

# SKPL FINAL

## SPEKIFIKASI KEBUTUHAN PERANGKAT LUNAK

### Class Scheduler Application

Dipersiapkan oleh:

Kelompok 1

Trisna Ari Roshinta (23518001)


Atika Rahmawati Y. (23518003)

Liptia Venica (23518004)

Nurlaili Rizki H. (23518019)

Program Magister Informatika – STEI ITB

Jl. Ganesha 10, Bandung 40132

	<b>Program Magister Informatika STEI ITB</b>	<b>Nomor Dokumen</b>		<b>Halaman</b>
		<i>SKPL-FINAL</i>		41
		<b>Revisi</b>	<i>B</i>	<i>Tanggal: 29/11/18</i>

## DAFTAR PERUBAHAN

Revisi	Deskripsi
A	Penambahan desain sequence diagram, class diagram, flow chart algoritma pembuatan jadwal, optimasi 1, dan optimasi 2
B	Perbaikan desain sequence diagram dan class diagram, detail kelas diagram bagian model, penambahan pendekatan pengembangan aplikasi

## Daftar Isi

Daftar Isi.....	3
Daftar Gambar.....	4
Daftar Tabel.....	5
1. Pendahuluan .....	6
1.1 Tujuan Penulisan Dokumen .....	6
1.2 Lingkup Masalah .....	6
1.3 Definisi, Istilah dan Singkatan .....	7
1.4 Aturan Penomoran .....	7
1.5 Referensi .....	8
1.6 Deskripsi umum Dokumen .....	8
2 Deskripsi Umum Perangkat Lunak.....	8
2.1 Deskripsi Umum Sistem .....	8
2.2 Fungsi Produk .....	9
2.3 Karakteristik Pengguna .....	9
2.4 Batasan Perancangan .....	9
2.5 Lingkungan Operasi .....	10
2.6 Pendekatan Pengembangan .....	10
3 Deskripsi Umum Kebutuhan .....	11
3.1 Kebutuhan Fungsional .....	11
3.2 Model Diagram .....	13
3.2.1 Diagram Aktivitas.....	14
3.2.2 Diagram <i>Use Case</i> .....	15
3.2.3 Definisi Aktor .....	16
3.2.4 Definisi <i>Use Case</i> .....	16
3.2.5 Diagram <i>Sequence</i> .....	20
3.2.6 Diagram Kelas .....	29
3.2.6.1 Detail Diagram Kelas Model.....	30
3.3 Flowchart Algoritma .....	35
3.3.1 Algoritma Pembuatan Jadwal Kelas .....	35
3.3.2 Algoritma Alokasi Kelas Mahasiswa (Optimasi 1) .....	36
3.3.3 Algoritma Alokasi Kelas Mahasiswa (Optimasi 2) .....	37
LAMPIRAN A: ACTIVITY LOG .....	38
LAMPIRAN B: LEMBAR ASISTENSI.....	41

## Daftar Gambar

Gambar 2-1 Diagram MVC.....	10
Gambar 3-1 Diagram Aktivitas .....	14
Gambar 3-2 Diagram <i>Use Case</i> .....	15
Gambar 3-3 Diagram <i>Sequence</i> untuk Admin.....	20
Gambar 3-4 Diagram <i>Sequence</i> untuk Admin (2).....	21
Gambar 3-5 Diagram <i>Sequence</i> untuk Admin (3).....	22
Gambar 3-6 Diagram <i>Sequence</i> untuk Admin (4).....	23
Gambar 3-7 Diagram <i>Sequence</i> untuk Admin (5).....	24
Gambar 3-8 Diagram <i>Sequence</i> untuk Dosen .....	25
Gambar 3-9 Diagram <i>Sequence</i> untuk Dosen (2).....	26
Gambar 3-10 Diagram <i>Sequence</i> untuk Mahasiswa.....	27
Gambar 3-11 Diagram <i>Sequence</i> untuk Mahasiswa (2) .....	28
Gambar 3-12 Diagram Kelas.....	29
Gambar 3-13 <i>Flowchart</i> Algoritma Pembuatan Jadwal Kelas .....	35
Gambar 3-14 <i>Flowchart</i> Algoritma (Optimasi 1).....	36
Gambar 3-15 <i>Flowchart</i> Algoritma (Optimasi 2).....	37

## Daftar Tabel

Tabel 1 Daftar Definisi, Akronim Singkatan.....	7
Tabel 2 Karakteristik Pengguna .....	9
Tabel 3 Lingkungan Operasi .....	10
Tabel 4 Kebutuhan Fungsional.....	11
Tabel 5 Definisi Aktor.....	16
Tabel 6 Definisi <i>Use Case</i> .....	16

## **1. Pendahuluan**

Dokumen ini akan berisi Spesifikasi Kebutuhan Perangkat Lunak (SKPL) sebagai acuan teknis pengembangan aplikasi Class Scheduler. Isi dari dokumen ini mengikuti template GL01 yang dikeluarkan oleh Program Magister STEI ITB.

### **1.1 Tujuan Penulisan Dokumen**

Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) ini dibuat untuk tujuan sebagai berikut:

1. Mendefinisikan dan menjelaskan hal-hal yang diperlukan dalam pengembangan aplikasi Class Scheduler.
2. Memperjelas detail spesifikasi kebutuhan dan lingkup kerja yang akan dilakukan dalam pengembangan aplikasi Class Scheduler beserta kendala-kendala yang mungkin dihadapi.
3. Mendefinisikan dan mendeskripsikan secara global aplikasi Class Scheduler yang akan dikembangkan, yang menggambarkan fungsionalitas, performansi, batasan perancangan, atribut, serta antarmuka eksternal aplikasi yang akan diimplementasikan.
4. Menjadi acuan teknis oleh pengembang perangkat lunak serta gambaran yang lebih detail untuk pada tahap-tahap selanjutnya serta memberikan gambaran secara lebih detail tentang proses.

Adapun pihak-pihak yang berkepentingan yang akan menggunakan dokumen SKPL ini adalah:

- a. Ibu Fazat dan Bapak Hari sebagai dosen matakuliah IF5121 dan asisten mata kuliah atau *reviewer* tugas SKPL Class Scheduler, yang akan menggunakan dokumen SKPL untuk dinilai dan dievaluasi konten dan formatnya.
- b. Kelompok 1 sebagai penulis yang menggunakan dokumen SKPL sebagai acuan pengembang perangkat lunak.

### **1.2 Lingkup Masalah**

Aplikasi Class Scheduler adalah sebuah aplikasi yang digunakan untuk merencanakan jadwal kelas mahasiswa berdasarkan kompetensi pekerjaan yang sesuai. Jadwal kelas direkomendasikan secara otomatis dengan menggunakan algoritma penjadwalan sesuai dengan optimasi mahasiswa yang memenuhi kompetensi pekerjaan atau optimasi penghasilan terbesar

yang diperoleh penyelenggara kelas (universitas). Kelas yang dibuka harus memenuhi persyaratan jumlah minimal mahasiswa yang mengikuti sejumlah 10 orang dengan batas maksimal 25 orang.

### 1.3 Definisi, Istilah dan Singkatan

Tabel 1.1 Daftar Definisi, Akronim Singkatan

No.	Istilah	Keterangan
1.	Admin	Pengguna yang merupakan administrator dari sistem
2.	Dosen	Pengguna yang merupakan pengajar kelas kompetensi
3.	Mahasiswa	Pengguna yang akan mengikuti kelas kompetensi untuk memenuhi kebutuhan kompetensi pekerjaan
5.	Pekerjaan	Nama pekerjaan terdaftar yang diinginkan oleh mahasiswa
4.	Kompetensi	Nama kebutuhan kompetensi yang harus dimiliki untuk memenuhi suatu pekerjaan
6.	Kelas	Kelas terkait kompetensi yang diajarkan satu dosen kepada beberapa mahasiswa yang membutuhkan kompetensi tersebut
7.	Jadwal	Jadwal kelas yang dihasilkan dari hasil optimasi yang ditentukan oleh admin

### 1.4 Aturan Penomoran

1. UC-xx adalah penomoran yang digunakan untuk merepresentasikan Use Case, UC adalah Use Case dan xx adalah nomor urut Use Case.
2. KF-xx adalah penomoran yang digunakan untuk merepresentasikan kebutuhan fungsional, KF adalah kebutuhan Fungsional dan xx adalah nomor urut kebutuhan.
3. KNF-xx adalah penomoran yang digunakan untuk merepresentasikan kebutuhan non-fungsional, KNF adalah Kebutuhan Non-Fungsional dan xx adalah nomor urut kebutuhan.

## 1.5 Referensi

Referensi yang digunakan pada perangkat lunak ini adalah:

1. GL01, Spesifikasi Kebutuhan Perangkat Lunak yang dikeluarkan oleh Program Magister Informatika ITB.

## 1.6 Deskripsi umum Dokumen

Sistematika pembahasan dokumen SKPL ini adalah sebagai berikut:

DAFTAR PERUBAHAN

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

### 1. PENDAHULUAN

Bagian ini berisi pengantar dokumen SKPL yang berisi tujuan penulisan dokumen, lingkup masalah pengembangan perangkat lunak, istilah dan singkatan yang digunakan, aturan penomoran yang digunakan dalam dokumen, referensi serta deskripsi umum yang merupakan ikhtisar dokumen SKPL.

### 2. DESKRIPSI UMUM PERANGKAT LUNAK

Bagian ini mendefinisikan deskripsi umum sistem, fungsi produk, karakteristik pengguna, batasan dan lingkungan operasi.

### 3. DEKRIPSI UMUM KEBUTUHAN

Bagian ini menjelaskan kebutuhan antarmuka eksternal, dekripsi fungsional yang digambarkan dengan *use case diagram*, *class diagram*, *component diagram*, *sequence diagram*, dan *state diagram*, kemudian menjelaskan data requirement, kebutuhan non-fungsional, batasan perancangan, keruntutan, dan ringkasan kebutuhan.

## 2 Deskripsi Umum Perangkat Lunak

Pada bagian ini akan didefinisikan deskripsi umum sistem, fungsi produk, karakteristik pengguna, batasan dan lingkungan operasi.

### 2.1 Deskripsi Umum Sistem

Untuk merencanakan jadwal kelas yang mengajarkan kompetensi untuk memenuhi kebutuhan suatu pekerjaan. Aplikasi ini memiliki tiga macam pengguna, yaitu admin



selaku pengelola data yang ada pada aplikasi, dosen sebagai pengajar kelas, dan mahasiswa sebagai peserta kelas. Aplikasi Class Scheduler menyediakan daftar kompetensi yang dapat dipelajari oleh mahasiswa untuk memenuhi kebutuhan dari pekerjaan yang terdaftar pada aplikasi. Suatu kelas kompetensi dapat dibuka ketika ada minimal 10 mahasiswa yang akan mengikuti kelas tersebut dengan batas jumlah 25 mahasiswa. Penjadwalan dilakukan secara otomatis dengan memilih jenis optimasi yang akan dilakukan. Optimasi ini bisa berdasarkan dari keterselesaiannya mahasiswa dalam memenuhi kompetensi suatu pekerjaan atau berdasarkan kombinasi terbesar untuk penghasilan terbesar bagi pihak penyelenggara kelas.

## 2.2 Fungsi Produk

Class Scheduler menyediakan fungsi untuk memberikan jadwal kelas yang sesuai dengan kebutuhan kelas dan ketersediaan waktu dosen pengampu.

