok. 5. Sistem menampilkan form masukan yang telah berisi nama *file* dan mengizinkan pengguna mengubah nama *file*. 6. Aktor dapat melakukan perubahan atau tidak. 7. Aktor menekan tombol *done*. 8. Setelah aktor mengunggah *file* yang dipilih, *file* tersebut akan disimpan di dalam sistem. | |
| **Alur Alternatif *(Alternative Flows)*** | |
| **Format *file* tidak sesuai** | 1. Sistem akan memberikan pemberitahuan kepada aktor bahwa format *file* yang diunggah oleh aktor tidak *valid*. 2. Sistem akan memberitahu aktor format *file* apa saja yang diizinkan untuk diunggah ke dalam sistem. 3. Sistem mengizinkan aktor untuk menutup notifikasi. 4. Aktor menutup notifikasi. 5. Skenario kembali ke alur utama (*basic flow*). |
| **Ukuran file melebihi batas maksimum** | 1. Sistem akan memberikan pemberitahuan kepada aktor bahwa ukuran *file* yang diunggah melebihi batas maksimum yang diizinkan oleh sistem. 2. Sistem memberikan pemberitahuan kepada aktor berapa ukuran *file* yang diizinkan untuk diunggah ke dalam sistem. 3. Sistem mengizinkan aktor untuk menutup notifikasi. 4. Aktor menutup notifikasi. 5. Skenario kembali ke alur utama (*basic flow*). |
| **Kondisi Akhir**  ***(Post-Conditions)*** | Jika *use case* berhasil, *file* materi akan ditambahkan ke dalam sistem. |

###### Skenario *Use Case* Kelola Materi

Kebutuhan fungsional selanjutnya yang harus disediakan oleh sistem adalah pengguna harus dapat mengelola materi yang direpresentasikan dengan *use case* kelola materi. Skenario *use case* kelola materi ditunjukkan pada Tabel 4.7.

Tabel 4. 7 **Skenario *Use Case* Kelola Materi**

|  |  |
| --- | --- |
| **Nama** | Kelola Materi |
| **Kode SRS** | SRS\_002\_002 |
| **Tujuan** | Mengelola Materi |
| **Deskripsi *(Brief Description)*** | Use case ini memungkinkan pengguna untuk mengelola materi, sehingga pengguna dapat mengunggah materi, menghapus materi, mengganti nama materi dan mengelompokkan materi dalam folder. |
| **Aktor** | Pengguna |
| **Kondisi Awal *(Pre-Conditions)*** | Aktor harus melakukan *login* terlebih dahulu kemudian masuk pada menu *materials*. |
| ***Flow of Events*** | |
| **Alur Utama** ***(Basic Flow)*** | |
| *Use case* ini dimulai ketika aktor ingin mengubah nama materi, menghapus materi, atau memindahkan materi ke dalam folder.   1. Aktor memberikan pilihan materi yang akan dikelola menekan kotak centang pada salah satu materi. 2. Sistem akan menampilkan tombol-tombol menu kelola materi, yaitu tombol *edit*, *delete* dan *move to*. 3. Sistem akan menjalankan fungsi sesuai dengan perilaku aktor.    1. Jika aktor menekan tombol *edit* maka alur “ubah nama materi” akan dijalankan.    2. Jika aktor menekan tombol *delete* maka alur “hapus materi” akan dijalankan.    3. Jika aktor menekan tombol *move to* maka alur pindah *folder* akan dijalankan. | |
| **Alur Bagian *(Sub Flow)*** | |
| **Ubah Nama Materi** | 1. Sistem akan menampilkan *form* inputan yang telah berisi nama materi 2. Sistem mengizinkan aktor untuk mengubah nama *file* materi. 3. Aktor mengubah nama *file* materi. 4. Aktor menekan tombol *save*. 5. Setelah aktor berhasil melakukan perubahan nama maka sistem memperbarui nama *file* materi dengan nama baru. |
| **Hapus Materi** | 1. Sistem akan meminta aktor untuk mengkonfirmasi penghapusan *file* materi yang telah dipilih untuk dihapus. 2. Aktor mengkonfirmasi penghapusan. 3. Sistem akan menghapus *file* materi dari sistem. |
| **Pindah *Folder*** | 1. Sistem menampilkan daftar folder yang dimiliki oleh aktor 2. Sistem mengizinkan aktor untuk memberikan pilihan folder yang menjadi tujuan. 3. Aktor memberikan pilihan folder. 4. Aktor menekan tombol ok. 5. Sistem memindahkan materi ke folder yang telah dipilih. |
| **Alur Alternatif *(Alternative Flows)*** | |
| **Hapus Materi Dibatalkan** | Jika pada *sub flow* “Hapus Materi” aktor memutuskan untuk tidak menghapus materi, maka penghapusan akan dibatalkan, dan kembali ke awal *basic flow*. |
| **Kondisi Akhir**  ***(Post-Conditions)*** | Jika *use case* berhasil, *file* materi akan terubah namanya, terhapus dari sistem atau berpindah ke dalam folder yang dipilih. |

###### Skenario *Use Case* Presentasi Materi

Kebutuhan fungsional selanjutnya yang harus disediakan oleh sistem adalah pengguna harus dapat menayangkan materi yang direpresentasikan dengan *use case* presentasi materi. Skenario *use case* presentasi materi ditunjukkan pada Tabel 4.8.

Tabel 4. 8 Skenario *Use Case* Presentasi Materi

|  |  |
| --- | --- |
| **Nama** | Presentasi Materi |
| **Kode SRS** | SRS\_002\_03 |
| **Tujuan** | Mempresentasikan materi |
| **Deskripsi *(Brief Description)*** | *Use case* ini memungkinkan pengguna untuk mempresentasikan materi dalam berbagai format media. |
| **Aktor** | Pengguna |
| **Kondisi Awal *(Pre-Conditions)*** | Aktor harus melakukan *login* terlebih dahulu kemudian masuk pada menu *materials*. |
| ***Flow of Events*** | |
| **Alur Utama** ***(Basic Flow)*** | |
| *Use case* ini dimulai ketika aktor ingin mempresentasikan materi.   1. Sistem menampilkan daftar materi yang dimiliki oleh aktor. 2. Aktor memberikan pilihan materi yang akan dipresentasikan dengan mengklik nama materi. 3. Sistem akan menjalankan fungsi sesuai dengan jenis *file* materi.    1. Jika jenis *file* materi memiliki formatPDF, maka alur bagian "format PDF" akan dijalankan.    2. Jika jenis *file* materi memiliki format PPTX, maka alur bagian "format PPTX" akan dijalankan.    3. Jika jenis *file* materi memiliki format video, maka alur bagian "format video" akan dijalankan | |
| **Alur Bagian *(Sub Flow)*** | |
| **Format PDF** | 1. Sistem akan menampilkan halaman pertama materi dan menu-menu penayangan (*previous*, *next*, dan *close*). 2. Sistem mengizinkan aktor untuk memilih menu. 3. Sistem akan menjalankan fungsi sesuai dengan menu yang dipilih oleh aktor.    1. Jika aktor memilih *previous*, maka sistem akan menampilkan halaman sebelumnya.    2. Jika aktor memilih *next*, maka sistem akan menampilkan halaman berikutnya.    3. Jika aktor memilih *close*, maka sistem akan menutup file dan kembali ke alur utama (*basic flow*). |
| **Format PPTX** | 1. Sistem akan menampilkan halaman pertama materi dan menu-menu penayangan (*previous*, *next*, dan *close*). 2. Sistem mengizinkan aktor untuk memilih menu. 3. Sistem akan menjalankan fungsi sesuai dengan menu yang dipilih oleh aktor.    1. Jika aktor memilih *previous*, maka sistem akan menampilkan halaman sebelumnya.    2. Jika aktor memilih *next*, maka sistem akan menampilkan halaman berikutnya.    3. Jika aktor memilih *close*, maka sistem akan menutup file dan kembali ke alur utama (*basic flow*). |
| **Format Video** | 1. Sistem akan menampilkan halaman pertama materi dan menu-menu penayangan (*play*, *pause*, dan *close*). 2. Sistem mengizinkan aktor untuk memilih menu. 3. Sistem akan menjalankan fungsi sesuai dengan menu yang dipilih oleh aktor.    1. Jika aktor memilih *previous*, maka sistem akan menampilkan halaman sebelumnya.    2. Jika aktor memilih *next*, maka sistem akan menampilkan halaman berikutnya.    3. Jika aktor memilih *close*, maka sistem akan menutup file dan kembali ke alur utama (*basic flow*). |
| **Kondisi Akhir *(Post-Conditions)*** | Jika *use case* berhasil, sistem dapat menampilkan *file* materi sesuai dengan perintah yang diberikan oleh aktor. |

###### Skenario *Use Case* Kelola *Folder*

Kebutuhan fungsional selanjutnya yang harus disediakan oleh sistem adalah pengguna harus dapat mengelola *folder* yang direpresentasikan dengan *use case* kelola *folder*. Skenario *use case* kelola *folder* ditunjukkan pada Tabel 4.9.

