



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Fakultas Hukum Universitas Brawijaya (FH UB) salah fakultas yang ada di universitas Brawijaya. FH UB menjadi salah satu fakultas yang paling tua di Universitas Brawijaya yang telah berdiri sejak tahun 1957 yang semula bernama Perguruan Tinggi Hukum dan Pengetahuan Masyarakat (PTHMP). FH UB saat ini memiliki lima program studi yakni program sarjana, program magister ilmu hukum, program magister kenotariatan, program doktor program ilmu hukum dan program doktor ilmu hukum – Jakarta.

FH UB mempunyai 6 gedung dan mempunyai 285 ruangan. 26 ruangan diantaranya adalah ruangan perkuliahan untuk melaksanakan kegiatan perkuliahan setiap harinya. Kegiatan perkuliahan pada dasarnya harus menyajikan informasi jadwal perkuliahan bagi mahasiswa, dosen dan semua orang yang terlibat. Kemajuan yang sangat pesat dibidang teknologi informasi mendorong munculnya inovasi baru dalam penyajian dan pengolahan informasi untuk memenuhi kebutuhan informasi yang cepat dan akurat (Nugraha & Nugraha, 2018). Sementara saat ini penyajian informasi jadwal kuliah di FH UB masih belum dapat diakses dengan cepat dan akurat. Jadwal kuliah FH UB masih dimuat pada file pdf yang kemudian harus diunduh terlebih dahulu untuk dapat dilihat. Dalam file pdf, jadwal dituliskan dalam bentuk tabel. Penyampaian informasi jadwal kuliah dengan cara tersebut dinilai kurang bagus karena sulit melakukan pencarian sebab jadwal harus dicek dari kolom pertama baris pertama sampai baris terakhir, dilanjut kolom kedua baris pertama sampai baris terakhir dan begitu seterusnya sampai kolom terakhir.

FH UB mengupayakan proses pelaksanaan kegiatan perkuliahan berjalan dengan lancar sesuai dengan jadwal perkuliahan yang telah disusun. Namun kelancaran proses perkuliahan sering diganggu oleh hari libur dan dosen sedang berhalangan. Hari libur atau dosen sedang berhalangan membuat kelas menjadi kosong dan meniadakan perkuliahan untuk sementara. Untuk mengganti pertemuan perkuliahan yang ditiadakan maka jadwal pengganti harus dibuat. Pergantian jadwal mengharuskan mahasiswa atau dosen harus melakukan booking ruangan kosong untuk tempat dilaksanakannya jadwal pengganti. Proses menentukan jadwal pengganti biasanya akan melibatkan semua peserta kuliah dan dosen yang kemudian dirundingkan bersama untuk menentukan hari, waktu dan ruangan yang akan digunakan. Proses booking ruangan dinilai masih sulit dilakukan karena mahasiswa atau dosen tidak dapat mengetahui ruangan-ruangan mana yang benar benar kosong. Hal ini karena pemetaan ruangan kosong dan berisi tidak dilakukan. Kehadiran sebuah media untuk memetakan ruangan kosong atau berisi akan sangat membantu dalam mengurus jadwal pengganti.

Selain untuk mengurus jadwal kuliah pengganti, ruangan juga dipinjam untuk kegiatan-kegiatan lain diluar perkuliahan. Kegiatan ini mengharuskan mahasiswa atau dosen untuk melakukan peminjaman ruangan. Prosedur peminjaman



ruangan untuk kegiatan non perkuliahan juga masih tergolong manual. Hal ini karena peminjam harus mengajukan surat permohonan ke petugas mengisi formulir peminjaman dan kemudian menghadap ke kasubag umum dan kasubag akademik untuk mendapatkan izin peminjaman dan tanda tangan. Hal ini dapat menjadi kendala saat orang-orang tersebut tidak ada, sedang rapat atau sedang diluar kota. Peminjam diharuskan untuk bersabar menunggu untuk mendapat tanda tangan dari orang-orang tersebut. Dengan perkembangan teknologi yang sangat cepat, pemakaian komputer sangat membantu dan mempermudah menyelesaikan pekerjaan yang semula dikerjakan secara manual (Nurwarsito, 2009). Proses peminjaman harusnya dapat dipermudah lagi bilamana peminjaman dapat divalidasi oleh kasubag secara online dimana peminjam tidak perlu menghadap lagi secara langsung.

Kegiatan non perkuliahan biasanya juga akan menggunakan sarana prasarana milik fakultas. Peminjaman barang atau sarana prasarana juga mengalami kendala yang sama dengan permasalahan peminjaman ruangan. Kendala lain yang ditemui dalam peminjaman barang adalah sulitnya untuk mengetahui posisi atau status barang sedang dipinjam atau tidak. Cara untuk mengetahui adalah petugas harus mengecek ketersediaan barang terlebih dahulu ke gudang.

Setiap ruangan perkuliahan biasanya dilengkapi dengan sarana prasarana pendukung seperti sebuah proyektor, ac, lampu, kursi dan meja. Selama proses perkuliahan tidak menutup kemungkinan sarana prasarana tersebut tidak berfungsi normal atau rusak. Saat terjadi kerusakan tersebut maka harus segera dilaporkan dan segera ditanggapi agar tidak mengganggu proses perkuliahan. Penyampaian keluhan pada era kemajuan teknologi saat ini tidak mengharuskan untuk menghadap langsung ke petugas. Penyampaian keluhan dapat disampaikan secara *online* melalui media online atau lebih sering disebut *e-complaint*.

Kehadiran sebuah sistem yang mampu melakukan manajemen peminjaman sarana dan prasarana di FH UB akan sangat berguna. Sistem yang mampu menyajikan informasi jadwal kuliah yang lebih baik. Mampu menyajikan informasi jadwal kuliah yang sudah dipetakan sehingga dapat dilihat lebih cepat dengan tersedianya filter jadwal. Sistem yang mampu memetakan status ketersediaan ruangan kosong atau berisi dan status ketersediaan barang atau sarana prasarana dapat dipinjam, sedang dipinjam atau tidak. Adanya sistem juga akan mampu merekam dan memberikan pelaporan terkait kegiatan peminjaman ruangan dan barang. Penyampaian keluhan secara *online* juga akan membuat setiap keluhan dapat direkam dengan baik.

Berdasarkan penjelasan di atas maka penulis berniat untuk melakukan penelitian yang berjudul "Pengembangan Sistem Manajemen Peminjaman Sarana dan Prasarana (Studi Kasus : Fakultas Hukum Universitas Brawijaya)". Sistem dibangun berbasis website dengan harapan dapat mengatasi permasalahan yang sudah dijelaskan.



1.2 Rumusan Masalah

Berdasarkan pada permasalahan yang telah dijelaskan pada bagian latar belakang, maka disusunlah rumusan masalah penelitian sebagai berikut:

1. Bagaimanakah hasil analisis dan perancangan “Pengembangan Sistem Manajemen Peminjaman Sarana dan Prasarana (Studi Kasus : Fakultas Hukum Universitas Brawijaya)” ?
2. Bagaimanakah hasil implementasi “Pengembangan Sistem Manajemen Peminjaman Sarana dan Prasarana (Studi Kasus : Fakultas Hukum Universitas Brawijaya)” ?
3. Bagaimanakah hasil pengujian *white-box testing* dan *black-box testing* “Pengembangan Sistem Manajemen Peminjaman Sarana dan Prasarana (Studi Kasus : Fakultas Hukum Universitas Brawijaya)” ?

1.3 Tujuan

Adapun tujuan penelitian adalah sebagai berikut :

1. Menganalisis dan melakukan rancangan pada sistem manajemen peminjaman sarana dan prasarana (studi kasus : Fakultas Hukum Universitas Brawijaya)
2. Mengimplementasikan hasil rancangan sistem manajemen peminjaman sarana dan prasarana (studi kasus : Fakultas Hukum Universitas Brawijaya)
3. Menguji sistem manajemen peminjaman sarana dan prasarana (studi kasus : Fakultas Hukum Universitas Brawijaya) menggunakan metode pengujian *white-box testing* dan *black-box testing*.

1.4 Manfaat

Penelitian ini diharapkan mampu menangani masalah pemetaan ruangan, sarana dan prasarana di fakultas hukum universitas Brawijaya. Dengan pemetaan yang baik maka manajemen ruang, sarana dan prasarana dapat dilakukan sehingga peminjaman ruangan, sarana dan prasarana dapat mudah dilakukan kapan saja.

1.5 Sistematika Pembahasan

Sistematika penulisan pada laporan ini bertujuan untuk gambaran dan uraian dari sistem yang dibangun dalam secara garis besar dengan meliputi beberapa bab sebagai berikut:

BAB 1 PENDAHULUAN

Pada pendahuluan mendeskripsikan tentang latar belakang dari permasalahan yang diangkat, rumusan masalah, tujuan penelitian, dan manfaat penelitian, serta batasan masalah dan sistematika penulisan.

BAB 2 LANDASAN KEPUSTAKAAN



Pada bab ini berisikan tentang kajian terhadap tentang teori, konsep, model, metode, atau sistem dari pustaka ilmiah, yang berkaitan dengan tema, masalah, atau pertanyaan penelitian.

BAB 3 METODOLOGI PENELITIAN

Pada bab ini menjelaskan tentang tipe penelitian, strategi dan rancangan penelitian seperti metode metode yang digunakan selama penelitian, langkah langkah kerja yang dilakukan selama penelitian

BAB 4 ANALISIS KEBUTUHAN

Pada bab ini menjelaskan terkait analisis kebutuhan dan perancangan sistem yang akan dibangun.

BAB 5 IMPLEMENTASI

Pada bab ini menjelaskan terkait implementasi dari sistem yang akan dibangun

BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini menjelaskan hasil pengujian dan pembahasannya dengan sub bab analisis kebutuhan, perancangan dan pengujian sistem yang akan dibangun.

BAB 7 PENUTUP

Pada bab ini menjelaskan tentang kesimpulan dan saran. Kesimpulan ini diperoleh dari hasil dan pembahasan dalam pembangunan “Sistem Manajemen Sarana Prasarana di Fakultas Hukum Universitas Brawijaya”. Serta saran ini nantinya bisa digunakan dalam pengembangan selanjutnya.



BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Dalam melakukan penelitian ini, penulis mengacu pada penelitian terkait yang sudah pernah dilakukan sebelumnya. Penelitian pertama berjudul “Sistem Informasi Manajemen Ruang Kuliah Berbasis Piranti Bergerak” (Gustriansyah & Suhandi, 2014). Penelitian tersebut membahas mengenai pengembangan sebuah sistem yang mampu melakukan manajemen ruangan kuliah di Universitas Indo Global Mandiri yang sebelumnya dilakukan secara manual. Peneliti tersebut membangun sebuah sistem berbasis mobile yang dapat digunakan untuk mencari informasi ketersediaan ruangan dan melakukan reservasi ruangan. Pengembangan sistem manajemen ruang kuliah berbasis piranti bergerak oleh Gustriansyah dan Suhandi ini dibangun menggunakan model prototype. Sistem yang dibangun mendapat respon yang baik dari para penggunaannya. Dari survey yang dilakukan, sistem yang dibangun mendapat nilai rata-rata 4,07 dari 5 poin. Namun pada sistem yang telah dibangun masih terdapat kekurangan yang dapat diperbaiki lagi. Proses reservasi pada dasarnya harus disertai dengan validasi dari pihak tertentu. Namun sistem informasi manajemen ruang kuliah yang dibangun tidak menyertakan validasi, sehingga setiap reservasi yang diajukan tepat sasaran dan tidak sembarangan digunakan untuk tujuan lain. Selain tidak ada validasi, sistem yang dibangun juga tidak menyertakan laporan atau rekap reservasi. Pelaporan dari reservasi perlu dibuat agar pimpinan dapat mengetahui rekap penggunaan ruangan di universitas Global Mandiri.

Penelitian kedua yang menjadi acuan penulis berjudul “Aplikasi Pengelolaan dan Peminjaman Ruang di Fakultas Ilmu Terapan, Universitas Telkom” (Annisa, 2017). Penelitian kedua ini membahas mengenai pembangunan sebuah aplikasi pengelolaan dan peminjaman ruangan di Fakultas Ilmu Terapan Universitas Telkom. Aplikasi yang dibangun untuk dapat membantu pengguna dalam meminjam ruangan secara online dan memiliki fitur pelaporan peminjaman ruangan. Sistem yang dibangun pada penelitian kedua ini mendapat nilai lebih dari sistem yang dibangun pada penelitian pertama. Sistem dibangun dengan menggunakan model *waterfall*. Sistem yang dibangun pada penelitian kedua ini sudah menyertakan laporan atau rekap peminjaman yang dapat dicetak. Laporan ini nantinya dapat membantu pengelola mengetahui rekap penggunaan ruangan. Selain itu sistem ini sudah membedakan antara peminjaman untuk akademik dan non akademik. Peminjaman akademik untuk tujuan perkuliahan dan non akademik untuk tujuan non kegiatan mahasiswa. Selain itu sistem yang dibangun sudah mempunyai fungsi validasi untuk peminjaman yang telah diajukan. Peminjaman kemudian dapat disetujui atau ditolak secara online.

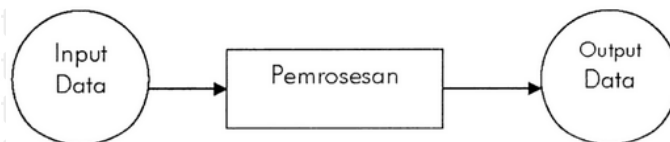
Penelitian ketiga yang menjadi acuan penulis berjudul “Sistem Informasi Manajemen Ruang (SIMERU) Kelas (Studi Kasus: FKTI Universitas Mulawarman)” (Khairina, Maharani, & Hatta, 2018). Penelitian ketiga ini membahas mengenai pembangunan sistem informasi yang dapat mengakomodir dan meminimalisir



adanya jadwal kuliah yang bertabrakan. Sistem dapat digunakan untuk reservasi ruang yang disertai dengan waktu dan tempat untuk menghindari jadwal yang berjalan pada tempat dan waktu yang sama. Sistem yang dibangun pada penelitian ketiga ini dibangun dengan pendekatan terstruktur. Sistem yang dibangun sudah dapat melakukan peminjaman ruangan atau reservasi ruangan namun masih mempunyai beberapa kekurangan. Pada sistem yang dibangun, setiap peminjaman masih belum disertai dengan proses validasi oleh pihak terkait. Sistem yang dibangun juga tidak membedakan antara peminjaman untuk kegiatan perkuliahan atau untuk keperluan kegiatan mahasiswa. Sistem yang dibangun juga tidak menyertakan laporan atau rekap peminjaman yang dapat dicetak.

2.2 Sistem Informasi

Sistem adalah suatu jaringan kerja dari prosedur -prosedur yang saling berhubungan, berkumpul Bersama untuk melakukan suatu kegiatan tertentu (Hutahaean, 2014). Informasi adalah data yang telah diproses atau data yang memiliki arti (Fatta, 2007). Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan laporan yang diperlukan (Sutabri, 2012). Sistem informasi manajemen adalah suatu alat yang menyajikan informasi dengan sedemikian rupa sehingga bermanfaat bagi penerimanya dengan tujuan untuk menyajikan informasi guna pengambilan keputusan pada pengendalian kegiatan operasi suatu perusahaan (Fatta, 2007). Adapun konsep sistem informasi dapat dilihat pada Gambar 2.1.



Gambar 2.1 Konsep Sistem Informasi

Sumber : Fatta (2007)

Penjelasan dari Gambar 2.1 adalah sebagai berikut.

1. Input data adalah masukan yang mewakili data yang masuk ke dalam sistem informasi. Input disini termasuk metode-metode media untuk menangkap data yang akan dimasukkan yang dapat berupa dokumen-dokumen dasar.
2. Output data adalah produk dari sistem informasi, informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkat manajemen serta semua pemakai sistem.
3. Pemrosesan adalah kegiatan yang menggunakan teknologi untuk menerima input, menyimpan dan mengakses data, menghasilkan sekaligus mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan (Fatta, 2007).

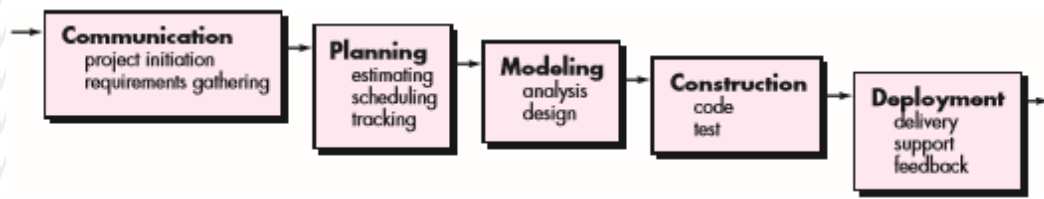


2.3 Software Development Life Cycle (SDLC)

Software development life cycle adalah sebuah metodologi yang digunakan untuk merancang, membangun, dan memelihara sebuah sistem. (Bahattab, Alshamrani, & Abdullah, 2015). Model SDLC saat ini sudah bermacam macam, seperti *Waterfall Model*, *V-Model*, *Spiral Model*, *Prototyping Model*, *Unified Software Development Process*, *Extreme Programming*, *Rapid Application Development* dan *Agile*.

2.4 Waterfall Model

Waterfall model atau *Classic Life Cycle* adalah model pengembangan perangkat lunak yang sistematis dan sekuensial yang dimulai dari perancangan, pemodelan, konstruksi, dan *deployment* (Pressman, 2010). Pengembangan perangkat lunak dilakukan tahap demi tahap, dimana setiap tahap pengembangan berlangsung berurutan tanpa ada tumpang tindih. Model pengembangan waterfall model dapat dilihat pada Gambar 2.1 berikut.



Gambar 2.1 *Waterfall Model*

Sumber : (Pressman, 2010)

Pada Gambar 2.1, Waterfall model mempunyai lima tahapan, yakni

1. **Communication**
Pada tahap ini dilakukan perjanjian antara klien dan pengembang untuk spesifikasi dan fitur perangkat lunak. Singkatnya, persyaratan dan kebutuhan dikumpulkan, dianalisis dan kemudian dokumentasi yang tepat disiapkan, yang membantu lebih lanjut dalam proses pengembangan sistem.
2. **Planning**
Pada tahap ini dilakukan penyusunan rencana pengerjaan dan perkiraan estimasi pengerjaan sistem dimulai dari tahap *modeling* sampai *deployment*.
3. **Modelling**
Tahap ini adalah tahap yang dilakukan untuk melakukan proses perencanaan dan pemecahan masalah untuk solusi perangkat lunak. Hal ini berkaitan dengan memilih desain algoritma yang sesuai, desain arsitektur perangkat lunak, database.



4. Construction

Pada fase ini dilakukan pengkodean hasil rancangan untuk menjadi sistem perangkat lunak menggunakan bahasa pemrograman.

5. Deployment

Setelah perangkat lunak sudah dirilis, mungkin perlu beberapa modifikasi, perbaikan, koreksi kesalahan, dan penyempurnaan yang sesuai.

Waterfall model dipilih dalam penelitian karena *waterfall model* adalah metode yang mudah diimplementasikan karena mempunyai model linier. Karena mempunyai model linier maka *waterfall* cocok digunakan pada pengembangan perangkat lunak yang mempunyai *requirement* yang sudah jelas. Fakultas Hukum UB yang sudah jelas mempunyai prosedur tertulis terkait dengan peminjaman ruangan dan barang dapat membuat proses mendapatkan *requirement* menjadi lebih mudah dan minim dari perubahan selama pengembangan sistem. Setiap fase dikerjakan satu per satu yang membuat waktu lama pengerjaan dapat dihitung (Balaji & Murugaiyan, 2012). *Waterfall* model juga digunakan karena sangat sederhana, tahapan didefinisikan dengan baik, dan mudah untuk diimplementasikan (Chandra, 2015).

2.5 Pendekatan Berorientasi Objek

Pendekatan berorientasi objek adalah sebuah metode pendekatan dalam pengembangan perangkat lunak yang memandang aplikasi perangkat lunak (*software*) sebagai kumpulan objek yang saling berinteraksi di dalam suatu sistem. Keuntungan yang diperoleh dari pendekatan object oriented ini adalah kemudahan untuk mengelola kompleksitas aplikasi yang dibuat dan kemudahan untuk melakukan perubahan dan pengembangan dari aplikasi. Bila terjadi kesalahan maka kesalahan tersebut dapat ditelusuri dan diperbaiki langsung pada class terkait tanpa mengakibatkan kesalahan baru pada class lain (Azis, 2005).

Pendekatan berorientasi objek terdiri dari tiga bagian yaitu Object Oriented Analysis (OOA), Object Oriented Design (OOD), Object Oriented Programming (OOP). Object Oriented Analysis (OOA) adalah tahapan untuk melakukan analisis kebutuhan dari sistem yang dibangun dengan menggunakan konsep yang berorientasi objek. Analisis kebutuhan menggunakan OOA harus dilakukan oleh orang yang memiliki pemahaman tentang sistem yang berorientasi objek, karena harus dipastikan setiap kebutuhan dapat diimplementasikan dengan berbasis objek. Object Oriented Design (OOD) adalah tahapan yang dilakukan untuk memetakan kebutuhan dari sistem yang dibangun dengan menggunakan konsep berorientasi objek ke dalam sebuah model desain. Setiap model desain harus dibuat agar mampu diimplementasikan dengan menggunakan pemrograman berorientasi objek. Model desain yang dibuat berorientasi objek dapat dibuat dengan menggunakan perangkat pemodelan berorientasi objek seperti UML (Unified Modelling Language) (A.S & Shalahuddin, 2015). Object Oriented Programming (OOP) adalah konsep yang memandang sebuah perangkat lunak sebagai kumpulan dari objek yang saling berinteraksi di dalam suatu sistem. Aplikasi dibangun dengan membagi fungsi-fungsi berdasarkan tugas tanggung



jadwal masing masing. Setiap tanggung jadwal ditetapkan pada setiap class. Setiap class menyediakan pelayanan untuk melakukan pekerjaan atau operasi tertentu. Operasi operasi ini kemudian dilakukan oleh objek-objek yang dibuat dari class tersebut (Azis, 2005).



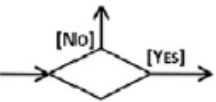
2.6 Unified Modelling Language (UML)

UML adalah bahasa pemodelan visual yang digunakan untuk menentukan, memvisualisasikan, mengontrak, dan mendokumentasikan artefak sistem perangkat lunak (James Rumbaugh, 1998). UML adalah hal pertama yang harus dibuat pengembang sebelum masuk tahap coding. UML nantinya akan dibuat untuk menerangkan model atau gambaran proses dari aplikasi yang akan dibuat sehingga nantinya dapat dijadikan acuan oleh pengembang pada saat implementasi menggunakan bahasa pemrograman tertentu. UML digunakan untuk pemodelan program berorientasi objek (OOP). Dalam penggunaannya UML yang biasa digunakan dalam pengembangan perangkat lunak adalah *Activity Diagram*, *Use Case Diagram*, *Sequence Diagram* dan *Class Diagram* (Pressman, 2010).

2.6.1 Activity Diagram

Activity diagram adalah diagram yang digunakan untuk membuat pemodelan dari beberapa objek untuk dapat saling bekerja sama. *Activity* dalam hal ini adalah *activities* yang digambarkan sebagai urutan perilaku subordinat, baik *action* atau aktivitas tingkat rendah yang terurai lebih lanjut, yang kemudian dihubungkan oleh aliran *control flow* dan *object flow*. Setiap perilaku dapat dimukai karena perilaku pendahulu atau sebuah peristiwa dari luar (Chonoles, 2017). *Activity diagram* menggunakan notasi dalam memodelkan sebuah aktivitas. Notasi tersebut dapat dilihat pada Tabel 2.1.

Tabel 2.1 Activity Diagram

No	Notasi	Simbol	Keterangan
1	Initial Node		Menandakan awal dari activity diagram
2	Final Node		Menandakan akhir dari sebuah aktivitas. Dalam hal ini activity masih dapat berlanjut jika masih ada proses lain yang masih berjalan
3	Decision Node		Notasi yang menandakan adanya sebuah keluaran berupa keputusan yang harus diambil dari flow yang masuk. Keputusan yang diambil hanya ada 2 yaitu ya atau tidak.



4	Merge Node		Notasi yang menandakan adanya gabungan dari beberapa flow yang masuk. Namun flow yang masuk hanya dapat keluar dari main edge
5	Action Roundangle		Notasi yang menandakan tindakan tindakan yang dilakukan pada activity
6	Control Flow		Notasi yang menerangkan aliran control dari sebuah node ke node lain

Sumber : (Chonoles, 2017)

2.6.2 Use Case Diagram

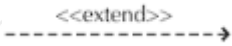

Use case diagram digunakan untuk menentukan fungsionalitas dan fitur perangkat lunak yang akan dibangun dari perspektif pengguna. Use case diagram menggambarkan bagaimana seorang pengguna berinteraksi dengan sistem, apa saja yang dapat dilakukan oleh pengguna saat berinteraksi dengan sistem (Pressman, 2010). Use Case digunakan untuk membuat daftar aktor dan kasus pengguna dan menunjukkan aktor mana yang berpartisipasi dalam setiap kasus penggunaan (James Rumbaugh, 1998).

Use case diagram dibuat dengan menggunakan notasi atau simbol pada Tabel 2.1 berikut:

Tabel 2.2 Simbol dan Deskripsi Use Case Diagram

No	Simbol	Deskripsi
1	Aktor 	Orang atau sistem yang mendapat manfaat dari perangkat dan bersifat eksternal terhadap subjek
2	Use Case 	Mewakili bagian utama dari fungsionalitas sistem
3	Assosiasi 	Menghubungkan Aktor dengan use case



4	Include 	Mewakili penyertaan fungsionalitas satu use case dalam yang lain
5	Extend 	Mewakili ekstensi use case untuk memasukkan perilaku opsional
6	Generalisasi 	Mewakili kasus penggunaan khusus untuk yang lebih umum

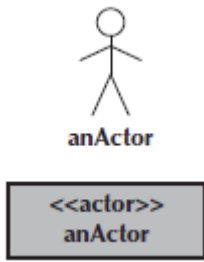

Sumber : (Cholones, 2017)

2.6.3 Sequence Diagram


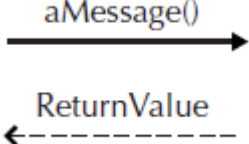
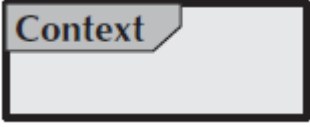
Sequence diagram berguna untuk menunjukkan urutan waktu, urutan proses yang terjadi dalam satu kasus penggunaan atau dalam satu skenario sistem perangkat lunak. *Sequence diagram* menunjukkan urutan pengiriman aksi, pemanggilan *method*, dan proses pembuatan dan pemanggilan objek (Pressman, 2010).

Sequence diagram dibuat dengan menggunakan notasi atau simbol pada Tabel 2.2 berikut:

Tabel 2.3 Simbol dan Deskripsi *Sequence Diagram*

No	Simbol	Deskripsi
1	Aktor 	Aktor dapat berupa seseorang atau juga sistem. Aktor adalah orang atau sistem yang berpartisipasi dalam <i>sequence</i> yang mengirimkan atau menerima pesan. Simbol orang digunakan jika aktornya adalah seseorang dan simbol persegi panjang jika aktor nya bukan manusia.
3	<i>LifeLine</i> 	Menunjukkan garis hidup suatu objek selama dalam <i>sequence</i>
	<i>Execution Occurence</i>	Sebuah persegi panjang berlebar sempit yang ditempatkan pada <i>life line</i> .



		Menunjukkan bahwa objek melakukan pengiriman atau penerimaan pesan
4	<p><i>Message</i></p> 	Menyampaikan informasi dari satu objek ke objek yang lain. Sebuah operasi diberikan label dengan pesan yang dikirim menggunakan panah penuh dan sebuah pesan return menggunakan symbol panah garis putus putus.
5	<p><i>Frame</i></p> 	Menunjukkan sebuah konteks yang terjadi dalam sequence diagram

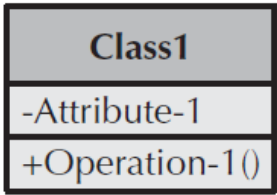
Sumber : (Dennis, Wixom, & Tegarden, 2015)

2.6.4 Class Diagram

Class diagram memberikan pandangan statis atau struktural dari suatu sistem. Kotak adalah elemen yang digunakan untuk mewakili kelas dan antar muka. Bagian atas class berisi nama kelas, bagian tengah berisi atribut-atribut kelas dan bagian bawah berisi operasi atau perilaku kelas (Pressman, 2010).

Class diagram dibuat dengan menggunakan notasi atau simbol pada Tabel 2.3 berikut:

Tabel 2.4 Simbol dan Deskripsi *Class Diagram*

No	Simbol	Deskripsi
1	<p><i>Class</i></p> 	<p><i>Class</i> memiliki nama yang diketik tebal, memiliki daftar atribut pada bagian tengah dan memiliki daftar operasi pada bagian bawah.</p> <p>Atribut merupakan properti yang menggambarkan sebuah keadaan dari objek.</p> <p>Operasi merupakan tindakan yang dapat dilakukan dalam sebuah <i>class</i></p>
2	<p>Asosiasi</p>	<p>Asosiasi merupakan hubungan antara banyak <i>class</i> dengan <i>class</i> lain atau dengan <i>class</i> itu sendiri.</p>



	<u>AssociatedWith</u> 0..* 1	
3	Generalisasi 	Merepresentasikan adanya semacam hubungan antara kelas yang satu dengan kelas yang lain. Sebuah kelas yang menjadi <i>sub class</i> akan mendapat pewarisan dari kelas lain yang menjadi <i>super class</i>
4	Agregasi 	Merupakan bentuk khusus dari asosiasi yang menandakan bahwa sebuah class adalah mempunyai hubungan a-part-of atau bagian dari <i>class</i> lain. Class yang memiliki hubungan ini dapat dibuat berdiri sendiri yang artinya, class tersebut tetap dapat berdiri meskipun class lain dihilangkan
5	Komposisi 	Merupakan bentuk khusus dari asosiasi yang menandakan bahwa sebuah class mempunyai hubungan a-part-of atau bagian dari class lain. Class yang memiliki hubungan ini tidak dapat berdiri sendiri. Hal ini karena setiap class harus saling terhubung agar dapat digunakan.

Sumber : (Dennis, Wixom, & Tegarden, 2015)

2.7 Hypertext Preprocessor (PHP)

PHP adalah salah satu bahasa pemrograman sederhana yang digunakan untuk melakukan pemrosesan HTML Form di dalam sebuah halaman *web*. PHP mempunyai struktur yang sederhana, hal ini membuat php dapat dengan mudah untuk dipelajari. Ketika dipanggil dari web browser, program yang ditulisdengan PHP akan di-*parsing* di dalam web server oleh *interpreter* PHP dan diterjemahkan ke dalam dokumen HTML, yang selanjutnya akan ditampilkan kembali ke *web browser* (Raharjo, 2016).

2.8 HyperText Markup Language (HTML)

HTML merupakan file text yang ditulis menggunakan aturan-aturan kode tertentu untuk kemudian disajikan ke user melalui suatu aplikasi web browser. Setiap informasi yang tampil di web selalu dibuat menggunakan kode HTML. Oleh karena itu, dokumen HTML sering disebut juga sebagai web page atau halaman web (Raharjo, 2016).

2.9 Cascading Style Sheets (CSS)

CSS adalah kumpulan kode yang digunakan untuk mendefenisikan desain dari bahasa markup. CSS digunakan untuk mengubah tampilan yang telah dituliskan



menggunakan kode HTML menjadi lebih cantik. Kode CSS dapat bekerja pada kode HTML dengan menggunakan mekanisme “tempel” pada kode HTML. (Enterprise, Pengenalan HTML dan CSS, 2016)

2.10 JavaScript

JavaScript merupakan bahasa pemrograman yang dipakai untuk membuat halaman website dapat berinteraksi dengan pengguna serta dapat memberikan respon event yang terjadi pada halaman. Javascript terintegrasi dengan HTML karena terkait dengan browser. Ketika sebuah browser memuat sebuah halaman website, server akan mengirim konten utuh dari dokumen, termasuk HTML dan statement javascript (Sianipar, 2015).

2.11 CodeIgniter

CodeIgniter adalah kerangka kerja PHP yang kuat dengan tapak yang sangat kecil, dibangun untuk pengembang yang membutuhkan *toolkit* sederhana dan elegan untuk membuat aplikasi web berfitur lengkap. Codeigniter memungkinkan pengembang untuk mengembangkan proyek lebih cepat dengan menyediakan serangkaian library yang biasanya dibutuhkan. Codeigniter memungkinkan pengerjaan proyek dengan meminimalkan jumlah kode yang diperlukan (CodeIgniter, 2019).

2.12 Bootstrap

Bootstrap digunakan untuk membangun tampilan dari situs web atau aplikasi web. Bootstrap merupakan framework paling populer di dunia untuk membangun tampilan website yang responsive dengan BootstrapCDN. Bootstrap adalah *toolkit* sumber terbuka untuk dikembangkan dengan HTML, CSS, dan JS (Bootstrap, 2019).

2.13 jQuery

jQuery adalah pustaka JavaScript yang cepat, kecil, dan kaya fitur. jQuery dapat membuat hal-hal seperti traversal dan manipulasi dokumen HTML, penanganan acara, animasi, dan Ajax jauh lebih sederhana dengan API yang mudah digunakan yang bekerja di banyak browser. Dengan kombinasi keserbagunaan dan ekstensibilitas, jQuery telah mengubah cara jutaan orang menulis JavaScript (jQuery, 2019).

2.14 DataTables

DataTables adalah plugin untuk pustaka Javascript jQuery. DataTables adalah alat yang sangat fleksibel, dibangun diatas dasar peningkatan progresif, yang menambahkan semua fitur canggih ini ke tabel HTML apa pun. (DataTables, 2019).



2.15 MySQL

MySQL digunakan untuk menyimpan data yang diperlukan oleh aplikasi (Raharjo, 2016). Mysql adalah salah satu database yang paling populer diantara database yang lain. MySQL mampu menangani data yang bwejumlah besar dan mampu mengirim dan menerima data dengan cepat (Komputer, 2010).

2.16 Pengujian Perangkat Lunak

Pengujian perangkat lunak adalah kegiatan yang dilakukan untuk memastikan suatu program dapat berjalan sesuai dengan apa yang diharapkan. Pengujian perangkat lunak juga ditujukan untuk dapat menemukan catat dalam sebuah program sebelum program tersebut digunakan. Pengujian perangkat lunak dilakukan untuk menerangkan kepada pengembang dan *costumer* bahwa perangkat lunak yang telah dibangun telah memenuhi semua *requirement*. Pengujian perangkat lunak dapat dilakukan dengan menggunakan beberapa metode pengujian seperti *white-box testing*, *black-box testing*, dan *compatibility testing* (Sommerville, 2011).

2.16.1 Black-Box Testing

Black-box testing adalah metode pengujian yang dilakukan untuk menguji perangkat lunak dimana penguji tidak memiliki akses terhadap kode program. Pengujian dilakukan dengan memperhatikan spesifikasi *requirement* dari sistem. Pengujian dengan menggunakan *black-box testing* harus memperhitungkan semua *requirement* sistem dapat berjalan dengan baik dan tidak gagal selama sistem digunakan. Pengujian ini dapat dilakukan dengan memasukkan nilai input dan memeriksa hasil output. Hasil output kemudian diperiksa apakah sudah sesuai sesuai dengan hasil yang diharapkan atau tidak (Sommerville, 2011).

2.16.2 White-Box Testing

White-box testing adalah sebuah metode pengujian yang digunakan untuk mengetahui tingkat cakupan program. Serangkaian tes harus dilakukan untuk memastikan bahwa program dapat melewati setiap jalur logis program saat dijalankan. Untuk melakukan pengujian dengan menggunakan *white-box testing* penguji dapat melihat stuktur program terlebih dahulu, setelah itu penguji dapat mengetahui *flow-graph* dari aliran program yang diuji. Penguji dapat melihat jumlah jalur yang ada pada *flow-graph* yang telah didapat. Penguji kemudian dapat memasukkan nilai input yang akan menyebabkan setiap jalur dapat dilewati (Sommerville, 2011).

2.16.3 Compatibility Testing

Compatibility testing adalah pengujian perangkat lunak untuk memastikan bahwa perangkat lunak yang dibangun dapat beroperasi baik jika dijalankan pada berbagai perangkat keras dan berbagai perangkat lunak. (Lonetti & Marchetti, 2017). *Compatibility testing* dapat dilakukan dengan menggunakan *tool sortsite*.

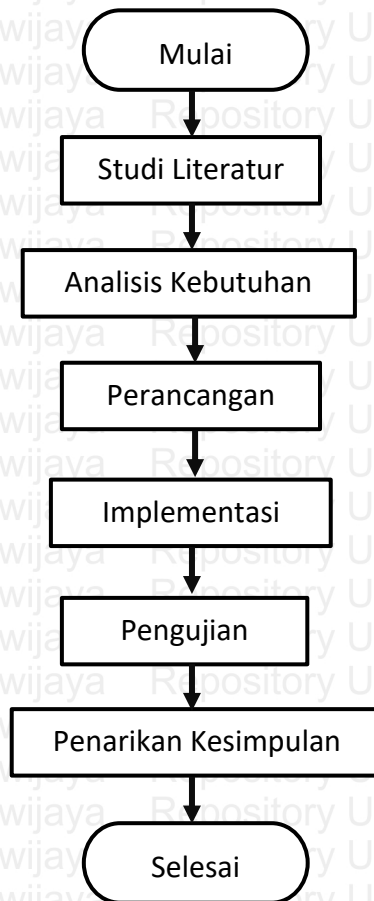


Tool ini dapat mengetahui tingkat *issue* dari sistem hanya dengan memasukkan alamat url sistem yang dibangun.



BAB 3 METODOLOGI PENELITIAN

Metodologi penelitian adalah langkah-langkah atau tahapan yang dilakukan selama proses penelitian. Pada penelitian ini, penulis menggunakan metode pengembangan perangkat lunak SDLC *waterfall model*. Adapun tahapan penelitian yang dilakukan adalah dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram Alir Metodologi Penelitian

3.1 Studi Literatur

Studi literatur adalah tahapan pertama dalam penelitian. Tahapan yang dilakukan untuk pengumpulan informasi dari berbagai sumber terpercaya yang memiliki hubungan dengan penelitian yang akan dilakukan. Pada tahapan studi literatur, penulis menggali dan mengumpulkan informasi terkait penelitian terlebih dahulu yang berkaitan dengan topik penelitian yang dilakukan. Penelitian terlebih dahulu dapat digunakan penulis sebagai pedoman dalam penelitian. dapat mengetahui kelebihan dan kekurangan dari penelitian terlebih dahulu sehingga dapat dijadikan acuan dalam melakukan penelitian yang lebih baik. Informasi dikumpulkan dari berbagai sumber seperti buku, jurnal, artikel online, dan laporan penelitian.



3.2 Analisis Kebutuhan

Analisis kebutuhan merupakan tahapan yang dilakukan untuk memperoleh informasi tentang spesifikasi sistem perangkat lunak yang akan dibangun. Dalam melakukan analisis kebutuhan, penulis melakukan wawancara kepada setiap pihak yang terlibat dalam kegiatan manajemen dan peminjaman ruangan dan barang. Penulis melakukan analisis kebutuhan dalam beberapa tahapan, yakni tahapan proses bisnis, elisitasi kebutuhan dan pemodelan kebutuhan.

3.2.1 Proses Bisnis

Proses bisnis adalah urutan aktivitas yang dilakukan dalam pekerjaan tertentu. Dalam mendapatkan informasi tentang proses bisnis, penulis melakukan wawancara dengan setiap pihak yang terlibat dalam kegiatan manajemen dan peminjaman ruangan dan barang. Selama melakukan wawancara, penulis juga mengemukakan artefak pendukung yang digunakan dalam kegiatan manajemen dan peminjaman ruangan dan barang. Setelah melakukan wawancara, penulis kemudian dapat mengetahui dan menggambarkan proses bisnis yang ada selama ini (*as is*). Dari proses bisnis *as is*, penulis kemudian dapat menyusun dan menggambarkan proses bisnis baru yang akan diterapkan (*to be*). Dari penjelasan diatas, maka hasil nantinya penulis dapatkan adalah informasi terkait prosedur manajemen dan peminjaman ruangan yang digambarkan dalam sebuah diagram *activity*.

3.2.2 Elisitasi Kebutuhan

Tahapan elisitasi kebutuhan adalah tahapan yang dilakukan untuk mengetahui informasi tentang kebutuhan-kebutuhan dari sistem perangkat lunak yang akan dibangun. Dalam tahapan elisitasi kebutuhan, penulis melakukan wawancara dengan setiap pihak yang terlibat dalam kegiatan manajemen dan peminjaman ruangan dan barang di FH UB. Dalam melakukan wawancara, penulis melakukan penggalian informasi lebih dalam terkait dengan kebutuhan-kebutuhan yang diperlukan dan diterapkan di sistem. Hasil yang nantinya didapatkan dalam tahapan analisis kebutuhan adalah penulis dapat mengetahui tentang gambaran umum sistem yang akan dibangun, dapat mengetahui siapa saja aktor yang terlibat dan dapat mengetahui apa saja kebutuhan dari sistem yang akan dibangun.

3.2.2.1 Gambaran Umum Sistem

Gambaran umum sistem adalah tahapan yang dilakukan untuk dapat mengetahui gambaran umum tentang sistem yang akan dibangun. Dalam melakukan tahapan ini, penulis melakukan analisis terkait dengan hasil wawancara yang dilakukan selama proses tahapan elisitasi kebutuhan. Hasil yang didapatkan dalam tahapan ini adalah penulis dapat memberikan penjelasan tentang gambaran umum sistem dan penjelasan fitur-fitur utama yang dapat dilakukan oleh sistem.



3.2.2.2 Identifikasi Aktor

Identifikasi aktor adalah tahapan yang dilakukan untuk mengetahui siapa saja aktor yang terlibat dalam penggunaan sistem beserta tugas masing masing. Dalam melakukan identifikasi aktor, penulis melakukan analisis berdasarkan hasil proses bisnis yang telah didapatkan sebelumnya. Penulis memperhatikan setiap proses bisnis, penulis melihat siapa siapa saja aktor yang terlibat dalam setiap proses bisnis. Penulis memperhatikan apa-apa saja tugas dari masing masing aktor yang terlibat dalam melakukan manajemen dan peminjaman ruangan dan barang di FH UB. Setelah melakukan hal tersebut, hasil yang didapatkan adalah penulis dapat merumuskan nama nama aktor yang terlibat di sistem beserta pembagian tugas masing masing.

3.2.2.3 Kebutuhan Sistem

Kebutuhan sistem adalah tahapan yang dilakukan untuk mengetahui apa saja kebutuhan yang harus disediakan oleh sistem. Dalam melakukan tahapan ini, penulis memperhatikan kembali proses bisnis yang telah didapatkan sebelumnya. Setiap proses yang ada dalam proses bisnis dapat menjadi sebuah kebutuhan. Penulis juga melakukan analisis setiap informasi yang didapatkan dalam proses elisitasi kebutuhan. Penulis dapat mendapatkan kebutuhan dengan memperhatikan setiap detail informasi yang didapatkan dalam proses elisitasi kebutuhan.

Kebutuhan sistem dibagi menjadi 2 jenis, kebutuhan fungsional dan kebutuhan non fungsional. Peneliti harus memperhatikan daftar kebutuhan dari hasil analisis ini harus dapat dilakukan, diukur, diuji, terkait dengan proses bisnis yang teridentifikasi, serta didefinisikan sampai tingkat detail yang memadai untuk desain sistem. Fitur-fitur yang telah diidentifikasi kemudian dispesifikasikan kedalam kalimat kebutuhan. Hasil yang didapatkan dalam proses ini adalah penulis mendapatkan kalimat kebutuhan dan spesifikasi kebutuhan beserta aktor.