## 2.3 Karakteristik Pengguna

Tabel 2 Karakteristik Pengguna

Kategori Pengguna	Tugas	Hak Akses ke aplikasi
Admin	Mengelola aplikasi, seperti menambahkan, menghapus, dan mengubah data terkait kompetensi, pekerjaan, dan akun. Melakukan registrasi akun dosen dan mahasiswa,	Menu registrasi, kelola akun, kelola data kompetensi, kelola data pekerjaan, daftar pemasukan dan tagihan
Dosen	Mengajukan ketersediaan jadwal mengajar dan kompetensi awal yang mampu diajarkan	Menu tanggal mengajar, jadwal kelas, dan kelola data diri
Mahasiswa	Mengajukan kompetensi awal yang dimiliki, pekerjaan pilihan, dan hari	Menu kelola pekerjaan pilihan, riwayat tagihan, jadwal kelas

## 2.4 Batasan Perancangan

Batasan pada perangkat lunak Class Scheduler adalah antara lain:

1. Sistem hanya menerima parameter masukan sesuai yang telah ditentukan
2. Kompetensi yang dipilih oleh dosen dan mahasiswa hanya yang sudah terdaftar pada sistem
3. Penjadwalan yang pertama kali dilakukan merupakan hasil yang paling baik dari segi kesesuaian jadwal mahasiswa
4. Sistem tidak menangani pemeriksaan pembayaran secara otomatis

## 2.5 Lingkungan Operasi

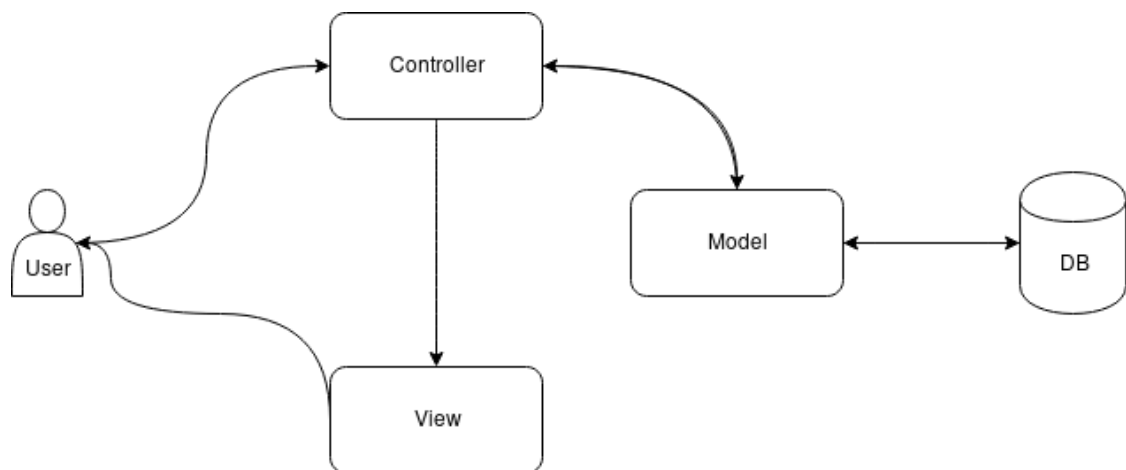
Berikut adalah lingkungan operasi yang dibutuhkan untuk menjalankan sistem Class Scheduler.

Tabel 3 Lingkungan Operasi

Hardware	Spesifikasi
PC (Personal Computer)	64 bit, RAM 4 GB
Storage	500 GB

## 2.6 Pendekatan Pengembangan

Dalam mengembangkan aplikasi Class Scheduler ini, kelompok 1 menggunakan pendekatan Object Oriented Programming (OOP) dengan metode MVC (Model, View, Controller). Berikut ini merupakan penjelasan singkat terkait MVC.



Gambar 2-1 Diagram MVC

- **Model:** Bertugas untuk mengatur, menyiapkan, memanipulasi dan mengorganisasikan data (dari database) sesuai dengan instruksi dari controller.
- **View:** Bertugas untuk menerima & menampilkan informasi dari dan oleh user sesuai dengan instruksi dari controller.
- **Controller:** Bertugas untuk mengatur apa yang harus dilakukan model, dan view mana yang harus ditampilkan berdasarkan permintaan dari user.

### 3 Deskripsi Umum Kebutuhan

Menjelaskan deskripsi umum kebutuhan dalam membangun sebuah sistem Class Scheduling

#### 3.1 Kebutuhan Fungsional

Tabel 4 Kebutuhan Fungsional

ID	Kebutuhan
KF-01	Sistem dapat menerima login dari admin, dosen, dan mahasiswa
KF-02	Sistem dapat menerima masukan data kompetensi untuk setiap pekerjaan
KF-03	Sistem dapat menerima masukan daftar pekerjaan yang dapat sesuai dengan minat dan bakat mahasiswa
KF-04	Sistem dapat memberhentikan periode pengisian data dosen dan mahasiswa
KF-05	Sistem dapat menampilkan seluruh tagihan mahasiswa
KF-06	Sistem dapat melihat seluruh peserta kelas dan dosen pengampunya
KF-07	Sistem dapat menampilkan total penghasilan yang diperoleh dari semua kelas yang dibuka
KF-08	Sistem dapat menampilkan daftar mahasiswa yang sudah dan belum membayar
KF-09	Sistem dapat menerima masukan validasi pembayaran tagihan mahasiswa

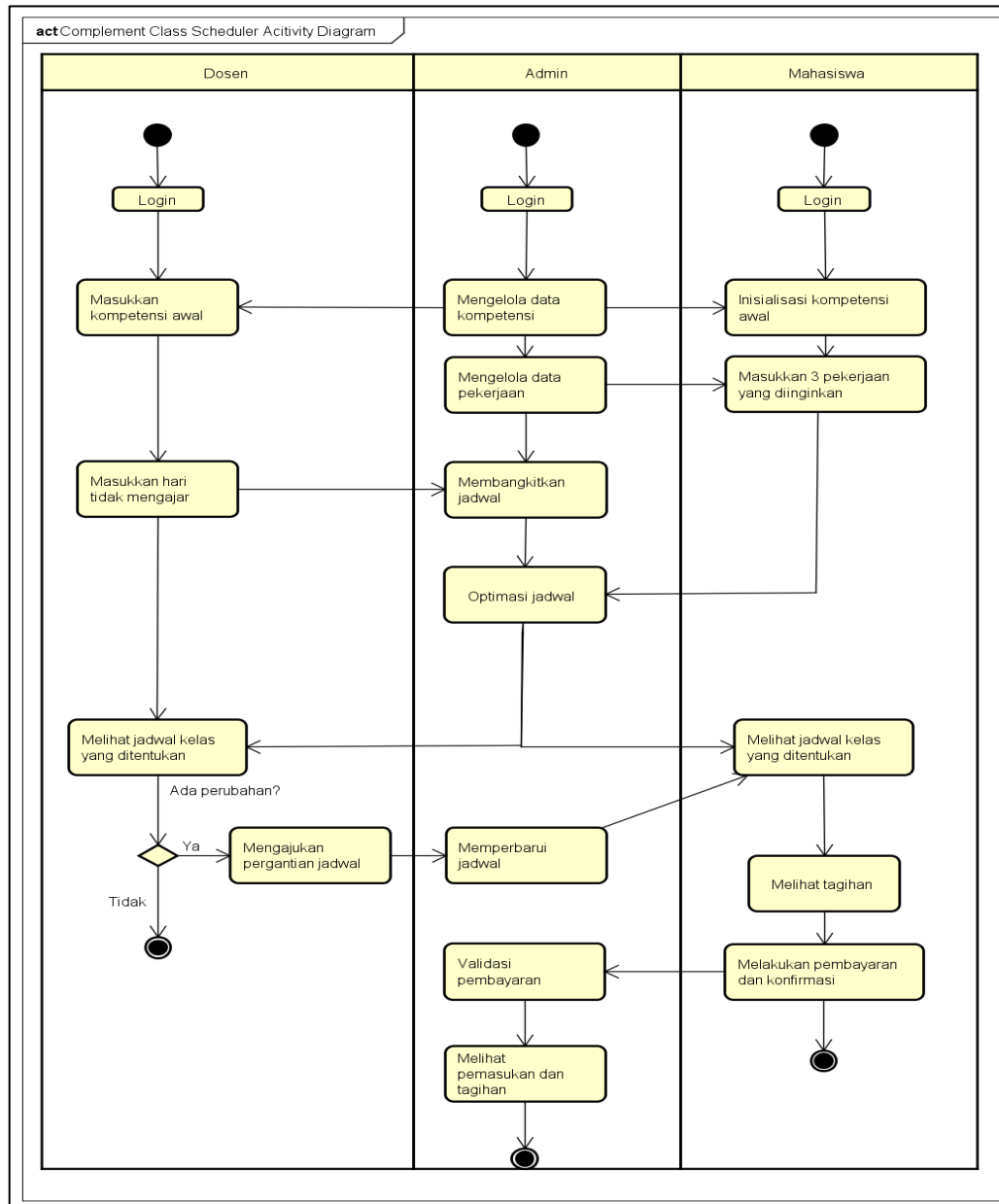
<b>ID</b>	<b>Kebutuhan</b>
KF-10	Sistem dapat menampilkan jumlah mahasiswa yang berhasil memenuhi kompetensi pekerjaan per periode pengisian
KF-11	Sistem dapat menampilkan daftar dosen dan mahasiswa yang terdaftar pada suatu kelas
KF-12	Sistem dapat menampilkan jadwal kelas yang diampu oleh dosen bersangkutan
KF-13	Sistem dapat menerima masukan permintaan perubahan jadwal kelas yang sudah ditentukan ke jadwal baru yang sesuai ketersediaan secara otomatis
KF-14	Sistem dapat menerima masukan permintaan pengubahan data diri pengguna (dosen/mahasiswa)
KF-15	Sistem dapat menerima masukan permintaan pengubahan password pengguna (dosen/mahasiswa)
KF-16	Sistem dapat menerima masukan kompetensi yang dimiliki dosen untuk diajarkan
KF-17	Sistem dapat menerima masukan daftar hari libur yang berlaku bagi dosen dan mahasiswa
KF-18	Sistem dapat menampilkan riwayat tagihan yang dikenakan pada mahasiswa
KF-19	Sistem dapat menerima masukan bukti pembayaran berupa kode transaksi
KF-20	Sistem dapat menerima masukan berupa tiga pekerjaan terdaftar yang diinginkan mahasiswa
KF-21	Sistem dapat menghasilkan jadwal berdasarkan pilihan optimasi
KF-21	Sistem dapat menerima masukan daftar kompetensi yang sudah dimiliki mahasiswa

### 3.2 Model Diagram

Berikut ini merupakan daftar diagram pemodelan sistem yang dijelaskan yang terdiri dari beberapa diagram sebagai berikut.

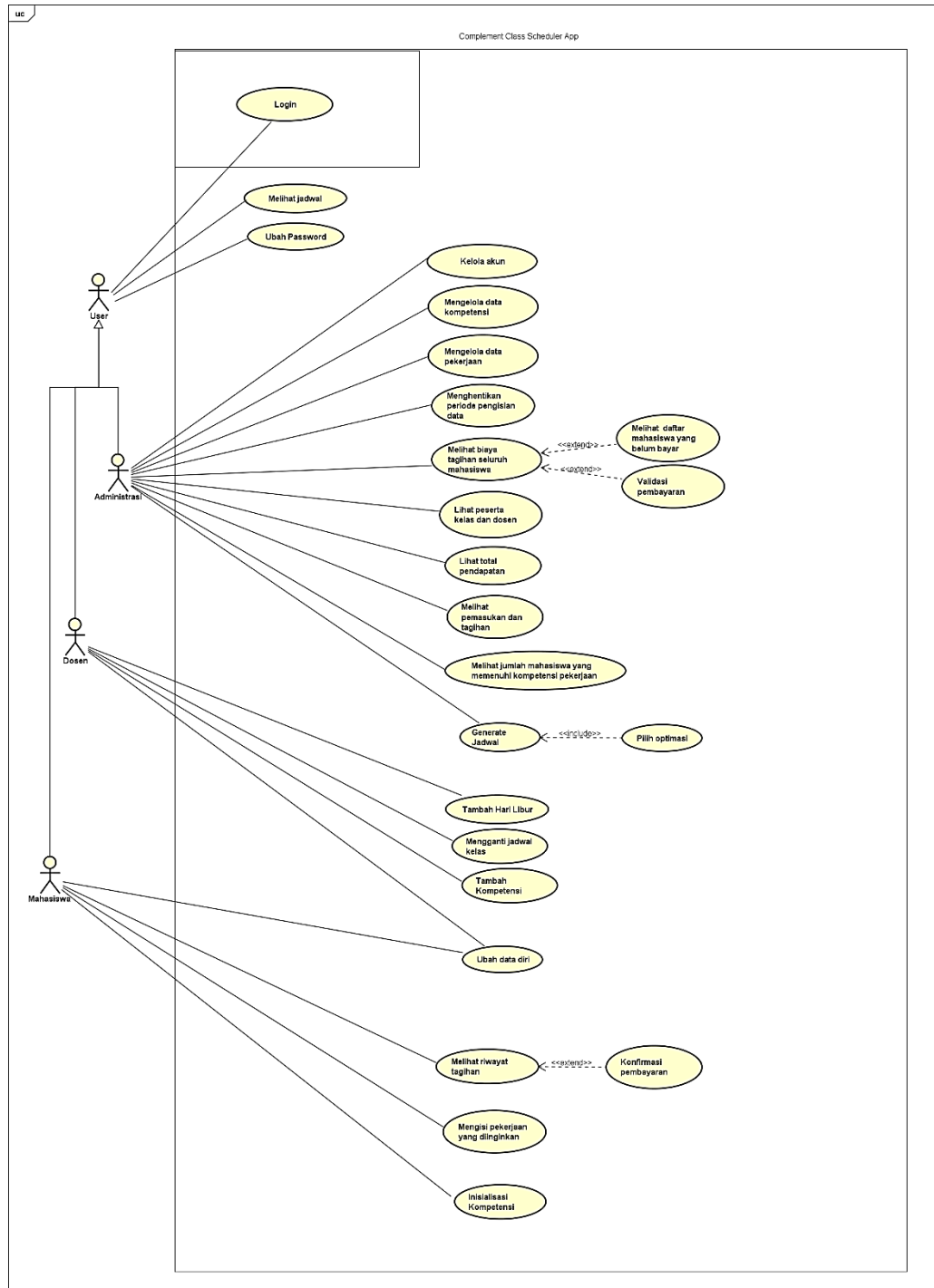
- Diagram Aktivitas: untuk menggambarkan alur kerja sistem secara keseluruhan
- Diagram *Use Case*: untuk menggambarkan kapabilitas penggunaan sistem bagi pengguna, yakni admin, dosen, dan mahasiswa
- Diagram *Sequence*: untuk menggambarkan alur keterhubungan fungsi antar satu sama lain.
- Diagram Kelas: untuk menggambarkan keterhubungan kelas Model, View, dan Controller.
  - Detail Kelas Model: untuk menggambarkan atribut dan fungsi utama dalam sistem

