

BAB I

PENDAHULUAN

1.1. Latar Belakang

Lembaga pendidikan tidak akan lepas dari kegiatan pengolahan data, baik secara manual ataupun dengan cara elektronik. Bukan sekedar kegiatan tulis menulis, menyimpan berkas. Pengolahan data merupakan suatu hal rutinitas organisasi sehingga perlu dibudayakan dengan baik dan benar supaya menghasilkan suatu informasi yang akurat. Informasi yang cepat dan akurat dalam pengambilan keputusan yang tepat perlu dikembangkan sistem pengolahan data dengan baik. Karena itu harus diperlukan alat yang dapat melakukan tugas-tugas tersebut secara cepat dan akurat yaitu komputer yang mempunyai keunggulan dalam pemrosesan data baik yang harus diolah, disimpan, disajikan, dianalisa ketepatannya dan kecepatan hasil yang dicapai.

Pendidikan memegang peranan yang sangat penting dalam meningkatkan kecerdasan bangsa dan merupakan faktor kunci dalam perkembangan pembangunan bangsa di segala bidang sesuai dengan perkembangan ilmu pengetahuan dan teknologi yang semakin pesat sekarang ini.

Laboratorium komputer merupakan tempat eksperimen, pelatihan, ataupun riset yang berhubungan dengan ilmu komputer dan memiliki beberapa komputer dalam satu jaringan yang digunakan oleh kalangan tertentu. Fakultas Ilmu Komputer Universitas Esa Unggul sebagai salah satu institusi pendidikan yang berkecimpung di dunia *ICT (Information and Communication Technologies)* sudah mempunyai Fasilitas Lab dengan sarana yang cukup baik untuk dikembangkan. Bahkan sumber

daya manusianya pun sudah siap untuk membantu mengembangkan aktivitas Lab.

Asisten Lab (Aslab) mempunyai wewenang dan tanggung jawab terhadap fasilitas lab juga memberikan kontribusi untuk aktivitas praktikum di Lab. Kegiatan praktikum ini berlangsung bersamaan ketika kegiatan perkuliahan dilakukan, sehingga jadwal perkuliahan dan jadwal praktikum dapat saling berbenturan. Hal tersebut dapat menyebabkan ketidakhadiran Asisten Lab yang mengambil mata kuliah praktikum tersebut dan menyebabkan mahasiswa tersebut tidak mendapatkan materi praktikum. Selain itu, pemilihan jadwal yang dilakukan asisten lab belum dimaksimalkan dengan baik dikarenakan proses pengolahan data tersebut harus menunggu data jadwal praktikum yang sudah disusun oleh fakultas. Dan penjadwalan tersebut membuat Asisten Lab seringkali tidak siap memandu kegiatan praktikum karena tidak ada pemilihan opsi antara kegiatan praktikum dan mata kuliah yang dilakukan. Sehingga asisten lab harus melakukan pertemuan untuk memilih jadwal praktikum dengan menggunakan *excel* yang disediakan oleh admin fakultas. Hal ini tidak efektif dalam dokumentasi kegiatan asisten, dan tidak efisien dalam penjadwalan kegiatan praktikum.

Masalah penjadwalan mata kuliah dalam dunia pendidikan sering kali terjadi. Dalam penelitian ini mencoba untuk menyelesaikan permasalahan pembuatan jadwal perkuliahan. Penyelesaian masalah penjadwalan perkuliahan dalam jumlah yang sangat besar hingga saat ini masih menjadi permasalahan yang rumit untuk diselesaikan secara manual.

Penjadwalan pada umumnya diperlukan untuk mengantisipasi adanya benturan jam kuliah dan juga waktu dosen dalam mengajar. Jadwal yang dihasilkan juga harus memenuhi batasan dan syarat yang

bertujuan agar jadwal yang dihasilkan sesuai saat digunakan. Masalah penjadwalan merupakan persoalan yang ditemukan pada situasi nyata. Masalah ini membutuhkan waktu komputasi yang cukup tinggi untuk pencarian solusinya, terlebih lagi jika ukuran permasalahan semakin besar dengan bertambahnya jumlah komponen dan tetapan.

Sistem Informasi Akademik di Universitas Esa Unggul adalah Sistem Informasi yang diperuntukan untuk dosen dalam hal mengajar dikelas yang digunakan untuk mengetahui kehadiran presensi dan jurnal mengajar, baik itu praktikum ataupun kuliah di kelas. Oleh karena itu diperlukan pemanfaatan sistem informasi penjadwalan untuk memudahkan Asisten Lab dalam melakukan koordinasi jadwal mengajar agar disesuaikan dengan matakuliah yang diambilnya, dengan membagi jadwal berdasarkan bidang rumpun ilmu yang dimiliki oleh masing masing Asisten Lab, agar proses *knowledge sharing* materi dapat terealisasi di kelas. Serta membatasi jadwal yang dimiliki, agar penjadwalan Asisten Lab dapat ter-realisis dengan baik dan merata oleh semua asisten lab.

Untuk itu perlunya perancangan dan pembangunan sebuah *Mobile Application* sebagai penetrasi terhadap penjadwalan asisten lab tersebut. *Mobile application* ini dipergunakan dalam aktivitas praktikum yang dapat mengatur jadwal praktikum berbasis notifikasi pengingat dengan dokumentasi *timesheet* kegiatan asisten lab. Dengan memanfaatkan Notifikasi pintar berbasis *mobile device smartphone*. Fitur notifikasi pintar yang memungkinkan asisten lab untuk mendapatkan peringatan saat ada pesan dan notifikasi masuk dari aplikasi pada perangkat *smartphone* yang dimiliki. sehingga akan menerima notifikasi yang sama dengan yang muncul pada layar perangkat *smartphone* yang dimiliki.

Notifikasi pada *mobile device* merupakan salah satu hal penting bagi pengguna *smartphone* karena berkaitan dengan notifikasi pada *mobile device*, notifikasi bisa datang dari aplikasi atau sistem dari *mobile device*. Sehingga pentingnya notifikasi *mobile device* bagi asisten lab menjadi hal penting untuk mempermudah dalam mengingat pemberitahuan jadwal praktikum dengan menampilkan notifikasi berupa catatan kehadiran pada saat mengajar di Lab pada saat praktikum. Kemudian Asisten Lab dapat mengontrol aktivitas praktikumnya dengan mengupdate informasi tentang praktikum.

Dengan adanya sistem informasi penjadwalan dengan notifikasi berbasis *mobile device* ini, kiranya dapat mempermudah dosen dalam berinteraksi dengan asisten lab sehingga terjadi interaksi yang signifikan antara keduanya. Sistem informasi ini juga diharapkan mampu meningkatkan perubahan perilaku asisten lab dan dosen terhadap aspek kedisiplinan dalam kehadiran praktikum dan dokumentasi *timesheet* asisten lab.

Oleh karena itu, berdasarkan uraian di atas maka akan dilakukan penelitian dengan judul ***“Rancang Bangun Sistem Informasi Penjadwalan Asisten Lab (Asisten lab) Fakultas Ilmu Komputer Dengan Notifikasi Berbasis Mobile Device”***

1.2. Identifikasi Masalah

Berdasarkan Masalah di atas, maka dapat diidentifikasi masalah sebagai berikut :

- 1) Bagaimana sistem informasi penjadwalan asisten lab fakultas ilmu komputer, yang dapat berguna berguna dalam koordinasi jadwal dan pembatasan jadwal praktikum ?

- 2) Bagaimana mengatur penjadwalan asisten lab berdasarkan bidang rumpun ilmu / keahlian yang dimiliki, agar penguasaan materi praktikum dapat ter-realisasikan di kelas ?
- 3) Bagaimana membangun sistem informasi penjadwalan yang berguna dalam *knowledge sharing* antar Asisten Lab?
- 4) Bagaimana manfaat notifikasi dalam proses penjadwalan asisten lab ?
- 5) Bagaimana manfaat sistem informasi penjadwalan untuk asisten lab fakultas ilmu komputer ?

1.3. Tujuan

Tujuan yang ingin dicapai dalam penelitian serta penulisan adalah sebagai berikut:

- 1) Merancang dan membangun sistem informasi penjadwalan asisten lab yang berguna dalam memudahkan memilih jadwal praktikum, serta membatasi jumlah matakuliah praktikum yang diambil oleh masing-masing asisten lab.
- 2) Merancang dan membangun sistem informasi penjadwalan asisten lab dengan menampilkan notifikasi pengingat jadwal praktikum, agar asisten lab dapat hadir tepat waktu dalam melakukan aktifitas mengajar praktikum dilab.
- 3) Membuat sistem informasi penjadwalan dengan mengatur bidang rumpun ilmu / keahlian yang dimiliki asisten lab, agar matakuliah tersebut tampil berdasarkan bidang rumpun ilmu yang dimiliki asisten lab.
- 4) Membangun sistem informasi penjadwalan asisten lab dengan membuat fitur diskusi seperti *chatting* antar asisten lab, agar *sharing* pengetahuan dapat merata.

1.4. Batasan Masalah

Berikut adalah batasan-batasan masalah pada penulisan ini :

- 1) Menganalisis Proses bisnis sistem yang berjalan saat ini.
- 2) Merancang sistem informasi penjadwalan asisten lab dengan menggunakan *Unified Modelling Language* (UML).
- 3) Membangun sistem informasi dengan menggunakan bahasa pemrograman PHP, Java, Ajax, JSON dan *Database* MySQL
- 4) Sistem informasi penjadwalan asisten lab ini tidak membahas materi pembelajaran praktikum dan nilai praktikum.
- 5) Pemberian notifikasi hanya berlaku ketika asisten lab sudah menentukan atau memilih jadwal praktikum yang diinginkannya.
- 6) Proses interaksi dosen dengan asisten lab dilakukan pada menu chat yang tersedia.
- 7) Asumsi koordinator asisten di berikan kewenangan untuk menyusun jadwal praktikum.
- 8) Pembatasan penjadwalan praktikum untuk masing masing asisten lab dibatasi hanya 6 matakuliah dan tidak boleh lebih.
- 9) Proses pembuatan *timesheet* hanya berfokus pada dokumentasi kegiatan asisten lab pada saat praktikum, tidak membahas mengenai pembayaran honor asisten lab melalui *timesheet*.

1.5. Manfaat Penelitian

Manfaat yang didapatkan dari penelitian ini adalah:

- 1) Sistem informasi penjadwalan asisten lab dengan (*Mobile Device*) yang tersedia menjadi berdaya guna untuk keperluan praktikum.
- 2) Mempermudah layanan informasi dan dokumentasi kegiatan asisten lab.

- 3) Mempercepat akses penyampaian informasi mengenai praktikum secara *mobile* dengan menampilkan notifikasi.

1.6. Metodologi Penelitian

Tahapan yang akan dilakukan dalam penelitian ini terdiri dari langkah-langkah berikut :

1) Studi Literatur

Studi literatur ini merupakan teori atau tinjauan pustaka yang dibutuhkan untuk penulisan proposal tugas akhir dengan cara membaca dan memahami berbagai literatur, buku, jurnal maupun bahan kepustakaan yang berhubungan dengan masalah yang sedang diteliti.

2) Studi lapangan

Studi ini merupakan pengumpulan data di lapangan dengan cara melakukan pengamatan terhadap objek penelitian di fakultas ilmu komputer universitas esa unggul.

3) Metode Pengembangann Perangkat Lunak RAD (*Rapid Application Development*)

Metodologi yang digunakan dalam penelitian ini yaitu metode RAD (*Rapid Application Development*). terdapat tahapan – tahapan sebagai berikut :

- 1. Fase Penentuan Syarat – Syarat**
- 2. Fase Perancangan**
- 3. Fase Pengkodean (*Coding Phase*)**
- 4. Fase Pelaksanaan**

BAB II

LANDASAN TEORI

2.1. Rancang Bangun

2.1.1. Perancangan

Menurut Kenneth Laudon dan Jane Laudon (2012), Perancangan Sistem merupakan keseluruhan rencana atau model untuk sistem yang terdiri dari semua spesifikasi sistem yang memberikan bentuk dan struktur.

Menurut Rosa A.S dan Shalahuddin (2016), Desain atau perancangan sistem dalam pembangunan perangkat lunak merupakan upaya untuk mengkonstruksi sebuah sistem yang memberikan (kepuasan informal) akan spesifikasi kebutuhan fungsional, memenuhi target, memenuhi kebutuhan secara implisit atau eksplisit dari segi performansi maupun penggunaan sumber daya, kepuasan batasan pada proses desain segi biaya, waktu, dan perangkat lunak.

2.1.1.1. Analisis Kebutuhan

Menurut Rosa A.S dan Shalahuddin (2016), *Requirement Analysis* atau analisis kebutuhan adalah proses perencanaan kebutuhan yang dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami kebutuhan perangkat lunak seperti apa yang dibutuhkan *user*.

Kebutuhan yang telah dikumpulkan menggunakan wawancara, observasi, kuisioner, atau gabungan dari ketiga hal tersebut dapat dikelompokkan menjadi beberapa kategori sebagai berikut (tidak semua kebutuhan ini harus ada) :

1) Kebutuhan Fungsional (*Functional Requirement*)

Kebutuhan yang terkait dengan fungsi produk, misalnya sistem informasi harus mampu mencetak laporan, sistem informasi harus mampu menampilkan grafik, dan lain lain.

2) Kebutuhan Lingkungan Sistem (*Deployment Requirement*)

Kebutuhan terkait dengan lingkungan dimana sistem informasi akan digunakan baik perangkat lunak maupun perangkat keras.

2.1.1.2. Desain Berorientasi Objek

Menurut Rosa A.S dan Shalahuddin (2016), Desain berorientasi objek atau *Object Oriented Design* (OOD) adalah tahapan perantara untuk memetakan spesifikasi atau kebutuhan sistem yang akan dibangun dengan konsep berorientasi objek ke desain pemodelan agar lebih mudah diimplementasikan.

Pemodelan berorientasi objek biasanya akan dituangkan dalam dokumentasi perangkat lunak dengan menggunakan perangkat pemodelan berorientasi objek salah satunya UML (*Unified Modelling Language*)

2.1.2. Pembangunan

Menurut Kenneth Laudon dan Jane Laudon (2012), Pembangunan atau pengembangan sistem adalah aktivitas untuk menghasilkan sebuah sistem informasi sebagai solusi untuk sebuah organisasi. Pembangunan sistem merupakan aktivitas terstruktur untuk menyelesaikan berbagai jenis masalah dengan kegiatan yang

berbeda yaitu analisis sistem, desain sistem, pemograman, pengujian, *conversion*, produksi dan pemeliharaan.

Menurut Rosa A.S dan Shalahuddin (2016), Pembangunan atau pengembangan sistem atau yang sering disebut SDLC (*System Devlopment Life Cycle*) adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model untuk mengembangkan sistem-sistem perangkat lunak sebelumnya (berdasarkan *best practice* atau cara-cara yang sudah teruji dengan baik).

2.2. Sistem

Menurut Kenneth Laudon dan Jane Laudon (2012), Sistem adalah komponen yang saling berhubungan dan bekerja sama untuk mengumpulkan, memproses, menyimpan, dan menyebarkan informasi untuk mendukung pengambilan keputusan, kordinasi, analisis, dan visualisasi pada organisasi.

Menurut Shelly dan Rosenblatt (2012), mendefinisikan sistem adalah suatu komponen yang saling berkaitan guna menghasilkan hasil yang spesifik.

2.3. Informasi

Menurut Kelly Rainer dan Casey Cegielski (2011), Informasi adalah data yang telah dikelola sehingga memiliki arti dan nilai bagi penerimanya.

Menurut Shelly dan Rosenblatt (2012), Informasi adalah data yang telah diubah menjadi suatu output yang memiliki nilai .

2.4. Sistem Informasi

Menurut Rochaety *et al* (2013) Sistem informasi merupakan kumpulan komponen dalam sebuah perusahaan yang berhubungan dengan proses penciptaan dan pengaliran informasi. Keandalan suatu sistem informasi dalam perusahaan terletak pada keterkaitan antara komponen yang ada sehingga dapat menghasilkan aliran informasi yang berguna, akurat, terpercaya, detail, cepat, relevan bagi kepentingan operasi perusahaan.

