

**DEPARTEMEN ILMU KOMPUTER**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**INSTITUT PERTANIAN BOGOR**

**BOGOR**

**2018**

**SUTRISNO**

**SISTEM INFORMASI VERIFIKASI SOAL UJIAN DI PROGRAM STUDI S1 ILMU KOMPUTER IPB**

**PERNYATAAN MENGENAI SKRIPSI DAN  
SUMBER INFORMASI SERTA PELIMPAHAN HAK CIPTA**

Dengan ini saya menyatakan bahwa skripsi berjudul Sistem Informasi Verifikasi Soal Ujian di Program Studi S1 Ilmu Komputer IPB adalah benar karya saya dengan arahan dari komisi pembimbing dan belum diajukan dalam bentuk apa pun kepada perguruan tinggi mana pun. Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan maupun tidak diterbitkan dari penulis lain telah disebutkan dalam teks dan dicantumkan dalam Daftar Pustaka di bagian akhir skripsi ini.

Dengan ini saya melimpahkan hak cipta dari karya tulis saya kepada Institut Pertanian Bogor.

Bogor, Agustus 2018

*Sutrisno*

NIM G64140045

**ABSTRAK**

SUTRISNO. Sistem Informasi Verifikasi Soal Ujian di Program Studi S1 Ilmu Komputer IPB. Dibimbing oleh HUSNUL KHOTIMAH.

Ujian merupakan alat yang bertujuan untuk menilai prestasi akademik mahasiswa setelah melakukan proses belajar dalam jangka waktu tertentu. Soal ujian diberikan berdasarkan rencana perkuliahan yang tertuang dalam satuan acara pengajaran (SAP). Oleh karena itu, soal ujian perlu diverifikasi oleh tim verifikasi agar sesuai dengan *learning outcome* yang telah disusun. Saat ini, proses verifikasi soal ujian di Program Studi S1 Ilmu Komputer IPB dari tim pengajar ke verifikator maupun sebaliknya masih dilakukan secara manual melalui perantara Komisi Pendidikan S1 Ilmu Komputer IPB. Penelitian ini bertujuan membangun sistem informasi verifikasi soal ujian (SIVES) untuk membantu dosen dalam mengelola fail draf soal ujian. SIVES berhasil dibangun dengan menggunakan metode *prototyping* dan memiliki fungsi utama seperti fungsi mengunggah, memverifikasi, dan mengunduh draf soal ujian maupun mengunduh hasil verifikasi. SIVES dapat mengirimkan pemberitahuan berupa surel kepada verifikator jika tim pengajar telah mengunggah soal ujian. SIVES membuat proses verifikasi tidak membutuhkan perantara dalam pengiriman fail, sehingga verifikasi dapat dilakukan lebih efektif dan efisien.

Kata kunci: metode *prototyping*, sistem informasi, verifikasi soal ujian.

**ABSTRACT**

SUTRISNO. Exam Verification Information System in the IPB Computer Science Study Program. Supervised by HUSNUL KHOTIMAH

Exam is a tool that aims to assess student academic achievement after doing the learning process within a certain period. The exam questions are given based on the lecture plan contained in the teaching events unit (SAP). Therefore, exam questions need to be verified by the verification team to fit the learning outcomes that have been prepared. Currently, the verification process of the exam in the IPB Computer Science Study Program from the teaching team to the verifier and vice versa is still done manually through an intermediary of the Undergraduate Commission of Computer Science IPB. This study aims to build verification exam information system (SIVES) to help the lecturer in managing draft files about the exam. SIVES is successfully built using the prototyping method and has the main functions such as function of uploading, verifying, and downloading the draft of exam and download the verification result. SIVES may send email notification to the verifier if the team has uploaded the exam. SIVES makes the verification process requires no intermediaries in file submissions, so verification can be made more effectively and efficiently.

Keywords: information system, prototyping method, verification exam.

Skripsi  
sebagai salah satu syarat untuk memperoleh gelar  
Sarjana Komputer   
pada  
Departemen Ilmu Komputer

**SUTRISNO**

**SISTEM INFORMASI VERIFIKASI SOAL UJIAN DI PROGRAM STUDI S1 ILMU KOMPUTER IPB**

**DEPARTEMEN ILMU KOMPUTER**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**INSTITUT PERTANIAN BOGOR**

**BOGOR**

**2018**

Penguji:

1. Dr Imas Sukaesih Sitanggang, SSi MKom
2. Wulandari, SKomp MAgrSc

Judul Skripsi : Sistem Informasi Verifikasi Soal Ujian di Program Studi S1 Ilmu Komputer IPB

Nama : Sutrisno

NIM : G64140045

Disetujui oleh

Husnul Khotimah, SKomp MKom

Pembimbing

Diketahui oleh

Prof Dr Ir Agus Buono, MSi MKom

Ketua Departemen

Tanggal Lulus:

**PRAKATA**

Puji dan syukur penulis panjatkan kepada Allah subhanahu wa ta’ala atas segala karunia-Nya sehingga karya ilmiah ini berhasil diselesaikan. Tema yang dipilih dalam penelitian yang dilaksanakan sejak bulan Januari 2018 ini ialah sistem informasi, dengan judul Sistem Informasi Verifikasi Soal Ujian di Program Studi S1 Ilmu Komputer IPB. Penyelesaian karya ilmiah ini tidak terlepas dari bantuan berbagai pihak. Oleh Karena itu, penulis ingin menyampaikan terima kasih kepada:

1. Kedua orang tua yang selalu mendoakan dan memberi dukungan selama kuliah dan pembuatan karya ilmiah ini.
2. Ibu Husnul Khotimah, SKomp MKom selaku dosen pembimbing yang senantiasa memberikan arahan, motivasi, dan solusi dari permasalahan dalam karya ilmiah ini serta keikhlasan waktunya untuk memberikan bimbingan selama penelitian.
3. Ibu Dr Imas Sukaesih Sitanggang, SSi MKom dan Ibu Wulandari, SKomp MAgrSc selaku dosen penguji yang telah memberikan arahan, dan masukan dalam pengujian tugas akhir.
4. Bapak Irman Hermadi, SKom MSc PhD, Bapak Muhammad Abrar Istiadi, SKomp MKom dan Bapak Rivo Rizki Maulana, SKomp yang telah memberikan bantuan serta arahan dalam implementasi LDAP IPB dalam penelitian tugas akhir.
5. Bapak Irvan dan Bapak Ridwan selaku Staf Administrasi Akademik yang telah membantu memudahkan urusan administrasi selama penulis berkuliah di Departemen Ilmu Komputer IPB.
6. Teman-teman sebimbingan yaitu Fahrendi, Nur Hadi, dan Ayuni yang selalu mendukung dan saling memberi motivasi.
7. Teman-teman Lab SEIS yaitu Widia, Fadlan, Nadia, Chaca, Hakim, Laras, Yoga, Alifka, dan Bang Hendrik yang selalu memberi bantuan, motivasi, dan saran.
8. Teman-teman Program S1 Ilmu Komputer IPB angkatan 51.

Semoga segala bantuan, motivasi, dan dukungan yang telah diberikan kepada penulis senantiasa dibalas kebaikan oleh Allah subhanahu wa ta’ala. Semoga karya ilmiah ini memberikan manfaat bagi pembaca.

Bogor, Agustus 2018

*Sutrisno*

**DAFTAR ISI**

DAFTAR TABEL vi

DAFTAR GAMBAR vi

DAFTAR LAMPIRAN vi

[PENDAHULUAN 1](#_Toc520851724)

[Latar Belakang 1](#_Toc520851725)

[Perumusan Masalah 1](#_Toc520851726)

[Tujuan Penelitian 2](#_Toc520851727)

[Manfaat Penelitian 2](#_Toc520851728)

[Ruang Lingkup Penelitian 2](#_Toc520851729)

[TINJAUAN PUSTAKA 2](#_Toc520851730)

[Metode *Prototyping* 2](#_Toc520851731)

[*Framework* Laravel 5.5 3](#_Toc520851732)

[METODE 3](#_Toc520851733)

[Data Penelitian 3](#_Toc520851734)

[Tahapan Penelitian 3](#_Toc520851735)

[Lingkungan Pengembangan 5](#_Toc520851736)

[HASIL DAN PEMBAHASAN 5](#_Toc520851737)

[Iterasi I 5](#_Toc520851738)

[Iterasi II 14](#_Toc520851739)

[SIMPULAN DAN SARAN 19](#_Toc520851740)

[Simpulan 19](#_Toc520851741)

[Saran 19](#_Toc520851742)

[DAFTAR PUSTAKA 19](#_Toc520851743)

[LAMPIRAN 20](#_Toc520851744)

[RIWAYAT HIDUP 34](#_Toc520851745)

**DAFTAR TABEL**

1. [Karakteristik pengguna SIVES 7](#_Toc520850288)
2. [Kebutuhan fungsional SIVES iterasi satu 7](#_Toc520850289)
3. [Hasil pengujian SIVES iterasi satu 13](#_Toc520850290)
4. [Kebutuhan fungsional sistem iterasi kedua 14](#_Toc520850291)
5. [Hasil pengujian SIVES iterasi kedua 18](#_Toc520850292)

**DAFTAR GAMBAR**

1. [Tahapan Metode Prototyping (Pressman 2010) 3](#_Toc520850430)
2. [Tahapan Penelitian 4](#_Toc520850431)
3. [*Flow chart* proses bisnis SIVES 6](#_Toc520850432)
4. [*Use case* *diagram* SIVES iterasi satu 8](#_Toc520850433)
5. [*Activity diagram* memverifikasi soal ujian 9](#_Toc520850434)
6. [*Class diagram* SIVES iterasi satu 9](#_Toc520850435)
7. [Potongan kode program Model verifikasi 10](#_Toc520850436)
8. [Halaman tambah mata kuliah tipe aktor admin 11](#_Toc520850437)
9. [Halaman unggah soal ujian tipe aktor pengajar 11](#_Toc520850438)
10. [Halaman verifikasi soal ujian tipe aktor verifikator 12](#_Toc520850439)
11. [Halaman melihat hasil verifikasi soal ujian 12](#_Toc520850440)
12. [*Form* hasil verifikasi soal ujian 13](#_Toc520850441)
13. [Potongan kode program *controller* verifikasi 13](#_Toc520850442)
14. [*Use case diagram* SIVES iterasi kedua 15](#_Toc520850443)
15. [*Class diagram* SIVESiterasi kedua 16](#_Toc520850444)
16. [Halaman melihat direktori soal ujian. 17](#_Toc520850445)
17. [Halaman mengunggah hasil verifikasi 17](#_Toc520850446)

**DAFTAR LAMPIRAN**

1. [Berkas pendukung verifikasi soal ujian 20](#_Toc520851785)
2. [*Activity diagram* SIVES iterasi satu 23](#_Toc520851786)
3. [Tampilan halaman SIVES iterasi satu 27](#_Toc520851787)
4. [*Activity diagram* SIVES iterasi kedua 31](#_Toc520851788)
5. [Tampilan halaman SIVES iterasi kedua 33](#_Toc520851789)

# PENDAHULUAN

## Latar Belakang

Ujian merupakan salah satu alat untuk mengukur dan menilai hasil belajar mahasiswa. Pelaksanaan ujian bertujuan untuk menilai prestasi akademik mahasiswa setelah melakukan proses belajar dalam jangka waktu tertentu dan mengevaluasi keberhasilan program pengajaran (Djaali dan Muljono 2008). Tujuan ujian dapat tercapai jika soal ujian mengacu pada satuan acara pengajaran (SAP) yang disusun untuk masing-masing mata kuliah. SAP merupakan rencana kegiatan pembelajaran yang digunakan untuk setiap topik pembahasan yang memberi petunjuk secara rinci pertemuan demi pertemuan. SAP berisi tujuan, ruang lingkup, materi yang diajarkan, proses pembelajaran, media pembelajaran, dan evaluasi yang digunakan. Salah satu cara untuk memastikan soal ujian yang dibuat telah sesuai dengan SAP adalah dengan melakukan verifikasi.

Berdasarkan prosedur operasional baku (POB) No POB/KOM-PP/05 tentang persiapan soal ujian, verifikasi merupakan pengecekan soal ujian terhadap kesesuaian dengan *learning outcomes* (kompetensi dasar) mata kuliah dan kesesuaian dengan materi perkuliahan yang diberikan kepada mahasiswa. Verifikasi yang dilakukan bertujuan untuk pengendalian mutu soal ujian. Proses persiapan soal ujian dimulai dengan penyusunan draf soal ujian oleh tim pengajar. Draf soal ujian kemudian diverifikasi kesesuaiannya dengan SAP oleh tim verifikasi soal ujian (verifikator). Draf soal ujian yang telah diverifikasi dan dinyatakan sesuai akan diserahkan kepada Staf Administrasi Akademik untuk dicek ulang format penulisannya sebelum soal ujian diperbanyak dan disebarkan.

Sebelum sistem dikembangkan, teknis pengiriman draf soal ujian baik dari tim pengajar ke verifikator maupun sebaliknya dilakukan melalui perantara orang ketiga yaitu Komisi Pendidikan S1 Ilmu Komputer IPB. Tim pengajar mengirimkan draf soal ujian ke Komisi Pendidikan S1 Ilmu Komputer yang kemudian akan diteruskan kepada verifikator yang bersangkutan. Jika draf soal ujian tidak lolos verifikasi oleh verifikator, maka draf soal ujian diserahkan kembali kepada tim pengajar untuk dilakukan revisi. Pengiriman draf soal ujian dengan cara tersebut tidak efektif karena seringkali surel yang berisi draf soal ujian tertimbun oleh surel lain sehingga menghambat proses penerusan draf soal ujian dari tim pengajar ke verifikator maupun sebaliknya.

Sistem yang akan dibangun pada penelitian ini adalah Sistem Informasi Verifikasi Soal Ujian (SIVES). SIVES merupakan sistem berbasis web untuk membantu dosen Program Studi S1 Ilmu Komputer IPB dalam mengelola fail draf soal ujian. SIVES memiliki fungsi utama seperti fungsi mengunggah, memverifikasi, dan mengunduh draf soal ujian maupun mengunduh hasil verifikasi. SIVES dikembangkan dengan mengikuti kaidah metode *prototyping* dan PHP *framework* Laravel 5.5 sebagai bahasa pemrograman.

## Perumusan Masalah

Rumusan permasalahan dalam penelitian ini adalah proses verifikasi soal ujian di Program Studi S1 Ilmu Komputer IPB dari tim pengajar ke verifikator maupun sebaliknya masih dilakukan secara manual melalui perantara Komisi Pendidikan S1 Ilmu Komputer IPB. Oleh karena itu, dibutuhkan sistem informasi untuk menunjang kegiatan verifikasi soal ujian yang meliputi pengiriman, verifikasi, dan pengelolaanfail agar lebih efektif dan efisien*.*

## Tujuan Penelitian

Tujuan penelitian ini adalah mengembangkan sistem informasi verifikasi soal ujian berbasis web yang menunjang proses pengiriman, verifikasi, dan pengelolaan fail di Program Studi S1 Ilmu Komputer IPB.

## Manfaat Penelitian

Pengembangan sistem pada penelitian ini diharapkan dapat bermanfaat bagi Komisi Pendidikan dan Staf Administrasi Akademik Program Studi S1 Ilmu Komputer IPB dalam mendukung proses persiapan pelaksanaan ujian.

## Ruang Lingkup Penelitian

Ruang lingkup dari penelitian ini adalah:

1. Data yang digunakan berasal dari Departemen Ilmu Komputer IPB tahun ajaran 2017/2018.
2. Jenis soal ujian yang dapat diverifikasi adalah soal Ujian Tengah Semester (UTS), Ujian Akhir Semester (UAS), dan Ujian Praktikum.
3. Data mata kuliah yang digunakan adalah mata kuliah yang ada pada kurikulum S1 Ilmu Komputer IPB tahun 2014.