Tabel 4.9 Skenario *Use Case* Kelola *Folder*

|  |  |
| --- | --- |
| **Nama** | Kelola *Folder* |
| **Kode SRS** | SRS\_002\_004 |
| **Tujuan** | Mengelola *Folder* |
| **Deskripsi**  ***(Brief Description)*** | *Use case* ini memungkinkan aktor untuk menambah, menghapus, atau mengubah nama *folder*. |
| **Aktor** | Pengguna |
| **Kondisi Awal**  ***(Pre-Conditions)*** | Aktor harus melakukan *login* terlebih dahulu kemudian masuk pada menu *materials*. |
| ***Flow of Events*** | |
| **Alur Utama** ***(Basic Flow)*** | |
| *Use case* ini dimulai ketika aktor ingin menambah, menambah dan menghapus *folder* serta memindahkan *file* materi ke sebuah *folder*.   1. Sistem menampilkan daftar mata kuliah yang diampu pengguna. 2. Aktor memilih salah satu mata kuliah. 3. Sistem akan menampilkan daftar materi-materi dan folder yang dimiliki mata kuliah yang dipilih. 4. Sistem mengizinkan aktor untuk menentukan tombol yang akan ditekan (*add folder* atau *checkbox* pada *folder*). 5. Sistem akan menjalankan fungsi sesuai dengan perilaku aktor.    1. Jika aktor menekan tombol *add folder*, maka alur bagian “tambah *folder*” akan dijalankan.    2. Jika aktor menekan tombol *checkbox* dan *edit*, maka alur bagian “ubah nama *folder*” akan dijalankan.    3. Jika aktor menekan tombol *checkbox* dan *delete*, maka alur bagian “hapus *folder*” akan dijalankan. | |
| **Alur Bagian *(Sub Flow)*** | |
| **Tambah Folder** | 1. Sistem akan memberikan *form* masukan untuk memberi nama folder. 2. Aktor memberikan nama untuk *folder* yang akan dibuat. 3. Sistem mengizinkan aktor untuk menekan tombol *create*. 4. Aktor menekan tombol *create*. 5. Sistem menambahkan folder ke dalam sistem. |
| **Ubah Nama *Folder*** | 1. Sistem akan menampilkan *form* inputan yang telah berisi nama *folder*. 2. Aktor mengubah nama *folder*. 3. Sistem mengizinkan pengguna menekan tombol *save*. 4. Aktor menekan tombol *save*. 5. Sistem memperbarui nama *folder* dengan nama yang baru. |
| **Hapus *Folder*** | 1. Sistem akan meminta aktor untuk mengkonfirmasi penghapusan *folder* yang telah dipilih untuk dihapus. 2. Aktor mengkonfirmasi penghapusan. 3. Sistem akan menghapus *folder* dari sistem. |
| **Alur Alternatif *(Alternative Flows)*** | |
| **Hapus *Folder* Dibatalkan** | Jika pada *sub flow* “Hapus *Folder*” aktor memutuskan untuk tidak menghapus *folder*, maka penghapusan akan dibatalkan, dan kembali ke awal *basic flow*. |
| **Kondisi Akhir**  ***(Post-Conditions)*** | Jika *use case* berhasil, *folder* akan ditambahkan atau dihapus dari sistem. Jika tidak, keadaan sistem tidak akan berubah. |

### Analisis Kebutuhan Non-Fungsional

Analisis kebutuhan fungsional didapatkan dengan menganalisis kebutuhan apa saja yang menentukan parameter yang dapat digunakan untuk menilai pengoperasian dari sistem. Hal ini sangat berbeda dengan analisis kebutuhan fungsional yang lebih berfokus pada penentuan kebutuhan perilaku atau fungsi sistem. Tabel 4.10 di bawah ini menjelaskan spesifikasi kebutuhan non-fungsional dari sistem :

Tabel 4. 10 Spesifikasi Kebutuhan Non-Fungsional

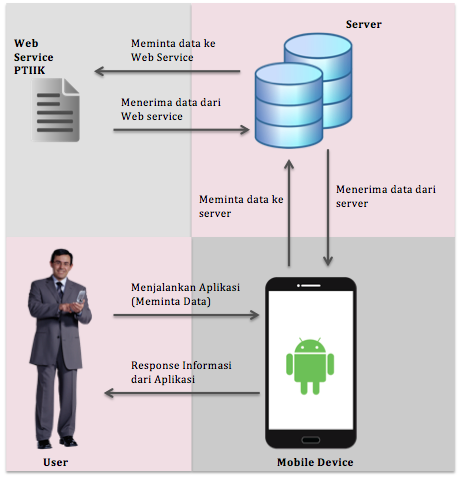
|  |  |
| --- | --- |
| Parameter | Deskripsi Kebutuhan |
| Compatibility | Aplikasi harus dapat berfungsi dengan baik ketika dijalankan di berbagai perangkat set-top-box dan *smartphone* versi Android dengan versi minimal *Honeycomb*, yaitu Android versi 3.0. |

## Perancangan

Proses perancangan terdiri dari dua bagian, yaitu perancangan umum sistem dan perancangan perangkat lunak. Perancangan umum sistem menjelaskan gambaran proses kerja sistem secara umum. Sedangkan proses perancangan perangkat lunak dibagi lagi menjadi empat bagian, yaitu perancangan arsitektural, pemodelan *activity diagram*, perancangan basis data dan perancangan antarmuka.

### Perancangan Umum Sistem

Perancangan umum sistem dilakukan untuk merepresentasikan arsitektur sistem yang akan dibuat secara umum. Arsitektur umum sistem terkait dengan proses kerja objek-objek yang menyusun sistem, yaitu pengguna, *web service* dari PTIIK yang berisi data dosen dan mata kuliah, *server* yang berisi penyimpanan data (MySQL) dan media yang digunakan. Proses kerja sistem dimulai dengan pengguna menjalankan aplikasi dan meminta data melalui aplikasi. Aplikasi akan meminta data ke server, lalu server akan meminta data dari *web service*, *web service* memberikan kembalian data, kemudian server mengolah data tersebut dan memberikan data yang diminta ke aplikasi. Kemudian aplikasi akan memberikan respon informasi kepada pengguna sehingga pengguna mendapatkan data yang diminta.