### 3.2.1 Diagram Aktivitas



Gambar 3-1 Diagram Aktivitas

### 3.2.2 Diagram Use Case



Gambar 3-2 Diagram Use Case

### 3.2.3 Definisi Aktor

Tabel 5 Definisi Aktor

No.	Aktor	Deskripsi
1.	Admin	Aktor dengan <i>role</i> ini mempunyai wewenang untuk masuk ke dalam sistem dan mengelola semua hal berkaitan dengan sistem, seperti mengelola jadwal, data kompetensi, data pekerjaan, biaya tagihan, pemasukan, dan sebagainya.
2.	Dosen	Aktor dengan <i>role</i> ini mempunyai wewenang untuk masuk ke dalam sistem dan mengelola hal-hal yang berkaitan dengan penggantian jadwal, hari libur, dan data akun pribadi.
3.	Mahasiswa	Aktor dengan <i>role</i> ini mempunyai wewenang untuk masuk ke dalam sistem dan melakukan pengisian data pekerjaan yang diinginkan, hari libur, dan data akun pribadi

### 3.2.4 Definisi Use Case

Berikut adalah daftar *use case* dan deskripsinya pada sistem Class Scheduler.

Tabel 6 Definisi Use Case

ID	Use Case	Deskripsi
UC-01	Login	Sistem menampilkan menu login untuk user
UC-02	Membangkitkan (generate) Jadwal	Sistem menampilkan pilihan optimasi untuk menghasilkan jadwal
UC-03	Mengelola Data Kompetensi	Sistem menampilkan formulir untuk menambah dan mengubah data terkait kompetensi pekerjaan yang tersedia



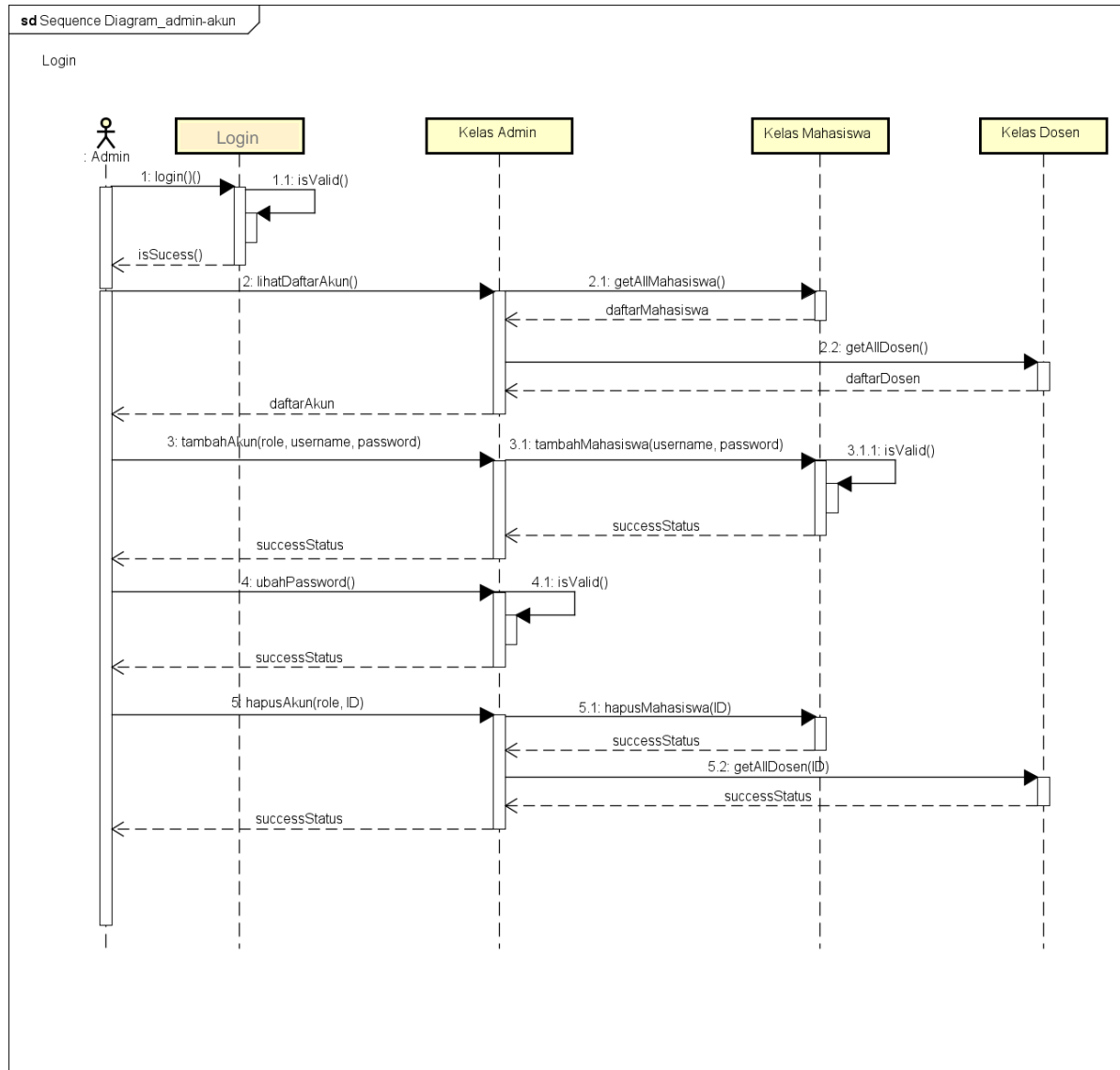
<b>ID</b>	<b>Use Case</b>	<b>Deskripsi</b>
UC-04	Mengelola Data Pekerjaan	Sistem menampilkan formulir untuk menambah dan mengubah data terkait pekerjaan yang sesuai kebutuhan
UC-05	Menghentikan Periode Pengisian Data	Sistem menampilkan tombol perintah yang dapat mematikan fungsi penerimaan masukan dari mahasiswa dan dosen
UC-06	Melihat Biaya Tagihan Seluruh Mahasiswa	Sistem menampilkan daftar tagihan untuk setiap mahasiswa
UC-07	Validasi Pembayaran	Sistem menampilkan tabel daftar mahasiswa yang sudah melakukan pembayaran dan tombol untuk validasi kebenaran pembayaran
UC-08	Melihat Daftar Mahasiswa yang Belum Membayar	Sistem menampilkan tabel daftar mahasiswa yang belum melakukan pembayaran
UC-09	Melihat Peserta Kelas dan Dosen	Sistem menampilkan daftar peserta dan dosen untuk setiap kelas kompetensi
UC-10	Melihat Total Pendapatan	Sistem menampilkan data total pendapatan yang diperoleh pada satu semester pembelajaran

<b>ID</b>	<b>Use Case</b>	<b>Deskripsi</b>
UC-11	Melihat Pemasukan dan Tagihan	Sistem menampilkan daftar tagihan yang sudah dibayar (pemasukan) dan yang belum dibayar (tagihan) oleh mahasiswa
UC-12	Melihat Jumlah Mahasiswa yang Memenuhi Kompetensi Pekerjaan	Sistem menampilkan daftar mahasiswa yang telah mengambil seluruh kelas 12kompetensi yang dibutuhkan dalam suatu pekerjaan
UC-13	Mengelola User Account	Sistem menampilkan tabel daftar akun dan perintah untuk mengubah atau menghapus akun
UC-14	Melihat Jadwal	Sistem menampilkan tabel jadwal kelas yang diampu oleh dosen yang bersangkutan
UC-15	Mengganti Jadwal Kelas	Sistem menampilkan formulir pengajuan pengubahan jadwal kelas dari jadwal semula ke jadwal lain yang sesuai dengan jadwal mahasiswa
UC-16	Ubah Data Diri	Sistem menampilkan formulir data diri yang datanya dapat diubah dan disimpan
UC-17	Ubah Password	Sistem menampilkan formulir pembaruan password dengan

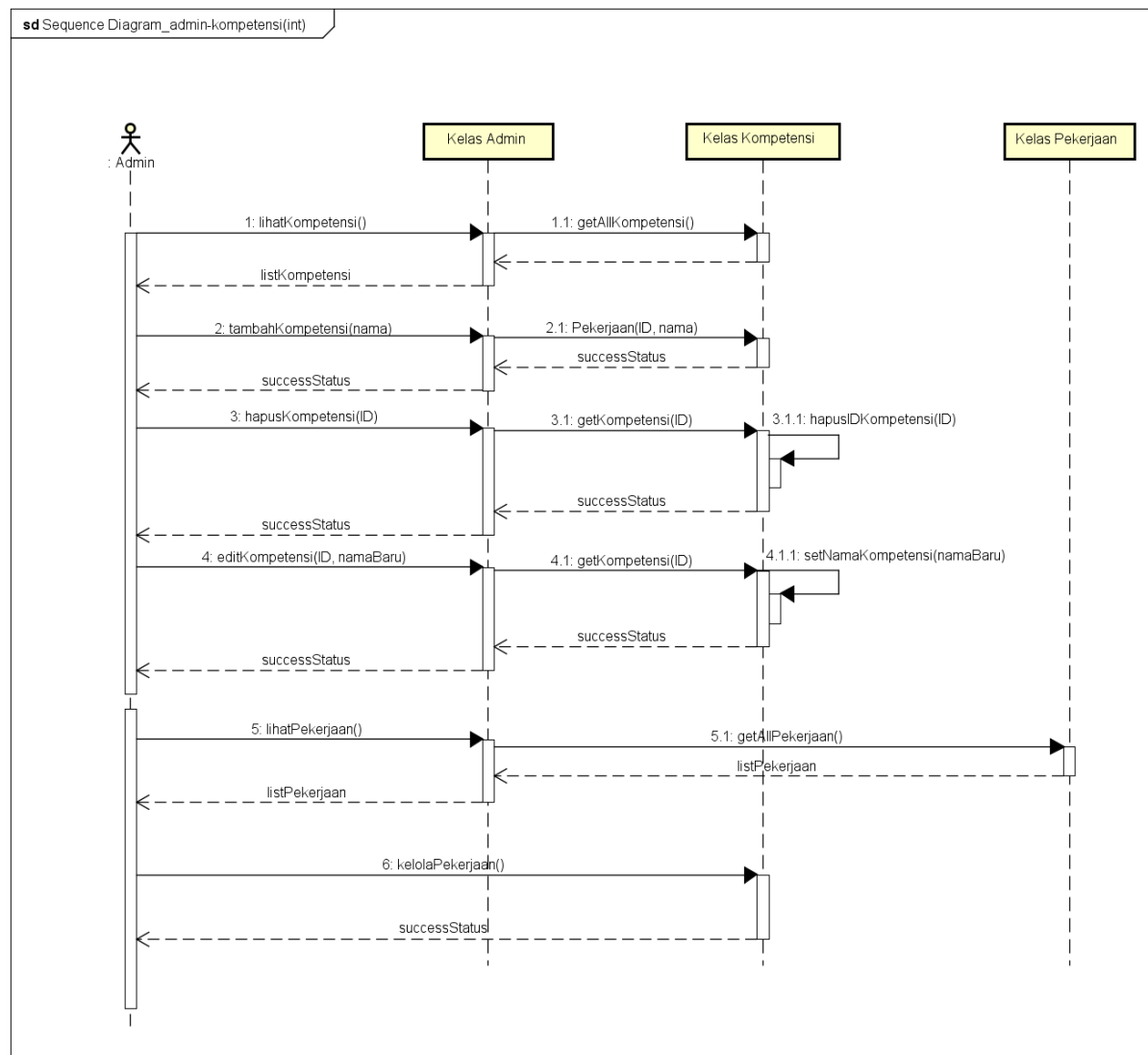
ID	Use Case	Deskripsi
		pengisian password lama dan password baru
UC-18	Tambah Kompetensi	Sistem menampilkan formulir untuk memasukkan kompetensi yang dapat diajarkan
UC-19	Tambah Hari Libur	Sistem menampilkan formulir untuk memasukkan hari libur untuk dosen
UC-20	Melihat Riwayat Tagihan	Sistem menampilkan tabel berisi seluruh daftar tagihan mahasiswa
UC-21	Konfirmasi Pembayaran	Sistem menampilkan formulir untuk memasukkan kode transaksi pembayaran yang sudah dilakukan
UC-22	Mengisi Pekerjaan yang Diinginkan	Sistem menampilkan formulir untuk memasukkan pekerjaan yang diinginkan dari daftar yang tersedia
UC-23	Inisialisasi Kompetensi	Sistem menampilkan formulir untuk memasukkan kompetensi yang sudah dimiliki dari daftar yang tersedia
UC-24	Memilih Optimasi dan Menghasilkan Jadwal	Sistem menampilkan menu untuk memilih pilihan optimasi jadwal