# TINJAUAN PUSTAKA

## Metode *Prototyping*

Metode *prototyping* merupakan metode pengembangan dan pengujian yang cepat dengan menggunakan model kerja (prototipe) aplikasi baru melalui proses interaksi berulang yang biasa digunakan oleh ahli sistem informasi dan ahli bisnis (O’Brien 2005). Metode *prototyping* membuat proses pengembangan sistem menjadi lebih cepat dan mudah, terutama bagi pengguna yang hanya mendefinisikan tujuan umum perangkat lunak namun tidak mengidentifikasi persyaratan rinci untuk fungsi dan fitur. Metode *prototyping* melibatkan *stackholder* dalam proses pembuatan sistem, sehingga sistem yang dikembangkan dievaluasi dan diperbaiki dalam beberapa iterasi sebelum akhirnya sistem dapat diterima.

Menurut Pressman (2010) metode *prototyping* mempunyai tahapan-tahapan yaitu *communication, quick plan, modeling quick design, construction of prototype,* dan *deployment, delivery, and feedback*. Tahapan metode *prototyping* dapat dilihat pada Gambar 1.

****

Gambar 1 Tahapan Metode Prototyping (Pressman 2010)

## *Framework* Laravel 5.5

Laravel merupakan kerangka kerja (*framework*) sistem web yang meminjam fitur terbaik dari kerangka poluler lainnya (Glimore 2015). Laravel mencoba untuk menghilangkan kesulitan saat pengembangan sistem dengan mengurangi tugas umum yang digunakan di sebagian besar proyek web. *Framework* Laravel didasarkan pada paradigma Model View Controller (MVC) (Bean 2015).

MVC memisahkan pengembangan sistem berdasarkan komponen utama yang membangun sebuah sistem, seperti bagian program yang menangani *query* dari *database* (*Model*), program yang mengatur tampilan dari respons pengguna(*View*), dan aliran kontrol antara tampilan dan model (*Controller*) (Chanchai 2011).

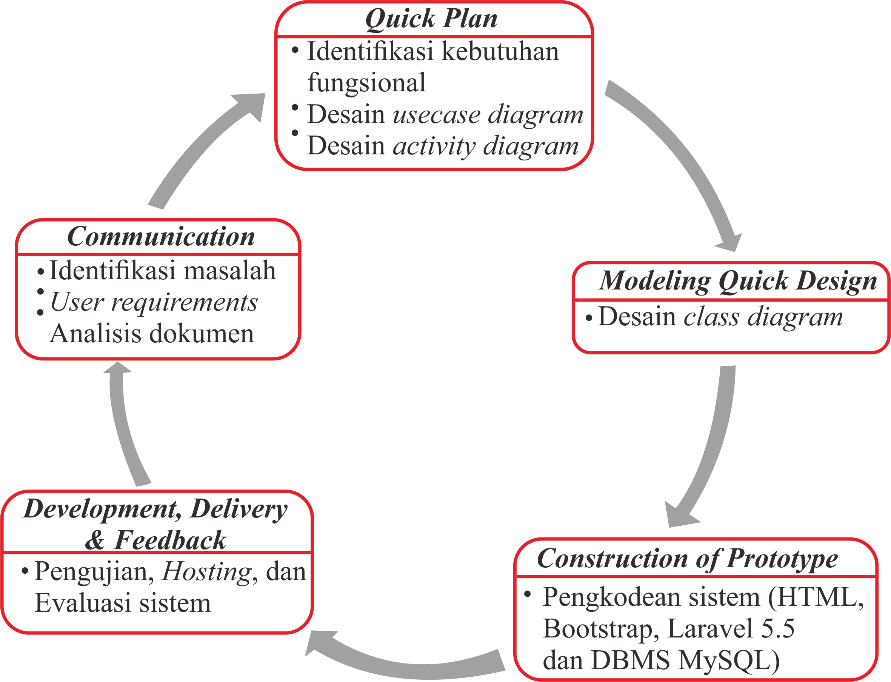
# METODE

## Data Penelitian

Data yang digunakan pada penelitian ini merupakan data yang berasal dari Program Studi S1 Ilmu Komputer IPB tahun ajaran 2017/2018. Data tersebut berupa data dosen, data mata kuliah yang ada pada kurikulum S1 Ilmu Komputer IPB tahun 2014, Prosedur Operasional Baku (POB) No POB/KOM-PP/05 Rev.00, serta contoh *form* verifikasi soal (POB/KOM-PP/FRM-01-00).

## Tahapan Penelitian

Tahapan penelitian sistem informasi verifikasi soal ujian mengikuti tahapan metode *prototyping*. Metode *prototyping* cocok diterapkan ketika *stakeholder* hanya memahami tujuan umum dari sistem yang akan dibangun karena memungkinkan pengembang dan *stakeholder* saling berinteraksi selama proses pembuatan sistem. Menurut Pressman (2010) metode *prorotyping* terdiri atas lima tahapan, yaitu *communication, quick plan*, *modeling quick design*, *construction of prototype* , dan tahapan *deployment, delivery, and feedback*. Adapun rincian setiap tahap yang dilakukan pada penelitian ini dapat dilihat pada Gambar 2.



Gambar 2 Tahapan Penelitian

***Communication***

Tahapan *communication* pada penelitian ini dilakukan dengan cara berkomunikasi langsung dengan *stakeholder* dan memahami berkas Prosedur Operasional Baku (POB) yang menjadi acuan dalam pembuatan sistem. *Stakeholder* pada penelitian ini adalah Komisi Pendidikan S1 Ilmu Komputer IPB. Komunikasi ini bertujuan untuk mengetahui masalah yang saat ini dihadapi dan mengetahui kebutuhaan sistem yang pengguna butuhkan. Keluaran dari tahapan ini berupa *user requirements* yang menjadi acuan dalam proses pengembangan sistem.

***Quick Plan***

Tahapan *quick plan* merupakan tahap menganalisis dan mendefinisikan kebutuhan fungsional dari sistem yang akan dibangun. Analisis ini mengacu pada *user requirements* yang yang telah didapatkan dari tahapan komunikasi dengan *stakeholder*. Hasil pendefinisian fungsi dapat direpresentasikan dengan *use case diagram* dan *activity diagram*. *Use case diagram* dibuat untuk menggambarkan *business rule* yang digunakan pada sistem. Sementara itu, detail alur bisnis digambarkan pada *activity diagram.*

***Modeling Quick Design***

Tahapan *modeling quick design* merupakan tahap perancangan desain sistem. Tahap ini dilakukan analisis *use case* untuk menentukan entitas yang akan digunakan pada basis data. Hubungan antar tabel pada basis datadigambarkan oleh *class diagram.*

***Construction of Prototype***

Tahapan *construction of prototype* merupakan tahap pembangunan sistem berdasarkan keluaran dari tahapan sebelumnya. Tahap ini dilakukan pengimplementasian *coding* dengan menggunakan HTML, Bootstrap, bahasa pemrograman PHP *framework* Laravel 5.5, dan DBMS MySQL.

***Deployment, Delivery, and Feedback***

Tahap *deployment, delivery, and feedback* terdiri atas tahap pengujian sistem, hosting sistem, dan evaluasi sistem oleh pengguna atau *stakeholder*. Pengujian sistem dilakukan secara *black box* oleh pengembang dan *stakeholder* guna mengidentifikasi apakah sistem yang dibuat sudah memenuhi *user requirements* yang diungkapkan pada tahap *communication*. Evaluasi diberikan sebagai umpan balik terhadap sistem yang telah dibangun sehingga dapat dilakukan proses penyempurnaan system.

## Lingkungan Pengembangan

Spesifikasi perangkat keras dan perangkat lunak yang digunakan pada penelitian ini yaitu sebagai berikut:

Perangkat keras:

1. Laptop Lenovo dengan *Processor* Intel Core i3
2. RAM sebesar 4 GB
3. 1 TB HDD
4. Sistem operasi Windows 10

Perangkat lunak:

1. PHP >= 7.0.0 sebagai bahasa pemrograman
2. *Framework* Laravel 5.5
3. DBMS MySQL
4. XAMPP sebagai server dan penerjemah bahasa pemrograman PHP
5. Sublime Text 3 (teks editor)
6. Google Chrome Version 62.0.3202.94 (*browser*)

# HASIL DAN PEMBAHASAN

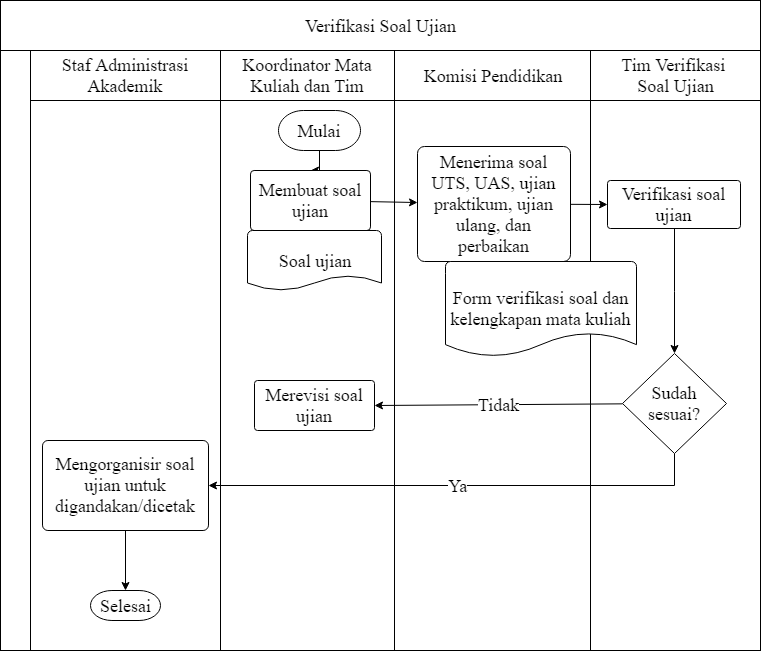
Proses analisis dan pengembangan SIVES berlangsung dari bulan Februari 2018. Sesuai dengan metode pengembangan yang telah direncanakan sebelumnya, SIVES dikembangkan dengan menggunakan metode *prototyping* yang meliputi *communication, quick plan, modeling quick design, construction of prototype,* dan *deployment, delivery, and feedback*. Tahapan metode *prototypiing* pada pengembangan SIVES dilakukan sebanyak dua kali iterasi.

## Iterasi I

### *Communication*

Tahapan *communication* merupakan tahap bertemunya pengembang dengan *stakeholder.* Pada tahapan ini pengembang melakukan diskusi bersama Ibu Husnul Khotimah, SKomp MKom selaku Komisi Pendidikan S1 Ilmu Komputer IPB yang berperan sebagai *stakeholder*. Komunikasi ini bertujuan untuk mengetahui masalah yang sedang dihadapi dan mengetahui kebutuhan sistem yang pengguna inginkan. Selain diskusi dengan *stakeholder*, pengembang juga melakukan analisis terhadap dokumen-dokumen terkait untuk memperjelas gambaran proses bisnis yang selama ini berjalan. Dokumen terkait yang pengembang analisis adalah dokumen Prosedur Operasional Baku (POB) Verifikasi Soal Ujian (POB/KOM-PP/05 Rev.00) dan *Form* Verifikasi Soal Ujian (POB/KOM-PP/05/FRM-01-00).

Berdasarkan hasil analisis dokumen, proses bisnis verifikasi soal ujian yang berlaku dan sedang berjalan di Departemen Ilmu Komputer IPB tertuang di dalam dokumen POB Verifikasi Soal Ujian (POB/KOM-PP/05 Rev.00). Rincian proses bisnis digambarkan dalam *flow chart* verifikasi soal ujian yang dapat dilihat pada Gambar 3.



Gambar 3 *Flow chart* proses bisnis SIVES

Koordinator mata kuliah bersama tim pengajar membuat soal ujian berupa soal UTS, UAS, ujian praktikum, ujian ulang, atau ujian perbaikan untuk diserahkan kepada Komisi Pendidikan S1 Ilmu Komputer IPB. Komisi Pendidikan S1 Ilmu Komputer IPB menerima soal ujian lalu mempersiapkan berkas berupa *form* verifikasi soal ujian, kontrak kuliah, SAP, dan silabus mata kuliah yang dapat dilihat pada Lampiran 1.

Tim verifikasi soal ujian (verifikator) memeriksa kesesuaian soal ujian dengan *learning outcomes* (kompetensi dasar) mata kuliah dan kesesuaian dengan materi perkuliahan yang diberikan kepada mahasiswa yang tertulis dalam SAP. Tim pengajar melakukan revisi soal ujian yang dinyatakan tidak sesuai dengan *learning outcomes* dan SAP berdasarkan hasil verifikasi. Soal yang sudah dinyatakan sesuai dengan *learning outcomes* dan SAP berdasarkan hasil verifikasi diserahkan kepada Staf Administrasi Akademik untuk dilakukan penggandaan atau dicetak dan dipersiapkan lembar jawaban soal sesuai kebutuhan.

### *Quick Plan*

Tahapan *quick plan* merupakan tahap mengidentifikasi karakteristik pengguna dan menganalisis kebutuhan fungsional dari sistem yang akan dikembangkan berdasarkan komunikasi yang telah dilakukan sebelumnya. Analisis karakteristik pengguna dilakukan untuk mengetahui dan mendefinisikan siapa saja aktor yang akan menggunakan sistem. Hasil analisis karakteristik pengguna dapat dilihat pada Tabel 1.

Tabel 1 Karakteristik pengguna SIVES

|  |  |  |
| --- | --- | --- |
| **Aktor** | **Definisi** | **Hak Akses** |
| Admin | Komisi Pendidikan dan Staf Administrasi Akademik S1 Ilmu Komputer IPB | Memiliki hak akses untuk mengelola keseluruhan sistem |
| Tim Pengajar | Dosen yang berdasarkan persyaratan pendidikan dan keahliannya ditugaskan untuk mengajar suatu mata kuliah | Memiliki hak akses untuk mengunggah, mengunduh, dan melihat hasil verifikasi soal ujian |
| Verifikator | Dosen yang ditetapkan oleh Ketua Departemen untuk melaksanakan verifikasi soal ujian yang terdiri atas Kepala Divisi atau dosen yang mewakili, pengelola program S1 Ilmu Komputer, dan Komisi Pendidikan S1 Ilmu Komputer. | Memiliki hak akses untuk mengunggah, mengunduh, memverifikasi, dan melihat hasil verifikasi soal ujian |

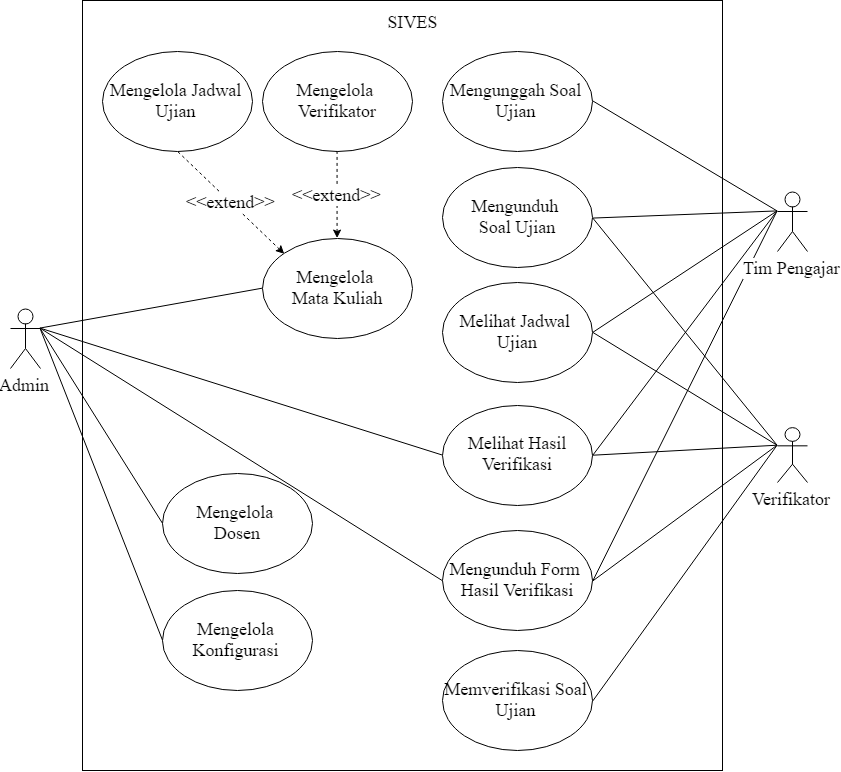
Kebutuhan fungsional merupakan aktifitas atau fungsi yang dapat dilakukan pada sistem yang akan dikembangkan. Kebutuhan fungsional dianalisis berdasarkan hasil komunikasi yang telah didapatkan sebelumnya. Adapun hasil analisis kebutuhan fungsional dapat dilihat pada Tabel 2.