2.5. Penjadwalan

Menurut Scroeder (2000), Penjadwalan dapat didefinisikan sebagai suatu petunjuk atau indikasi apa saja yang harus dilakukan, dengan siapa, dan dengan peralatan apa yang digunakan untuk menyelesaikan suatu pekerjaan pada waktu tertentu.

Menurut Pinedo (2002), Penjadwalan selalu berhubungan dengan pengalokasian sumber daya yang ada pada jangka waktu tertentu, hal tersebut adalah proses pengambilan keputusan yang tujuannya adalah untuk optimalitas.

2.5.1. Tujuan Penjadwalan

Penjadwalan memiliki tujuan sebagai berikut :

- 1) Efeksiensi pemakaian sumber daya dan meminimasi makespan
- 2) Agar lebih responsif terhadap permintaan. Meminimasi rata rata *flowtime* atau rata rata waktu menunggu (*tardiness*)
- 3) Meningkatkan produktivitas mesin dengan jalan meminimai waktu menganggur.

- 4) Penjadwalan menambah kapasitas dan fleksibilitas yang terkait memberikan waktu pengiriman yang lebih cepat dan dengan demikian pelayanan akan lebih baik.
- 5) Memenuhi batas waktu (*due date*), serta mengurangi lateness atau keterlambatan.

2.6. Aktifitas Praktikum

Menurut Anton Mulyono (2010), Aktivitas artinya “kegiatan atau keaktifan”. Jadi segala sesuatu yang dilakukan atau kegiatan-kegiatan yang terjadi baik fisik maupun nonfisik, merupakan suatu aktivitas.

Kegiatan kerja di laboratorium (*laboratory work*) lazimnya dilakukan untuk melatih mahasiswa dalam penguasaan kompetensi (*skill*) dalam bidang studi tertentu. Dalam konteks *laboratory work*, kompetensi lebih ditekankan pada apa yang dapat dilakukan oleh mahasiswa dan bukan hanya sekedar mengetahui.

Dalam aspek kognitif, ditekankan agar teori dapat diaplikasikan pada kondisi atau problema konkret, sehingga kegiatan yang dilakukan ditekankan pada aspek latihan. Hal ini sejalan dengan teori belajar yang banyak diterapkan dalam proses belajar-mengajar di perguruan tinggi melalui pendekatan terapan (*Applied Approach*) yang merekomendasikan bahwa setelah mahasiswa diberikan orientasi (isi pengetahuan dan metodenya). Kemudian harus diikuti dengan latihan dan umpan balik.

2.7. Asisten Lab

Asisten Lab adalah sumber daya manusia yang ada di fakultas ilmu komputer yang bertugas sebagai asisten dosen pada saat praktikum dan mengontrol aktivitas praktikum serta bertanggung jawab terhadap kondisi lab.

2.8. Aplikasi Mobile

Menurut Rangsang Purnama (2010), Kata *mobile* mempunyai arti bergerak atau berpindah, sehingga aplikasi *mobile* adalah sebutan untuk aplikasi yang berjalan di *mobile device*. Dengan menggunakan aplikasi *mobile*, dapat dengan mudah melakukan berbagai macam aktifitas mulai dari hiburan, berjualan, belajar, mengerjakan pekerjaan kantor, *browsing* dan lain sebagainya.

2.9. Notifikasi

Menurut Kamus Besar Bahasa Indonesia, Notifikasi berarti pemberitahuan atau kabar. Notifikasi dalam tugas akhir ini dijabarkan sebagai pemberitahuan yang dikirimkan oleh aplikasi *mobile*.

2.10. Android

Menurut Suhas Holla dan Katti Mahima (2012), *Android* merupakan sistem operasi untuk perangkat *mobile* baru yang berjalan pada kernel *Linux*. Sistem operasi ini menyediakan lingkungan pengembangan aplikasi yang fleksibel bagi pengembang aplikasi *mobile*. Aplikasi *Android* dibangun dengan menggunakan Bahasa pemrograman Java, Namun aplikasi tersebut tidak dapat dijalankan dengan menggunakan *Java Virtual Machine* (JVM) yang standar, melainkan menggunakan *Dalvik Virtual Machine*. Adapun *Dalvik Virtual Machine* adalah mesin virtual buatan Google yang digunakan untuk mengkonversi dan mengeksekusi kode *Java*. Dalvik dibuat dengan tujuan untuk mendukung sifat sistem operasi *mobile* yang ringan, yang dibutuhkan oleh perangkat *mobile* yang notabene memiliki kemampuan yang terbatas jika dibandingkan dengan desktop atau laptop.

Menurut Abdur Kadir (2013), Dalam mengembangkan aplikasi Android, pengembang membutuhkan piranti yang dapat memudahkan pembuatan aplikasi. Piranti yang diperlukan untuk pengembangan aplikasi Android mencakup *Java Development Kit* (JDK), Android Studio dan Android SDK.

a) *Java Development Kit* (JDK)

Java Development Kit (JDK) adalah perangkat pengembangan aplikasi berbasis bahasa pemrograman Java. JDK digunakan untuk melakukan proses kompilasi dari kode java ke *bytecode* yang dapat dimengerti dan dijalankan oleh *Java Runtime Environment*. Dalam mengembangkan aplikasi *Android*, pengembang dapat memanfaatkan *library* yang disediakan di dalam Java kecuali *library swing*.

b) *Android Studio*

Android Studio adalah sebuah media untuk mengembangkan aplikasi (*IDE/Integrated Development Environment*) berbasis *Android* yang resmi. *Android Studio* menyediakan *tools* dengan kecepatan yang tinggi sehingga memungkinkan pengembang untuk membuat suatu aplikasi *Android* dengan baik. IDE ini juga menyediakan sistem *build/deploy* yang fleksibel bagi penggunaanya.

c) *Android SDK*

Android Software Development Kit (SDK) adalah kumpulan perangkat lunak yang berisi *library*, *debugger* (perangkat untuk mencari kesalahan pada program), *emulator* (perangkat yang berfungsi sebagai peniru perangkat *mobile* yang sebenarnya), dokumentasi, kode contoh dan panduan. Keberadaan *emulator* membuat pengembang dapat membuat dan menguji aplikasi

Android tanpa harus memasang aplikasi terlebih dahulu pada perangkat *Android*.

2.10.1. Arsitektur Android

Menurut Safaat Nazaruddin (2015), Secara garis besar arsitektur *Android* dapat dijelaskan dan digambarkan sebagai berikut:

a) *Applications dan Widgets*

Adalah layer dimana kita berhubungan dengan aplikasi saja, dimana biasanya kita download aplikasi kemudian kita lakukan instalasi dan jalankan aplikasi tersebut. Di layer terdapat aplikasi inti termasuk klien email, program SMS, kalender, peta, browser, kontak, dan lain-lain. Semua aplikasi ditulis menggunakan bahasa pemrograman Java.

b) *Applications Framework*

Android adalah “Open Development Platform”, yaitu *Android* menawarkan kepada pengembang atau member kemampuan kepada pengembang untuk membangun aplikasi yang bagus dan inovatif. Pengembangan bebas untuk mengakses perangkat keras, akses informasi resources, menjalankan service background, mengatur alarm dan menambah status notifications, dan sebagainya.

c) *Libraries*

Layer dimana fitur-fitur *Android* berbeda, biasanya para pembuat aplikasi mengakses libraries untuk menjalankan aplikasinya. Berjalan diatas kernel, Layer ini meliputi berbagai library C/C++ inti seperti Libc dan SSL.

d) *Android Runtime*

Berisi *Core libraries* dan *Dalvik Virtual Machine (DVM)*. *Core libraries* mencakup serangkaian *library* Java, sedangkan *Dalvik Virtual Machine (DVM)* merupakan *virtual machine* yang dijalankan didalam sistem operasi android. Dalvik di konversi dari *Java Virtual Machine (JVM)* yang mana *.class files* di ubah menjadi *.dex*.

e) *Linux kernel*

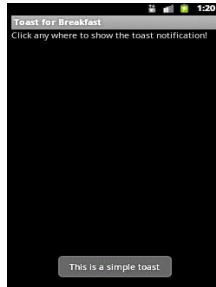
Adalah layer dimana inti dari operating sistem dari Android itu berada. Berisi file-file system yang mengatur sistem processing, memory, resource, drivers, dan system-sistem operasi android lainnya. Linuk kernel yang digunakan adalah linux kernelrelase 2.

2.10.2. Notifikasi Android

Menurut www.developer.android.com yang Diakses pada 5 Oktober 2017, pada Android, sistem notifikasi memungkinkan pengguna untuk mendapatkan informasi tentang peristiwa yang relevan dan tepat waktu dalam aplikasi, seperti pesan obrolan baru dari seorang teman atau acara kalender. Notifikasi sebagai saluran berita yang memberitahu pengguna untuk peristiwa penting yang terjadi sementara pengguna tidak memperhatikan perangkat Androidnya. Tiga jenis utama tampilan notifikasi pada Android :

1) *Toast Notification*

Pesan sementara yang muncul pada jendela tampilan sekarang yang secara otomatis menghilang dari layar di mana tidak ada interaksi atau respon dengan pengguna.

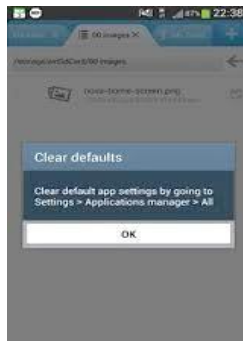


Gambar 2.10.2.1 Toast Notifikasi

Sumber : www.developer.android.com

2) *Dialog Notification*

Sebuah jendela kecil yang muncul (popping up) secara otomatis dan tiba-tiba di depan jendela aplikasi yang sedang aktif serta meminta pengguna untuk membuat keputusan atau memasukkan informasi tambahan. Sebuah dialog tidak memenuhi layar dan biasanya digunakan untuk event yang mengharuskan pengguna untuk mengambil tindakan sebelum action dapat dilanjutkan.



Gambar 2.10.2.2 Dialog Notification

Sumber : www.developer.android.com

3) *Notification Bar Navigation*

Pemberitahuan navigasi bar menunjukkan beberapa latar belakang layanan yang sedang berlangsung seperti *download* atau bermain musik dan menampilkan peringatan atau informasi. Untuk melihat notifikasi, pengguna perlu membuka *drawer notification*.



Gambar 2.10.2.3 *Navigation Bar Notification*

Sumber : www.developer.android.com

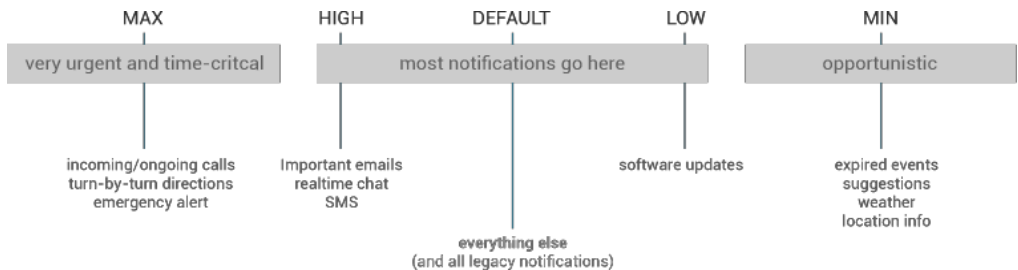
4) *Prioritas Notifikasi Android*

Android mendukung tingkat prioritas untuk pemberitahuan. Ini memungkinkan pengguna mengubah notifikasi mana yang akan muncul, bersifat relatif terhadap pemberitahuan lainnya dan membantu memastikan bahwa pengguna selalu melihat notifikasi tersebut yang paling penting pertama kalinya. Tingkat prioritas pada notifikasi :

Tabel 2.10.2 Prioritas Notifikasi Android

Prioritas	Penggunaan
<i>MAX</i>	Gunakan untuk pemberitahuan kritis dan mendesak yang mengingatkan pengguna untuk kondisi yang <i>critical</i> atau perlu diselesaikan sebelum mereka dapat melanjutkan tugas tertentu.
<i>HIGH</i>	Gunakan terutama untuk komunikasi penting, seperti pesan atau chatting dengan konten yang sangat menarik bagi pengguna. Pemberitahuan prioritas tinggi memicu tampilan notifikasi <i>pop-up</i> .
<i>DEFAULT</i>	Gunakan untuk semua pemberitahuan yang tidak termasuk ke dalam salah satu prioritas lain yang dijelaskan di sini.
<i>LOW</i>	Gunakan untuk pemberitahuan yang mana pengguna inginkan agar diinformasikan, tapi yang kurang mendesak. Pemberitahuan prioritas rendah cenderung muncul di bagian bawah daftar, seperti <i>social updates</i> : Pengguna telah meminta untuk diberitahu tentang konten tersebut, tapi pemberitahuan ini tidak lebih diutamakan dari prioritas yang mendesak atau komunikasi langsung.
<i>MIN</i>	Gunakan untuk informasi kontekstual atau konten latar belakang seperti informasi cuaca atau informasi lokasi. Pemberitahuan prioritas ini tidak muncul dalam status bar. Pengguna menemukan pemberitahuan ini sekedar untuk mengetahuinya saja.

Sumber : www.developer.android.com



Gambar 2.10.2.4 Prioritas Notifikasi Android

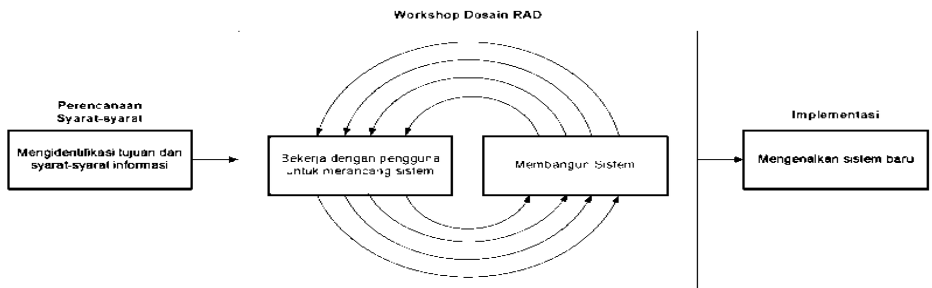
Sumber : www.developer.android.com

2.11. *Rapid Application Development (RAD)*

Menurut Roger S Pressman. (2010), *Rapid Application Development (RAD)* adalah sebuah model proses perkembangan perangkat lunak sekuensial linear yang menekankan pada siklus perkembangan cepat dengan menggunakan pendekatan konstruksi berbasis komponen. Sehingga apabila kebutuhan dipahami dengan baik, maka sistem fungsional yang utuh dapat diselesaikan dalam waktu kira-kira 60-90 hari.

Menurut Bentley Lonnie and Whitten Jefrey (2007), *Rapid Application Development (RAD)* adalah sebuah strategi pengembangan sistem yang menekankan kecepatan pengembangan melalui keterlibatan pengguna yang ekstensif dalam konstruksi, cepat, berulang dan bertambah serangkain prototype / prototipe bekerja sebuah sistem yang pada akhirnya berkembang kedalam sistem final (atau sebuah versi).

Menurut Kenneth E. Kendall dan Jullie E Kendall (2010), Metode *RAD* mempunyai 4 tahapan utama, yaitu :



Gambar 2.11. Metode RAD

Sumber : Kenneth E. Kendall dan Jullie E Kendall (2010)

Model pengembangan RAD memiliki yaitu fase perencanaan syarat-syarat, fase perancangan, fase konstruksi, dan fase pelaksanaan. Berikut adalah penjelasan masing-masing fase dalam penelitian ini :

1. Fase Perencanaan Syarat-Syarat

Pada tahap ini dilakukan pengidentifikasian tujuan aplikasi atau sistem serta untuk mengidentifikasi syarat-syarat informasi yang ditimbulkan dari tujuan-tujuan tersebut. Pada tahap ini peneliti melakukan analisis kebutuhan.