### Perancangan Perangkat Lunak

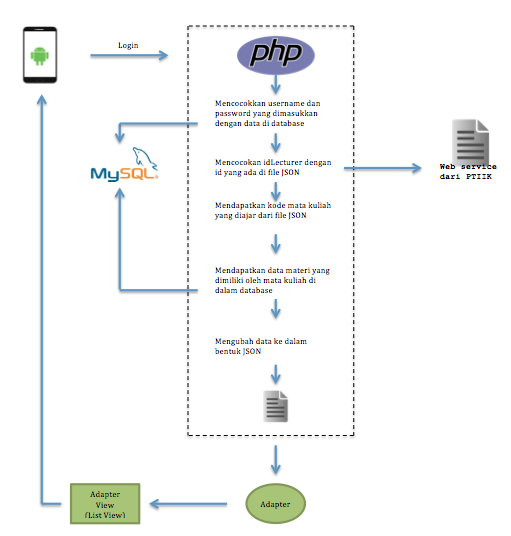
Perancangan perangkat lunak dimulai dengan perancangan arsitektur sistem, perancangan alur aktifitas pengguna dengan sistem yang dimodelkan dengan *activity diagram*, perancangan basis data yang dimodelkan dengan *Entity Relationship Diagram* (ERD), dan tahap perancangan yang terakhir adalah perancangan antarmuka.

##### Perancangan Arsitektur Sistem

Sistem terdiri dari aplikasi *client* dan aplikasi *server*. Perangkat lunak pada sisi *client* dibangun menggunakan konsep *native* Android *Application*. Teknik penyimpanan data menggunakan *remote server* dengan DBMS MySQL. Untuk dapat mendapatkan data dari basis data, dibutuhkan *script* PHP yang meminta data ke basis data, kemudian mengubahnya dalam bentuk JSON, lalu mengirimkan data dalam bentuk JSON ke aplikasi *client* yang menggunakan bahasa pemrograman Java.

##### Perancangan Komunikasi Data dengan Web Service

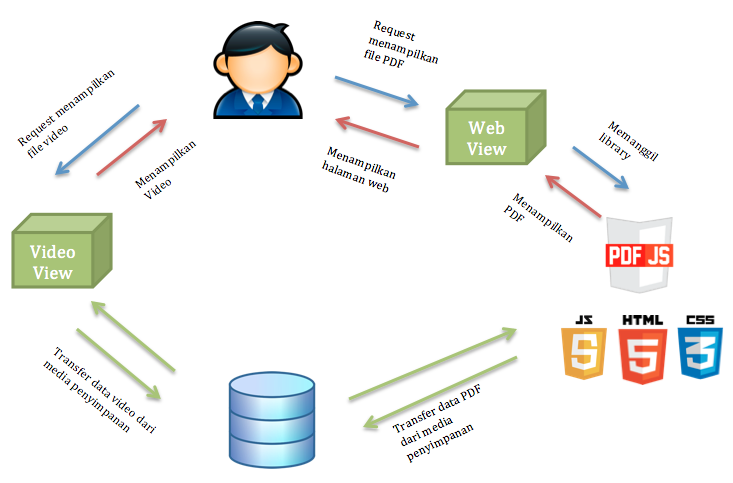
Sistem ini menggunakan web service dari PTIIK untuk mendapatkan informasi mengenai data dosen seperti id dosen dan mata kuliah apa saja yang diampu oleh dosen tersebut. Web service dari PTIIK memberikan kembalian berupa file json dari data jadwal pengajaran pada halaman website PTIIK, yaitu http://filkom.ub.ac.id/info/jadwal.



##### Perancangan Pemutar Materi Kuliah

Sistem ini dapat memutar presentasi dalam bentuk video dan PDF. Presentasi dengan bentuk video ditangani oleh Android API *class*, yaitu VideoView. Sedangkan untuk presentasi dalam bentuk PDF ditangani oleh PDF.js *library*. Perancangan pemutar materi kuliah ini ditunjukkan oleh Gambar 4.xx. Dalam gambar tersebut ditunjukkan bahwa ketika pengguna memilih untuk memutar presentasi video, maka aplikasi akan memanggil kelas VideoView. *File* video disimpan di suatu tempat di dalam *server*. Sehingga untuk mengatur video mana yang akan ditampilkan, kelas VideoView memiliki fungsi setVideoURI yang parameternya diisi dengan nilai String dari *path* dimana video tersebut disimpan di dalam server. Setelah *path* video diatur, video dapat ditampilkan dengan memanggil fungsi start dari kelas VideoView.

Sedangkan jika pengguna memilih untuk memutar presentasi PDF, maka aplikasi akan membutuhkan *library* PDF.js. Namun PDF.js merupakan *library* yang berjalan pada lingkungan *website*, sehingga tidak dapat langsung dipanggil di oleh aplikasi. Untuk itu diperlukan WebView Android API *class*. WebView dapat menampilkan konten *online* sebuah halaman web. Maka, ketika pengguna memilih untuk memutar presentasi PDF, yang pertama kali dipanggil adalah kelas WebView, kemudian kelas WebView akan memanggil halaman tampilan PDF.js, dimana PDF.js akan memproses tampilan file PDF.

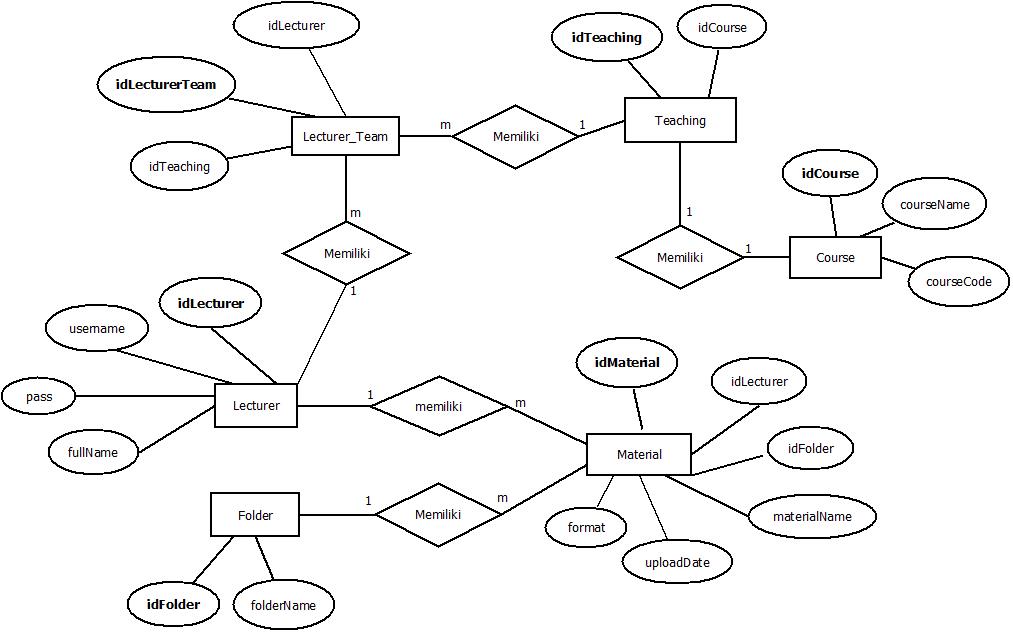


##### Perancangan Basis Data

Penyimpanan data pada aplikasi ini terpusat pada basis data dalam satu server. Perancangan basis data dimodelkan dalam bentuk *Entity Relationship Diagram*. ERD terdiri dari entitas, atribut, dan relasi antar entitas. Entitas merupakan objek dan disusun oleh atribut-atribut yang dapat merepresentasikannya. Entitas dapat memiliki relasi dengan entitas lainnya, yaitu berupa relasi *one-to-one*, *one-to-many*, atau *many-to-many*.

Penentuan rancangan entitas, atribut dan relasi didapat dari hasil analisis yang mengacu pada analisis kebutuhan sistem. Entitas yang dibutuhkan oleh sistem ini adalah dosen sebagai pengguna sistem, mata kuliah yang diampu oleh dosen, kelompok dosen yang mengajar mata kuliah yang sama dan materi yang akan dipresentasikan oleh dosen. Sehingga dibentuk 6 entitas dalam basis data yaitu entitas *Lecturer*, *Teaching*, *Lecturer\_Team,* *Course*, *Material, dan Folder*.