Tabel 2 Kebutuhan fungsional SIVES iterasi satu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Kode Fungsi** | **Nama Fungsi** | **Aktor** | | |
| **Admin** | **Pengajar** | **Verifikator** |
| SIVES-F01 | *Login* | √ | √ | √ |
| SIVES-F02 | *Logout* | √ | √ | √ |
| SIVES-F03 | CRUD Dosen | √ | - | - |
| SIVES-F04 | CRUD Mata Kuliah | √ | - | - |
| SIVES-F05 | CRU Verifikator | √ | - | - |
| SIVES-F06-1 | Verifikasi Soal Ujian | - | - | √ |
| SIVES-F06-2 | Melihat Hasil Verifikasi | √ | √ | √ |
| SIVES-F07 | CRUD Jadwal Ujian | √ | - | - |
| SIVES-F08 | CRU Konfigurasi | √ | - | - |
| SIVES-F09 | Unggah Soal Ujian | - | √ | √ |
| SIVES-F10 | Unduh Soal Ujian | √ | √ | √ |
| SIVES-F11 | Unduh *Form* Hasil Verifikasi | √ | √ | √ |

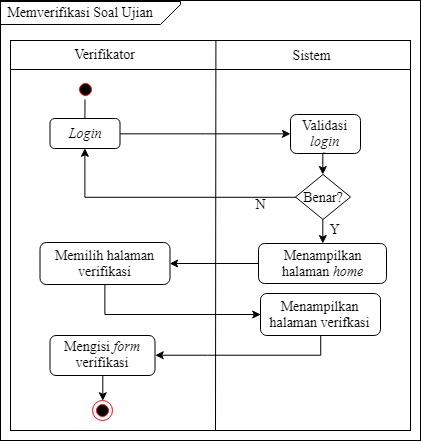
Ket : \*CRUD : *Create, Read, Update, Delete*

Berdasarkan hasil komunikasi dan pendefinisian fungsi, didapatkan *use case diagram* dan *activity diagram*. *Use case diagram* dibuat untuk menggambarkan *business rule* yang digunakan pada sistem. Sementara itu, detail alur bisnis digambarkan pada *activity diagram.* Terdapat 3 aktor dan 11 aktivitas utama pada *use case* yang dihasilkan. Admin memiliki aktivitas mengelola mata kuliah, mengelola verifikator, mengelola dosen, mengelola jadwal ujian, mengelola konfigurasi, melihat hasil verifikasi, mengunduh soal ujian dan mengunduh *form* hasil verifikasi. Tim pengajar memiliki aktivitas melihat jadwal ujian, mengunggah soal ujian, mengunduh soal ujian dan melihat hasil verifikasi. Verifikator memiliki aktivitas memverifikasi soal ujian dan melihat hasil verifikasi. *Use case* *diagram* SIVES iterasi satu dapat dilihat pada Gambar 4.



Gambar 4 *Use case* *diagram* SIVES iterasi satu

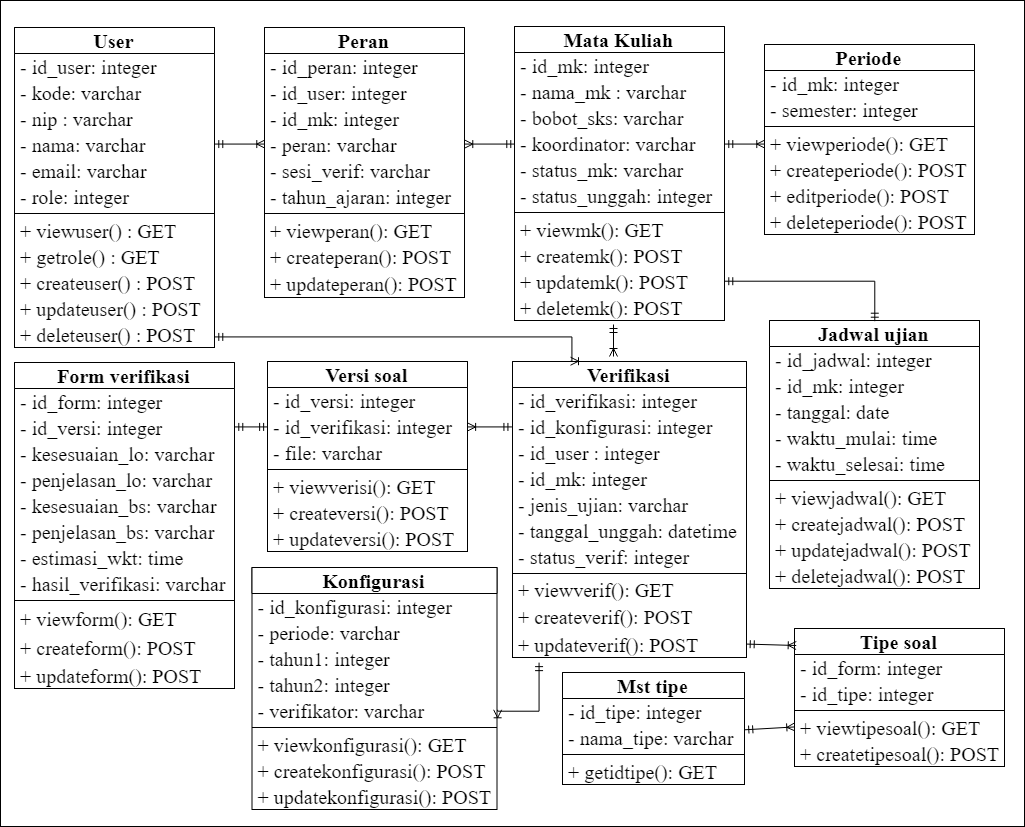
Setelah membuat *use case diagram* kemudian dilakukan pembuatan *activity diagram*. *Activity diagram* merupakan representasi alur bisnis dan penjabaran dari satu fungsi *use case*. Gambar 5 merupakan salah satu contoh *activity diagram* fungsi memverifikasi soal ujian yang dilakukan oleh verifikator. Verifikasi soal ujian diawali dengan verifikator melakukan proses *login*, kemudian sistem akan menampilkan halaman *home* lalu verifikator memilih halaman verifikasi. Kemudian verifikator diminta untuk mengisi *form* verifikasi soal ujian berdasarkan kesesuaian soal ujian dengan *learning outcomes* (kompetensi dasar) mata kuliah dan kesesuaian dengan materi perkuliahan yang diberikan kepada mahasiswa. *Activity diagram* SIVES iterasi satu lainnya dapat dilihat pada Lampiran 2.



Gambar 5 *Activity diagram* memverifikasi soal ujian

### *Modeling Quick Design*

Tahapan *modeling quick design* dilakukan perancangan basis data yang mengacu pada *use case* yang telah dibuat. Analisis *use case* dilakukan guna menentukan entitas apa saja yang akan dimasukan pada basis data*.* Hubungan antar tabel pada basis datadigambarkan oleh *class diagram* seperti pada Gambar 6.



Gambar 6 *Class diagram* SIVES iterasi satu

*Class diagram* SIVES iterasi satu pada Gambar 6 menunjukkan hubungan antar kelas dan *method* atau fungsi yang ada. Pada kelas *user* terdapat hubungan *many to many* dengan kelas mata kuliah, sehingga dibuat satu kelas baru yang diberi nama kelas peran. Kelas *user* memiliki hubungan *one to many* dengan kelas peran, artinya satu *user* dapat memiliki lebih dari satu peran, baik sebagai pengajar maupun sebagai verifikator. Kelas mata kuliah memiliki hubungan *one to many* dengan kelas peran, artinya satu mata kuliah dapat diajar atau diverifikatori lebih dari satu *user*. Pada kelas verifikasi terdapat *method viewverif*( ), *createverif*( ), dan *updateverif*( ) yang digunakan untuk melihat, menambah, dan mengubah data verifikasi*.* *Class diagram* pada Gambar 6 akan digunakan sebagai acuan dalam membuat *model* di MVC *framework* Laravel 5.5.

### *Construction of Prototype*

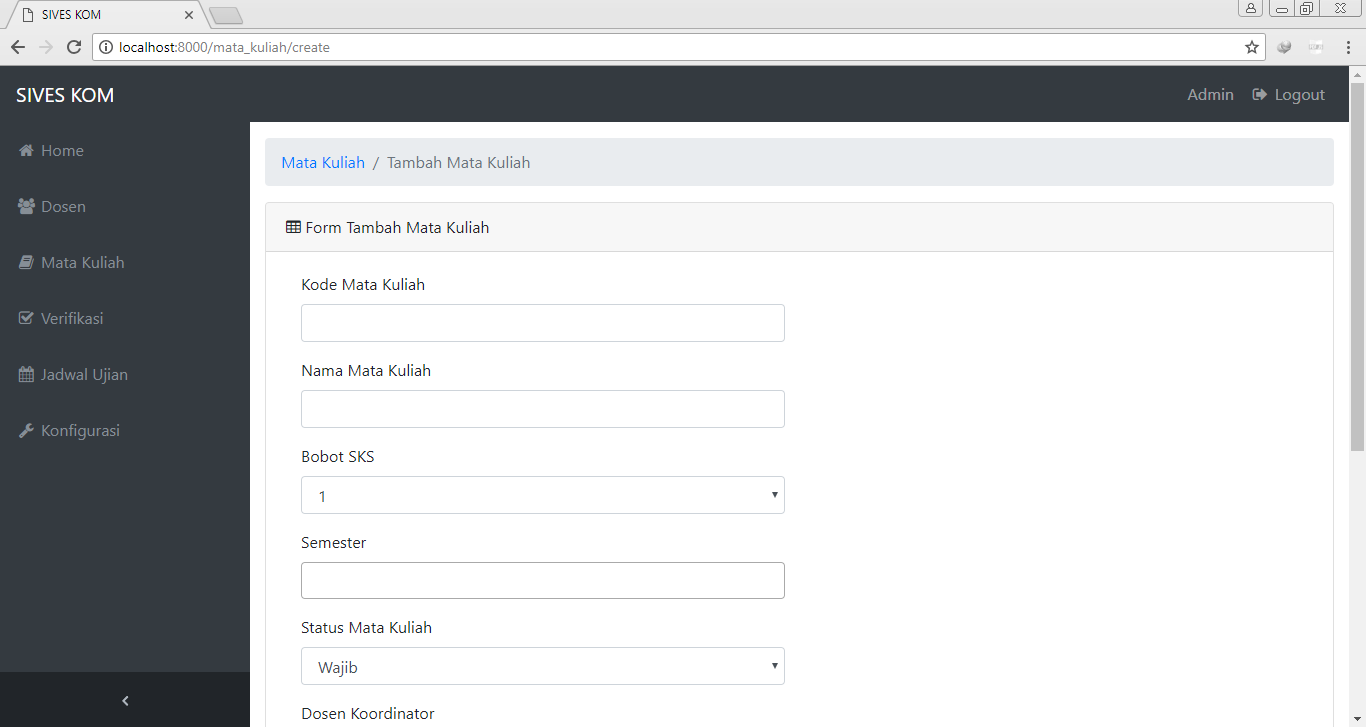
*Construction of prototype* merupakan tahap dimana pengkodean sistem dilakukan. Pengkodean sistem dibuat dengan menggunakan bahasa pemrograman web PHP *framework* Laravel 5.5dan DBMSMySQL dengan menggunakan *templete Bootstrap* sebagai acuan tampilan. Paradigma MVC pada laravel diimplementasikan dalam pengkodean sistem. Representasi basis data dilakukan pada bagian *model*, pengkodean tampilan sistem dilakukan pada bagian *view*, dan pengkodean fungsi dilakukan pada bagian *controller*. Berikut merupakan salah satu hasil implementasi *model* verifikasi pada *framework* Laravel 5.5 yang dapat dilihat pada Gambar 7.

|  |
| --- |
| 7 class Verifikasi extends Model  8 {  9 /\*\*  10 \* Indicates if the model should be timestamped.  11 \*  12 \* @var bool  13 \*/  14 protected $table = 'verifikasi';  15 public $timestamps = false;  16 protected $fillable = ['id\_verifikasi', 'id\_konfigurasi','verifikator','id\_mk','jenis\_ujian','status\_verif'];  17 protected $primaryKey = 'id\_verifikasi'; |

Gambar 7 Potongan kode program Model verifikasi

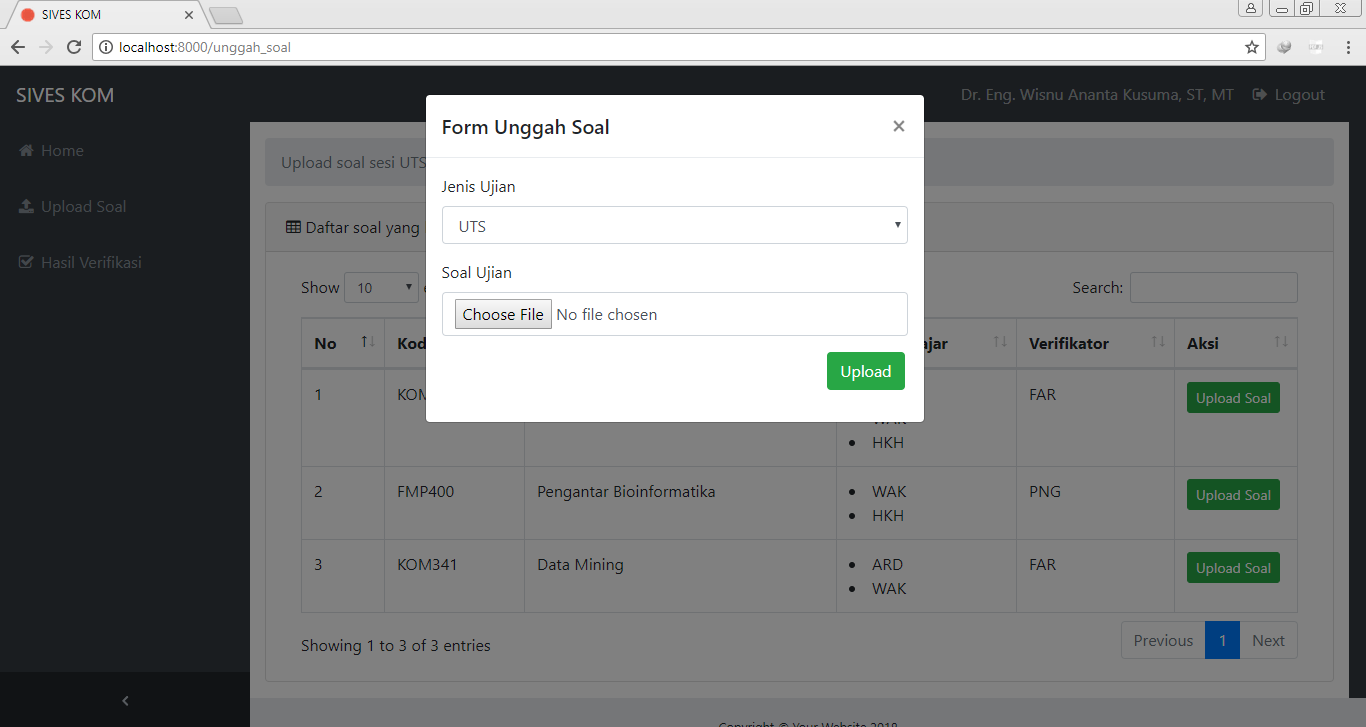
*Model* verifikasi terdiri atas beberapa atribut seperti: id\_verifikasi, id\_konfigurasi, id\_user, verifikator, id\_mk, jenis\_ujian, dan status\_verif. Atribut pada *model* verifikasi dikodekan pada bagian ‘protected $fillable’ yang berarti atribut tersebut dapat diakses oleh kelasitu sendiri. Protected $table digunakan untuk menandai tabel yang peneliti gunakan pada basis data dengan *primary key* ‘id\_verifikasi’. *Model* tersebut dapat digunakan untuk memanipulasi basis data dan membuat hubungan antar tabel.