2. Fase Perancangan

Pada tahap ini adalah melakukan proses desain dan melakukan perbaikan-perbaikan apabila masih terdapat ketidaksesuaian desain antara *user* dan *analyst*. Untuk tahap ini keaktifan user yang terlibat sangat menentukan untuk mencapai tujuan, karena user bisa langsung memberikan komentar apabila terdapat ketidaksesuaian pada desain.

3. Fase Konstruksi

Setelah desain dari sistem yang akan dibuat sudah disetujui

baik itu oleh *user* dan *analyst*, maka pada tahap ini *programmer* mengembangkan desain menjadi suatu program. Hal terpenting adalah keterlibatan *user* sangat diperlukan supaya sistem yang dikembangkan dapat sesuai dengan permintaan.

4. Fase Pelaksanaan

Pada tahap ini dilakukan pengujian aplikasi, dengan menggunakan pengujian *black box testing*. Yaitu sebuah pengujian dengan menjalankan setiap fungsi dari aplikasi.

2.12. Pemodelan Perangkat Lunak

Menurut Rosa A.S dan M. Shalahuddin (2016) Pemodelan perangkat lunak adalah gambaran dari realita yang simple dan dituangkan dalam bentuk pemetaan dengan aturan tertentu. Pemodelan perangkat lunak digunakan untuk mempermudah langkah berikutnya dari pengembangan sebuah sistem informasi sehingga lebih terencana dan didigunakan untuk memvisualisasi perangkat lunak yang akan dibuat. Salah satu perangkat pemodelan yaitu *Unified Modelling Language*.

2.12.1. Unified Modelling Language (UML)

Menurut Rosa A.S dan M. Shalahuddin (2016), *Unified Modeling Language* (UML) adalah salah satu standar bahasa yang banyak digunakan di dunia industry untuk mengidentifikasi requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.


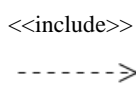
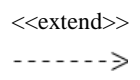

Dari definisi di atas, dapat di simpulkan bahwa UML adalah sebuah bahasa yang digunakan untuk menggambarkan sistem informasi yang berkaitan dengan objek yang digunakan untuk menjelaskan suatu informasi sesuai dengan kebutuhan pengguna.



2.12.1.1. Use Case Diagram

Menurut Shelly dan Rosenblatt (2012), *Use case* merupakan suatu diagram yang melukiskan interaksi antara sistem dengan para pemakai. Dengan kata lain, *use case* diagram dengan nyata menguraikan siapa yang akan menggunakan sistem dan dengan cara apa pemakai dapat saling berhubungan dengan sistem.

Dari definisi diatas dapat disimpulkan *use case* diagram adalah diagram yang menggambarkan bagaimana kita berinteraksi atau berhubungan dengan sistem. Secara umum *use case* diagram terdiri dari:

Tabel 2.12.1. Tabel Simbol *Use Case* Diagram

No.	Simbol	Nama	Keterangan
1.		<i>Actor</i>	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat
2.		<i>Include</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> , dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.
3.		<i>Extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> , dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri.
4.		<i>Association</i>	Komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.

No.	Simbol	Nama	Keterangan
5.		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor

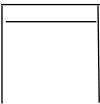




Sumber : Rosa A.S dan Shalahuddin (2016)

2.12.1.2. *Activity Diagram*

Shelly dan Rosenblatt (2012) mengungkapkan, *Activity Diagram* adalah teknik untuk mendeksripsikan logika prosedural proses bisnis dan aliran kerja dalam banyak kasus. *Activity Diagram* mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah *Activity Diagram* bisa mendukung perilaku paralel sedangkan *flowchart* tidak bisa. *Activity Diagram* menunjukkan urutan dimana tindakan berlangsung dan mengidentifikasi hasil

Dari definisi diatas dapat ditarik kesimpulan bahwa *Activity Diagram* adalah diagram yang dapat menggambarkan bagaimana aktivitas suatu aktor atau pengguna dari kegiatan yang dilakukan didalam sistem, dari mulai hingga terselesaikannya proses tersebut. *Activity Diagram* juga dapat melibatkan satu aktor atau lebih, Simbol-simbol yang digunakan pada saat pembuatan *Activity Diagram* adalah sebagai berikut:

Tabel 2.12.1.2 Simbol pada *Activity Diagram*

No.	Simbol	Nama	Keterangan
1.		<i>Swimlane</i>	Merupakan sebuah cara untuk mengelompokkan <i>activity</i> berdasarkan <i>actor</i> (mengelompokkan <i>activity</i> dari sebuah urutan yang sama)
2.		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
3.		<i>Decision</i> <i>/percabangan</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
4.		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
5.		<i>Activity Final</i> <i>Node</i>	Bagaimana objek dibentuk dan dihancurkan

Sumber : Rosa A.S dan Shalahuddin (2016)



2.12.1.3. Class Diagram

Menurut Rosa A.S dan Shalahuddin (2016), *Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Dari definisi diatas dapat ditarik kesimpulan bahwa *Class Diagram* adalah suatu yang diagram yang menggambarkan

perancangan kelas yang nantinya akan dibuat oleh *programmer* untuk membuat *database* perangkat lunak. Berikut adalah simbol-simbol yang ada pada *Class Diagram* :

Tabel 2.12.1.3. Simbol pada *Class Diagram*

No.	Simbol	Nama	Keterangan
1.		<i>Class</i>	Kelas pada struktur sistem.
2.		<i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>

Sumber : Rosa A.S dan Shalahuddin (2016)


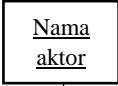

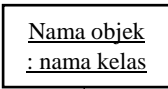

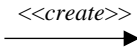
2.12.1.4. Sequence Diagram

Menurut Rosa A.S dan Shalahuddin (2016), *Sequence Diagram* atau Diagram Sekuen menggambarkan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh Karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*.

Dari definisi diatas dapat ditarik kesimpulan diagram sekuen atau *sequence diagram* digunakan untuk menggambarkan apa yang dilakukan oleh objek pada *use case*, dideskripsikan waktu dan apa yang terjadi pada keterkaitan

antar objek. Berikut ini merupakan simbol-simbol yang ada pada diagram sekuen :

Tabel 2.12.1.4. Simbol pada *Sequence Diagram*

No.	Simbol	Nama	Keterangan
1.	 atau 	Aktor	Orang yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri. Biasanya dinyatakan menggunakan kata benda diawal frase nama aktor.
2.		Garis hidup/ <i>lifeline</i>	Menyatakan kehidupan suatu objek.
3.		Objek	Menyatakan objek yang berinteraksi pesan.
4.		Waktu aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dalam waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya.
5.		Pesan tipe <i>create</i>	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah

Sumber : Rosa A.S dan Shalahuddin (2016)

BAB III

GAMBARAN UMUM INSTANSI

3.1. Tempat dan Waktu Penelitian

Pelaksanaan penelitian ini dilakukan mulai pada awal bulan Oktober, bertempat di Labolatorium dan Fakultas Ilmu Komputer Ilmu Komputer, bertempat di Lt.4 & 5 Gedung Biru Universitas Esa Unggul Jalan Arjuna Utara No. 9.

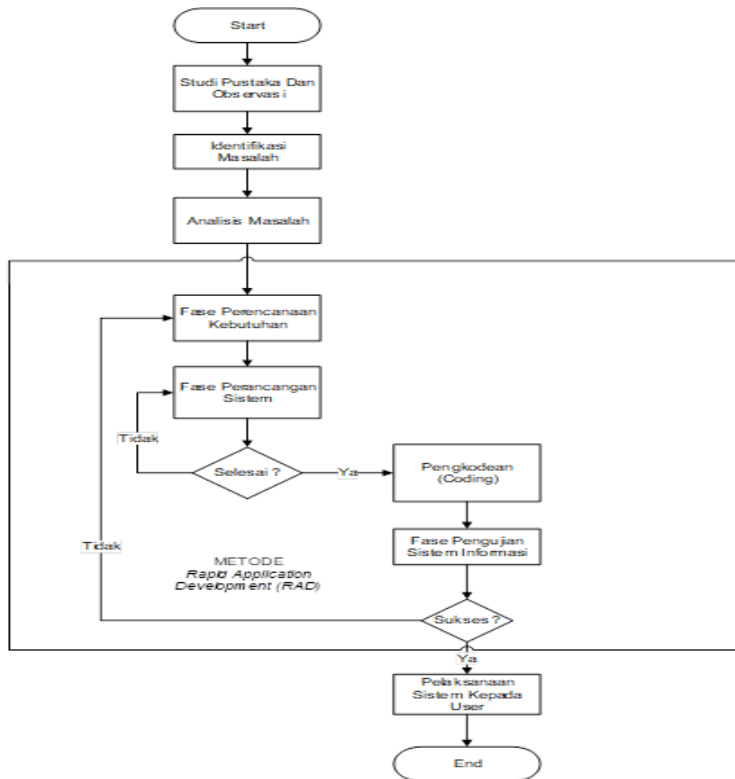
3.2. Tugas dan Tanggung Jawab Asisten Lab (ASLAB)

Berikut ini merupakan tugas dan tanggung jawab seorang asisten lab:

- Mengawasi jalannya praktikum.
- Memberikan materi dan tugas praktikum.
- Merekapitulasi hasil praktikum selama satu semester.
- Membantu dosen dalam praktikum
- Menangani mahasiswa/i yang memiliki absen bermasalah
- Membantu membuat soal ujian praktikum
- Memberikan materi kepada mahasiswa/i sesuai dengan mata kuliah yang di ajar.

3.3. Kerangka Pemikiran

Kerangka pemikiran pada gambar di bawah ini merupakan serangkaian bagan-bagan menggambarkan alur dari proses penelitian dalam pembuatan Rancang bangun sistem informasi penjadwalan *asisten lab* (asisten lab) fakultas ilmu komputer dengan notifikasi berbasis *mobile device*.. Berikut merupakan penjelasan mengenai kerangka pemikiran yang telah dibuat :



Gambar 3.3 Kerangka Pemikiran

Keterangan Kerangka Pemikiran :

1. Studi Pustaka dan Observasi

Melakukan pengumpulan studi pustaka berdasarkan jurnal, dan buku. Untuk mendukung proses pembuatan tugas akhir, dan melakukan observasi di Fakultas Ilmu Komputer Universitas Esa Unggul.

2. Identifikasi Masalah

Melakukanidentifikasi masalah perihal tentang penjadwalan praktikum di Universitas Esa Unggul Fakultas Ilmu Komputer.

3. Analisis Masalah

Melakukan analisis masalah berdasarkan fase penjadwalan yang berjalan di Fakultas Ilmu Komputer. Universitas Esa Unggul

4. Fase Perencanaan Kebutuhan

Melakukan perencanaan kebutuhan dari sistem yang akan dibuat berdasarkan analisis masalah yang telah dibuat.

5. Fase Perancangan Sistem Dengan UML

Melakukan perancangan sistem dengan UML berdasarkan perencanaan kebutuhan sistem yang dibuat.

6. Fase Pengkodean (*Coding Phase*)

Melakukan kontruksi pembangunann sistem berdasarkan kebutuhan dan desain yang telah dibuat dengan bahasa pemrograman Java, dan *Database* MySQL.

7. Fase Pengujian Sistem

Melakukan pengujian sistem terhadap sistem yang telah dibangun dengan metode *blackbox testing*.

8. Pengenalan Sistem Kepada User

Tahap terakhir yaitu mengenalkan sistem kepada user yaitu asisten lab, admin koordinator asisten lab dan dosen. Untuk meninjau kembali penggunaan aplikasi.

3.4. Proses Bisnis Berjalan

3.4.1. Proses Pembuatan Jadwal Praktikum

Admin koordinator asisten lab bertugas untuk merekap jadwal matakuliah praktikum yang telah diatur oleh fakultas, dan dibuat dalam bentuk *excel* dengan contoh di bawah ini :

Jadwal Mengajar Asisten Semester Genap 2016-2017 (Protected View) - Excel

Be careful—files from the Internet can contain viruses. Unless you need to edit, it's safer to stay in Protected View. [Enable Editing]

B4 D Teknik Informatika

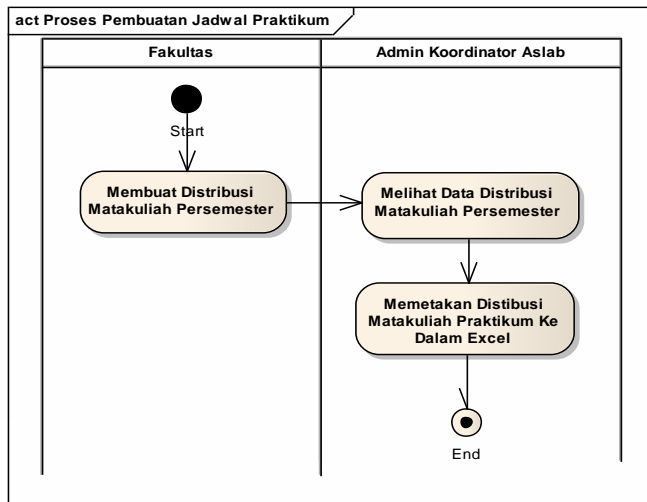
Alokasi Jadwal Mengajar Asisten Fakultas Ilmu Komputer Semester Genap Tahun Akademik 2016/2017

	Kode	Nama HK	Sesi	Kelompok	Jadwal	Ruang	Sesi	Asisten	Kode
1	1	CI1230 ALGORITMA LANJUT	1	1	Rabu, 09:00-10:00	LAB-C	2	Giang	7614
2	2	CI1230 ALGORITMA LANJUT	1	2	Rabu, 10:00-12:00	LAB-C	2	Ricko	7437
3	3	CI1230 ALGORITMA LANJUT	2	2	Rabu, 10:00-12:00	LAB-D	2	Vienne	7417
4	4	CI1230 ALGORITMA LANJUT	2	1	Rabu, 09:00-10:00	LAB-D	2	Ade	7281
5	5	CI1230 ALGORITMA LANJUT	3	1	Rabu, 09:00-10:00	LAB-E	2	Wenu	7262
6	6	CI1230 ALGORITMA LANJUT	1	1	Selasa, 13:00-14:00	LAB-C	4	Lingga	7423
7	7	CI1230 ANALISIS PERANC. SISTEM	1	2	Selasa, 10:00-10:00	LAB-C	4	Sarvel Wahyuni	7174
8	8	CI1230 ANALISIS PERANC. SISTEM	2	1	Selasa, 13:00-14:00	LAB-D	4	Sarvel Wahyuni	7174
9	9	CI1230 ANALISIS PERANC. SISTEM	2	2	Selasa, 10:00-10:00	LAB-D	4	Fajar	7180
10	10	CI1230 METODE NUMERIK	1	1	Rabu, 14:40-16:20	LAB-C	4	Fajar	7180
11	11	CI1230 METODE NUMERIK	1	2	Rabu, 10:00-10:00	LAB-C	4	Fajar	7180
12	12	CI1230 METODE NUMERIK	2	1	Rabu, 14:40-16:20	LAB-D	4	Alan Dufren	7267
13	13	CI1230 METODE NUMERIK	2	2	Rabu, 10:00-10:00	LAB-D	4		
14	14	CI1230 PEMROGRAMAN BERORIENTASI OBJEK	1	2	Kamis, 14:30-16:00	LAB-C	4	Wenu	7263
15	15	CI1230 PEMROGRAMAN BERORIENTASI OBJEK	1	1	Kamis, 14:40-16:20	LAB-C	4	Lingga	7423
16	16	CI1230 PEMROGRAMAN BERORIENTASI OBJEK	2	2	Kamis, 10:00-10:00	LAB-D	4	Taufik	7172
17	17	CI1230 PEMROGRAMAN BERORIENTASI OBJEK	2	1	Kamis, 14:40-16:20	LAB-D	4	Taufik	7172
18	18	CI1230 PEMROGRAMAN JAVA	1	1	Rabu, 07:30-09:10	LAB-D	6	Wenu	7263
19	19	CI1230 PEMROGRAMAN JAVA	1	2	Rabu, 07:30-09:10	LAB-C	6	Ade	7281
20	20	CI1230 PERANCANGAN SISTEM BERBASIS KOMPOSER	1	1	Kamis, 14:40-16:20	LAB-E	6	Fajar	7180