### 3.2.5 Diagram Sequence

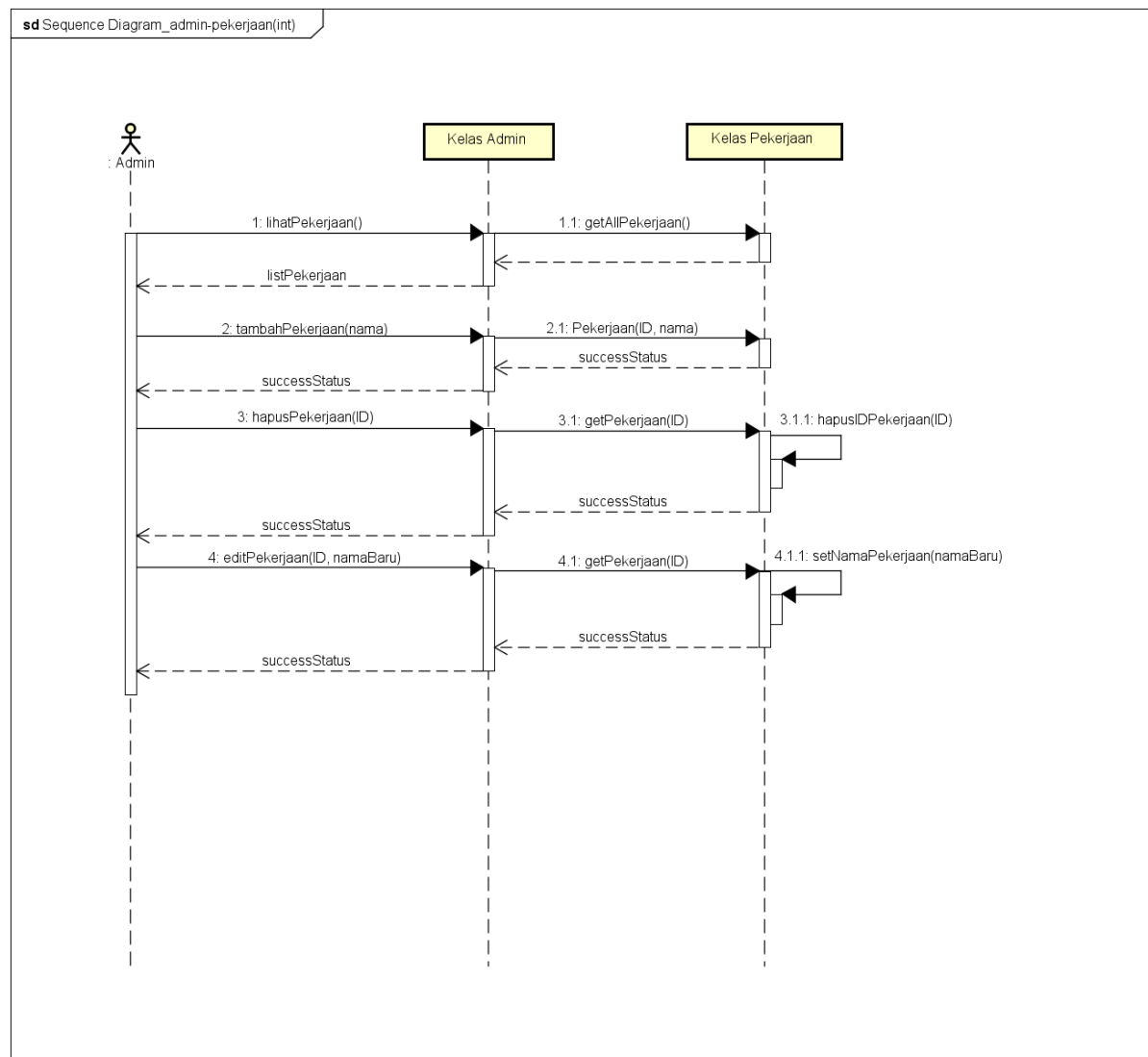
#### Role Admin



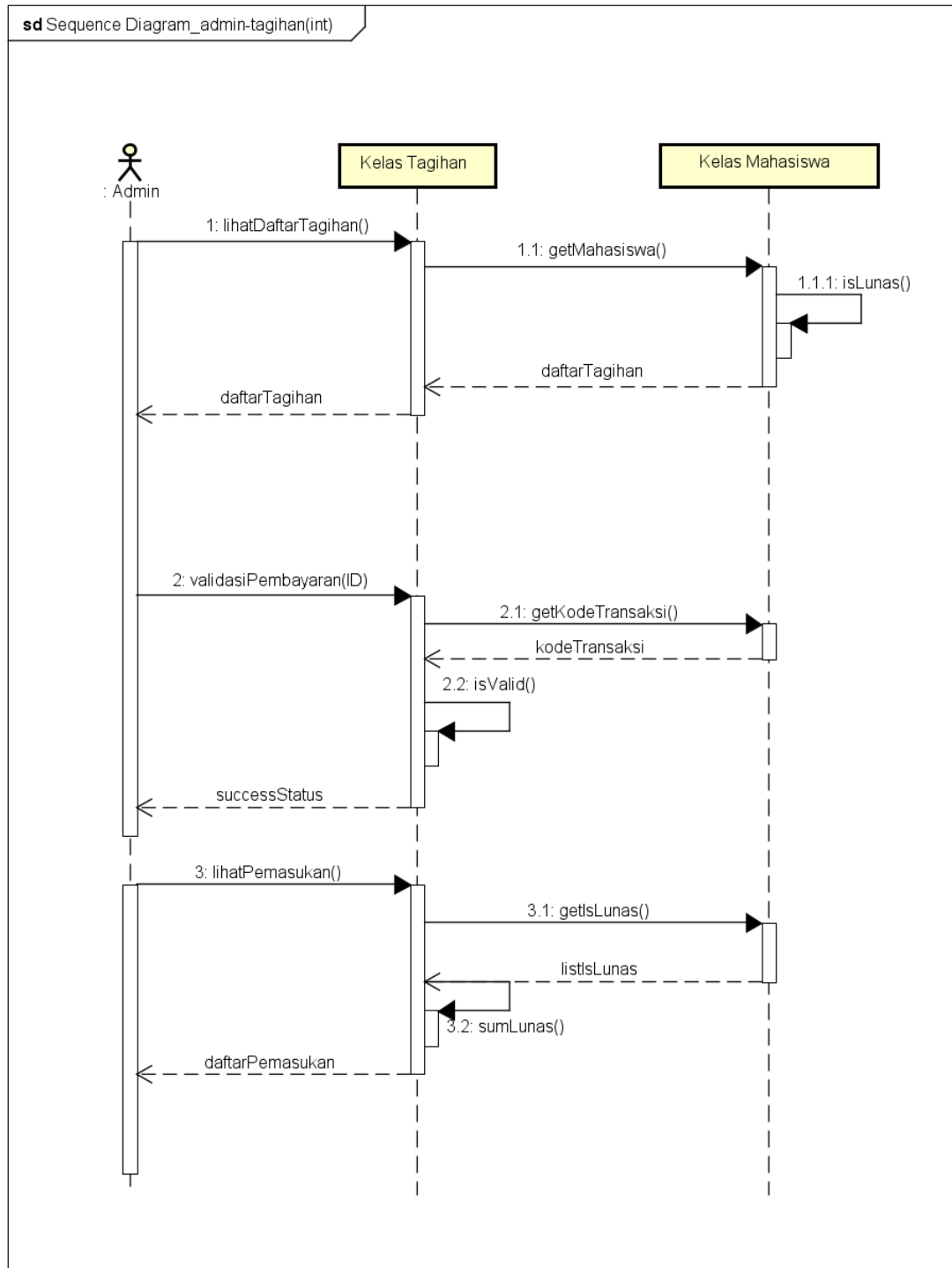
Gambar 3-3 Diagram Sequence untuk Admin



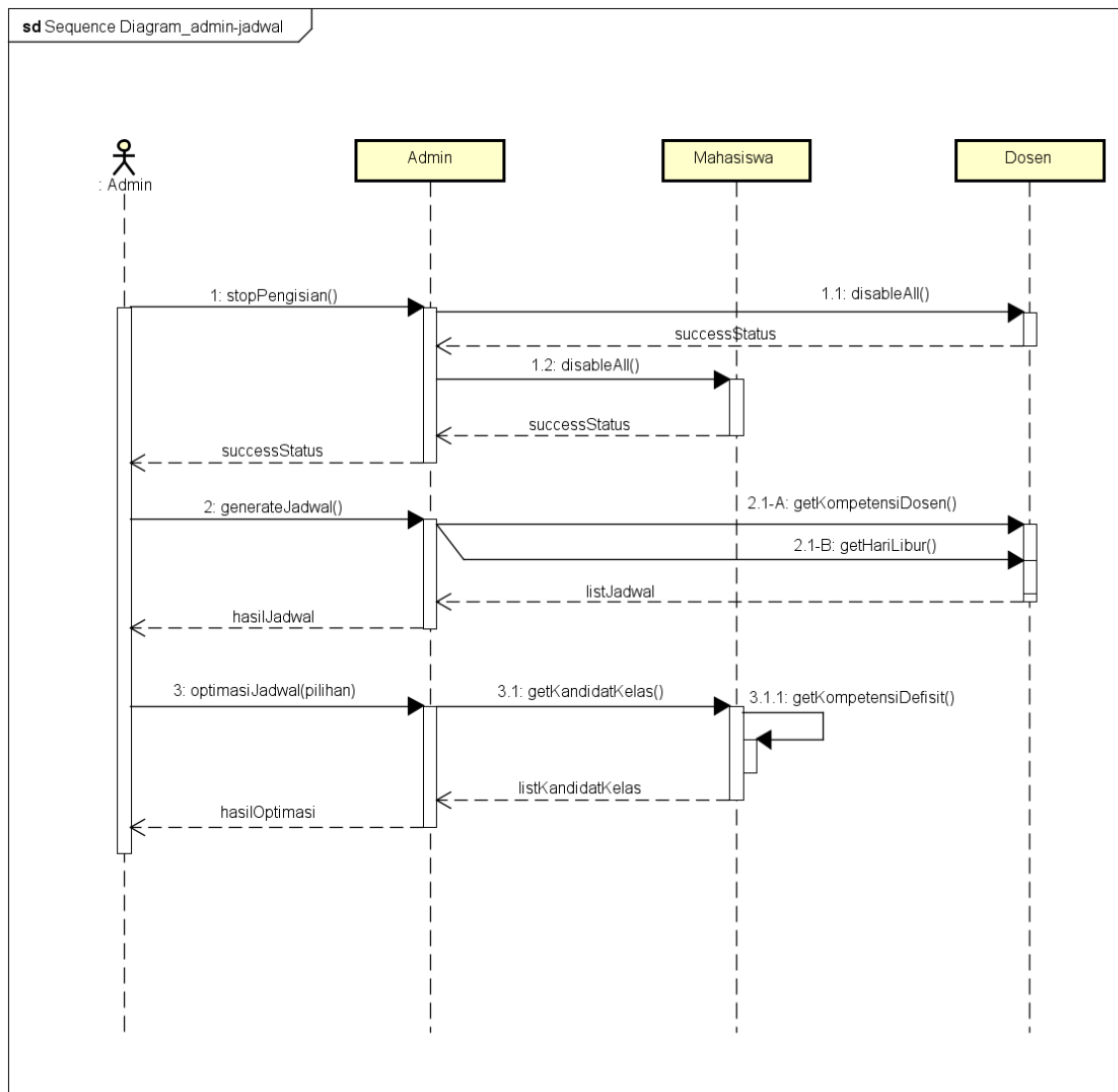
Gambar 3-4 Diagram *Sequence* untuk Admin (2)