Tampilan sistem dikodekan pada bagian *view*, tampilan tersebut dibedakan berdasarkan kategori dan hak akses masing-masing aktor. Salah satu contoh hasil pengkodean *view* dapat dilihat pada Gambar 8 yang merupakan tampilan halaman tambah mata kuliah aktor admin. Selain menambahkan mata kuliah, admin juga dapat mengubah dan menghapus mata kuliah.



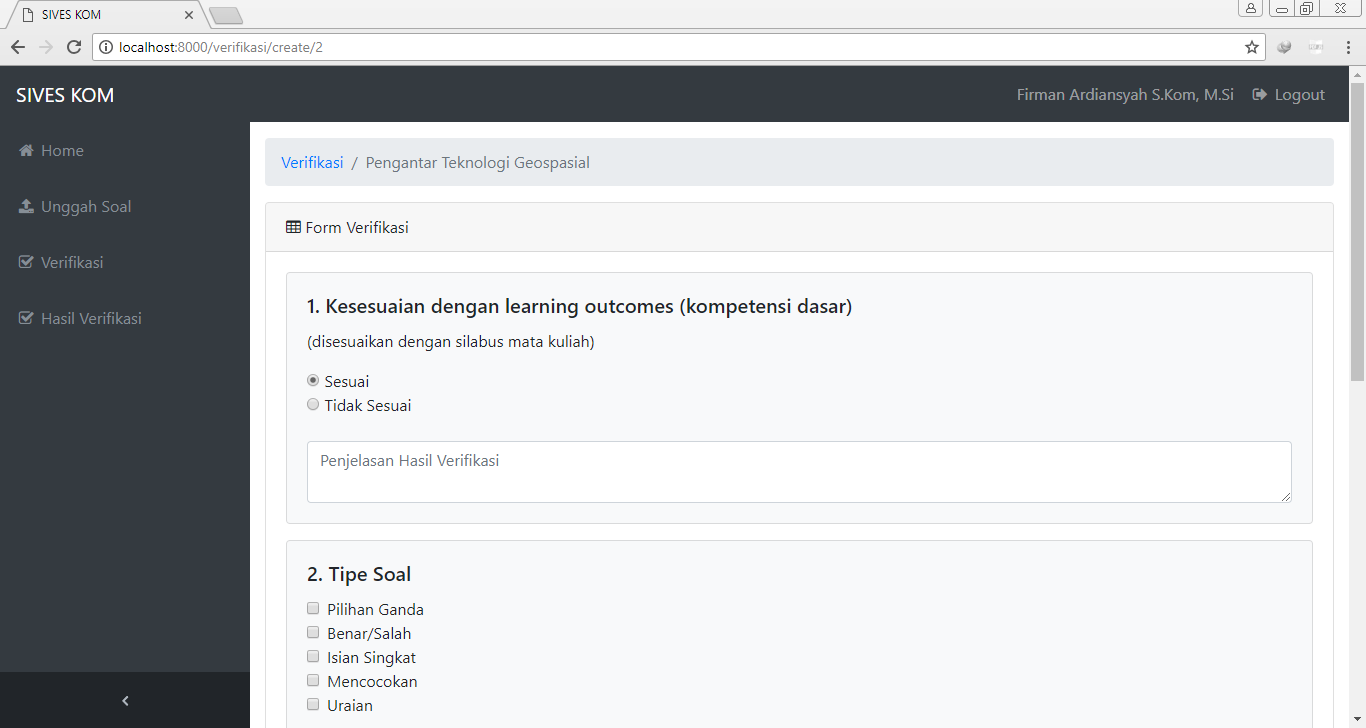
Gambar 8 Halaman tambah mata kuliah tipe aktor admin

Gambar 9 merupakan halaman unggah soal ujian untuk aktorpengajar. Tim pengajar bisa melihat daftar mata kuliah yang dikelolanya dan mengunggah soal ujian untuk mata kuliah tersebut. Surel pemberitahuan bahwa pengajar telah mengunggah soal ujian akan otomatis terkirim kepada verifikator yang bersangkutan.



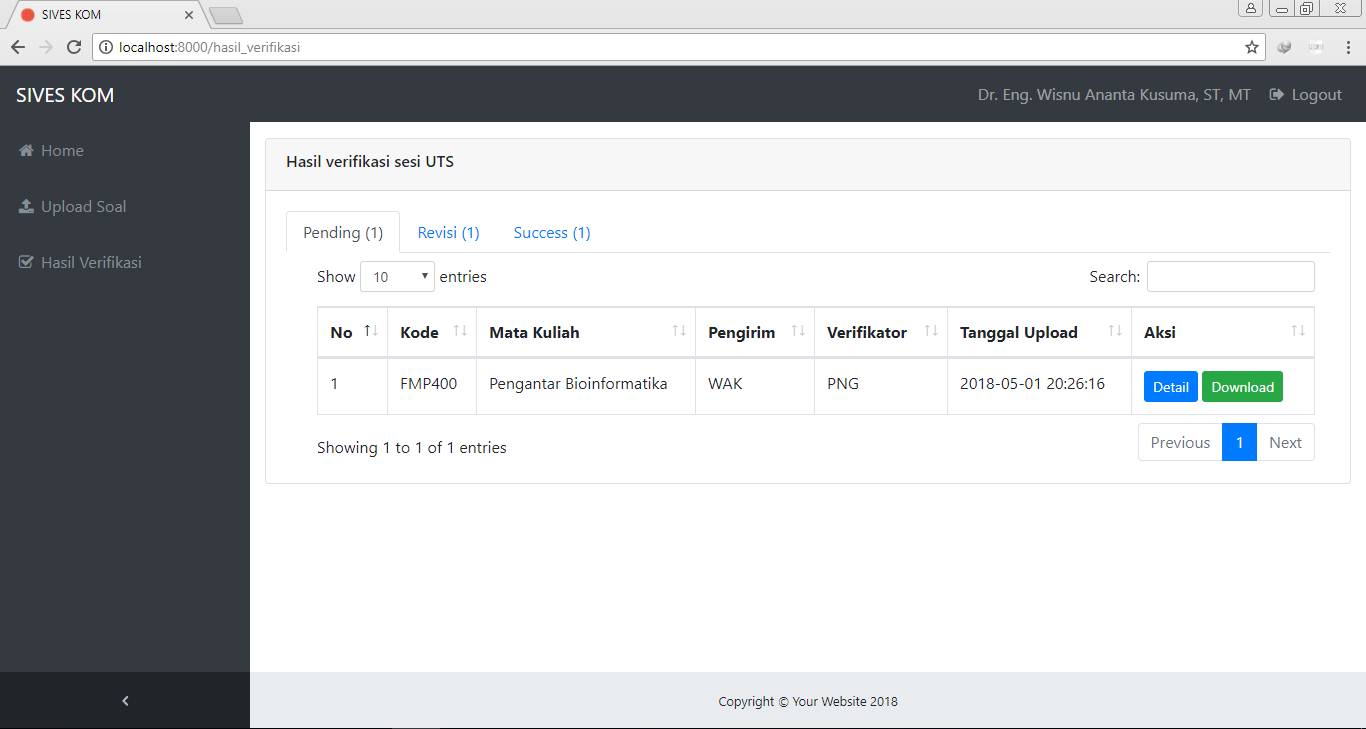
Gambar 9 Halaman unggah soal ujian tipe aktor pengajar

Gambar 10 merupakan halaman *form* verifikasi soal ujian yang ada pada aktorverifikator. Verifikator akan mengisikan *form* verifikasi berdasarkan kesesuaian soal ujian dengan *learning outcomes* (kompetensi dasar) mata kuliah dan kesesuaian dengan materi perkuliahan yang diberikan kepada mahasiswa. Soal ujian yang dinyatakan revisi oleh verifikator akan dikirimkan kembali kepada tim pengajar, sedangkan soal ujian yang dinyatakan diterima akan diteruskan kepada Staf Administrasi Akademik untuk dicek ulang format penulisannya kemudian diperbanyak.



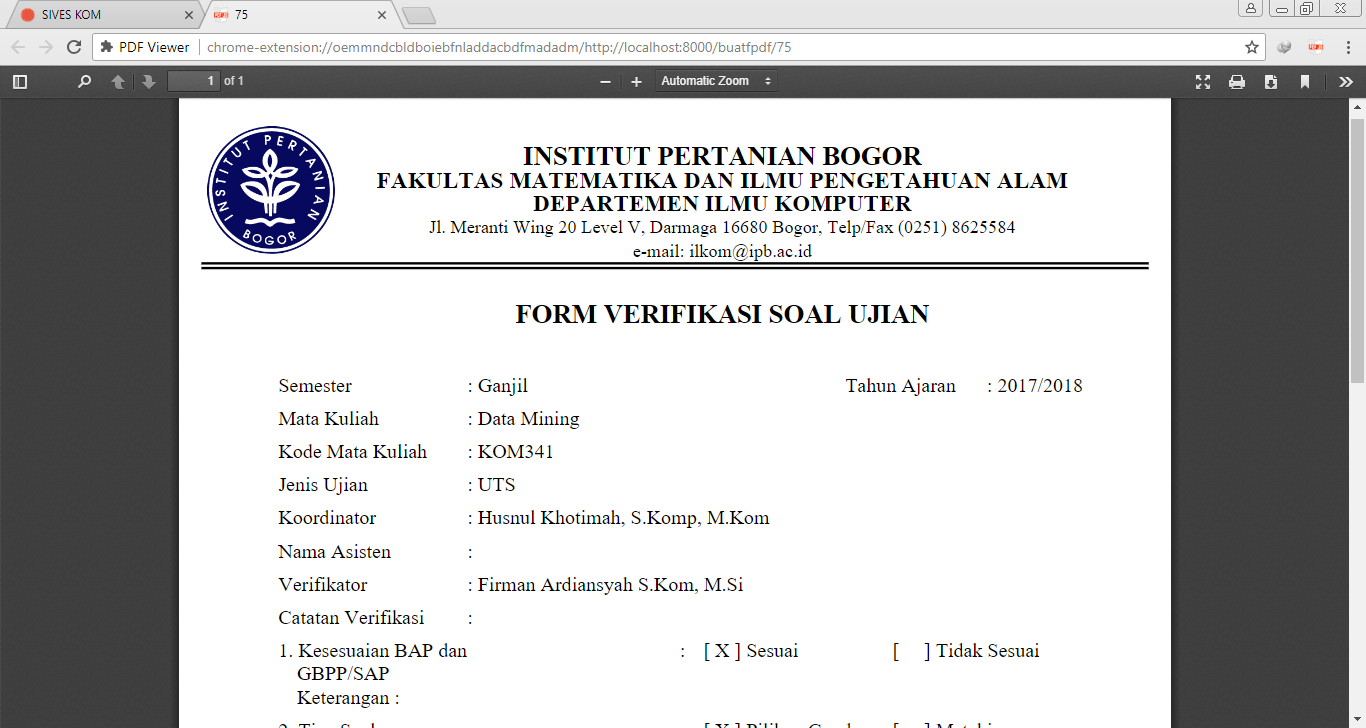
Gambar 10 Halaman verifikasi soal ujian tipe aktor verifikator

Gambar 11 merupakan halaman melihat hasil verifikasi soal ujian. Hasil verifikasi dikelompokan menjadi 3, yaitu *pending,* revisi, dan sukses*.* *Pending* merupakan status ketika soal ujian telah diunggah oleh tim pengajar namun belum diverifikasi oleh verifikator. Revisi merupakan status ketika soal ujian yang telah diunggah dinyatakan tidak sesuai dan harus diperbaiki, sedangkan sukses merupakan status ketika soal ujian yang telah diunggah dinyatakan sesuai dan diterima untuk diperbanyak.



Gambar 11 Halaman melihat hasil verifikasi soal ujian

Gambar 12 merupakan *form* hasil verifikasi soal ujian. Setiap soal ujian yang diunggah memiliki hasil verifikasinya masing-masing. *Form* hasil verifikasi akan tercetak secara langsung berdasarkan data hasil verifikasi yang dimasukan oleh verifikator*.* Halaman hasil pengkodean iterasi satu lainnya dapat dilihat pada Lampiran 3.



Gambar 12 *Form* hasil verifikasi soal ujian

Setelah dilakukan pengkodean *view*, proses selanjutnya adalah pengkodean bagian *controller*. *Controller* merupakan bagian yang menghubungkan *view* dan *model*. *Controller* berfungsiuntuk memproses fungsi atau perintah dari *user* kemudian menentukan bagaimana aplikasi dijalankan. Fungsi-fungsi yang terdapat pada *Controller* terdiri atas fungsi dasar seperti *create*( )*, read*( )*, update*( )*,* dan *delete*( )(CRUD). Gambar 13 merupakan potongan kode program pada *controller* verifikasi.

|  |
| --- |
| 52 public function create($id\_mk)  53 {  54 $mata\_kuliah = MataKuliah::where('id\_mk',$id\_mk)  ->first();  55 return view('pengajar/form\_verifikasi', compact('mata\_kuliah'));  56 } |

Gambar 13 Potongan kode program *controller* verifikasi

### *Deployment, Delivery, and Feedback*

SIVES yang telah dikembangkan dalam tahap *construction of prototype* kemudian dilakukan pengujiandan dievaluasi fungsinya. Pengujiandan evaluasi dilakukan dengan cara *black box*. Hasil pengujiandan evaluasi SIVES iterasi satu dapat dilihat pada Tabel 3.

Tabel 3 Hasil pengujian SIVES iterasi satu

| Kode Fungsi | Nama Fungsi | Hasil yang diharapkan | Hasil uji |
| --- | --- | --- | --- |
| SIVES-F01 | *Login* | *User* dapat melakukan *login* dan masuk ke halaman *home* sesuai dengan tipe *user* nya masing-masing. | Berhasil |
| SIVES-F02 | *Logout* | *User* dapat melakukan proses *logout* | Berhasil |
| SIVES-F03 | CRUD Dosen | *User* dapat mengelola data dosen | Berhasil |
| SIVES-F04 | CRUD Mata Kuliah | *User* dapat mengelola data mata kuliah | Berhasil |
| SIVES-F05 | CRU Verifikator | *User* dapat mengelola verifikator | Berhasil |
| SIVES-F06-1 | Verifikasi Soal Ujian | *User* dapat membuat verifikasi | Berhasil |
| SIVES-F06-2 | Melihat Hasil Verifikasi | *User* dapat melihat hasil verifikasi | Berhasil |
| SIVES-F07 | CRUD Jadwal Ujian | *User* dapat mengelola data jadwal ujian | Berhasil |
| SIVES-F08 | CRU Konfigurasi | *User* dapat mengelola konfigurasi | Berhasil |
| SIVES-F09 | Unggah Soal Ujian | *User* dapat mengunggah soal ujian | Berhasil |
| SIVES-F10 | Unduh Soal Ujian | *User* dapat mengunduh soal ujian | Berhasil |
| SIVES-F11 | Unduh *Form* Hasil Verifikasi | *User* dapat mengunduh *form* verifikasi | Berhasil |

Ket : \*CRUD : *Create, Read, Update, Delete*

Tabel 3 (*lanjutan*)

Berdasarkan hasil pengujian pada Tabel 3, semua kebutuhan fungsi yang didefinisikan pada tahapan komunikasi telah berhasil dibangun. Evaluasi sistem didapatkan dari *stakeholder* yang

## Iterasi II

### *Communication*

Tahapan *communication* pada *prototyping* iterasi kedua dilakukan bersama *stakeholder*. Hasil dari komunikasi ini didapatkan *requirements* berupa penambahan enam fungsi baru dan perubahan satu fungsi yang sudah ada. Fungsi yang ditambahkan berupa fungsi melihat direktori soal ujian, mengunggah berkas hasil verifikasi, mengunggah kontrak kuliah, mengunggah berita acara pengajaran (BAP), mengunduh kontrak kuliah, dan mengunduh BAP. Perubahan fungsi yang dilakukan yaitu perubahan fungsi *login* yang diintegrasikan dengan LDAP IPB. Tabel 4 menunjukkan seluruh kebutuhan fungsional sistem sampai prototipeiterasi kedua.