3.2.3 Pemodelan Kebutuhan

Pemodelan kebutuhan adalah tahapan yang dilakukan untuk memberikan penjelasan tentang bagaimana hubungan interaksi antara aktor dengan masing masing kebutuhan yang ada di sistem. Dalam melakukan pemodelan kebutuhan, penulis dapat memodelkan kebutuhan dengan cara memperhatikan kembali hasil yang didapatkan dalam kalimat atau spesifikasi kebutuhan sistem. Penulis kemudian dapat memodelkan setiap kebutuhan yang telah didapatkan dan siapa aktor yang menjalankannya. Pemodelan setiap kebutuhan dengan aktor yang terlibat kemudian dapat digambarkan ke dalam sebuah *use case diagram*. Setelah mendapatkan model *use case diagram*, penulis kemudian dapat menjelaskan detail urutan langkah-langkah dalam mencapai tujuan dari sebuah *use case*. Urutan langkah didapatkan dengan memperhatikan kembali proses bisnis yang telah didapatkan pada tahapan sebelumnya. Langkah-langkah tersebut kemudian digambarkan ke dalam sebuah *use case scenario*. Hasil yang didapatkan dalam



pemodelan kebutuhan nantinya adalah berupa model kebutuhan yang dituangkan ke dalam *use case diagram* dan *use case scenario*.

3.3 Perancangan

Perancangan adalah tahapan yang dilakukan untuk merancang sistem sesuai dengan hasil analisis kebutuhan yang telah didapatkan. Perancangan sistem dilakukan dengan menggunakan perancangan arsitektur dan perancangan komponen.

3.3.1 Perancangan Arsitektur

Perancangan arsitektur adalah tahapan yang dilakukan untuk membuat rancangan arsitektur sistem. Dalam melakukan perancangan arsitektur, penulis memperhatikan kembali hasil pemodelan kebutuhan yang telah diperoleh dalam bentuk *use case diagram* dan *use case scenario*. Peneliti melakukan identifikasi terkait dengan objek yang ada dan interaksi yang terjadi antara objek yang satu dengan objek yang lain. Setelah melakukan identifikasi, peneliti kemudian dapat memodelkannya ke dalam sebuah *sequence diagram*.

Setiap objek yang telah diidentifikasi kemudian diperhatikan masing-masing karakteristik dan perilakunya. Setiap objek yang memiliki persamaan karakteristik kemudian dapat dibuat menjadi sebuah class. Setiap objek yang telah dikelompokkan berdasarkan persamaan karakteristik namun memiliki hubungan dengan objek lain yang berada dalam class yang berbeda kemudian dapat direlasikan dengan menggunakan hubungan antar class. Hubungan ini dapat berupa asosiasi, generalisasi, komposisi atau agregasi. Hasil pengelompokan class dan masing-masing relasinya dapat dimodelkan ke dalam sebuah *class diagram*.

3.3.2 Perancangan Komponen

Perancangan komponen adalah tahapan yang dilakukan untuk membuat rancangan komponen dari sistem yang akan dibangun. Perancangan komponen dilakukan dua jenis perancangan, yakni perancangan algoritma dan perancangan komponen.

3.3.2.1 Perancangan Algoritma

Perancangan algoritma adalah tahapan yang dilakukan untuk membuat rancangan alur logika dari program yang akan digunakan. Dalam melakukan perancangan algoritma, penulis merancang algoritma dengan memperhatikan kembali *sequence diagram* yang telah didapatkan. Method yang terdapat dalam *sequence diagram*, diperhatikan parameter input dan output. Penulis kemudian membuat rancangan algoritma untuk mendapatkan output yang diinginkan berdasarkan input yang ada tersedia. Rancangan algoritma tersebut digambarkan dalam bentuk *flow chart / pseudocode*.



3.3.2.2 Perancangan Antarmuka

Perancangan antarmuka adalah tahapan yang dilakukan untuk membuat rancangan tampilan antarmuka sistem yang akan dibangun. Dalam melakukan perancangan antarmuka, penulis memperhatikan hasil analisis kebutuhan yang telah didapatkan sebelumnya. Penulis menggunakan spesifikasi kebutuhan seperti detail data yang harus disimpan atau ditampilkan oleh sistem untuk merancang antarmuka sistem. Rancangan tersebut berupa gambar yang memuat bentuk formulir, halaman, tabel, tombol, ikon.

3.4 Implementasi

Implementasi adalah tahapan yang dilakukan untuk mengimplementasikan hasil rancangan menjadi sebuah sistem yang siap digunakan. Penulis melakukan implementasi dengan mengubah hasil rancangan menjadi sebuah sistem menggunakan bahasa pemrograman. Implementasi dimulai dengan membangun antarmuka atau *front end sistem* kemudian dilanjutkan dengan pemberian fungsi pada sistem. Sistem akan diimplementasikan menggunakan bahasa pemrograman PHP, HTML, CSS, JS dan MySql untuk sebagai *database* sistem. Untuk memudahkan implementasi maka penulis juga menggunakan kerangka kerja. Adapun kerangka kerja yang digunakan adalah *framework bootstrap* untuk implementasi tampilan sistem, *codeignitier* untuk *back end*. Selain itu penulis juga menggunakan plug-in bantuan *jQuery* dan *dataTable* untuk membuat sistem berjalan lebih baik. Hasil yang didapatkan dalam tahapan implementasi adalah sebuah sistem perangkat lunak yang sudah dapat digunakan, dengan catatan bahwa sistem masih perlu untuk dilakukan pengujian.

3.5 Pengujian

Tahap pengujian atau testing adalah tahapan yang dilakukan mengetahui apakah sistem yang dikembangkan sudah memenuhi semua persyaratan atau tidak. Dalam tahapan pengujian, peneliti mengoperasikan sistem yang telah diimplementasi menemukan sebanyak mungkin kesalahan yang ada sebelum digunakan oleh pengguna. Dalam penelitian ini penulis menggunakan tiga teknik pengujian, *white-box testing*, *black-box-testing* dan *compatibility-testing*.

3.5.1 White-box testing

White-box testing adalah teknik pengujian yang dilakukan untuk menguji stuktur dan alur kode program yang telah ditulis. Penulis melakukan pengujian white-box testing dengan cara menggunakan teknik *basis-path testing*. Teknik pengujian ini dilakukan dengan melihat langsung kode program yang telah dibuat. Penulis memperhatikan struktur alur program untuk dapat menyusun sebuah *flow graph*. *Flow graph* kemudian dapat digunakan untuk mengetahui jalur indenden dari program yang telah dibuat. Setiap jalur independent kemudian akan dijadikan kasus uji. Kasus uji yang telah didapatkan kemudian dilakukan pengecekan dengan memasukkan nilai input, apakah kasus uji sesuai dengan hasil yang diharapkan



atau tidak. Hasil pengujian yang akan didapatkan dari pengujian ini adalah untuk mengetahui apakah semua jalur independen dapat dilalui atau tidak.

3.5.2 *Black-box testing*

Black-box testing adalah teknik pengujian yang dilakukan untuk mengetahui apakah semua kebutuhan fungsional yang telah ditetapkan dapat berjalan baik atau tidak. *Black-box testing*, dilakukan dengan cara memeriksa kembali daftar kebutuhan sistem yang telah dapat dari tahapan analisis kebutuhan dengan apa yang sudah diimplementasikan. Setiap kebutuhan fungsional yang telah ditetapkan akan menjadi kasus uji. Peneliti melakukan pengujian pada setiap kasus uji dengan cara memasukkan nilai masukan, setelah nilai masukan diinput, penulis memperhatikan hasil akhir kasus uji. Pengecekan kemudian dilakukan dengan membandingkan hasil yang diharapkan dengan hasil akhir yang sebenarnya terjadi. Jika hasil yang diharapkan sama dengan hasil yang sebenarnya terjadi maka kasus uji tersebut berstatus valid. Hasil akhir yang didapatkan dalam melakukan *black-box testing* adalah mengetahui status valid dari kasus uji setiap kebutuhan fungsional yang telah ditetapkan.

3.5.3 *Compatibility testing*

Compatibility testing adalah Teknik pengujian untuk mengetahui apakah sebuah sistem perangkat lunak dapat berjalan pada lingkungan perangkat yang berbeda beda atau tidak. *Compatibility testing* dengan cara menggunakan aplikasi *sortsite*. Penulis menggunakan aplikasi *sortsite* untuk mengetahui tingkat *compatibility* dari sebuah sistem hanya dengan memasukkan alamat url sistem. Tool *sortsite* kemudian akan memberikan hasil pengujian dengan memberikan keterangan apakah terdapat *issue* atau tidak dari sistem yang diuji.

3.6 Penarikan Kesimpulan

Pengambilan kesimpulan merupakan tahapan dimana semua tahanan sebelumnya telah selesai dilakukan untuk ditarik sebuah kesimpulan berdasarkan penelitian yang dilakukan. Pada tahap ini, peneliti akan memberikan saran tentang pengalaman yang didapatkan selama proses penelitian.



BAB 4 ANALISIS KEBUTUHAN

Bab analisis kebutuhan ini menerangkan tentang gambaran umum sistem yang dibangun. Penjelasan dibagi kedalam dua bagian yakni penjelasan proses bisnis dan penjelasan analisis kebutuhan.

4.1 Proses Bisnis

Proses rekayasa kebutuhan dilakukan untuk menggali kebutuhan sistem yang dibangun. Untuk mengetahuinya maka proses bisnis atau proses bisnis harus diketahui lebih dalam. Hal ini agar sistem yang akan dibangun tidak merubah total prosedur yang telah ada sebelumnya. Prosedur atau proses bisnis terkait manajemen ruangan di FH UB dilakukan dengan melakukan wawancara kepada setiap pemangku kepentingan yang terlibat dalam proses manajemen sarana prasarana di FH UB.

4.1.1 Proses Bisnis Saat Ini

Berdasarkan hasil wawancara yang dilakukan dengan pihak Fakultas Hukum Universitas Brawijaya, maka penulis mendapatkan prosedur terkait kegiatan penggunaan sarana prasarana. Adapun kegiatannya adalah prosedur peminjaman ruangan untuk tujuan perkuliahan, peminjaman ruangan untuk kegiatan diluar perkuliahan, peminjaman barang, dan pengembalian barang.

4.1.1.1 Prosedur Peminjaman Ruangan Untuk Tujuan Perkuliahan

Berdasarkan hasil wawancara yang dilakukan dengan petugas yang terlibat dalam mengelola sarana prasarana di Fakultas Hukum Universitas Brawijaya, maka dapat digambarkan prosedur peminjaman ruangan untuk tujuan perkuliahan pada Gambar 4.1. Pada Gambar 4.1 dapat dilihat bahwa aktor yang terlibat ada 2 yakni mahasiswa / dosen sebagai pemohon dan sekretariat perkuliahan. Prosedur peminjaman ruangan untuk tujuan perkuliahan pada Gambar 4.1 diatas terlihat sederhana. Pertama pemohon cukup mengajukan peminjaman untuk tujuan perkuliahan ke sekretariat kuliah. Kedua sekretariat perkuliahan melakukan pengecekan ketersediaan ruangan. Jika ruangan tersedia untuk dipinjam maka mahasiswa/dosen dapat langsung menggunakannya. Proses bisnis peminjaman ruangan untuk tujuan perkuliahan saat ini memang sudah terlihat sederhana dan mudah diikuti. Namun perlu diingat bahwa prosedur tersebut masih dapat diperbaiki lagi. Mahasiswa atau dosen sebaiknya dapat langsung mengetahui status ketersediaan ruangan kosong atau tidak pada hari, tanggal dan jam tertentu. Dimana nantinya akan mempermudah mahasiswa atau dosen dalam menentukan jam dan ruangan tempat berlangsungnya jadwal kuliah pengganti.

4.1.1.2 Prosedur Peminjaman Ruangan Untuk Tujuan Non Perkuliahan

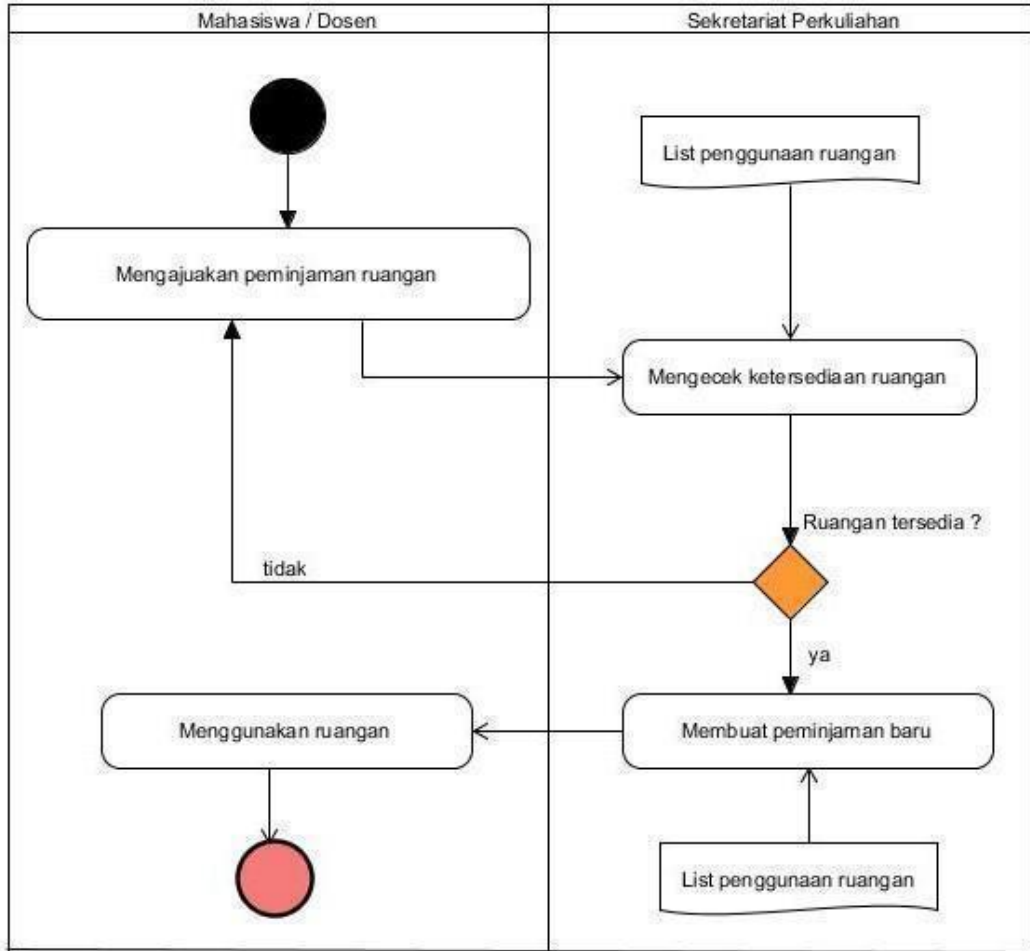
Berdasarkan hasil wawancara yang dilakukan dengan petugas yang terlibat dalam mengelola sarana prasarana di Fakultas Hukum Universitas Brawijaya, maka dapat digambarkan prosedur peminjaman ruangan untuk tujuan perkuliahan pada



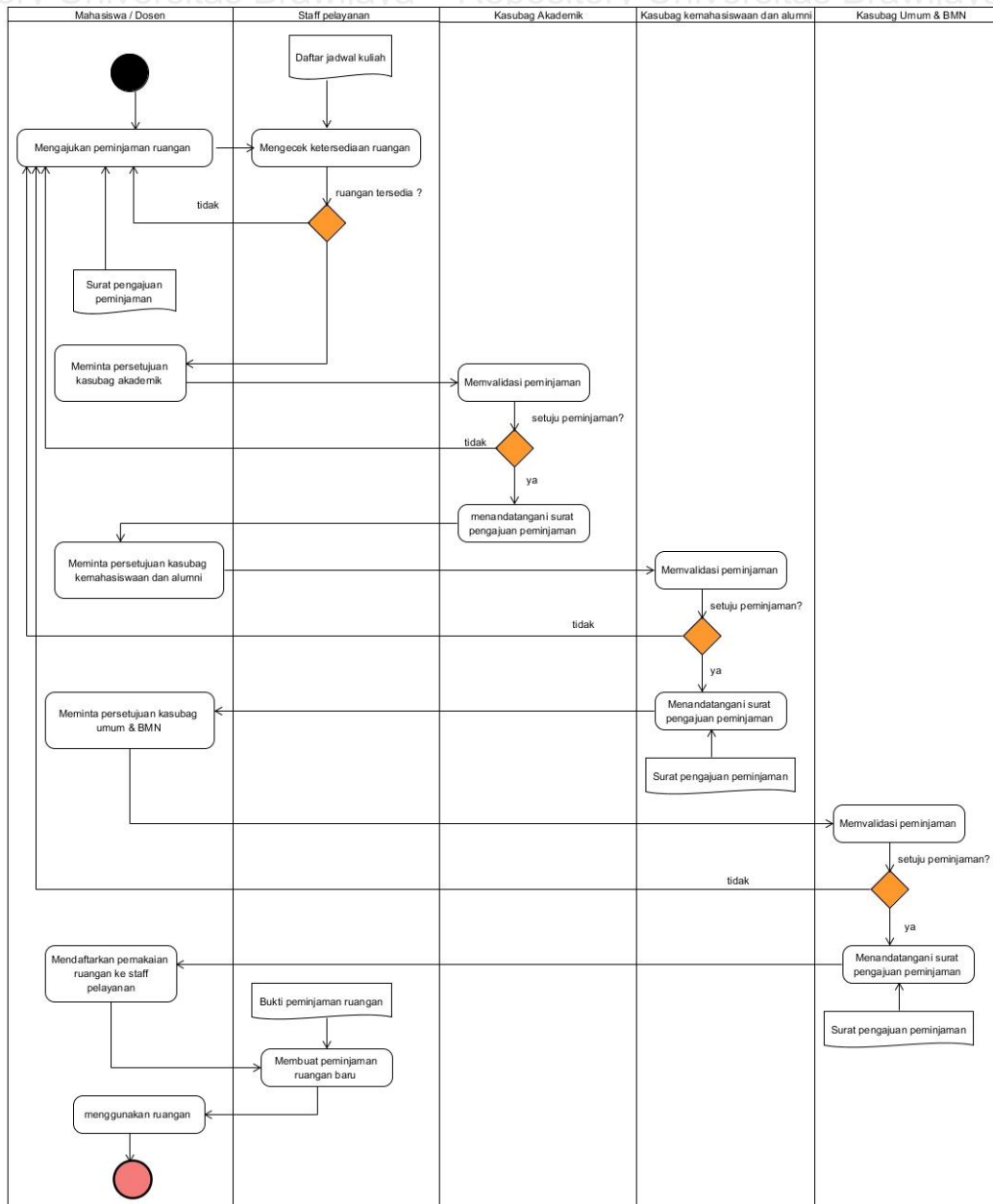
Gambar 4.2. Pada Gambar 4.2 dapat dilihat peminjaman ruangan untuk tujuan kegiatan diluar perkuliahan membutuhkan proses yang lebih panjang dibandingkan dengan peminjaman ruangan untuk tujuan perkuliahan. Pada peminjaman ini mahasiswa atau dosen harus mendapatkan validasi atau persetujuan dari kasubag akademik, kasubag kemahasiswaan dan kasubag umum. Peminjaman yang disetujui akan mendapat tanda tangan dari masing masing kasubag pada kertas permohonan peminjaman. Setelah semua kasubag sudah setuju maka mahasiswa atau dosen menyerahkan surat permohonan ke petugas pelayanan untuk dibuatkan peminjaman pemakaian ruangan baru. Setelah itu, mahasiswa akan mengisi sebuah bukti peminjaman. Prosedur peminjaman ini sering mendapat kendala. Kendala yang sering dihadapi adalah pada saat kasubag tidak berada di ruangan saat meminta tanda tangan persetujuan. Hal ini berakibat proses peminjaman akan memakan waktu lama.

4.1.1.3 Prosedur Peminjaman Barang

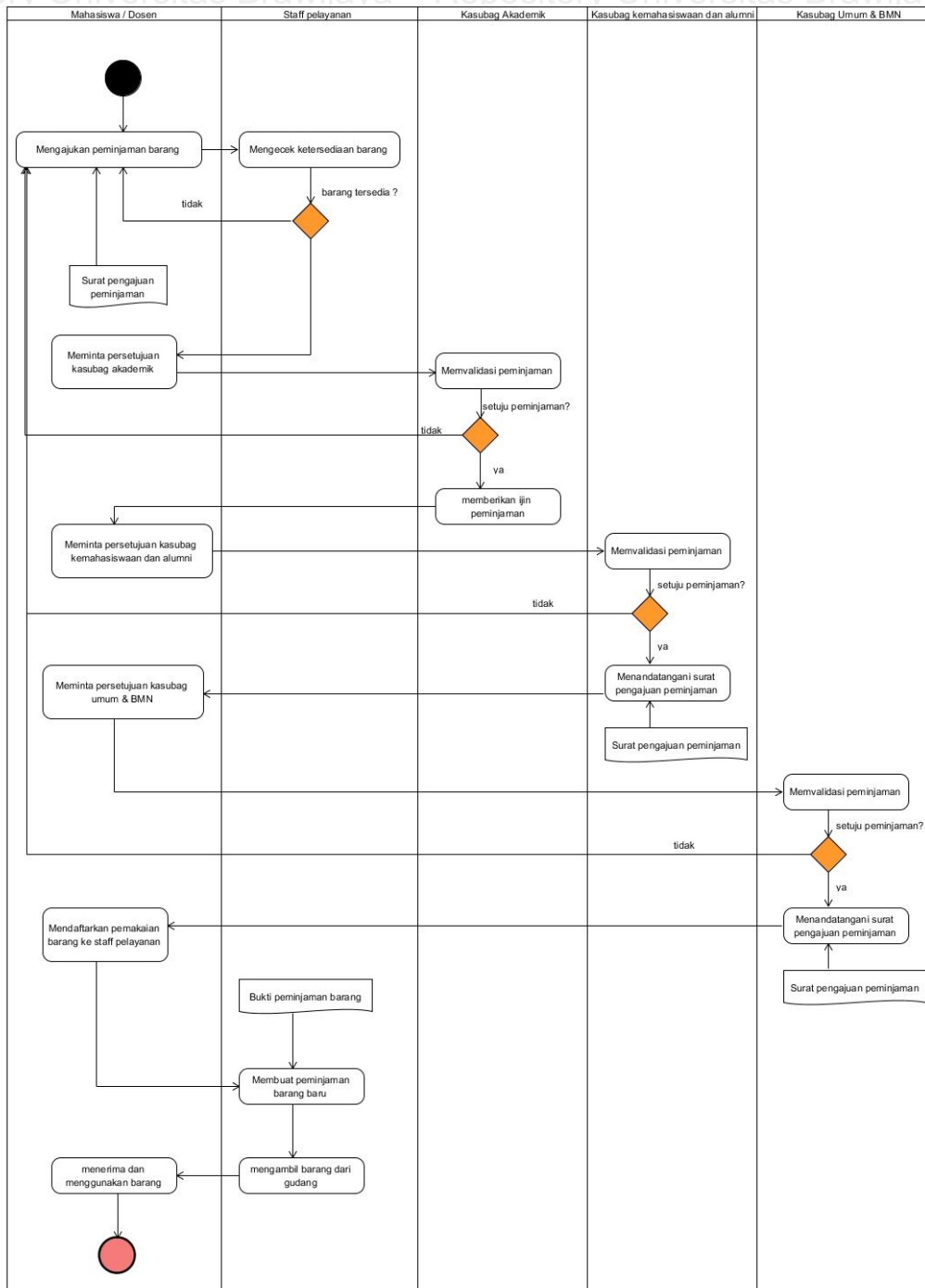
Berdasarkan hasil wawancara yang dilakukan dengan petugas yang terlibat dalam mengelola sarana prasarana di Fakultas Hukum Universitas Brawijaya, maka dapat digambarkan prosedur peminjaman barang pada Gambar 4.3. Prosedur peminjaman barang hampir sama dengan prosedur peminjaman ruangan untuk tujuan non perkuliahan. Hal yang membedakan adalah status ketersediaan barang harus ditanyakan ke staf pelayanan barang terlebih dahulu. Setelah mengetahui ketersediaan barang, mahasiswa atau dosen dapat meminta persetujuan kasubag akademik, kasubag kemahasiswaan, dan kasubag umum. Jika sudah mendapat persetujuan semua kasubag maka surat permohonan diberikan kepada staf pelayanan untuk didaftarkan menjadi peminjaman barang dan membuat bukti peminjaman barang. Kendala yang ditemui pada prosedur peminjaman barang sama dengan yang ditemui pada peminjaman ruangan. Namun peminjaman barang mendapat kendala baru dimana barang harus di cek terlebih dahulu ke gudang untuk mengetahui status ketersediaan barang.



Gambar 4.1 Prosedur Peminjaman Ruang untuk Tujuan Perkuliahan



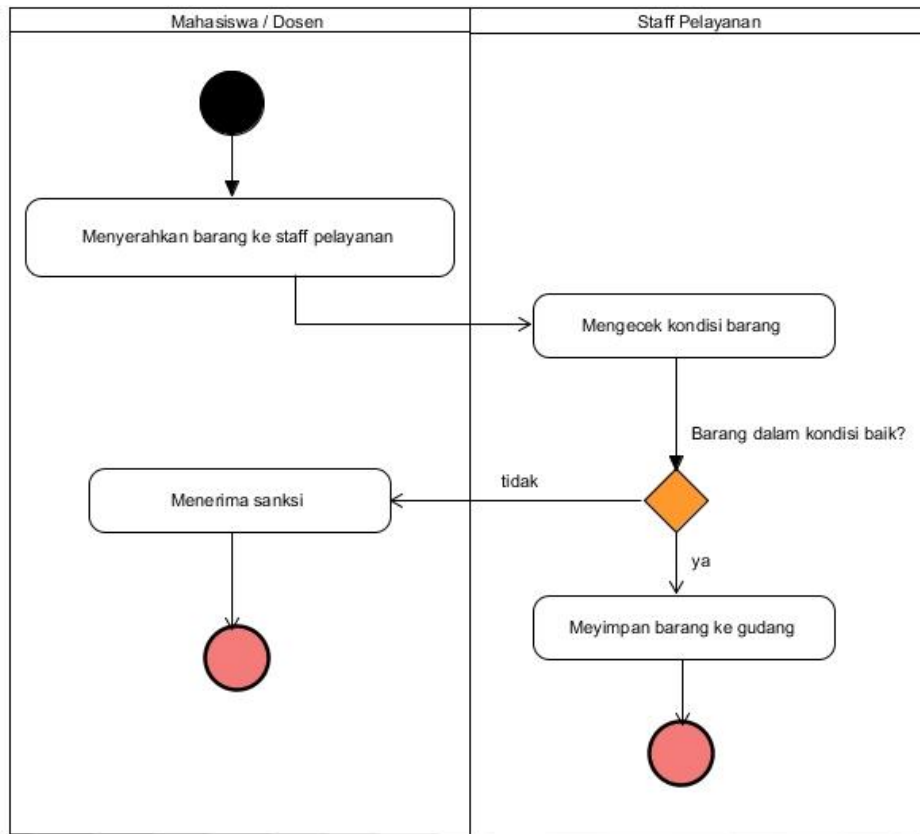
Gambar 4.2 Prosedur Peminjaman Ruang untuk Tujuan Non Perkuliahan



Gambar 4.3 Prosedur Peminjaman Barang

4.1.1.4 Prosedur Pengembalian Barang

Berdasarkan hasil wawancara yang dilakukan dengan petugas yang terlibat dalam mengelola sarana prasarana di Fakultas Hukum Universitas Brawijaya, maka dapat digambarkan prosedur peminjaman ruangan untuk tujuan perkuliahan pada Gambar 4.4.



Gambar 4.4 Prosedur Pengembalian Barang

Pada saat mahasiswa atau dosen meminjam barang maka barang tersebut harus dikembalikan. Pada prosedur pengembalian yang sudah ada, barang yang dikembalikan akan dicek oleh staf pelayanan. Peminjam harus membayar sanksi jika barang tidak kembali dengan kondisi baik. Jika kondisi barang normal maka staf pelayanan mengembalikan barang ke gudang. Pada prosedur pengembalian barang ini terdapat hal yang perlu diperbaiki. Pelaporan atau rekapan pengembalian barang tidak ada. Sehingga setiap barang yang ada di Fakultas Hukum tidak mempunyai catatan kerusakan pada setiap barang yang sudah dipinjam.

4.1.2 Proses Bisnis yang Ditawarkan

Berdasarkan kendala yang ditemui dalam prosedur kegiatan terkait penggunaan sarana prasarana saat ini. Maka penulis membuat prosedur baru yang ditawarkan kepada Fakultas Hukum Universitas Brawijaya. Prosedur baru dalam peminjaman ruangan untuk tujuan perkuliahan, peminjaman ruangan untuk tujuan non perkuliahan, peminjaman barang dan pengembalian barang.



4.1.2.1 Prosedur Peminjaman Ruang untuk Tujuan Perkuliahan yang Ditawarkan

Berdasarkan hasil wawancara yang dilakukan dengan petugas yang terlibat dalam mengelola sarana prasarana di Fakultas Hukum Universitas Brawijaya, maka dapat digambarkan prosedur peminjaman ruangan untuk tujuan perkuliahan pada Gambar 4.5. Pada Gambar 4.5 dapat dilihat bahwa mahasiswa tidak perlu menemui sekretariat lagi untuk mengetahui kondisi status ketersediaan ruangan. Mahasiswa atau dosen cukup melihat status penggunaan ruangan melalui sistem. Jika ruangan sudah dipilih maka mahasiswa dapat mengajukan peminjaman kepada sekretariat kuliah. Sekretariat kemudian membuat peminjaman baru dan harus mengganti status penggunaan ruangan yang ada di sistem agar informasi di sistem selalu *update*.

4.1.2.2 Prosedur Peminjaman Ruang untuk Tujuan Non Perkuliahan yang Ditawarkan

Berdasarkan hasil wawancara yang dilakukan dengan petugas yang terlibat dalam mengelola sarana prasarana di Fakultas Hukum Universitas Brawijaya, maka dapat digambarkan prosedur peminjaman ruangan untuk tujuan non perkuliahan pada Gambar 4.6. Pada Gambar 4.6 dapat dilihat prosedur peminjaman ruangan yang ditawarkan hampir sama dengan prosedur peminjaman ruangan untuk tujuan non perkuliahan sebelum adanya sistem. Hal yang membedakan adalah pada prosedur yang ditawarkan, mahasiswa atau dosen dapat langsung melakukan pengecekan status penggunaan ruangan melalui sistem tanpa menemui sekretariat kuliah. Pada prosedur yang ditawarkan, proses persetujuan validasi peminjaman oleh kasubag akademik, kasubag kemahasiswaan dan kasubag umum dilakukan melalui sistem. Hal ini membuat peminjam tidak perlu lagi menemui kasubag untuk meminta tanda tangan persetujuan peminjaman. Prosedur yang ditawarkan ini akan membuat kasubag dapat melakukan validasi kapan dan dimana saja.

4.1.2.3 Prosedur Peminjaman Barang yang Ditawarkan

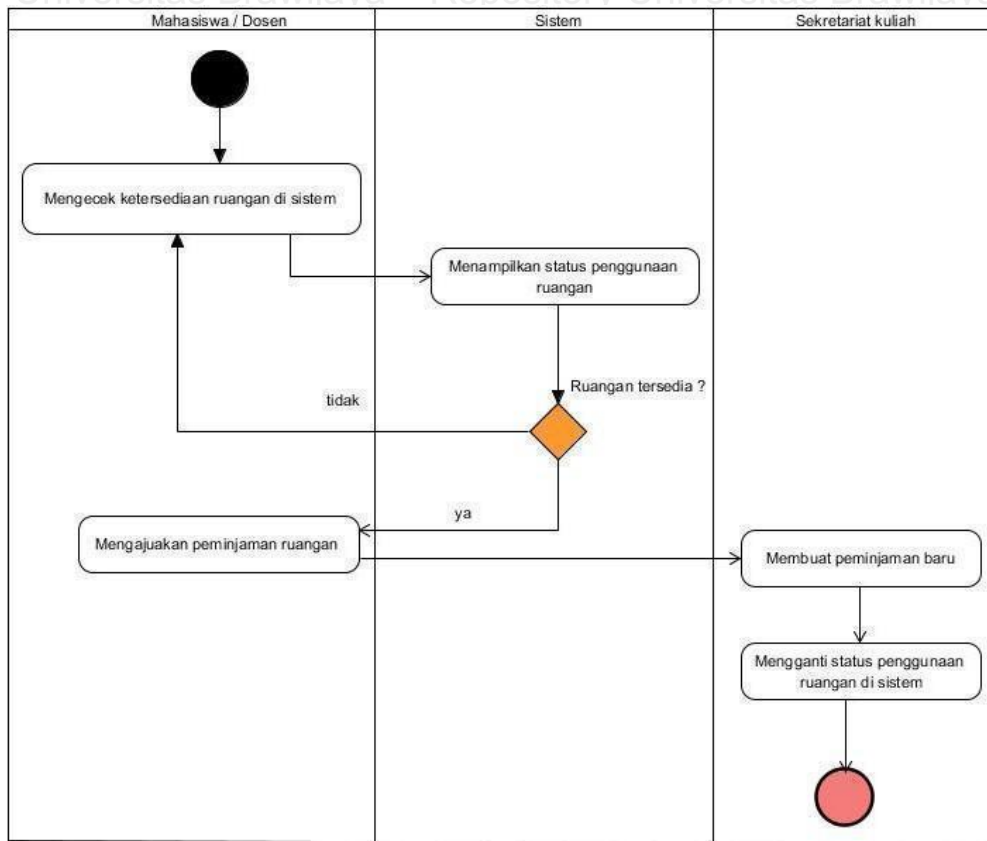
Berdasarkan hasil wawancara yang dilakukan dengan petugas yang terlibat dalam mengelola sarana prasarana di Fakultas Hukum Universitas Brawijaya, maka dapat digambarkan prosedur peminjaman barang pada Gambar 4.7. Prosedur peminjaman barang yang ditawarkan ini hampir sama dengan prosedur peminjaman ruangan non perkuliahan yang ditawarkan. Perbedaannya hanya terletak pada sarana yang dipinjam. Perbedaan yang lain adalah pada peminjaman barang, barang akan dikembalikan.

4.1.2.4 Prosedur Pengembalian Barang yang Ditawarkan

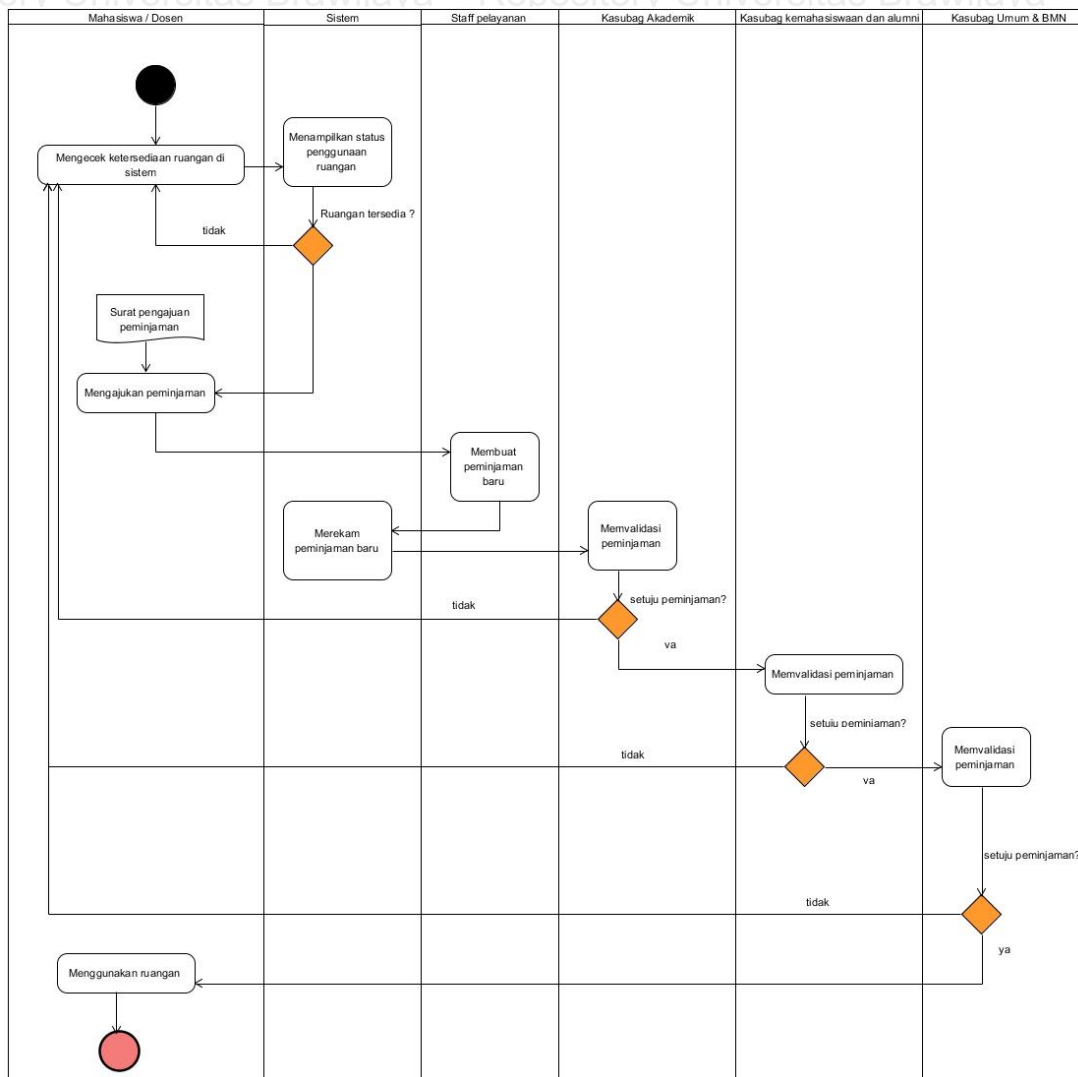
Berdasarkan hasil wawancara yang dilakukan dengan petugas yang terlibat dalam mengelola sarana prasarana di Fakultas Hukum Universitas Brawijaya, maka dapat digambarkan prosedur pengembalian barang pada Gambar 4.8. Prosedur pengembalian barang pada Gambar 4.8 tidak jauh berbeda dengan prosedur yang telah ada. Hal yang membedakan adalah staf pelayanan akan menggunakan



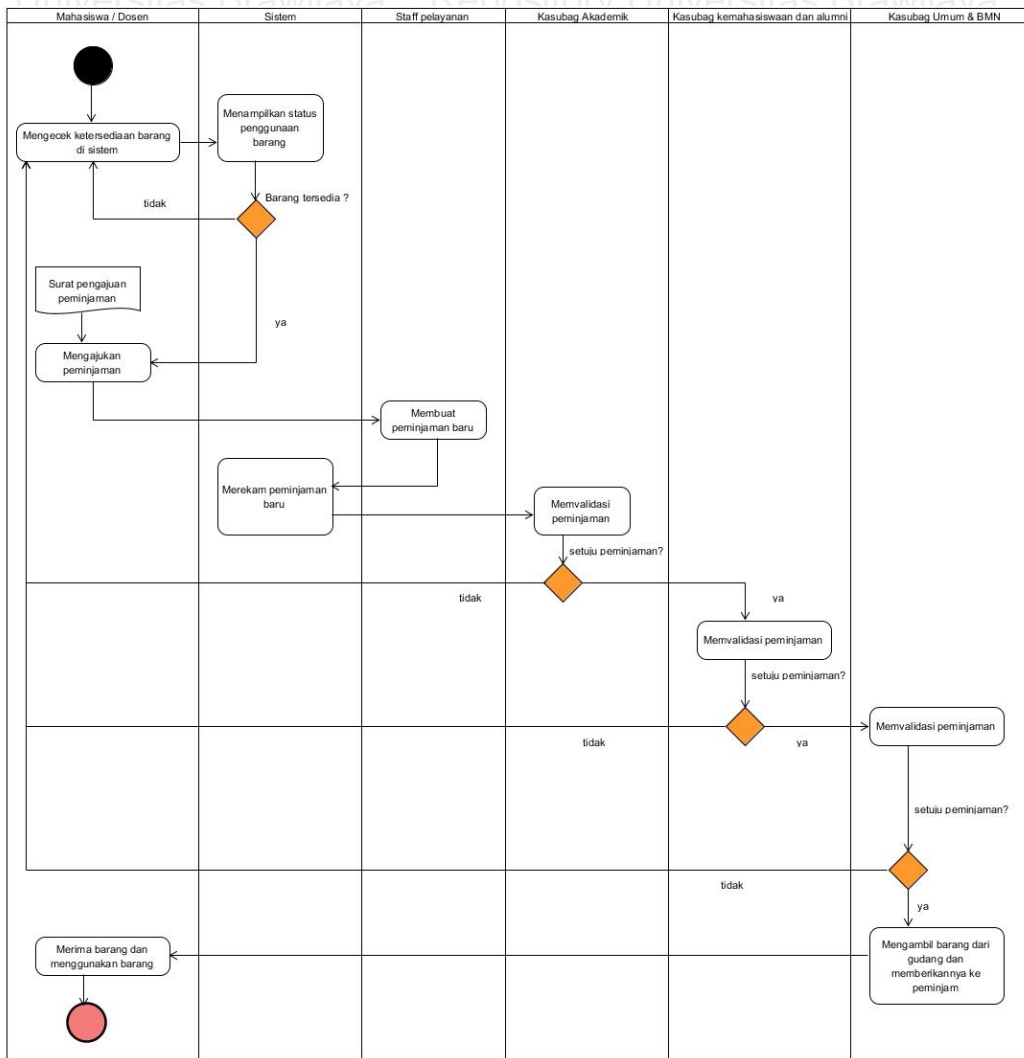
sistem dalam merekam status dan catatan peminjaman. Dengan menggunakan prosedur yang ditawarkan ini, status keberadaan barang akan terekam dengan baik. Sistem akan merekam segala kegiatan peminjaman barang, mulai dari awal proses peminjaman sampai pengembalian barang. Hal ini akan membuat staf pelayanan dapat dengan mudah mengetahui posisi barang. Apakah barang tersedia di gudang dan dapat dipinjamkan? Apakah barang sedang dipinjam? Apakah barang sudah dikembalikan? Pada tanggal dan waktu berapa barang dapat dipinjam? Semua pertanyaan itu akan dapat mudah terjawab hanya dengan melihat satu layar tampilan di sistem.