Gambar 3-5 Diagram *Sequence* untuk Admin (3)



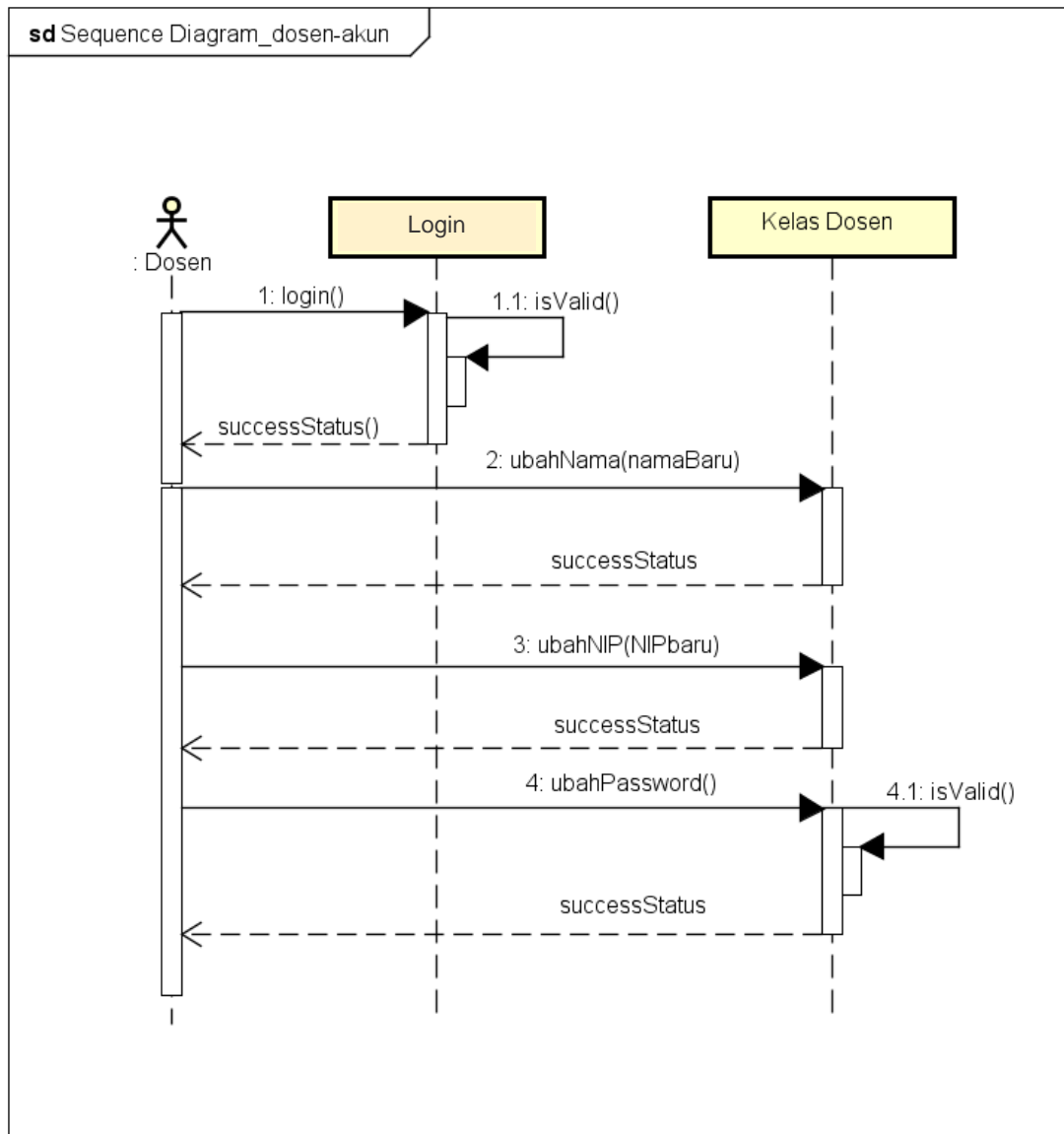
Gambar 3-6 Diagram *Sequence* untuk Admin (4)



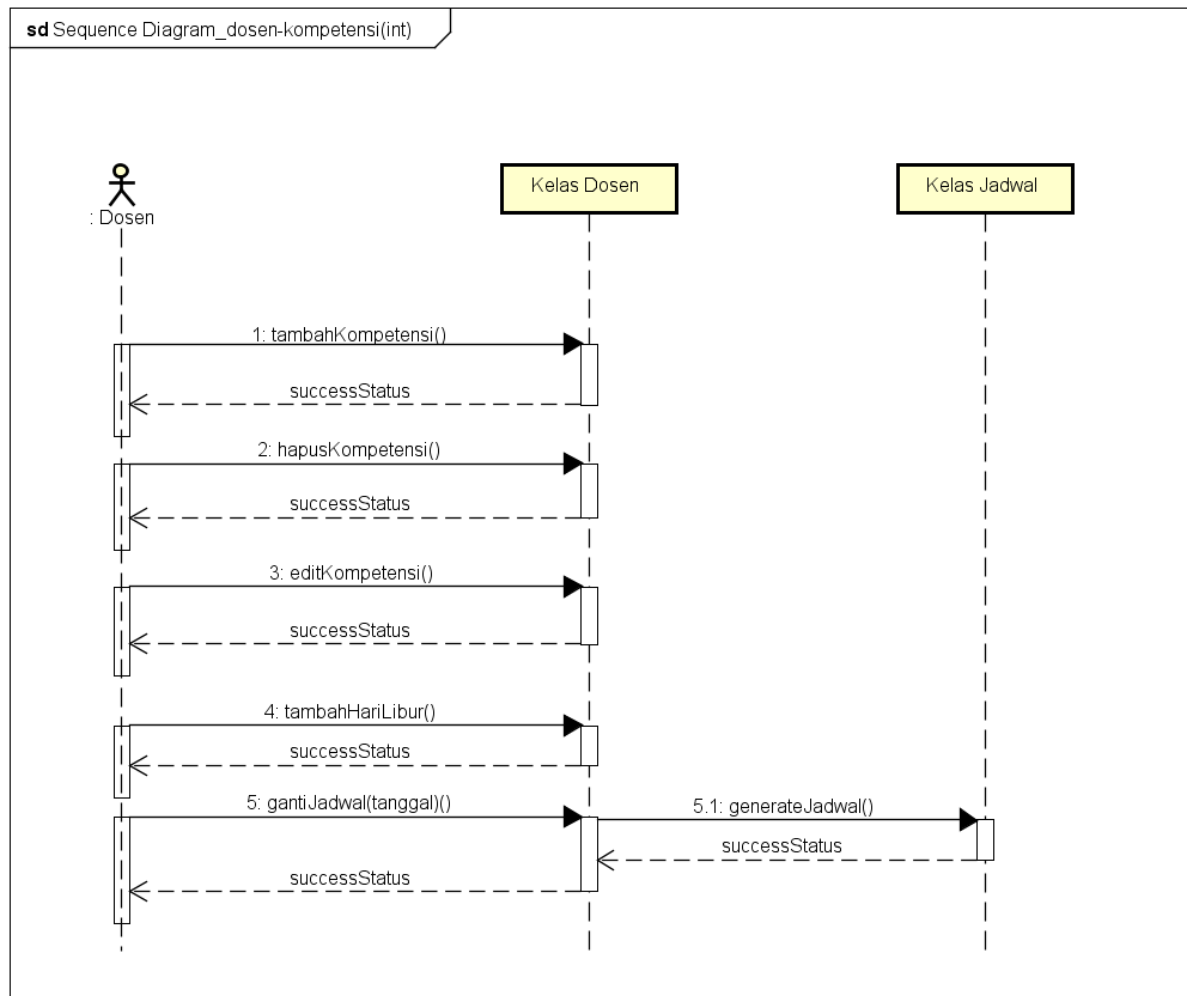
Gambar 3-7 Diagram *Sequence* untuk Admin (5)