Tabel 4 Kebutuhan fungsional sistem iterasi kedua

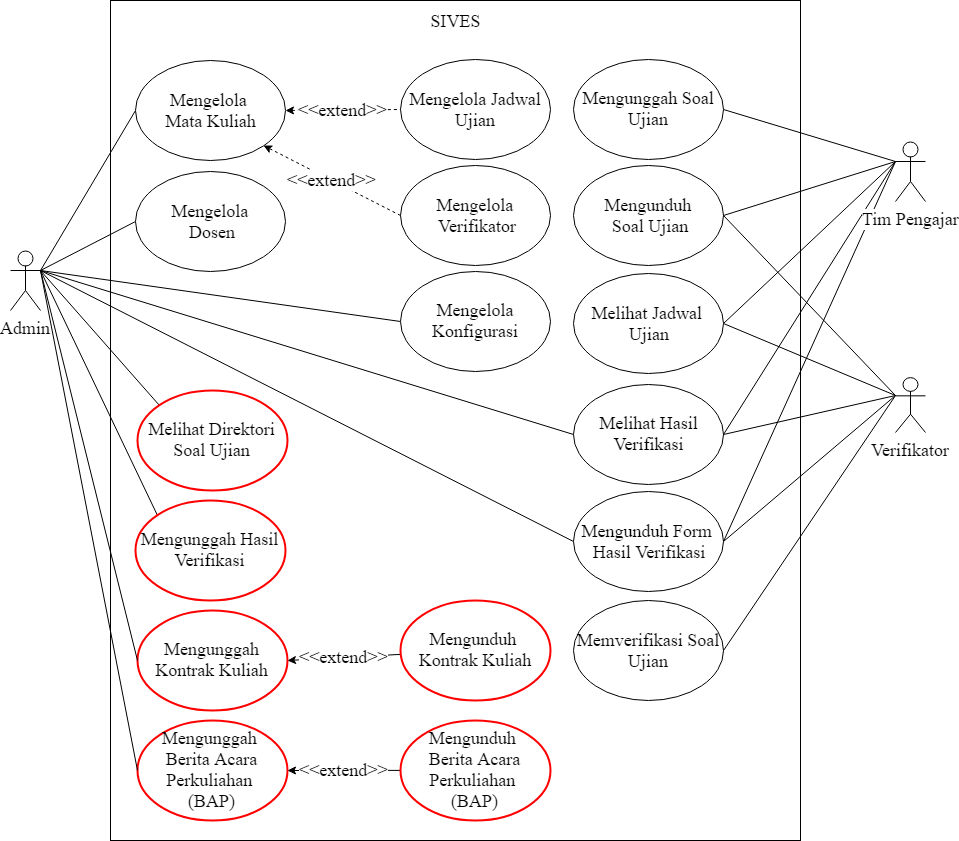
| Kode Fungsi | Nama Fungsi | Aktor | | |
| --- | --- | --- | --- | --- |
| Admin | Pengajar | Verifikator |
| SIVES-F01 | *Login* | √ | √ | √ |
| SIVES-F02 | *Logout* | √ | √ | √ |
| SIVES-F03 | CRUD Dosen | √ | - | - |
| SIVES-F04 | CRUD Mata Kuliah | √ | - | - |
| SIVES-F05 | CRU Verifikator | √ | - | - |
| SIVES-F06-1 | Verifikasi Soal Ujian | - | - | √ |
| SIVES-F06-2 | Melihat Hasil Verifikasi | √ | √ | √ |
| SIVES-F07 | CRUD Jadwal Ujian | √ | - | - |
| SIVES-F08 | CRU Konfigurasi | √ | - | - |
| SIVES-F09 | Unggah Soal Ujian | - | √ | √ |
| SIVES-F10 | Unduh Soal Ujian | √ | √ | √ |
| SIVES-F11 | Unduh *Form* Hasil Verifikasi | √ | √ | √ |
| SIVES-F12 | Melihat Direktori Soal Ujian | √ | - | - |
| SIVES-F13 | Mengunggah Hasil Verifikasi | √ | - | - |
| SIVES-F14 | Mengunggah Kontrak Kuliah | √ | - | - |
| SIVES-F15 | Mengunggah Berita Acara Perkuliahan (BAP) | √ | - | - |
| SIVES-F16 | Mengunduh Kontrak Kuliah | √ | - | √ |
| SIVES-F17 | Mengunduh BAP | √ | - | √ |

Ket : \*CRUD : *Create, Read, Update, Delete*

Tabel 4 (*lanjutan*)

### *Quick Plan*

Tahapan *quick plan* pada *prototyping* iterasi kedua dilakukan penambahan *use case* sesuai dengan kebutuhan sistem terbaru. Gambar 15 merupakan *use case diagram* iterasi kedua. *Activity diagram* dari fungsi *use case* iterasi kedua dapat dilihat pada Lampiran 4.



Gambar 14 *Use case diagram* SIVES iterasi kedua

Berdasarkan evaluasi pada iterasi satu, penambahan fungsi dilakukan pada aktor admin. Fungsi tersebut berupa fungsimelihat direktori soal ujian, mengunggah hasil verifikasi, mengunggah kontrak kuliah, mengunggah berita acara pengajaran (BAP), mengunduh kontrak kuliah, dan mengunduh BAP yang akan dibangun pada tahapan *construction of prototype*.

### *Modeling Quick Design*

Tahapan *modeling quick design* pada *prototyping* iterasi kedua dilakukan penambahan *class* yaitu *class* berkas dan *class* tipe berkas. Kedua *class* tersebut berfungsi untuk menyimpan berkas hasil verifikasi, kontrak kuliah, dan BAP yang diunggah oleh admin. Gambar 16 merupakan *class diagram* SIVES iterasi kedua.

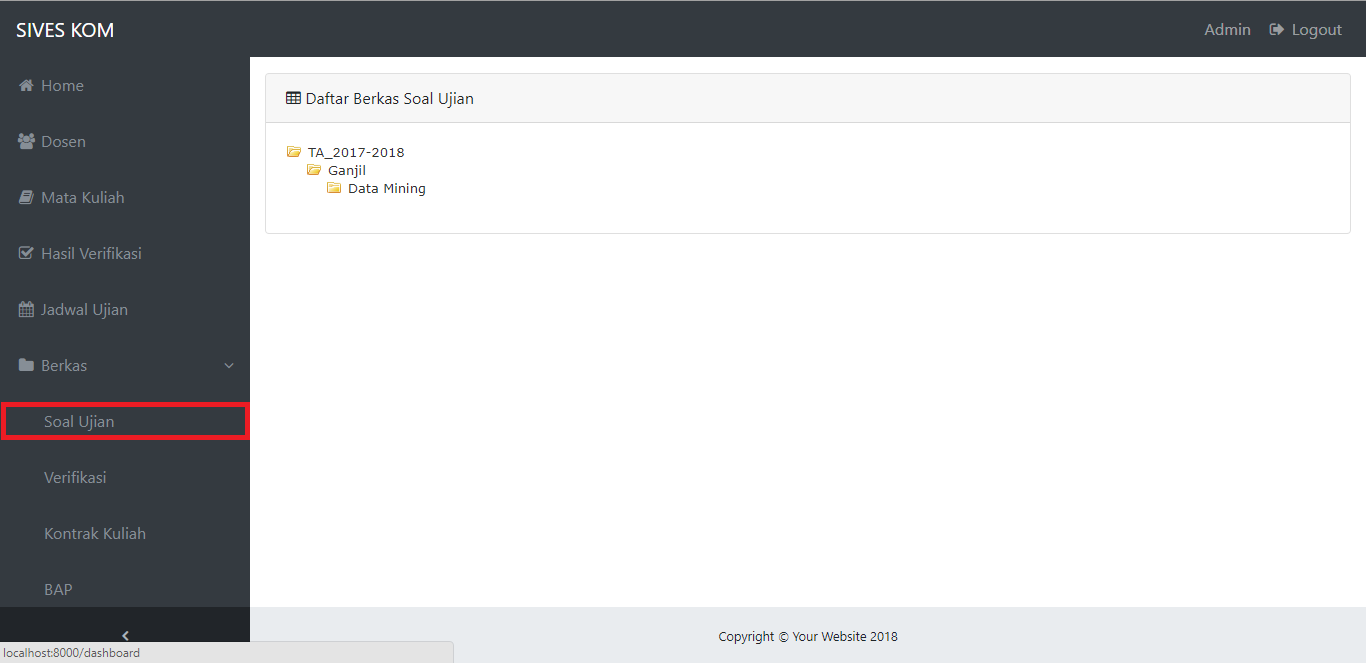


Gambar 15 *Class diagram* SIVESiterasi kedua

*Class* mata kuliah memiliki hubungan *one to many* dengan *class* berkas, yang artinya satu mata kuliah bisa memiliki banyak berkas. *Class* tipe berkas memiliki hubungan *one to many* dengan *class* berkas, yang artinya satu tipe berkas bisa dimiliki oleh banyak berkas.

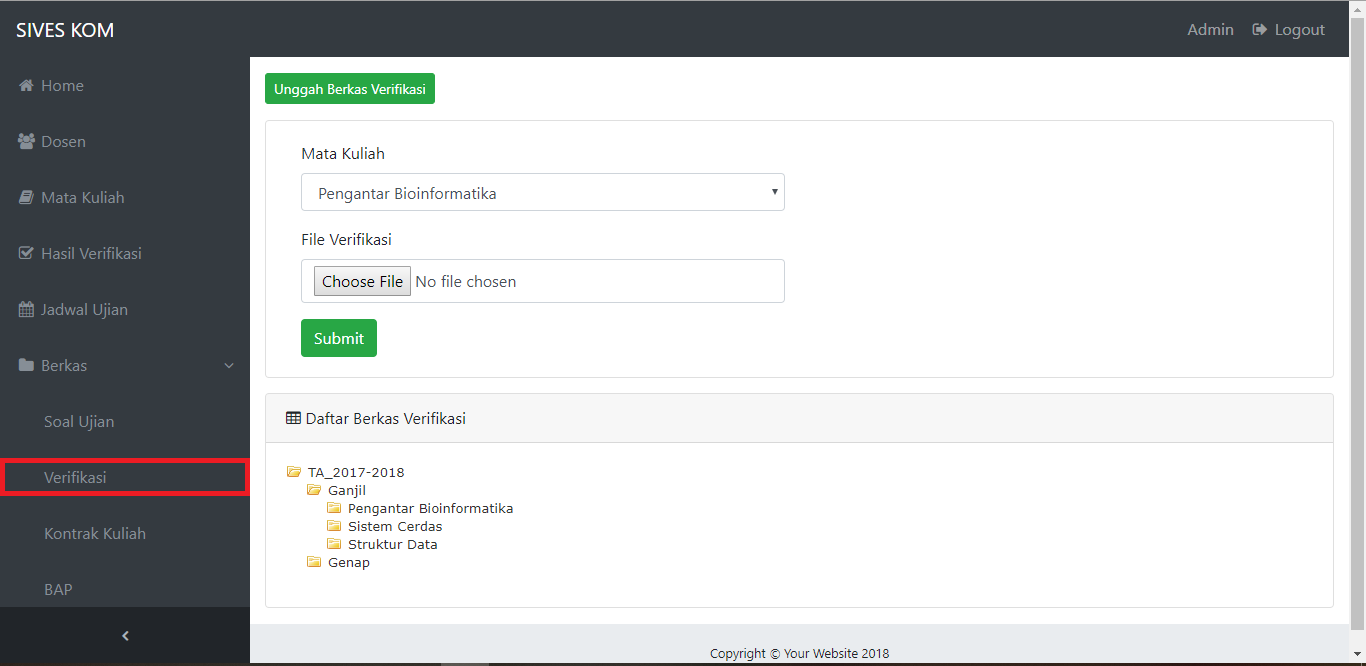
### *Construction of Prototype*

Tahapan *construction of prototype* iterasi kedua dilakukan pengkodean untuk fungsi baru pada sistem. Prototipe tersebut adalah halaman melihat direktori soal ujian, mengunggah berkas hasil verifikasi, mengunggah kontrak kuliah, mengunggah berita acara pengajaran (BAP), mengunduh kontrak kuliah, dan mengunduh BAP. Gambar 16 dan 17 merupakan tampilan halaman melihat direktori berkas soal ujian dan mengunggah hasil verifikasi, sedangkan tampilan halaman mengunggah kontrak kuliah dan menggunggah BAP dapat dilihat pada Lampiran 5.



Gambar 16 Halaman melihat direktori soal ujian.

Gambar 16 merupakan halaman melihat direktori soal ujian. Admin dapat mencari dan mengunduh soal ujian yang disimpan berdasarkan tahun ajaran, periode semester, dan mata kuliah. Penyimpanan berdasarkan folder bertujuan untuk memudahkan dalam pencarian soal ujian.



Gambar 17 Halaman mengunggah hasil verifikasi

Gambar 17 merupakan halaman mengunggah hasil verifikasi, admin harus memilih mata kuliah yang akan diunggah berkas hasil verifikasinya. Satu mata kuliah bisa mempunyai banyak hasil verifikasi.

### *Deployment, Delivery, and Feedback*

SIVES yang telah dikembangkan dalam tahap *construction of prototype* iterasi kedualalu dilakukan *testing* dan dievaluasi fungsinya. *Testing* dan evaluasi dilakukan dengan cara *black box*. Berdasarkan *testing* dan evaluasi didapatkan hasil yang dapat dilihat pada Tabel 5.