Ready

Gambar 3.4.1. Alokasi Jadwal Mengajar Asisten Dalam eXcel

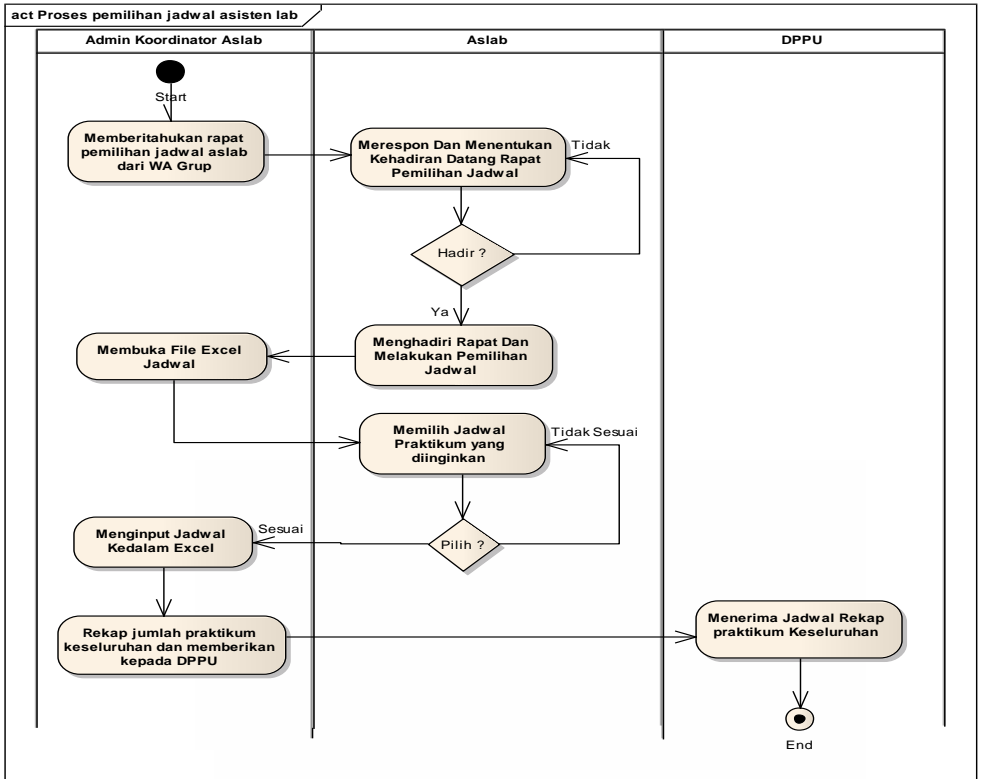
Berikut Adalah Fase proses pembuatan jadwal praktikum, tergambar dalam *activity diagram* di bawah ini :



Gambar 3.4.1.2. Proses Pembuatan Jadwal Praktikum

3.4.2. Proses Pemilihan Jadwal Praktikum Asisten lab

Adapun proses tersebut tergambar dalam *activity diagram* di bawah ini :



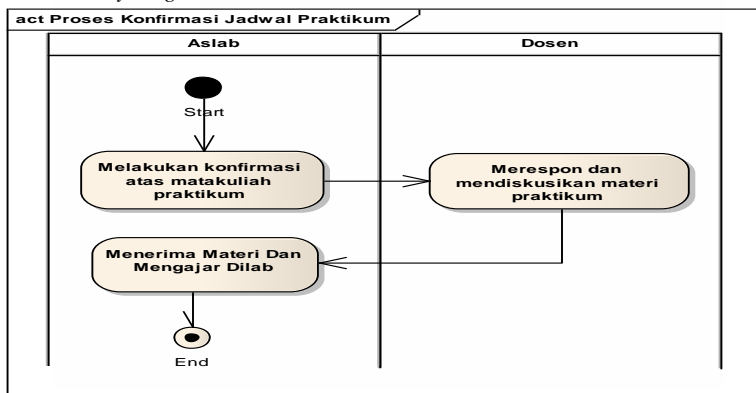
Gambar 3.4.2. Proses Pemilihan Jadwal Praktikum

Ketika admin koordinator asisten lab selesai membuat pemetaan distribusi matakuliah praktikum ke dalam *excel*, kemudian admin memberitahukan rapat pemilihan jadwal dari WA grup. Kemudian asisten lab merespon pemberitahuan dari WA grup, asisten lab menentukan jika iya datang maka asisten lab melakukan

pemilihan jadwal. Jika tidak maka asisten lab merespon WA grup. Kemudian admin membuka *file excel* jadwal pada saat rapat. Asisten lab memilih jadwal praktikum sesuai yang diinginkannya. Jika asisten lab memilih ya maka admin menginput jadwal ke dalam *excel*, jika terjadi bentrok dengan matakuliah yang diikuti asisten lab, maka asisten lab memilih jadwal lain. Setelah selesai admin membuat rekap jumlah praktikum keseluruhan dan memberikan laporan kepada DPPU. Kemudian pihak dppu menerima rekap jadwal praktikum keseluruhan.

3.4.3. Proses Konfirmasi Jadwal Praktikum Kepada Dosen Bersangkutan

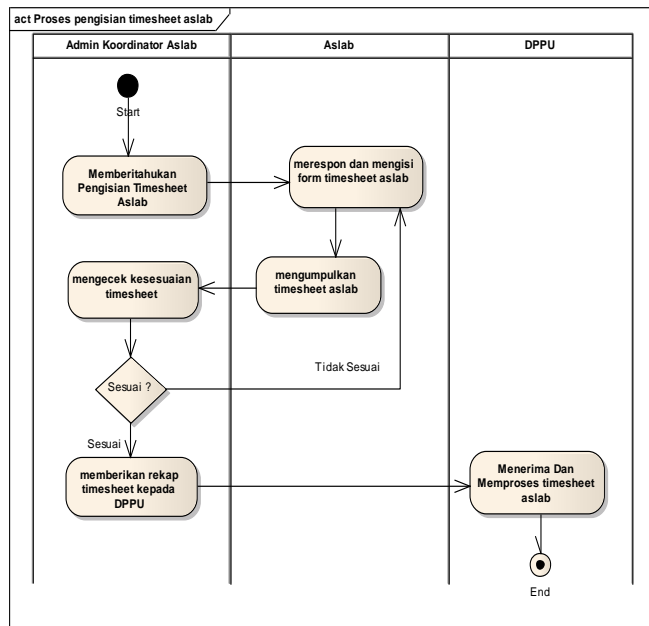
Setelah asisten lab mendapatkan jadwal praktikumnya, maka asisten lab diwajibkan melakukan konfirmasi atas matakuliah praktikum yang diampuhnya kepada dosen yang bersangkutan. Kemudian dosen merespon dan mendiskusikan materi tersebut. Setelah selesai melakukan konfirmasi, maka asisten lab menerima materi dan mengajar di lab. Adapun proses tersebut tergambar dalam *activity diagram* di bawah ini :



Gambar 3.4.3. Proses Konfirmasi Jadwal Praktikum

3.4.4. Proses Pengisian *Timesheet* Kegiatan Praktikum Asisten lab

Adapun proses tersebut tergambar dalam *activity diagram* di bawah ini :



Gambar 3.4.4. Proses Pengisian *Timesheet* Praktikum

Setelah asisten lab mengajar selama satu bulan kurang lebih empat minggu, asisten lab diwajibkan mengisi *timesheet* kegiatan praktikum, dan admin biasanya memberitahukan jadwal pengisian *timesheet* yaitu pada saat menjelang akhir bulan, kemudian asisten lab mengisi *form timesheet* berda sarkan jumlah tatap muka yang

dilakukan selama praktikum dalam satu bulan. Ketika selesai diisi maka asisten lab mengumpulkan *timesheet* tersebut kepada admin, dan admin mengecek *timesheet* tersebut, jika sesuai maka admin merekap *timesheet* asisten lab, jika tidak maka asisten lab harus mengisi ulang *timesheet* tersebut. Setelah selesai admin memberikan hasil rekap tersebut kepada pihak DPPU, dan pihak DPPU memeriksa dan memproses *timesheet* asisten lab.

UNIVERSITAS ESA UNGGUL
Jl. Taman Asri, Td. Tembung, Kebon Jeruk, Jakarta Barat 13110
Telp : 021-567-4223
Fax : 021-566-2065, 567-4248 Indonesia
<http://www.esaunggul.ac.id>

Header Student Asisten **Paralel / Kelas :**

Nama Asisten : Nama Dosen :
Kode Asisten : Kode Dosen :
Mata Kuliah : Diklat MKK : (MKK)
Fakultas : Hari / Jam :
Jurusan : Ruang :
Besi / Kef : Basi / Kef :

Tanggal Kul	Tanggal	Topik dan Materi Pembelajaran	Jam Mengajar		Tanda Tangan	Keterangan
			Mulai	Selesai		

Catatan
1. Pengisian honorarium dimulai tanggal 1 sampai dengan tanggal terakhir bulan berjalan
2. Batas tanggal pengisian Time sheet sampai dengan minggu terakhir aktivitas pembelajaran di 1 bulan yang dijadwalkan
3. Time sheet harus diisi dengan jelas dan lengkap

Gambar 3.4.4.2 Form Timesheet Praktikum

3.5. Analisis Masalah Penjadwalan Praktikum

Berdasarkan fase penjadwalan pada proses bisnis berjalan tersebut, maka terdapat beberapa masalah yang muncul, adapun masalah tersebut adalah sebagai berikut :

1) Fase Pembuatan Jadwal Praktikum

Masalah yang terjadi pada fase ini adalah sebagai berikut :

Belum terkomputerisasinya rekapitulasi jadwal praktikum, sehingga harus dipindahkan dahulu ke dalam *excel*.

2) Fase Proses pemilihan jadwal praktikum

Masalah yang terjadi pada fase ini adalah sebagai berikut :

- a) Tidak meratanya informasi pemberitahuan pemilihan jadwal praktikum kepada asisten lab
- b) Tidak adanya batasan dalam pemilihan jadwal praktikum, sehingga terdapat asisten lab yang tidak mendapatkan jadwal praktikum
- c) Proses pemilihan jadwal asisten lab tidak merata dan mengalami kesalahan informasi, sehingga mengharuskan asisten lab berkomunikasi melalui email untuk informasi penambahan kelas atau pengurangan kelas dalam pemilihan jadwal.
- d) Adanya kesamaan jadwal dengan jadwal mata kuliah yang dimiliki oleh masing-masing asisten lab.

3) Fase Proses Konfirmasi Jadwal Kepada Dosen

- a) Proses konfirmasi harus dilakukan secara langsung melalui komunikasi Telepon/*Chatting* atau dengan mendatangi dosen yang bersangkutan.
- b) Terjadinya *Miss Communication* antara asisten lab dan dosen dalam mendiskusikan materi praktikum.

4) Fase Proses Pengisian *Timesheet* Kegiatan Asisten lab

- a) Informasi pemberitahuan pengisian *timesheet* asisten lab tidak merata, sering kali ada asisten lab yang telat atau tidak mengisi *timesheet* pada akhir bulan.

- b) Penggunaan banyak kertas/form pada saat pengisian *timesheet*.
 - c) Terjadinya kehilangan dokumen *timesheet*.
 - d) Adanya kesalahan pengisian *timesheet* yang dilakukan oleh asisten lab
- 5) Masalah lain diantaranya yaitu :
- a) Proses pemberian jadwal tidak direalisasikan berdasarkan bidang ilmu / keahlian yang dimiliki asisten lab.
 - b) Tidak adanya *knowledge sharing* antar asisten lab, jika kelas tersebut memiliki kesamaan matakuliah, sehingga pemberian materi praktikum tidak maksimal dilaksanakan.
 - c) Dosen tidak mengetahui asisten lab pengampu matakuliah dalam Jadwal praktikumnya.

3.6. Rencana Solusi Pemecahan Masalah

Berdasarkan analisis masalah di atas, maka rencana solusi pemecahan masalah yang akan dibuat yaitu :

1. Membangun sistem informasi penjadwalan asisten lab untuk menyelesaikan pokok permasalahan yang terjadi di Fakultas Ilmu Komputer Universitas Esa Unggul.
2. Membuat usulan penyelesaian masalah dengan menggunakan Metode RAD (*Rapid Application Development*) untuk mempercepat proses pembuatan aplikasi dengan fase-fase yang dimilikinya.
3. Membuat sistem informasi penjadwalan asisten lab yang berguna untuk mempermudah asisten lab dalam menerima notifikasi dan melihat jadwal praktikum .

4. Membuat sistem informasi penjadwalan asisten lab yang menghasilkan dokumentasi *timesheet* mengajar asisten lab.
5. Merancang kebutuhan sistem secara kebutuhan fungsional, kebutuhan lingkungan sistem dan kebutuhan pendukung dalam menyelesaikan masalah yang terjadi.
6. Membuat gambaran umum usulan perancangan sistem informasi penjadwalan, yang meliputi desain *use case*, proses bisnis usulan, *class* serta *sequence* diagram.

BAB IV

PEMBAHASAN

4.1. Fase Persencanaan Syarat-Syarat

Permasalahan yang terjadi dalam hal penjadwalan dijadikan sebagai acuan dalam melakukan perencanaan. Oleh karena itu, proses ini akan berfokus untuk mendapatkan gambaran fungsi atau fitur dari perangkat lunak yang akan dibangun. Perencanaan dimulai dengan membuat kesimpulan dari kumpulan cerita yang telah didapat dari hasil penelitian dan akan menjadi gambaran dasar dari sistem tersebut. Adapun proses perencanaan yang dibuat agar dapat mengatasi masalah sebagai berikut :

4.1.1. Perencanaan Kebutuhan Sistem Penjadwalan

Studi kasus yang diambil pada penelitian ini yaitu Asisten Lab Fakultas Ilmu Komputer Universitas Esa Unggul. Hasil (output) uji coba dari aplikasi diverifikasi secara manual untuk membuktikan tidak ada jadwal yang melanggar aturan yang telah ditetapkan. Agar suatu jadwal dapat dibuat dengan benar, terdapat sejumlah faktor dan aturan penjadwalan harus diperhatikan. Faktor-faktor yang berpengaruh dalam pembentukan jadwal meliputi:

1) Asisten Lab

Seorang Asisten Lab tidak dapat mengajar beberapa mata kuliah pada jam yang sama. Selain itu, hanya dapat mengajar pada jam-jam dan hari-hari tertentu saja, sehingga perlu untuk memesan jadwal khusus yang tidak dapat diganggu matakuliah yang dimiliki oleh asisten lab lain.

2) Ruang

Pemakaian ruang praktikum dilakukan di lab komputer universitas esa unggul, maka perlu diperhatikan ruang lab yang tersedia agar tidak mengganggu jalannya perkuliahan.

3) Waktu

Waktu merupakan batasan berapa menit yang diperlukan untuk satu jam kuliah. Selain itu, ada hari-hari yang jam kuliah dibatasi sampai dengan jam tertentu (misalnya jam kuliah malam dimulai jam 18.00 sampai 21.20 dan jam kuliah hari Jumat, Sabtu, Minggu dibatasi mulai jam 07.30 sampai jam 11.40 kemudian dimulai kembali jam 13.00 Sampai 21.20). Dengan batasan-batasan waktu ini, jadwal hanya akan berada pada waktu yang ditentukan.