## Role Dosen

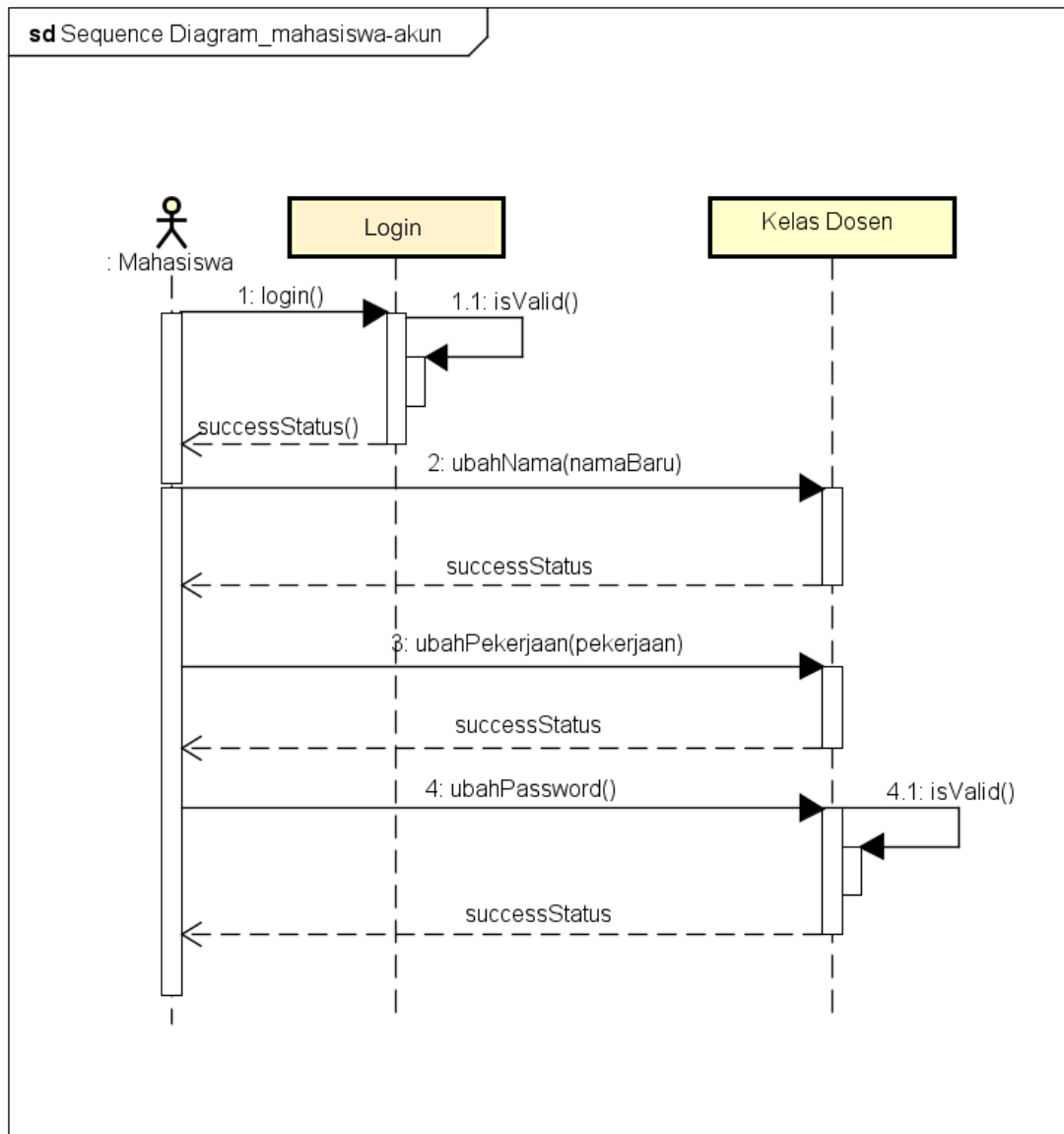


Gambar 3-8 Diagram *Sequence* untuk Dosen

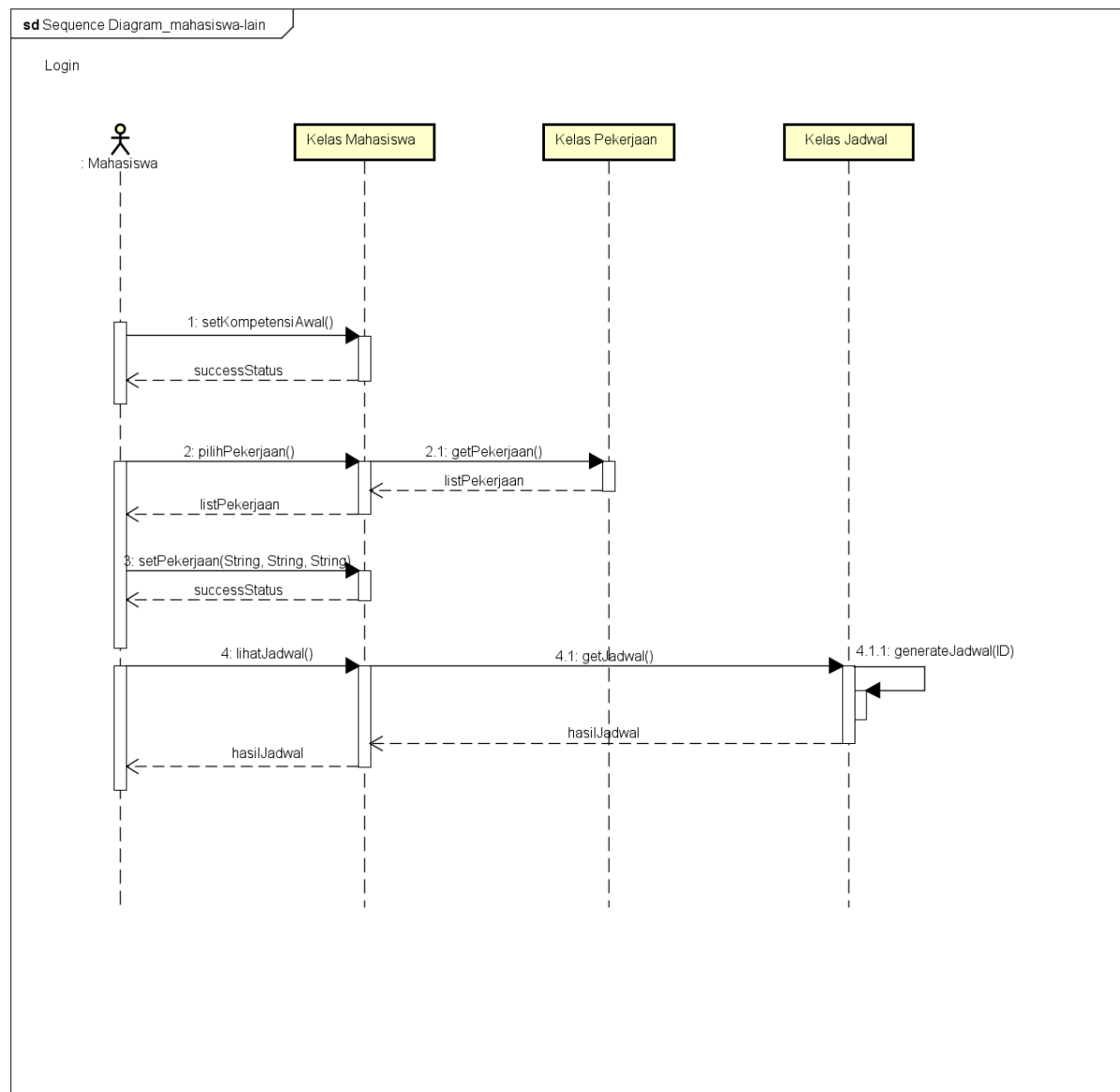


Gambar 3-9 Diagram *Sequence* untuk Dosen (2)

## Role Mahasiswa

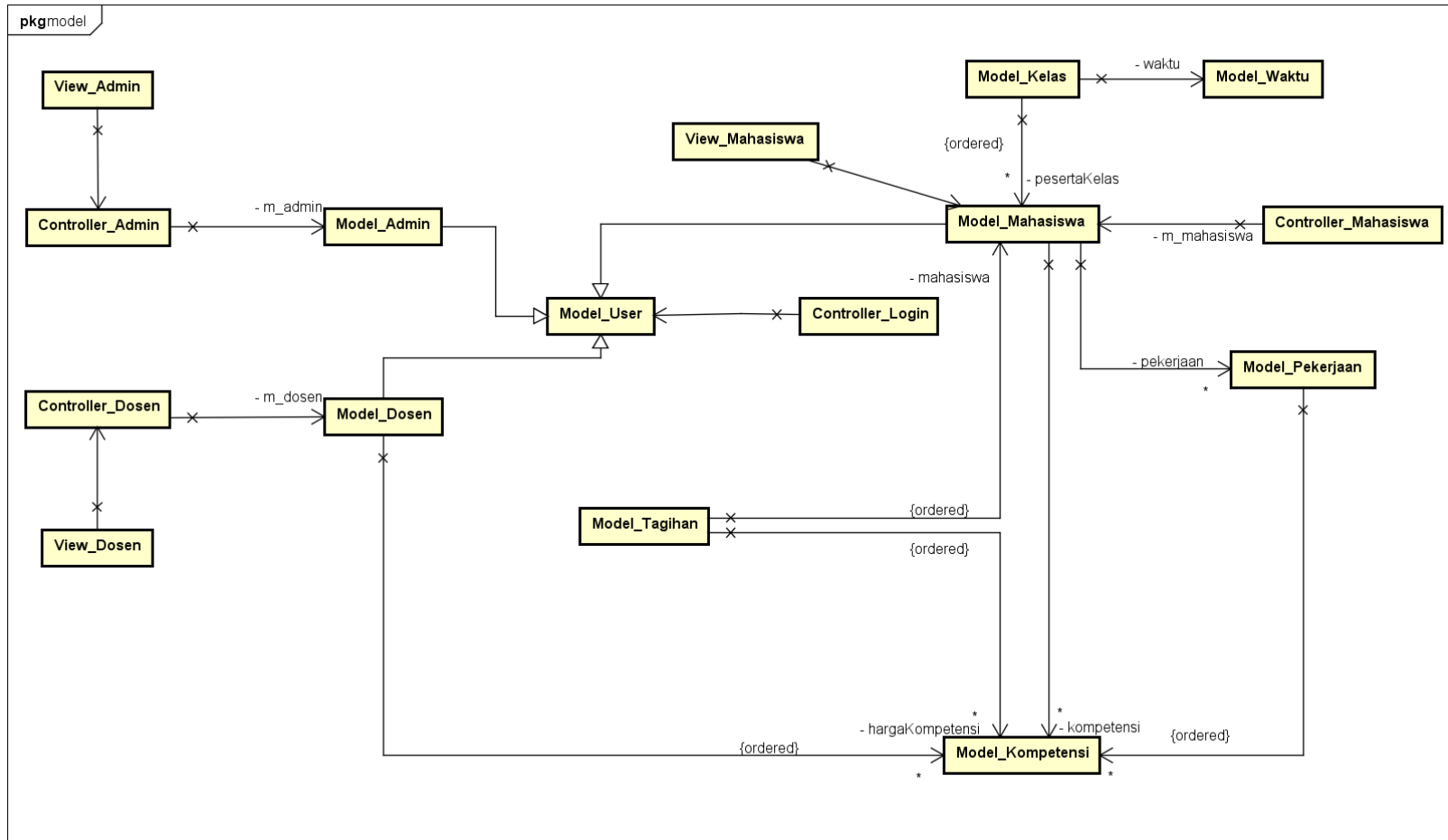


Gambar 3-10 Diagram *Sequence* untuk Mahasiswa



Gambar 3-11 Diagram *Sequence* untuk Mahasiswa (2)

### 3.2.6 Diagram Kelas



Gambar 3-12 Diagram Kelas

### 3.2.6.1 Detail Diagram Kelas Model

Model_User
- nama : String - username : String - password : String - role : int
+ Model_User() + Model_User(nama : String) + getUsername() : String + getPassword() : String + getNama() : String + setUsername(username : String) : void + setPassword(password : String) : void + setNama(nama : String) : void + getRole() : int + setRole(role : int) : void + jsonFileGetRole(username : String, password : String) : int + jsonFileSetPassword(password : String) : boolean + jsonFileSetNama(newNama : String) : boolean + login(role : int, username : String) : void + setIsLockedJson(isLocked : boolean) : void + getIsLockedJson() : boolean

Model_Admin
+ Model_Admin() + Model_Admin(nama : String) + tambahMahasiswa(nama : String, nim : String, kompetensi : int) : boolean + tambahDosen(nama : String, nip : String, kompetensi : int) : boolean + isUserExist(id : String) : boolean + hapusMahasiswa(nim : String) : boolean + hapusDosen(nip : String) : boolean + getAllDosen() : int + getAllMahasiswa() : int + getAdminFromJson(username : String) : Model_Admin + ubahKompetensiUser(username : String, kompetensi : int) : boolean + getAllKompetensi() : int + tambahKompetensi(idKompetensi : String, namaKompetensi : String, isPraktikum : String, daftarKompetensi : int, bobotKompetensi : int) : boolean + isKompetensiExist(id : String) : boolean + ubahNamaKompetensi(id : String, nama : String) : boolean + ubahBobotKompetensi(id : String, bobot : int) : boolean + ubahIsPraktikumKompetensi(id : String, isPraktikum : String) : boolean + ubahReqKompetensi(id : String, reqKompetensi : int) : boolean + hapusKompetensi(id : String) : boolean + hapusForeignKeyKompetensiDiKompetensi(id : String) : boolean + hapusForeignKeyKompetensiDiPekerjaan(id : String) : boolean + hapusForeignKeyKompetensiDiUser(id : String) : boolean + getAllPekerjaan() : int + tambahPekerjaan(id : String, nama : String, kompetensi : int) : boolean + isPekerjaanExist(id : String, nama : String) : boolean + ubahReqKompetensiUntukPekerjaan(id : String, reqKompetensi : int) : boolean + ubahNamaPekerjaan(id : String, nama : String) : boolean + hapusPekerjaan(id : String) : boolean + getAllKelas() : int + getAllTagihan() : int + validasiPembayaran(username : String) : boolean

<b>Model_Mahasiswa</b>
- nim : String
+ Model_Mahasiswa() + Model_Mahasiswa(nama : String, nim : String, kompetensi : int) + Model_Mahasiswa(nama : String, nim : String) + getNim() : String + setNim(nim : String) : void + getKompetensi() : int + setKompetensi(kompetensi : int) : void + getPekerjaan() : Model_Pekerjaan[] + setPekerjaan(pekerjaan : Model_Pekerjaan[]) : void + jsonFileSetNim(nim : String) : boolean + getMahasiswaFromJson(username : String) : Model_Mahasiswa + jsonFileSetKompetensiMahasiswa(kompetensi : int) : boolean + jsonFileSetPekerjaanMahasiswa(pekerjaan : String[]) : boolean + jsonFileSetPembayaran(kodeTransaksi : String, tanggalBayar : String) : boolean + getKelasByUsernameMahasiswa() : int + getTagihanMahasiswa() : Model_Tagihan

<b>Model_Dosen</b>
- nip : String - hariLibur : int
+ Model_Dosen() + Model_Dosen(nama : String, nip : String) + Model_Dosen(nama : String, nip : String, kompetensi : int) + getNip() : String + setNip(nip : String) : void + getKompetensi() : int + setKompetensi(kompetensi : int) : void + getHariLibur() : int + setHariLibur(hariLibur : int) : void + jsonFileSetNip(nip : String) : boolean + getDosenFromJson(username : String) : Model_Dosen + jsonFileSetKompetensiDosen(kompetensi : int) : boolean + jsonFileSetHariLiburDosen(hariLibur : int) : boolean + getKelasByUsernameDosen() : int

<b>Model_Pekerjaan</b>
- idPekerjaan : String - namaPekerjaan : String
+ Model_Pekerjaan() + Model_Pekerjaan(id : String) + Model_Pekerjaan(id : String, nama : String, mk : int) + getIdPekerjaan() : String + setIdPekerjaan(idPekerjaan : String) : void + getNamaPekerjaan() : String + setNamaPekerjaan(namaPekerjaan : String) : void + getDaftarKompetensi() : int + setDaftarKompetensi(daftarKompetensi : int) : void + getPekerjaanById() : Model_Pekerjaan

<b>Model_Kompetensi</b>
<ul style="list-style-type: none"> <li>- idKompetensi : String</li> <li>- namaKompetensi : String</li> <li>- isPraktikum : String</li> <li>- bobotKompetensi : int</li> </ul>
<ul style="list-style-type: none"> <li>+ Model_Kompetensi()</li> <li>+ Model_Kompetensi(id : String)</li> <li>+ Model_Kompetensi(idKompetensi : String, namaKompetensi : String, bobotKompetensi : int)</li> <li>+ Model_Kompetensi(idKompetensi : String, namaKompetensi : String, isPraktikum : String, bobotKompetensi : int)</li> <li>+ Model_Kompetensi(idKompetensi : String, namaKompetensi : String, isPraktikum : String, daftarKompetensi : int, bobotKompetensi : int)</li> <li>+ getIsPraktikum() : String</li> <li>+ setIsPraktikum(isPraktikum : String) : void</li> <li>+ getDaftarReqKompetensi() : int</li> <li>+ setDaftarReqKompetensi(daftarReqKompetensi : int) : void</li> <li>+ getIdKompetensi() : String</li> <li>+ setIdKompetensi(idKompetensi : String) : void</li> <li>+ getNamaKompetensi() : String</li> <li>+ setNamaKompetensi(namaKompetensi : String) : void</li> <li>+ getBobotKompetensi() : int</li> <li>+ setBobotKompetensi(bobotKompetensi : int) : void</li> <li>+ getKompetensiByID() : Model_Kompetensi</li> </ul>