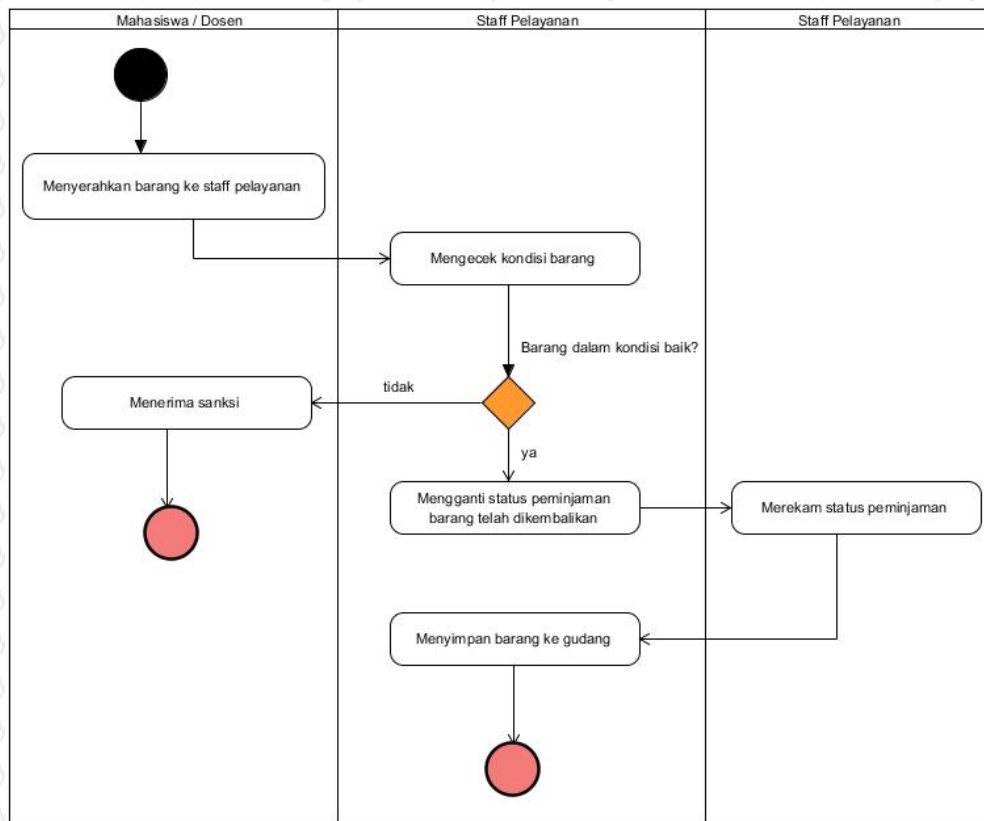
Gambar 4.5 Prosedur Peminjaman Ruang untuk Perkuliahan yang Ditawarkan



Gambar 4.6 Prosedur Peminjaman Ruang untuk Tujuan Non Perkuliahan



Gambar 4.7 Prosedur Peminjaman Barang yang Ditawarkan



Gambar 4.8 Prosedur Pengembalian Barang yang Ditawarkan

4.2 Analisis Kebutuhan

4.2.1 Gambaran Umum Sistem

Sistem manajemen sarana prasarana melingkupi pekerjaan sebagai berikut :

1. Pemetaan jadwal kuliah
2. Pemetaan status penggunaan ruangan dan barang
4. Peminjaman ruangan dan barang
5. Perekaman pelaporan peminjaman
6. Perekaman *complaint* ruangan dan barang

Dalam sistem ini terdapat 7 aktor yang terlibat yakni mahasiswa, dosen, kasubag akademik, kasubag kemahasiswaan, kasubag umum dan staf pelayanan. Seluruh actor dapat menggunakan sistem untuk melihat jadwal kuliah, status penggunaan ruangan, dan status penggunaan barang. Seluruh aktor juga akan dapat menggunakan sistem untuk menyampaikan keluhan terkait masalah sarana



prasarana. Mahasiswa dan dosen dapat juga menggunakan sistem ini untuk meminjam ruangan dan barang.

4.2.2 Identifikasi Aktor

Aktor adalah orang yang ikut ambil bagian dalam menjalankan sebuah proses bisnis. Aktor yang terlibat dalam sistem manajemen sarana prasarana ini terdiri dari lima aktor yang memiliki perannya masing-masing. Adapun aktor yang terlibat dijelaskan pada Tabel 4.1 berikut.

Tabel 4.1 Identifikasi Aktor

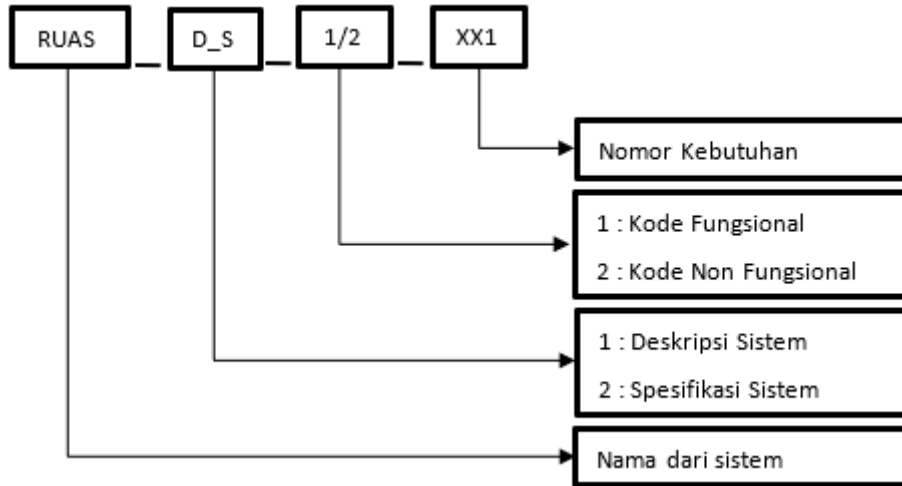
No	Aktor	Deskripsi
1	Guest	Guest adalah aktor untuk mahasiswa atau dosen. Aktor guest hanya dapat melihat informasi yang ditampilkan oleh sistem saja. Informasi seperti informasi jadwal kuliah, status penggunaan ruangan dan lainnya.
2	Sekretariat Perkuliahan	Sekretariat Perkuliahan adalah aktor yang melakukan mengatur jadwal kuliah pengganti dan mengganti status penggunaan ruangan perkuliahan.
3	staf Pelayanan	staf Pelayanan adalah aktor yang melakukan penambahan peminjaman ruangan untuk kegiatan diluar perkuliahan dan peminjaman barang serta aktor yang mengatur status pengembalian barang.
4	Validator	Validator dibedakan menjadi tiga jenis yaitu validator kasubag akademik, validator kasubag umum dan kasubag kemahasiswaan. Validator kasubag akademik adalah aktor yang melakukan validasi peminjaman mewakili kasubag akademik. Kasubag umum adalah aktor yang melakukan validasi peminjaman mewakili kasubag umum dan perlengkapan. Kasubag kemahasiswaan adalah aktor yang melakukan validasi peminjaman mewakili kasubag kemahasiswaan
5	Admin	Admin adalah aktor yang mengatur sistem secara menyeluruh

4.2.3 Kebutuhan Sistem

Daftar kebutuhan sistem adalah daftar yang menjelaskan kemampuan yang harus dimiliki oleh sistem. Daftar kebutuhan sistem ini membagi kebutuhan menjadi dua jenis kebutuhan, yakni kebutuhan fungsional dan kebutuhan non fungsional. Untuk lebih memperjelas kebutuhan fungsional dilengkapi dengan spesifikasi kebutuhan dan kode penomoran. Pada Gambar 4.9 dibawah dapat



dilihat aturan penomoran daftar kebutuhan sistem. Kode penomoran menggunakan format RUAS_D/S_1/2_XX1. RUAS menggambarkan nama dari sistem yang dibangun. D/S merupakan inisial menandakan sebuah daftar adalah deskripsi atau spesifikasi. 1/2 merupakan inisial yang menandakan tipe kebutuhan fungsional atau non fungsional. XX1 merupakan inisial untuk memberikan penomoran daftar kebutuhan.



Gambar 4.9 Aturan Penomoran

4.2.3.1 Kebutuhan Fungsional

Kebutuhan fungsional merupakan kemampuan atau layanan yang harus dimiliki oleh sistem. Kebutuhan fungsional dijelaskan pada Tabel 4.2 berikut.

Tabel 4.2 Kebutuhan Fungsional

No	Kode Kebutuhan	Deskripsi/Spesifikasi
1	RUAS_D_1_01	Sistem menyediakan fungsi login bagi pengguna sistem
2	RUAS_S_1_01	Spesifikasi kebutuhan <ul style="list-style-type: none"> • Login menggunakan <i>username</i> dan <i>password</i> • Sistem menampilkan pesan peringatan jika melakukan login dengan <i>username</i> dan <i>password</i> salah
3	RUAS_D_1_02	Sistem menyediakan fungsi menambah data mahasiswa untuk admin
4	RUAS_S_1_02	Spesifikasi Kebutuhan <ul style="list-style-type: none"> • Sistem menyediakan tombol "Tambah Mahasiswa"



		<ul style="list-style-type: none"> ● Sistem menyediakan form untuk menambah data mahasiswa baru <p>Dilakukan oleh admin dengan mengisi NIM dan Nama</p>
5	RUAS_D_1_03	Sistem menyediakan fungsi menambah data dosen untuk admin
6	RUAS_S_1_03	<p>Spesifikasi Kebutuhan</p> <ul style="list-style-type: none"> ● Sistem menyediakan tombol “Tambah Dosen” ● Sistem menyediakan form untuk menambah data dosen baru <p>Dilakukan oleh admin dengan mengisi NIP dan Nama</p>
7	RUAS_S_1_04	Sistem menyediakan fungsi menambah data operator untuk admin
8	RUAS_S_1_04	<p>Spesifikasi Kebutuhan</p> <ul style="list-style-type: none"> ● Sistem menyediakan tombol “Tambah Operator” ● Sistem menyediakan form untuk menambah operator baru ● Dilakukan oleh admin dengan mengisikan nama, username, password dan jenis validator ● Jenis operator terdiri dari kasubag akademik, kasubag kemahasiswaan dan kasubag umum, sekretariat kuliah, dan staf pelayanan
9	RUAS_D_1_05	Sistem menyediakan fungsi menambah data ruangan untuk admin
10	RUAS_S_1_05	<p>Spesifikasi Kebutuhan</p> <ul style="list-style-type: none"> ● Sistem menyediakan tombol “Tambah Ruang” ● Sistem menyediakan form untuk menambah data ruangan baru ● Dilakukan oleh admin dengan mengisikan nama ruangan dan jenis ruangan. ● Jenis ruangan ada ruangan kuliah dan ruangan non kuliah
11	RUAS_D_1_06	Sistem menyediakan fungsi menambah data barang untuk admin
12	RUAS_S_1_06	Spesifikasi Kebutuhan



		<ul style="list-style-type: none"> ● Sistem menyediakan tombol “Tambah Barang” ● Sistem menyediakan form untuk menambah data barang baru ● Dilakukan oleh admin dengan mengisikan nama barang dan jenis barang ● Jenis barang digunakan untuk menggolongkan barang barang sama meskipun mempunyai merek berbeda
13	RUAS_D_1_07	Sistem menyediakan fungsi menambah data semester baru untuk admin
14	RUAS_S_1_07	<p>Spesifikasi Kebutuhan</p> <ul style="list-style-type: none"> ● Sistem menyediakan tombol “Tambah Semester” ● Sistem menyediakan form untuk menambah data semester ajaran baru ● Dilakukan oleh admin dengan mengisikan nama semester, tanggal mulai dan tanggal selesai
15	RUAS_D_1_08	Sistem menyediakan fungsi menambah data jadwal kuliah baru untuk admin
16	RUAS_S_1_08	<p>Spesifikasi Kebutuhan</p> <ul style="list-style-type: none"> ● Sistem menyediakan tombol “Tambah Jadwal” ● Sistem menyediakan form untuk menambah data jadwal kuliah baru ● Data yang diisi adalah hari, jam kuliah, ruangan, kelas, program studi dan dosen
17	RUAS_D_1_09	Sistem menyediakan fungsi menambah data lembaga baru untuk admin
18	RUAS_S_1_09	<p>Spesifikasi Kebutuhan</p> <ul style="list-style-type: none"> ● Sistem menyediakan tombol “Tambah Lembaga” ● Sistem menyediakan form untuk menambah data lembaga baru <p>Dilakukan oleh admin dengan mengisikan nama lembaga</p>
19	RUAS_D_1_10	Sistem menyediakan fungsi menambah data mata kuliah baru untuk admin
20	RUAS_S_1_10	<p>Spesifikasi Kebutuhan</p> <ul style="list-style-type: none"> ● Sistem menyediakan tombol “Tambah Mata Kuliah”



		<ul style="list-style-type: none"> ● Sistem menyediakan form untuk menambah data mata kuliah <p>Dilakukan oleh admin dengan mengisikan kode mata kuliah dan nama mata kuliah</p>
21	RUAS_D_1_11	Sistem menyediakan fungsi menambah data jam kuliah baru untuk admin
22	RUAS_S_1_11	<p>Spesifikasi Kebutuhan</p> <ul style="list-style-type: none"> ● Sistem menyediakan tombol “Tambah Jam Kuliah” ● Sistem menyediakan form untuk menambah data jam kuliah baru <p>Dilakukan oleh admin dengan mengisikan jam kuliah</p>
23	RUAS_D_1_12	Sistem menyediakan fungsi menambah data program studi untuk admin
24	RUAS_S_1_12	<p>Spesifikasi Kebutuhan</p> <ul style="list-style-type: none"> ● Sistem menyediakan tombol “Tambah Program Studi” ● Sistem menyediakan form untuk menambah program studi baru <p>Dilakukan oleh admin dengan mengisikan nama program studi</p>
25	RUAS_D_1_13	Sistem menyediakan fungsi <i>edit</i> data mahasiswa untuk admin
26	RUAS_S_1_13	<p>Spesifikasi Kebutuhan</p> <ul style="list-style-type: none"> ● Sistem menyediakan tombol “<i>Edit</i> Mahasiswa” ● Sistem menyediakan form untuk <i>edit</i> data mahasiswa baru ● Data yang dapat diubah adalah NIM, Nama, dan status aktif
27	RUAS_D_1_14	Sistem menyediakan fungsi <i>edit</i> data dosen untuk admin
28	RUAS_S_1_14	<p>Spesifikasi Kebutuhan</p> <ul style="list-style-type: none"> ● Sistem menyediakan tombol “<i>Edit</i> Dosen” ● Sistem menyediakan form untuk <i>edit</i> data dosen ● Data yang dapat diubah adalah NIM, Nama dan status aktif



29	RUAS_S_1_15	Sistem menyediakan fungsi <i>edit</i> data operator untuk admin
30	RUAS_S_1_15	Spesifikasi Kebutuhan <ul style="list-style-type: none"> ● Sistem menyediakan tombol “Edit Operator” ● Sistem menyediakan form untuk <i>edit</i> operator baru ● Data yang dapat diubah adalah password dan status operator
31	RUAS_D_1_16	Sistem menyediakan fungsi <i>edit</i> data ruangan untuk admin
32	RUAS_S_1_16	Spesifikasi Kebutuhan <ul style="list-style-type: none"> ● Sistem menyediakan tombol “Edit Ruangan” ● Sistem menyediakan form untuk <i>edit</i> data ruangan ● Data yang dapat diedit adalah nama ruangan dan status ruangan
33	RUAS_D_1_17	Sistem menyediakan fungsi <i>edit</i> data barang untuk admin
34	RUAS_S_1_17	Spesifikasi Kebutuhan <ul style="list-style-type: none"> ● Sistem menyediakan tombol “Edit Barang” ● Sistem menyediakan form untuk <i>edit</i> data barang ● Data yang dapat diubah adalah nama barang dan jenis barang
35	RUAS_D_1_18	Sistem menyediakan fungsi <i>edit</i> data semester untuk admin
36	RUAS_S_1_18	Spesifikasi Kebutuhan <ul style="list-style-type: none"> ● Sistem menyediakan tombol “Edit Semester” ● Sistem menyediakan form untuk <i>edit</i> data semester ● Data yang dapat diubah adalah nama semester, tanggal mulai dan tanggal selesai
37	RUAS_D_1_19	Sistem menyediakan fungsi <i>edit</i> data jadwal kuliah untuk admin
38	RUAS_S_1_19	Spesifikasi Kebutuhan



		<ul style="list-style-type: none"> ● Sistem menyediakan tombol “<i>Edit Jadwal</i>” ● Sistem menyediakan form untuk <i>edit</i> data jadwal kuliah ● Data yang dapat diubah adalah hari, jam kuliah, ruangan, kelas, program studi dan dosen
39	RUAS_D_1_20	Sistem menyediakan fungsi <i>edit</i> data lembaga untuk admin
40	RUAS_S_1_20	Spesifikasi Kebutuhan <ul style="list-style-type: none"> ● Sistem menyediakan tombol “<i>Edit Lembaga</i>” ● Sistem menyediakan form untuk <i>edit</i> data Lembaga ● Data yang dapat diubah adalah nama lembaga
41	RUAS_D_1_21	Sistem menyediakan fungsi <i>edit</i> data mata kuliah untuk admin
42	RUAS_S_1_21	Spesifikasi Kebutuhan <ul style="list-style-type: none"> ● Sistem menyediakan tombol “<i>Edit Mata Kuliah</i>” ● Sistem menyediakan form untuk <i>edit</i> data mata kuliah ● Data yang dapat diubah adalah nama mata kuliah
43	RUAS_D_1_22	Sistem menyediakan fungsi <i>edit</i> data jam kuliah untuk admin
44	RUAS_S_1_22	Spesifikasi Kebutuhan <ul style="list-style-type: none"> ● Sistem menyediakan tombol “<i>Edit Jam kuliah</i>” ● Sistem menyediakan form untuk <i>edit</i> data jam kuliah ● Data yang dapat diubah adalah jam kuliah
45	RUAS_D_1_23	Sistem menyediakan fungsi <i>edit</i> data program studi untuk admin
46	RUAS_S_1_23	Spesifikasi Kebutuhan <ul style="list-style-type: none"> ● Sistem menyediakan tombol “<i>Edit Program Studi</i>” ● Sistem menyediakan form untuk menambah program studi baru ● Data yang dapat diubah adalah nama program studi



47	RUAS_D_1_24	Sistem menyediakan fungsi hapus data mahasiswa untuk admin
48	RUAS_S_1_24	Spesifikasi Kebutuhan <ul style="list-style-type: none"> ● Sistem menyediakan tombol dengan <i>icon</i> tong sampah untuk menghapus data mahasiswa ● Sistem menyediakan form untuk hapus data mahasiswa
49	RUAS_D_1_25	Sistem menyediakan fungsi hapus data dosen untuk admin
50	RUAS_S_1_25	Spesifikasi Kebutuhan <ul style="list-style-type: none"> ● Sistem menyediakan tombol dengan <i>icon</i> tong sampah untuk menghapus data dosen ● Sistem menampilkan pesan pemberitahuan jika data dosen telah berhasil dihapus
51	RUAS_S_1_26	Sistem menyediakan fungsi hapus data operator untuk admin
52	RUAS_S_1_26	Spesifikasi Kebutuhan <ul style="list-style-type: none"> ● Sistem menyediakan tombol dengan <i>icon</i> tong sampah untuk menghapus data operator ● Sistem menampilkan pesan pemberitahuan jika data operator telah berhasil dihapus
53	RUAS_D_1_27	Sistem menyediakan fungsi hapus data ruangan untuk admin
54	RUAS_S_1_27	Spesifikasi Kebutuhan <ul style="list-style-type: none"> ● Sistem menyediakan tombol dengan <i>icon</i> tong sampah untuk menghapus data ruangan <p>Sistem menampilkan pesan pemberitahuan jika data ruangan telah berhasil dihapus</p>
55	RUAS_D_1_28	Sistem menyediakan fungsi hapus data barang untuk admin
56	RUAS_S_1_28	Spesifikasi Kebutuhan <ul style="list-style-type: none"> ● Sistem menyediakan tombol dengan <i>icon</i> tong sampah untuk menghapus data barang



		Sistem menampilkan pesan pemberitahuan jika data barang telah berhasil dihapus
57	RUAS_D_1_29	Sistem menyediakan fungsi hapus data semester untuk admin
58	RUAS_S_1_29	Spesifikasi Kebutuhan <ul style="list-style-type: none"> • Sistem menyediakan tombol dengan <i>icon</i> tong sampah untuk menghapus data semester Sistem menampilkan pesan pemberitahuan jika data semester telah berhasil dihapus
59	RUAS_D_1_30	Sistem menyediakan fungsi hapus data jadwal kuliah untuk admin
60	RUAS_S_1_30	Spesifikasi Kebutuhan <ul style="list-style-type: none"> • Sistem menyediakan tombol dengan <i>icon</i> tong sampah untuk menghapus data jadwal kuliah Sistem menampilkan pesan pemberitahuan jika data jadwal kuliah telah berhasil dihapus
61	RUAS_D_1_31	Sistem menyediakan fungsi hapus data lembaga untuk admin
62	RUAS_S_1_31	Spesifikasi Kebutuhan <ul style="list-style-type: none"> • Sistem menyediakan tombol dengan <i>icon</i> tong sampah untuk menghapus data lembaga Sistem menampilkan pesan pemberitahuan jika data lembaga telah berhasil dihapus
63	RUAS_D_1_32	Sistem menyediakan fungsi hapus data mata kuliah untuk admin
64	RUAS_S_1_32	Spesifikasi Kebutuhan <ul style="list-style-type: none"> • Sistem menyediakan tombol dengan <i>icon</i> tong sampah untuk menghapus data mata kuliah Sistem menampilkan pesan pemberitahuan jika data mata kuliah telah berhasil dihapus
65	RUAS_D_1_33	Sistem menyediakan fungsi hapus data jam kuliah untuk admin
66	RUAS_S_1_33	Spesifikasi Kebutuhan



		<ul style="list-style-type: none"> ● Sistem menyediakan tombol dengan <i>icon</i> tong sampah untuk menghapus data jam kuliah <p>Sistem menampilkan pesan pemberitahuan jika data jam kuliah telah berhasil dihapus</p>
67	RUAS_D_1_34	Sistem menyediakan fungsi hapus data program studi untuk admin
68	RUAS_S_1_34	<p>Spesifikasi Kebutuhan</p> <ul style="list-style-type: none"> ● Sistem menyediakan tombol dengan <i>icon</i> tong sampah untuk menghapus data program studi <p>Sistem menampilkan pesan pemberitahuan jika data program studi telah berhasil dihapus</p>
69	RUAS_D_1_35	Sistem menyediakan fungsi untuk melihat rekap peminjaman untuk admin
70	RUAS_S_1_35	<p>Spesifikasi Kebutuhan</p> <ul style="list-style-type: none"> ● Sistem menampilkan total jumlah peminjaman dalam perbulan yang ● Data ditampilkan dalam bentuk tabel yang terdiri dari kolom nama bulan januari sampai desember, dan baris diisi dengan jumlah peminjaman dari masing masing jenis peminjaman selama setahun ● Sistem menyediakan fungsi untuk menampilkan rekap berdasarkan tahun
71	RUAS_D_1_36	Sistem menyediakan fungsi untuk melihat rekap penggunaan ruangan untuk admin
72	RUAS_S_1_36	<p>Spesifikasi Kebutuhan</p> <ul style="list-style-type: none"> ● Sistem menampilkan total jumlah penggunaan ruangan yang dipinjam dalam perbulan ● Data ditampilkan dalam bentuk tabel yang terdiri dari kolom nama bulan januari sampai desember, dan baris diisi dengan jumlah penggunaan ruangan dari masing masing ruangan ● Sistem menyediakan fungsi untuk menampilkan rekap penggunaan ruangan berdasarkan tahun



73	RUAS_D_1_37	Sistem menyediakan fungsi untuk melihat rekap penggunaan barang untuk admin
74	RUAS_S_1_37	Spesifikasi Kebutuhan <ul style="list-style-type: none"> ● Sistem menampilkan total jumlah penggunaan barang yang dipinjam dalam perbulan ● Data ditampilkan dalam bentuk tabel yang terdiri dari kolom nama bulan januari sampai desember, dan baris diisi dengan jumlah penggunaan barang dari masing masing barang ● Sistem menyediakan fungsi untuk menampilkan rekap penggunaan raungan berdasarkan tahun
75	RUAS_D_1_38	Sistem menyediakan fungsi untuk melihat rekap <i>complaint</i> untuk admin
76	RUAS_S_1_38	Spesifikasi Kebutuhan <ul style="list-style-type: none"> ● Sistem menampilkan total jumlah <i>complaint</i> selama setahun ● Data ditampilkan dalam bentuk tabel yang terdiri kolom nama bulan dan jumlah <i>complaint</i>. ● Sistem menyediakan fungsi untuk menampilkan rekap <i>complaint</i> berdasarkan tahun
77	RUAS_D_1_39	Sistem menyediakan fungsi untuk melihat jadwal kuliah untuk guest, validator, staf pelayanan, dan sekretariat kuliah
78	RUAS_S_1_39	Spesifikasi Kebutuhan <ul style="list-style-type: none"> ● Sistem menampilkan jadwal kuliah dalam bentuk tabel ● Tabel terdiri dari jam pelajaran sebagai kolom dan ruangan sebagai baris ● Setiap sel dari tabel menampilkan data jadwal kuliah yang berisi matakuliah, dosen, dan kelas ● Setiap sel menampilkan status pengganti jika jadwal kuliah adalah jadwal pengganti ● Sistem menyedikan filter berdasarkan hari, ruangan, program studi, dosen, mata kuliah dan status pengganti



79	RUAS_D_1_40	Sistem menyediakan fungsi untuk melihat status penggunaan ruangan kelas untuk guest, validator, staf pelayanan, dan sekretariat kuliah
80	RUAS_S_1_40	<p>Spesifikasi Kebutuhan</p> <ul style="list-style-type: none"> ● Data ditampilkan dalam tabel yang terdiri dari rentang per jam kuliah sebagai kolom dan nama ruangan sebagai baris ● ketersediaan ruangan dalam bentuk <i>icon</i> centang dan <i>icon</i> silang ● <i>Icon</i> centang untuk menandakan ruangan tersedia untuk dipinjam dan <i>icon</i> silang menandakan ruangan tidak tersedia untuk dipinjam atau digunakan ● Setiap sel dari tabel status ketersediaan ruangan dalam bentuk <i>icon</i> centang dan <i>icon</i> silang ● <i>Icon</i> centang untuk menandakan ruangan tersedia untuk dipinjam dan <i>icon</i> silang menandakan ruangan tidak tersedia untuk dipinjam atau digunakan
81	RUAS_D_1_41	Sistem menyediakan fungsi untuk melihat status penggunaan ruangan non kelas untuk guest, validator, staf pelayanan, dan sekretariat kuliah
82	RUAS_S_1_41	<p>Spesifikasi Kebutuhan</p> <ul style="list-style-type: none"> ● Data ditampilkan dalam tabel yang terdiri dari rentang per jam sebagai kolom dan nama ruangan sebagai baris ● Rentang jam dalam tabel penggunaan ruangan kelas dan non kelas dimulai dari pukul 06:00 sampai pukul 21:00 ● Setiap sel dari tabel status ketersediaan ruangan dalam bentuk <i>icon</i> centang dan <i>icon</i> silang ● <i>Icon</i> centang untuk menandakan ruangan tersedia untuk dipinjam dan <i>icon</i> silang menandakan ruangan tidak tersedia untuk dipinjam atau digunakan



		<ul style="list-style-type: none"> • Setiap sel dari tabel status ketersediaan ruangan dalam bentuk <i>icon</i> centang dan <i>icon</i> silang • <i>Icon</i> centang untuk menandakan ruangan tersedia untuk dipinjam dan <i>icon</i> silang menandakan ruangan tidak tersedia untuk dipinjam atau digunakan
83	RUAS_D_1_42	Sistem menyediakan fungsi untuk melihat status penggunaan barang untuk guest, validator, staf pelayanan, dan sekretariat kuliah
84	RUAS_S_1_42	<p>Spesifikasi Kebutuhan</p> <ul style="list-style-type: none"> • Data ditampilkan dalam tabel yang terdiri dari rentang per jam sebagai kolom dan barang sebagai baris • Rentang jam dalam tabel penggunaan ruangan kelas dan non kelas dimulai dari pukul 06:00 sampai pukul 21:00 • Setiap sel dari tabel status ketersediaan ruangan dalam bentuk <i>icon</i> centang dan <i>icon</i> silang • <i>Icon</i> centang untuk menandakan ruangan tersedia untuk dipinjam dan <i>icon</i> silang menandakan ruangan tidak tersedia untuk dipinjam atau digunakan • Setiap sel dari tabel status ketersediaan ruangan dalam bentuk <i>icon</i> centang dan <i>icon</i> silang • <i>Icon</i> centang untuk menandakan ruangan tersedia untuk dipinjam dan <i>icon</i> silang menandakan ruangan tidak tersedia untuk dipinjam atau digunakan
85	RUAS_D_1_43	Sistem menyediakan fungsi untuk mengecek status peminjaman untuk guest
86	RUAS_S_1_43	<p>Spesifikasi Kebutuhan</p> <ul style="list-style-type: none"> • Pengecekan dilakukan dengan menggunakan kode peminjaman • Sistem menampilkan halaman untuk menerima input kode peminjaman



		<ul style="list-style-type: none"> ● Sistem menyediakan tombol pencarian ● Sistem menampilkan detail peminjaman setelah tombol pencarian ditekan ● Data yang ditampilkan adalah kode peminjaman, nama peminjam, tanggal peminjaman, sarana yang digunakan dan status peminjaman.
87	RUAS_D_1_44	Sistem menyediakan fungsi tambah peminjaman ruangan kelas untuk sekretariat kuliah
88	RUAS_S_1_44	<p>Spesifikasi Kebutuhan</p> <ul style="list-style-type: none"> ● Sistem menampilkan halaman status penggunaan ruangan kelas ● Sekretariat kuliah dapat mengklik salah satu icon bertanda centang ● Sistem menampilkan pop up berisi form yang akan diisi oleh staf pelayanan ● Data yang harus diisi adalah nim/nip peminjam, tanggal penggunaan, jam kuliah, ruangan, kode mata kuliah, nip dosen, program studi, kelas dan keterangan peminjaman ● Sistem menyediakan tombol submit untuk membuat peminjaman kelas baru ● Sistem menampilkan pesan peringatan untuk data yang dibiarkan kosong saat menekan tombol submit ● Sistem menampilkan pesan pemberitahuan jika peminjaman telah berhasil dilakukan
89	RUAS_D_1_45	Sistem menyediakan fungsi tambah peminjaman ruangan non kelas untuk staf pelayanan
90	RUAS_S_1_45	<p>Spesifikasi Kebutuhan</p> <ul style="list-style-type: none"> ● Sistem menampilkan halaman berisi form untuk mengisi data peminjaman ruangan non kelas ● Data yang harus diisi adalah nim/nip peminjam, nama lembaga, tanggal penggunaan, jam mulai pemakaian ruangan, jam selesai



		<p>pemakaian, nama kegiatan, keterangan dan ruangan yang digunakan</p> <ul style="list-style-type: none"> ● Sistem menyediakan tombol submit untuk membuat peminjaman kelas baru ● Sistem menampilkan pesan peringatan untuk data yang dibiarkan kosong saat menekan tombol submit ● Jam pemakaian dimulai dari pukul 06:00 sampai pukul 21:00
91	RUAS_D_1_46	Sistem menyediakan fungsi tambah peminjaman barang untuk staf pelayanan
92	RUAS_S_1_46	<p>Spesifikasi Kebutuhan</p> <ul style="list-style-type: none"> ● Sistem menampilkan halaman berisi form untuk mengisi data peminjaman barang ● Data yang harus diisi adalah nim/nip peminjam, nama lembaga, tanggal penggunaan, jam mulai pemakaian barang, jam selesai pemakaian barang, nama kegiatan, keterangan dan nama barang yang dipinjam ● Sistem menyediakan tombol submit untuk membuat peminjaman kelas baru ● Sistem menampilkan pesan peringatan untuk data yang dibiarkan kosong saat menekan tombol submit ● Jam pemakaian dimulai dari pukul 06:00 sampai pukul 21:00
93	RUAS_D_1_47	Sistem menyediakan fungsi tambah sarana peminjaman untuk staff pelayanan
94	RUAS_S_1_47	Fungsi ini ditujukan agar staff pelayanan dapat menginputkan ruangan atau barang yang akan diinputkan dalam proses peminjaman non kelas dan barang
95	RUAS_D_1_48	Sistem menyediakan fungsi melihat <i>history</i> peminjaman untuk sekretariat kuliah, staf pelayanan, validator dan admin.
96	RUAS_S_1_48	<p>Spesifikasi Kebutuhan</p> <ul style="list-style-type: none"> ● Sistem menampilkan history peminjaman dalam bentuk tabel



		<ul style="list-style-type: none"> • Data history peminjaman dibedakan untuk sekretariat kuliah, staf pelayanan dan validator • Sekretariat kuliah hanya melihat data peminjaman kelas sedangkan staf pelayanan dan validator hanya dapat melihat data peminjaman non kelas dan data peminjaman barang • Data ditampilkan dalam bentuk tabel yang terdiri dari kolom id peminjaman, nama peminjaman, tanggal peminjaman, status validasi akademik, kemahasiswaan dan umum.
97	RUAS_D_1_49	Sistem menyediakan fungsi melakukan validasi peminjaman untuk validator
98	RUAS_S_1_49	<p>Spesifikasi Kebutuhan</p> <ul style="list-style-type: none"> • Sistem menampilkan tabel <i>history</i> peminjaman • Sistem menyediakan tombol 'setuju' untuk memvalidasi peminjaman • Validasi dilakukan berurutan selama tiga kali. • Validasi pertama dilakukan oleh validator jenis kasubag akademik • Validasi kedua dilakukan oleh validator jenis kasubag kemahasiswaan • Validasi ketiga dilakukan oleh validator jenis kasubag umum • Sistem menampilkan pesan pemberitahuan jika proses validasi berhasil dilakukan
99	RUAS_D_1_50	Sistem menyediakan fungsi melakukan menolak untuk validator
100	RUAS_S_1_50	<p>Spesifikasi Kebutuhan</p> <ul style="list-style-type: none"> • Sistem menampilkan tabel <i>history</i> peminjaman • Sistem menyediakan tombol 'tolak' untuk menolak peminjaman • Tolak peminjaman dapat dilakukan oleh ketiga jenis validator • Tolak peminjaman hanya dapat dilakukan oleh validator pertama selama validator kedua masih belum melakukan validasi



		<ul style="list-style-type: none"> ● Tolak peminjaman hanya dapat dilakukan oleh validator kedua selama validator ketiga masih belum melakukan validasi ● Validator wajib mengisi alasan penolakan ● Sistem menampilkan pesan pemberitahuan jika proses tolak peminjaman berhasil dilakukan
101	RUAS_D_1_51	Sistem menyediakan fungsi melakukan membatalkan peminjaman untuk sekretariat kuliah dan staf pelayanan
102	RUAS_S_1_51	Spesifikasi Kebutuhan <ul style="list-style-type: none"> ● Sistem menampilkan tabel <i>history</i> peminjaman ● Sistem menyediakan tombol 'batal' untuk membatalkan peminjaman ● Sistem menampilkan pesan pemberitahuan jika proses membatalkan peminjaman berhasil dilakukan
103	RUAS_D_1_52	Sistem menyediakan fungsi <i>edit password</i> untuk admin, staf pelayanan, sekretariat kuliah, dan validator
104	RUAS_S_1_52	Spesifikasi Kebutuhan <ul style="list-style-type: none"> ● Sistem menyediakan form untuk menerima inputan <i>password</i> lama dan <i>password</i> baru ● Sistem menyediakan tombol <i>Edit Password</i> untuk menyimpan <i>password</i> baru ● Sistem menampilkan pesan pemberitahuan jika <i>password</i> telah berhasil diubah
105	RUAS_D_1_53	Sistem menyediakan fungsi menambah <i>complaint</i> baru untuk guest
106	RUAS_S_1_53	Spesifikasi Kebutuhan <ul style="list-style-type: none"> ● Sistem menyediakan form untuk menerima inputan <i>complaint</i> baru ● Data yang harus diinputkan adalah nama pelapor dan isi keluhan ● Sistem menyediakan tombol <i>submit</i> untuk menyimpan <i>complaint</i> baru ● Sistem menampilkan pesan pemberitahuan jika <i>complaint</i> telah berhasil dibuat



107	RUAS_D_1_54	Sistem menyediakan fungsi melihat history <i>complaint</i> untuk staf pelayanan
108	RUAS_S_1_54	Spesifikasi Kebutuhan <ul style="list-style-type: none"> ● Sistem menampilkan <i>history complaint</i> dalam bentuk tabel ● Tabel menampilkan data nama pelapor, isi keluhan dan status <i>complaint</i>
109	RUAS_D_1_55	Sistem menyediakan fungsi mengubah status tinjau <i>complaint</i>
110	RUAS_S_1_55	Spesifikasi Kebutuhan <ul style="list-style-type: none"> ● Sistem menampilkan tombol aksi dengan icon centang untuk menandakan <i>complaint</i> telah ditinjau oleh staf pelayanan ● Sistem menampilkan pemberitahuan jika status <i>complaint</i> telah berhasil diubah
111	RUAS_D_1_56	Sistem menyediakan fungsi untuk melihat rekap <i>complaint</i> untuk admin
112	RUAS_S_1_56	Spesifikasi Kebutuhan <ul style="list-style-type: none"> ● Sistem menampilkan total jumlah <i>complaint</i> dalam perbulan ● Data ditampilkan dalam bentuk tabel yang terdiri dari kolom nama bulan januari sampai desember, dan kolom tahun. Sedangkan baris tabel menampilkan angka jumlah <i>complaint</i> perbulan dan total <i>complaint</i> selama setahun ● Sistem menyediakan fungsi untuk menampilkan rekap berdasarkan tahun
113	RUAS_D_1_57	Sistem menyediakan fungsi <i>logout</i> untuk admin, staf pelayanan, sekretariat kuliah, dan validator
114	RUAS_S_1_57	Spesifikasi Kebutuhan <ul style="list-style-type: none"> ● Sistem menampilkan tombol <i>logout</i> pada menu navigasi
115	RUAS_D_1_58	Sistem menyediakan fungsi untuk melihat grafik data peminjaman dan <i>complaint</i>
116	RUAS_S_1_58	Spesifikasi Kebutuhan



		<ul style="list-style-type: none"> • Grafik terdiri dari grafik jumlah peminjaman, grafik status peminjaman, grafik perbandingan peminjaman dan grafik <i>complaint</i> • Grafik peminjaman dibuat dalam bentuk donat yang menampilkan angka jumlah peminjaman kelas, non kelas, dan barang selama setahun • Grafik status peminjaman dibuat dalam bentuk donat yang menampilkan angka jumlah peminjaman yang telah disetujui, ditolak atau belum diproses. • Grafik perbandingan peminjaman dibuat dalam bentuk diagram batang yang menampilkan jumlah peminjaman kelas, non kelas, dan barang dalam perbulannya <p>Grafik <i>complaint</i> dibuat dalam bentuk diagram batang yang menampilkan jumlah <i>complaint</i> dalam perbulannya.</p>
117	RUAS_D_1_59	Sistem menyediakan fungsi untuk dapat melakukan <i>import</i> data jadwal kuliah dalam bentuk excel untuk admin
118	RUAS_S_1_59	<p>Spesifikasi Kebutuhan</p> <ul style="list-style-type: none"> • Sebelum data di <i>import</i> ke database, sistem menampilkan <i>preview</i> dari data yang akan <i>diimport</i> terlebih dahulu

4.2.3.2 Kebutuhan Non Fungsional

Kebutuhan non fungsional merupakan kemampuan pendukung dimiliki oleh sistem yang membuat sistem dapat berjalan lebih baik. Kebutuhan non fungsional dijelaskan pada Tabel 4.3 berikut.

Tabel 4.3 Kebutuhan Non Fungsional

No	Kode	Nama Kebutuhan	Deskripsi
1	RUAS_S_2_01	<i>Compatibility</i>	Sistem mempunyai tampilan <i>responsive</i> yang dapat menyesuaikan ukuran layar perangkat yang digunakan

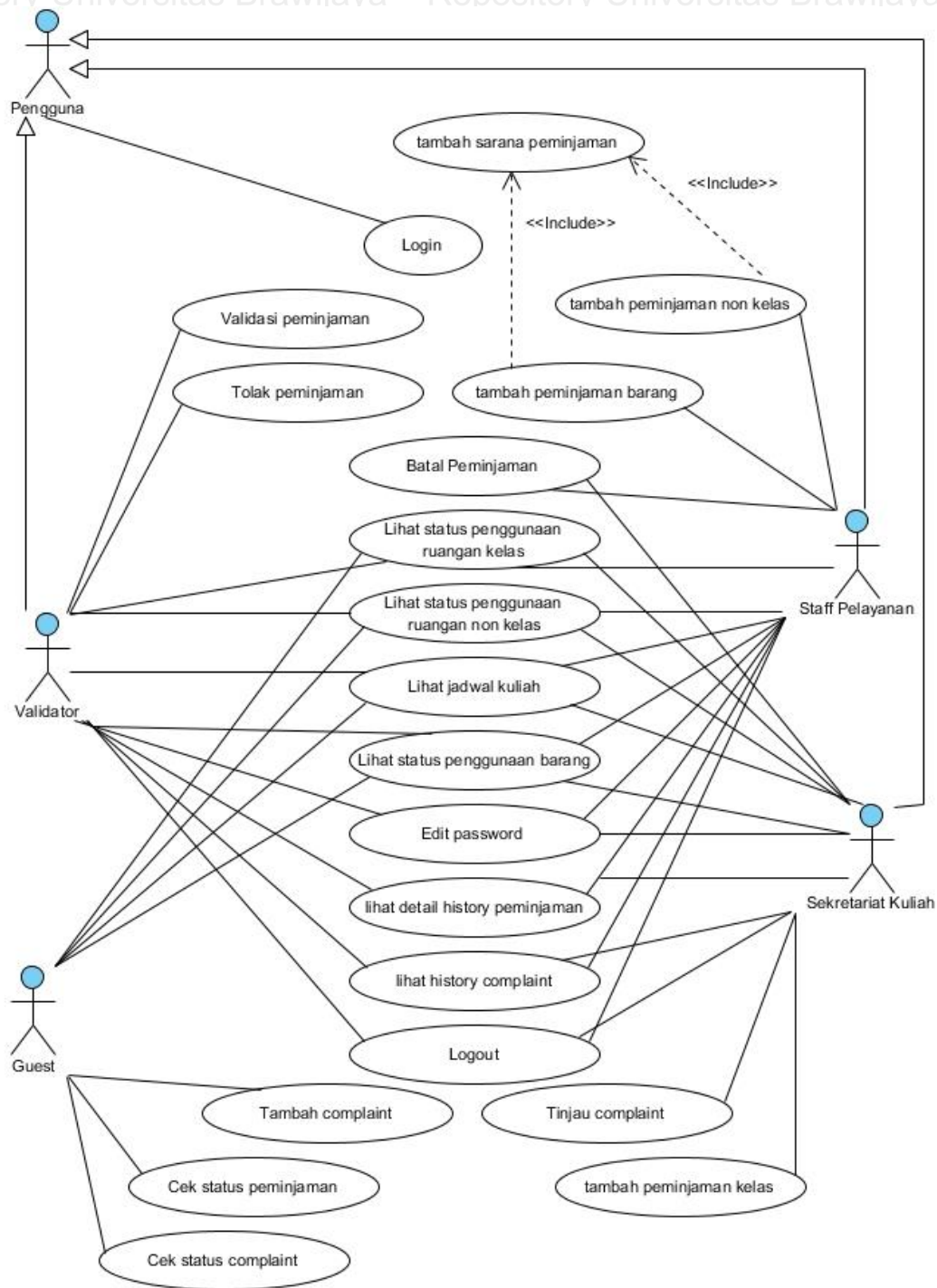


4.3 Pemodelan Kebutuhan

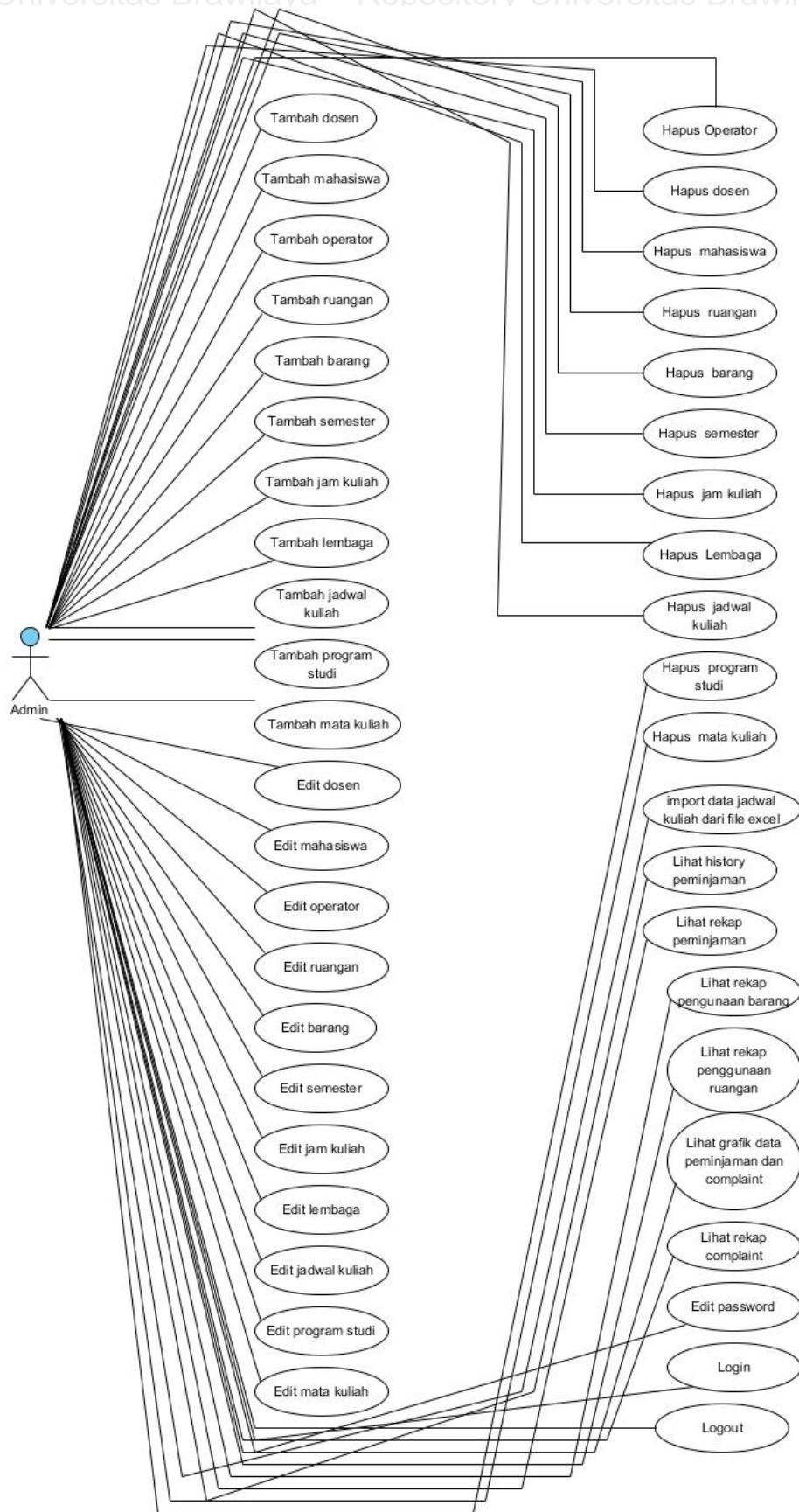
Pemodelan kebutuhan dilakukan setelah melakukan analisis kebutuhan. Pemodelan kebutuhan dibuat untuk menghubungkan antara deskripsi sistem dan model perancangan. Pemodelan kebutuhan dilakukan dengan pendekatan *Object Oriented Design* dengan menggunakan UML (*Unified Modelling Language*). Pemodelan kebutuhan menggunakan UML akan memodelkan kebutuhan ke dalam *use case diagram* dan *use case scenario*.