Tabel 5 Hasil pengujian SIVES iterasi kedua

|  |  |  |  |
| --- | --- | --- | --- |
| Kode Fungsi | Nama Fungsi | Hasil yang diharapkan | Hasil uji |
| SIVES-F01 | *Login* terintegrasi  LDAP IPB | *User* dapat melakukan *login* dan masuk ke halaman *home* sesuai dengan tipe *user* nya masing-masing. | Belum berhasil |
| SIVES-F02 | *Logout* | *User* dapat melakukan proses *logout* | Belum berhasil |
| SIVES-F03 | CRUD Dosen | *User* dapat mengelola data dosen | Berhasil |
| SIVES-F04 | CRUD Mata Kuliah | *User* dapat mengelola data mata kuliah | Berhasil |
| SIVES-F05 | CRU Verifikator | *User* dapat mengelola verifikator | Berhasil |
| SIVES-F06-1 | Verifikasi Soal Ujian | *User* dapat mengelola data verifikasi | Berhasil |
| SIVES-F06-2 | Melihat Hasil Verifikasi | *User* dapat mengelola data jadwal ujian | Berhasil |
| SIVES-F07 | CRUD Jadwal Ujian | *User* dapat mengelola konfigurasi | Berhasil |
| SIVES-F08 | CRU Konfigurasi | *User* dapat mengunggah soal ujian | Berhasil |
| SIVES-F09 | Unggah Soal Ujian | *User* dapat mengunduh soal ujian | Berhasil |
| SIVES-F10 | Unduh Soal Ujian | *User* dapat mengunduh *form* verifikasi | Berhasil |
| SIVES-F11 | Unduh *Form* Hasil Verifikasi | *User* dapat melihat direktori soal ujian | Berhasil |
| SIVES-F12 | Melihat Direktori Soal Ujian | *User* dapat melihat direktori hasil verifikasi | Berhasil |
| SIVES-F13 | Mengunggah Hasil Verifikasi | *User* dapat mengunggah hasil verifikasi | Berhasil |
| SIVES-F14 | Mengunggah Kontrak Kuliah | *User* dapat mengunggah kontrak kuliah | Berhasil |
| SIVES-F15 | Mengunggah Berita Acara Perkuliahan (BAP) | *User* dapat mengunggah BAP | Berhasil |
| SIVES-F16 | Mengunduh Kontrak Kuliah | *User* dapat mengunduh kontrak kuliah | Berhasil |
| SIVES-F17 | Mengunduh BAP | *User* dapat mengunduh BAP | Berhasil |

Ket : \*CRUD : *Create, Read, Update, Delete*

Fungsi *login* terintegrasi LDAP IPB sudah dicoba untuk diimplementasikan, namun terdapat beberapa kendala. Diantaranya, saat ini Direktorat Integrasi Data dan Sistem Informasi (DIDSI) IPB sedang melakukan perubahan akses data dari *lightweight directory access protocol* (LDAP) ke *central authentication service* (CAS). Semua aplikasi yang terhubung melalui LDAP akan dipindahkan menggunakan CAS, namun CAS masih dalam proses persiapan dan belum dapat di implementasikan.

# SIMPULAN DAN SARAN

## Simpulan

Pengembangan sistem informasi verifikasi soal ujian (SIVES) telah berhasil dilakukan dengan pendekatan metode *prototyping* sebanyak dua iterasi. Fungsi utama pada SIVES seperti fungsi mengunggah, memverifikasi, dan mengunduh soal ujian maupun mengunduh hasil verifikasi telah berhasil dikembangkan. Fungsi *login* terintegrasi LDAP IPB sudah dicoba untuk diimplementasikan namun masih belum berhasil karena adanya proses perubahan akses data di DIDSI IPB. SIVES diharapkan dapat mengotomatisasi proses verifikasi soal ujian sehingga dapat dilakukan secara efektif dan efisien.

## Saran

Soal ujian merupakan berkas yang bersifat penting dan rahasia, oleh karena itu sistem harus memiliki keamanan yang baik. Penelitian selanjutnya diharapkan dapat menerapkan keamanan seperti melakukan *encode* padaID ketika melakukan proses tambah, ubah, maupun hapus data. Selain keamanan, penelitian selanjutnya diharapkan dapat mengintegrasikan jadwal ujian dengan pemberitahuan berupa surel yang dikirimkan kepada tim pengajar beberapa hari sebelum ujian.

# DAFTAR PUSTAKA

Bean M. 2015. *Laravel 5 Essentials*. Birmingham (UK): Packt Publishing Ltd.

Chanchai S. 2011. *PHP Framework for Database Management Based on MVC Pattern*. *International Jornal of Computer Science & Information Technology* (*IJCSIT*). 3(2):251-252.

Djaali, Muljono P. 2008. *Pengukuran dalam Bidang Pendidikan*. Jakarta (ID): Grasindo.

Gilmore W J. (2015). *Easy Laravel 5 A Hands On Introduction Using a Real-World Project*. Victoria (CA): Leanpub.

Indrajit, Richardus E. 2000. *Pengantar Konsep Dasar: Manajemen Sistem Informasi dan Teknologi Informasi*. Jakarta (ID): PT Elex Media Komputindo.

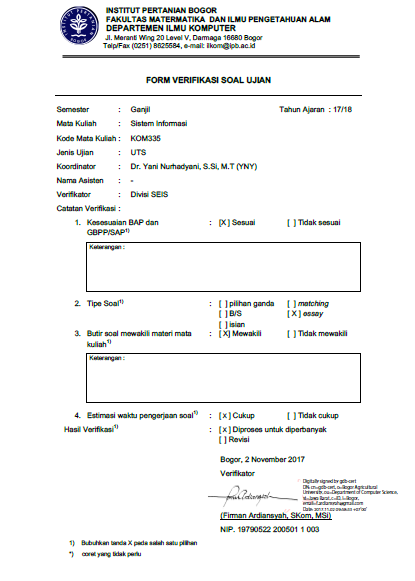
O’Brien James A. 2005. *Introduction of Information System*. Ed ke-12. Northern Arizona (US): McGraw-Hill.

Pressman RS. 2010. *Software Engineering: A Practitioner’s Approach*. Ed ke-7. New York (US): McGraw-Hill.

# LAMPIRAN

Lampiran 1 Berkas pendukung verifikasi soal ujian

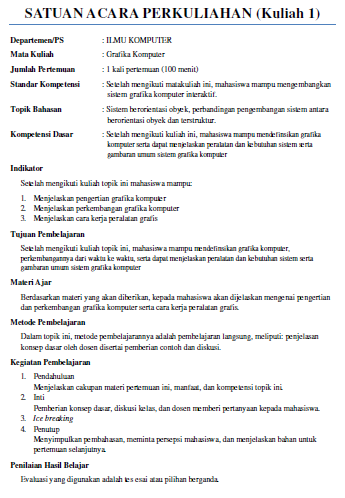
1. Form verifikasi soal ujian (POB/KOM-PP/05/FRM-01-00)