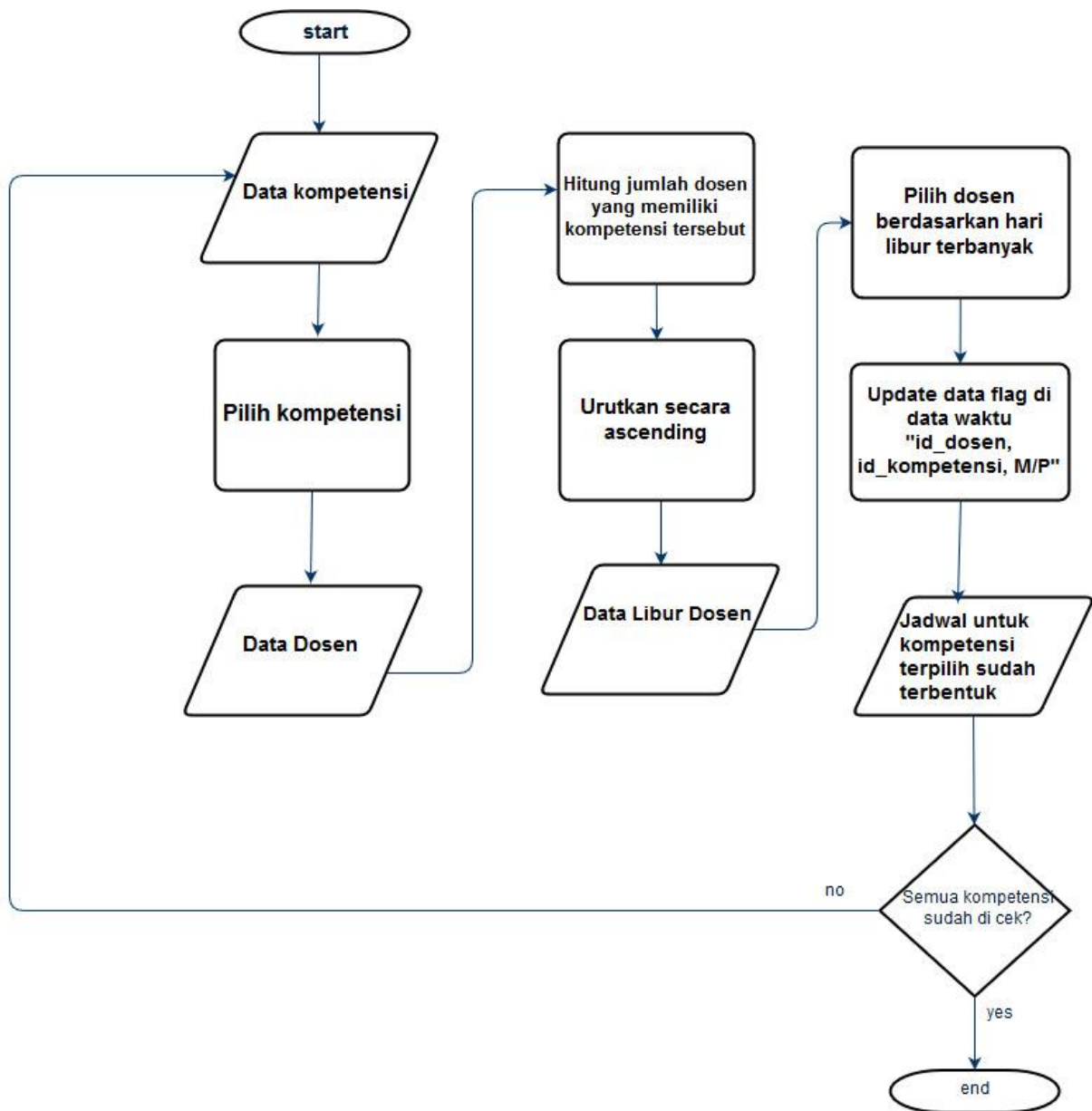
Model_Tagihan
<ul style="list-style-type: none"> <li>- totalTagihan : int</li> <li>- isLunas : boolean</li> <li>- kodeTransaksi : String</li> <li>- tanggalBayar : String</li> </ul>
<ul style="list-style-type: none"> <li>+ Model_Tagihan()</li> <li>+ Model_Tagihan(mahasiswa : Model_Mahasiswa, arrKelas : int)</li> <li>+ Model_Tagihan(mahasiswa : Model_Mahasiswa, arrKelas : int, totalTagihan : int)</li> <li>+ Model_Tagihan(mahasiswa : Model_Mahasiswa, arrKelas : int, totalTagihan : int, isLunas : boolean, kodeTransaksi : String, tanggalBayar : String)</li> <li>+ generateTagihanMahasiswaToJson(arrKelasMahasiswa : int, usernameMahasiswa : String) : void</li> <li>+ isUserExist(usernameMahasiswa : String) : boolean</li> <li>+ getMahasiswa() : Model_Mahasiswa</li> <li>+ setMahasiswa(mahasiswa : Model_Mahasiswa) : void</li> <li>+ getKelas() : int</li> <li>+ setKelas(kelas : int) : void</li> <li>+ getTotalTagihan() : int</li> <li>+ setTotalTagihan(totalTagihan : int) : void</li> <li>+ getIsLunas() : boolean</li> <li>+ setIsLunas(isLunas : boolean) : void</li> <li>+ getKodeTransaksi() : String</li> <li>+ setKodeTransaksi(kodeTransaksi : String) : void</li> <li>+ getTanggalBayar() : String</li> <li>+ setTanggalBayar(tanggalBayar : String) : void</li> <li>+ jsonFileSetIsLunas() : boolean</li> <li>+ jsonFileSetPembayaran(kodeTransaksi : String, tanggalBayar : String) : boolean</li> </ul>

Model_Waktu
<ul style="list-style-type: none"> <li>- idWaktu : String</li> <li>- Tanggal : String</li> <li>- Jam : String</li> <li>- flagOwner : int</li> </ul>
<ul style="list-style-type: none"> <li>+ Model_Waktu()</li> <li>+ Model_Waktu(idWaktu : String)</li> <li>+ Model_Waktu(idWaktu : String, Tanggal : String, Jam : String, flagOwner : int)</li> <li>+ getIdWaktu() : String</li> <li>+ setIdWaktu(idWaktu : String) : void</li> <li>+ getTanggal() : String</li> <li>+ setTanggal(Tanggal : String) : void</li> <li>+ getJam() : String</li> <li>+ setJam(Jam : String) : void</li> <li>+ getFlagOwner() : int</li> <li>+ setFlagOwner(flagOwner : int) : void</li> <li>+ getAllWaktu() : int</li> <li>+ jsonFileGetFlagOwner() : int</li> <li>+ jsonFileSetFlagOwner(newFlag : int) : boolean</li> <li>+ getWaktuById() : Model_Waktu</li> </ul>

Model_Kelas
<ul style="list-style-type: none"> <li>- idKelas : String</li> <li>- idKompetensi : String</li> <li>- idDosen : String</li> <li>- slotWaktu : int</li> <li>- kuotaKelas : int</li> </ul>
<ul style="list-style-type: none"> <li>+ Model_Kelas()</li> <li>+ Model_Kelas(id_kelas : String, id_kompetensi : String, id_dosen : String, slotWaktu : int)</li> <li>+ Model_Kelas(idKelas : String, idKompetensi : String, idDosen : String, slotWaktu : int, kuotaKelas : int)</li> <li>+ Model_Kelas(idKelas : String, idKompetensi : String, id_dosen : String, slotWaktu : int, kuotaKelas : int, pesertaKelas : int)</li> <li>+ getIdDosen() : String</li> <li>+ setIdDosen(id_dosen : String) : void</li> <li>+ getIdKelas() : String</li> <li>+ setIdKelas(id_kelas : String) : void</li> <li>+ getIdKompetensi() : String</li> <li>+ setIdKompetensi(id_kompetensi : String) : void</li> <li>+ getSlotWaktu() : int</li> <li>+ setSlotWaktu(slotWaktu : int) : void</li> <li>+ getKuotaKelas() : int</li> <li>+ setKuotaKelas(kuotaKelas : int) : void</li> <li>+ getPesertaKelas() : int</li> <li>+ setPesertaKelas(pesertaKelas : int) : void</li> <li>+ addKelasToJsonFile(kelas : Model_Kelas) : void</li> <li>+ addKelasFinalToJsonFile(kelas : Model_Kelas) : void</li> <li>+ isKelasExist(idKelas : String) : boolean</li> <li>+ isKelasFinalExist(idKelas : String) : boolean</li> <li>+ hitungKompetensiInDosen() : int</li> <li>- hitungKompetensiInDosenSort(data : int) : int</li> <li>+ hitungLiburInDosen(dosen : int) : int</li> <li>- hitungLiburInDosenSort(data : int) : int</li> <li>+ plotJadwal() : void</li> <li>+ optimasi1() : void</li> <li>+ optimasi2() : void</li> <li>- sortDefisitKompetensiMahasiswaAsc(data : int) : int</li> <li>- sortDefisitKompetensiMahasiswaDesc(data : int) : int</li> <li>- isJadwalMahasiswaTidakBentrok(usernameMahasiswa : String, idKompetensiBaru : String) : boolean</li> <li>- addPesertaKelasToJson(idKompetensi : String, usernameMahasiswa : String) : void</li> <li>- getPesertaKelasFromJson(idKompetensi : String) : int</li> <li>- getKuotaKelasFromJson(idKompetensi : String) : int</li> <li>- setKuotaKelasToJson(idKompetensi : String, kuota : int) : void</li> <li>- getSlotWaktuByIdKompetensiFromJson(idKompetensi : String) : int</li> <li>- getSlotWaktuKelasMilikMahasiswaFromJson(usernameMahasiswa : String) : int</li> <li>- deleteKelasYangTidakMemenuhiSyarat() : void</li> </ul>