Representasi hubungan entitas pada sistem ini ditnjukkan oleh gambar 4.11 di bawah ini :



Gambar 4. 3 *Entity Relational Diagram*

Entitas *Lecturer* berisi atribut-atribut yang menyimpan data pengguna dan memiliki idLecturer yang bertindak sebagai *primary key*.Entitas *Material* berisi atribut-atribut yang menyimpan data materi dan memiliki satu atribut dari entitas *Lecturer* yaitu *idLecturer* yang menunjukan kode pengguna yang mengunggah materi tersebut. Atribut *idLecturer* ini berfungsi sebagai kunci relasi dari entitas *Lecturer* dan *Material* yang memiliki relasi *one-to-many*, karena satu dosen dapat mengunggah banyak materi sedangkan satu *idMaterial* pasti hanya diunggah oleh satu dosen. Entitas *Folder* memiliki berisi atribut-atribut yang menyimpan data *folder*. Entitas ini memiliki relasi *one-to-many* dengan entitas *Material* karena satu materi hanya berada pada satu *folder*, sedangkan satu *folder* bisa memiliki banyak materi. Entitas *Course* berisi atribut-atribut yang menyimpan data mata kuliah dan memiliki atribut *idCourse* sebagai *primary key*. Entitas *Course* memiliki relasi dengan entitas *Teaching* dengan atribut *idCourse* bertindak sebagai *foreign key di dalam* entitas *Teaching.* Entitas *Teaching* berisi atribut *idTeaching* dan *idCourse*. Atribut *idTeaching* hanya akan dibuat sejumlah mata kuliah yang ada pada sistem, sehingga relasi dari kedua entitas ini adalah *one-to-one*. Entitas yang terakhir adalah entitas *Lecturer\_Team*. Entitas ini menunjukkan kelompok pengajaran dosen yang mengampu mata kuliah yang sama, sehingga entitas ini memiliki relasi dengan entitas *Lecturer* dan *Course*. Hubungan ketiga entitas tersebut ditunjukkan dengan keberadaan kedua *primary key* dari entitas *Lecturer* dan *Course* di dalam entitas *Lecturer\_*Team, yaitu atribut *idTeaching* dan *idLecturer* yang menjadi *foreign key* pada entitas *Lecturer\_Team*.

Entitas di dalam ERD merepresentasikan tabel di dalam basis data, sedangkan atribut merepresentasikan *field* dari masing-masing tabel. Berikut ini merupakan struktur dari masing-masing tabel serta keterangan tiap *field* yang ada pada basis data. Entitas *Lecturer* merepresentasikan tabel *Lecturer* di dalam basis data. Tabel *Lecturer* melakukan penyimpanan data ketika pengguna melakukan perubahan data akun. Data yang disimpan yaitu berupa kode pengguna, nama lengkap pengguna, serta *username* dan *password* yang digunakan untuk melakukan *login*. Struktur tabel *Lecturer* ditunjukkan pada Tabel 4.11.

Tabel 4. 11 Struktur Tabel *Lecturer*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Field** | **Tipe Data** | **Keterangan** |
| 1 | idLecturer (PK) | Integer | Kode dari pengguna (1,2,3) |
| 2 | username | Varchar | Username pengguna, berupa *e-mail* UB (pengguna@ub.ac.id) |
| 3 | pass | Varchar | Password pengguna (r4h4s1as3k4l1) |
| 4 | fullName | Varchar | Nama lengkap pengguna |
|  |  |  |  |

Entitas *Material* merepresentasikan tabel *Material* yang berisi data-data dari materi yang terdapat di dalam basis data, yaitu kode materi, nama materi, format *file* materi, dan kode pengguna yang mengunggah materi tersebut. Tabel *Material* melakukan penyimpanan data ketika pengguna mengunggah materi ke sistem. Selain itu tabel *Material* juga melakukan penyimpanan data ketika pengguna melakukan aktifitas *edit material* dan melakukan penghapusan data ketika pengguna melakukan aktifitas *delete material*. Struktur tabel *Material* ditunjukkan pada Tabel 4.12.

Tabel 4. 12 Strutur Tabel *Material*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Field** | **Tipe Data** | **Keterangan** |
| 1 | idMaterial (PK) | Integer | Kode dari materi (1,2,3) |
| 2 | materialName | Varchar | Nama materi (OSI Layer) |
| 3 | format | Varchar | Password dari pengguna (r4h4s1as3k4l1) |
| 4 | uploadDate | Datetime | Tanggal materi diunggah |
| 5 | idLecturer (FK) | Integer | Kode dosen yang mengunggah materi dan merupakan *foreign key* |
| 6 | idFolder | Integer | Kode *folder* tempat materi berada (1,2,3) |

Entitas *Folder* merepresentasikan tabel *Folder* yang berisi kode *folder* dan nama folder. Tabel *Folder* melakukan penyimpanan data ketika pengguna mengunggah materi atau memindahkan materi ke *folder* lainnya. Struktur tabel *Folder* ditunjukkan pada Tabel 4.13.

Tabel 4. 13 Struktur Tabel *Folder*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Field** | **Tipe Data** | **Keterangan** |
| 1 | idFolder | Integer | Kode dari *folder* (1,2,3) |
| 2 | folderName | Varchar | Nama *folder* (Kumpulan Jurnal Fuzzy Metode Tsukamoto) |

Entitas *Course* merepresentasikan tabel *Course* yang berisi data mata kuliah, yaitu kode mata kuliah yang dibuat oleh sistem, nama mata kuliah, dan kode mata kuliah yang menjadi standar di PTIIK. Struktur tabel *Course* ditunjukkan pada Tabel 4.14.

Tabel 4. 14 Struktur Tabel *Course*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Field** | **Tipe Data** | **Keterangan** |
| 1 | idCourse (PK) | Integer | Kode dari mata kuliah yang dibuat oleh sistem (1,2,3) |
| 2 | courseName | Varchar | Nama mata kuliah (Keamanan Jaringan) |
| 3 | courseCode | Varchar | Kode dari mata kuliah yang menjadi standar di PTIIK (KOM15008) |

Entitas *Teaching* merepresentasikan tabel *Teaching* pada basis data yang berisi kode pengajaran dan kode mata kuliah. Struktur tabel *Teaching* ditunjukkan pada Tabel 4.15.

Tabel 4. 15 Struktur Tabel *Teaching*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Field** | **Tipe Data** | **Keterangan** |
| 1 | idTeaching (PK) | Integer | Kode pengajaran (1,2,3) |
| 2 | idCourse (FK) | Integer | Kode dari mata kuliah yang menjadi *foreign key* (1,2,3) |

Entitas *Lecturer\_Team* merepresentasikan tabel *Lecturer\_Team* pada basis data yang berisi kode *Lecturer\_Team*, kode *Teaching* yang bertindak *foreign key* dari tabel *Teaching*, dan kode *Lecturer* yang bertindak *foreign key* dari Tabel *Lecturer*.

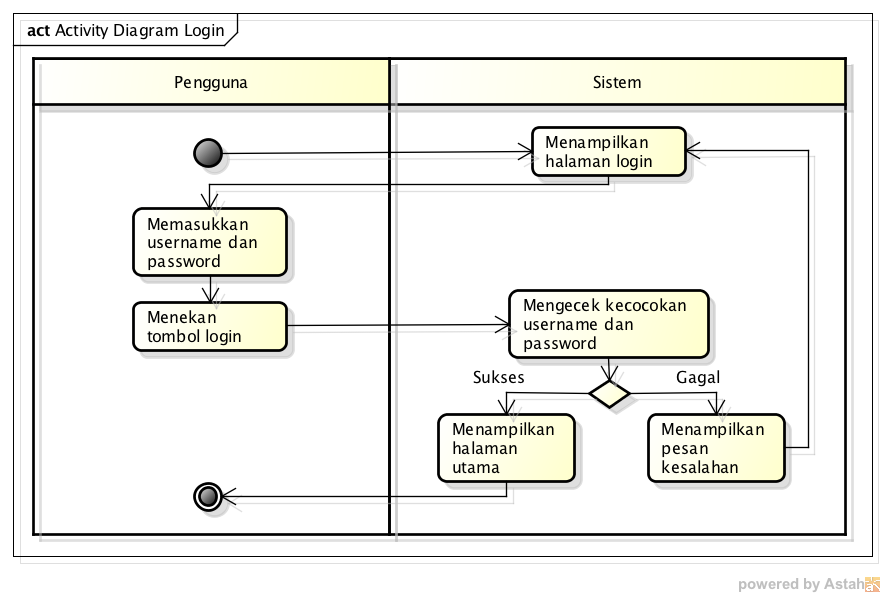
Tabel 4. 16 Struktur Tabel *Lecturer\_Team*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Field** | **Tipe Data** | **Keterangan** |
| 1 | idLecturerTeam | Integer | Kode dari pengguna (1,2,3) |
| 2 | idTeaching | Integer | Kode pengajaran dari tabel Teaching yang bertindak sebagai *foreign key* pada tabel Lecturer\_Team |
| 3 | idLecturer | Integer | Kode pengguna dari tabel Lecturer yang bertindak sebagai *foreign key* pada tabel LecturerTeam |