4) Matakuliah

Mengingat setiap matakuliah memiliki semester mata kuliah itu diajarkan, maka perlu adanya aturan yang membatasi penjadwalan matakuliah, agar mata-kuliah itu sesuai dengan aturan-aturan penjadwalan. Aturan-aturan yang harus diperhatikan dalam membuat jadwal meliputi:

- a) Tidak boleh ada Nomor Induk Asisten Lab yang sama pada jam yang sama.
- b) Kemunculan matakuliah pada semester yang sama dibatasi maksimal dua matakuliah praktikum pada satu hari, dan enam matakuliah praktikum dalam satu minggu.
- c) Pada masing-masing semester, Asisten Lab dapat mengambil matakuliah praktikum sesuai dengan kurikulum yang ditentukan program studi sehingga jika jadwal praktikum bentok dengan matakuliah yang diambil

maka jadwal praktikum untuk asisten lab tidak bisa diambil karena jadwal yang bersamaan dengan jadwal matakuliah lain pada semester yang sama.

- d) Dalam satu hari, Asisten Lab mengajar dua matakuliah dengan maksimal dua kelas praktikum dengan jam yang berbeda, dan dalam seminggu maksimal asisten lab mengajar enam matakuliah dengan satu atau dua kelompok praktikum..
- e) Pembagian jadwal matakuliah praktikum dilakukan sesuai bidang rumpun ilmu / bidang keahlian asisten lab agar tidak terjadi kesalahan pemberian materi praktikum, adapun bidang rumpun ilmu yang diusulkan adalah sebagai berikut :

Tabel 4.1.1. Bidang Rumpun Ilmu

No.	Bidang Rumpun Ilmu	Bidang Keahlian	Deskripsi Keahlian
1.	Sistem Informasi	Pemrograman	<ul style="list-style-type: none"> a) Lulus matakuliah pemrograman dasar dengan minimal nilai B+, jika kurang dari nilai minimal tersebut, maka tidak boleh mengajar. b) Memiliki IPK min 3,25 c) Memahami logika dan struktur algoritma d) Mampu membuat program terstruktur e) Mampu berkomunikasi dengan baik f) Mampu mempresentasikan materi dengan baik dan benar.
		Desain Basis Data	<ul style="list-style-type: none"> a) Lulus matakuliah pemrograman dengan minimal nilai B+, jika kurang dari nilai minimal tersebut, maka tidak boleh mengajar.

			<ul style="list-style-type: none"> b) Memiliki IPK min 3,25 c) Menguasai desain basis data seperti : ERD, Conceptual Desain dan lain lain d) Menguasai tools, oracle, MySQL Server, MySQL dan lain lain e) Mampu berkomunikasi dengan baik dan Mampu mempresentasikan materi dengan baik dan benar.
		Analisis Dan Desain Sistem	<ul style="list-style-type: none"> a) Lulus matakuliah Dasar Sistem Informasi / Pemodelan Proses Bisnis / Anallisis Kebutuhan Informasi / Sistem Informasi Manajemen dengan minimal 3 Matkul Dengan nilai B+, jika kurang dari nilai minimal tersebut, maka tidak boleh mengajar. b) Memiliki IPK min 3,25 c) Menguasai konsep <i>flowchart</i> d) Menguasai konsep objek dengan UML (Unified Modelling Language) e) Menguasai konsep desain interface dalam analisis perancangan sistem. f) Menguasai Tools Seperti : <i>Enterprise Arsitechtur, Ms. Visio, Balsamiq Mockup</i> Dan Lain Lain. g) Mampu berkomunikasi dengan baik dan Mampu mempresentasikan materi dengan baik dan benar.
		Sistem Operasi Dan Jaringan	<ul style="list-style-type: none"> a) Lulus matakuliah Organisasi Dan Manajemen / Arsitektur dan organisasi komputer / Pemrograman dengan minimal 2 Matkul Dengan nilai B+, jika kurang dari nilai minimal tersebut, maka tidak boleh mengajar. b) Memiliki IPK 3,25

			c) Menguasai Sistem operasi Linux d) Memahami <i>Shell Programming</i> e) menguasai jaringan dan konfigurasi komputer f) Mampu berkomunikasi dengan baik dan Mampu mempresentasikan materi dengan baik dan benar.
2.	Teknik Informatika	Pemrograman	a) Lulus matakuliah pemrograman dasar dengan minimal nilai B+, jika kurang dari nilai minimal tersebut, maka tidak boleh mengajar. b) Memiliki IPK min 3,25 c) Memahami logika dan struktur algoritma d) Mampu membuat program terstruktur e) Mampu berkomunikasi dengan baik dan Mampu mempresentasikan materi dengan baik dan benar.
		Kriptografi	a) Lulus matakuliah Matematika Diskrit / Algoritma / Pemrograman Dasar dengan minimal 3 Matkul Dengan nilai B+, jika kurang dari nilai minimal tersebut, maka tidak boleh mengajar. b) Memiliki IPK min 3,25 c) Memahami logika dan struktur algoritma d) Menguasai bit, binary, XOR dalam enkripsi dan deskripsi e) Mampu membuat program terstruktur dalam melakukan enkripsi dan deskripsi f) Mampu berkomunikasi dengan baik dan Mampu mempresentasikan materi dengan baik dan benar.
		Sistem Operasi Dan Jaringan	a) Lulus matakuliah Organisasi Dan Manajemen / Arsitektur dan organisasi komputer / Pemrograman dengan minimal 2 Matkul Dengan nilai B+, jika

			<p>kurang dari nilai minimal tersebut, maka tidak boleh mengajar.</p> <p>b) Memiliki IPK 3,25</p> <p>c) Menguasai Sistem operasi Linux</p> <p>d) Memahami <i>Shell Programming</i></p> <p>e) menguasai jaringan dan konfigurasi komputer</p> <p>f) Mampu berkomunikasi dengan baik dan Mampu mempresentasikan materi dengan baik dan benar.</p>
		Analisis Dan Desain Sistem	<p>a) Lulus matakuliah Dasar Sistem Informasi / Desain Dan Analisa Algoritma / Rekayasa Perangkat Lunak dengan minimal 2 Matkul Dengan nilai B+, jika kurang dari nilai minimal tersebut, maka tidak boleh mengajar.</p> <p>b) Memiliki IPK min 3,25</p> <p>c) Menguasai konsep <i>flowchart</i></p> <p>d) Menguasai konsep objek dengan UML (Unified Modelling Language)</p> <p>e) Menguasai konsep desain interface dalam analisis perancangan sistem.</p> <p>f) Menguasai Tools Seperti : <i>Enterprise Arsitektur, Ms. Visio, Balsamiq Mockup</i> Dan Lain Lain.</p> <p>h) Mampu berkomunikasi dengan baik dan Mampu mempresentasikan materi dengan baik dan benar.</p>
		Distribusi Query Basis Data	<p>a) Lulus matakuliah basis data dengan minimal nilai B+, jika kurang dari nilai minimal tersebut, maka tidak boleh mengajar.</p> <p>b) Memiliki IPK min 3,25</p> <p>c) Memahami NoSQL seperti MongoDB</p>

			d) Menguasai desain basis data seperti : ERD, Conceptual Desain dan lain lain. e) Menguasai Pemrograman Minimal PHP <i>Framework / PHP Native</i> . f) Mampu berkomunikasi dengan baik dan Mampu mempresentasikan materi dengan baik dan benar.
--	--	--	---

4.1.1.1. Perencanaan Prioritas Notifikasi Android

Berdasarkan hasil analisis dan rancangan sistem usulan penjadwalan, maka Sistem notifikasi yang akan dibuat memungkinkan pengguna untuk mendapatkan informasi tentang peristiwa yang relevan dan tepat waktu dalam aplikasi. Berikut ini prioritas notifikasi dalam Sistem Informasi penjadwalan Asisten Lab :

Tabel 4.1.1.1. Kebutuhan Prioritas Notifikasi Android

No.	Jenis Notifikasi	Nama Notifikasi	Keterangan
1.	<i>Highest (HIGH)</i>	Pemberitahuan jadwal praktikum selama 3 kondisi yaitu : 1 jam sebelum kelas, 30 menit sebelum kelas dan 10 menit sebelum kelas.	mendapat notifikasi mengajar praktikum, dengan 3 kondisi waktu notifikasi, agar asisten lab dapat melakukan konfirmasi kehadiran di notifikasi tersebut.
2.	<i>Maximum (MAX)</i>	<ul style="list-style-type: none"> ➤ Pemberitahuan chat antar asisten, serta asisten dengan dosen ➤ Pemberitahuan pengisian <i>timesheet</i> jadwal praktikum 	<ul style="list-style-type: none"> ➤ Melakukan komunikasi antar asisten lab dan dosen pengampu agar tidak terjadi kesalahan dalam praktikum.

			➤ Memberikan notifikasi pemberitahuan pengisian <i>timesheet</i> agar lebih terorganisir dan tepat.
3.	<i>Lowest (LOW)</i>	➤ Pemberitahuan konfirmasi kedatangan jam mengajar praktikum	➤ Melakukan konfirmasi Ya/Tidak kehadiran mengajar.
4.	<i>Minimalization (MIN)</i>	➤ Pemberitahuan pengisian jadwal koordinasi ➤ Pemberitahuan pengisian jadwal praktikum	➤ Melakukan pembuatan jadwal pengisian jadwal koordinasi untuk menentukan waktu kosong asisten lab ➤ Melakukan pengisian jadwal praktikum sesuai dengan bidang rumpun ilmu

4.1.1.2. Perencanaan Kebutuhan Fungsional

Setelah menganalisa masalah yang terjadi pada proses bisnis yang dilakukan oleh Sistem Penjadwalan yang berjalan, Maka Analisa kebutuhan bisnis terhadap kebutuhan sistem merupakan langkah awal untuk menentukan perangkat lunak seperti apa yang akan dihasilkan, ketika melaksanakan pembangunan perangkat lunak. Berikut adalah analisis kebutuhan fungsional bisnis dan sistem :

Tabel 4.1.1.2. Kebutuhan Fungsional Bisnis Dan Sistem

No.	Permasalahan	Kebutuhan Pengguna	Kebutuhan Sistem	Use Case
1.	Tidak adanya batasan dalam pemilihan jadwal praktikum, sehingga terdapat asisten lab yang tidak mendapatkan jadwal praktikum.	Asisten lab harus Mengisi <i>Form</i> Keahlian Bidang Ilmu, agar pembagian jadwal merata	Sistem menampilkan form keahlian bidang ilmu, dan menampilkan jadwal praktikum sesuai dengan keahlian bidang ilmu yang dimiliki.	Mengisi <i>Form</i> Keahlian Bidang Ilmu
2.	Adanya jadwal bentrok dengan jadwal kuliah yang dimiliki oleh masing-masing asisten lab.	Asisten lab harus mengisi <i>form</i> jadwal tersedia, agar jadwal kuliah yang dimiliki asisten lab, tidak bentrok dengan jadwal mengajar.	Sistem menampilkan form jadwal tersedia, agar asisten dapat mengisi dengan menampilkan hari dan jam waktu kosong yang dimiliki asisten lab.	Mengisi <i>Form</i> Jadwal Tersedia
3.	Tidak meratanya informasi pemberitahuan pemilihan jadwal praktikum kepada asisten lab	Asisten lab dapat menerima notifikasi pemberitahuan kegiatan seperti notifikasi jadwal.	Sistem dapat menampilkan notifikasi pemberitahuan dalam priority seperti (Max, Min, Low Dan High) yang telah ditentukan	Pemberitahuan Jadwal

No.	Permasalahan	Kebutuhan Pengguna	Kebutuhan Sistem	Use Case
4.	Proses konfirmasi harus dilakukan secara langsung melalui komunikasi telepon / chatting Atau dengan mendatangi dosen yang bersangkutan, dan Terjadinya <i>Miss Communication</i> antara asisten lab dan dosen dalam mendiskusikan materi praktikum.	Asisten dapat melakukan konfirmasi jadwal melalui aplikasi <i>mobile device</i> dan menggunakan fitur chatting sebagai media komunikasi antara dosen pengampu dan asisten lab.	Sistem dapat menampilkan fitur <i>chatting</i> sebagai media komunikasi internal dalam aplikasi.	Penjadwalan Dan Chatting
5.	Belum terkomputerisasinya rekapitulasi jadwal praktikum dan Tidak meratanya informasi pemberitahuan pemilihan jadwal praktikum kepada asisten lab	Koordinator Asisten Lab dapat melakukan penjadwalan praktikum ketika asisten lab telah mengisi <i>form</i> keahlian bidang ilmu yang dimilikinya.	Sistem dapat melakukan penjadwalan praktikum sesuai dengan bidang ilmu asisten lab.	Penjadwalan Praktikum

No.	Permasalahan	Kebutuhan Pengguna	Kebutuhan Sistem	Use Case
6.	Adanya kesalahan pengisian <i>timesheet</i> yang dilakukan oleh asisten lab, dan Terjadinya kehilangan dokumen <i>timesheet</i> .	Asisten lab dapat melakukan pengisian <i>timesheet</i> setelah kegiatan praktikum selesai, dan koordinator asisten lab dapat mengelola <i>timesheet</i> tersebut.	Sistem dapat menampilkan form pengisian <i>timesheet</i> asisten lab, dan sistem dapat mengorganisir <i>timesheet</i> asisten lab.	Form pengisian <i>timesheet</i> dan mengelola <i>timesheet</i> asisten lab
7.	Dosen tidak mengetahui Asisten lab pengampu matakuliah dalam Jadwal praktikumnya.	Dosen dapat melihat detail jadwal praktikum yang dimilikinya untuk mengetahui asisten lab pengampu dimatakuliah praktikumnya	Sistem dapat menampilkan jadwal praktikum yang dimiliki oleh dosen pengampu.	Lihat Jadwal Praktikum

4.1.1.3. Perencanaan Kebutuhan Lingkungan Sistem

A. Perangkat Keras

Spesifikasi perangkat keras untuk *server* yang dibutuhkan sebagai berikut :

1. *Processor* dengan kecepatan minimal 1.8 Ghz.
2. RAM dengan kapasitas minimal 2GB
3. *Monitor, Keyboard dan Mouse.*
4. *Wifi adapter.*

Spesifikasi perangkat keras untuk *User* yang dibutuhkan yaitu *Android Device.*

B. Perangkat Lunak

Spesifikasi perangkat lunak untuk server yang dibutuhkan sebagai berikut :

1. Sistem Operasi Windows 7.
2. XAMPP.

Spesifikasi perangkat lunak Android untuk *User* yang dibutuhkan minimal sistem operasi android *Jellybean*

4.2. Fase Perancangan

Rancangan atau pemodelan sistem yang akan dibangun digambarkan menggunakan diagram UML antara lain :

1. *Use Case Diagram*
2. *Activity Diagram*
3. *Class Diagram*
4. *Sequence Diagram*

Pada proses desain sistem ini ada 3 aktor yang akan berhubungan langsung dengan sistem yaitu :

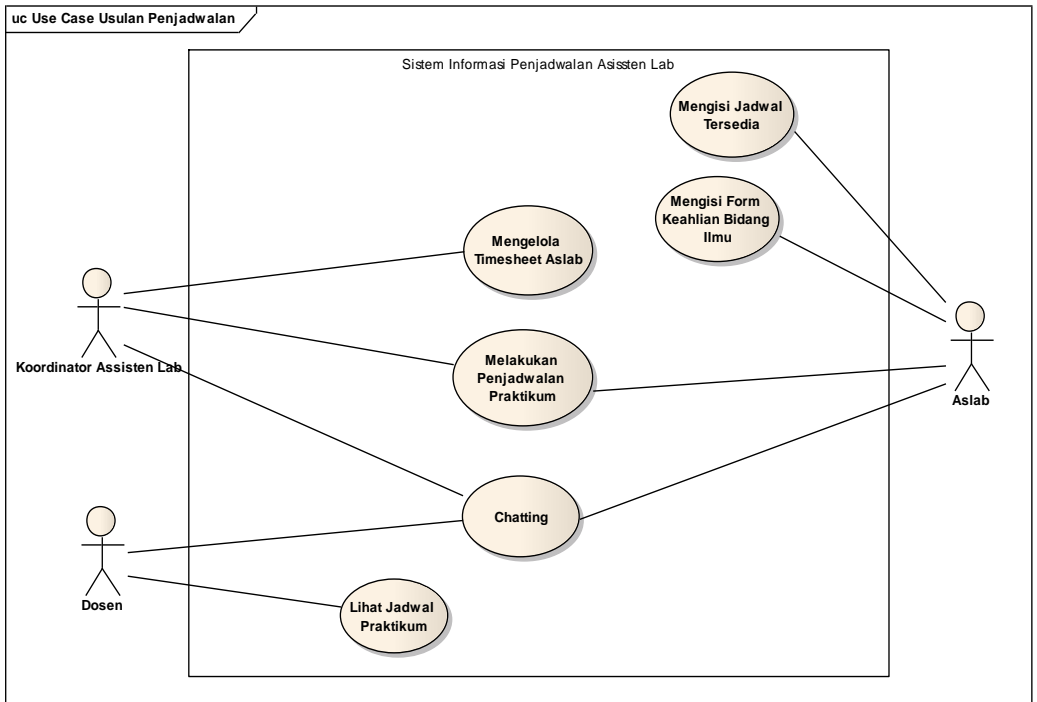
1. Kordinator Asisten Lab

2. Asisten Lab
3. Dosen

4.2.1. Use Case Diagram

Diagram ini merupakan pemodelan untuk mendeskripsikan sebuah interaksi antar satu atau lebih aktor dengan sistem yang akan dibangun. *Use Case Diagram* menggambarkan bagaimana dan dengan cara apa untuk berinteraksi dengan sistem.