Lampiran 1 Lanjutan

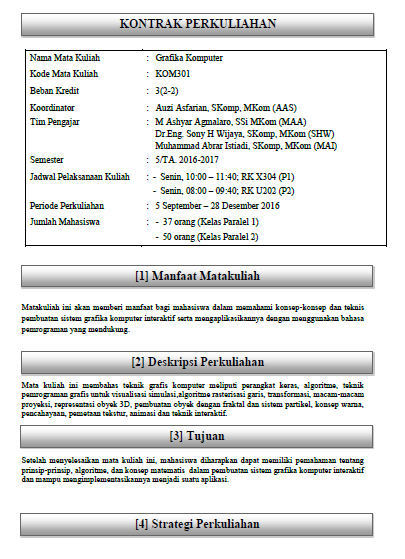
1. Contoh silabus dan SAP mata kuliah





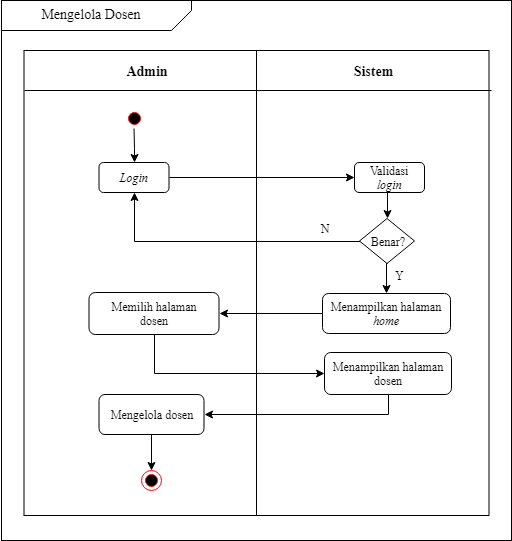
Lampiran 1 Lanjutan

1. Contoh kontrak perkuliahan

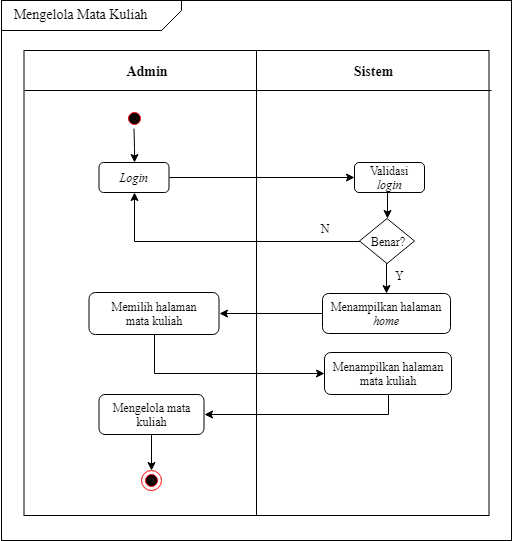


Lampiran 2 *Activity diagram* SIVES iterasi satu

1. *Activity diagram* mengelola dosen

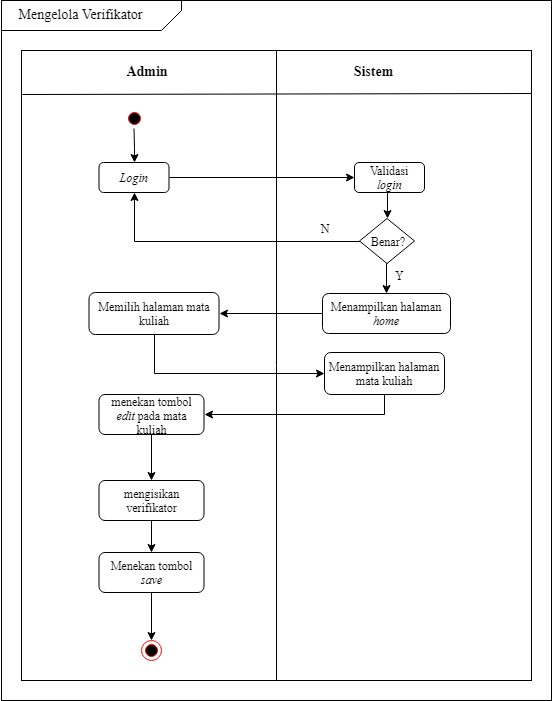


1. *Activity diagram* mengelola mata kuliah

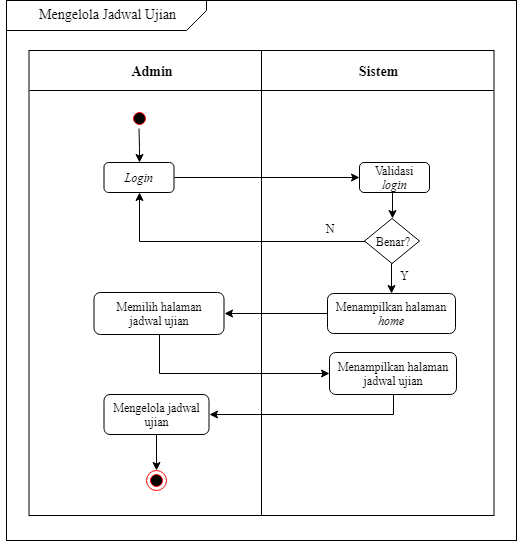


Lampiran 2 Lanjutan

1. *Activity diagram* mengelola verifikator

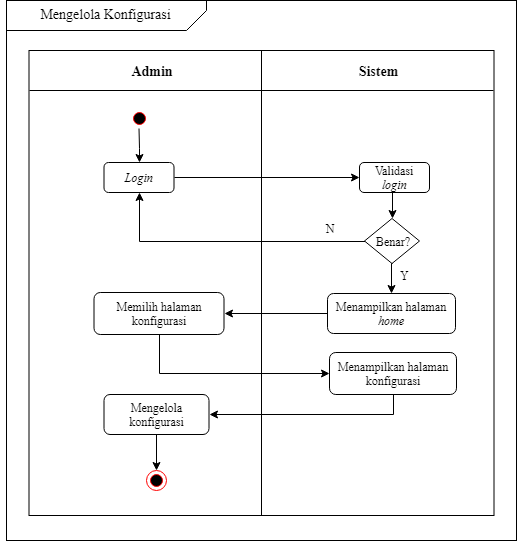


1. *Activity diagram* mengelola jadwal ujian

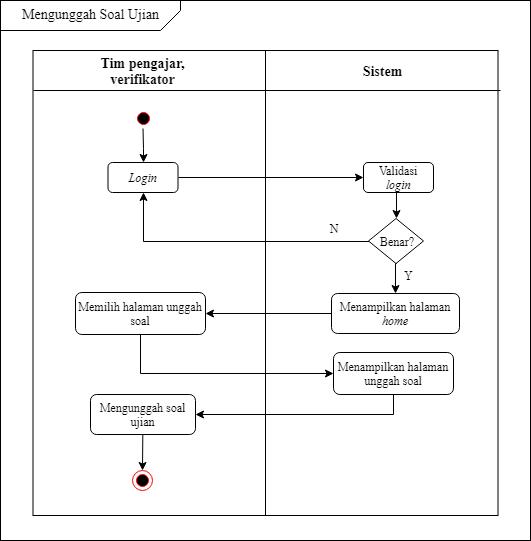


Lampiran 2 Lanjutan

1. *Activity diagram* mengelola konfigurasi

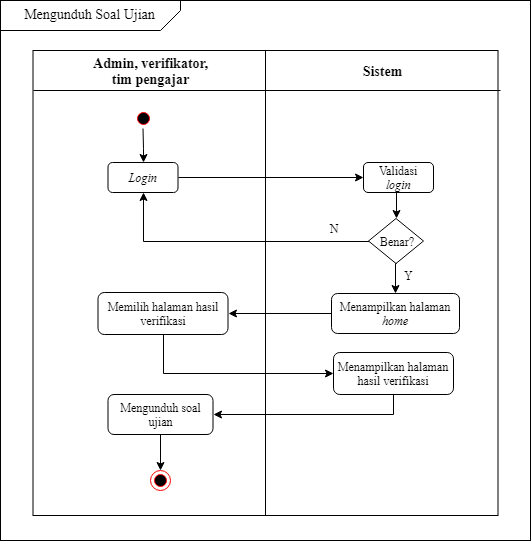


1. *Activity diagram* mengunggah soal ujian

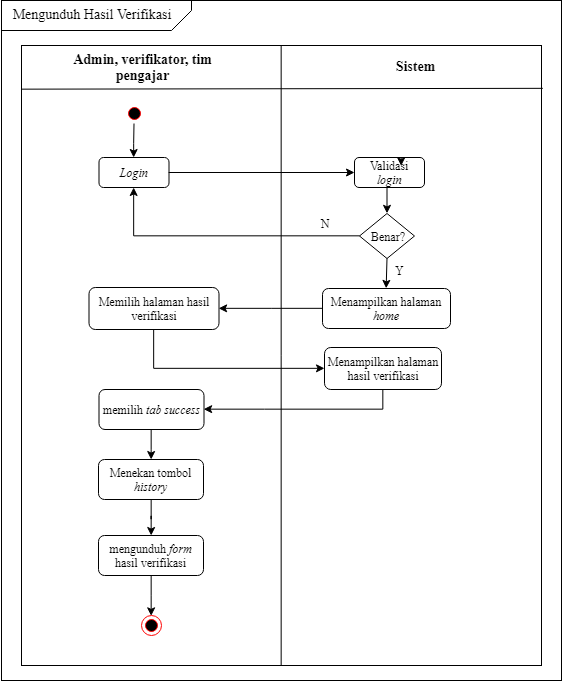


Lampiran 2 Lanjutan

1. *Activity diagram* mengunduh soal ujian

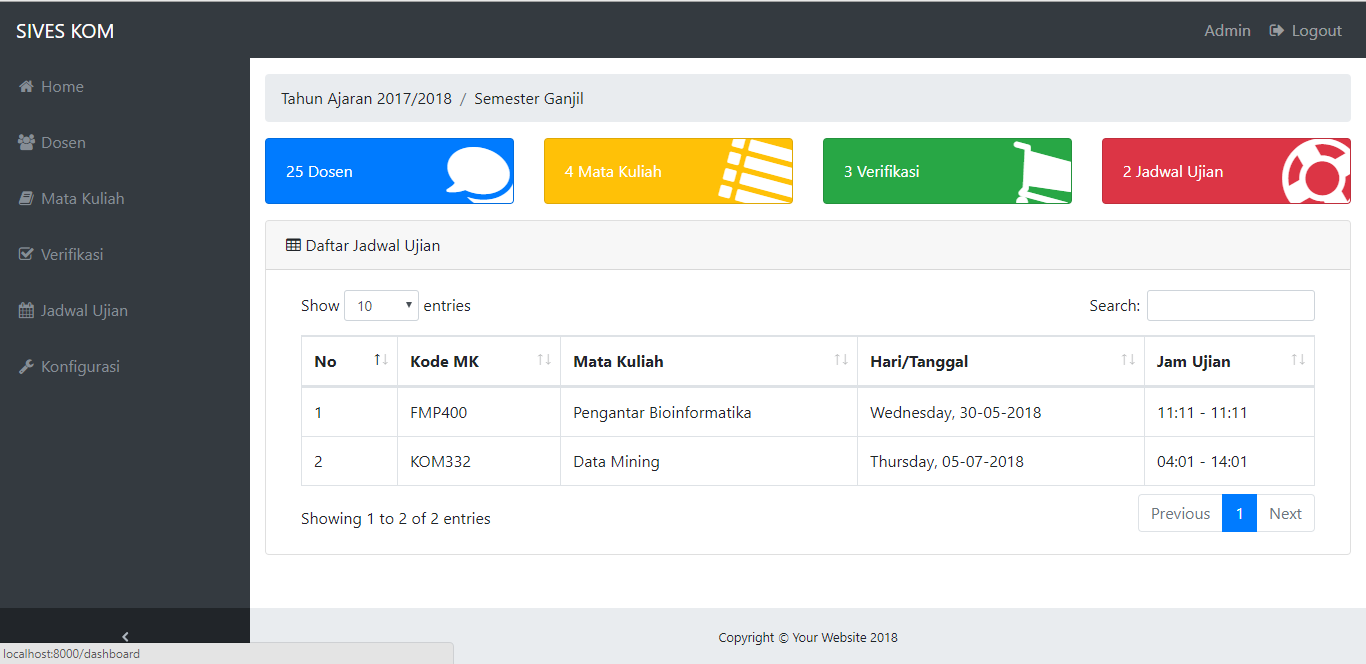


1. *Activity diagram* mengunduh *form* hasil verifikasi

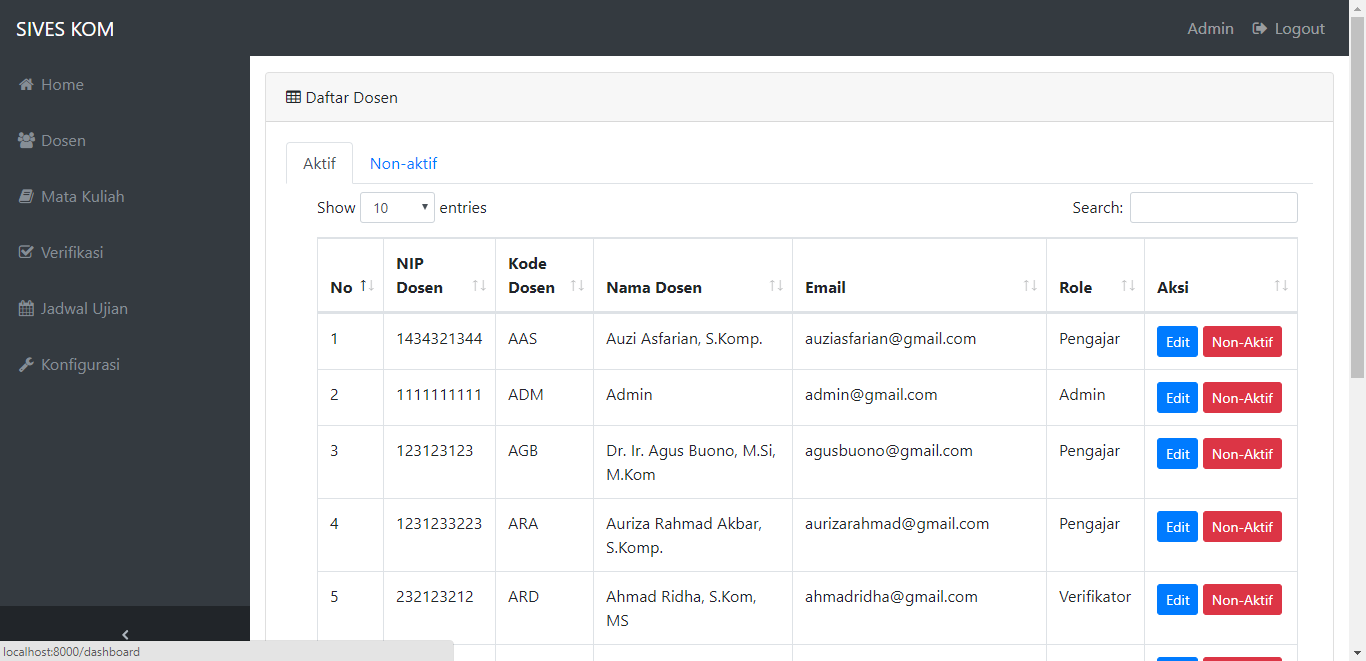


Lampiran 3 Tampilan halaman SIVES iterasi satu

1. Halaman *home* aktor admin

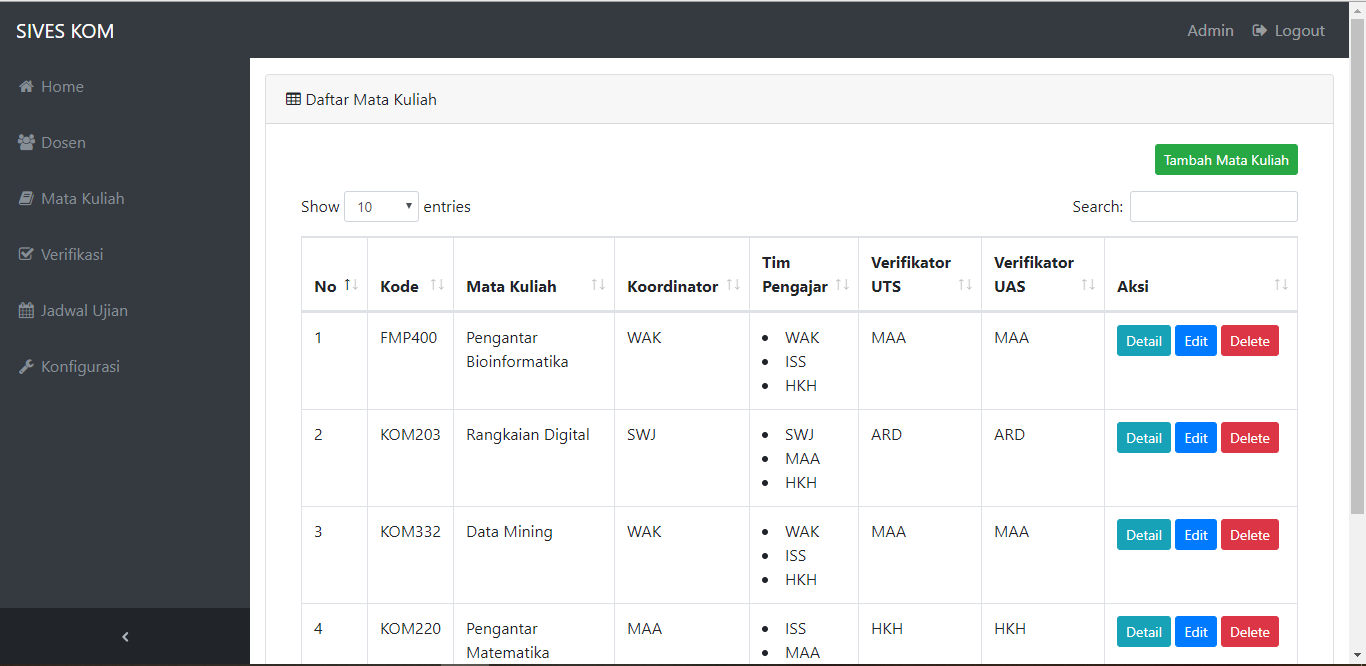


1. Halaman dosen aktor admin

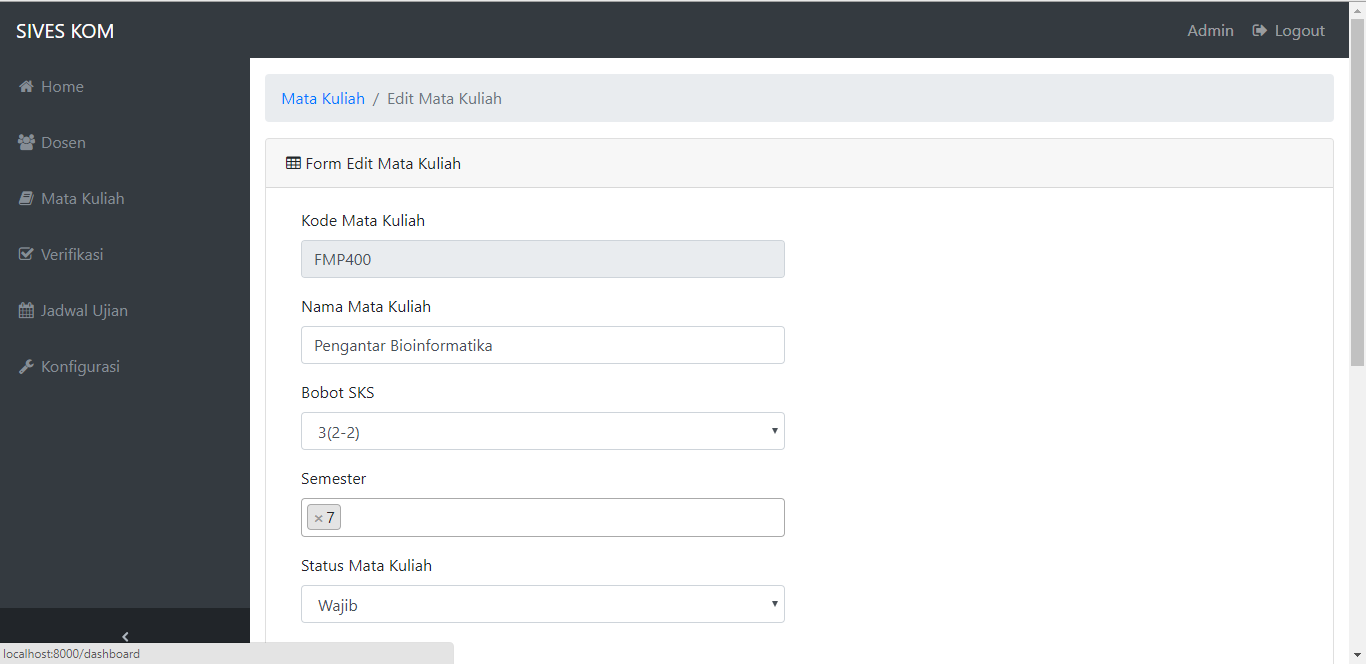


Lampiran 3 Lanjutan

1. Halaman mata kuliah aktor admin

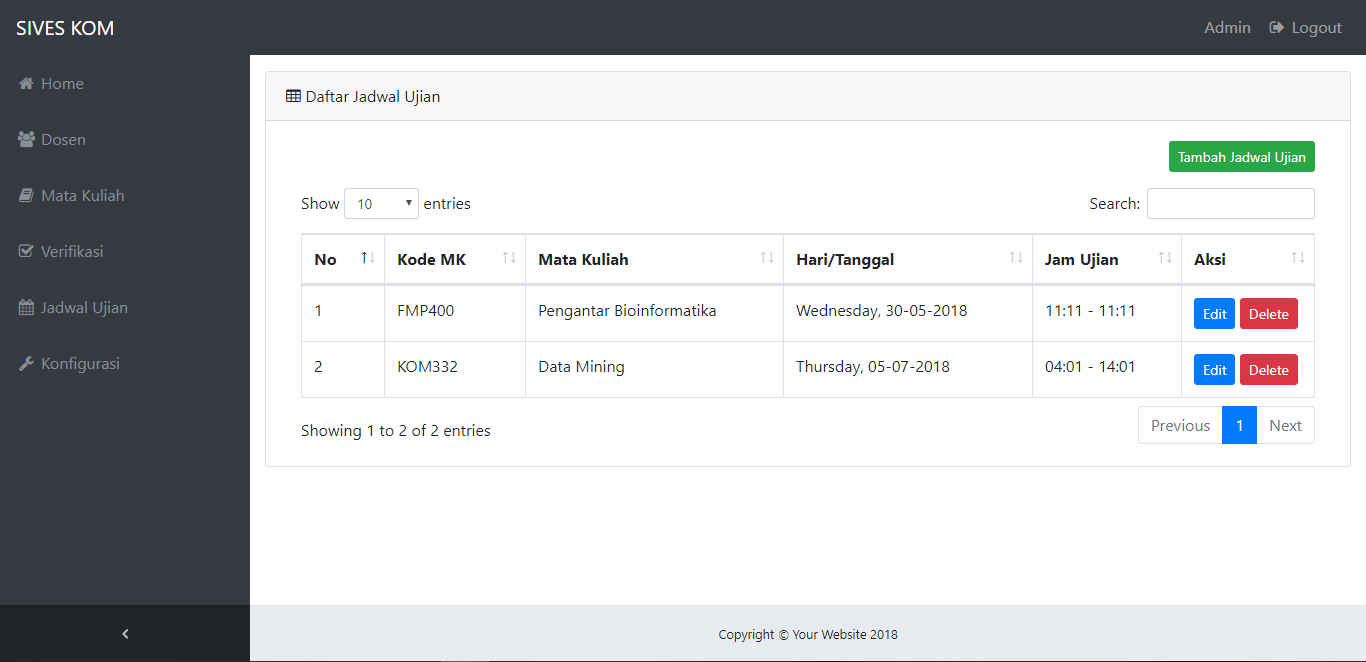


1. Fungsi mengubah mata kuliah aktor admin

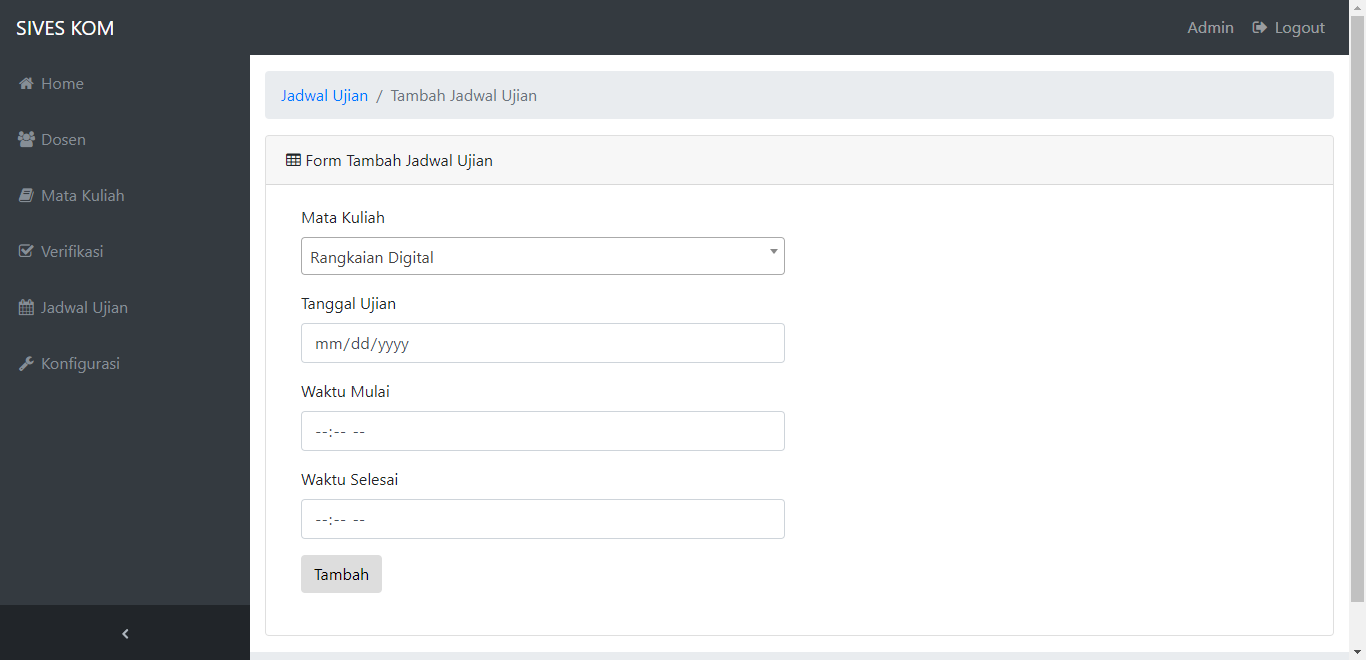


Lampiran 3 Lanjutan

1. Halaman jadwal ujian aktor admin

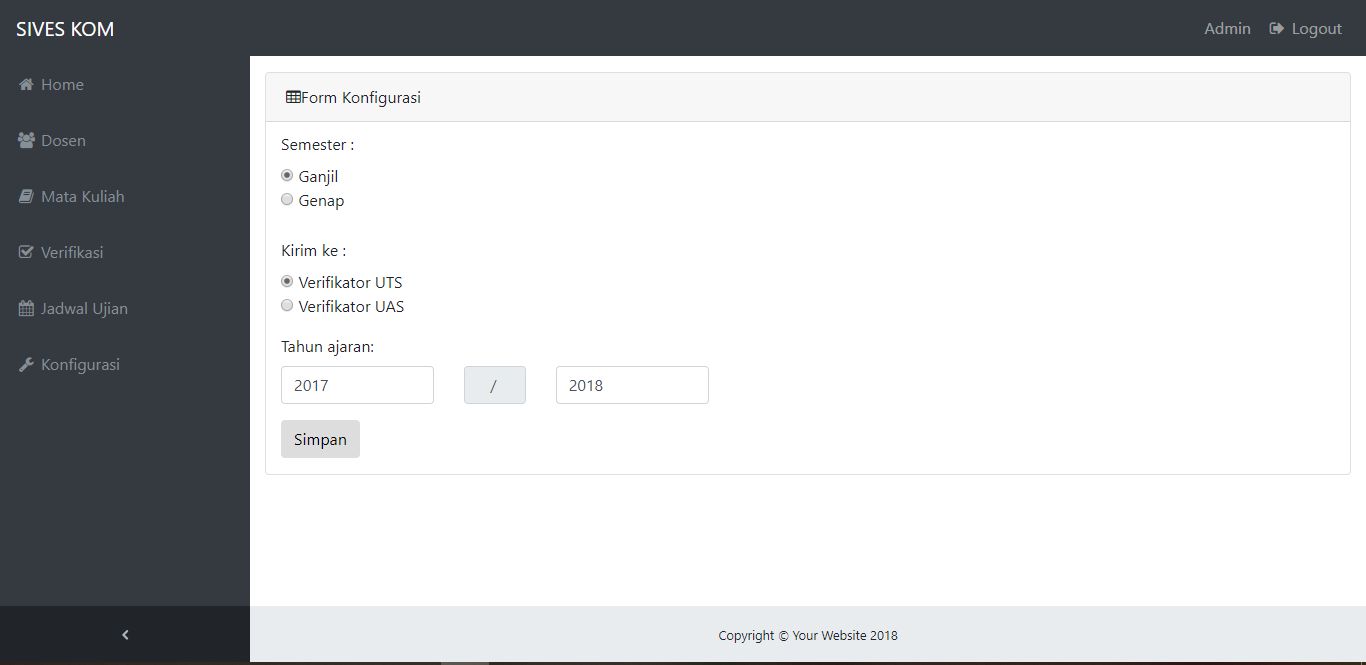


1. Fungsi menambahkan jadwal ujian aktor admin



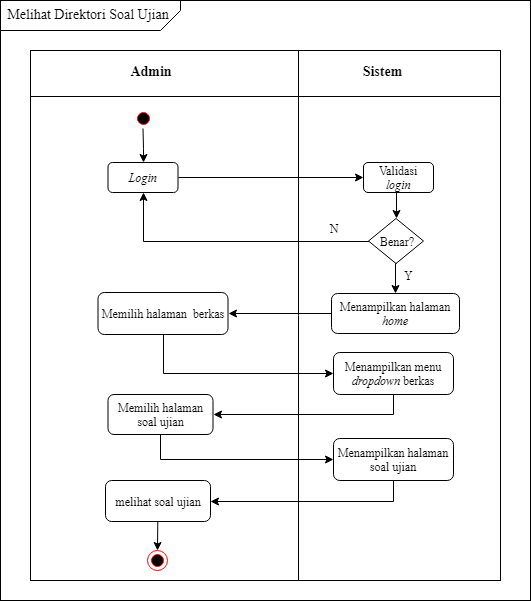
Lampiran 3 Lanjutan

1. Halaman konfigurasi aktor admin