4.3.1 Use Case Diagram

Use case diagram memodelkan interaksi apa saja yang terjadi di sistem. Interaksi terjadi antara actor dan *use case*. *Use case diagram* sistem manajemen ruang digambarkan pada Gambar 4.10 dan Gambar 4.11. Gambar 4.10 menjelaskan *use case* yang dapat dilakukan oleh aktor *guest*, sekretariat kuliah, staf pelayanan, dan validator. Pada Gambar 4.11 menjelaskan *use case* yang dapat dilakukan oleh aktor admin. *Use case diagram* terdiri dari aktor dan *use case*. Aktor dan *use case* dihubungkan dengan menggunakan garis penghubung.



Gambar 4.10 Use Case Diagram Guest, Validator, Staf Pelayanan, dan Sekretariat Kuliah



Gambar 4.11 Use Case Diagram Admin



4.3.2 Use Case Scenario

Use case scenario menggambarkan alur urutan yang terjadi pada saat aktor melakukan interaksi dengan sistem. *Use case scenario* menerangkan tujuan dari masing masing use case, aktor yang menjalankan, kondisi sebelum menjelaskan alur utama, alur utama, alur alternatif, dan kondisi akhir. Adapun use case scenario sistem manajemen sarana prasarana adalah sebagai berikut.

4.3.2.1 Use Case Scenario Login

Use case scenario login menggambarkan alur proses yang dilakukan aktor untuk dapat login ke sistem. *Use scenario login* dapat dilihat pada Tabel 4.4 berikut.

Tabel 4.4 Use Case Scenario Login

Item	Deskripsi
Objektif	Aktor dapat login ke sistem
Aktor	Pengguna
Kondisi Sebelumnya	Aktor sedang mengakses halaman login
Alur Utama	<ol style="list-style-type: none"> 1. Aktor mengisi username dan password lalu menekan tombol login 2. Sistem menampilkan halaman jadwal kuliah
Alur Alternatif	<p>2a. Terdapat <i>field required</i> yang kosong saat menekan tombol login</p> <p>2a1. Sistem menampilkan field harus diisi</p> <p>2b. Username atau password salah</p> <p>2b1. Sistem menampilkan username atau password salah</p>
Kondisi Akhir	Aktor berhasil login ke sistem



4.3.2.2 Use Case Scenario Tambah Dosen

Use case scenario tambah dosen menggambarkan urutan proses yang dilakukan oleh admin untuk dapat menambahkan data dosen baru ke sistem. *Use case scenario* tambah pegawai dapat dilihat pada Tabel 4.5 berikut.

Tabel 4.5 Use Case Scenario Tambah Dosen

Item	Deskripsi
Objektif	Admin dapat menambah data dosen baru
Aktor	Admin
Kondisi Sebelumnya	Admin sedang mengakses halaman tambah dosen
Alur Utama	<ol style="list-style-type: none"> 1. Admin mengisi form tambah dosen 2. Admin menekan tombol tambah dosen 3. Sistem menyimpan data dosen baru ke sistem dan menampilkan halaman data dosen
Alur Alternatif	<p>3a. Nip yang dimasukkan sudah tersimpan di database</p> <p>3a1. Sistem menampilkan pesan peringatan nip sudah digunakan</p>
Kondisi Akhir	Admin berhasil menambahkan data dosen baru

4.3.2.3 Use Case Scenario Tambah Mahasiswa

Use case scenario tambah mahasiswa menggambarkan urutan proses yang dilakukan oleh admin untuk dapat menambahkan data mahasiswa baru ke sistem. *Use case scenario* tambah mahasiswa dapat dilihat pada Tabel 4.6 berikut.

Tabel 4.6 Use Case Scenario Tambah Mahasiswa

Item	Deskripsi
Objektif	Admin dapat menambah data mahasiswa baru
Aktor	Admin
Kondisi Sebelumnya	Admin sedang mengakses halaman tambah mahasiswa
Alur Utama	<ol style="list-style-type: none"> 1. Admin mengisi form tambah mahasiswa 2. Admin menekan tombol tambah mahasiswa



	3. Sistem menyimpan data mahasiswa baru ke sistem dan menampilkan halaman data mahasiswa
Alur Alternatif	3a. Nim yang dimasukkan sudah tersimpan di database 3a1. Sistem menampilkan pesan peringatan nim sudah digunakan
Kondisi Akhir	Admin berhasil menambahkan data dosen baru

4.3.2.4 Use Case Scenario Tambah Operator

Use case scenario tambah operator menggambarkan urutan proses yang dilakukan oleh admin untuk dapat menambahkan data operator baru ke sistem. *Use case scenario* tambah operator dapat dilihat pada Tabel 4.7 berikut.

Tabel 4.7 Use Case Scenario Tambah Operator

Item	Deskripsi
Objektif	Admin dapat menambah data operator baru
Aktor	Admin
Kondisi Sebelumnya	Admin sedang mengakses halaman tambah operator
Alur Utama	1. Admin mengisi form tambah operator 2. Admin menekan tombol tambah operator 3. Sistem menyimpan data operator baru ke sistem dan menampilkan halaman data operator
Alur Alternatif	2a. Terdapat field yang kosong saat tombol tambah Operator ditekan 2a1. Sistem menampilkan pesan pemberitahuan "isi isian ini" 3a. username yang dimasukkan sudah tersimpan di database 3a1. Sistem menampilkan pesan peringatan username sudah digunakan



Kondisi Akhir	Admin berhasil menambahkan data operator baru
---------------	---

4.3.2.5 Use Case Scenario Tambah Ruangan

Use case scenario tambah ruangan menggambarkan urutan proses yang dilakukan oleh admin untuk dapat menambahkan data ruangan baru ke sistem. *Use case scenario* tambah ruangan dapat dilihat pada Tabel 4.8 berikut.

Tabel 4.8 Use Case Scenario Tambah Ruangan

Item	Deskripsi
Objektif	Admin dapat menambah data ruangan baru
Aktor	Admin
Kondisi Sebelumnya	Admin sedang mengakses halaman tambah ruangan
Alur Utama	<ol style="list-style-type: none"> 1. Admin mengisi form tambah ruangan 2. Admin menekan tombol tambah ruangan 3. Sistem menyimpan data ruangan baru ke sistem dan menampilkan halaman data ruangan
Alur Alternatif	
Kondisi Akhir	Admin berhasil menambahkan data ruangan baru

4.3.2.6 Use Case Scenario Tambah Barang

Use case scenario tambah barang menggambarkan urutan proses yang dilakukan oleh admin untuk dapat menambahkan data barang baru ke sistem. *Use case scenario* tambah barang dapat dilihat pada Tabel 4.9 berikut.

Tabel 4.9 Use Case Scenario Tambah Barang

Item	Deskripsi
Objektif	Admin dapat menambah data barang baru
Aktor	Admin
Kondisi Sebelumnya	Admin sedang mengakses halaman tambah barang
Alur Utama	<ol style="list-style-type: none"> 1. Admin mengisi form tambah barang 2. Admin menekan tombol tambah barang 3. Sistem menyimpan data barang baru ke sistem dan menampilkan halaman data barang



Alur Alternatif	
Kondisi Akhir	Admin berhasil menambahkan data barang baru

4.3.2.7 Use Case Scenario Tambah Semester

Use case scenario tambah semester menggambarkan urutan proses yang dilakukan oleh admin untuk dapat menambahkan data semester baru ke sistem. *Use case scenario* tambah semester dapat dilihat pada Tabel 4.10 berikut.

Tabel 4.10 Use Case Scenario Tambah Semester

Item	Deskripsi
Objektif	Admin dapat menambah data semester baru
Aktor	Admin
Kondisi Sebelumnya	Admin sedang mengakses halaman tambah semester
Alur Utama	<ol style="list-style-type: none"> 1. Admin mengisi form tambah semester 2. Admin menekan tombol tambah semester 3. Sistem menyimpan data semester baru ke sistem dan menampilkan halaman data semester
Alur Alternatif	
Kondisi Akhir	Admin berhasil menambahkan data semester baru

4.3.2.8 Use Case Scenario Tambah Jam Kuliah

Use case scenario tambah jam kuliah menggambarkan urutan proses yang dilakukan oleh admin untuk dapat menambahkan data jam kuliah baru ke sistem. *Use case scenario* tambah jam kuliah dapat dilihat pada Tabel 4.11 berikut.

Tabel 4.11 Use Case Scenario Tambah Jam Kuliah

Item	Deskripsi
Objektif	Admin dapat menambah data jam kuliah baru
Aktor	Admin
Kondisi Sebelumnya	Admin sedang mengakses halaman tambah jam kuliah
Alur Utama	<ol style="list-style-type: none"> 1. Admin mengisi form tambah jam kuliah 2. Admin menekan tombol tambah jam kuliah



	3. Sistem menyimpan data jam kuliah baru ke sistem dan menampilkan halaman data jam kuliah
Alur Alternatif	
Kondisi Akhir	Admin berhasil menambahkan data jam kuliah baru

4.3.2.9 Use Case Scenario Tambah Lembaga

Use case scenario tambah lembaga menggambarkan urutan proses yang dilakukan oleh admin untuk dapat menambahkan data lembaga baru ke sistem. *Use case scenario* tambah lembaga dapat dilihat pada Tabel 4.12 berikut.

Tabel 4.12 Use Case Scenario Tambah Lembaga

Item	Deskripsi
Objektif	Admin dapat menambah data lembaga baru
Aktor	Admin
Kondisi Sebelumnya	Admin sedang mengakses halaman tambah lembaga
Alur Utama	<ol style="list-style-type: none"> 1. Admin mengisi form tambah lembaga 2. Admin menekan tombol tambah lembaga 3. Sistem menyimpan data lembaga baru ke sistem dan menampilkan halaman data lembaga
Alur Alternatif	
Kondisi Akhir	Admin berhasil menambahkan data lembaga baru

4.3.2.10 Use Case Scenario Tambah Jadwal Kuliah

Use case scenario tambah jadwal kuliah menggambarkan urutan proses yang dilakukan oleh admin untuk dapat menambahkan data jadwal kuliah baru ke sistem. *Use case scenario* tambah jadwal kuliah dapat dilihat pada Tabel 4.13 berikut.

Tabel 4.13 Use Case Scenario Tambah Jadwal Kuliah

Item	Deskripsi
Objektif	Admin dapat menambah data jadwal kuliah baru
Aktor	Admin



Kondisi Sebelumnya	Admin sedang mengakses halaman tambah jadwal kuliah
Alur Utama	<ol style="list-style-type: none"> 1. Admin mengisi form tambah jadwal kuliah dan menekan tombol tambah jadwal kuliah 2. Sistem menyimpan data jadwal kuliah baru ke sistem dan menampilkan halaman data jadwal kuliah
Alur Alternatif	<ol style="list-style-type: none"> 1a. Terdapat field yang kosong saat tombol tambah jadwal kuliah ditekan <ol style="list-style-type: none"> 1a1. Sistem menampilkan field harus diisi 2a. Terdapat jadwal kuliah yang sama dengan yang sudah ada di database <ol style="list-style-type: none"> 2a1. Sistem menampilkan pesan jadwal kuliah gagal ditambahkan, jadwal ada yang bertabrakan
Kondisi Akhir	Admin berhasil menambahkan jadwal kuliah baru

4.3.2.11 Use Case Scenario Tambah Program Studi

Use case scenario tambah program studi menggambarkan urutan proses yang dilakukan oleh admin untuk dapat menambahkan data program studi baru ke sistem. *Use case scenario* tambah program studi dapat dilihat pada Tabel 4.14 berikut.

Tabel 4.14 Use Case Scenario Tambah Program Studi

Item	Deskripsi
Objektif	Admin dapat menambah data program studi baru
Aktor	Admin
Kondisi Sebelumnya	Admin sedang mengakses halaman tambah program studi
Alur Utama	<ol style="list-style-type: none"> 1. Admin mengisi form tambah program studi dan menekan tombol tambah program studi 2. Sistem menyimpan data program studi baru ke sistem dan menampilkan halaman data program studi
Alur Alternatif	1a. Terdapat field yang kosong saat tombol tambah



	program studi ditekan 1a1. Sistem menampilkan field harus diisi
Kondisi Akhir	Admin berhasil menambah program studi baru

4.3.2.12 Use Case Scenario Tambah Mata Kuliah

Use case scenario tambah mata kuliah menggambarkan urutan proses yang dilakukan oleh admin untuk dapat menambahkan data mata kuliah baru ke sistem.

Use case scenario tambah mata kuliah dapat dilihat pada Tabel 4.15 berikut.

Tabel 4.15 Use Case Scenario Tambah Mata Kuliah

Item	Deskripsi
Objektif	Admin dapat menambah data mata kuliah baru
Aktor	Admin
Kondisi Sebelumnya	Admin sedang mengakses halaman tambah mata kuliah
Alur Utama	<ol style="list-style-type: none"> Admin mengisi form tambah mata kuliah dan menekan tombol tambah mata kuliah Sistem menyimpan data mata kuliah baru ke sistem dan menampilkan halaman data mata kuliah
Alur Alternatif	<ol style="list-style-type: none"> 1a. Terdapat field yang kosong saat tombol tambah mata kuliah ditekan <ol style="list-style-type: none"> 1a1. Sistem menampilkan field harus diisi 2a. Terdapat kode mata kuliah yang sama dengan yang ada di database <ol style="list-style-type: none"> 2a1. Sistem menampilkan pesan kode mata kuliah sudah digunakan
Kondisi Akhir	Admin berhasil menambah mata kuliah baru



4.3.2.13 Use Case Scenario Edit Dosen

Use case scenario edit dosen menggambarkan urutan proses yang dilakukan oleh admin untuk dapat mengubah data dosen di sistem. *Use case scenario edit* dosen dapat dilihat pada Tabel 4.16 berikut.

Tabel 4.16 Use Case Scenario Edit Dosen

Item	Deskripsi
Objektif	Admin dapat mengubah data dosen
Aktor	Admin
Kondisi Sebelumnya	Admin sedang mengakses halaman <i>edit</i> dosen
Alur Utama	<ol style="list-style-type: none"> Admin mengisi form <i>edit</i> dosen dan menekan tombol edit data dosen Sistem menyimpan data dosen yang baru ke sistem dan menampilkan halaman data dosen
Alur Alternatif	<p>1a. Terdapat <i>field</i> yang kosong saat tombol <i>edit</i> Dosen ditekan</p> <p>1a1. Sistem menampilkan <i>field</i> harus diisi</p>
Kondisi Akhir	Admin berhasil mengubah data dosen

4.3.2.14 Use Case Scenario Edit Mahasiswa

Use case scenario edit mahasiswa menggambarkan urutan proses yang dilakukan oleh admin untuk dapat mengubah data mahasiswa di sistem. *Use case scenario edit* mahasiswa dapat dilihat pada Tabel 4.17 berikut.

Tabel 4.17 Use Case Scenario Edit Mahasiswa

Item	Deskripsi
Objektif	Admin dapat mengubah data mahasiswa
Aktor	Admin
Kondisi Sebelumnya	Admin sedang mengakses halaman <i>edit</i> mahasiswa
Alur Utama	<ol style="list-style-type: none"> Admin mengisi form <i>edit</i> mahasiswa dan menekan tombol edit mahasiswa



	2. Sistem menyimpan data mahasiswa yang baru ke sistem dan menampilkan halaman data mahasiswa
Alur Alternatif	1a. Terdapat <i>field</i> yang kosong saat tombol <i>edit</i> mahasiswa ditekan 1a1. Sistem menampilkan <i>field</i> harus diisi
Kondisi Akhir	Admin berhasil mengubah data mahasiswa

4.3.2.15 Use Case Scenario Edit Operator

Use case scenario edit operator menggambarkan urutan proses yang dilakukan oleh admin untuk dapat mengubah data operator di sistem. *Use case scenario edit operator* dapat dilihat pada Tabel 4.18 berikut.

Tabel 4.18 Use Case Scenario Edit Operator

Item	Deskripsi
Objektif	Admin dapat mengubah data operator
Aktor	Admin
Kondisi Sebelumnya	Admin sedang mengakses halaman <i>edit</i> operator
Alur Utama	1. Admin mengisi form <i>edit</i> operator dan menekan tombol edit data operator 2. Sistem menyimpan data operator yang baru ke sistem dan menampilkan halaman data operator
Alur Alternatif	1a. Terdapat <i>field</i> yang kosong saat tombol <i>edit</i> operator ditekan 1a1. Sistem menampilkan <i>field</i> harus diisi
Kondisi Akhir	Admin berhasil mengubah data operator

4.3.2.16 Use Case Scenario Edit Ruangan

Use case scenario edit ruangan menggambarkan urutan proses yang dilakukan oleh admin untuk dapat mengubah data ruangan di sistem. *Use case scenario edit ruangan* dapat dilihat pada Tabel 4.19 berikut.



Tabel 4.19 Use Case Scenario Edit Ruang

Item	Deskripsi
Objektif	Admin dapat mengubah data ruangan
Aktor	Admin
Kondisi Sebelumnya	Admin sedang mengakses halaman <i>edit</i> ruangan
Alur Utama	<ol style="list-style-type: none"> Admin mengisi form <i>edit</i> ruangan dan menekan tombol edit data ruangan Sistem menyimpan data ruangan yang baru ke sistem dan menampilkan halaman data ruangan
Alur Alternatif	<ol style="list-style-type: none"> 1a. Terdapat <i>field</i> yang kosong saat tombol <i>edit</i> ruangan ditekan <ol style="list-style-type: none"> 1a1. Sistem menampilkan <i>field</i> harus diisi
Kondisi Akhir	Admin berhasil mengubah data ruangan

4.3.2.17 Use Case Scenario Edit Barang

Use case scenario edit barang menggambarkan urutan proses yang dilakukan oleh admin untuk dapat mengubah data barang di sistem. *Use case scenario edit* barang dapat dilihat pada Tabel 4.20 berikut.

Tabel 4.20 Use Case Scenario Edit Barang

Item	Deskripsi
Objektif	Admin dapat mengubah data barang
Aktor	Admin
Kondisi Sebelumnya	Admin sedang mengakses halaman <i>edit</i> barang
Alur Utama	<ol style="list-style-type: none"> Admin mengisi form <i>edit</i> dosen dan menekan tombol edit data barang Sistem menyimpan data barang yang baru ke sistem dan menampilkan halaman data barang
Alur Alternatif	<ol style="list-style-type: none"> 1a. Terdapat <i>field</i> yang kosong saat tombol <i>edit</i> barang ditekan <ol style="list-style-type: none"> 1a1. Sistem menampilkan <i>field</i> harus diisi



Kondisi Akhir	Admin berhasil mengubah data barang
---------------	-------------------------------------

4.3.2.18 Use Case Scenario Edit Semester

Use case scenario edit semester menggambarkan urutan proses yang dilakukan oleh admin untuk dapat mengubah data semester di sistem. *Use case scenario edit semester* dapat dilihat pada Tabel 4.21 berikut.

Tabel 4.21 Use Case Scenario Edit Semester

Item	Deskripsi
Objektif	Admin dapat mengubah data semester
Aktor	Admin
Kondisi Sebelumnya	Admin sedang mengakses halaman <i>edit</i> semester
Alur Utama	1. Admin mengisi form <i>edit</i> dosen dan menekan tombol edit data semester 2. Sistem menyimpan data semester yang baru ke sistem dan menampilkan halaman data semester
Alur Alternatif	1a. Terdapat <i>field</i> yang kosong saat tombol <i>edit</i> semester ditekan 1a1. Sistem menampilkan <i>field</i> harus diisi
Kondisi Akhir	Admin berhasil mengubah data semester

4.3.2.19 Use Case Scenario Edit Jam Kuliah

Use case scenario edit jam kuliah menggambarkan urutan proses yang dilakukan oleh admin untuk dapat mengubah data jam kuliah di sistem. *Use case scenario edit jam kuliah* dapat dilihat pada Tabel 4.22 berikut.

Tabel 4.22 Use Case Scenario Edit Jam Kuliah

Item	Deskripsi
Objektif	Admin dapat mengubah data jam kuliah
Aktor	Admin
Kondisi Sebelumnya	Admin sedang mengakses halaman <i>edit</i> jam kuliah



Alur Utama	<ol style="list-style-type: none"> 1. Admin mengisi form <i>edit</i> dosen dan menekan tombol edit data jam kuliah 2. Sistem menyimpan data jam kuliah yang baru ke sistem dan menampilkan halaman data jam kuliah
Alur Alternatif	<ol style="list-style-type: none"> 1a. Terdapat <i>field</i> yang kosong saat tombol <i>edit</i> jam kuliah ditekan 1a1. Sistem menampilkan <i>field</i> harus diisi
Kondisi Akhir	Admin berhasil mengubah data jam kuliah

4.3.2.20 Use Case Scenario Edit Lembaga

Use case scenario edit lembaga menggambarkan urutan proses yang dilakukan oleh admin untuk dapat mengubah data lembaga di sistem. *Use case scenario edit* lembaga dapat dilihat pada Tabel 4.23 berikut.

Tabel 4.23 Use Case Scenario Edit Lembaga

Item	Deskripsi
Objektif	Admin dapat mengubah data lembaga
Aktor	Admin
Kondisi Sebelumnya	Admin sedang mengakses halaman <i>edit</i> lembaga
Alur Utama	<ol style="list-style-type: none"> 1. Admin mengisi form <i>edit</i> lembaga dan menekan tombol edit data lembaga 2. Sistem menyimpan data lembaga yang baru ke sistem dan menampilkan halaman data lembaga
Alur Alternatif	<ol style="list-style-type: none"> 1a. Terdapat <i>field</i> yang kosong saat tombol <i>edit</i> lembaga ditekan 1a1. Sistem menampilkan <i>field</i> harus diisi
Kondisi Akhir	Admin berhasil mengubah data lembaga



4.3.2.21 Use Case Scenario Edit Jadwal Kuliah

Use case scenario edit jadwal kuliah menggambarkan urutan proses yang dilakukan oleh admin untuk dapat mengubah data jadwal kuliah di sistem. *Use case scenario edit* jadwal kuliah dapat dilihat pada Tabel 4.24 berikut.

Tabel 4.24 Use Case Scenario Edit Jadwal Kuliah

Item	Deskripsi
Objektif	Admin dapat mengubah data jadwal kuliah
Aktor	Admin
Kondisi Sebelumnya	Admin sedang mengakses halaman <i>edit</i> jadwal kuliah
Alur Utama	<ol style="list-style-type: none"> Admin mengisi form <i>edit</i> jadwal kuliah dan menekan tombol edit data jadwal kuliah Sistem menyimpan data jadwal kuliah yang baru ke sistem dan menampilkan halaman data jadwal kuliah
Alur Alternatif	<ol style="list-style-type: none"> 1a. Terdapat <i>field</i> yang kosong saat tombol <i>edit</i> jadwal kuliah ditekan <ol style="list-style-type: none"> 1a1. Sistem menampilkan <i>field</i> harus diisi
Kondisi Akhir	Admin berhasil mengubah data jadwal kuliah

4.3.2.22 Use Case Scenario Edit Program Studi

Use case scenario edit program studi menggambarkan urutan proses yang dilakukan oleh admin untuk dapat mengubah data program studi di sistem. *Use case scenario edit* program studi dapat dilihat pada Tabel 4.25 berikut.

Tabel 4.25 Use Case Scenario Edit Program Studi

Item	Deskripsi
Objektif	Admin dapat mengubah data program studi
Aktor	Admin
Kondisi Sebelumnya	Admin sedang mengakses halaman <i>edit</i> program studi
Alur Utama	<ol style="list-style-type: none"> Admin mengisi form <i>edit</i> program studi dan menekan tombol edit data program studi



	2. Sistem menyimpan data program studi yang baru ke sistem dan menampilkan halaman data program studi
Alur Alternatif	1a. Terdapat <i>field</i> yang kosong saat tombol <i>edit</i> program studi ditekan 1a1. Sistem menampilkan <i>field</i> harus diisi
Kondisi Akhir	Admin berhasil mengubah data program studi

4.3.2.23 Use Case Scenario Edit Mata Kuliah

Use case scenario edit mata kuliah menggambarkan urutan proses yang dilakukan oleh admin untuk dapat mengubah data mata kuliah di sistem. *Use case scenario edit* mata kuliah dapat dilihat pada Tabel 4.26 berikut.

Tabel 4.26 Use Case Scenario Edit Mata Kuliah

Item	Deskripsi
Objektif	Admin dapat mengubah data mata kuliah
Aktor	Admin
Kondisi Sebelumnya	Admin sedang mengakses halaman <i>edit</i> mata kuliah
Alur Utama	1. Admin mengisi form <i>edit</i> lembaga dan menekan tombol edit data mata kuliah 2. Sistem menyimpan data mata kuliah yang baru ke sistem dan menampilkan halaman data mata kuliah
Alur Alternatif	1a. Terdapat <i>field</i> yang kosong saat tombol <i>edit</i> mata kuliah ditekan 1a1. Sistem menampilkan <i>field</i> harus diisi
Kondisi Akhir	Admin berhasil mengubah data mata kuliah



4.3.2.24 Use Case Scenario Hapus Dosen

Use case scenario hapus dosen menggambarkan urutan proses yang dilakukan oleh admin untuk dapat menghapus data dosen dari sistem. *Use case scenario* hapus dosen dapat dilihat pada Tabel 4.27 berikut.

Tabel 4.27 Use Case Scenario Hapus Dosen

Item	Deskripsi
Objektif	Admin dapat menghapus data dosen
Aktor	Admin
Kondisi Sebelumnya	Admin sedang mengakses halaman data dosen
Alur Utama	<ol style="list-style-type: none"> 1. Admin mengklik tombol hapus pada data dosen yang ingin dihapus 2. Sistem menghapus data dosen dan menampilkan halaman data dosen dengan pesan data dosen berhasil dihapus
Alur Alternatif	
Kondisi Akhir	Admin berhasil menghapus data dosen

4.3.2.25 Use Case Scenario Hapus Mahasiswa

Use case scenario hapus mahasiswa menggambarkan urutan proses yang dilakukan oleh admin untuk dapat menghapus data mahasiswa dari sistem. *Use case scenario* hapus mahasiswa dapat dilihat pada Tabel 4.28 berikut.

Tabel 4.28 Use Case Scenario Hapus Mahasiswa

Item	Deskripsi
Objektif	Admin dapat menghapus data mahasiswa
Aktor	Admin
Kondisi Sebelumnya	Admin sedang mengakses halaman data mahasiswa
Alur Utama	<ol style="list-style-type: none"> 1. Admin mengklik tombol hapus pada data mahasiswa yang ingin dihapus 2. Sistem menghapus data mahasiswa dan menampilkan halaman data mahasiswa dengan pesan data mahasiswa berhasil dihapus



Alur Alternatif	
Kondisi Akhir	Admin berhasil menghapus data mahasiswa

4.3.2.26 Use Case Scenario Hapus Operator

Use case scenario hapus operator menggambarkan urutan proses yang dilakukan oleh admin untuk dapat menghapus data operator dari sistem. *Use case scenario* hapus operator dapat dilihat pada Tabel 4.29 berikut.

Tabel 4.29 Use Case Scenario Hapus Operator

Item	Deskripsi
Objektif	Admin dapat menghapus data operator
Aktor	Admin
Kondisi Sebelumnya	Admin sedang mengakses halaman data operator
Alur Utama	<ol style="list-style-type: none"> 1. Admin mengklik tombol hapus pada data operator yang ingin dihapus 2. Sistem menghapus data operator dan menampilkan halaman data operator dengan pesan data operator berhasil dihapus
Alur Alternatif	
Kondisi Akhir	Admin berhasil menghapus data operator

4.3.2.27 Use Case Scenario Hapus Ruangan

Use case scenario hapus ruangan menggambarkan urutan proses yang dilakukan oleh admin untuk dapat menghapus data ruangan dari sistem. *Use case scenario* hapus ruangan dapat dilihat pada Tabel 4.30 berikut.

Tabel 4.30 Use Case Scenario Hapus Ruangan

Item	Deskripsi
Objektif	Admin dapat menghapus data ruangan
Aktor	Admin
Kondisi Sebelumnya	Admin sedang mengakses halaman data ruangan
Alur Utama	<ol style="list-style-type: none"> 1. Admin mengklik tombol hapus pada data ruangan yang ingin dihapus



	2. Sistem menghapus data ruangan dan menampilkan halaman data ruangan dengan pesan pemberitahuan bahwa data ruangan berhasil dihapus
Alur Alternatif	
Kondisi Akhir	Admin berhasil menghapus data ruangan

4.3.2.28 Use Case Scenario Hapus Barang

Use case scenario hapus barang menggambarkan urutan proses yang dilakukan oleh admin untuk dapat menghapus data barang dari sistem. *Use case scenario* hapus barang dapat dilihat pada Tabel 4.31 berikut.

Tabel 4.31 Use Case Scenario Hapus Barang

Item	Deskripsi
Objektif	Admin dapat menghapus data barang
Aktor	Admin
Kondisi Sebelumnya	Admin sedang mengakses halaman data barang
Alur Utama	<ol style="list-style-type: none"> 1. Admin mengklik tombol hapus pada data barang yang ingin dihapus 2. Sistem menghapus data barang dan menampilkan halaman data barang dengan pesan data barang berhasil dihapus
Alur Alternatif	
Kondisi Akhir	Admin berhasil menghapus data barang

4.3.2.29 Use Case Scenario Hapus Semester

Use case scenario hapus semester menggambarkan urutan proses yang dilakukan oleh admin untuk dapat menghapus data semester dari sistem. *Use case scenario* hapus semester dapat dilihat pada Tabel 4.32 berikut.

Tabel 4.32 Use Case Scenario Hapus Semester

Item	Deskripsi
Objektif	Admin dapat menghapus data semester
Aktor	Admin



Kondisi Sebelumnya	Admin sedang mengakses halaman data semester
Alur Utama	<ol style="list-style-type: none"> 1. Admin mengklik tombol hapus pada data semester yang ingin dihapus 2. Sistem menghapus data semester dan menampilkan halaman data semester dengan pesan data semester berhasil dihapus
Alur Alternatif	
Kondisi Akhir	Admin berhasil menghapus data semester

4.3.2.30 Use Case Scenario Hapus Jam Kuliah

Use case scenario hapus jam kuliah menggambarkan urutan proses yang dilakukan oleh admin untuk dapat menghapus data jam kuliah dari sistem. *Use case scenario* hapus jam kuliah dapat dilihat pada Tabel 4.33 berikut.

Tabel 4.33 Use Case Scenario Hapus Jam Kuliah

Item	Deskripsi
Objektif	Admin dapat menghapus data jam kuliah
Aktor	Admin
Kondisi Sebelumnya	Admin sedang mengakses halaman data jam kuliah
Alur Utama	<ol style="list-style-type: none"> 1. Admin mengklik tombol hapus pada data jam kuliah yang ingin dihapus 2. Sistem menghapus data jam kuliah dan menampilkan halaman data jam kuliah dengan pesan data jam kuliah berhasil dihapus
Alur Alternatif	
Kondisi Akhir	Admin berhasil menghapus data jam kuliah



4.3.2.31 Use Case Scenario Hapus Lembaga

Use case scenario hapus lembaga menggambarkan urutan proses yang dilakukan oleh admin untuk dapat menghapus data lembaga dari sistem. *Use case scenario* hapus lembaga dapat dilihat pada Tabel 4.34 berikut.

Tabel 4.34 Use Case Scenario Hapus Lembaga

Item	Deskripsi
Objektif	Admin dapat menghapus data lembaga
Aktor	Admin
Kondisi Sebelumnya	Admin sedang mengakses halaman data lembaga
Alur Utama	<ol style="list-style-type: none"> 1. Admin mengklik tombol hapus pada data lembaga yang ingin dihapus 2. Sistem menghapus data lembaga dan menampilkan halaman data lembaga dengan pesan data lembaga kuliah berhasil dihapus
Alur Alternatif	
Kondisi Akhir	Admin berhasil menghapus data lembaga

4.3.2.32 Use Case Scenario Hapus Jadwal Kuliah

Use case scenario hapus jadwal kuliah menggambarkan urutan proses yang dilakukan oleh admin untuk dapat menghapus data jadwal kuliah dari sistem. *Use case scenario* hapus jadwal kuliah dapat dilihat pada Tabel 4.35 berikut.

Tabel 4.35 Use Case Scenario Hapus Jadwal Kuliah

Item	Deskripsi
Objektif	Admin dapat menghapus data jadwal kuliah
Aktor	Admin
Kondisi Sebelumnya	Admin sedang mengakses halaman data jadwal kuliah
Alur Utama	<ol style="list-style-type: none"> 1. Admin mengklik tombol hapus pada data jadwal kuliah yang ingin dihapus 2. Sistem menghapus data jadwal kuliah dan menampilkan halaman data jadwal kuliah



	dengan pesan data jadwal kuliah kuliah berhasil dihapus
Alur Alternatif	
Kondisi Akhir	Admin berhasil menghapus data jadwal kuliah

4.3.2.33 Use Case Scenario Hapus Program Studi

Use case scenario hapus program studi menggambarkan urutan proses yang dilakukan oleh admin untuk dapat menghapus data program studi dari sistem. *Use case scenario* hapus program studi dapat dilihat pada Tabel 4.36 berikut.

Tabel 4.36 Use Case Scenario Hapus Program Studi

Item	Deskripsi
Objektif	Admin dapat menghapus data program studi
Aktor	Admin
Kondisi Sebelumnya	Admin sedang mengakses halaman data program studi
Alur Utama	<ol style="list-style-type: none"> 1. Admin mengklik tombol hapus pada data program studi yang ingin dihapus 2. Sistem menghapus data program studi dan menampilkan halaman data program studi dengan pesan data program studi berhasil dihapus
Alur Alternatif	
Kondisi Akhir	Admin berhasil menghapus data program studi

4.3.2.34 Use Case Scenario Hapus Mata Kuliah

Use case scenario hapus mata kuliah menggambarkan urutan proses yang dilakukan oleh admin untuk dapat menghapus data mata kuliah dari sistem. *Use case scenario* hapus mata kuliah dapat dilihat pada Tabel 4.37 berikut.

Tabel 4.37 Use Case Scenario Hapus Mata Kuliah

Item	Deskripsi
Objektif	Admin dapat menghapus data mata kuliah
Aktor	Admin



Kondisi Sebelumnya	Admin sedang mengakses halaman data mata kuliah
Alur Utama	<ol style="list-style-type: none"> 1. Admin mengklik tombol hapus pada data mata kuliah yang ingin dihapus 2. Sistem menghapus data mata kuliah dan menampilkan halaman data mata kuliah dengan pesan data mata kuliah berhasil dihapus
Alur Alternatif	
Kondisi Akhir	Admin berhasil menghapus data mata kuliah

4.3.2.35 Use Case Scenario Lihat History Peminjaman

Use case scenario history peminjaman menggambarkan urutan proses yang dilakukan oleh admin untuk dapat melihat *history* peminjaman *Use case scenario* lihat *history* peminjaman dapat dilihat pada Tabel 4.38 berikut.

Tabel 4.38 Use Case Scenario Lihat History Peminjaman

Item	Deskripsi
Objektif	Admin dapat melihat data <i>history</i> peminjaman
Aktor	Admin
Kondisi Sebelumnya	Aktor sudah login dan masuk ke sistem
Alur Utama	<ol style="list-style-type: none"> 1. Aktor mengklik tombol <i>history</i> peminjaman pada menu navigasi 2. Sistem menampilkan halaman <i>history</i> peminjaman
Alur Alternatif	
Kondisi Akhir	Admin melihat data <i>history</i> peminjaman



4.3.2.36 Use Case Scenario Lihat Rekap Peminjaman

Use case scenario lihat rekap peminjaman menggambarkan urutan proses yang dilakukan oleh admin untuk dapat melihat rekap peminjaman *Use case scenario* lihat rekap peminjaman dapat dilihat pada Tabel 4.39 berikut.

Tabel 4.39 Lihat Rekap Peminjaman

Item	Deskripsi
Objektif	Admin dapat melihat data rekap peminjaman
Aktor	Admin
Kondisi Sebelumnya	Admin sudah login dan masuk ke sistem
Alur Utama	<ol style="list-style-type: none"> 1. Admin mengklik tombol rekap peminjaman pada menu navigasi 2. Sistem menampilkan halaman rekap peminjaman
Alur Alternatif	
Kondisi Akhir	Admin melihat data rekap peminjaman

4.3.2.37 Use Case Scenario Lihat Status Penggunaan Ruang Kelas

Use case scenario lihat status penggunaan ruangan kelas menggambarkan urutan proses yang dilakukan oleh pengguna untuk dapat melihat status penggunaan ruangan kelas. *Use case scenario* lihat status penggunaan ruangan kelas dapat dilihat pada Tabel 4.40 berikut.

Tabel 4.40 Use Case Scenario Lihat Status Penggunaan Ruang Kelas

Item	Deskripsi
Objektif	Pengguna dapat melihat status penggunaan ruangan kelas
Aktor	Pengguna
Kondisi Sebelumnya	Pengguna sedang mengakses sistem
Alur Utama	<ol style="list-style-type: none"> 1. Pengguna mengklik menu ruangan kelas pada navbar sistem 2. Sistem menampilkan halaman penggunaan ruangan kelas



Alur Alternatif	
Kondisi Akhir	Pengguna status penggunaan ruangan kelas

4.3.2.38 Use Case Scenario Lihat Status Penggunaan Ruang Non Kelas

Use case scenario lihat status penggunaan ruangan kelas menggambarkan urutan proses yang dilakukan oleh pengguna untuk dapat melihat status penggunaan ruangan non kelas. *Use case scenario* lihat status penggunaan ruangan non kelas dapat dilihat pada Tabel 4.41 berikut.

Tabel 4.41 Use Case Scenario Lihat Status Penggunaan Ruang Non Kelas

Item	Deskripsi
Objektif	Pengguna dapat melihat status penggunaan ruangan non kelas
Aktor	Pengguna
Kondisi Sebelumnya	Pengguna sedang mengakses sistem
Alur Utama	<ol style="list-style-type: none"> 1. Pengguna mengklik menu ruangan non kelas pada navbar sistem 2. Sistem menampilkan halaman penggunaan ruangan non kelas
Alur Alternatif	
Kondisi Akhir	Pengguna status penggunaan ruangan non kelas

4.3.2.39 Use Case Scenario Lihat Status Penggunaan Barang

Use case scenario lihat status penggunaan ruangan kelas menggambarkan urutan proses yang dilakukan oleh pengguna untuk dapat melihat status penggunaan ruangan non kelas. *Use case scenario* lihat status penggunaan ruangan non kelas dapat dilihat pada Tabel 4.42 berikut.

Tabel 4.42 Use Case Scenario Lihat Status Penggunaan Barang

Item	Deskripsi
Objektif	Pengguna dapat melihat status penggunaan ruangan non kelas
Aktor	Pengguna



Kondisi Sebelumnya	Pengguna sedang mengakses sistem
Alur Utama	<ol style="list-style-type: none"> 1. Pengguna mengklik menu ruangan non kelas pada navbar sistem 2. Sistem menampilkan halaman penggunaan ruangan non kelas
Alur Alternatif	
Kondisi Akhir	Pengguna status penggunaan ruangan non kelas

4.3.2.40 Use Case Scenario Lihat Jadwal Kuliah

Use case scenario lihat jadwal kuliah menggambarkan urutan proses yang dilakukan oleh pengguna untuk dapat melihat jadwal kuliah. *Use case scenario* lihat jadwal kuliah dapat dilihat pada Tabel 4.43 berikut.

Tabel 4.43 Use Case Scenario Lihat Jadwal Kuliah

Item	Deskripsi
Objektif	Pengguna dapat melihat jadwal kuliah
Aktor	Pengguna
Kondisi Sebelumnya	Pengguna sedang mengakses sistem
Alur Utama	<ol style="list-style-type: none"> 1. Pengguna mengklik menu jadwal kuliah pada navbar sistem 2. Sistem menampilkan halaman jadwal kuliah
Alur Alternatif	
Kondisi Akhir	Pengguna melihat jadwal kuliah



4.3.2.41 Use Case Scenario Cek Status Complaint

Use case scenario cek status *complaint* menggambarkan urutan proses yang dilakukan oleh *guest* untuk dapat cek status *complaint*. *Use case scenario* cek status *complaint* dapat dilihat pada Tabel 4.45 berikut.