Adapun desain sistem usulan digambarkan pada *use case* diagram berikut ini :



Gambar 4.2.1. *Use Case Diagram* Usulan

Penjelasan :

1. Kordinator Asisten Lab dapat mengelola menu yang terdapat didalam sistem, yaitu : Mengelola *timesheet* asisten lab, melakukan penjadwalan praktikum, dapat menggunakan fitur *chatting* pada aplikasi.
2. Asisten Lab dapat mengisi jadwal yang tersedia sesuai dengan kesediannya mereka dan mengisi *form* keahlian bidang ilmu.
3. Dosen dapat melihat jadwal praktikum serta menggunakan fitur *chat* untuk berkomunikasi dengan kordinator dan asisten lab.

4.2.2. Proses Bisnis Usulan Penjadwalan Asisten Lab Dengan Notifikasi Berbasis *Mobile Device*

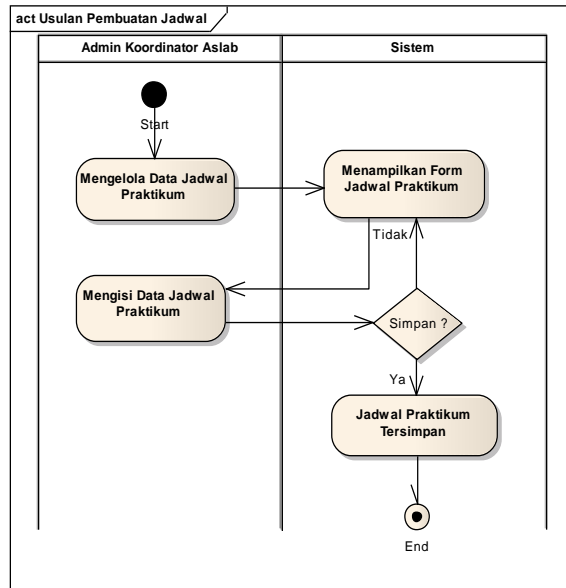
Berikut ini akan dijelaskan rancangan proses bisnis usulan untuk sistem penjadwalan asisten lab. Rancangan proses bisnis usulan digambarkan dalam bentuk *Activity Diagram* dibawah ini :

4.2.2.1. *Activity Diagram* – Fase Pembuatan Jadwal Praktikum

Koordinator Asisten Lab selaku admin bertugas untuk mengelola data jadwal praktikum yang nantinya akan menjadi pilihan kepada para asisten lab untuk mendaftar sebagai pengajar mata kuliah praktek.

Untuk mengelola data tersebut, admin dapat membuka menu kelola data jadwal praktikum, kemudian sistem akan menampilkan *form* jadwal praktikum yang akan diatur oleh Kordinator Asisten Lab.

Adapun proses bisnis yang terjadi dapat dilihat pada *activity diagram* dibawah ini :

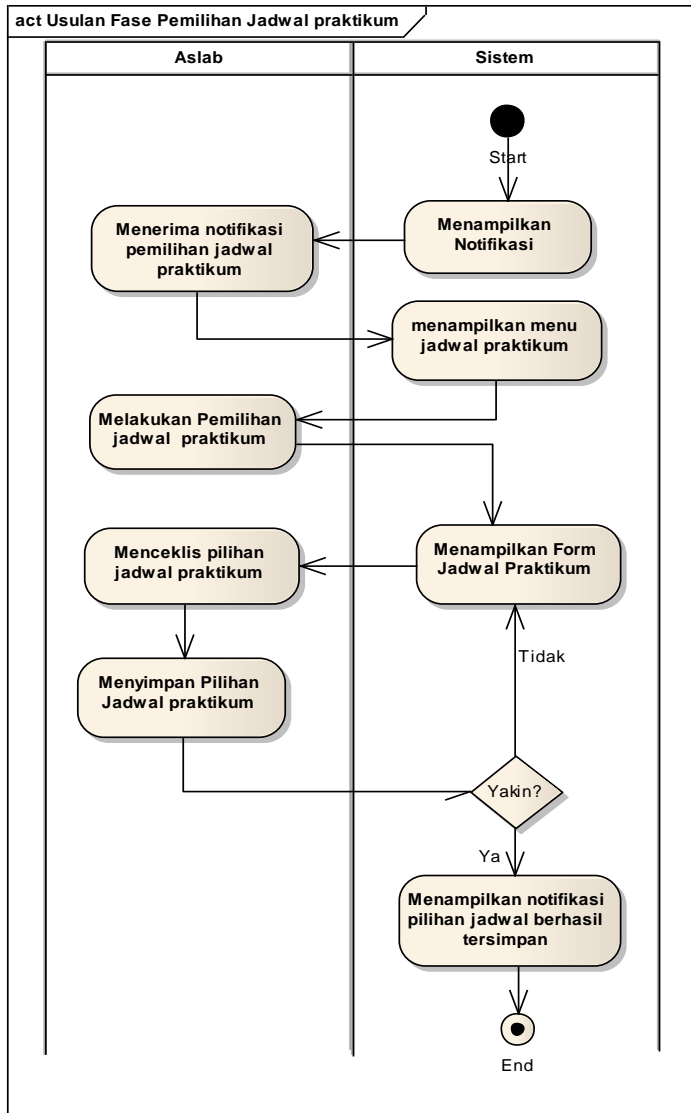


Gambar 4.2.2.1. Activity Diagram Pembuatan Jadwal Praktikum

4.2.2.2. Activity Diagram – Pemilihan Jadwal Praktikum

Setelah jadwal yang telah di input oleh koordinator asisten lab, kemudian sistem akan memberikan notifikasi kepada asisten lab bahwa jadwal telah tersedia. Setelah itu, asisten lab dapat memilih jadwal mengajar sesuai dengan ketersediaan waktu mereka.

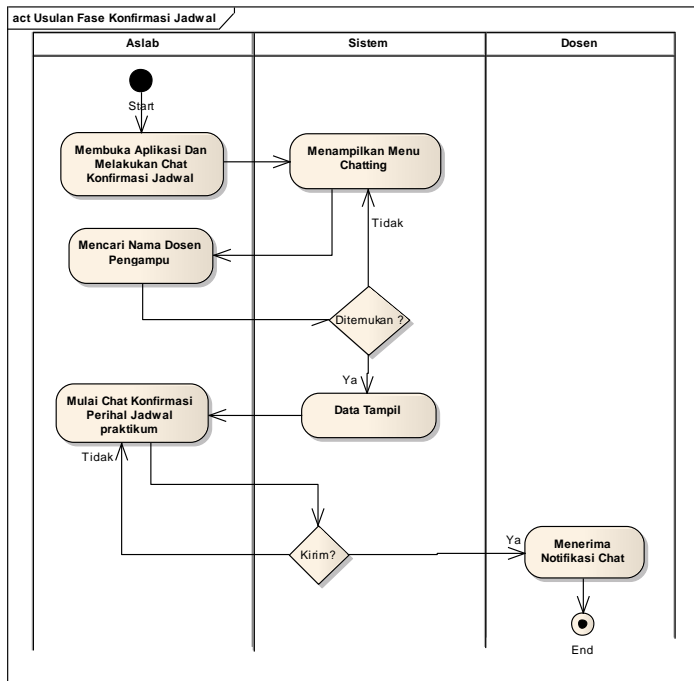
Setelah memilih jadwal, asisten lab dapat menyimpan jadwal tersebut dan sistem akan menampilkan notifikasi bahwa jadwal telah dipilih. Adapun proses bisnis yang terjadi dapat dilihat pada *activity diagram* dibawah ini :



Gambar 4.2.2.2. *Activity Diagram* Pemilihan Jadwal Praktikum

4.2.2.3. Activity Diagram – Konfirmasi Jadwal Praktikum Kepada Dosen

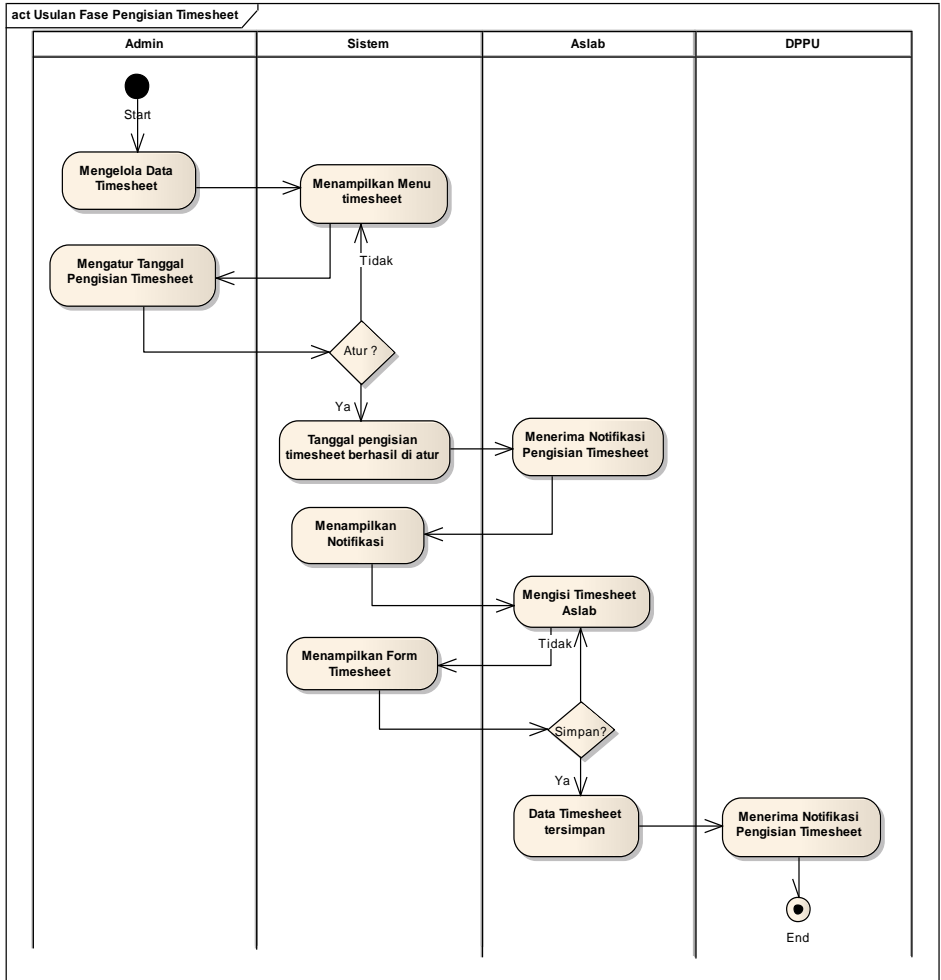
Setelah memilih dan menyimpan jadwal. Asisten lab diharuskan melakukan konfirmasi kepada dosen pengampu. Dilakukan dengan cara membuka aplikasi lalu memilih fitur *chat*. Lalu, mencari dan memilih dosen pengampu. Jika data yang dicari ada, maka sistem akan menampilkan nama dosen. Setelah itu, asisten lab dapat memulai *chat* untuk melakukan konfirmasi perihal jadwal praktikum. Adapun proses bisnis yang terjadi dapat dilihat pada *activity diagram* dibawah ini :



Gambar 4.2.2.3. Activity Diagram Konfirmasi Jadwal Praktikum

4.2.2.4. Activity Diagram – Pengisian Timesheet

Adapun proses bisnis yang terjadi tergambar dalam *activity diagram* dibawah ini :



Gambar 4.2.2.4. Activity Diagram Pengisian Timesheet

Data *timesheet* pada aplikasi terlebih dahulu diatur oleh admin. Ketika berhasil, sistem akan memberikan notifikasi kepada masing-masing asisten lab. Lalu aslab dapat mengisi *timesheet* pada *form input* yang telah disediakan.

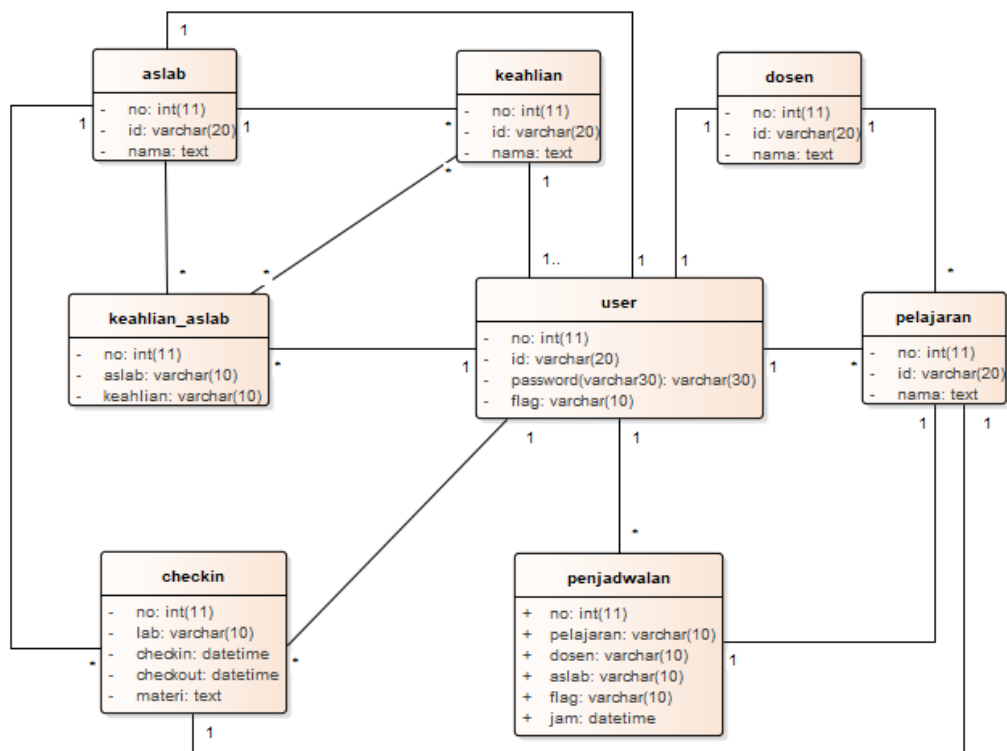
Setelah selesai, asisten lab dapat menyimpan data *timesheet* mereka yang nantinya data tersebut dapat diterima oleh DPPU.