##### Perancangan *Activity Diagram*

*Activity diagram* merepresentaikan alur aktifitas antara pengguna dengan sistem yang mengacu pada scenario *use case* yang telah dibuat sebelumnya.

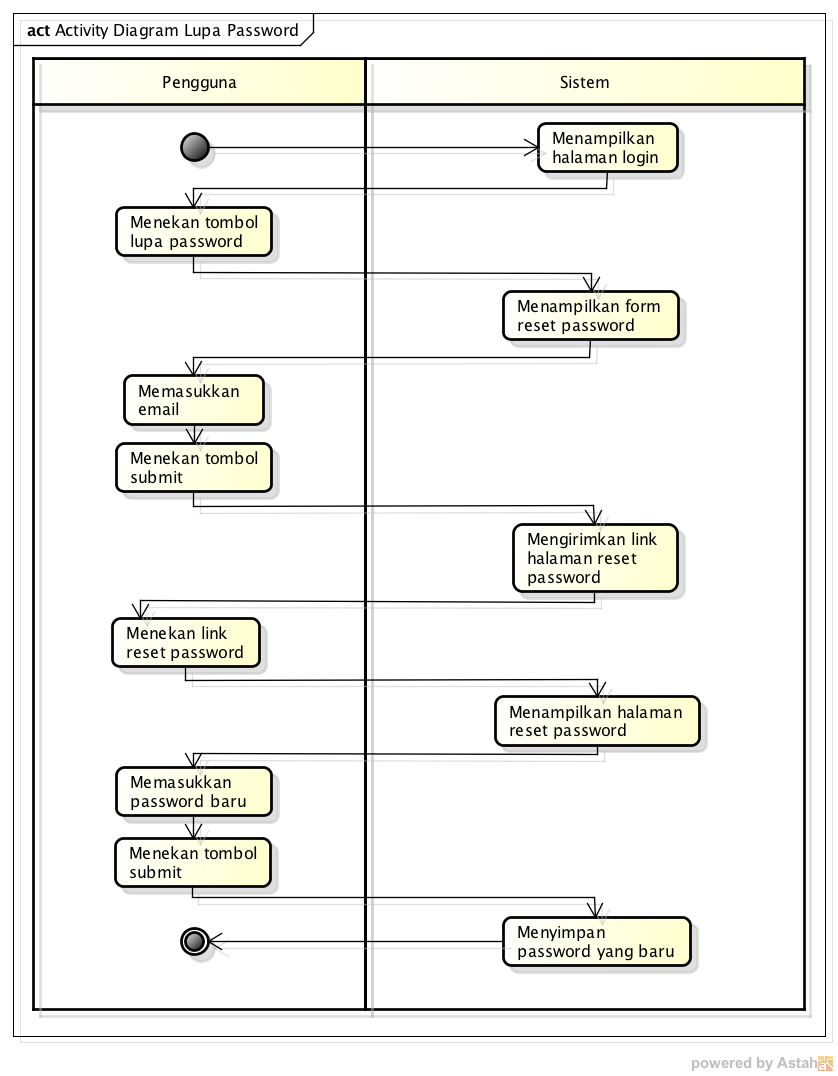
###### *Activity Diagram Login*



Gambar 4. 4 *Activity Diagram Login*

*Activity diagram login* dibuat sesuai dengan skenario pada *use case login* yang ditunjukkan pada Tabel 4.3. Dalam Gambar 4.3 ditunjukkan interaksi yang dilakukan oleh pengguna dan sistem dalam melakukan aktifitas *login*, yaitu diawali ketika pengguna menjalankan aplikasi, selanjutnya sistem akan menampilkan halaman *login*. Pengguna akan memasukkan *usename* dan *password*, lalu sistem akan mengecek kecocokan *username* dan *password* yang dimasukkan oleh pengguna dengan yang tersimpan di dalam sistem.

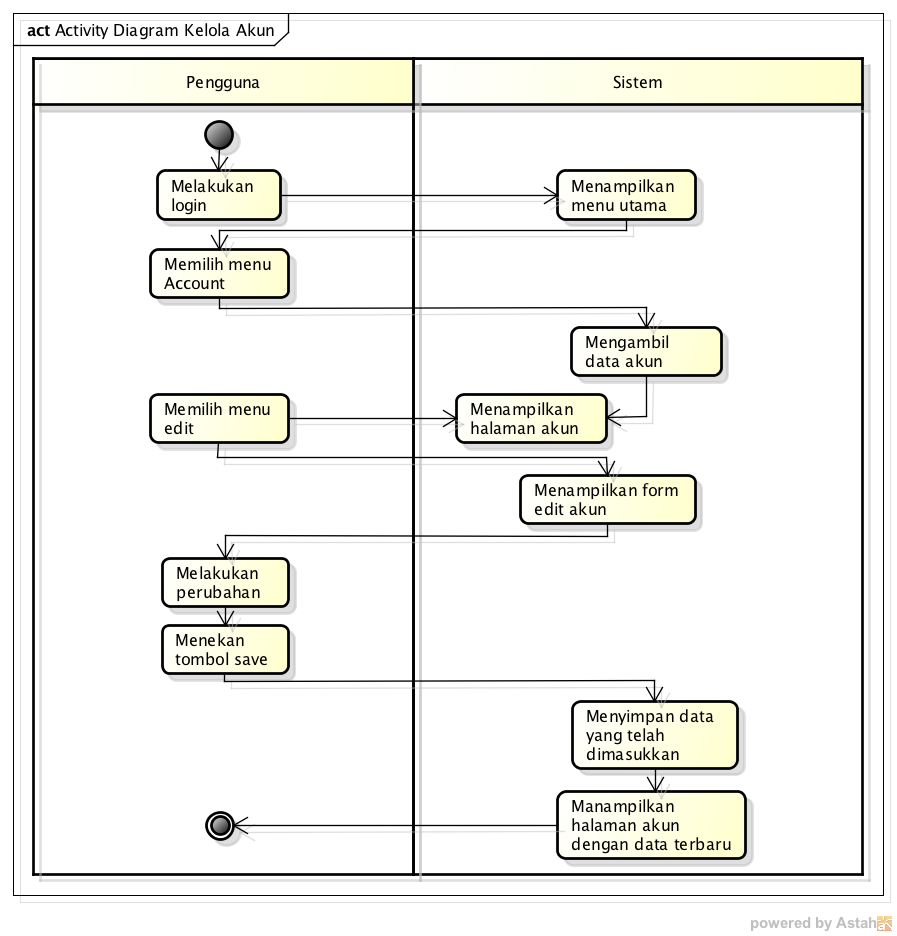
###### *Activity Diagram* Lupa *Password*