Tabel 4.44 Use Case Scenario Cek Status Complaint

Item	Deskripsi
Objektif	<i>Guest</i> dapat mengecek status peminjaman
Aktor	<i>Guest</i>
Kondisi Sebelumnya	<i>Guest</i> sedang mengakses halaman cek status peminjaman
Alur Utama	<ol style="list-style-type: none"> 1. <i>Guest</i> mengisi <i>form</i> cek status peminjaman dan menekan tombol cek status peminjaman 2. Sistem menampilkan data status peminjaman pada halaman detail peminjaman
Alur Alternatif	<ol style="list-style-type: none"> 1a. Terdapat <i>field</i> yang kosong 1a1. Sistem menampilkan pesan <i>field</i> harus diisi
Kondisi Akhir	<i>Guest</i> berhasil melakukan pengecekan status peminjaman

4.3.2.42 Use Case Scenario Cek Status Peminjaman

Use case scenario cek status peminjaman menggambarkan urutan proses yang dilakukan oleh *guest* untuk dapat melakukan pengecekan status peminjaman. *Use case scenario* cek status peminjaman dapat dilihat pada Tabel 4.45 berikut.

Tabel 4.45 Use Case Scenario Cek Status Peminjaman

Item	Deskripsi
Objektif	<i>Guest</i> dapat mengecek status peminjaman
Aktor	<i>Guest</i>
Kondisi Sebelumnya	<i>Guest</i> sedang mengakses halaman cek peminjaman
Alur Utama	<ol style="list-style-type: none"> 1. Sistem menampilkan halaman cek status peminjaman yang memuat formulir cek peminjaman



	<ol style="list-style-type: none"> 2. Guest mengisi form cek peminjaman yang terdiri dari id peminjaman 3. Guest menekan tombol cek status peminjaman 4. Sistem melakukan pengecekan form cek peminjaman sudah diisi lengkap atau tidak 5. Sistem melakukan pengecekan validitas data peminjaman berdasarkan id peminjaman 6. Sistem mengambil data peminjaman sesuai id peminjaman dari data peminjaman 7. Sistem mengambil data sarana peminjaman berdasarkan id peminjaman 8. Sistem menampilkan data peminjaman pada halaman detail peminjaman
Alur Alternatif	<p>4a. Terdapat <i>field</i> yang kosong saat tombol cek status peminjaman di tekan</p> <p>4a1. Sistem menampilkan halaman cek peminjaman dengan pesan peringatan field kosong harus diisi</p> <p>5a. Id peminjaman tidak ditemukan pada data peminjaman</p> <p>5a1. Sistem menampilkan halaman cek peminjaman dengan pesan pemberitahuan id peminjaman tidak ditemukan</p>
Kondisi Akhir	Guest berhasil melakukan pengecekan status peminjaman

4.3.2.43 Use Case Scenario Validasi Peminjaman

Use case scenario validasi peminjaman menggambarkan urutan proses yang dilakukan oleh validator untuk dapat melakukan validasi peminjaman. *Use case scenario* validasi peminjaman dapat dilihat pada Tabel 4.46 berikut.

Tabel 4.46 Use Case Scenario Validasi Peminjaman

Item	Deskripsi
Objektif	Validator dapat melakukan validasi peminjaman
Aktor	Validator
Kondisi Sebelumnya	Validator sedang mengakses halaman <i>history</i> peminjaman



Alur Utama	<ol style="list-style-type: none"> 1. Validator menekan tombol setuju pada peminjaman yang ingin divalidasi 2. Sistem mengubah status peminjaman yang divalidasi dan menampilkan halaman <i>history</i> peminjaman
Alur Alternatif	
Kondisi Akhir	Admin berhasil melakukan validasi peminjaman

4.3.2.44 Use Case Scenario Tolak Peminjaman

Use case scenario tolak peminjaman menggambarkan urutan proses yang dilakukan oleh validator untuk dapat melakukan penolakan peminjaman. *Use case scenario* tolak peminjaman dapat dilihat pada Tabel 4.47 berikut.

Tabel 4.47 Use Case Scenario Tolak Peminjaman

Item	Deskripsi
Objektif	Validator dapat melakukan tolak peminjaman
Aktor	Validator
Kondisi Sebelumnya	Validator sedang mengakses halaman <i>history</i> peminjaman
Alur Utama	<ol style="list-style-type: none"> 1. Validator menekan tombol tolak pada peminjaman yang ingin ditolak 2. Sistem mengubah status peminjaman yang di unvalidasi dan menampilkan halaman <i>history</i> peminjaman
Alur Alternatif	
Kondisi Akhir	Admin berhasil melakukan tolak peminjaman



4.3.2.45 Use Case Scenario Batal Peminjaman

Use case scenario tolak peminjaman menggambarkan urutan proses yang dilakukan oleh sekretariat kuliah dan staf pelayanan untuk dapat melakukan membatalkan peminjaman. *Use case scenario* batal peminjaman dapat dilihat pada Tabel 4.48 berikut.

Tabel 4.48 Use Case Scenario Batal Peminjaman

Item	Deskripsi
Objektif	Validator dapat membatalkan peminjaman
Aktor	Validator
Kondisi Sebelumnya	Validator sedang mengakses halaman <i>history</i> peminjaman
Alur Utama	3. Validator menekan tombol batal pada peminjaman yang ingin dibatalkan 4. Sistem menghapus data peminjaman yang dibatalkan dari database dan menampilkan halaman <i>history</i> peminjaman
Alur Alternatif	
Kondisi Akhir	Admin berhasil melakukan batal peminjaman

4.3.2.46 Use Case Scenario Tambah Peminjaman Kelas

Use case scenario tambah peminjaman kelas menggambarkan urutan proses yang dilakukan oleh sekretariat kuliah untuk dapat melakukan tambah peminjaman kelas. *Use case scenario* tambah peminjaman kelas dapat dilihat pada Tabel 4.49 berikut.

Tabel 4.49 Use Case Scenario Tambah Peminjaman Kelas

Item	Deskripsi
Objektif	Sekretariat kuliah dapat melakukan tambah peminjaman kelas
Aktor	Sekretariat kuliah
Kondisi Sebelumnya	Sekretariat kuliah sedang mengakses halaman status penggunaan ruangan kelas
Alur Utama	1. Sistem menampilkan halaman peta ruangan kelas yang berisi informasi tanggal, ruangan dan jam kuliah



	<ol style="list-style-type: none"> 2. Sekretariat kuliah menekan tombol dengan <i>icon</i> centang pada halaman peta ruangan kelas 3. Sistem melakukan pengecekan pemakaian ruangan berdasarkan hari, ruangan dan jam kuliah dari data jadwal kuliah reguler 4. Sistem melakukan pengecekan pemakaian ruangan berdasarkan tanggal, ruangan dan jam dari data peminjaman kelas 5. Sistem menampilkan halaman tambah peminjaman kelas yang berisi form tambah peminjaman kelas 6. Sekretariat kuliah mengisi <i>form</i> tambah peminjaman kelas yang terdiri dari nim, kode mata kuliah, nip dosen, tanggal penggunaan, jam kuliah, ruangan, program studi, kelas dan keterangan. Sekretariat kuliah kemudian menekan tombol tambah peminjaman kelas 7. Sistem melakukan pengecekan form tambah peminjaman kelas sudah diisi lengkap atau tidak 8. Sistem melakukan pengecekan validitas data mahasiswa berdasarkan nim 9. Sistem menyimpan data peminjaman kelas baru dan menampilkan halaman <i>history</i> peminjaman dengan pesan peminjaman kelas telah berhasil ditambahkan.
Alur Alternatif	<p>3a. Terdapat data peminjaman tanggal, id ruangan dan id jam kuliah yang bertabrakan dengan data pada model jadwal kuliah</p> <p>3a1. Sistem menampilkan halaman peta ruangan kelas dengan pesan pemberitahuan “ruangan tidak dapat dipinjam”</p> <p>4a. Terdapat data peminjaman tanggal, id ruangan dan id jam kuliah yang bertabrakan dengan data pada model peminjaman</p> <p>4a1. Sistem menampilkan halaman peta ruangan kelas dengan pesan pemberitahuan “ruangan tidak dapat dipinjam”</p>



	<p>7a. Terdapat field yang kosong saat tombol tambah Peminjaman kelas ditekan</p> <p>7a1. Sistem menampilkan halaman tambah peminjaman kelas dan menampilkan pesan field harus diisi</p> <p>8a. nim mahasiswa yang diinput tidak ditemukan pada data mahasiswa</p> <p>8a1. Sistem menampilkan halaman tambah peminjaman kelas dengan pesan “data mahasiswa tidak ditemukan”</p>
Kondisi Akhir	Sekretariat kuliah berhasil melakukan tambah peminjaman kelas

4.3.2.47 Use Case Scenario Tambah Peminjaman Non Kelas

Use case scenario tambah peminjaman non kelas menggambarkan urutan proses yang dilakukan oleh staf pelayanan untuk dapat melakukan tambah peminjaman non kelas. *Use case scenario* tambah peminjaman non kelas dapat dilihat pada Tabel 4.50 berikut.

Tabel 4.50 Use Case Scenario Tambah Peminjaman Non Kelas

Item	Deskripsi
Objektif	Staf pelayanan dapat melakukan tambah peminjaman non kelas
Aktor	Staf pelayanan
Kondisi Sebelumnya	Staf pelayanan sedang mengakses halaman tambah peminjaman kelas
Alur Utama	<ol style="list-style-type: none"> 1. Staf pelayanan mengisi <i>form</i> tambah peminjaman non kelas yang terdiri dari nim peminjam, tanggal mulai penggunaan, tanggal selesai penggunaan, jam mulai, jam selesai, lembaga dan keterangan. 2. Staf pelayanan menekan tombol tambah peminjaman non kelas 3. Sistem melakukan pengecekan form tambah peminjaman barang sudah diisi lengkap atau tidak



	<p>4. Sistem melakukan pengecekan id mahasiswa berdasarkan data mahasiswa</p> <p>5. Sistem menyimpan data peminjaman non kelas baru dan menampilkan halaman tambah sarana peminjaman barang berisi daftar ruangan yang dapat dipinjam</p> <p>6. Menjalankan <i>use case</i> tambah sarana peminjaman</p>
Alur Alternatif	<p>3a. Terdapat field yang kosong saat tombol tambah peminjaman non kelas ditekan</p> <p>3a1. Sistem menampilkan halaman tambah peminjaman non kelas dengan pesan pemberitahuan field kosong harus diisi</p> <p>4a. id mahasiswa yang diinput tidak ditemukan pada data mahasiswa</p> <p>4a1. Sistem menampilkan halaman tambah peminjaman non kelas dengan pesan pemberitahuan id mahasiswa tidak ditemukan</p>
Kondisi Akhir	Staf pelayanan berhasil melakukan tambah peminjaman non kelas

4.3.2.48 Use Case Scenario Tambah Peminjaman Barang

Use case scenario tambah peminjaman barang menggambarkan urutan proses yang dilakukan oleh staf pelayanan untuk dapat melakukan tambah peminjaman barang. *Use case scenario* tambah peminjaman barang dapat dilihat pada Tabel 4.51 berikut.

Tabel 4.51 Use Case Scenario Tambah Peminjaman Barang

Item	Deskripsi
Objektif	Staf pelayanan dapat melakukan tambah peminjaman barang
Aktor	Staf pelayanan
Kondisi Sebelumnya	Staf pelayanan sedang mengakses halaman tambah peminjaman barang
Alur Utama	1. Staf pelayanan mengisi <i>form</i> tambah peminjaman barang yang terdiri dari nim



	<p>peminjam, tanggal mulai penggunaan, tanggal selesai penggunaan, jam mulai, jam selesai, lembaga dan keterangan.</p> <p>2. Staf pelayanan menekan tombol tambah peminjaman barang</p> <p>3. Sistem melakukan pengecekan form sudah diisi lengkap atau tidak</p> <p>4. Sistem melakukan pengecekan validitas data mahasiswa berdasarkan nim</p> <p>5. Sistem menyimpan data peminjaman barang baru dan menampilkan halaman tambah sarana peminjaman barang berisi daftar barang yang dapat dipinjam</p> <p>6. Menjalankan <i>use case</i> tambah sarana peminjaman</p>
Alur Alternatif	<p>3a. Terdapat field yang kosong saat tombol tambah peminjaman barang ditekan</p> <p>3a1. Sistem menampilkan halaman tambah peminjaman barang dengan pesan pemberitahuan field kosong harus diisi</p> <p>4a. id mahasiswa yang diinput tidak ditemukan pada data mahasiswa</p> <p>4a1. Sistem menampilkan halaman tambah peminjaman barang dengan pesan pemberitahuan id mahasiswa tidak ditemukan</p>
Kondisi Akhir	Staf pelayanan berhasil menambahkan peminjaman barang baru



4.3.2.49 Use Case Scenario Tambah Sarana Peminjaman

Use case scenario tambah sarana peminjaman menggambarkan urutan proses yang dilakukan oleh staf pelayanan untuk dapat melakukan penambahan ruangan atau barang yang dilakukan dalam melakukan proses peminjaman barang. *Use case scenario* tambah sarana peminjaman dapat dilihat pada Tabel 4.52 berikut.

Tabel 4.52 Use Case Scenario Tambah Sarana Peminjaman

Item	Deskripsi
Objektif	Staf pelayanan dapat melakukan tambah sarana peminjaman
Aktor	Staf pelayanan
Kondisi Sebelumnya	Staf pelayanan telah melakukan proses tambah peminjaman barang dan berada di halaman tambah sarana peminjaman
Alur Utama	<ol style="list-style-type: none"> 1. Staf pelayanan memilih ruangan atau barang yang akan dipinjam 2. Sistem menyimpan data peminjaman baru dan menampilkan halaman tambah sarana peminjaman barang yang berisi daftar barang yang dapat dipinjam
Alur Alternatif	
Kondisi Akhir	Staf pelayanan berhasil menambahkan ruangan atau barang yang akan dipinjamkan

4.3.2.50 Use Case Scenario Tambah Complaint

Use case scenario tambah *complaint* menggambarkan urutan proses yang dilakukan oleh guest untuk dapat membuat *complaint* baru. *Use case scenario* tambah *complaint* dapat dilihat pada Tabel 4.53 berikut.

Tabel 4.53 Use Case Scenario Tambah Complaint

Item	Deskripsi
Objektif	Guest dapat membuat <i>complaint</i> baru
Aktor	Guest
Kondisi Sebelumnya	Guest sedang mengakses halaman tambah <i>complaint</i>



Alur Utama	<ol style="list-style-type: none"> 1. Guest mengisi form tambah <i>complaint</i> dan menekan tombol tambah <i>complaint</i> 2. Sistem menyimpan <i>complaint</i> baru dan menampilkan halaman tambah <i>complaint</i> dengan pesan <i>complaint</i> telah berhasil dikirimkan
Alur Alternatif	<ol style="list-style-type: none"> 1a. Terdapat field yang kosong saat tombol tambah <i>Complaint</i> ditekan 1a1. Sistem menampilkan pesan field harus diisi
Kondisi Akhir	Guest telah membuat <i>complaint</i> baru

4.3.2.51 Use Case Scenario Lihat History Complaint

Use case scenario lihat *history complaint* menggambarkan urutan proses yang dilakukan oleh admin dan staf pelayanan untuk dapat melihat *history complaint*. *Use case scenario* lihat *history complaint* dapat dilihat pada Tabel 4.54 berikut.

Tabel 4.54 Use Case Scenario Lihat History Complaint

Item	Deskripsi
Objektif	admin, staf pelayanan dapat melihat <i>history complaint</i>
Aktor	Admin dan staf pelayanan
Kondisi Sebelumnya	Admin, staf pelayanan sudah <i>login</i> ke sistem
Alur Utama	<ol style="list-style-type: none"> 1. Admin, staf pelayanan mengklik menu <i>history complaint</i> pada navbar sistem 2. Sistem menampilkan halaman <i>history complaint</i>
Alur Alternatif	
Kondisi Akhir	Admin dan staf pelayanan melihat data <i>history complaint</i>



4.3.2.52 Use Case Scenario Tinjau Complaint

Use case scenario tinjau complaint menggambarkan urutan proses yang dilakukan oleh staf pelayanan untuk dapat mengubah status *complaint* menjadi telah ditinjau. *Use case scenario* ubah status *complaint* dapat dilihat pada Tabel 4.55 berikut.

Tabel 4.55 Use Case Scenario Tinjau Complaint

Item	Deskripsi
Objektif	admin, staf pelayanan dapat mengubah status <i>complaint</i>
Aktor	Staf pelayanan
Kondisi Sebelumnya	Staf pelayanan sedang mengakses halaman <i>history complaint</i>
Alur Utama	<ol style="list-style-type: none"> 1. Staf pelayanan menekan tombol selesai pada <i>complaint</i> yang dipilih 2. Sistem mengubah status <i>complaint</i> menjadi selesai dan menampilkan halaman <i>history complaint</i>
Alur Alternatif	
Kondisi Akhir	Staf pelayanan mengubah status <i>complaint</i>

4.3.2.53 Use Case Scenario Lihat Rekap Penggunaan Ruangan

Use case scenario lihat rekap penggunaan ruangan menggambarkan urutan proses yang dilakukan oleh admin untuk dapat melihat rekap penggunaan ruangan. *Use case scenario* lihat rekap peminjaman dapat dilihat pada Tabel 4.56 berikut.

Tabel 4.56 Use Case Scenario Lihat Rekap Penggunaan Ruangan

Item	Deskripsi
Objektif	Admin dapat melihat data rekap penggunaan ruangan
Aktor	Admin
Kondisi Sebelumnya	Admin sudah login dan masuk ke sistem
Alur Utama	<ol style="list-style-type: none"> 1. Admin mengklik tombol rekap penggunaan ruangan pada menu navigasi



	2. Sistem menampilkan halaman rekap penggunaan ruangan
Alur Alternatif	
Kondisi Akhir	Admin melihat data rekap penggunaan ruangan

4.3.2.54 Use Case Scenario Lihat Rekap Penggunaan Barang

Use case scenario lihat rekap penggunaan barang menggambarkan urutan proses yang dilakukan oleh admin untuk dapat melihat rekap penggunaan barang. *Use case scenario* lihat rekap peminjaman dapat dilihat pada Tabel 4.57 berikut.

Tabel 4.57 Use Case Scenario Lihat Rekap Penggunaan Barang

Item	Deskripsi
Objektif	Admin dapat melihat data rekap penggunaan barang
Aktor	Admin
Kondisi Sebelumnya	Admin sudah login dan masuk ke sistem
Alur Utama	1. Admin mengklik tombol rekap penggunaan barang pada menu navigasi 2. Sistem menampilkan halaman rekap penggunaan barang
Alur Alternatif	
Kondisi Akhir	Admin melihat data rekap penggunaan barang

4.3.2.55 Use Case Scenario Lihat Rekap Complaint

Use case scenario lihat rekap complaint menggambarkan urutan proses yang dilakukan oleh admin untuk dapat melihat rekap peminjaman *Use case scenario* lihat rekap complaint dapat dilihat pada Tabel 4.58 berikut.

Tabel 4.58 Use Case Scenario Lihat Rekap Complaint

Item	Deskripsi
Objektif	Admin dapat melihat data rekap <i>complaint</i>
Aktor	Admin
Kondisi Sebelumnya	Admin sudah <i>login</i> dan masuk ke sistem



Alur Utama	<ol style="list-style-type: none"> Admin mengklik tombol rekap complaint pada menu navigasi Sistem menampilkan halaman rekap complaint
Alur Alternatif	
Kondisi Akhir	Admin melihat data rekap <i>complaint</i>

4.3.2.56 Use Case Scenario Edit Password

Use case scenario edit password menggambarkan urutan proses yang dilakukan oleh admin, validator, staff pelayanan, dan sekretariat kuliah untuk dapat mengubah password. *Use case scenario edit password* dapat dilihat pada Tabel 4.59 berikut.

Tabel 4.59 Use Case Scenario Edit Password

Item	Deskripsi
Objektif	admin, validator, staff pelayanan, dan sekretariat kuliah dapat mengubah <i>password</i>
Aktor	Admin, validator, staff pelayanan, dan sekretariat kuliah
Kondisi Sebelumnya	Admin, validator, staff pelayanan, dan sekretariat kuliah sedang mengakses halaman <i>edit password</i>
Alur Utama	<ol style="list-style-type: none"> Admin, validator, staff pelayanan, dan sekretariat kuliah mengisi form <i>edit password</i> Sistem menyimpan <i>password</i> baru dan menampilkan halaman <i>edit password</i> dengan pesan <i>password</i> telah diganti
Alur Alternatif	<ol style="list-style-type: none"> terdapat <i>field</i> kosong saat tombol <i>edit password</i> ditekan <ol style="list-style-type: none"> Sistem menampilkan pesan <i>field</i> harus diisi
Kondisi Akhir	admin, validator, staff pelayanan, dan sekretariat kuliah berhasil mengubah password

4.3.2.57 Use Case Scenario Import Data Excel

Use case scenario import data excel menggambarkan urutan proses yang dilakukan oleh admin untuk dapat melakukan import data dalam file excel ke sistem. *Use case scenario import data excel* dapat dilihat pada Tabel 4.60 berikut.



Tabel 4.60 Use Case Scenario Import Data Excel

Item	Deskripsi
Objektif	Admin dapat melakukan import data jadwal kuliah berbentuk file excel ke sistem
Aktor	Admin
Kondisi Sebelumnya	Admin sedang mengakses halaman <i>import</i> jadwal kuliah
Alur Utama	<ol style="list-style-type: none"> 1. Admin memilih file excel dan menekan tombol preview 2. Sistem menampilkan preview dari data yang berada dalam file excel 3. Admin menekan tombol import 4. Sistem menyimpan data jadwal kuliah dan menampilkan halaman jadwal kuliah dengan pesan jadwal kuliah telah berhasil diimport
Alur Alternatif	<ol style="list-style-type: none"> 1a. File excel tidak dipilih saat tombol preview ditekan <ol style="list-style-type: none"> 1a1. Sistem menampilkan pesan pemberitahuan tidak ada yang dipilih 2a. File yang diimport tidak sesuai dengan format <ol style="list-style-type: none"> 2a1. Sistem menampilkan pesan import gagal, kolom tidak sesuai
Kondisi Akhir	Admin berhasil melakukan <i>import</i> data jadwal kuliah dalam bentuk file excel



4.3.2.58 Use Case Scenario Lihat Grafik Rekap Data Peminjaman dan Complaint

Use case scenario import data excel menggambarkan urutan proses yang dilakukan oleh admin untuk dapat melakukan import data dalam file excel ke sistem. *Use case scenario import data excel* dapat dilihat pada Tabel 4.61 berikut.

Tabel 4.61 Use Case Scenario Lihat Grafik Data Peminjaman dan Complaint

Item	Deskripsi
Objektif	Admin dapat melihat grafik data peminjaman dan <i>complaint</i>
Aktor	Admin
Kondisi Sebelumnya	Admin sudah login
Alur Utama	<ol style="list-style-type: none"> Admin memilih menu dashboard yang ada pada navigasi Sistem menampilkan halaman dashboard yang berisi grafik peminjaman dan <i>complaint</i>
Alur Alternatif	-
Kondisi Akhir	Admin melihat grafik data peminjaman dan <i>complaint</i>

4.3.2.59 Use Case Scenario Logout

Use case scenario logout menggambarkan urutan proses yang dilakukan oleh admin, validator, staff pelayanan, dan sekretariat kuliah untuk dapat *logout* dari sistem. *Use case scenario logout* dapat dilihat pada Tabel 4.62 berikut.

Tabel 4.62 Use Case Scenario Logout

Item	Deskripsi
Objektif	admin, validator, staff pelayanan, dan sekretariat kuliah dapat logout dari sistem
Aktor	Admin, validator, staff pelayanan, dan sekretariat kuliah
Kondisi Sebelumnya	Admin, validator, staff pelayanan, dan sekretariat kuliah sudah <i>login</i> dan mengakses sistem
Alur Utama	<ol style="list-style-type: none"> Admin, validator, staff pelayanan, dan sekretariat kuliah menekan tombol <i>logout</i> pada menu navigasi



	2. Sistem mengeluarkan akun yang login dari sistem
Alur Alternatif	
Kondisi Akhir	admin, validator, staff pelayanan, dan sekretariat kuliah berhasil <i>logout</i> dari sistem



BAB 5 PERANCANGAN DAN IMPLEMENTASI

5.1 Perancangan Perangkat Lunak

Pada tahap ini perangkat lunak perancangan yang dilakukan adalah perancangan alur proses program menggunakan *sequence diagram*, perancangan kelas menggunakan *class diagram*, perancangan komponen, dan perancangan antar muka.

5.1.1 Sequence Diagram

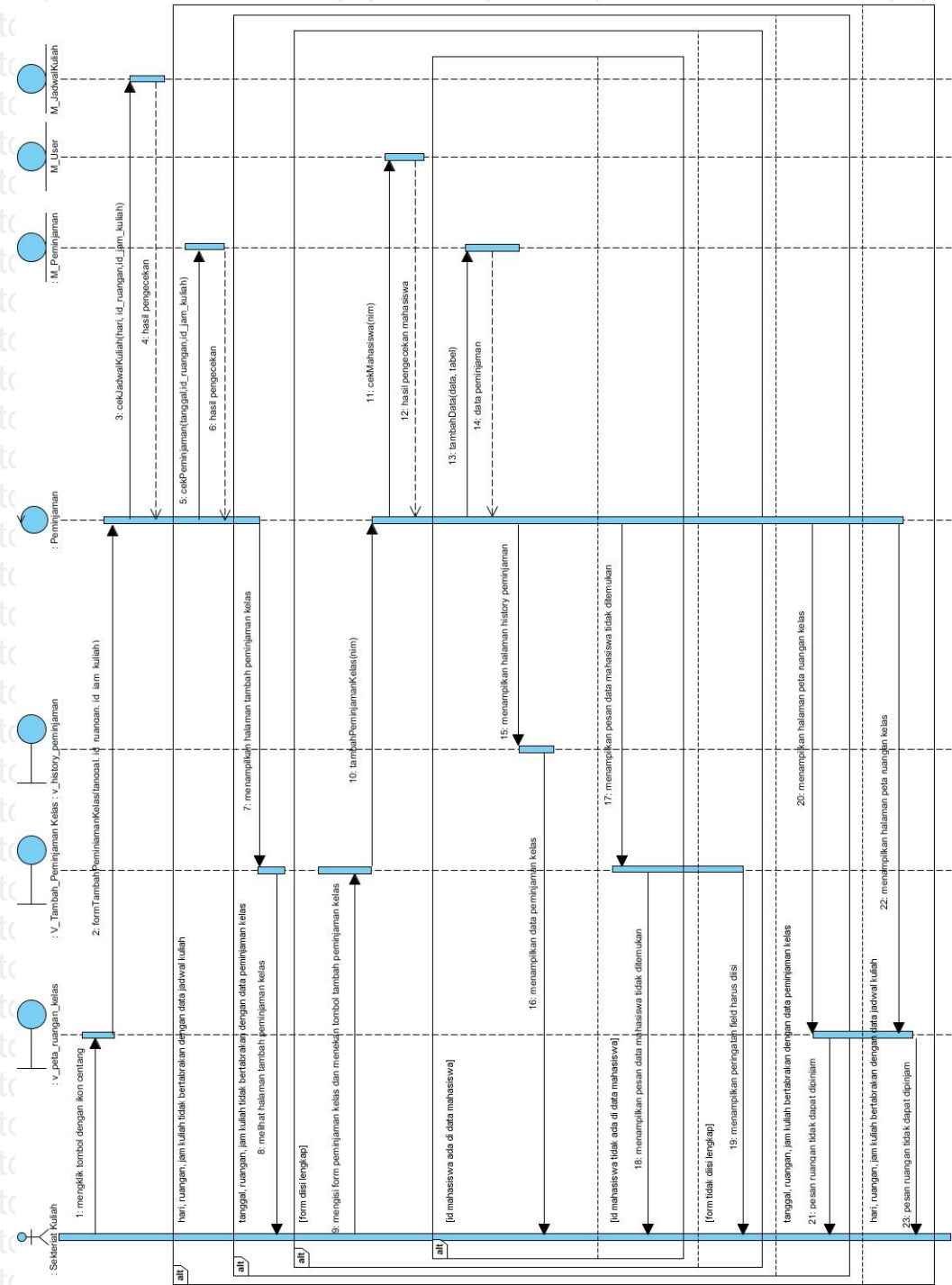
Sequence diagram adalah perancangan perangkat lunak yang menggambarkan urutan dari alur proses yang terjadi disaat aktor berinteraksi dengan sistem. *Sequence diagram* yang akan dijelaskan adalah alur dari proses tambah peminjaman ruang kelas, cek status peminjaman dan tambah peminjaman barang.

5.1.1.1 Sequence Diagram Tambah Peminjaman Kelas

Tambah peminjaman kelas adalah kegiatan yang dilakukan oleh sekretariat kuliah untuk dapat membuat peminjaman kelas baru. *Sequence diagram* dari tambah peminjaman kelas dapat dilihat pada Gambar 5.1. Hal pertama yang dilakukan sekretariat kuliah adalah menekan tombol centang pada halaman peta ruangan kelas. Saat tombol centang ditekan sistem akan memanggil method `formTambahPeminjamanKelas()` dengan memasukkan nilai parameter tanggal, id ruangan dan id jam kuliah. Setelah itu method `formTambahPeminjamanKelas()` akan memanggil method `cekJadwalkuliah()` pada model jadwal kuliah untuk melakukan pengecekan data peminjaman baru dengan data jadwal kuliah. Jika data peminjaman baru tidak bertabrakan dengan data jadwal kuliah, maka akan dilakukan pengecekan kembali dengan memanggil method `cekPeminjaman()` pada model peminjaman. Method `cekPeminjaman()` melakukan pengecekan apakah data peminjaman baru bertabrakan dengan data peminjaman atau tidak. Jika hasil pengecekan menyatakan tidak bertabrakan maka method `formTambahPeminjamanKelas` akan menampilkan halaman tambah peminjaman kelas. Hasil pengecekan data peminjaman baru yang bertabrakan dengan data jadwal kuliah pada model jadwal kuliah akan membuat method `formTambahPeminjamanKelas` kembali ke halaman peta ruangan kelas. Hal ini juga sama jika data peminjaman baru bertabrakan dengan data peminjaman pada model peminjaman.

Setelah halaman tambah jadwal kuliah ditampilkan, sekretariat kuliah dapat mengisi form tambah peminjaman kelas dengan lengkap dan menekan tombol tambah peminjaman kelas. Jika form benar diisi lengkap, sistem akan memanggil method `tambahPeminjamanKelas()`. Method `tambahPeminjamanKelas` akan memanggil method `cekMahasiswa()` dari model user untuk melakukan pengecekan data mahasiswa. Jika data mahasiswa benar ada di model user maka method `tambahPeminjamanKelas` akan memanggil method `tambahData()` dari model peminjaman untuk menyimpan data peminjaman kelas baru ke database. Namun, jika data mahasiswa tidak ada di model user maka sistem akan

menampilkan kembali halaman tambah peminjaman kelas. Hal ini juga sama ketika form tambah peminjaman kelas tidak diisi lengkap. Method tambahPeminjamanKelas tidak akan dipanggil sebelum form diisi lengkap.

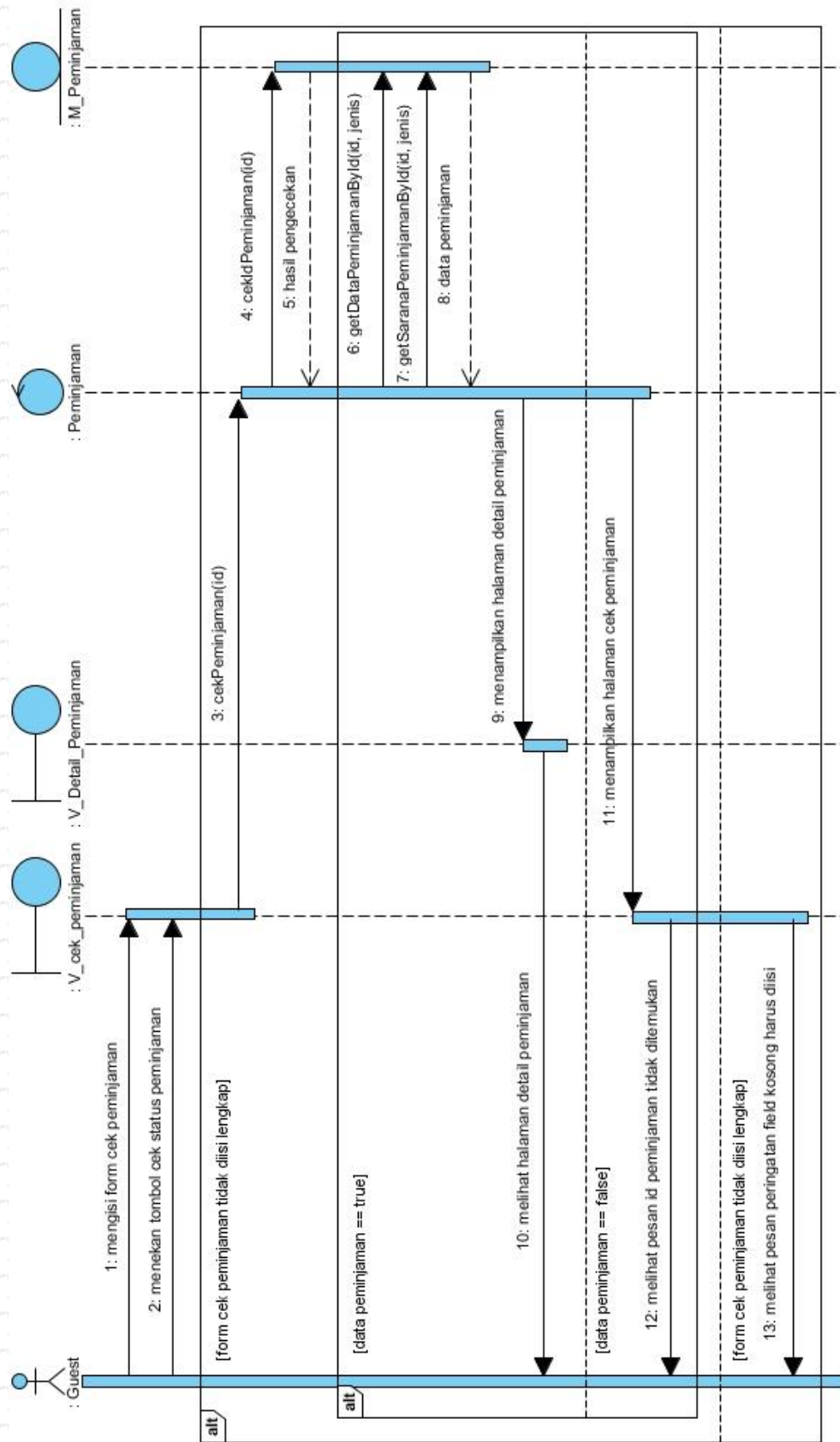


Gambar 5.1 Sequence Diagram Tambah Peminjaman Kelas



5.1.1.2 Sequence Diagram Cek Status Peminjaman

Cek status peminjaman adalah proses yang dilakukan oleh guest untuk dapat melakukan pengecekan status peminjaman. *Sequence diagram* cek status peminjaman dapat dilihat pada Gambar 5.2. Cek status peminjaman dapat dimulai ketika aktor sedang mengakses halaman `v_cek_peminjaman`. Pada halaman `v_cek_peminjaman` guest dapat mengisi form cek peminjaman dan menekan tombol cek status peminjaman. Saat tombol cek status peminjaman ditekan maka sistem akan melakukan pengecekan kelengkapan isian form pada halaman `v_cek_peminjaman`. Jika form cek peminjaman masih kosong maka sistem akan menampilkan pesan peringatan *field* kosong harus diisi. Sedangkan jika form cek peminjaman sudah diisi maka sistem akan memanggil method `cekPeminjaman()`. Method `cekPeminjaman()` akan melakukan pengecekan ketersediaan id peminjaman dengan memanggil method `cekIdPeminjaman()` pada model peminjaman. Jika id peminjaman tidak ditemukan pada model peminjaman maka method `cekPeminjaman()` akan menampilkan kembali halaman cek peminjaman dengan pesan id peminjaman tidak ditemukan. Sedangkan jika id peminjaman ada pada model peminjaman maka method `cekPeminjaman()` akan memanggil method `getDataPeminjaman()` dari model peminjaman untuk mengambil data peminjaman dan method `getSaranaPeminjaman()` dari model peminjaman untuk mengambil data sarana peminjaman. Setelah itu method `cekPeminjaman()` akan menampilkan halaman detail peminjaman.

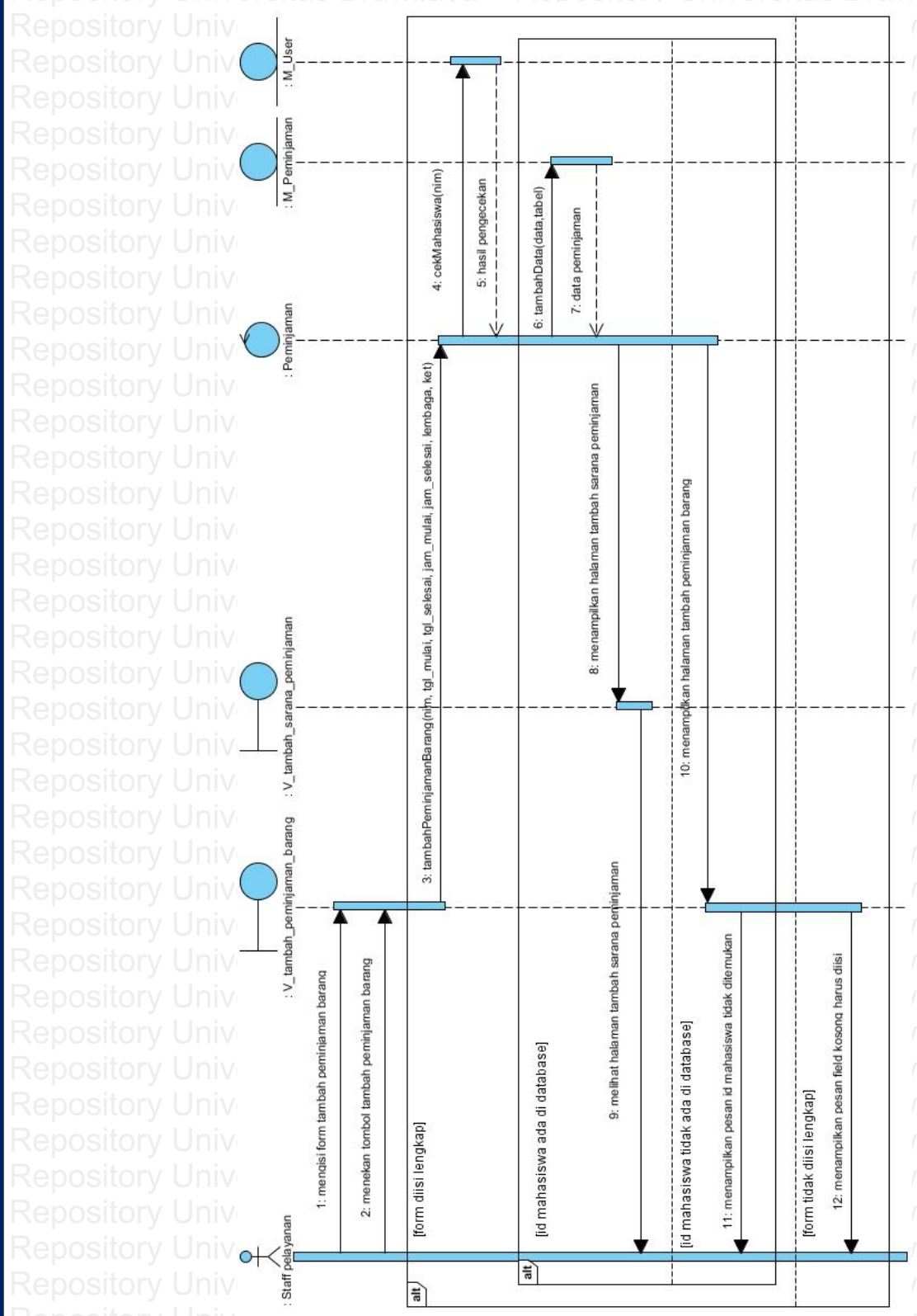


Gambar 5.2 Sequence Diagram Cek Status Peminjaman



5.1.1.3 Sequence Diagram Tambah Peminjaman Barang

Tambah peminjaman barang adalah proses yang dilakukan oleh staf pelayanan untuk dapat melakukan penambahan peminjaman barang baru. *Sequence diagram* tambah peminjaman barang dapat dilihat pada Gambar 5.3. Proses tambah peminjaman barang dapat dimulai ketika aktor sedang mengakses halaman `v_tambah_peminjaman_barang`. Pada halaman `v_tambah_peminjaman_barang` staf pelayanan dapat mengisi form peminjaman barang dan menekan tombol tambah peminjaman barang. Setiap *field* pada form harus diisi dengan lengkap. Jika form tidak diisi dengan lengkap maka sistem akan merespon dengan pesan peringatan *field* harus diisi. Jika form sudah diisi lengkap maka tombol tambah peminjaman barang dapat ditekan. Pada saat tombol tambah peminjaman barang ditekan, maka sistem akan memanggil *method* `tambahPeminjamanBarang()` dari *controller* peminjaman. *Method* `tambahPeminjamanBarang()` akan melakukan pengecekan data mahasiswa yang diinput. Pengecekan dilakukan untuk mengetahui apakah data mahasiswa ada di database atau tidak dengan memanggil *method* `cekMahasiswa()` dari *class* model `M_User`. Jika data mahasiswa tidak ada di *database* maka sistem akan kembali ke halaman `v_tambah_peminjaman_barang`. Sedangkan jika data mahasiswa benar maka *method* `tambahPeminjamanBarang()` akan memanggil *method* `tambahPeminjaman()` dari model `M_peminjaman` untuk menyimpan data peminjaman barang di *database*. Setelah data peminjaman barang disimpan maka sistem akan *redirect* ke halaman `v_tambah_sarana_peminjaman`.