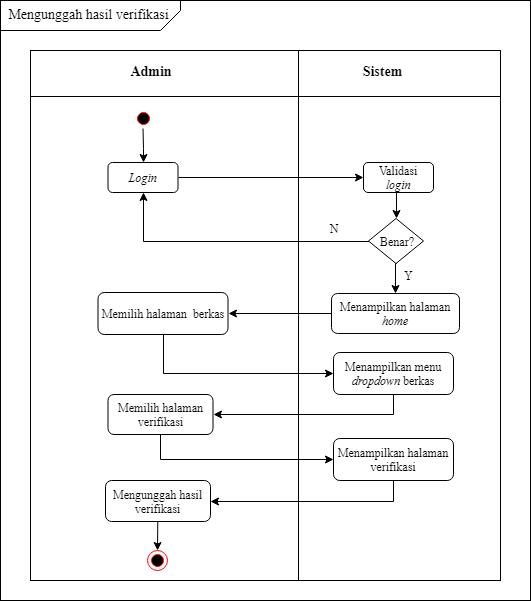


Lampiran 4 *Activity diagram* SIVES iterasi kedua

1. *Activity Diagram* melihat direktori soal ujian

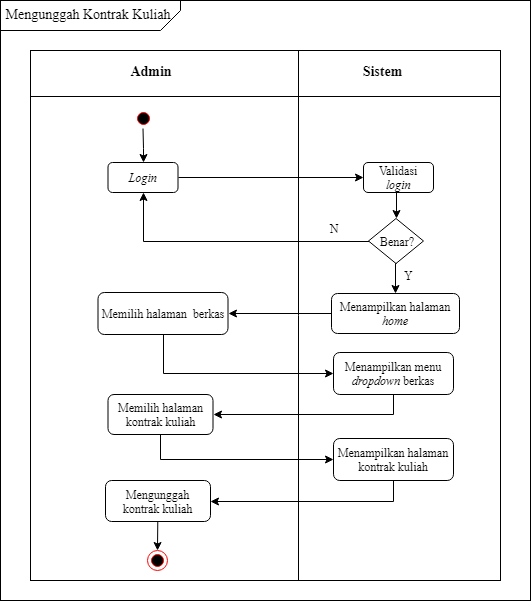


1. *Activity diagram* mengunggah hasil verifikasi

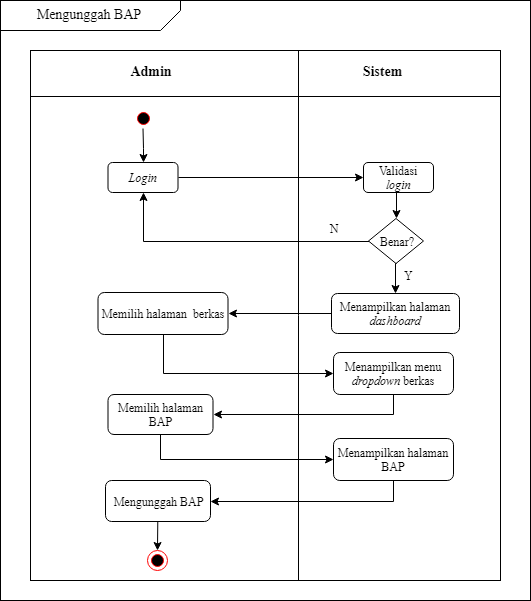


Lampiran 4 Lanjutan

1. *Activity diagram* mengunggah kontrak kuliah

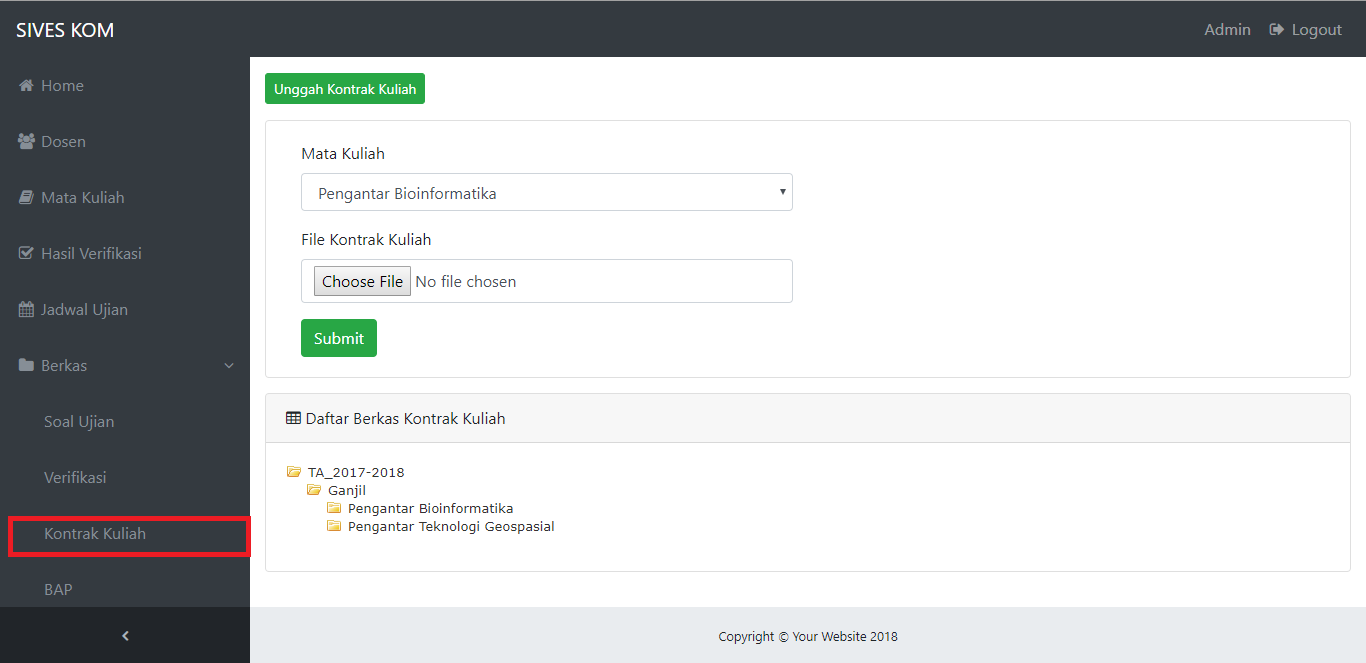


1. Activity diagram mengunggah BAP

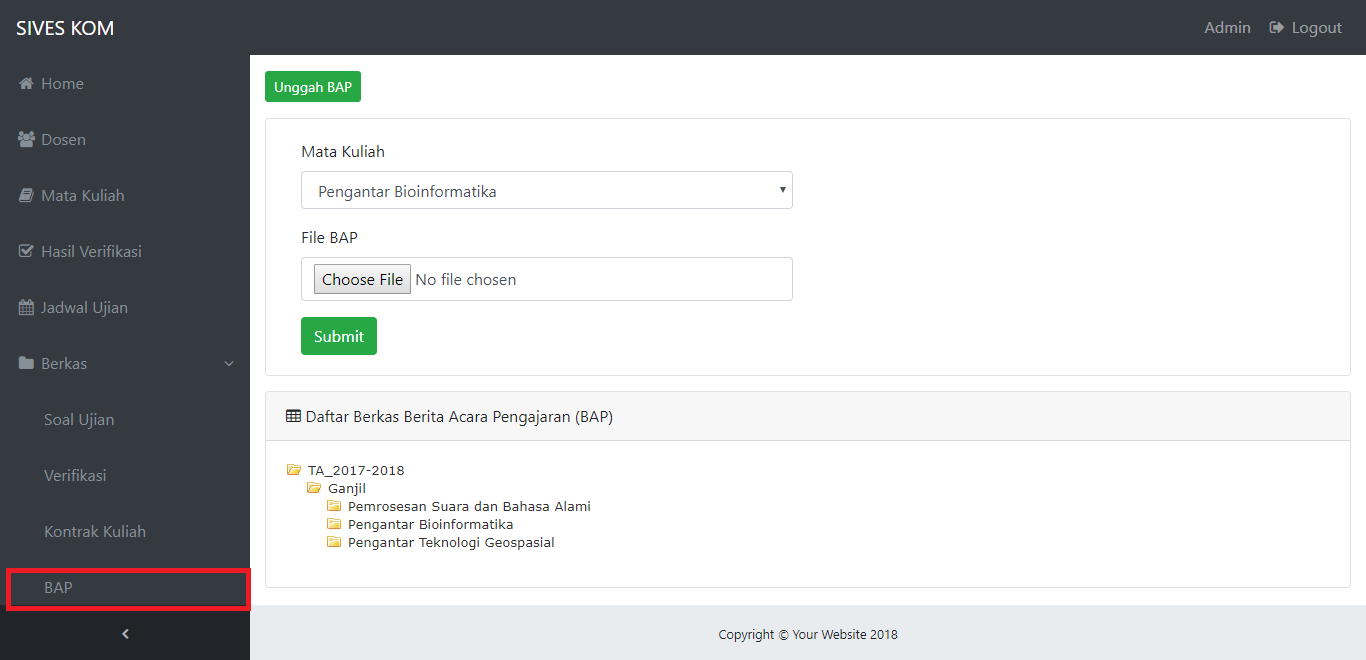


Lampiran 5 Tampilan halaman SIVES iterasi kedua

1. Halaman mengunggah kontrak kuliah aktor admin



1. Halaman mengunggah BAP aktor admin



# RIWAYAT HIDUP

Penulis lahir di Indramayu pada tanggal 28 Maret 1996, anak pertama dari pasangan Rokhmat dan Yomi. Penulis lulus dari SMAN 1 Anjatan pada tahun 2014. Pada tahun yang sama, Penulis lulus SNMPTN dan diterima di Departemen Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam (FMIPA) Institut Pertanian Bogor.

Selama mengikuti perkuliahan, penulis pernah menjadi anggota Divisi Kominfo Ikatan Keluarga dan Mahasiswa Darma Ayu (IKADA Bogor) tahun 2015/2016 dan mengikuti beberapa kegiatan kepanitiaan yang ada di Institut Pertanian Bogor. Di bidang non akademik, Penulis pernah menjadi juara dalam bidang olahraga di acara Olimpiade Mahasiwa IPB (OMI) 2016/2017.

Penulis pernah menerima beasiswa Peningkatan Prestasi Akademik (PPA), dan penerima beasiswa KEB Hana Bank Korea yang sangat membantu penulis dalam mendukung kegiatan akademik. Selain itu, Penulis pernah melaksanakan Praktik Kerja Lapang (PKL) di PT Trans Retail Indonesia dan menjadi Pemakalah pada Seminar Ilmiah Ilmu Komputer (SEMILKOM) 2018 dari Departemen Ilmu Komputer IPB.