Gambar 4. 5 *Activity Diagram* Lupa *Password*

*Activity diagram* lupa *password* dibuat sesuai dengan skenario pada *use case* lupa *password* yang ditunjukkan pada Tabel 4.4. Dalam Gambar 4.4 ditunjukkan interaksi yang dilakukan oleh pengguna dan sistem dalam melakukan aktifitas lupa *password*, yaitu diawali ketika pengguna menjalankan aplikasi, selanjutnya sistem akan menampilkan halaman *login*. Pengguna memilih menu *forgot password* kemudian sistem akan menjalankan prosedur yang dibutuhkan agar pengguna dapat membuat *password* baru untuk melakukan *login.*

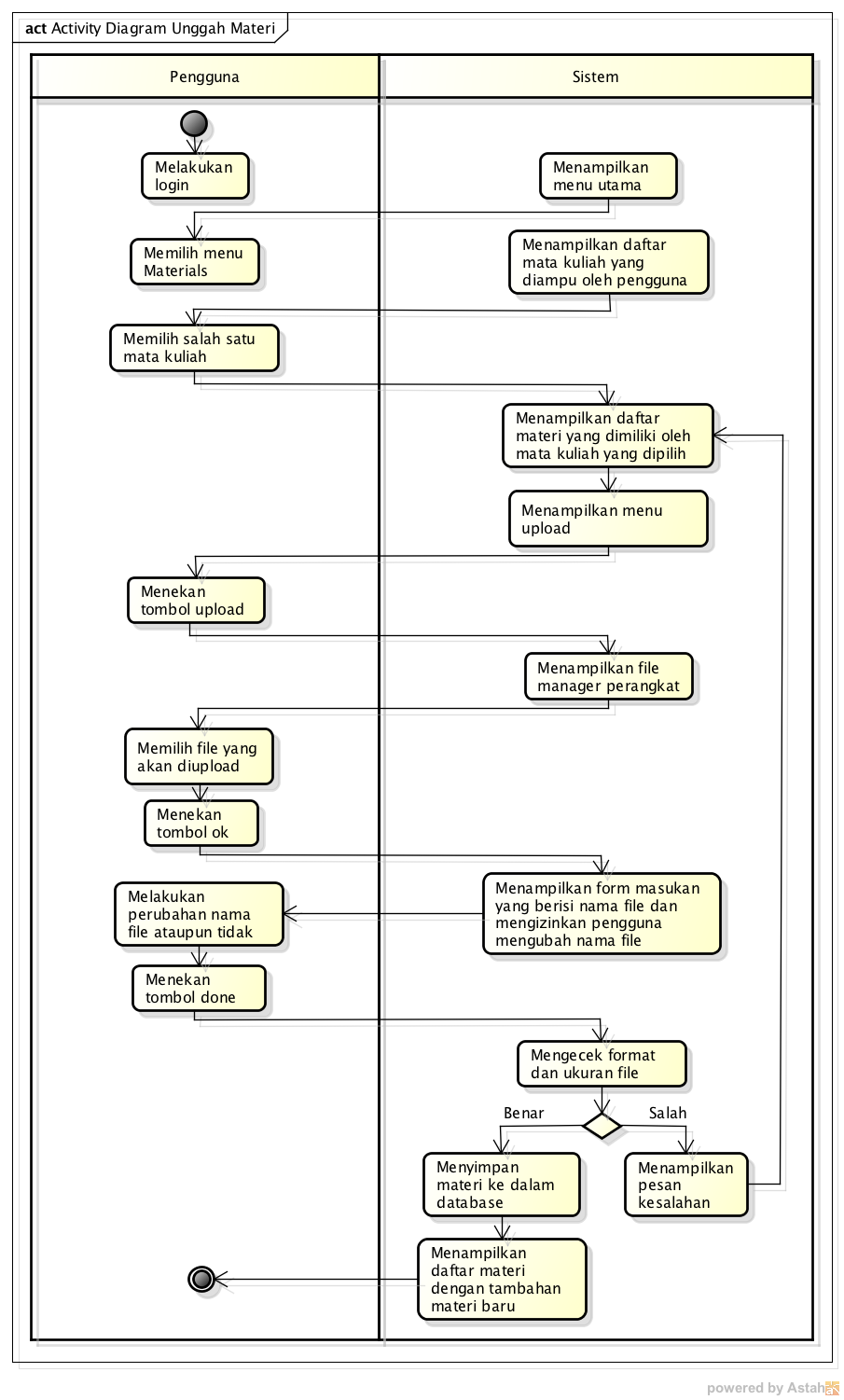
###### *Activity Diagram* Kelola Akun



Gambar 4. 6 *Activity Diagram* Kelola Akun

*Activity diagram* kelola akundibuat sesuai dengan skenario pada *use case* kelola akunyang ditunjukkan pada Tabel 4.5. Dalam Gambar 4.5 ditunjukkan interaksi yang dilakukan oleh pengguna dan sistem dalam melakukan aktifitas kelola akun, yaitu diawali ketika pengguna melakukan login dan memilih menu akun, selanjutnya sistem akan mengambil data akun lalu menampilkan data akun dan menu *edit* akun.