4.3. Class Diagram

Diagram kelas atau *Class Diagram* merupakan diagram yang dibuat untuk menggambarkan kelas-kelas yang ada didalam sistem yang akan dibangun. Berikut ini merupakan *class diagram* sistem informasi penjadwalan yang akan dibangun dengan keterangan :

1. Satu *user* (aslab) dapat mempunyai banyak keahlian
2. Satu *user* (aslab) dapat melakukan banyak *checkin*
3. Satu *user* (admin/koord. Aslab) dapat mengelola banyak jadwal
4. Satu *user* (aslab) dapat mengajar banyak pelajaran
5. Satu *user* (aslab) dapat berkomunikasi dengan banyak dosen

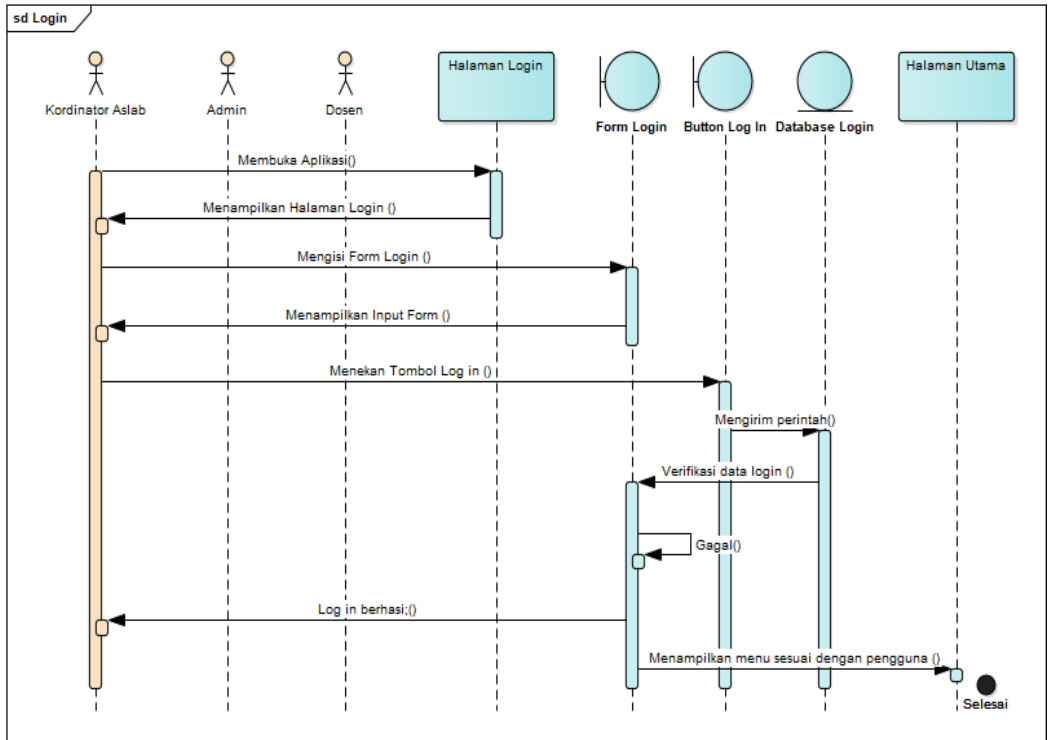
class Class Penjadwalan Aslab



Gambar 4.3. Class Diagram

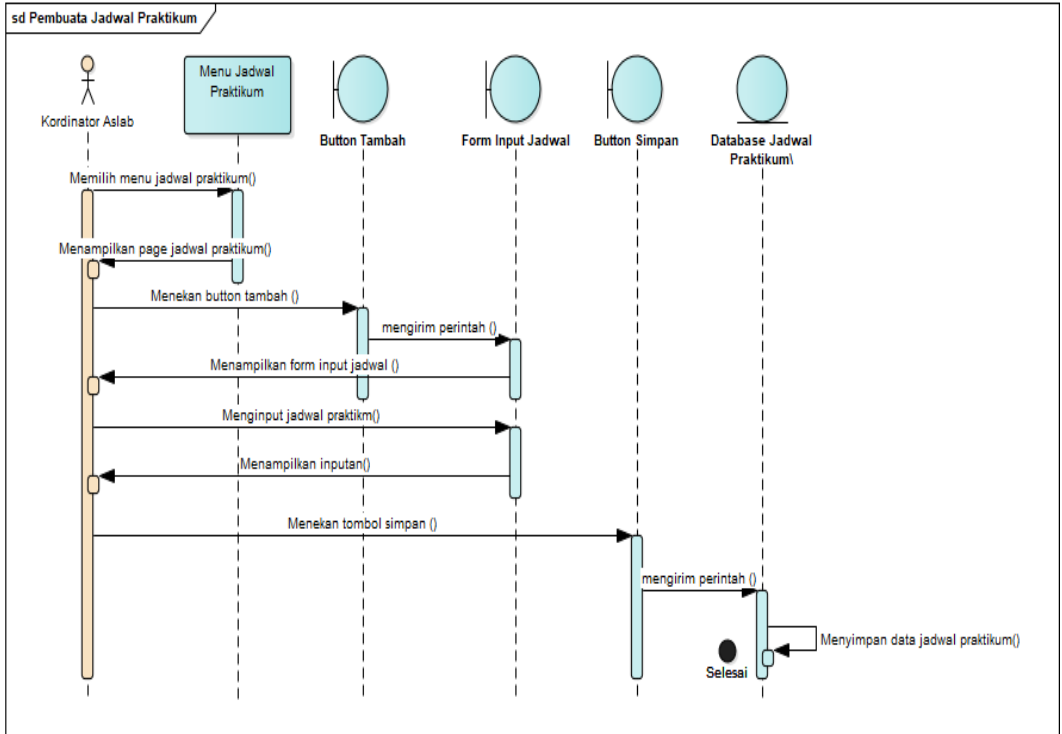
4.4. Sequence Diagram

4.4.1. Sequence Diagram Login



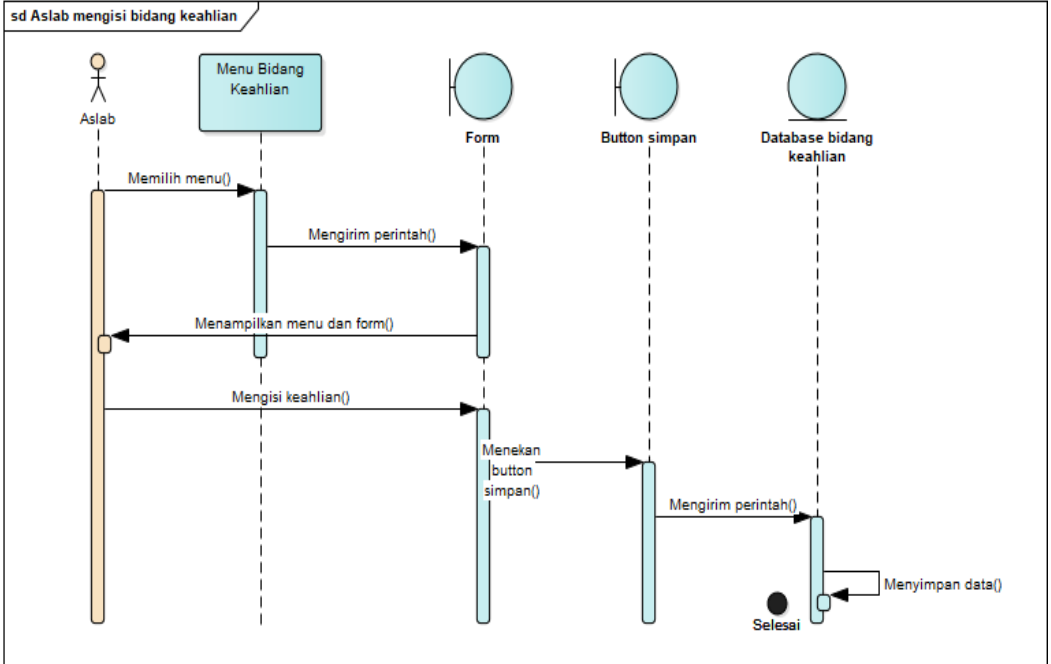
Gambar 4.4.1. Sequence Diagram Login

4.4.2. Sequence Diagram Pembuatan Jadwal Praktikum



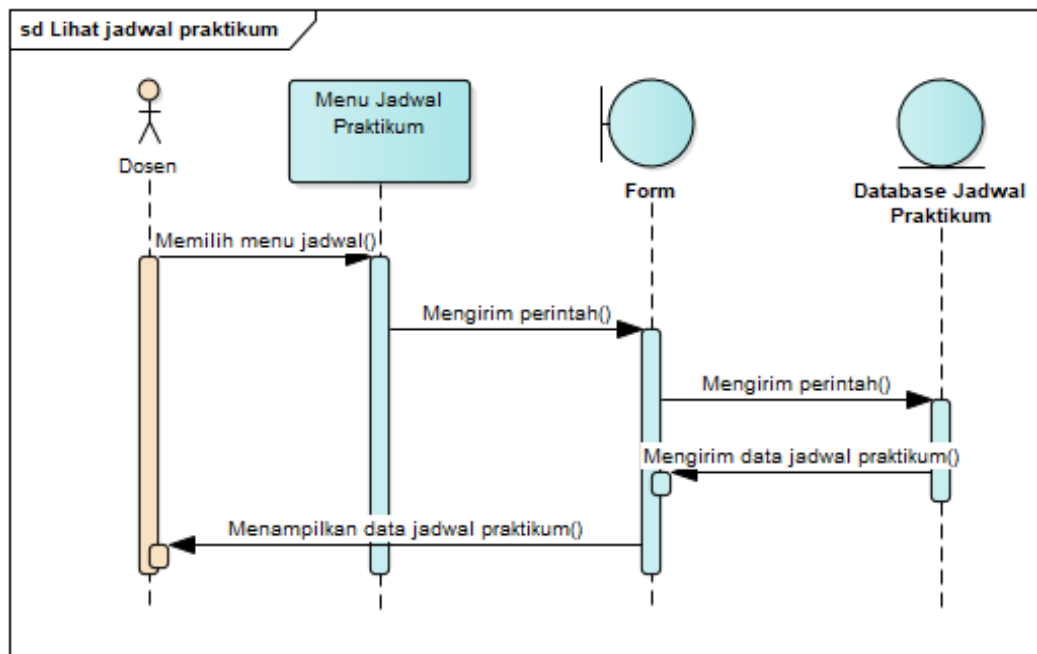
Gambar 4.4.2. Sequence Diagram Mengisi Jadwal Praktikum

4.4.3. Sequence Diagram Mengisi Bidang Keahlian



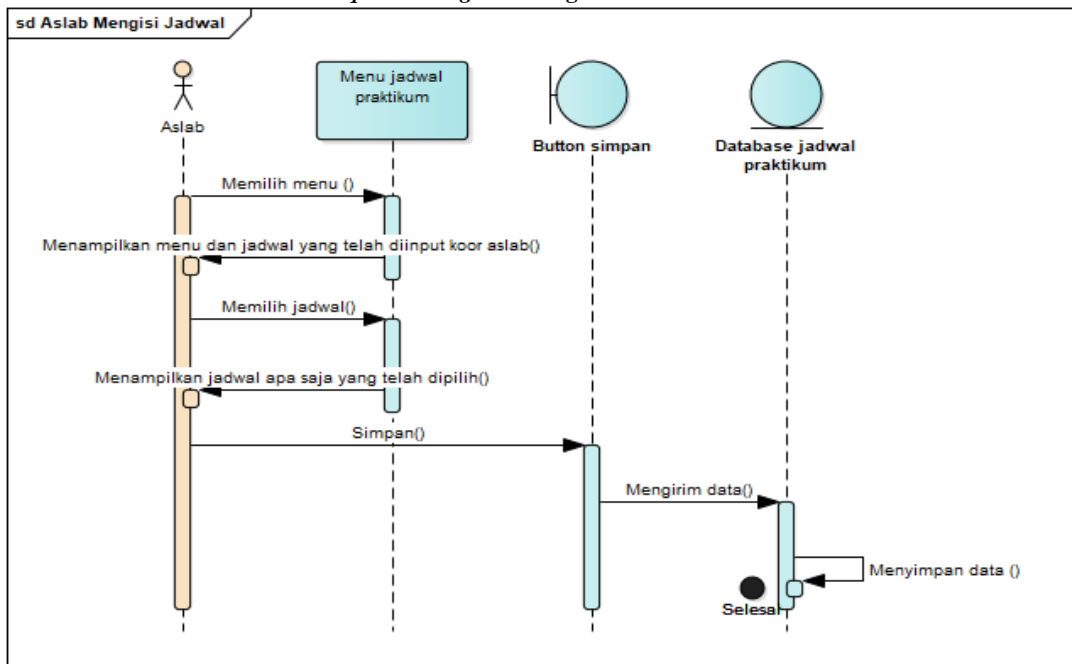
Gambar 4.4.3. Sequence Diagram Mengisi Bidang Keahlian

4.4.4. Sequence Diagram Lihat Jadwal Praktikum



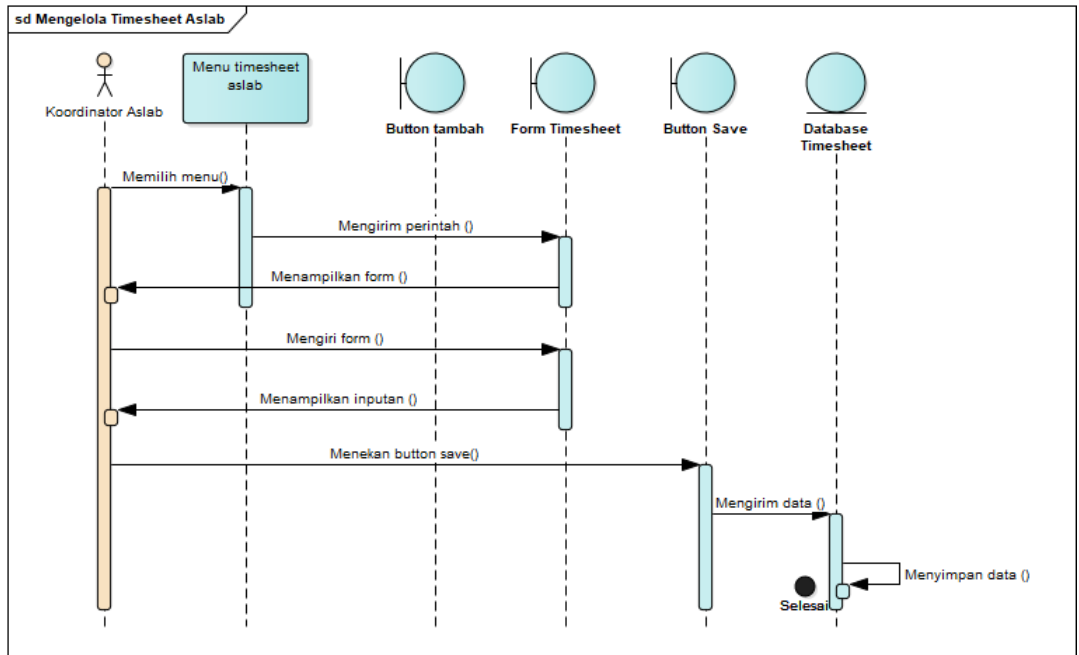
Gambar 4.4.4. Sequence Diagram Lihat Jadwal Praktikum