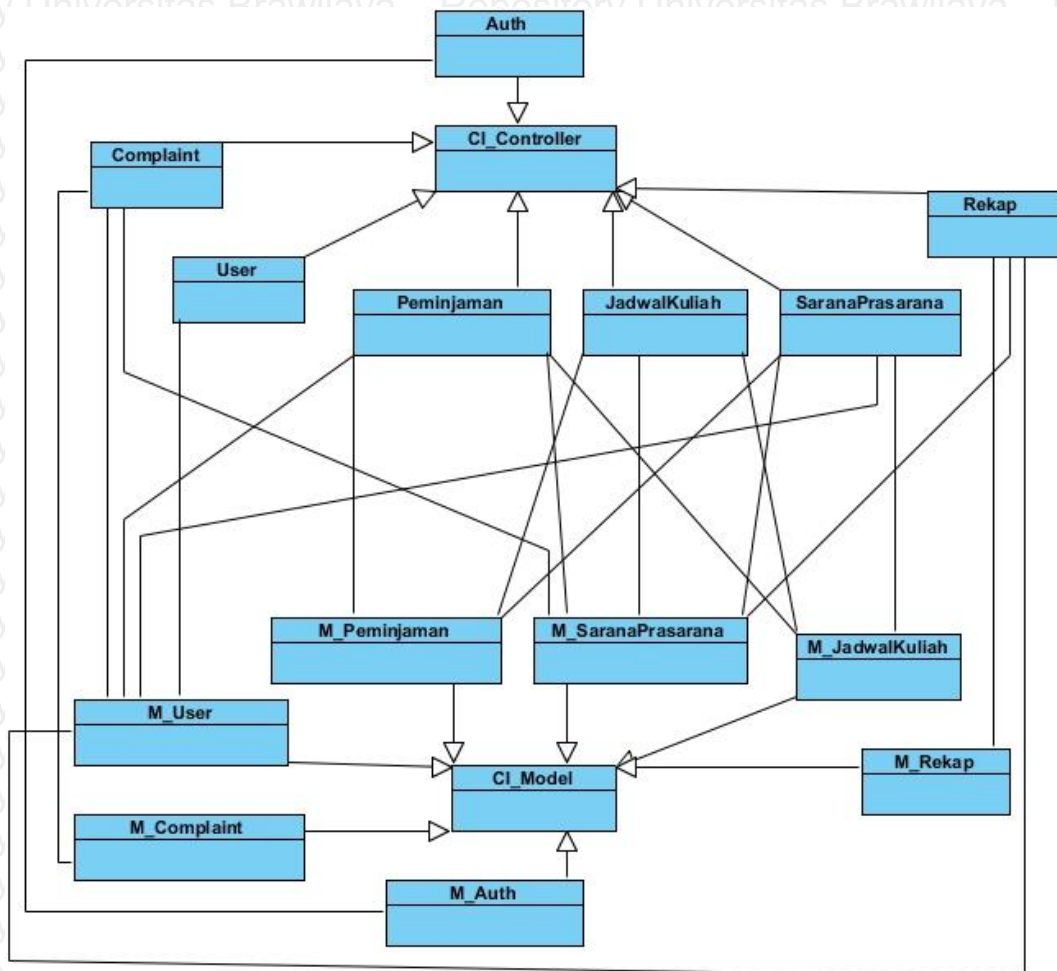


Gambar 5.3 Sequence Diagram Tambah Peminjaman Barang



5.1.2 Class Diagram

Class diagram adalah perancangan perangkat lunak yang menggambarkan *class* apa saja yang digunakan untuk membangun sistem. *Class diagram* sistem manajemen ruang dan sarana prasarana dapat dilihat pada Gambar 5.4



Gambar 5.4 Class Diagram Sistem Manajemen Sarana Prasarana

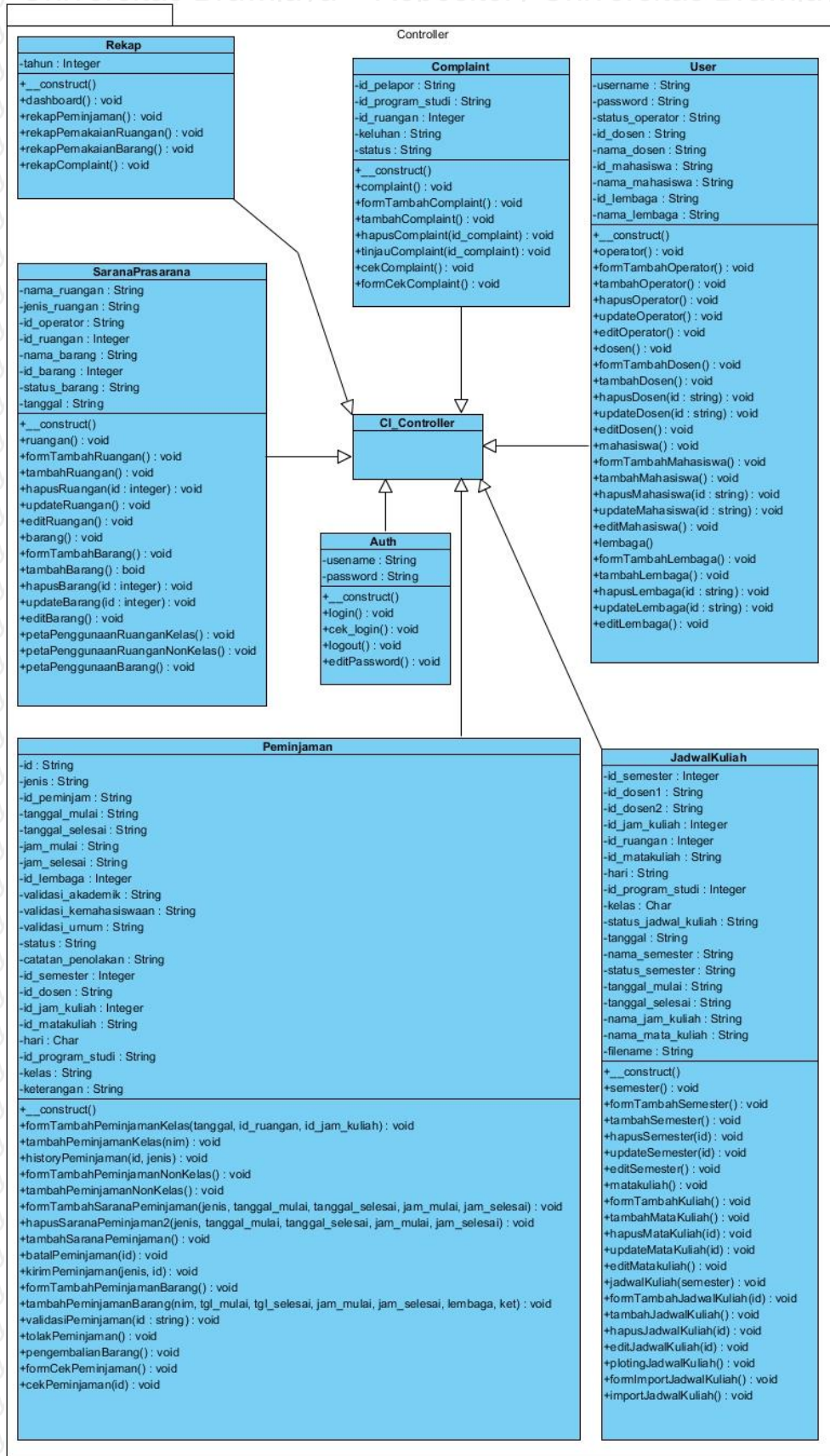
Sistem manajemen peminjaman ruangan dan barang dibuat dengan mengikuti konsep MVC dari *framework* yang digunakan untuk membangun sistem. *Framework* yang digunakan untuk membangun sistem adalah *framework* CodeIgniter. Class diagram pada Gambar 5.4 terdiri dari *class controller* dan *class model*. Class controller berguna untuk menerima aksi dari view atau halaman sistem dan memberikan aksi balik ke halaman sistem. Sedangkan *class model* berguna untuk mengambil data dari database dan menerima data inputan dari *controller* dan menyimpannya ke dalam *database* sistem. Sistem manajemen sarana prasarana dirancang dengan mempunyai *class* sebanyak 14. 7 *class* diantaranya adalah *class controller* dan sisanya adalah *class model*. Class controller terdiri dari *class* auth, user, peminjaman, sarana_prasarana,



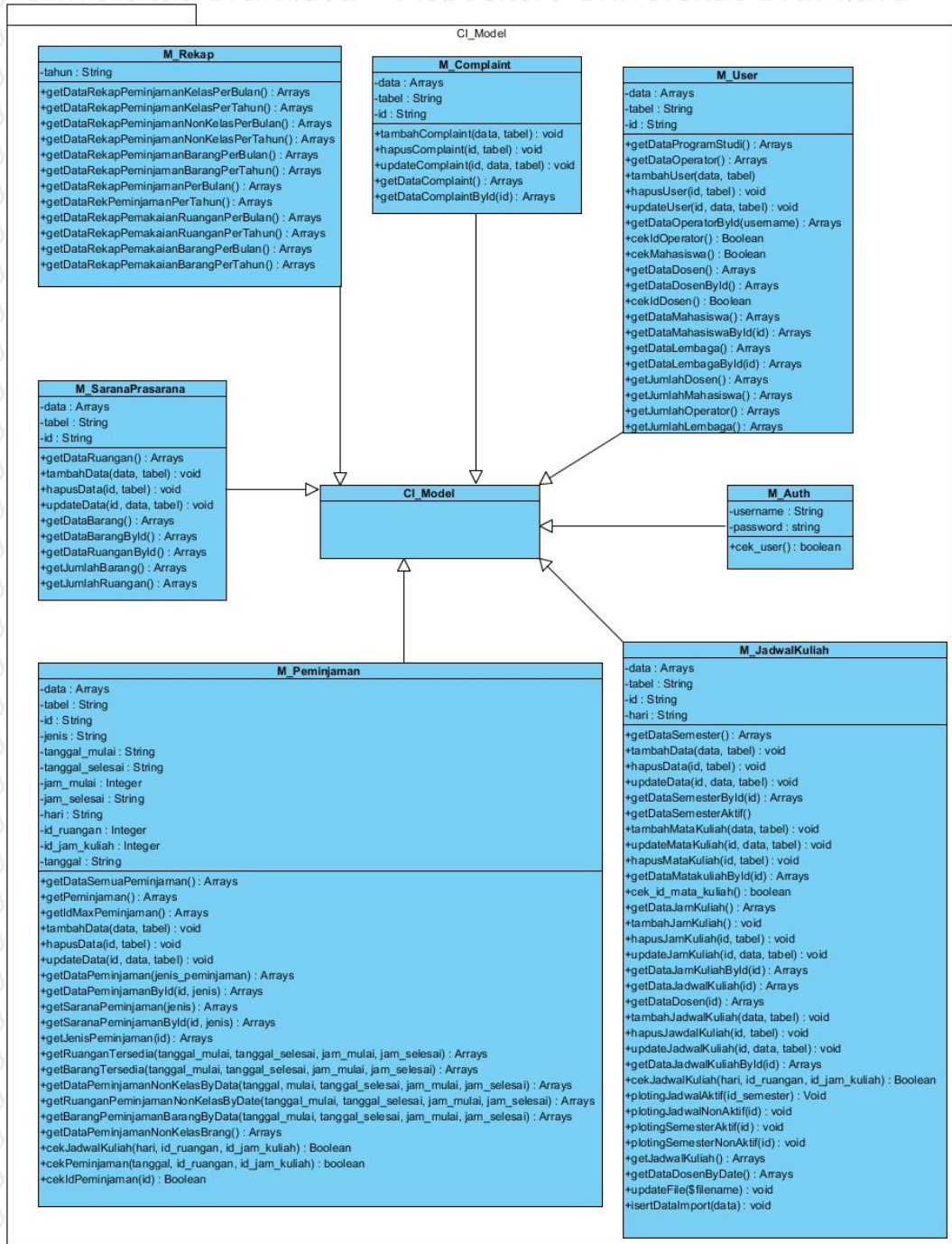
jadwal_kuliah, *complaint* dan rekap. Sedangkan class model terdiri dari class M_auth, M_user, M_peminjaman, M_sarana_prasarana, M_jadwal_kuliah, M_complaint dan M_Rekap. *Class diagram* yang ditunjukkan pada Gambar 5.4 adalah gambaran class diagram secara umum tanpa memberikan keterangan penjelasan terkait atribut dan fungsi yang akan digunakan di sistem. Detail dari masing masing class dapat dilihat pada Gambar 5.5 dan Gambar 5.6

Gambar 5.5 adalah detail dari class diagram yang menjadi controller. *Class auth* digunakan untuk menampung *method* yang digunakan ketika *user* melakukan aksi *login*, ganti *password* dan *logout*. *Class user* digunakan untuk menampung *method* yang digunakan ketika *user* melakukan aksi tambah data pengguna, *edit* data pengguna, dan menghapus data pengguna, dimana pengguna dalam hal ini adalah dosen, mahasiswa, validator, sekretariat kuliah, dan staf pelayanan. *Class peminjaman* digunakan untuk menampung *method* yang digunakan ketika *user* melakukan menambah peminjaman, cek status peminjaman, memvalidasi peminjaman, menolak peminjaman, melihat rekap peminjaman dan melihat detail rekap peminjaman. *Class sarana_prasarana* digunakan untuk menampung *method* yang digunakan ketika *user* melakukan aksi tambah, *edit*, dan hapus data ruangan, data barang, dan data jenis barang. *Class jadwal_kuliah* digunakan untuk menampung *method* yang digunakan ketika *user* melakukan aksi tambah, edit dan hapus data yang berkaitan dengan jadwal kuliah seperti data mata kuliah, jam kuliah, semester, dan jadwal kuliah. *Class complaint* digunakan untuk menampung *method* yang digunakan ketika *user* melakukan aksi tambah *complaint*, cek status *complaint*, menanggapi *complaint*, melihat rekap keluhan dan melihat detail rekap keluhan. *complaint* digunakan untuk menampung *method* yang digunakan ketika *user* ingin melihat data rekap peminjaman, rekap penggunaan ruangan atau barang dan rekap *complaint*.

Gambar 5.6 adalah detail dari *class diagram* yang menjadi model. *Class model* adalah class yang menampung semua query yang digunakan untuk mengambil data dari dan ke *database*. Model peminjaman digunakan untuk menampung *query* yang digunakan untuk mengambil semua data dari tabel peminjaman di *database*. Model jadwal kuliah digunakan untuk menampung *query* yang digunakan untuk mengambil dan memasukkan data ke tabel jadwal kuliah. Model sarana prasarana digunakan untuk mengambil data dan memasukkan data sarana prasarana di tabel sarana prasarana. Model *user* digunakan untuk menampung *query* yang digunakan untuk mengambil data yang berkaitan dengan user pada tabel user. Model *complaint* digunakan untuk menampung *query* yang digunakan untuk mengambil data dan memasukkan data *complaint* pada tabel *complaint*. Model *auth* digunakan untuk menampung *query* yang akan *digunakan* dalam melakukan proses *autentifikasi*. Model rekap digunakan untuk menampung *query* yang digunakan dalam proses menampilkan data laporan dengan mengambil data dari tabel peminjaman, tabel sarana prasarana, tabel user, dan tabel *complaint*.



Gambar 5.5 Detail Class Diagram Bagian Controller



Gambar 5.6 Detail Class Diagram Bagian Model



5.1.3 Perancangan Algoritma

Perancangan algoritma digunakan untuk menjelaskan algoritma yang akan diterapkan pada fungsi fungsi sistem. Adapun fungsi yang akan dijelaskan dalam perancangan algoritma ini adalah algoritma fungsi tambahJadwalKuliah(), tambahPeminjamanBarang(), validasiPeminjaman().

5.1.3.1 Algoritma fungsi tambahJadwalKuliah()

Fungsi tambahPeminjamanKelas() yang dijelaskan adalah fungsi yang berada dalam *class controller* peminjaman. Fungsi ini digunakan untuk menangani aksi peminjaman ruangan kelas perkuliahan yang dilakukan oleh sekretariat kuliah. Perancangan komponen algoritma fungsi tambahJadwalKuliah dapat dilihat pada Tabel 5.1 berikut.

Tabel 5.1 Algoritma fungsi tambahJadwalKuliah()

No	Algoritma
1	Mulai
2	Get data_input_jadwal_kuliah[]
3	If cek_jadwal_kuliah() = true
4	Output = jadwal kuliah gagal ditambahkan
5	Redirect halaman tambah jadwal kuliah
6	Else
7	data = data_input_jadwal_kuliah[]
8	tambahJadwalKuliah(data)
9	Output = jadwal kuliah berhasil ditambahkan jadwal
10	ada yang bertabrakan
11	Redirect halaman jadwal kuliah
12	Selesai

5.1.3.2 Algoritma fungsi tambahPeminjamanBarang()

Fungsi tambahPeminjamanBarang() yang dijelaskan adalah fungsi yang berada dalam *class controller* peminjaman. Fungsi ini digunakan untuk menangani aksi peminjaman barang yang dilakukan oleh staf pelayanan. Perancangan komponen algoritma fungsi tambahPeminjamanBarang() dapat dilihat pada Tabel 5.2 berikut.

Tabel 5.2 Algoritma fungsi tambahPeminjamanBarang()

No	Algoritma
1	Mulai
2	Get data_input_peminjaman[]
3	Set jenis_peminjaman = barang, validasi_akademik = pending,



```

validasi_kemahasiswaan = pending,
validasi_umum = pending
4 Peminjaman = getIdMaxPeminjaman()
5 If id_mahasiswa != false
6   if peminjaman != null
7     id_peminjaman = id_peminjaman + 1
8   else
9     id_peminjaman = 3+tanggal_mulai_penggunaan+0001
10 data = data_input_peminjaman[]
11 tambahPeminjaman(data)
12 output = peminjaman berhasil ditambahkan
13 redirect halaman tambah sarana peminjaman
14 else
15 output = data mahasiswa tidak ditemukan
16 redirect halaman tambah peminjaman barang

```

5.1.3.3 Algoritma fungsi validasiPeminjaman()

Fungsi validasiPeminjaman() yang dijelaskan adalah fungsi yang berada dalam *class controller* peminjaman. Fungsi ini digunakan untuk menangani aksi untuk melakukan validasi peminjaman yang dilakukan oleh seorang validator. Perancangan komponen algoritma fungsi validasiPeminjaman() dapat dilihat pada Tabel 5.3 berikut.

Tabel 5.3 Algoritma fungsi validasiPeminjaman()

No	Algoritma
1	Mulai
2	Get jenis_peminjaman, id_peminjaman
3	Set status = setuju
4	Operator = userdata(status_operator)
5	if jenis_peminjaman = kelas
6	data[] = validasi_akademik = setuju,
7	validasi_kemahasiswaan = setuju,
8	validasi_umum = setuju
9	else if jenis_peminjaman = non kelas
10	if operator = akademik
11	data[] = validasi_akademik = setuju
12	else if operator = kemahasiswaan



```

13 data[] = validasi_kemahasiswaan = setuju
14 else
15 data[] = validas_umum = setuju
16 else
17 if operator = akademik
18 data[] = validasi_akademik = setuju
19 else if operator = kemahasiswaan
20 data[] = validasi_kemahasiswaan = setuju
21 else
22 data[] = validas_umum = setuju
23 Update detailPeminjaman(data)
24 Output = data peminjaman telah berhasil disetujui
25 Redirect halaman history peminjaman

```

5.1.4 Perancangan Antarmuka

Perancangan antar muka adalah perancangan sistem yang bertujuan untuk memberikan gambaran tampilan sistem yang dibangun. Tampilan dirancang terlebih dahulu sebelum diimplementasikan menggunakan bahasa pemrograman. Adapun rancangan antar muka yang dijelaskan adalah antarmuka dari halaman peta jadwal kuliah, halaman peta status penggunaan ruangan, halaman history peminjaman, halaman rekam peminjaman, halaman tambah operator.

5.1.4.1 Rancangan Antarmuka Halaman Jadwal Kuliah

Rancangan antarmuka halaman jadwal kuliah dapat dilihat pada Gambar 5.5 berikut.

Ruang \ Jam	06.45 - 08.25	08.30 - 10.10	10.15 - 11.55	12.00-13.00	13.15-14.55	15.30-17.10	18.00-20.00
Ruang 1	Kuliah Nama Dosen Kelas	Mata Kuliah Nama Dosen Kelas	Mata Kuliah Nama Dosen Kelas	Mata Kuliah Nama Dosen Kelas	Mata Kuliah Nama Dosen Kelas	Mata Kuliah Nama Dosen Kelas	Mata Kuliah Nama Dosen Kelas
Ruang 2	Mata Kuliah Nama Dosen Kelas	Mata Kuliah Nama Dosen Kelas	Mata Kuliah Nama Dosen Kelas	Mata Kuliah Nama Dosen Kelas	Mata Kuliah Nama Dosen Kelas	Mata Kuliah Nama Dosen Kelas	Mata Kuliah Nama Dosen Kelas
Ruang 3	Mata Kuliah Nama Dosen Kelas	Mata Kuliah Nama Dosen Kelas	Mata Kuliah Nama Dosen Kelas	Mata Kuliah Nama Dosen Kelas	Mata Kuliah Nama Dosen Kelas	Mata Kuliah Nama Dosen Kelas	Mata Kuliah Nama Dosen Kelas
Ruang 4	Mata Kuliah Nama Dosen Kelas	Mata Kuliah Nama Dosen Kelas	Mata Kuliah Nama Dosen Kelas	Mata Kuliah Nama Dosen Kelas	Mata Kuliah Nama Dosen Kelas	Mata Kuliah Nama Dosen Kelas	Mata Kuliah Nama Dosen Kelas

Gambar 5.7 Rancangan Antarmuka Halaman Jadwal Kuliah

Penjelasan dari Gambar 5.7 diatas dapat dilihat pada Tabel 5.4.



Tabel 5.4 Penjelasan Rancangan Antarmuka Halaman Jadwal Kuliah

No	Nama Objek	Keterangan
1	Navbar	Menampilkan pilihan menu pada sistem Ruang FH
2	Judul Halaman	Menampilkan judul halaman yang sedang diakses
3	Tombol Filter	Sebuah tombol yang disediakan untuk melakukan filter jadwal kuliah
4	Tabel Jadwal Kuliah	Tabel yang berisi informasi jadwal kuliah

5.1.4.2 Rancangan Antarmuka Halaman Status Penggunaan Ruang Non Kelas

Rancangan antarmuka halaman status penggunaan ruangan non kelas dapat dilihat pada Gambar 5.6 berikut.

Ruang/Jam	06:00-06:59	07:00-07:59	08:00-08:59	09:00-09:59	10:00-10:59	11:00-11:59	12:00-12:59	13:00-13:59	14:00-14:59	15:00-15:59	16:00-16:59	17:00-17:59
Ruang 1	✓	✓	✓	✓	✓	✗	✗	✓	✓	✓	✓	✓
Ruang 2	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓
Ruang 3	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗
Ruang 4	✓	✓	✓	✓	✓	✓	✗	✓	✗	✓	✓	✗

Gambar 5.8 Rancangan Antarmuka Halaman Status Penggunaan Ruang Non Kelas

Penjelasan dari Gambar 5.6 diatas dapat dilihat pada Tabel 5.5 berikut.

Tabel 5.5 Penjelasan Rancangan Antarmuka Halaman Status Penggunaan Rungan Non Kelas

No	Nama Objek	Keterangan
1	Navbar	Menampilkan pilihan menu pada sistem Ruang FH
2	Judul Halaman	Menampilkan judul halaman yang sedang diakses



3	Tombol Filter	Sebuah tombol yang disediakan untuk melakukan filter jadwal kuliah
4	Tabel Jadwal Kuliah	Tabel yang berisi informasi status penggunaan ruangan

5.1.4.3 Rancangan Antarmuka Halaman Tambah Peminjaman Kelas

Rancangan antarmuka halaman tambah peminjaman kelas dapat dilihat pada Gambar 5.9 berikut.

Gambar 5.9 Rancangan Antarmuka Halaman Tambah Peminjaman Kelas

Penjelasan dari Gambar 5.7 diatas dapat dilihat pada Tabel 5.6 berikut.

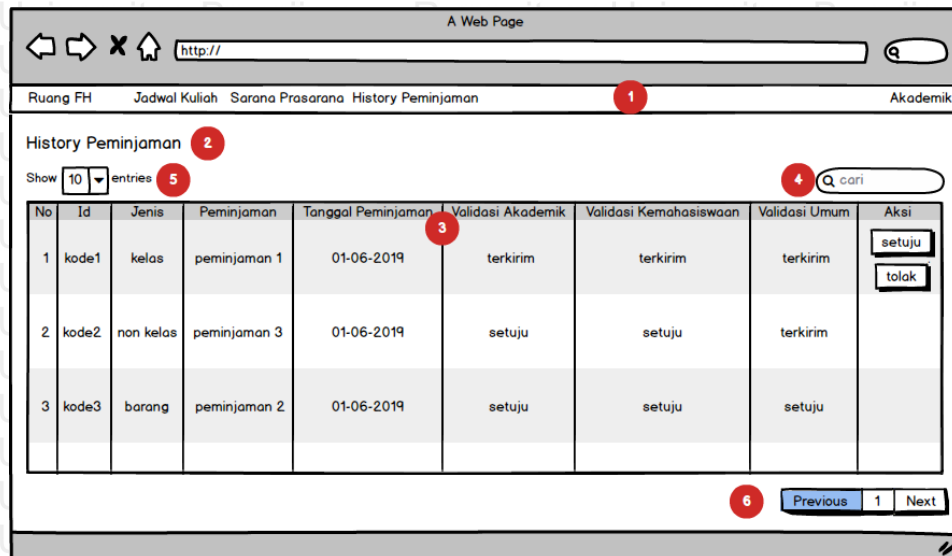
Tabel 5.6 Penjelasan Rancangan Antarmuka Halaman Tambah Peminjaman Kelas

No	Nama Objek	Keterangan
1	Navbar	Menampilkan pilihan menu pada sistem Ruang FH
2	Judul Halaman	Menampilkan judul halaman yang sedang diakses
3	Form tambah peminjaman ruang kelas	Form untuk menerima inputan data yang diperlukan untuk melakukan peminjaman ruang kelas untuk tujuan perkuliahan



5.1.4.4 Rancangan Antarmuka Halaman *History* Peminjaman

Rancangan antarmuka halaman *history* peminjaman dapat dilihat pada Gambar 5.8 berikut.



Gambar 5.10 Rancangan Antarmuka Halaman *History* Peminjaman

Penjelasan dari Gambar 5.8 diatas dapat dilihat pada Tabel 5.7 berikut.

Tabel 5.7 Penjelasan Rancangan Antarmuka Halaman *History* Peminjaman

No	Nama Objek	Keterangan
1	Navbar	Menampilkan pilihan menu pada sistem Ruang FH
2	Judul Halaman	Menampilkan judul halaman yang sedang diakses
3	Tabel <i>History</i> Peminjaman	Tabel berisi informasi terkait peminjaman yang pernah dilakukan.
4	Field search	Sebuah <i>field</i> yang disediakan untuk digunakan dalam melakukan pencarian data pada tabel <i>history</i> peminjaman
5	<i>Option Entries</i>	Sebuah <i>option</i> untuk menampilkan jumlah data yang tampil pada tabel <i>history</i> peminjaman
6	<i>Pagination</i>	Sebuah pengaturan untuk menampilkan data selanjutnya atau data sebelumnya yang belum terlihat pada tabel <i>history</i> peminjaman



5.1.4.5 Rancangan Antarmuka Halaman Rekap Peminjaman

Rancangan antarmuka halaman rekap peminjaman dapat dilihat pada Gambar 5.9 berikut.

Peminjaman	Jan	Feb	Mar	Apr	Mai	Jun	Jul	Agu	Sep	Okt	Nov	Des	Total
Semua	2	5	11	11	16	21	13	16	17	18	20	13	
Kelas	0	4	5	1	5	7	9	3	7	5	4	11	4
Non Kelas	1	2	4	5	2	7	5	4	7	2	5	4	1
Barang	1	0	4	6	2	7	8	2	5	6	1	2	1

Gambar 5.11 Rancangan Antarmuka Halaman Rekap Peminjaman

Penjelasan dari Gambar 5.8 diatas dapat dilihat pada Tabel 5.7 berikut.

Tabel 5.8 Penjelasan Rancangan Antarmuka Halaman Rekap Peminjaman

No	Nama Objek	Keterangan
1	Navbar	Menampilkan pilihan menu pada sistem Ruang FH
2	Judul Halaman	Menampilkan judul halaman yang sedang diakses
3	Tabel <i>History</i> Peminjaman	Tabel berisi informasi terkait peminjaman yang pernah dilakukan.
4	Field search	Sebuah <i>field</i> yang disediakan untuk digunakan dalam melakukan pencarian rekap peminjaman berdasarkan tahun

5.2 Implementasi Sistem

Pada tahap ini perangkat lunak diimplementasikan sesuai dengan apa yang telah dirancang. Pada sub bab implementasi sistem membahas tentang spesifikasi sistem, Batasan implementasi, dan implementasi antarmuka.

5.2.1 Spesifikasi Sistem

Spesifikasi sistem menerangkan tentang penjelasan tentang peralatan yang digunakan dalam proses implementasi. Penjelasan spesifikasi sistem dibagi kedalam dua bagian yakni spesifikasi perangkat keras dan spesifikasi perangkat lunak.



5.2.1.1 Spesifikasi Perangkat Keras

Spesifikasi perangkat keras menerangkan tentang peralatan perangkat keras yang digunakan dalam proses implementasi sistem. Spesifikasi perangkat keras dapat dilihat pada Tabel 5.9 berikut.

Tabel 5.9 Spesifikasi Perangkat Keras

No	Nama Komponen	Spesifikasi
1	Model	HP Notebook
2	Processor	Intel(R) Core™ i7-5500U CPU @2.40 GHz
3	Memory	4GB
4	Display	Intel(R) HD Graphics 5500

5.2.1.2 Spesifikasi Perangkat Lunak

Spesifikasi perangkat lunak menerangkan tentang peralatan perangkat lunak yang digunakan dalam proses implementasi sistem. Spesifikasi perangkat lunak dapat dilihat pada Tabel 5.10 berikut.

Tabel 5.10 Spesifikasi Perangkat Lunak

No	Nama Komponen	Spesifikasi
1	Sistem Informasi	Windows 10 Pro 64 Bit
2	Editor Perancangan	Visual Paradigm for UML 8.0, Balsamiq Mockups 3
3	Editor Pemrograman	Visual Studio Code
4	Framework dan Bahasa Pemrograman	Codeigniter, Bootstrap, Datatable, Chartist, fontawesome, PHP, HTML, CSS, JS
5	DBMS	Mysql
6	Database Server	PhpMyAdmin
7	Aplikasi Pendukung	XAMPP
8	Browser	Mozilla Firefox, Google Chrome, Microsoft Edge

5.2.2 Implementasi Kelas

Implementasi kelas menerangkan tentang penjelasan kelas kelas yang ada pada sistem. Penjelasan tentang kelas dan file pada sistem yang telah diimplementasi dapat dilihat pada Tabel 5.12 berikut



Tabel 5.11 Implementasi Kelas

No	Package	Nama Kelas	Nama File Sistem
1	<i>controller</i>	Auth	Auth.php
2	<i>controller</i>	Complaint	Complaint.php
3	<i>controller</i>	JadwalKuliah	Jadwalkuliah.php
4	<i>controller</i>	Peminjaman	Peminjaman.php
5	<i>controller</i>	Rekap	Rekap.php
6	<i>controller</i>	SaranaPrasarana	SaranaPrasarana.php
7	<i>controller</i>	User	User.php
8	<i>model</i>	M_Auth	M_Auth.php
9	<i>model</i>	M_Complaint	M_Complaint.php
10	<i>model</i>	M_JadwalKuliah	M_Jadwalkuliah.php
11	<i>model</i>	M_Peminjaman	M_Peminjaman.php
12	<i>model</i>	M_Rekap	M_Rekap.php
13	<i>model</i>	M_SaranaPrasarana	M_SaranaPrasarana.php
14	<i>model</i>	M_User	M_User.php
15	<i>view</i>	File HTML	V_login.php
16	<i>view</i>	File HTML	V_GantiPassword.php
17	<i>view</i>	File HTML	V_TambahComplaint.php
18	<i>view</i>	File HTML	V_ListComplaint.php
19	<i>view</i>	File HTML	V_EditJadwalKuliah.php
20	<i>view</i>	File HTML	V_EditJamKuliah.php
21	<i>view</i>	File HTML	V_EditMataKuliah.php
22	<i>view</i>	File HTML	V_EditSemester.php
23	<i>view</i>	File HTML	V_ListJadwalKuliah.php
24	<i>view</i>	File HTML	V_ListJamKuliah.php
25	<i>view</i>	File HTML	V_ListMataKuliah.php
26	<i>view</i>	File HTML	V_ListSemester.php
27	<i>view</i>	File HTML	V_PetaJadwalKuliah.php
28	<i>view</i>	File HTML	V_TambahJadwalKuliah.php
29	<i>view</i>	File HTML	V_TambahJamKuliah.php



30	<i>view</i>	<i>File HTML</i>	V_TambahMataKuliah.php
31	<i>view</i>	<i>File HTML</i>	V_TambahSemester.php
32	<i>view</i>	<i>File HTML</i>	V_FormCekPeminjaman.php
33	<i>view</i>	<i>File HTML</i>	V_DetailPeminjaman.php
34	<i>view</i>	<i>File HTML</i>	V_HistoryPeminjaman.php
35	<i>view</i>	<i>File HTML</i>	V_TambahPeminjamanBarang.php
36	<i>view</i>	<i>File HTML</i>	V_TambahPeminjamanKelas.php
37	<i>view</i>	<i>File HTML</i>	V_TambahPeminjamanNonKelas.php
38	<i>view</i>	<i>File HTML</i>	V_TambahSaranaPeminjaman.php
39	<i>view</i>	<i>File HTML</i>	V_RekapPemakaianBarang.php
40	<i>view</i>	<i>File HTML</i>	V_RekapPemakaianRuangan.php
41	<i>view</i>	<i>File HTML</i>	V_RekapPeminjaman.php
42	<i>view</i>	<i>File HTML</i>	V_Dashboard.php
43	<i>view</i>	<i>File HTML</i>	V_EditBarang.php
44	<i>view</i>	<i>File HTML</i>	V_EditRuangan.php
45	<i>view</i>	<i>File HTML</i>	V_ListBarang.php
46	<i>view</i>	<i>File HTML</i>	V_ListRuangan.php
47	<i>view</i>	<i>File HTML</i>	V_PetaPenggunaanBarang.php
48	<i>view</i>	<i>File HTML</i>	V_PetaPenggunaanRuanganKelas.php
49	<i>view</i>	<i>File HTML</i>	V_PetaPenggunaanRuanganNonKelas.php
50	<i>view</i>	<i>File HTML</i>	V_TambahBarang.php
51	<i>view</i>	<i>File HTML</i>	V_TambahRuangan.php
52	<i>view</i>	<i>File HTML</i>	V_TemplateAdmin.php
53	<i>view</i>	<i>File HTML</i>	V_TemplateOperator.php
54	<i>view</i>	<i>File HTML</i>	V_EditDosen.php
55	<i>view</i>	<i>File HTML</i>	V_EditMahasiswa.php
56	<i>view</i>	<i>File HTML</i>	V_EditOperator.php
57	<i>view</i>	<i>File HTML</i>	V_ListDosen.php
58	<i>view</i>	<i>File HTML</i>	V_ListMahasiswa.php
59	<i>view</i>	<i>File HTML</i>	V_ListOperator.php
60	<i>view</i>	<i>File HTML</i>	V_TambahDosen.php



61	view	File HTML	V_TambahMahasiswa.php
62	view	File HTML	V_TambahOperator.php
63	view	File HTML	V_ImportJadwalKuliah.php

5.2.3 Implementasi Algoritma

Implementasi algoritma menerangkan tentang susunan kode program yang ditulis dalam proses implementasi. Pada sub bab implementasi algoritma akan dijelaskan tiga fungsi yang ada di sistem. Fungsi yang dijelaskan adalah fungsi tambahJadwalKuliah(), tambahPeminjamanBarang(), validasiPeminjaman().

5.2.3.1 Implementasi *Method* tambahJadwalKuliah()

Method tambahJadwalKuliah() adalah *method* yang digunakan dalam melakukan kegiatan tambah jadwal kuliah oleh admin. Implementasi *method* tambah jadwal kuliah dapat dilihat pada Tabel 5.12 berikut

Tabel 5.12 Implementasi *Method* tambahJadwalKuliah()

No	Kode Program
1	public function tambahJadwalKuliah(){
2	\$id_semester = \$this->input->post('id_semester');
3	\$id_dosen = \$this->input->post('id_dosen');
4	\$id_jam_kuliah = \$this->input->post('id_jam_kuliah');
5	\$id_ruangan = \$this->input->post('id_ruangan');
6	\$id_matakuliah = \$this->input->post('id_matakuliah');
7	\$hari = \$this->input->post('hari');
8	\$id_program_studi = \$this->input->
9	post('id_program_studi');
10	\$kelas = \$this->input->post('kelas');
11	\$status_jadwal_kuliah = \$this->input->
12	post('status_jadwal_kuliah');
13	if(\$this->M_JadwalKuliah->cekJadwalKuliah() == TRUE){
14	\$this->session->set_flashdata('notif', "jadwal
15	kuliah gagal ditambahkan, jadwal
16	ada yang bertabrakan");
17	redirect('JadwalKuliah/jadwalKuliah/'
18	.\$id_semester);
19	}else{
20	\$data = array(
21	'id_semester' => \$id_semester,
22	'id_dosen' => \$id_dosen,



```

23     'id_jam_kuliah' => $id_jam_kuliah,
24     'id_matakuliah' => $id_matakuliah,
25     'id_ruangan' => $id_ruangan,
26     'hari' => $hari,
27     'id_program_studi' => $id_program_studi,
28     'kelas' => $kelas,
29     'status_jadwal_kuliah' => $status_jadwal_kuliah
30 );
31 $this->M_JadwalKuliah->tambahJadwalKuliah
32     ($data, 'jadwal_kuliah');
33 $this->session->set_flashdata('notifsukses',
34     "jadwal kuliah berhasil ditambahkan");
35
36     redirect('JadwalKuliah/jadwalKuliah/'.$id_semester);
37 }
38 }

```

Penjelasan

- Baris 1 : Deklarasi nama fungsi tambahJadwalKuliah()
- Baris 2-12 : Berfungsi untuk mengambil data inputan dari form tambah Jadwal kuliah
- Baris 13 : Berfungsi untuk melakukan pengecekan jadwal bertabrakan antara data jadwal kuliah baru dengan data jadwal kuliah yang ada di *database*
- Baris 14-17 : Berfungsi untuk memberitahukan user bahwa proses menambah jadwal kuliah baru tidak berhasil
- Baris 19 : Berfungsi untuk pengecekan jadwal kuliah jika jadwal yang baru tidak bertabrakan dengan jadwal yang ada di *database*
- Baris 20-30 : Berfungsi untuk memberikan menyimpan data inputan ke dalam satu variabel array
- Baris 31 : Berfungsi memanggil method tambahJadwalKuliah dari model M_JadwalKuliah untuk menyimpan data jadwal kuliah baru ke Database
- Baris 32-33 : Berfungsi untuk memberitahukan bahwa jadwal telah berhasil Ditambahkan
- Baris 34 : Akhir method tambahJadwalKuliah()



5.2.3.2 Implementasi *Method* tambahPeminjamanBarang()

Method tambahPeminjamanBarang() adalah *method* yang digunakan dalam melakukan kegiatan tambah peminjaman barang oleh staf pelayanan. Implementasi *method* tambah peminjaman barang dapat dilihat pada Tabel 5.13 berikut.

Tabel 5.13 Implementasi *Method* tambahPeminjamanBarang()

No	Kode Program
1	public function tambahPeminjamanBarang(){
2	\$tanggal_mulai_penggunaan = \$this->input->post(
3	'tanggal_mulai_penggunaan');
4	\$tanggal_selesai_penggunaan = \$this->input->
5	post('tanggal_selesai_penggunaan');
6	\$jam_mulai = \$this->input->post('jam_mulai');
7	\$jam_selesai = \$this->input->post('jam_selesai');
8	\$id_peminjam = \$this->input->post('id_peminjam');
9	\$id_lembaga = \$this->input->post('id_lembaga');
10	\$jenis_peminjaman = "barang";
11	\$validasi_akademik = 'pending';
12	\$validasi_kemahasiswaan = 'pending';
13	\$validasi_umum = 'pending';
14	\$keterangan = \$this->input->post('keterangan');
15	if(\$this->M_user->cek_mahasiswa() == TRUE){
16	\$peminjaman = \$this->M_Peminjaman->
17	getIdMaxPeminjaman();
18	\$id = null;
19	foreach (\$peminjaman as \$u){ echo \$id=\$u->
20	id_peminjaman;}
21	if(\$id != null){
22	\$id_peminjaman = \$id+1;
23	}else{
24	\$kode_tgl = str_replace("-", "",
25	\$tanggal_mulai_penggunaan);
26	\$id_peminjaman = "3".\$kode_tgl."0001";
27	}
28	\$data = array(
29	'id_peminjaman' => \$id_peminjaman,
30	'jenis_peminjaman' => \$jenis_peminjaman,



```

26         'id_peminjam' => $id_peminjam,
27         'tanggal_mulai_penggunaan' =>
           $tanggal_mulai_penggunaan,
28         'tanggal_selesai_penggunaan' =>
           $tanggal_selesai_penggunaan,
29         'jam_mulai' => $jam_mulai,
30         'id_lembaga' => $id_lembaga,
31         'jam_selesai' => $jam_selesai,
32         'validasi_akademik' => $validasi_akademik,
33         'validasi_kemahasiswaan' =>
           $validasi_kemahasiswaan,
34         'validasi_umum' => $validasi_umum,
35         'keterangan' => $keterangan
36     );
37     $this->M_Peminjaman-
           >TambahPeminjaman($data, 'peminjaman');
38     $this->session->set_flashdata('notifsukses',
           "peminjaman berhasil ditambahkan");
39     redirect('Peminjaman/
           formTambahSaranaPeminjaman/barang/'
           .$tanggal_mulai_penggunaan.'/'.
           .$tanggal_selesai_penggunaan.'/'.
           .$jam_mulai.'/'. $jam_selesai);
40     }else{
41         $this->session->set_flashdata('notif',
           "data mahasiswa tidak ditemukan");
42         redirect('Peminjaman/
           formTambahPeminjamanBarang/');
43     }
44 }

```

Penjelasan

- Baris 1 : Deklarasi nama fungsi tambahPeminjamanBarang()
- Baris 2-12 : Berfungsi untuk mengambil data inputan dari form tambah peminjaman barang
- Baris 13 : Berfungsi untuk melakukan pengecekan data mahasiswa apakah Terdaftar di database atau tidak



- Baris 14 : Berfungsi untuk mendapatkan id peminjaman terbesar di Database
- Baris 15-18 : Berfungsi melakukan pengecekan id peminjaman terbesar yang ada di database. Jika database tidak kosong maka id peminjaman yang baru dapat dibuat dengan id peminjaman terbesar + 1
- Baris 19-22 : Berfungsi membuat id peminjaman baru dengan menggunakan Kombinasi dari angka 3, kode tanggal, dan 0001 peminjaman lama ditambah 1
- Baris 23-36 : Berfungsi untuk menyimpan data inputan peminjaman barang ke dalam variabel data yang berbentuk array
- Baris 37 : Berfungsi untuk memanggil method tambahPeminjaman dari model M_Peminjaman untuk menyimpan data peminjaman barang ke dalam database
- Baris 38-39 : Berfungsi untuk memberi pemberitahuan kepada user bahwa peminjaman barang telah berhasil dilakukan
- Baris 40-43 : Berfungsi untuk memberi pemberitahuan kepada user jika data mahasiswa yang diinput tidak terdaftar di database
- Baris 52 : Akhir dari *method* tambahPeminjamanBarang()

5.2.3.3 Implementasi *Method* validasiPeminjaman()

Method validasiPeminjaman() adalah *method* yang digunakan dalam melakukan validasi peminjaman kelas, peminjaman non kelas dan peminjaman barang. Implementasi *method* validasi peminjaman dapat dilihat pada Tabel 5.16 berikut

Tabel 5.14 Implementasi *Method* validasiPeminjaman()

No	Kode Program
1	public function validasiPeminjaman(\$id_peminjaman){
2	\$status = "setuju";
3	\$operator = \$this->session->
4	userdata('status_operator');
5	if(\$operator == 'kasubag akademik'){
6	\$data = array(
7	'validasi_akademik' => \$status
8);
9	}else if(\$operator == 'kasubag kemahasiswaan'){
10	\$data = array(



```

11     'validasi_kemahasiswaan' => $status
12     );
13     }else{
14         $data = array(
15             'validasi_umum' => $status
16         );
17     }
18     $where = array('id_peminjaman' => $id_peminjaman);
19     $this->M_Peminjaman-
20         >updatePeminjaman($where,$data,'peminjaman');
21     $this->session->set_flashdata('notifsukses', "Data
22         peminjaman telah berhasil disetujui");
23     redirect('Peminjaman/historyPeminjaman');
24 }

```

Penjelasan

- Baris 1 : Deklarasi nama fungsi validasiPeminjaman()
- Baris 2 : Berfungsi untuk inisialisasi variabel status = setuju
- Baris 3-4 : Berfungsi menyimpan status user saat login ke dalam variabel operator
- Baris 5 - 8 : Berfungsi untuk melakukan pengecekan status operator yang melakukan validasi. Jika status adalah akademik maka hanya status validasi akademik yang akan menjadi setuju
- Baris 9 - 12 : Berfungsi untuk melakukan pengecekan status operator yang melakukan validasi. Jika status adalah kemahasiswaan maka hanya status validasi kemahasiswaan yang akan menjadi setuju
- Baris 13 - 17 : Berfungsi untuk melakukan pengecekan status operator yang melakukan validasi. Jika status adalah umum maka hanya status validasi umum yang akan menjadi setuju
- Baris 18 - 20 : Berfungsi untuk memanggil method updatePeminjaman untuk mengubah nilai dari validasi akademik, validasi kemahasiswaan, validasi umum di database
- Baris 21 - 23 : Berfungsi untuk memberitahukan user bahwa data peminjaman telah disetujui
- Baris 24 : akhir dari method validasiPeminjaman()



5.2.4 Implementasi Antarmuka

Implementasi antarmuka memuat penjelasan tentang hasil implementasi rancangan antarmuka sistem. Antarmuka atau tampilan sistem diimplementasi dengan menggunakan *framework* bootstrap. Pada sub bab ini akan dijelaskan antar muka dari halaman jadwal kuliah, halaman penggunaan ruangan, halaman tambah peminjaman kelas, halaman history peminjaman dan halaman rekap peminjaman.