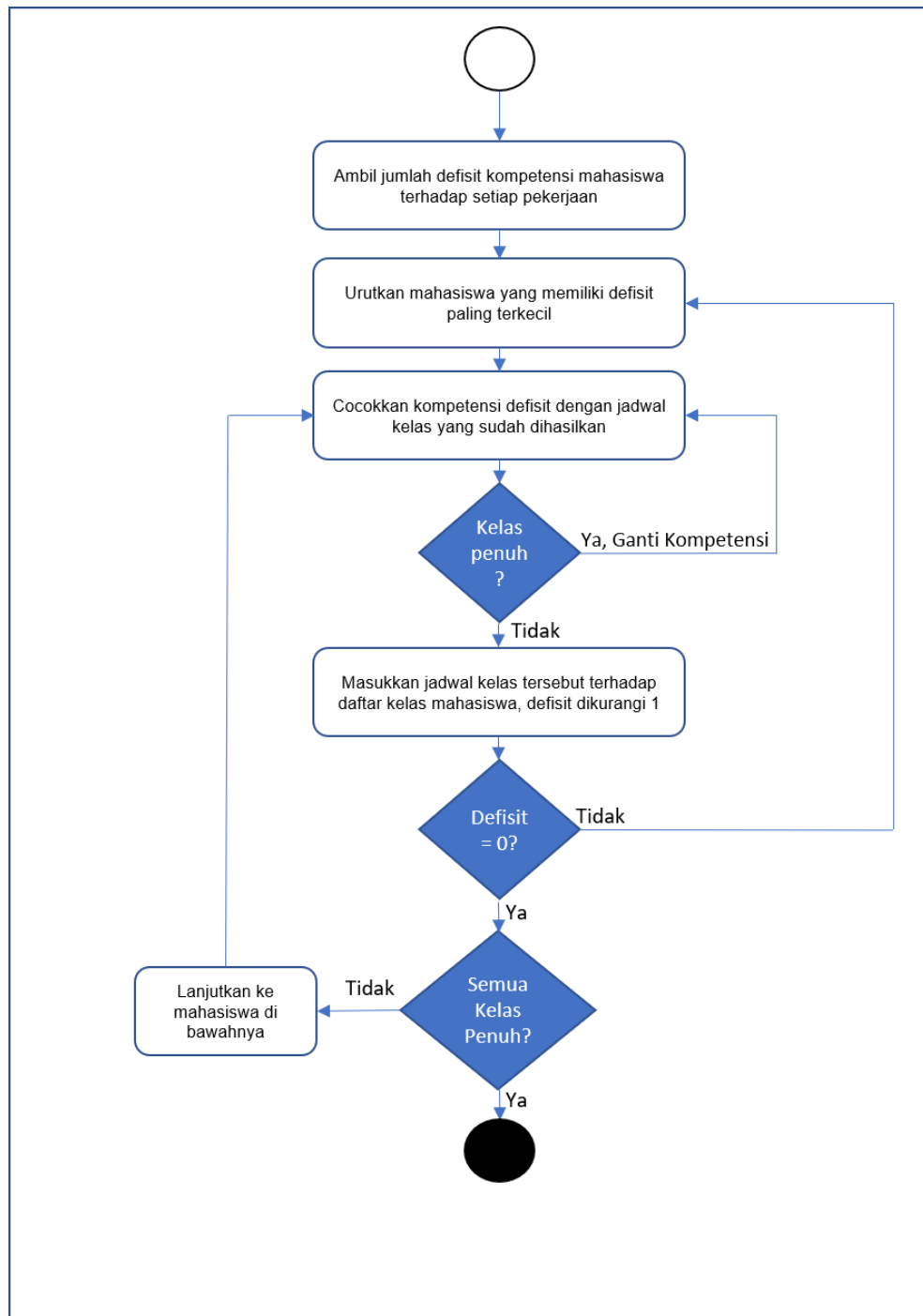
### 3.3 Flowchart Algoritma

#### 3.3.1 Algoritma Pembuatan Jadwal Kelas



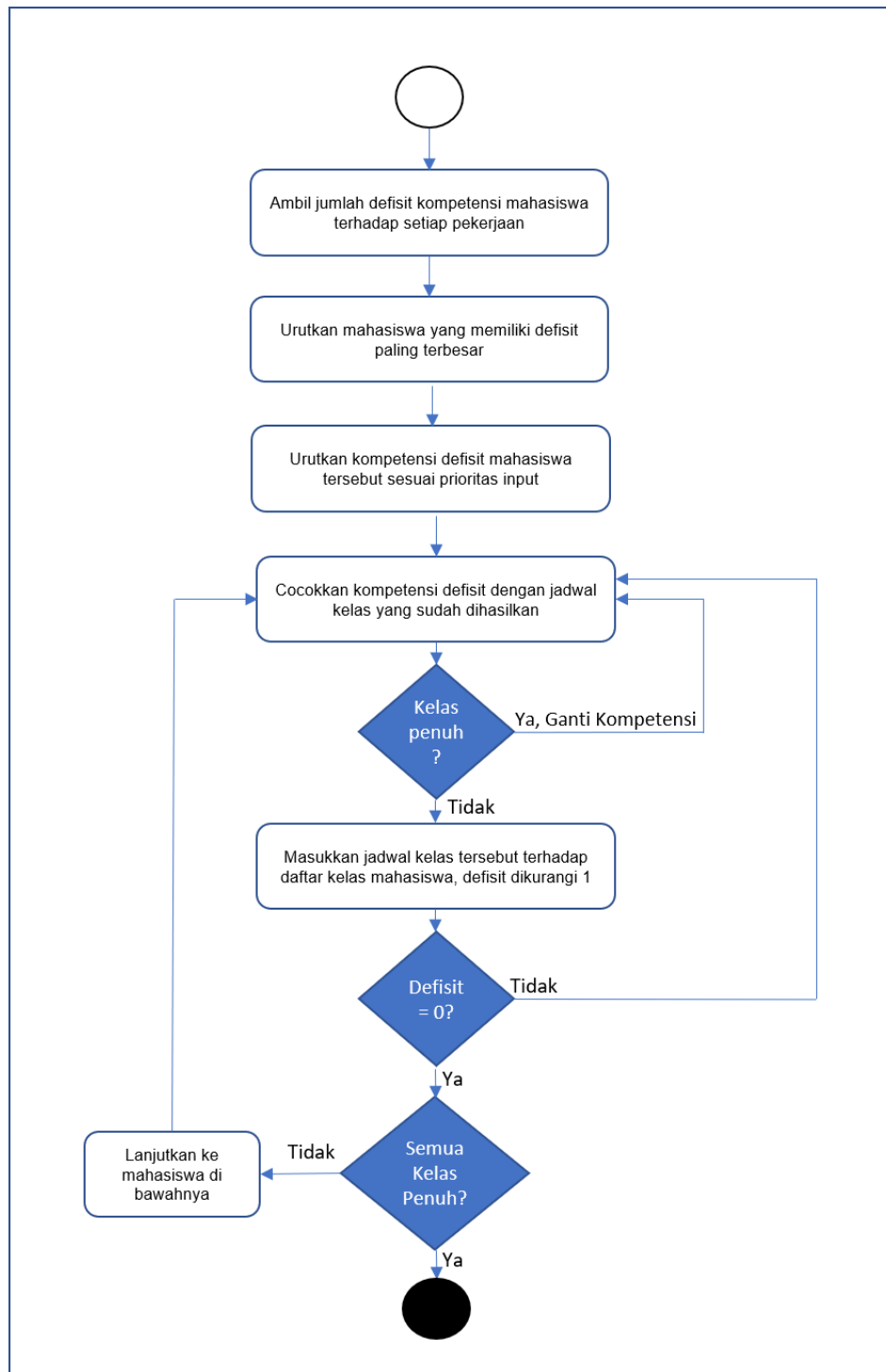
Gambar 3-13 Flowchart Algoritma Pembuatan Jadwal Kelas

### 3.3.2 Algoritma Alokasi Kelas Mahasiswa (Optimasi 1)



Gambar 3-14 *Flowchart* Algoritma (Optimasi 1)

### 3.3.3 Algoritma Alokasi Kelas Mahasiswa (Optimasi 2)



Gambar 3-15 Flowchart Algoritma (Optimasi 2)

## LAMPIRAN A: ACTIVITY LOG

Nama	Pembagian Tugas
Trisna Ari Roshinta 23518001	<ul style="list-style-type: none"><li>- CRUD kelas User</li><li>- Generate database waktu dan model waktu</li><li>- Generate Jadwal Kelas</li><li>- Optimasi 1</li><li>- Optimasi 2</li><li>- Menambahkan hari libur ke database waktu</li><li>- Memperbaiki fitur lihat tagihan mahasiswa</li><li>- Membuat fitur untuk menampilkan mahasiswa yang memenuhi kompetensi di pekerjaan cita-cita.</li><li>- Menambah kompetensi mahasiswa ketika mahasiswa tersebut sudah berhasil mengambil kelas</li></ul>
Atika Rahmawati Y. 23518001	<ul style="list-style-type: none"><li>- Diagram Aktivitas</li><li>- Diagram Use Case</li><li>- Diagram Sequence</li><li>- Diagram Kelas</li><li>- Laporan Dokumentasi</li><li>- Presentasi</li><li>- Algoritma Optimasi 1</li><li>- Algoritma Optimasi 2</li><li>- Algoritma Generate Jadwal</li><li>- Algoritma Penggantian Jadwal</li><li>- View dan Controller Tagihan dan Pemasukan</li></ul>
Liptia Venica 23518004	<ul style="list-style-type: none"><li>- CRUD kelas Kompetensi</li><li>- Diagram Use Case</li><li>- CRUD Kompetensi, Pekerjaan</li><li>- Membuat fitur menambahkan hari libur dosen</li><li>- Generate Jadwal Kelas</li><li>- Optimasi 1</li><li>- Optimasi 2</li><li>- Menyederhanakan struktur kode program untuk view, dibuat menjadi view per-user</li><li>- Membuat Fitur Logout</li><li>- Memperbaiki fitur lihat jadwal untuk setiap role</li><li>- Menambahkan fitur lock menu ketika optimasi telah dilakukan untuk menghentikan pengisian jadwal dan mencegah data diubah</li></ul>

Nama	Pembagian Tugas
Nurlaili Rizki H. 23518001	<ul style="list-style-type: none"> <li>- CRUD kelas Pekerjaan</li> <li>- Optimasi 1</li> <li>- Optimasi 2</li> <li>- Model Tagihan dan Pemasukan</li> <li>- Membuat fitur validasi pembayaran</li> <li>- Membuat fitur konfirmasi pembayaran</li> <li>- Membuat fitur mengganti jadwal</li> <li>- Membuat jadwal hasil revisi</li> <li>- Penggabungan kode program</li> </ul>

### Detail Timeline Pengerjaan

Tanggal	Kegiatan	
	Aktivitas Utama	Aktivitas Pendukung
9 November 2018	<ul style="list-style-type: none"> <li>• Menentukan alur, membuat diagram use case, diagram aktivitas</li> <li>• Pembagian tugas membuat <i>model, view, controller</i>.</li> </ul>	<ul style="list-style-type: none"> <li>• Melakukan diskusi pembagian tugas masing-masing melalui aplikasi pesan <i>whatsapp</i></li> <li>• Menyusun laporan</li> </ul>
10 – 14 November 2018	<ul style="list-style-type: none"> <li>• Pengerjaan kode program CRUD untuk kelas Pekerjaan, Kompetensi, dan User</li> </ul>	<ul style="list-style-type: none"> <li>• Menyusun laporan</li> <li>• Kordinasi melalui aplikasi pesan <i>whatsapp</i></li> </ul>
15 November 2018	<ul style="list-style-type: none"> <li>• Penggabungan kode program sehingga CRUD berjalan baik</li> </ul>	<ul style="list-style-type: none"> <li>• Kordinasi melalui aplikasi pesan <i>whatsapp</i></li> </ul>
19 November 2018	<ul style="list-style-type: none"> <li>• Pembahasan Algoritma pembuatan jadwal kelas</li> </ul>	<ul style="list-style-type: none"> <li>• Kordinasi melalui aplikasi pesan <i>whatsapp</i></li> </ul>
20 – 22 November 2018	<ul style="list-style-type: none"> <li>• Pembuatan diagram dan flow chart algoritma</li> <li>• Pembuatan program untuk pembuatan jadwal kelas</li> </ul>	<ul style="list-style-type: none"> <li>• Menyusun laporan dan presentasi</li> <li>• Kordinasi melalui aplikasi pesan <i>whatsapp</i></li> </ul>
23 – 24 November 2018	<ul style="list-style-type: none"> <li>• Pembuatan program untuk pembuatan jadwal kelas</li> </ul>	<ul style="list-style-type: none"> <li>• Kordinasi melalui aplikasi pesan <i>whatsapp</i></li> </ul>
25 – 28 November 2018	<ul style="list-style-type: none"> <li>• Pembuatan optimasi jadwal kelas</li> <li>• Pembuatan tagihan dan pemasukan</li> </ul>	<ul style="list-style-type: none"> <li>• Menyusun laporan dokumentasi lengkap</li> <li>• Kordinasi melalui aplikasi pesan <i>whatsapp</i></li> </ul>

Tanggal	Kegiatan	
	Aktivitas Utama	Aktivitas Pendukung
	<ul style="list-style-type: none"> <li>Pembuatan konfirmasi dan validasi pembayaran</li> </ul>	<ul style="list-style-type: none"> <li>Kordinasi tatap muka sesuai pembagian tugas</li> </ul>



## LAMPIRAN B: LEMBAR ASISTENSI

Form Asistensi Tugas Besar  
IF5021/Algoritma dan Pemrograman B  
Sem. 1 2018/2019

### ASISTENSI 1

No. Kelompok/Kelas : 01  
Tanggal : 15 November 2018

Anggota kelompok

	NIM / Nama	Tanda Tangan Kehadiran
1	Thina / 23518001	1 Kena
2	Abika / 23518003	2 Angl.
3	Lipta / 23518004	3 Jephew
4	Murtali / 23518019	4 Ruf
	NIM / Nama	Tanda Tangan Kehadiran
	Tori A.B. / 23517015	

Asisten pembimbing

Catatan Asistensi:

#### Rangkuman Diskusi

- Reset password (admin)
- mahasiswa ga ada ganti libur
- optimasi diganti alurnya. Jadwal kelas → ~~optimasi~~ pekerjaan → optimasi
  - ① mapping kompetensi ke pekerjaan, mapping deficit kompetensi terkecil. → ada pekerjaan yang full -
  - ② mapping kelas terbanyak → ngambil banyak kelas
- ganti jadwal harus sebelum kelas
- ganti jadwal harus nyesuaikan jadwal mahasiswa, min. 10 bus (asumsi).
- generate jadwal langsung 12 minggu.
- kelas punya atribut Jadwal. Mahasiswa punya array of kelas.

#### Tindak Lanjut