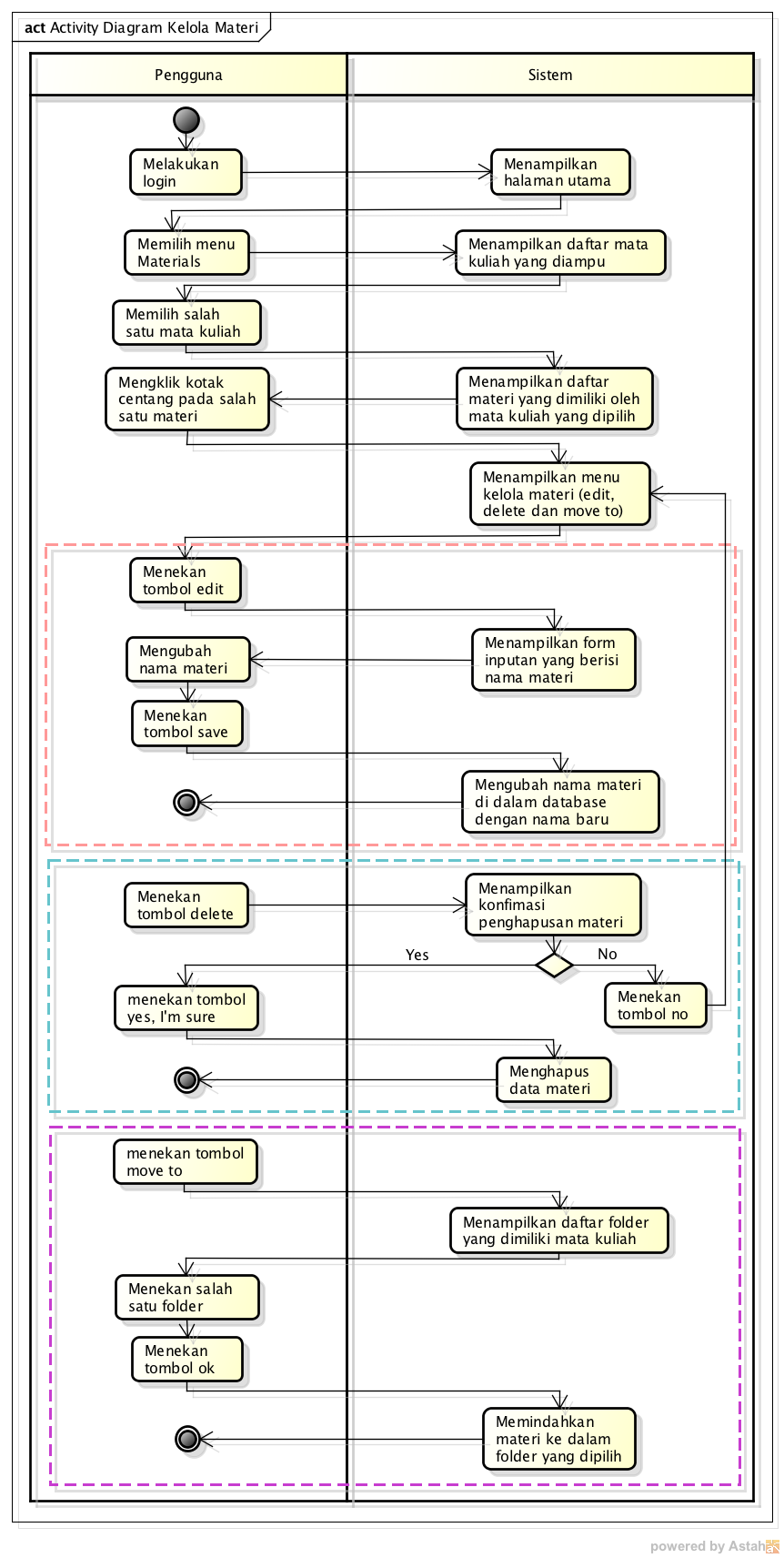
###### *Activity Diagram* Unggah Materi

******

Gambar 4. 7 *Acitvity Diagram* Unggah Materi

*Activity diagram* unggah materidibuat sesuai dengan skenario pada *use case* unggah materiyang ditunjukkan pada Tabel 4.6. Dalam Gambar 4.6 ditunjukkan interaksi yang dilakukan oleh pengguna dan sistem dalam melakukan aktifitas unggah materi, yaitu diawali ketika pengguna menjalankan aplikasi dan berhasil melakukan login, selanjutnya sistem akan menampilkan halamanutama. Pengguna memilih menu *materials* kemudian menekan tombol *upload*. Selanjutnya sistem akan menjalankan prosedur yang dibutuhkan agar aktifitas unggah materi berhasil dilakukan.

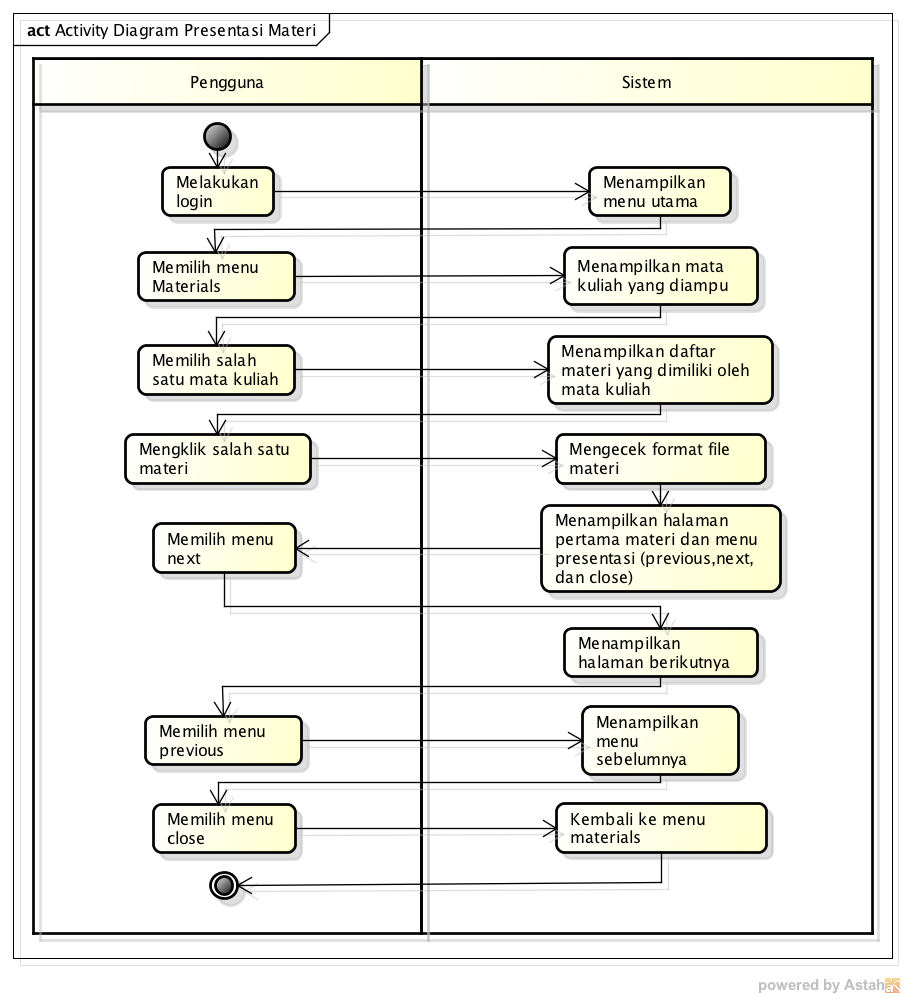
###### *Activity Diagram* Kelola Materi



Gambar 4. 8 *Activity Diagram* Kelola Materi

*Activity diagram* kelola materidibuat sesuai dengan skenario pada *use case* kelola materiyang ditunjukkan pada Tabel 4.7. Dalam Gambar 4.7 ditunjukkan interaksi yang dilakukan oleh pengguna dan sistem dalam melakukan aktifitas kelola materi, yaitu diawali ketika pengguna menjalankan aplikasi dan berhasil melakukan *login*, selanjutnya sistem akan menampilkan halaman halaman utama. Pengguna memilih menu *materials* dan melakukan instruksi sesuai dengan menu yang diberikan, yaitu mengubah nama materi, menghapus materi, dan memindahkan materi ke *folder* lain.

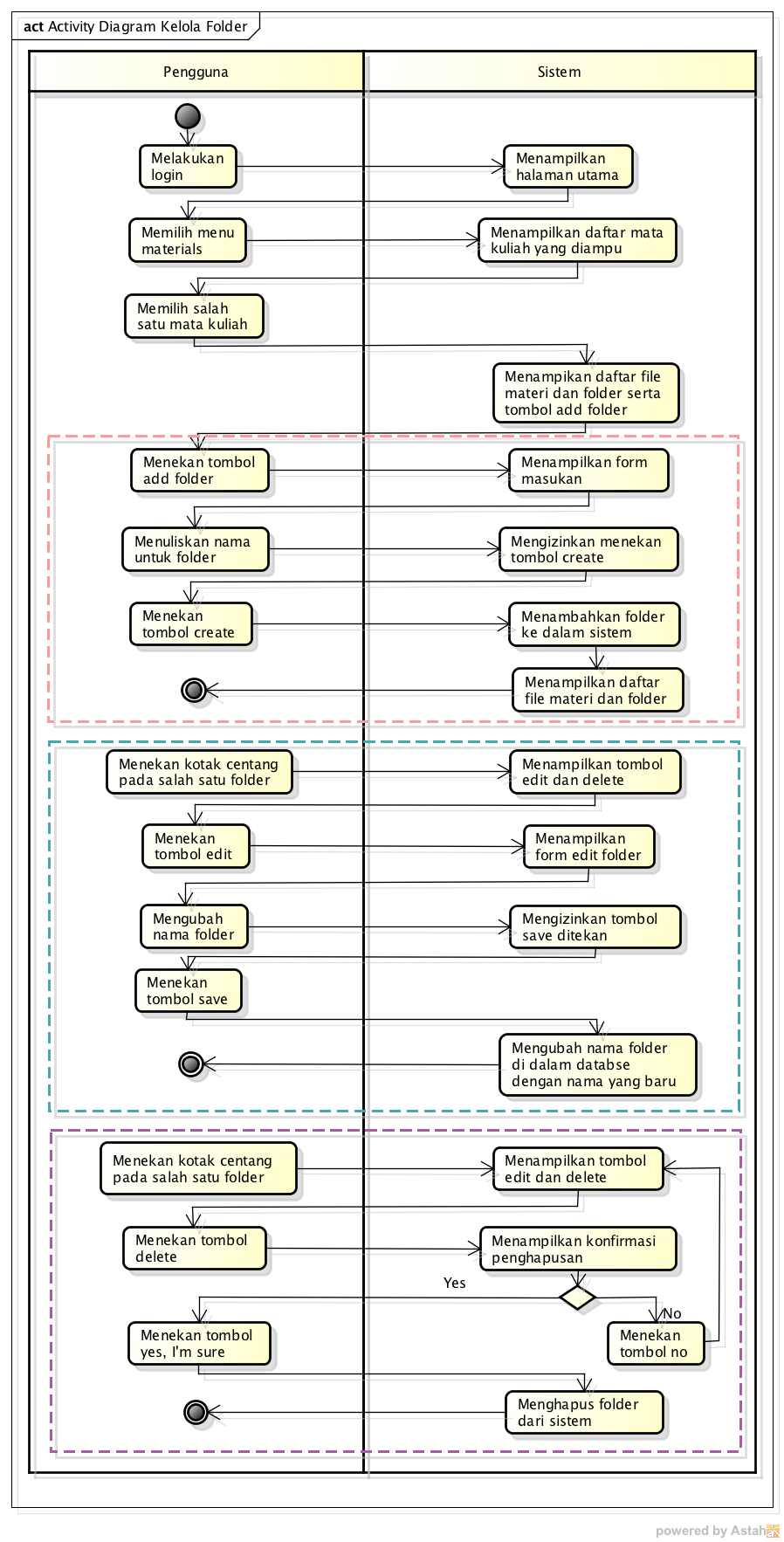
###### *Activity Diagram* Presentasi Materi