4.4.5. Sequence Diagram Mengisi Jadwal Praktikum



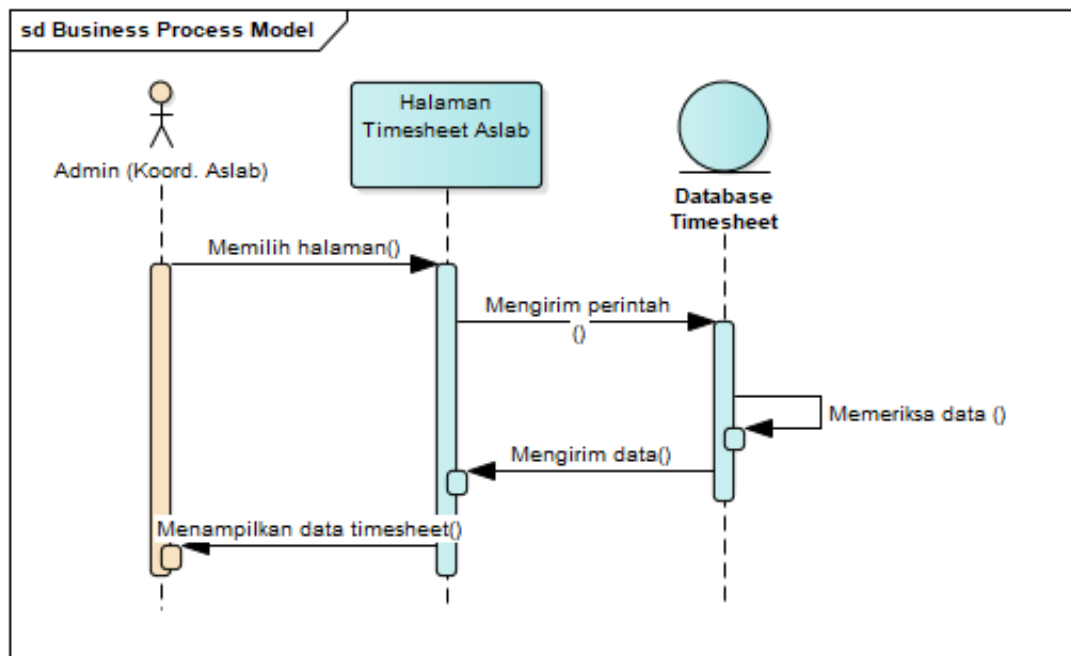
Gambar 4.4.5. Sequence Diagram Mengisi Jadwal Praktikum

4.4.6. Sequence Diagram Mengelola Timesheet



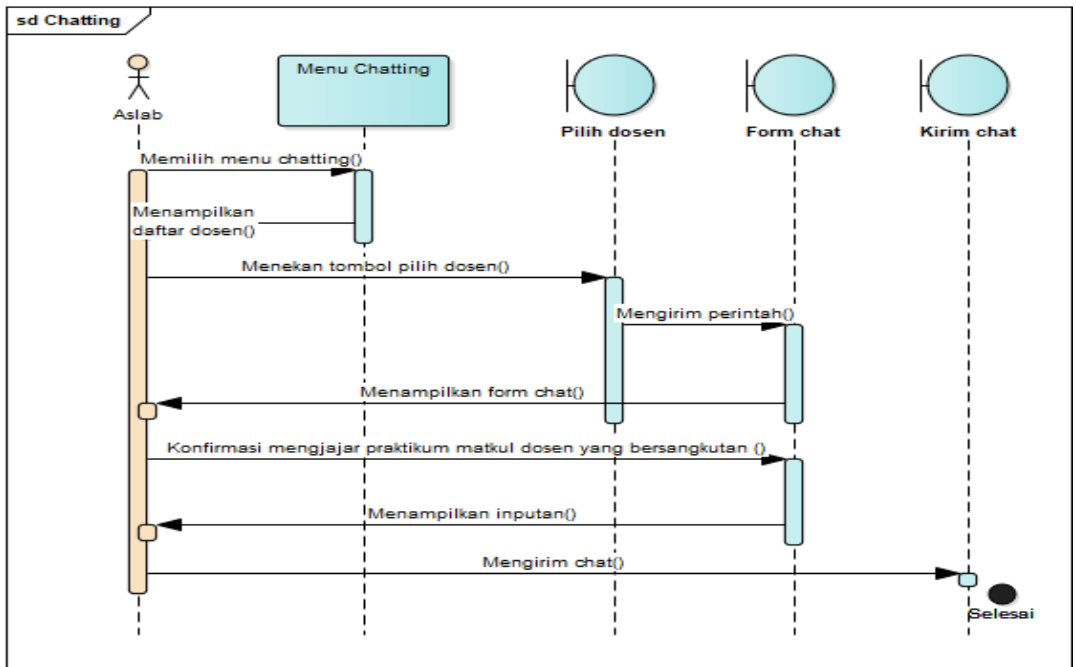
Gambar 4.4.6. *Sequence Diagram* Mengelola Time Sheet Aslab

4.4.7. Sequence Diagram Lihat Timesheet



Gambar 4.4.7. Sequence Diagram Lihat Timesheet Aslab

4.4.8. Sequence Diagram Chatting



Gambar 4.4.8. Sequence Diagram Chatting

4.5. Fase Pengkodean (*Coding Phase*)

Setelah tahap desain, selanjutnya membangun sistem berdasarkan rancangan yang telah dibuat. Tahap ini dibangun database yang dibutuhkan menggunakan MySQL dan pembuatan fungsi pada sistem dengan menggunakan bahasa pemrograman Java serta PHP untuk menjalankan fungsi *web service*.

4.6. Fase Pengujian (*Testing*)

Tahapan ini merupakan tahap akhir dalam metode pengembangan *Rapid Application Development* (RAD), dimana sistem yang telah berhasil dibangun siap untuk diuji coba. *Testing* akan dilakukan dengan metode *Blackbox* yang akan dijelaskan di bawah ini.

4.6.1. *Blackbox Testing*

Pengujian dilakukan untuk mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Pengujian akan dilakukan terhadap fungsi *interface*, dan *form handle* dari sistem.

4.6.1.1. Pengujian *Blackbox* Terhadap *User Interface* Sistem

Pengujian *interface* sistem bertujuan untuk mengetahui fungsionalitas dari elemen-elemen *interface* yang terdapat didalam halaman sistem. Elemen – elemen yang diujikan terutama dalam elemen *button*.

Pengujian interface sistem kali ini yaitu pada proses mengelo data di menu aslab pada aplikasi penjadwalan aslab yang akan dijelaskan didalam tabel dibawah ini :

Tabel 4.6.1.1. Blackbox Interface Sistem

No.	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1.	Pengujian tombol tambah aslab	Aplikasi akan menampilkan form untuk menambahkan data aslab	Sesuai harapan	Valid
2.	Pengujian tombol simpan/save	Sistem akan menyimpan data aslab dan menampilkan daftar aslab yang telah ditambah	Sesuai harapan	Valid
3.	Pengujian Tahan dan tekan nama aslab	Sistem akan menampilkan data diri aslab yang dipilih	Sesuai harapan	Valid
4.	Pengujian tombol hapus	Menghapus daftar aslab yang dipilih dari tampilan	Sesuai harapan	Valid
5.	Pengujian Tombol edit	Sistem akan mengganti data aslab yang lama dengan data yang baru	Sesuai harapan	Valid

4.7. User Interface

4.7.1. Log in



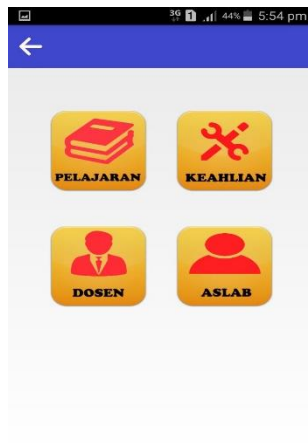
Gambar 4.7.1. User Interface Log in

4.7.2. Menu Utama Admin



Gambar 4.7.2. User Interface Menu Admin

4.7.3. Menu Master Data



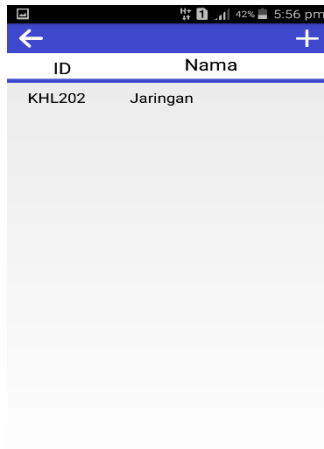
Gambar 4.7.3. *User Interface* Menu Master Data

4.7.4. Menu Mata Kuliah

ID	Nama
CCS220	Struktur Data
CSP212	Perancangan Basis Data
CCJ121	Jaringan Komputer
CSA123	Algoritma Dan
CCS111	Sistem Basis Data
CCS113	Sistem Operasi
CSJ205	Pemrograman Java
CCP119	Pemrograman Berbasis

Gambar 4.7.4. *User Interface* Menu Mata Kuliah

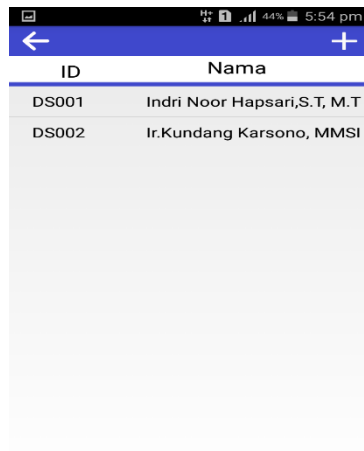
4.7.5. Menu Keahlian Aslab



ID	Nama
KHL202	Jaringan

Gambar 4.7.5. *User Interface* Menu Keahlian Aslab


4.7.6. Menu Tambah Dosen



ID	Nama
DS001	Indri Noor Hapsari,S.T, M.T
DS002	Ir.Kundang Karsono, MMSI

Gambar 4.7.6. *User Interface* Menu Mata Tambah Dosen

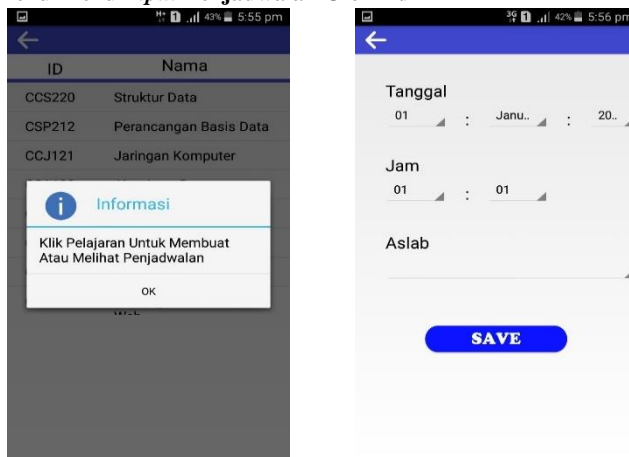
4.7.7. Menu Admin *Input Data Aslab*



ID	Nama
ASL101	Sawali Wahyu
ASL102	Gilang
ASL103	Elandi

Gambar 4.7.7. *User Interface Menu Input Data Aslab*

4.7.8. Menu Menu *Input Penjadwalan Oleh Admin*



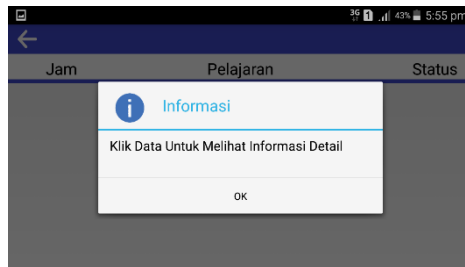
The left screenshot shows a list of courses with an 'Informasi' dialog box overlay. The dialog box contains the text: 'Klik Pelajaran Untuk Membuat Atau Melihat Penjadwalan' and an 'OK' button.

The right screenshot shows the scheduling form with the following fields:

- Tanggal: 01 : Janu.. : 20..
- Jam: 01 : 01
- Aslab: [Empty field]
- SAVE button

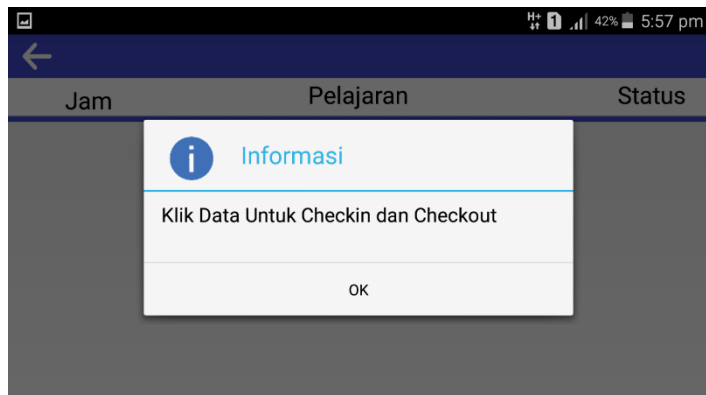
Gambar 4.7.8. *User Interface Menu Input Penjadwalan Oleh Admin*

4.7.9. Menu Memeriksa Hasil *Input Timesheet*



Gambar 4.5.9. *User Interface Menu Input Timesheet*

4.7.10. Menu *Input Timesheet* Oleh Aslab



Gambar 4.7.10. *User Interface Menu Timesheet Oleh Aslab*

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Setelah merancang dan membangun sistem informasi penjadwalan asisten lab (asisten lab) fakultas ilmu komputer dengan notifikasi berbasis *mobile device*, maka dapat ditarik kesimpulan sebagai berikut :

- 1) Penerapan aplikasi penjadwalan berbasis notifikasi dapat membantu mengingatkan asisten lab Fakultas Ilmu Komputer terkait jadwal mengajar praktikum yang telah diambil.
- 2) Dengan menerapkan aplikasi yang mengatur penjadwalan berdasarkan keahlian yang dimiliki aslab, penguasaan materi oleh dapat terealisasi di kelas.
- 3) Dengan menerapkan aplikasi penjadwalan, asisten lab memiliki jadwal mengajar yang jelas dan kesamaan jadwal mengajar ataupun adanya mata kuliah dapat dihindari.
- 4) Dengan adanya fitur *chatting* dalam aplikasi penjadwalan, *sharing* informasi terkait jadwal mengajar mata kuliah dengan dosen yang bersangkutan dapat dilakukan secara mudah.

5.2. Saran

Dalam Tugas Akhir ini masih memiliki kekurangan, maka diperlukan penyampaian saran yang nantinya dapat digunakan peneliti selanjutnya untuk menyempurnakan sistem dimasa yang akan datang. Adapun saran yang diberikan untuk pengembangan sistem untuk kedepannya :

- 1) Aplikasi dapat menghitung gaji aslab secara langsung terkait dengan *timesheet* aslab.

- 2) Terintegrasi dengan Departemen Administrasi Akademik (DAA), agar mempermudah dapat melakukan pembayaran gaji aslab.

DAFTAR PUSTAKA

BUKU :

- A.S, Rosa dan Shalauhuiddin, M. (2016). *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung : Informatika Bandung.
- Anton, M, Mulyono. (2010). *Aktivitas Belajar*. Bandung :Yrama
- H, Nazaruddin Safaat (2015). *Android Pemrograman Aplikasi Mobile Smatrphone dan Tablet PC Berbasis Android*. Bandung : Informatika Bandung.
- Kadir, Abdul.(2013). *From Zero to a pro javascript & jquery*.Yogyakarta : Andi
- Kendall, kenneth E. & Kendall, Jullie E. (2010). *System Analysis and Design Edisi 5*. Jakarta : Prenhallindo
- Laudon, Kenneth C., Laudon Jane P. (2012). *Management Information Systems – Managing the Digital Firm*. New Jersey : Pearson Education Inc.
- Lonnie D. Bentley., and Jefrey L. Whitten.(2007). *System Analysis and Design for The Global Enterprise seventh edition*. The McGraw-Hill Companies : USA.
- Pinedo, Michael.(2002). *Scheduling Theory, Algorithms, and System, Third Edition*. New York : Prentice Hall, Inc.

Purnama, Rangsang. (2010). *Mari Mengenal J2ME Java 2 Micro Edition*. Prestasi Pustaka, Jakarta.

Pressman, Roger S. (2010). *Software Engineering: a practioner's approach*. McGraw-Hill, New York.

Rainer, Kelly dan Cegieslky, Casey. (2011). *Introduction to Information System Enabling and Transforming Business*. New York : John Wiley & Sons

Rochaety, Eti, dkk. (2013), *Sistem Informasi Manajemen Edisi 2*, Jakarta:Mitra Wacana Media.

Schroeder, Roger. (2000). *Operations Management*. The Mc-Graw Hill Companies. Inc. North America

Shelly, B. Garry dan Harry J. Rosenblatt. (2012). *System Analysis and Design(9th Edition)*. Canada : Course Technology.

JURNAL :

Holla, Suhas dan Mahima M Katti.(2012). *Android Based Mobile Application Development and its Security*. International Journal of Computer Trends and Technology – Volume 3 Issue 3.

SITUS :

<https://kbbi.web.id/>

www.developer.android.com. Diakses pada 5 Oktober 2017