5.2.4.1 Implementasi Antarmuka Halaman Jadwal Kuliah

Antarmuka halaman jadwal kuliah berisi informasi pemetaan jadwal kuliah setiap harinya. Informasi jadwal kuliah dapat di filter berdasarkan tanggal, program studi, dosen, ruang, mata kuliah, dan status jadwal. Halaman antarmuka jadwal kuliah dapat dilihat pada Gambar 5.10 berikut

R/J	06:45 - 08:25	08:30 - 10:10	10:15 - 11:55	12:00 - 13:00	13:15 - 14:55	15:30 - 17:10	18:00 - 20:00
B.1/1	Hukum Acara Perdata Dr. BUDI SANTOSO, SH, LL.M. Kelas-B	Hukum Perdata MOCH. ZAIRUL ALAM, SH, MH. Kelas-D	Tindak Pidana dalam KUHP FINES FATIMAH S.H., M.H. Dr. ABDUL MADJID, SH, M.Hum. Kelas-J		Hukum Jaminan Prof.Dr. SUHARININGSIH, SH, SU. RUMI SUWARDNYATI, SH, M.Kn. Kelas-B	Kapita Selekta Hukum Internasional ANAK AGUNG AYU NANDA SARASWATI, S.H., M.H. FRANSISKA AVULISTIYA SUSANTO, SH, LL.M. Kelas-A	
B.1/2		Hukum Perdata RUMI SUWARDNYATI, SH, M.Kn. Kelas-C	Perselisihan Hubungan Industrial Prof.Dr. SUHARININGSIH, SH, SU. Kelas-A		Hukum Kesehatan Dr. AAN EKO WIDIARTO, SH, M.Hum. Dr. YULIATI, SH, LL.M. Kelas-I	Kejahatan Korporasi Prof.Dr. I NYOMAN NURJAYA, SH, MS. ENY HARJATI, SH, M.Hum. Kelas-G	Sistem Peradilan Pidana Dr. PRJIA DJATMIKA, SH, M.S. ALFONS ZAKARIA, SH, LL.M. Kelas-A
B.1/3	Hukum Acara Pidana	Hukum Perikatan PRAWATYA IDO NURHAWATI,	Hukum Lingkungan DEWI CAHYANDARI, S.H., M.H.		Hukum Laut Internasional	Kriminologi Dr. PRJIA DJATMIKA, SH,	

Gambar 5.12 Implementasi Antarmuka Halaman Jadwal Kuliah

5.2.4.2 Implementasi Antarmuka Halaman Penggunaan Ruangan Non Kelas

Antarmuka halaman penggunaan ruangan non kelas menampilkan informasi status penggunaan ruangan non kelas. Status penggunaan ruangan kelas ada dua, digunakan dan tidak digunakan. Pada halaman penggunaan ruangan non kelas ruangan yang digunakan dilambangkan dengan tanda silang berwarna merah dan ruangan yang kosong dilambangkan dengan tanda centang berwarna biru. Informasi status penggunaan ruangan non kelas dapat di filter berdasarkan tanggal. Filter ini berguna untuk menampilkan informasi status penggunaan ruangan di hari selanjutnya. Antarmuka halaman penggunaan ruangan non kelas dapat dilihat pada Gambar 5.11 berikut.



Ruang FH Jadwal Kuliah Sarana Prasarana History Peminjaman sekretariatkuliah

Penggunaan Ruang Non Kelas

Monday, 27-05-2019

R/J	06.00 - 06.59	07.00 - 07.59	08.00 - 08.59	09.00 - 09.59	10.00 - 10.59	11.00 - 11.59	12.00 - 12.59	13.00 - 13.59	14.00 - 14.59	15.00 - 15.59	16.00 - 16.59	17.00 - 17.59	18.00 - 18.59	19.00 - 19.59	20.00 - 20.59	21.00 - 21.59
Mimbar	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Aula	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Lobby	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Ruang Rapat	✓	✓	✓	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Auditorium	✓	✓	✓	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Lapangan Depan	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Gambar 5.13 Implementasi Antarmuka Halaman Penggunaan Ruang Non Kelas

5.2.4.3 Implementasi Antarmuka Tambah Peminjaman Kelas

Antarmuka halaman tambah peminjaman kelas digunakan untuk melakukan proses penambahan peminjaman ruangan kelas. Halaman tambah peminjaman kelas berisi form yang harus diisi untuk membuat peminjaman kelas baru. Form peminjaman kelas yang harus diisi terdapat nim peminjam, semester, mata kuliah, dosen, program studi, kelas dan keterangan. Sedangkan tanggal penggunaan, jam kuliah dan ruangan akan terisi otomatis disaat aktor mengklik ikon centang biru pada halaman penggunaan ruangan kelas. Antarmuka halaman tambah peminjaman kelas dapat dilihat pada Gambar 5.12 berikut.

Ruang FH Jadwal Kuliah Sarana Prasarana History Peminjaman sekretariatkuliah

Tambah Peminjaman Kelas

NIM Peminjam
masukkan nim/nik sebagai identitas peminjam

Mata Kuliah
HKA4001 - Hukum Administrasi Daerah

Dosen
195005261980022001 - Prof.Dr. SUHARININGSIH, SH., SU.

Tanggal Penggunaan
Friday, 05-07-2019

Jam Kuliah
10:15 - 11:55

Ruangan
B.1/2

Program Studi
Sarjana Ilmu Hukum

Kelas

Keterangan

Tambah Peminjaman Kelas

Gambar 5.14 Implementasi Antarmuka Halaman Tambah Peminjaman Kelas



5.2.4.4 Implementasi *History* Peminjaman

Antarmuka halaman *history* peminjaman digunakan untuk menampilkan informasi riwayat peminjaman yang pernah dilakukan. Halaman *history* peminjaman terdiri dari tabel *history* peminjaman. Tabel *history* peminjaman tidak menampilkan semua data peminjaman secara lengkap. Data peminjaman yang dapat dilihat pada halaman ini adalah id peminjaman, jenis peminjaman, nama peminjam, tanggal peminjaman, status validasi akademik, validasi kemahasiswaan, dan validasi umum. Untuk melihat data peminjaman secara lengkap maka id peminjaman dapat diklik agar detail peminjaman ditampilkan. Pada tabel *history* peminjaman terdapat kolom validasi yang berisi tombol setuju dan tolak. Tombol setuju digunakan untuk menyetujui peminjaman dan tombol tolak digunakan untuk menolak peminjaman. Tombol ini hanya akan tampil jika status operator yang mengakses adalah sekretariat kuliah kasubag akademik, kasubag kemahasiswaan dan kasubag umum. Antarmuka halaman *history* peminjaman dapat dilihat pada Gambar 5.13 berikut.

No	Id	Jenis	Peminjam	Tanggal Peminjaman	Validasi Akademik	Validasi Kemahasiswaan	Validasi Umum	Validasi
1	1201905270005	barang	Sueddi Sihotang	Sunday, 26-05-2019	tolak	terkirim	terkirim	Setuju Tolak
2	1201905270011	non kelas	Sueddi Sihotang	Thursday, 23-05-2019	setuju	setuju	setuju	Setuju Tolak
3	1201905270012	non kelas	Sueddi Sihotang	Tuesday, 23-04-2019	setuju	setuju	setuju	Setuju Tolak
4	1201905270014	non kelas	Sueddi Sihotang	Tuesday, 23-04-2019	terkirim	terkirim	terkirim	Setuju Tolak
5	1201905270015	non kelas	Sueddi Sihotang	Tuesday, 23-04-2019	terkirim	terkirim	terkirim	Setuju Tolak
6	1201905270016	barang	Sueddi Sihotang	Tuesday, 23-04-2019	setuju	setuju	setuju	Setuju Tolak
7	1201905270017	barang	Sueddi Sihotang	Monday, 15-04-2019	setuju	setuju	setuju	Setuju Tolak
8	1201905270018	barang	Sueddi Sihotang	Saturday, 13-04-2019	tolak	terkirim	terkirim	Setuju Tolak
9	1201905270019	barang	Sueddi Sihotang	Sunday, 21-04-2019	setuju	setuju	setuju	Setuju Tolak
10	1201905270020	barang	Sueddi Sihotang	Wednesday, 23-01-2019	setuju	setuju	setuju	Setuju Tolak

Gambar 5.15 Implementasi Antarmuka Halaman *History* Peminjaman

5.2.4.5 Implementasi Rekap Peminjaman

Antarmuka halaman rekap peminjaman digunakan untuk menampilkan informasi rekap peminjaman. Halaman rekap peminjaman hanya dapat diakses oleh aktor admin. Halaman rekap peminjaman terdiri dari tabel untuk menampilkan data rekap. Data rekap peminjaman yang ditampilkan adalah jumlah peminjaman kelas, non kelas, barang, dan jumlah keseluruhan. Data rekap peminjaman ditampilkan dalam perbulan selama setahun. Pada halaman rekap peminjaman juga terdapat tombol filter tahun untuk menampilkan data rekap



peminjaman pada tahun yang diinginkan. Antarmuka halaman rekap peminjaman dapat dilihat pada Gambar 5.14 berikut.

Rekap Jumlah Peminjaman
Tahun 2019

Peminjaman	Jan	Feb	Mar	Apr	Mei	Jun	Jul	Agu	Sep	Okt	Nov	Des	Total
Semua	3	2	0	9	7	0	0	1	0	0	0	0	22
Kelas	0	1	0	2	4	0	0	0	0	0	0	0	7
Non Kelas	0	0	0	3	2	0	0	0	0	0	0	0	5
Barang	3	1	0	4	1	0	0	1	0	0	0	0	10

Gambar 5.16 Implementasi Antarmuka Halaman Rekap Peminjaman



BAB 6 PENGUJIAN

6.1 Pengujian *White-Box*

Pengujian *white-box* dilakukan untuk melakukan menguji alur program sistem. Pengujian *white-box* dilakukan pada tiga sampel *method* yang dipilih. Adapun *method* yang dipilih adalah *method* tambahJadwalKuliah(), historyPeminjaman(), dan validasiPeminjaman().

6.1.1 Pengujian Unit pada Method tambahJadwalKuliah()

Method tambahJadwalKuliah adalah *method* yang digunakan untuk proses menambah jadwal kuliah oleh admin. Adapun algoritma pada *method* tambahJadwalKuliah() dapat dilihat pada Tabel 6.1 berikut.

1. Algoritma *method* tambahJadwalKuliah()

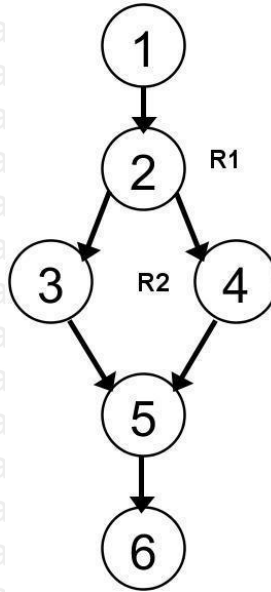
Tabel 6.1 Algoritma *Method* tambahJadwalKuliah()

Nama method		tambahJadwalKuliah()
No	<i>Pseudocode</i>	
		Node
1	Mulai	1
2	Get data input_jadwal_kuliah[]	2
3	if cek_jadwal_kuliah() = true	
4	Output = jadwal kuliah gagal ditambahkan	3
5	Redirect halaman tambah jadwal kuliah	
6	else	4
7	data = data_input_jadwal_kuliah[]	5
8	tambahJadwalKuliah(data)	
9	Output = jadwal kuliah berhasil ditambahkan	
10	Redirect halaman jadwal kuliah	
11	end If	6
12	Selesai	7

2. Basis Path Testing

a. Flow Graph

Flow Graph method tambahJadwalKuliah dapat dilihat pada Gambar 6.1.



Gambar 6.1 Flow Graph Method tambahJadwalKuliah()

b. Cyclomatic Complexity

$$- V(G) = R = 2$$

$$- V(G) = E - N + 2 = 6 - 6 + 2 = 2$$

$$- V(G) = P + 1 = 1 + 1 = 2$$

c. Independent Path

$$- \text{Jalur 1} = 1-2-3-5-6$$

$$- \text{Jalur 2} = 1-2-4-5-6$$

Berdasarkan penjelasan pada poin c, didapatkan dua jalur independen yang menjadi kasus uji *method* tambahJadwalKuliah(). Tabel 6.2 menunjukkan penjelasan kasus uji dari *method* tambahPeminjamanKelas().

Tabel 6.2 Test Case Method tambahJadwalKuliah()

Nomor Jalur	Prosedur Pengujian	Expected Result	Actual Result	Status
1	Form diisi lengkap dengan data inputan jam kuliah, ruangan dan hari sama dengan yang sudah ada di database. Data inputan	Jadwal kuliah gagal ditambahkan, Sistem menampilkan halaman data mata kuliah dan menampilkan pesan	Jadwal kuliah gagal ditambahkan, Sistem menampilkan halaman data mata kuliah dan menampilkan pesan	valid



	<ul style="list-style-type: none"> - Mata kuliah = HKA4003 - Dosen : 9036346765 - Jam Kuliah : 06:45-08:25 - Ruang : A.23 - Hari : Senin - Program Studi : Sarjana Ilmu Hukum - Kelas : A - Status Jadwal : aktif 	<p>pemberitahuan "jadwal kuliah gagal ditambahkan, jadwal ada yang bertabrakan"</p>	<p>pemberitahuan "jadwal kuliah gagal ditambahkan, jadwal ada yang bertabrakan"</p>	
2	<p>Form diisi lengkap dengan data inputan jam kuliah, ruangan dan hari sama dengan yang sudah ada di database.</p> <p>Data inputan</p> <ul style="list-style-type: none"> - Mata kuliah = HKA4003 - Dosen : 9036346765 - Jam Kuliah : 06:45-08:25 - Ruang : A.23 - Hari : Selasa - Program Studi : Sarjana Ilmu Hukum - Kelas : A - Status Jadwal : aktif 	<p>Jadwal kuliah gagal ditambahkan, Sistem menampilkan halaman data mata kuliah dan pesan pemberitahuan "jadwal kuliah gagal ditambahkan, jadwal ada yang bertabrakan"</p>	<p>Jadwal kuliah berhasil ditambahkan, Sistem menampilkan halaman data mata kuliah dan pesan pemberitahuan "jadwal kuliah berhasil ditambahkan"</p>	valid

6.1.2 Pengujian Unit pada *Method* cekPeminjaman()

Method cekPeminjaman() adalah *method* yang digunakan untuk melakukan pengecekan status peminjaman oleh *guest*. Adapun algoritma pada *method* cekPeminjaman() dapat dilihat pada Tabel 6.3 berikut.

1. Algoritma *method* cekPeminjaman ()

Tabel 6.3 Algoritma *Method* cekPeminjaman()

Nama Method		cekPeminjaman ()
No	Pseudocode	
1	Mulai	1
2	Get Id Peminjaman	2
3	Get cekIdPeminjaman (id peminjaman)	

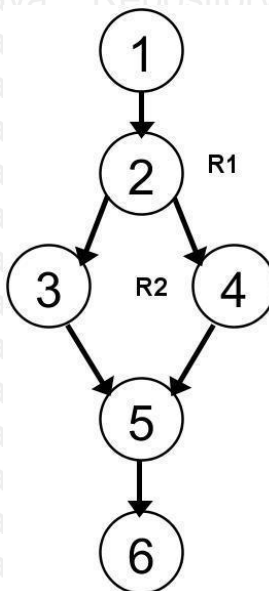


4	if data peminjaman != false	
5	data = get detail peminjaman	3
	redirect halaman detail peminjaman	
6	else	4
7	Output = id peminjaman tidak ditemukan	
8	Redirect method form cek peminjaman	
9	end if	5
10	Selesai	6

2. Basis Path Testing

a. Flow Graph

Flow Graph method cekPeminjaman dapat dilihat pada Gambar 6.4.



Gambar 6.2 Flow Graph Method cekPeminjaman()

b. Cyclomatic Complexity

$$- V(G) = R = 2$$

$$- V(G) = E - N + 2 = 6 - 6 + 2 = 2$$

$$- V(G) = P + 1 = 1 + 1 = 2$$

c. Independent Path

$$- \text{Jalur 1} = 1-2-3-5-6$$

$$- \text{Jalur 2} = 1-2-4-5-6$$



Berdasarkan penjelasan pada poin c, didapatkan dua jalur independen yang menjadi kasus uji *method* cekPeminjaman(). Tabel 6.4 menunjukkan penjelasan kasus uji dari *method* cekPeminjaman().

Tabel 6.4 Test Case Method cekPeminjaman()

Nomor Jalur	Prosedur Pengujian	Expected Result	Actual Result	Status
1	Form diisi lengkap dengan data inputan id peminjaman sama dengan yang ada tersimpan di database Data inputan - Id peminjaman = 201905290003	Sistem menampilkan halaman detail peminjaman berisi detail peminjaman dengan id 201905290003	Sistem menampilkan halaman detail peminjaman berisi detail peminjaman dengan id 201905290003	valid
2	Form diisi lengkap dengan data inputan id peminjaman tidak ada yang sama dengan yang ada tersimpan di database Data inputan - Id peminjaman = 201805290003	Sistem menampilkan halaman detail peminjaman berisi detail peminjaman dengan id 201805290003	Sistem menampilkan halaman detail peminjaman berisi detail peminjaman dengan id 201805290003	valid

6.1.3 Pengujian Unit pada *Method* formTambahPeminjamanKelas()

Method formTambahPeminjamanKelas() adalah *method* yang digunakan untuk menambah data peminjaman kelas oleh staf pelayanan.

1. Algoritma *method* formTambahPeminjamanKelas()

Tabel 6.5 Algoritma *Method* formTambahPeminjamanKelas()

Nama method	formTambahPeminjamanKelas(tanggal, id_ruangan, id_jam_kuliah)	
No	Pseudocode	Node
1	Mulai	1
2	Get data peminjaman[]	2
3	if cekPenggunaanRuanganDiJadwalKuliah == false	
4	if cekPenggunaanRuanganDiPeminjaman == false	3
5	load view v_tambah_peminjaman_kelas	4

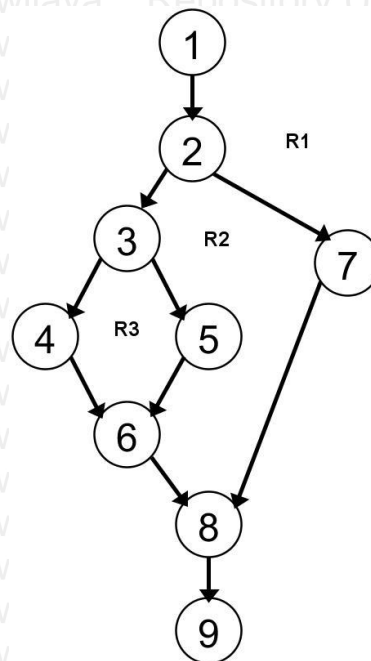


6	else	5
7	Output = ruangan tidak dapat dipinjam	
8	redirect view peta penggunaan ruangan kelas	
9	end if	6
10	else	7
11	Output = ruangan tidak dapat dipinjam	
12	redirect view peta penggunaan ruangan kelas	
13	end if	8
14	selesai	9

2. Basis Path Testing

a. Flow Graph

Flow Graph method form `TambahPeminjamanKelas()` dapat dilihat pada Gambar 6.4.



Gambar 6.3 Glow Graph Method form `TambahPeminjamanKelas()`

b. Cyclomatic Complexity

$$- V(G) = R = 3$$

$$- V(G) = E - N + 2 = 10 - 1 + 2 = 2$$

$$- V(G) = P + 1 = 2 + 1 = 3$$

c. Independent Path

$$- \text{Jalur 1} = 1-2-3-4-6-8-9$$



- Jalur 2 = 1-2-3-5-6-8-9

- Jalur 3 = 1-2-7-8-9

Berdasarkan penjelasan pada poin c, didapatkan dua jalur independen yang menjadi kasus uji *method* tambahPeminjamanBarang(). Tabel 6.6 menunjukkan penjelasan kasus uji dari *method* tambahPeminjamanBarang ().

Tabel 6.6 Test Case Method formTambahPeminjamanKelas()

Nomor Jalur	Prosedur Pengujian	Expected Result	Actual Result	Status
1	Memanggil method formTambahPeminjamanKelas(tanggal, id_ruangan, id_jam_kuliah) dengan nilai atribut tanggal = 2019-05-22, id_ruangan = 2, id_jam_kuliah = 2. Data atribut dibuat hingga tidak ada yang sama dengan data yang ada di database tabel jadwal kuliah dan database tabel peminjaman	Sistem menampilkan halaman tambah peminjaman kelas dengan keadaan field tanggal = data atribut tanggal, field ruangan = data atribut id ruangan, dan field jam kuliah = berdasarkan id jam kuliah	Sistem menampilkan halaman tambah peminjaman kelas dengan keadaan field tanggal = data atribut tanggal, field ruangan = data atribut id ruangan, dan field jam kuliah = berdasarkan id jam kuliah	valid
2	Memanggil method formTambahPeminjamanKelas(tanggal, id_ruangan, id_jam_kuliah) dengan nilai atribut tanggal = 2019-05-22, id_ruangan = 2, id_jam_kuliah = 2. Data atribut dibuat hingga tidak ada yang sama dengan data yang ada di database tabel jadwal kuliah namun ada yang sama dengan data yang ada di database tabel peminjaman	Sistem menampilkan halaman penggunaan ruangan kelas	Sistem menampilkan halaman penggunaan ruangan kelas	valid
	Memanggil method formTambahPeminjam	Sistem menampilkan	Sistem menampilkan	valid



anKelas(tanggal, id_ruangan, id_jam_kuliah) dengan nilai atribut tanggal = 2019-05-22, id_ruangan = 2, id_jam_kuliah = 2. Data atribut dibuat ada yang sama dengan data yang ada di database tabel jadwal kuliah	halaman penggunaan ruangan kelas	halaman penggunaan ruangan kelas	
--	----------------------------------	----------------------------------	--

6.2 Pengujian Validasi

Pengujian validasi dilakukan untuk menguji apakah semua fungsionalitas sudah sesuai dengan *use case scenario* yang sudah didefinisikan sebelumnya. Pengujian validasi dilakukan dengan menggunakan pengujian *black box*.

6.2.1 Pengujian Validasi Login

Hasil pengujian validasi login dapat dilihat pada Tabel 6.7 berikut.

Tabel 6.7 Pengujian Validasi Login

Nomo Kasus Uji	VAL_01			
Nama Kasus Uji	Kasus Uji Validasi <i>Login</i>			
No	Test Case	Input	Expected	Result Test
1	Aktor login dengan mengisi username dan password lengkap sesuai dengan yang terdaftar di sistem	Username = admin, Password = admin	Berhasil login	valid
2	Aktor login dengan mengisi username dan password lengkap namun tidak terdaftar di sistem	Username = admin Password = 123	Gagal login, sistem menampilkan pesan pemberitahuan "username atau password salah"	valid



3	Aktor login dengan username atau password tidak diisi	Username dan password kosong	Gagal login, sistem menampilkan pesan pemberitahuan "field harus diisi"	valid
---	---	------------------------------	---	-------

6.2.2 Pengujian Validasi Tambah Dosen

Hasil pengujian validasi tambah dosen dapat dilihat pada Tabel 6.8 berikut.

Tabel 6.8 Pengujian Validasi Tambah Dosen

Nomor Kasus Uji		VAL_02			
Nama Kasus Uji		Kasus Uji Validasi Tambah Dosen			
No	Test Case	Input	Expected	Result Test	
1	Aktor mengisi form tambah dosen dengan lengkap dengan nik terdFTAR di database	NIK = 155150201111309 Nama = Sueddi	Data dosen baru berhasil disimpan, sistem menampilkan pesan "data dosen berhasil ditambahkan"	valid	
2	Aktor mengisi form tambah dosen secara lengkap dengan NIK terdFTAR di database	NIK = 155150201111309 Nama = Sueddi	Data dosen baru gagal ditambahkan, sistem menampilkan pesan "nik 155150201111309 sudah terdFTAR di database"	valid	
3	Aktor mengisi form tambah dosen secara tidak lengkap	NIK = kosong Nama = kosong	Data dosen baru gagal ditambahkan, sistem menampilkan pesan "field harus diisi"	valid	



6.2.3 Pengujian Validasi Tambah Mahasiswa

Hasil pengujian validasi tambah mahasiswa dapat dilihat pada Tabel 6.9 berikut.

Tabel 6.9 Pengujian Validasi Tambah Mahasiswa

Nomor Uji	Kasus	VAL_03			
Nama Uji	Kasus	Kasus Uji Validasi Tambah Mahasiswa			
No	Test Case	Input	Expected	Result Test	
1	Aktor mengisi form tambah mahasiswa dengan lengkap dengan nik belum terdaftar di database	NIM = 155150201111309 Nama = Sueddi	Data mahasiswa baru berhasil disimpan, sistem menampilkan pesan "data mahasiswa berhasil ditambahkan"	valid	
2	Aktor mengisi form tambah dosen secara mahasiswa dengan NIM sudah terdaftar di database	NIM = 155150201111309 Nama = Sueddi	Data mahasiswa baru gagal ditambahkan, sistem menampilkan pesan "nim 155150201111309 sudah terdaftar di database"	valid	
3	Aktor mengisi form tambah mahasiswa secara tidak lengkap	NIK = kosong Nama = kosong	Data mahasiswa baru gagal ditambahkan, sistem menampilkan pesan "field harus diisi"	valid	



6.2.4 Pengujian Validasi Tambah Operator

Hasil pengujian validasi tambah operator dapat dilihat pada Tabel 6.10 berikut.

Tabel 6.10 Pengujian Validasi Tambah Operator

Nomor Kasus Uji		VAL_04			
Nama Kasus Uji		Kasus Uji Validasi Tambah Operator			
No	Test Case	Input	Expected	Result Test	
1	Aktor mengisi form tambah operator dengan lengkap dengan nik terdaftar di database	username = sueddi, password = sueddi, status = staff pelayanan	Data operator baru berhasil disimpan, sistem menampilkan pesan "data operator berhasil ditambahkan"	valid	
2	Aktor mengisi form dosen secara mahasiswa operator username sudah terdaftar di database	username = sueddi, password = sueddi, status = staff pelayanan	Data operator baru gagal ditambahkan, sistem menampilkan pesan "username sueddi sudah terdaftar di database"	valid	
3	Aktor mengisi form tambah operator secara tidak lengkap	username = kosong, password = kosong, status = kosong	Data operator baru gagal ditambahkan, sistem menampilkan pesan "field harus diisi"	valid	



6.2.5 Pengujian Validasi Tambah Ruangan

Hasil pengujian validasi tambah ruangan dapat dilihat pada Tabel 6.11 berikut.

Tabel 6.11 Pengujian Validasi Tambah Ruangan

Nomor Uji	Kasus	VAL_05			
Nama Uji	Kasus	Kasus Uji Validasi Tambah Ruangan			
No	Test Case		Input	Expected	Result Test
1	Aktor mengisi form tambah ruangan dengan lengkap	mengisi tambah ruangan dengan lengkap	Nama ruangan = E.12 Jenis ruangan = kelas	Data ruangan baru berhasil ditambahkan, sistem menampilkan pesan "data ruangan berhasil ditambahkan"	valid
2	Aktor mengisi form tambah ruangan dengan tidak lengkap	mengisi tambah ruangan dengan tidak lengkap	Nama ruangan = kosong, Jenis ruangan = kosong	Data ruangan baru gagal ditambahkan, sistem menampilkan pesan "field harus diisi"	valid

6.2.6 Pengujian Validasi Tambah Barang

Hasil pengujian validasi tambah barang dapat dilihat pada Tabel 6.12 berikut.

Tabel 6.12 Pengujian Validasi Tambah Barang

Nomor Uji	Kasus	VAL_06			
Nama Uji	Kasus	Kasus Uji Validasi Tambah Barang			
No	Test Case		Input	Expected	Result Test
1	Aktor mengisi form tambah barang dengan lengkap	mengisi tambah barang dengan lengkap	Nama barang = Mic Wireless	Data barang baru berhasil ditambahkan, sistem	valid



			menampilkan pesan “data barang berhasil ditambahkan”	
2	Aktor mengisi form tambah ruangan dengan tidak lengkap	Nama barang = kosong	Data barang baru gagal ditambahkan, sistem menampilkan pesan “field harus diisi”	valid

6.2.7 Pengujian Validasi Tambah Semester

Hasil pengujian validasi tambah semester dapat dilihat pada Tabel 6.13 berikut.

Tabel 6.13 Pengujian Validasi Tambah Semester

Nomor Kasus Uji	VAL_07			
Nama Kasus Uji	Kasus Uji Validasi Tambah Semester			
No	Test Case	Input	Expected	Result Test
1	Aktor mengisi form tambah semester dengan lengkap	Nama semester = Semester genap 2019/2020, Tanggal mulai = 01-08-2019, Tanggal selesai = 01-01-2020	Data semester baru berhasil ditambahkan, sistem menampilkan pesan “data semester berhasil ditambahkan”	valid
2	Aktor mengisi form tambah semester dengan tidak lengkap	Nama semester = kosong, Tanggal mulai = kosong, Tanggal selesai = kosong	Data semester baru gagal ditambahkan, sistem menampilkan pesan “field harus diisi”	valid



6.2.8 Pengujian Validasi Tambah Jam Kuliah

Hasil pengujian validasi tambah jam kuliah dapat dilihat pada Tabel 6.14 berikut.

Tabel 6.14 Pengujian Validasi Tambah Jam Kuliah

Nomor Uji	Kasus	VAL_08			
Nama Uji	Kasus	Kasus Uji Validasi Tambah Jam Kuliah			
No	Test Case	Input	Expected	Result Test	
1	Aktor mengisi form tambah jam kuliah dengan lengkap	Jam Kuliah = 10:15 - 11:55	Data jam kuliah baru berhasil ditambahkan, sistem menampilkan pesan "data jam kuliah berhasil ditambahkan"	valid	
2	Aktor mengisi form tambah jam kuliah dengan tidak lengkap	Jam kuliah = kosong	Data jam kuliah baru gagal ditambahkan, sistem menampilkan pesan "field harus diisi"	valid	

6.2.9 Pengujian Validasi Tambah Lembaga

Hasil pengujian validasi tambah lembaga dapat dilihat pada Tabel 6.15 berikut.

Tabel 6.15 Pengujian Validasi Tambah Lembaga

Nomor Uji	Kasus	VAL_09			
Nama Uji	Kasus	Kasus Uji Validasi Tambah Lembaga			
No	Test Case	Input	Expected	Result Test	
1	Aktor mengisi form tambah	Nama lembaga = ALSA	Data lembaga baru berhasil ditambahkan,	valid	



	lengkap dengan lembaga		sistem menampilkan pesan "data lembaga berhasil ditambahkan"	
2	Aktor mengisi form tambah lembaga dengan tidak lengkap	Nama lembaga = kosong	Data lembaga baru gagal ditambahkan, sistem menampilkan pesan "field harus diisi"	valid

6.2.10 Pengujian Validasi Tambah Jadwal Kuliah

Hasil pengujian validasi tambah jadwal kuliah dapat dilihat pada Tabel 6.15 berikut.

Tabel 6.16 Pengujian Validasi Tambah Jadwal Kuliah

Nomor Kasus Uji	VAL_10			
Nama Kasus Uji	Kasus Uji Validasi Tambah Jadwal Kuliah			
No	Test Case	Input	Expected	Result Test
1	Aktor mengisi form tambah jadwal kuliah dengan lengkap. Jadwal diisi nama ruangan, jam, dan hari tidak ada sama dengan yang sudah ada di <i>database</i>	Mata kuliah = HKA400 - Hukum Lingkungan, Dosen = 9240035234 - Sudarsono, Jam kuliah = 06:45 – 08:25, Ruangan = A 2.3, Hari = senin, Program studi = sarjana ilmu hukum, Kelas = A	Data jadwal kuliah baru berhasil ditambahkan, sistem menampilkan pesan "data lembaga berhasil ditambahkan"	valid



		Status aktif	Jadwal =	
2	Aktor mengisi form tambah jadwal kuliah dengan lengkap. Jadwal nama ruangan, jam, dan hari sama dengan yang sudah ada di <i>database</i>	Mata kuliah HKA400 - Hukum Lingkungan, Dosen = 9240035234 - Sudarsono, Jam kuliah = 06:45 - 08:25, Ruangan = A 2.3, Hari = senin, Program studi = sarjana ilmu hukum, Kelas = A	Status Jadwal = aktif	Data jadwal kuliah baru gagal ditambahkan, sistem menampilkan pesan "jadwal kuliah gagal ditambahkan, jadwal ada yang bertabrakan"
3	Aktor mengisi form tambah jadwal dengan tidak lengkap	Mata kuliah kosong, Dosen = kosong, Jam kuliah kosong, Ruangan = kosong, Hari = kosong, Program studi = kosong, Kelas = kosong	Status Jadwal = aktif	Data jadwal kuliah baru gagal ditambahkan, sistem menampilkan pesan " <i>field</i> harus diisi" valid



6.2.11 Pengujian Validasi Tambah Program Studi

Hasil pengujian validasi tambah program studi dapat dilihat pada Tabel 6.17 berikut.

Tabel 6.17 Pengujian Validasi Tambah Program Studi

No	Test Case	Input	Expected	Result Test
1	Aktor mengisi form tambah program studi dengan lengkap	Program studi= Sarjana Hukum	Data program studi baru berhasil ditambahkan, sistem menampilkan pesan "data program studi berhasil ditambahkan"	valid
2	Aktor mengisi form tambah program studi dengan tidak lengkap	Program studi= kosong	Data program studi baru gagal ditambahkan, sistem menampilkan pesan "field harus diisi"	valid



6.2.12 Pengujian Validasi Tambah Mata Kuliah

Hasil pengujian validasi tambah mata kuliah dapat dilihat pada Tabel 6.18 berikut.

Tabel 6.18 Pengujian Validasi Tambah Jadwal Kuliah

Nomor Kasus Uji		VAL_11			
Nama Kasus Uji		Kasus Uji Validasi Tambah Mata Kuliah			
No	Test Case	Input	Expected	Result Test	
1	Aktor mengisi form tambah mata kuliah dengan lengkap. Kode mata kuliah diisi dengan inputan yang tidak ada sama di database	Kode mata kuliah = HKA4003, Nama mata kuliah = Hukum Lingkungan	Data mata kuliah baru berhasil ditambahkan, sistem menampilkan pesan "data mata kuliah berhasil ditambahkan"	valid	
2	Aktor mengisi form tambah mata kuliah dengan lengkap. Kode mata kuliah diisi dengan inputan yang ada yang sama dengan data yang ada di database	Kode mata kuliah = HKA4003, Nama mata kuliah = Hukum Lingkungan	Data mata kuliah baru gagal ditambahkan, sistem menampilkan pesan "kode mata kuliah HKA4003 sudah terdaftar di database"	valid	
3	Aktor mengisi form tambah mata kuliah dengan tidak lengkap	Kode mata kuliah = kosong, Nama mata kuliah = kosong	Data mata kuliah baru gagal ditambahkan, sistem menampilkan pesan "field harus diisi"	valid	



6.2.13 Pengujian Validasi *Edit* Dosen

Hasil pengujian validasi edit dosen dapat dilihat pada Tabel 6.19 berikut.

Tabel 6.19 Pengujian Validasi *Edit* Dosen

Nomor Uji	Kasus	VAL_13			
Nama Uji	Kasus	Kasus Uji Validasi <i>Edit</i> Dosen			
No	Test Case	Input	Expected	Result Test	
1	Aktor mengisi form <i>edit</i> dosen dengan lengkap	NIK = 155150201111309 Nama = Sueddi	Data dosen baru berhasil disimpan, sistem menampilkan pesan "data dosen berhasil diubah"	valid	
2	Aktor mengisi form <i>edit</i> dosen secara tidak lengkap	NIK = 155150201111309 Nama = kosong	Data dosen baru gagal diedit, sistem menampilkan pesan "field harus diisi"	valid	

6.2.14 Pengujian Validasi *Edit* Mahasiswa

Hasil pengujian validasi *edit* mahasiswa dapat dilihat pada Tabel 6.20 berikut.

Tabel 6.20 Pengujian Validasi *Edit* Mahasiswa

Nomor Uji	Kasus	VAL_14			
Nama Uji	Kasus	Kasus Uji Validasi <i>Edit</i> Mahasiswa			
No	Test Case	Input	Expected	Result Test	
1	Aktor mengisi form <i>edit</i> mahasiswa dengan lengkap	NIM = 155150201111309 Nama = Sueddi	Data mahasiswa baru berhasil disimpan, sistem menampilkan pesan "data mahasiswa berhasil diubah"	valid	



2	Aktor mengisi form mahasiswa secara lengkap	mengisi <i>edit</i> tidak	NIK = 155150201111309 Nama = kosong	Data mahasiswa baru gagal diubah, sistem menampilkan pesan "field harus diisi"	valid
---	---	---------------------------	--	--	-------

6.2.15 Pengujian Validasi *Edit* Operator

Hasil pengujian validasi *edit* operator dapat dilihat pada Tabel 6.21 berikut.

Tabel 6.21 Pengujian Validasi *Edit* Operator

Nomor Kasus Uji		VAL_15			
Nama Kasus Uji		Kasus Uji Validasi <i>Edit</i> Operator			
No	<i>Test Case</i>	<i>Input</i>	<i>Expected</i>	<i>Result Test</i>	
1	Aktor mengisi form operator lengkap	dengan	username = sueddi, password = sueddi, status = staff pelayanan	Data operator baru berhasil disimpan, sistem menampilkan pesan "data operator berhasil diubah"	valid
2	Aktor mengisi form operator secara tidak lengkap		username = sueddi, password = kosong, status = kosong	Data operator baru gagal diubah, sistem menampilkan pesan "field harus diisi"	valid



6.2.16 Pengujian Validasi *Edit* Ruangan

Hasil pengujian validasi *edit* ruangan dapat dilihat pada Tabel 6.21 berikut.

Tabel 6.22 Pengujian Validasi *Edit* Ruangan

Nomor Uji	Kasus	VAL_16			
Nama Uji	Kasus	Kasus Uji Validasi <i>Edit</i> Ruangan			
No	Test Case	Input	Expected	Result Test	
1	Aktor mengisi form <i>edit</i> ruangan dengan lengkap	Nama ruangan = E.12 Jenis ruangan = kelas	Data ruangan baru berhasil diubah, sistem menampilkan pesan "data ruangan berhasil diubah"	valid	
2	Aktor mengisi form <i>edit</i> ruangan dengan tidak lengkap	Nama ruangan = kosong, Jenis ruangan = kosong	Data ruangan baru gagal diubah, sistem menampilkan pesan "field harus diisi"	valid	

6.2.17 Pengujian Validasi *Edit* Barang

Hasil pengujian validasi *edit* barang dapat dilihat pada Tabel 6.22 berikut.

Tabel 6.23 Pengujian Validasi *Edit* Barang

Nomor Uji	Kasus	VAL_17			
Nama Uji	Kasus	Kasus Uji Validasi <i>Edit</i> Barang			
No	Test Case	Input	Expected	Result Test	
1	Aktor mengisi form <i>edit</i> barang dengan lengkap	Nama barang = <i>Mic Wireless</i>	Data barang baru berhasil diubah, sistem menampilkan pesan "data	valid	



			barang berhasil diubah”	
2	Aktor mengisi form <i>edit</i> barang dengan tidak lengkap	Nama barang = kosong	Data barang baru gagal diubah, sistem menampilkan pesan “ <i>field</i> harus diisi”	valid

6.2.18 Pengujian Validasi *Edit* Semester

Hasil pengujian validasi *edit* semester dapat dilihat pada Tabel 6.24 berikut.

Tabel 6.24 Pengujian Validasi *Edit* Semester

Nomor Kasus Uji	VAL_18			
Nama Kasus Uji	Kasus Uji Validasi <i>Edit</i> Semester			
No	<i>Test Case</i>	<i>Input</i>	<i>Expected</i>	<i>Result Test</i>
1	Aktor mengisi form <i>edit</i> semester dengan lengkap	Nama semester = Semester genap 2019/2020, Tanggal mulai = 01-08-2019, Tanggal selesai = 01-01-2020	Data semester baru berhasil diubah, sistem menampilkan pesan “data semester berhasil diubah”	valid
2	Aktor mengisi form <i>edit</i> semester dengan tidak lengkap	Nama semester = kosong, Tanggal mulai = kosong, Tanggal selesai = kosong	Data semester baru gagal diubah, sistem menampilkan pesan “ <i>field</i> harus diisi”	valid



6.2.19 Pengujian Validasi *Edit* Jam Kuliah

Hasil pengujian validasi *edit* jam kuliah dapat dilihat pada Tabel 6.25 berikut.

Tabel 6.25 Pengujian Validasi *Edit* Jam Kuliah

Nomor Uji	Kasus	VAL_19			
Nama Uji	Kasus	Kasus Uji Validasi <i>Edit</i> Jam Kuliah			
No	Test Case	Input	Expected	Result Test	
1	Aktor mengisi form <i>edit</i> jam kuliah dengan lengkap	Jam kuliah = 10:15 - 11:55	Data jam kuliah baru berhasil diubah, sistem menampilkan pesan "data jam kuliah berhasil ditambahkan"	valid	
2	Aktor mengisi form <i>edit</i> jam kuliah dengan tidak lengkap	Jam kuliah = kosong	Data jam kuliah baru gagal diubah, sistem menampilkan pesan "field harus diisi"	valid	

6.2.20 Pengujian Validasi *Edit* Lembaga

Hasil pengujian validasi *edit* lembaga dapat dilihat pada Tabel 6.26 berikut.

Tabel 6.26 Pengujian Validasi *Edit* Lembaga

Nomor Uji	Kasus	VAL_20			
Nama Uji	Kasus	Kasus Uji Validasi <i>Edit</i> Lembaga			
No	Test Case	Input	Expected	Result Test	
1	Aktor mengisi form <i>edit</i> lembaga dengan lengkap	Nama lembaga = ALSA	Data lembaga baru berhasil diubah, sistem menampilkan pesan "data	valid	



			lembaga berhasil diubah”	
2	Aktor mengisi form <i>edit</i> lembaga dengan tidak lengkap	Nama lembaga = kosong	Data lembaga baru gagal diubah, sistem menampilkan pesan “ <i>field</i> harus diisi”	valid

6.2.21 Pengujian Validasi *Edit* Jadwal Kuliah

Hasil pengujian validasi *edit* jadwal kuliah dapat dilihat pada Tabel 6.27 berikut.