Gambar 4. 9 *Activity Diagram* Presentasi Materi

*Activity diagram* presentasi materidibuat sesuai dengan skenario pada *use case* presentasi materiyang ditunjukkan pada Tabel 4.8. Dalam Gambar 4.8 ditunjukkan interaksi yang dilakukan oleh pengguna dan sistem dalam melakukan aktifitas presentasi materi, yaitu diawali ketika pengguna menjalankan aplikasi dan berhasil melakukan *login*, kemudian sistem akan menampilkan halaman utama. Pengguna memilih *menu materials* dan menekan salah satu materi. Sistem selanjutnya akan menjalankan prosedur penayangan materi sesuai dengan format *file* dari materi yang dipilih.

###### *Activity Diagram* Kelola Folder



Gambar 4. 10 *Activity Diagram* Kelola *Folder*

*Activity diagram* kelola *folder* dibuat sesuai dengan skenario pada *use case login* yang ditunjukkan pada Tabel 4.9. Dalam Gambar 4.9 ditunjukkan interaksi yang dilakukan oleh pengguna dan sistem dalam melakukan aktifitas kelola *folder*, yaitu diawali ketika pengguna menjalankan aplikasi dan berhasil melakukan *login*, selanjutnya sistem akan menampilkan halaman utama. Pengguna memilih menu *materials* dan melakukan instruksi sesuai dengan yang disediakan, yaitu menambahkan *folder*, mengubah nama *folder*, dan menghapus *folder*.

##### Perancangan Antar Muka

Proses perancangan antar muka akan menggambarkan perancangan antar muka aplikasi dalam bentuk *layout*.

###### Halaman *Login*

Ketika pengguna menjalankan aplikasi, maka halaman pertama yang akan muncuk adalah halaman *login*. Halaman *login* terdiri dari logo aplikasi, form *login* dan menu *forgot password*. Gambar 4.10 adalah gambar rancangan antarmuka *login*.

|  |  |
| --- | --- |
| Server HD:Users:ditaoktaria:Documents:login.jpg | **Keterangan :**  1. Logo Aplikasi  2. *Tagline* Aplikasi  3. *Form username*  4. *Form password*  5. Tombol *login* |

Gambar 4. 11 Antarmuka Halaman *Login*

###### Antarmuka Halaman Utama

Jika pengguna sukses melakukan *login*, halaman pertama yang akan ditampilkan adalah halaman utama. Halaman utama menampilkan menu *Materials*, menu *Account*, menu *Upload*, menu *Add Folder*, serta daftar mata kuliah yang dimiliki oleh dosen. Gambar 4.11 adalah gambar perancangan antarmuka halaman utama.

|  |  |
| --- | --- |
| Server HD:Users:ditaoktaria:Documents:home.jpg | **Keterangan :**   * + - 1. Menu *Materials*       2. Menu *Account*       3. Menu *Upload*       4. Menu *Add Folder*       5. Daftar mata kuliah yang dimiliki akun |

Gambar 4. 12 Antarmuka Halaman Utama

###### Antarmuka Halaman Kelola Materi

Halaman kelola materi ditampilkan ketika pengguna mengklik *checkbox* pada materi. Pada halaman ini akan ditampilkan daftar materi yang dimiliki oleh mata kuliah yang dipilih sebelumnya oleh pengguna. Selain itu halaman kelola materi juga menampilkan menu-menu untuk mengelola materi, yaitu *edit, delete,* dan *move to.* Gambar 4.12 adalah gambar perancangan halaman kelola materi.

|  |  |
| --- | --- |
| Server HD:Users:ditaoktaria:Documents:kelola materi.jpg | **Keterangan :**   1. Menu *Materials* 2. Menu *Account* 3. Menu *Upload Materials* 4. Menu *Add Folder* 5. Daftar Materi 6. Menu *Edit* 7. Menu *Delete* 8. Menu *Move to* |

Gambar 4. 13 Antarmuka Halaman Kelola Materi

###### Antarmuka Kelola Akun

Halaman kelola akun ditampilkan ketika pengguna menekan *icon* pengguna pada bagian atas halaman. Terdapat 3 menu dari kelola akun, yaitu *view account, edit account,* dan *sign out*. Gambar 4.13 adalah gambar perancangan halaman kelola akun.

|  |  |
| --- | --- |
| Server HD:Users:ditaoktaria:Documents:akun.jpg | **Keterangan :**   1. Menu *Materials* 2. Menu *Account* 3. Sub-menu *Account* (*view account, edit account*, *sign out*) 4. Menu *Add Folder* 5. Daftar materi |

Gambar 4. 14 Antarmuka Halaman Kelola Akun

###### Antarmuka Halaman *Edit* *Account*

Halaman *edit account* ditampilkan ketika pengguna memilih sub-menu *edit account*. Halaman ini menampilkan *form* masukan yang telah berisi data-data akun dan tombol *save.* Gambar 4.14 adalah gambar perancangan halaman *edit account*.

|  |  |
| --- | --- |
| Macintosh HD:Users:ditaoktaria:Documents:skripsi pak aryo:skripsi dita:Kelola akun.jpg | **Keterangan :**   1. Menu *Materials* 2. Menu *Account* 3. *Form edit account* 4. Tombol *save*. |

Gambar 4. 15 Antarmuka Halaman *Edit Account*

# DAFTAR PUSTAKA

[AS-98] Atkins-Sayre, W., Hopkins, S., Mohundro, S., & Sayre, W. (1998, November). Rewards and liabilities of presentation software as an ancillary tool: Prison or paradise? Paper presented at the National Communication Association Eighty Fourth Annual Convention, New York.

[AN-09] Nugroho, Adi. 2009. "Rekayasa Perangkat Lunak Menggunakan UML dan JAVA". Andi., Yogyakarta.

[LK-12] Lorenz Klopfenstein et al.2012. “Introducing openBOXware for Android: The Convergence between Mobile Devices and Set-Top Boxes”.Information Science and Technology Division of DiSBeF University of Urbino.

[KG-00] Goetz, Karin. 2000. "Perspective on team teaching". Volume 1 Nomor 4.

[AND-14] Androydz. 2014. New Android Set Top Box from ENY Technology Coming Soon. WWW [terhubung berkala]. http://androydz.com [diakses pada tanggal 20 Oktober 2014].

[WS-14] W3schools. 2014. W3Schools Online Web Tutorials. WWW [terhubung berkala]. http://www.w3schools.com [diakses pada tanggal 20 Oktober 2014].

[TP-15] http://www.tutorialspoint.com/json/json\_java\_example.htm [diakses pada tanggal 19 Maret 2015].

[PHP-15] http://php.net/manual/en/intro-whatcando.php [diakses pada tanggal 19 Maret 2015].

[VJS-15] http://viewerjs.org [diakses pada tanggal 07 April 2015].