Tabel 6.27 Pengujian Validasi *Edit* Jadwal Kuliah

Nomor Kasus Uji	VAL_21			
Nama Kasus Uji	Kasus Uji Validasi <i>Edit</i> Jadwal Kuliah			
No	<i>Test Case</i>	<i>Input</i>	<i>Expected</i>	<i>Result Test</i>
1	Aktor mengisi form <i>edit</i> jadwal kuliah dengan lengkap. Jadwal diisi nama ruangan, jam, dan hari tidak ada sama dengan yang sudah ada di <i>database</i>	Mata kuliah = HKA400 - Hukum Lingkungan, Dosen = 9240035234 - Sudarsono, Jam kuliah = 06:45 – 08:25, Ruangan = A 2.3, Hari = senin, Program studi = sarjana ilmu hukum, Kelas = A Status Jadwal = aktif	Data jadwal kuliah baru berhasil diubah, sistem menampilkan pesan “data lembaga berhasil diubah”	valid



2	Aktor mengisi form <i>edit</i> jadwal kuliah dengan lengkap. Jadwal nama ruangan, jam, dan hari sama dengan yang sudah ada di <i>database</i>	Mata kuliah = HKA400 - Hukum Lingkungan, Dosen = 9240035234 - Sudarsono, Jam kuliah = 06:45 – 08:25, Ruangan = A 2.3, Hari = senin, Program studi = sarjana ilmu hukum, Kelas = A Status Jadwal = aktif	Data jadwal kuliah baru gagal diubah, sistem menampilkan pesan “jadwal kuliah gagal diubah, jadwal ada yang bertabrakan”	
3	Aktor mengisi form <i>edit</i> jadwal kuliah dengan tidak lengkap	Mata kuliah = kosong, Dosen = kosong, Jam kuliah = kosong, Ruangan = kosong, Hari = kosong, Program studi = kosong, Kelas = kosong Status Jadwal = aktif	Data jadwal kuliah baru gagal diubah, sistem menampilkan pesan “ <i>field</i> harus diisi”	valid



6.2.22 Pengujian Validasi *Edit* Program Studi

Hasil pengujian validasi *edit* program studi dapat dilihat pada Tabel 6.28 berikut.

Tabel 6.28 Pengujian Validasi *Edit* Program Studi

Nomor Uji	Kasus	VAL_22			
Nama Uji	Kasus	Kasus Uji Validasi <i>Edit</i> Program Studi			
No	Test Case		Input	Expected	Result Test
1	Aktor mengisi form <i>edit</i> program studi dengan lengkap	mengisi <i>edit</i> studi	Program studi= Sarjana Hukum	Data program studi baru berhasil diubah, sistem menampilkan pesan “data program studi berhasil diubah”	valid
2	Aktor mengisi form <i>edit</i> program studi dengan tidak lengkap	mengisi <i>edit</i> studi tidak	Program studi= kosong	Data program studi baru gagal diubah, sistem menampilkan pesan “field harus diisi”	valid

6.2.23 Pengujian Validasi *Edit* Mata Kuliah

Hasil pengujian validasi *edit* mata kuliah dapat dilihat pada Tabel 6.29 berikut.

Tabel 6.29 Pengujian Validasi *Edit* Mata Kuliah

Nomor Uji	Kasus	VAL_23			
Nama Uji	Kasus	Kasus Uji Validasi <i>Edit</i> Mata Kuliah			
No	Test Case		Input	Expected	Result Test
1	Aktor mengisi form <i>edit</i> mata kuliah dengan lengkap.	mengisi <i>edit</i> mata kuliah dengan	Kode mata kuliah = HKA4003,	Data mata kuliah baru berhasil diubah, sistem menampilkan	valid



	Kode mata kuliah diisi dengan inputan yang tidak ada yang sama di database	Nama mata kuliah = Hukum Lingkungan	pesan "data mata kuliah berhasil diubah"	
2	Aktor mengisi form <i>edit</i> mata kuliah dengan tidak lengkap	Kode mata kuliah = HKA4003, Nama mata kuliah = kosong	Data mata kuliah baru gagal diubah, sistem menampilkan pesan " <i>field</i> harus diisi"	valid

6.2.24 Pengujian Validasi Hapus Dosen

Hasil pengujian validasi hapus dosen dapat dilihat pada Tabel 6.30 berikut

Tabel 6.30 Pengujian Validasi Hapus Dosen

Nomor Kasus Uji	VAL_24			
Nama Kasus Uji	Kasus Uji Validasi Hapus Dosen			
No	Test Case	Input	Expected	Result Test
1	Aktor menghapus data dosen	Id dosen = 155150201111309	Data dosen berhasil dihapus, sistem menampilkan pesan "data dosen berhasil dihapus"	valid



6.2.25 Pengujian Validasi Hapus Mahasiswa

Hasil pengujian validasi hapus mahasiswa dapat dilihat pada Tabel 6.30 berikut.

Tabel 6.31 Pengujian Validasi Hapus Mahasiswa

Nomor Uji	Kasus	VAL_25		
Nama Uji	Kasus	Kasus Uji Validasi Hapus Mahasiswa		
No	Test Case	Input	Expected	Result Test
1	Aktor menghapus data mahasiswa	Id mahasiswa = 155150201111309	Data mahasiswa berhasil dihapus, sistem menampilkan pesan "data mahasiswa berhasil dihapus"	valid

6.2.26 Pengujian Validasi Hapus Operator

Hasil pengujian validasi hapus operator dapat dilihat pada Tabel 6.32 berikut.

Tabel 6.32 Pengujian Validasi Hapus Operator

Nomor Uji	Kasus	VAL_26		
Nama Uji	Kasus	Kasus Uji Validasi Hapus Operator		
No	Test Case	Input	Expected	Result Test
1	Aktor menghapus data operator	username = sueddi,	Data operator berhasil dihapus, sistem menampilkan pesan "data operator berhasil dihapus"	valid



6.2.27 Pengujian Validasi Hapus Ruangan

Hasil pengujian validasi hapus ruangan dapat dilihat pada Tabel 6.33 berikut.

Tabel 6.33 Pengujian Validasi Hapus Ruangan

Nomor Uji	Kasus	VAL_27		
Nama Uji	Kasus	Kasus Uji Validasi Hapus Ruangan		
No	Test Case	Input	Expected	Result Test
1	Aktor menghapus data ruangan	Id_ruangan = 2	Data ruangan berhasil dihapus, sistem menampilkan pesan "data ruangan berhasil dihapus"	valid

6.2.28 Pengujian Validasi Hapus Barang

Hasil pengujian validasi hapus barang dapat dilihat pada Tabel 6.34 berikut.

Tabel 6.34 Pengujian Validasi Hapus Barang

Nomor Uji	Kasus	VAL_28		
Nama Uji	Kasus	Kasus Uji Validasi Hapus Barang		
No	Test Case	Input	Expected	Result Test
1	Aktor menghapus data barang	Id barang = 2	Data barang berhasil dihapus, sistem menampilkan pesan "data barang berhasil dihapus"	valid



6.2.29 Pengujian Validasi Hapus Semester

Hasil pengujian validasi hapus semester dapat dilihat pada Tabel 6.34 berikut.

Tabel 6.35 Pengujian Validasi Hapus Semester

Nomor Uji	Kasus	VAL_29		
Nama Uji	Kasus	Kasus Uji Validasi Hapus Semester		
No	Test Case	Input	Expected	Result Test
1	Aktor menghapus data semester	Id semester = 2	Data semester dan data jadwal kuliah dengan id 2 berhasil dihapus, sistem menampilkan pesan "data semester berhasil dihapus"	valid

6.2.30 Pengujian Validasi Hapus Jam Kuliah

Hasil pengujian validasi hapus jam kuliah dapat dilihat pada Tabel 6.36 berikut.

Tabel 6.36 Pengujian Validasi Hapus Jam Kuliah

Nomor Uji	Kasus	VAL_30		
Nama Uji	Kasus	Kasus Uji Validasi Hapus Jam Kuliah		
No	Test Case	Input	Expected	Result Test
1	Aktor menghapus data jam kuliah	Id jam kuliah = 1	Data jam kuliah berhasil dihapus, sistem menampilkan pesan "data jam kuliah berhasil dihapus"	valid



6.2.31 Pengujian Validasi Hapus Lembaga

Hasil pengujian validasi hapus lembaga dapat dilihat pada Tabel 6.37 berikut.

Tabel 6.37 Pengujian Validasi Hapus Lembaga

Nomor Uji	Kasus	VAL_31			
Nama Uji	Kasus	Kasus Uji Validasi Hapus Lembaga			
No	Test Case	Input	Expected	Result Test	
1	Aktor menghapus data lembaga	Nama lembaga = ALSA	Data lembaga baru berhasil dihapus, sistem menampilkan pesan "data lembaga berhasil dihapus"	valid	

6.2.32 Pengujian Validasi Hapus Jadwal Kuliah

Hasil pengujian validasi hapus jadwal kuliah dapat dilihat pada Tabel 6.38 berikut.

Tabel 6.38 Pengujian Validasi Hapus Jadwal Kuliah

Nomor Uji	Kasus	VAL_32			
Nama Uji	Kasus	Kasus Uji Validasi Hapus Jadwal Kuliah			
No	Test Case	Input	Expected	Result Test	
1	Aktor menghapus data jadwal kuliah	Id jadwal kuliah = 123	Data jadwal kuliah baru berhasil dihapus, sistem menampilkan pesan "data lembaga berhasil dihapus"	valid	



6.2.33 Pengujian Validasi Hapus Program Studi

Hasil pengujian validasi hapus program studi dapat dilihat pada Tabel 6.39 berikut.

Tabel 6.39 Pengujian Validasi Hapus Program Studi

Nomor Uji	Kasus	VAL_33		
Nama Uji	Kasus	Kasus Uji Validasi Hapus Program Studi		
No	Test Case	Input	Expected	Result Test
1	Aktor menghapus data program studi	Id program studi = 1	Data program studi baru berhasil dihapus, sistem menampilkan pesan “data program studi berhasil dihapus”	valid

6.2.34 Pengujian Validasi Hapus Mata Kuliah

Hasil pengujian validasi hapus mata kuliah dapat dilihat pada Tabel 6.40 berikut.

Tabel 6.40 Pengujian Validasi Hapus Mata Kuliah

Nomor Uji	Kasus	VAL_34		
Nama Uji	Kasus	Kasus Uji Validasi Hapus Mata Kuliah		
No	Test Case	Input	Expected	Result Test
1	Aktor menghapus data mata kuliah	Id mata kuliah = 18	Data mata kuliah baru berhasil dihapus, sistem menampilkan pesan “data mata kuliah berhasil dihapus”	valid



6.2.35 Pengujian Validasi Lihat *History* Peminjaman

Hasil pengujian validasi lihat *history* peminjaman dapat dilihat pada Tabel 6.41 berikut.

Tabel 6.41 Pengujian Validasi Lihat *History* Peminjaman

Nomor Uji	Kasus	VAL_35			
Nama Uji	Kasus	Kasus Uji Validasi Melihat <i>History</i> Peminjaman			
No	<i>Test Case</i>		<i>Input</i>	<i>Expected</i>	<i>Result Test</i>
1	Aktor data peminjaman	melihat <i>history</i>	status operator = sekretariat kuliah	Sistem menampilkan halaman <i>history</i> peminjaman	valid

6.2.36 Pengujian Validasi Lihat Rekap Peminjaman

Hasil pengujian validasi lihat rekap peminjaman dapat dilihat pada Tabel 6.42 berikut.

Tabel 6.42 Pengujian Validasi Lihat Rekap Peminjaman

Nomor Uji	Kasus	VAL_36			
Nama Uji	Kasus	Kasus Uji Validasi Lihat Rekap Peminjaman			
No	<i>Test Case</i>		<i>Input</i>	<i>Expected</i>	<i>Result Test</i>
1	Aktor data peminjaman	melihat rekap	Tahun = 2019	Sistem menampilkan halaman rekap peminjaman	valid



6.2.37 Pengujian Validasi Lihat Rekap Penggunaan Ruangan

Hasil pengujian validasi lihat rekap penggunaan ruangan dapat dilihat pada Tabel 6.43 berikut.

Tabel 6.43 Pengujian Validasi Lihat Rekap Penggunaan Ruangan

Nomor Uji	Kasus	VAL_37			
Nama Uji	Kasus	Kasus Uji Validasi Lihat Rekap Penggunaan Ruangan			
No	Test Case		Input	Expected	Result Test
1	Aktor melihat data rekap penggunaan ruangan		Tahun = 2019	Sistem menampilkan halaman rekap penggunaan ruangan	valid

6.2.38 Pengujian Validasi Lihat Rekap Penggunaan Barang

Hasil pengujian validasi lihat rekap penggunaan barang dapat dilihat pada Tabel 6.44 berikut.

Tabel 6.44 Pengujian Validasi Lihat Rekap Penggunaan Barang

Nomor Uji	Kasus	VAL_38			
Nama Uji	Kasus	Kasus Uji Validasi Lihat Rekap Penggunaan Barang			
No	Test Case		Input	Expected	Result Test
1	Aktor melihat data rekap penggunaan barang		Tahun = 2019	Sistem menampilkan halaman rekap penggunaan barang	valid



6.2.39 Pengujian Validasi Lihat *History Complaint*

Hasil pengujian validasi melihat *history complaint* barang dapat dilihat pada Tabel 6.45 berikut.

Tabel 6.45 Pengujian Validasi Lihat *History Complaint*

Nomor Uji	Kasus	VAL_39			
Nama Uji	Kasus	Kasus Uji Validasi Melihat <i>History Complaint</i>			
No	Test Case		Input	Expected	Result Test
1	Aktor data <i>complaint</i>	melihat <i>history</i>	Status operator = sekretariat kuliah	Sistem menampilkan halaman <i>history complaint</i>	valid

6.2.40 Pengujian Validasi Lihat Rekap *Complaint*

Hasil pengujian validasi lihat rekap *complaint* dapat dilihat pada Tabel 6.45 berikut.

Tabel 6.46 Pengujian Validasi Lihat Rekap *Complaint*

Nomor Uji	Kasus	VAL_40			
Nama Uji	Kasus	Kasus Uji Validasi Melihat Rekap <i>Complaint</i>			
No	Test Case		Input	Expected	Result Test
1	Aktor data <i>complaint</i>	melihat rekap	Tahun = 2019	Sistem menampilkan halaman rekap <i>complaint</i>	valid



6.2.41 Pengujian Validasi Lihat Jadwal Kuliah

Hasil pengujian validasi lihat jadwal kuliah dapat dilihat pada Tabel 6.47 berikut.

Tabel 6.47 Pengujian Validasi Lihat Jadwal Kuliah

Nomor Uji	Kasus	VAL_41		
Nama Uji	Kasus	Kasus Uji Validasi Lihat Jadwal Kuliah		
No	Test Case	Input	Expected	Result Test
1	Aktor melihat jadwal kuliah	Tanggal = 30-05-2019	Sistem menampilkan halaman jadwal kuliah dengan data jadwal kuliah yang tampil adalah tanggal 30-05-2019	valid

6.2.42 Pengujian Validasi Tambah *Complaint*

Hasil pengujian validasi tambah *complaint* dapat dilihat pada Tabel 6.48 berikut.

Tabel 6.48 Pengujian Validasi Tambah *Complaint*

Nomor Uji	Kasus	VAL_42		
Nama Uji	Kasus	Kasus Uji Validasi Tambah <i>Complaint</i>		
No	Test Case	Input	Expected	Result Test
1	Aktor mengisi form tambah <i>complaint</i> dengan lengkap	Nim = 155150201111309 Program studi = 2, Ruangan = E.21, Keluhan proyektor menyala tidak	Data complaint baru berhasil ditambahkan. Sistem menampilkan pesan " <i>complaint</i> berhasil dikirim"	valid



2	Aktor mengisi form tambah <i>complaint</i> dengan tidak lengkap	Nim = kosong, Program studi = kosong, Ruangan = kosong, Keluhan = kosong	Data complaint baru gagal ditambahkan. Sistem menampilkan pesan “field ini harus diisi”	valid
---	---	---	---	-------

6.2.43 Pengujian Validasi Cek Status Complaint

Hasil pengujian validasi cek status *complaint* dapat dilihat pada Tabel 6.49 berikut.

Tabel 6.49 Pengujian Cek Status Complaint

Nomor Kasus Uji	VAL_43			
Nama Kasus Uji	Kasus Uji Validasi Cek Status <i>Complaint</i>			
No	Test Case	Input	Expected	Result Test
1	Aktor mengisi form cek status <i>complaint</i> dengan lengkap	Id <i>complaint</i> = 1201905270005	Data complaint berhasil ditampilkan. Sistem menampilkan halaman detail <i>complaint</i>	valid
2	Aktor mengisi form cek status <i>complaint</i> dengan tidak lengkap	Id <i>complaint</i> = kosong	Data complaint tidak ditampilkan. Sistem menampilkan pesan “isi isian ini”	valid



6.2.44 Pengujian Validasi Cek Status Peminjaman

Hasil pengujian validasi cek status peminjaman dapat dilihat pada Tabel 6.50 berikut.

Tabel 6.50 Pengujian Validasi Cek Status Peminjaman

Nomor Kasus Uji	VAL_44			
Nama Kasus Uji	Kasus Uji Validasi Cek Status <i>Peminjaman</i>			
No	Test Case	Input	Expected	Result Test
1	Aktor mengisi form cek status peminjaman dengan lengkap	Id peminjaman = 1201905270005	Data peminjaman berhasil ditampilkan. Sistem menampilkan halaman detail peminjaman	valid
2	Aktor mengisi form cek status peminjaman dengan tidak lengkap	Id peminjaman = kosong	Data peminjaman tidak ditampilkan. Sistem menampilkan pesan "isi isian ini"	valid

6.2.45 Pengujian Validasi Lihat Status Penggunaan Ruang Kelas

Hasil pengujian validasi lihat status penggunaan ruangan kelas dapat dilihat pada Tabel 6.51 berikut.

Tabel 6.51 Pengujian Validasi Lihat Status Penggunaan Ruang Kelas

Nomor Kasus Uji	VAL_45			
Nama Kasus Uji	Kasus Uji Validasi Lihat Status Penggunaan Ruang Kelas			
No	Test Case	Input	Expected	Result Test
1	Aktor melihat status penggunaan ruangan kelas	Tanggal = 30-05-2019	Sistem menampilkan halaman status penggunaan ruangan kelas	valid



			dengan data penggunaan ruangan yang tampil adalah tanggal 30-05-2019	
--	--	--	--	--

6.2.46 Pengujian Validasi Lihat Status Penggunaan Ruang Non Kelas

Hasil pengujian validasi lihat status penggunaan ruangan non kelas dapat dilihat pada Tabel 6.52 berikut.

Tabel 6.52 Pengujian Validasi Lihat Status Penggunaan Ruang Non Kelas

Nomor Kasus Uji	VAL_46			
Nama Kasus Uji	Kasus Uji Validasi Lihat Status Penggunaan Ruang Non Kelas			
No	Test Case	Input	Expected	Result Test
1	Aktor melihat status penggunaan ruangan non kelas	Tanggal = 30-05-2019	Sistem menampilkan halaman status penggunaan ruangan non kelas dengan data penggunaan ruangan yang tampil adalah tanggal 30-05-2019	valid



6.2.47 Pengujian Validasi Lihat Status Penggunaan Barang

Hasil pengujian validasi lihat status penggunaan Barang dapat dilihat pada Tabel 6.53 berikut.

Tabel 6.53 Pengujian Validasi Lihat Status Penggunaan Barang

Nomor Uji	Kasus	VAL_47		
Nama Uji	Kasus	Kasus Uji Validasi Lihat Status Penggunaan Barang		
No	Test Case	Input	Expected	Result Test
1	Aktor melihat status penggunaan barang	Tanggal = 30-05-2019	Sistem menampilkan halaman status penggunaan barang dengan data penggunaan barang yang tampil adalah tanggal 30-05-2019	valid

6.2.48 Pengujian Validasi Validasi Peminjaman

Hasil pengujian validasi validasi peminjaman dapat dilihat pada Tabel 6.54 berikut.

Tabel 6.54 Pengujian Validasi Validasi Peminjaman

Nomor Uji	Kasus	VAL_48		
Nama Uji	Kasus	Kasus Uji Validasi Validasi Peminjaman		
No	Test Case	Input	Expected	Result Test
1	Aktor dapat melakukan validasi peminjaman	Id peminjaman = 1201905270005	Peminjaman berhasil divalidasi. Sistem menampilkan pesan "data peminjaman berhasil disetujui"	valid



6.2.49 Pengujian Validasi Tolak Peminjaman

Hasil pengujian validasi tolak peminjaman dapat dilihat pada Tabel 6.55 berikut.

Tabel 6.55 Pengujian Validasi Tolak Peminjaman

Nomor Uji	Kasus	VAL_49		
Nama Uji	Kasus	Kasus Uji Validasi Tolak Peminjaman		
No	Test Case	Input	Expected	Result Test
1	Aktor mengisi form tolak peminjaman dengan lengkap	Id peminjaman = 1201905270005, Catatan penolakan sarana sedang diperbaiki	Peminjaman berhasil ditolak. Sistem menampilkan pesan "data peminjaman berhasil ditolak"	valid
2	Aktor mengisi form tolak peminjaman dengan tidak lengkap	Id peminjaman = 1201905270005, Catatan penolakan kosong	Peminjaman gagal ditolak, sistem menampilkan pesan "isi isian ini"	valid

6.2.50 Pengujian Validasi Batal Peminjaman

Hasil pengujian validasi batal peminjaman dapat dilihat pada Tabel 6.56 berikut.

Tabel 6.56 Pengujian Validasi Batal Peminjaman

Nomor Uji	Kasus	VAL_50		
Nama Uji	Kasus	Kasus Uji Validasi Batal Peminjaman		
No	Test Case	Input	Expected	Result Test
1	Aktor dapat melakukan batal peminjaman	Id peminjaman = 1201905270005	Peminjaman berhasil dibatalkan. Sistem menampilkan pesan "data	valid



			peminjaman berhasil dibatalkan”	
--	--	--	---------------------------------	--

6.2.51 Pengujian Validasi Tambah Peminjaman Kelas

Hasil pengujian validasi tambah peminjaman kelas dapat dilihat pada Tabel 6.56 berikut.

Tabel 6.57 Pengujian Validasi Tambah Peminjaman Kelas

Nomor Kasus Uji	VAL_51			
Nama Kasus Uji	Kasus Uji Validasi Lihat Tambah Peminjaman Kelas			
No	Test Case	Input	Expected	Result Test
1	Aktor mengisi form tambah peminjaman kelas dengan lengkap, id peminjaman dimasukkan tidak dengan id mahasiswa yang terdaftar di <i>database</i>	Nim peminjam = 15515020111130, Semester = semester genap 2019, Mata kuliah = HKA4003 – Hukum Lingkungan, Dosen = 9240035234 – Sudarsono, Tanggal penggunaan = 30-05-2019, Jam kuliah = 12:00 – 13:00, Ruangan = A.23, Program studi = sarjana ilmu hukum, Kelas = A, Keterangan = jadwal pengganti	Data peminjaman kelas baru berhasil ditambahkan. Sistem menampilkan pesan “peminjaman kelas berhasil ditambahkan”	valid



2	Aktor mengisi form tambah peminjaman kelas dengan lengkap, id peminjaman dimasukkan tidak dengan id mahasiswa yang tidak terdaftar di <i>database</i>	Nim peminjam = 15515313132132, Semester = semester genap 2019, Mata kuliah = HKA4003 – Hukum Lingkungan, Dosen = 9240035234 – Sudarsono, Tanggal penggunaan = 30-05-2019, Jam kuliah = 12:00 – 13:00, Ruangan = A.23, Program studi = sarjana ilmu hukum, Kelas = A, Keterangan = jadwal pengganti	Data peminjaman kelas baru gagal ditambahkan. Sistem menampilkan halaman tambah peminjaman kelas dengan menampilkan pesan “data mahasiswa tidak ditemukan”	valid
3	Aktor mengisi form tambah peminjaman kelas dengan tidak lengkap	Nim peminjaman = kosong, Semester = kosong, Mata kuliah = kosong, Dosen = kosong, Tanggal penggunaan =kosong, Jam kuliah = kosong, Ruangan = kosong,	Data peminjaman kelas baru gagal ditambahkan. Sistem menampilkan halaman tambah peminjaman kelas dengan menampilkan pesan “isi isian ini”	valid



		Program studi = kosong, Kelas = kosong, Keterangan = kosong		
--	--	---	--	--

6.2.52 Pengujian Validasi Tambah Peminjaman Non Kelas

Hasil pengujian validasi tambah peminjaman non kelas dapat dilihat pada Tabel 6.58 berikut.

Tabel 6.58 Pengujian Validasi Tambah Peminjaman Non Kelas

Nomor Kasus Uji	VAL_52			
Nama Kasus Uji	Kasus Uji Validasi Lihat Tambah Peminjaman Non Kelas			
No	Test Case	Input	Expected	Result Test
1	Aktor mengisi form tambah peminjaman non kelas secara lengkap dengan id peminjam menggunakan id mahasiswa yang terdaftar di database	Nim peminjam = 155150201111309 , Tanggal mulai penggunaan = 30-05-2019, Tanggal selesai penggunaan = 02-06-2019, Jam mulai = 06.00, Jam selesai = 09.59, Lembaga = ALSA, Keterangan = acara pelatihan, Id ruangan = 28	Data peminjaman non kelas baru berhasil ditambahkan. Sistem menampilkan halaman tambah sarana peminjaman	valid
2	Aktor mengisi form tambah peminjaman non kelas secara	Nim peminjaman = 1551504949459,	Data peminjaman non kelas baru gagal ditambahkan.	valid



	lengkap dengan id peminjam menggunakan id mahasiswa yang tidak terdaftar di database	Tanggal mulai penggunaan = 30-05-2019, Tanggal selesai penggunaan = 02-06-2019, Jam mulai = 06.00, Jam selesai = 09.59, Lembaga = ALSA, Keterangan = acara pelatihan	Sistem menampilkan halaman tambah peminjaman non kelas dengan menampilkan pesan “data mahasiswa tidak ditemukan”	
3	Aktor mengisi form tambah peminjaman non kelas dengan tidak lengkap	Nim peminjaman = kosong, Tanggal mulai penggunaan = kosong, Tanggal selesai penggunaan = kosong, Jam mulai = kosong, Jam selesai = kosong, Lembaga = kosong, Keterangan = kosong	Data peminjaman non kelas baru gagal ditambahkan. Sistem menampilkan halaman tambah peminjaman non kelas dengan menampilkan pesan “isi isian ini”	valid



6.2.53 Pengujian Validasi Tambah Peminjaman Barang

Hasil pengujian validasi tambah peminjaman barang dapat dilihat pada Tabel 6.59 berikut.

Tabel 6.59 Pengujian Validasi Tambah Peminjaman Barang

Nomor Kasus Uji		VAL_55		
Nama Kasus Uji		Kasus Uji Validasi Lihat Tambah Peminjaman Barang		
No	Test Case	Input	Expected	Result Test
1	Aktor mengisi form tambah peminjaman barang secara lengkap dengan id peminjam menggunakan id mahasiswa yang terdaftar di database	Nim peminjam = 155150201111309 Tanggal mulai penggunaan = 30-05-2019, Tanggal selesai penggunaan = 02-06-2019, Jam mulai = 06.00, Jam selesai = 09.59, Lembaga = ALSA, Keterangan = acara pelatihan, Id ruangan = 28	Data peminjaman non kelas baru berhasil ditambahkan. Sistem menampilkan halaman tambah sarana peminjaman	valid
2	Aktor mengisi form tambah peminjaman barang secara lengkap dengan id peminjam menggunakan id mahasiswa yang tidak terdaftar di database	Nim peminjaman = 1551504949459, Tanggal mulai penggunaan = 30-05-2019, Tanggal selesai penggunaan = 02-06-2019, Jam mulai = 06.00, Jam selesai = 09.59,	Data peminjaman non kelas baru gagal ditambahkan. Sistem menampilkan halaman tambah peminjaman barang dengan menampilkan "data mahasiswa tidak ditemukan"	valid



		Lembaga = ALSA, Keterangan = acara pelatihan		
3	Aktor mengisi form tambah peminjaman barang dengan tidak lengkap	Nim peminjaman = kosong, Tanggal mulai penggunaan = kosong, Tanggal selesai penggunaan = kosong, Jam mulai = kosong, Jam selesai = kosong, Lembaga = kosong, Keterangan = kosong	Data peminjaman non kelas baru gagal ditambahkan. Sistem menampilkan halaman tambah peminjaman barang dengan pesan "isi isian ini"	valid

6.2.54 Pengujian Validasi Tambah Sarana Peminjaman

Hasil pengujian validasi tambah sarana peminjaman dapat dilihat pada Tabel 6.60 berikut.

Tabel 6.60 Pengujian Tambah Sarana Peminjaman

Nomor Kasus Uji	VAL_54			
Nama Kasus Uji	Kasus Uji Validasi Tambah Sarana Peminjaman			
No	Test Case	Input	Expected	Result Test
1	Aktor dapat menambahkan ruangan atau barang dalam proses peminjaman ruangan non	Id peminjaman = 1201905300001, id barang = 5	Barang yang di pilih tersimpan di database untuk menjadi sarana yang digunakan dalam peminjaman ber	valid



	kelas dan barang		id 1201905300001. Sistem menampilkan halaman tambah sarana peminjaman dengan menampilkan pesan “data sarana peminjaman berhasil ditambahkan”	
--	------------------	--	--	--

6.2.55 Pengujian Validasi Tinjau *Complaint*

Hasil pengujian validasi tinjau *complaint* dapat dilihat pada Tabel 6.61 berikut.

Tabel 6.61 Pengujian Validasi Tinjau *Complaint*

Nomor Kasus Uji	VAL_54			
Nama Kasus Uji	Kasus Uji Validasi Tinjau <i>Complaint</i>			
No	Test Case	Input	Expected	Result Test
1	Aktor dapat mengubah status <i>complaint</i> menjadi telah ditinjau	Id <i>complaint</i> = 1201905300001	Status <i>complaint</i> berubah menjadi ditinjau. Sistem menampilkan pesan “ <i>complaint</i> telah berhasil ditinjau:	valid



6.2.56 Pengujian Validasi Edit Password

Hasil pengujian validasi *edit password* dapat dilihat pada Tabel 6.62 berikut.

Tabel 6.62 Pengujian Validasi Edit Password

Nomor Uji	Kasus	VAL_55		
Nama Uji	Kasus	Kasus Uji Validasi <i>Edit Password</i>		
No	Test Case	Input	Expected	Result Test
1	Aktor mengisi form <i>edit password</i> dengan lengkap	Username = admin, Password lama = admin, Password baru = admins	Password berhasil diubah, sistem menampilkan pesan <i>password</i> telah berhasil diubah	valid
2	Aktor mengisi form <i>edit password</i> dengan tidak lengkap	Username = admin, Password lama = kosong, Password baru = kosong	Password gagal diubah, sistem menampilkan pesan "isi isian ini"	valid

6.2.57 Pengujian Validasi Import Data Excel

Hasil pengujian validasi *import data excel* dapat dilihat pada Tabel 6.63 berikut.

Tabel 6.63 Pengujian Validasi Import Data Excel

Nomor Uji	Kasus	VAL_56		
Nama Uji	Kasus	Kasus Uji Validasi <i>Import Data Excel</i>		
No	Test Case	Input	Expected	Result Test
1	Aktor memilih file yang akan diimport dengan menggunakan	File jadwalkuliah.xlsx	<i>Import</i> data jadwal kuliah berhasil dilakukan, sistem menampilkan	valid



	format yang benar		halaman jadwal kuliah	
2	Aktor memilih file yang akan diimport dengan menggunakan format yang salah	File jadwalkuliah.xlsx	Import data jadwal kuliah gagal dilakukan, sistem menampilkan halaman import jadwal kuliah	valid

6.2.58 Pengujian Validasi Lihat Grafik Rekap Data Peminjaman dan Complaint

Hasil pengujian validasi Lihat Grafik Rekap Data Peminjaman dan Complaint dapat dilihat pada Tabel 6.64 berikut.

Tabel 6.64 Pengujian Validasi Lihat Grafik Rekap Data Peminjaman dan Complaint

Nomor Kasus Uji	VAL_57			
Nama Kasus Uji	Kasus Uji Validasi Lihat Grafik Data Peminjaman dan Complaint			
No	Test Case	Input	Expected	Result Test
1	Aktor dapat melihat grafik data peminjaman dan complaint	Aktor memilih menu dashboard pada menu navigasi	Sistem menampilkan halaman dashboard yang berisi grafik data peminjaman dan complaint	valid

6.2.59 Pengujian Validasi Logout

Hasil pengujian validasi *logout* dapat dilihat pada Tabel 6.65 berikut.

Tabel 6.65 Pengujian Validasi Logout

Nomor Kasus Uji	VAL_58			
Nama Kasus Uji	Kasus Uji Validasi Logout			
No	Test Case	Input	Expected	Result Test



1	Aktor dapat logout dari sistem	Username =admin	Aktor menampilkan halaman login	valid
---	--------------------------------	-----------------	---------------------------------	-------

6.3 Pengujian Compatibility

Pengujian *compatibility* dilakukan untuk menguji kebutuhan non fungsional yang telah didefinisikan. Pengujian *compatibility* dilakukan untuk menguji sistem perangkat lunak agar dapat mampu berjalan dengan baik pada lingkungan yang berbeda-beda. Sistem perangkat lunak diharapkan mampu berjalan baik pada beberapa browser yang berbeda, seperti *internet explorer*, *microsoft edge*, *safari*, *opera*, *chrome*, *ios* dan *android*

Pengujian *compatibility* dilakukan dengan menggunakan *software* bantuan yang bernama *Sortsite*. *Sortsite* yang digunakan adalah versi *trial* 5.33.384.0. Untuk menguji *compatibility* sistem, *sortsite* menggunakan tiga kategori penilaian. Adapun kateori penilaian dapat dilihat pada Tabel 5.63 berikut.

Tabel 6.66 Kategori Masalah *Compatibility* Pada *Sortsite*

No	Simbol	Nama Kategori	Keterangan
1		<i>Critical issues</i>	menandakan adanya masalah fungsionalitas sistem atau adanya konten yang hilang pada browser tertentu
2		<i>Major issues</i>	Menandakan adanya masalah pada <i>layout</i> utama atau masalah performa browser
3		<i>Minor Issues</i>	Menandakan adanya masalah pada <i>layout</i> minor atau masalah performa browser

Gambar 6.4 menunjukkan hasil pengujian *compatibility* dengan menggunakan *sortsite*.



Errors	Accessibility	Compatibility	Search	Standards	Usability
--------	---------------	---------------	--------	-----------	-----------

This tab shows pages that exhibit browser-specific behavior, or trigger browser bugs.

Browser	IE		Edge	Firefox	Safari	Opera	Chrome	iOS			Android	
	Version	11	18	66	12	60	74	≤ 10	11	12	≤ 3	4*
Critical Issues	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Major Issues	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Minor Issues	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Key

- Missing content or functionality
- Major layout or performance problems
- Minor layout or performance problems

* Most Android devices from 4.4 onwards use Chrome as the default browser, older versions use the original Android stock browser

Gambar 6.4 Hasil Pengujian Compatibility

Berdasarkan Gambar 6.1 dapat dilihat bahwa hasil pengujian menunjukkan hasil yang baik. Dari ketiga kategori permasalahan yang ada, tidak ada satu *issues* yang menjadi masalah pada sistem. Hal ini dapat disimpulkan bahwa sistem mampu berjalan baik setiap browser yang diuji. Sistem mampu berjalan baik pada perangkat seperti *internet explorer, microsoft edge, safari, opera, chrome, dan android*



BAB 7 PENUTUP

7.1 Kesimpulan

Kesimpulan diambil berdasarkan hasil penelitian yang dilakukan dalam pembangunan sistem manajemen sarana prasarana di fakultas hukum Universitas Brawijaya. Kesimpulan dibagi kedalam tiga penjelasan yakni penjelasan kesimpulan terkait dengan hasil analisis dan perancangan, hasil implementasi dan hasil pengujian sistem manajemen sarana prasarana di fakultas hukum Universitas Brawijaya. Hasil analisis kebutuhan didapatkan dengan melakukan elisitasi kebutuhan menggunakan metode wawancara kepada pemangku kepentingan yang ada di fakultas hukum Universitas Brawijaya. Dari hasil analisis kebutuhan yang dilakukan maka didapatkan 58 kebutuhan fungsional dan 1 kebutuhan non fungsional. Sistem dibangun untuk dapat menangani proses peminjaman ruangan kelas, peminjaman ruangan non kelas, peminjaman barang, penyampaian informasi jadwal kuliah, penyampaian informasi status penggunaan ruangan, penyampaian informasi status penggunaan barang serta penyampaian keluhan. Sistem dioperasikan oleh lima aktor, yakni guest, sekretariat kuliah, staf pelayanan, validator, dan admin. Sedangkan proses perancangan menggunakan pendekatan *object oriented design* OOD, menggunakan empat jenis rancangan yakni *sequence diagram*, *class diagram*, perancangan komponen dan perancangan antar muka. Rancangan *sequence diagram* dan rancangan komponen dibuat dengan menggunakan tiga sampel dan semua sampel dapat dibuatkan rancangannya. Sedangkan hasil rancangan *class diagram* menghasilkan 14 *class*.

Implementasi sistem dilakukan dengan menggunakan pendekatan *object oriented programming* (OOP). Sistem diimplementasi dengan menggunakan *codeigniter* sebagai kerangka kerja *back-end* dan *bootstrap* sebagai kerangka kerja *front end*. Implementasi sistem menghasilkan jumlah kelas sebanyak 63 kelas. Dari hasil analisis kebutuhan dan perancangan yang telah dilakukan, semua kebutuhan dapat diimplementasi ke dalam sistem.

Pengujian sistem dilakukan dengan menggunakan tiga jenis pengujian, yakni pengujian unit, pengujian validasi dan pengujian *compatibility*. Pengujian unit dilakukan dengan menggunakan metode pengujian *white-box*. Hasil pengujian unit dilakukan dengan mengambil tiga sampel uji yang menunjukkan hasil 100% hasil pengujian adalah valid. Pengujian validasi dilakukan dengan menggunakan metode pengujian *black-box*. Pengujian validasi dilakukan pada semua fungsional sistem yang menunjukkan hasil 100 % valid. Sedangkan pengujian *compatibility* dilakukan untuk menguji non fungsional. Dari hasil pengujian *compatibility* yang dilakukan menunjukkan sistem sudah mampu dioperasikan dari lingkungan akses yang berbeda-beda. Dari ketiga hasil pengujian yang dilakukan, dapat disimpulkan bahwa sistem mendapatkan hasil pengujian 100% valid.



7.2 Saran

Saran yang dapat diberikan untuk pengembangan sistem adalah penambahan fitur-fitur tambahan yang mampu membuat sistem menjadi lebih baik. Fitur pertama yang dapat ditambahkan adalah fitur *whatsapp gateway*. Fitur ini nantinya akan membuat setiap peminjaman yang dilakukan akan lebih mudah dilacak. Setiap peminjaman yang disetujui dan ditolak akan diberitahu langsung kepada peminjam melalui pesan *whatsapp*. Fitur kedua yang disarankan adalah data mahasiswa, dosen sudah terintegrasi dengan sistem pusat, sehingga setiap data mahasiswa dan dosen akan selalu diperbaharui setiap ada perubahan yang dilakukan.



DAFTAR REFERENSI

- Annisa, F., 2017. Aplikasi Pengelolaan dan Peminjaman Ruangan di Fakultas Ilmu Terapan, Universitas Telkom. *e-Proceeding of Applied Science*, 3(2), 600-605.
- Azis, M.F., 2005. *Object Oriented Programming dengan PHP5*. [e-book]. Jakarta: PT Elex Media Komputindo. Tersedia melalui : Google Books <<https://books.google.co.id>> [Diakses 10 Februari 2019]
- Bahattab, Alshamrani, A., & Abdullah., 2015. A Comparison Between Three SDLC Models Waterfall Model, Spiral Model, and Incremental/Iterative Model. *IJCSI International Journal of Computer Science Issues*, 106-111.
- Balaji, S., & Murugaiyan, S., 2012. WATERFALLVs V-MODEL Vs AGILE: A COMPARATIVE STUDY ON SDLC. *International Journal of Information Technology and Business Management*, 26-30.
- Bootstrap, 2019. *Bootstrap*, [online] Tersedia di <<https://getbootstrap.com>> [Diakses 19 Januari 2019]
- Chandra, V., 2015. Comparison between Various Software Development Methodologies. *International Journal of Computer Applications*, 7-10.
- Chonoles, M.J., 2017. *OCUP 2 Certification Guide: Preparing for the OMG Certified UML 2.5 Professional 2 Foundation Exam*. [e-book]. Cambridge: Elsevier. Tersedia melalui : Science Direct <<https://www.sciencedirect.com>> [Diakses 10 Februari 2019]
- CodeIgniter, 2019. *CodeIgniter Rocks*, [online] Tersedia di <<https://codeigniter.com/>> [Diakses 19 Januari 2019]
- DataTables, D., 2019. *Add advanced interaction controls to your HTML tables the free & easy way*. Retrieved January 19, 2019, from <https://datatables.net/>
- Enterprise, J., 2016. *Pengenalan HTML dan CSS*. [e-book]. Jakarta: Elex Media Komputindo. Tersedia melalui : Google Books <<https://books.google.co.id>> [Diakses 10 Februari 2019]
- Fatta, H.A., 2007. *Analisis & Perancangan Sistem Informasi*. [e-book]. Yogyakarta: Andi. Tersedia melalui : Google Books <<https://books.google.co.id>> [Diakses 10 Februari 2019]
- Gustriansyah, R., & Suhandi, N., 2014. Sistem Informasi Manajemenn Ruang Kuliah Berbasis Piranti Bergerak. Makassar: KNSI2014-12.
- Hutahaean, J., 2014. *Konsep Sistem Informasi*. [e-book]. Yogyakarta: Deepublish. Tersedia melalui : Google Books <<https://books.google.co.id>> [Diakses 10 Februari 2019]



Khairina, D. M., Maharani, S., & Hatta, H. R., 2018. Sistem Informasi Manajemen Ruang (SIMERU) Kelas (Studi Kasus: FKTI Universitas Mulawarman). *Jurnal Ilmiah Ilmu Komputer*, 30-32.

James, R., Jacobson, I., & Booch, G., 1998. *The Unified Modelling Language Reference Manual*. [e-book]. Canada: Addison Wesley Longman. Tersedia melalui : Google Books <<https://people.ucalgary.ca>> [Diakses 10 Februari 2019]

Jquery, 2019 . *What is jQuery*, [online] Tersedia di <<https://jquery.com>> [Diakses 22 Maret 2019]

Komputer, W., 2010. *Panduan Belajar MySQL Database Serve*. [e-book]. Jakarta: Mediakita. Tersedia melalui : Google Books <<https://books.google.co.id>> [Diakses 10 Februari 2019]

Lonetti, F., & Marchetti, E., 2017. Emerging Software Testing Technologies. [e-book]. Pisa: Academic Press. Tersedia melalui : Science Direct <<https://www.sciencedirect.com>> [Diakses 10 Februari 2019]

Nugraha, M., & Nugraha, Y., 2018. Selamat Tinggal Bentrok! Efisiensi Pengelolaan Jadwal Kuliah dengan SIM-DRaW (Sistem Informasi Manajemen Dosen, Ruang, dan Waktu). *Fountain of Informatics Journal*, 24-29.

Nurwarsito, H., 2009. Sistem Informasi Jadwal Perkuliahan dengan. *Jurnal EECIS*, 57-61.

Pressman, R. S., 2010. *Software Engineering A Practitioner's Approach*. 7th ed. [e-book]. New York: McGraw-Hill. Universitas Dian Nuswantoro <<https://dinus.ac.id>> [Diakses 12 Februari 2019]

Raharjo, B., 2016. *Modul Pemrograman Web (HTML, PHP, & MYSQL)*. 7th ed. Bandung: Modula.

Rosa, A.S., & Shalahuddin, M., 2015. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika Bandung.

Sianipar, R., 2015. *Pemrograman Javascript*. [e-book]. Bandung: Informatika. Tersedia melalui : Google Books <<https://books.google.co.id>> [Diakses 10 Februari 2019]

Sommerville, I., 2011. *Software Engineering*. 9th ed. [e-book]. Boston: Addison-Wesley. Universitas Dian Nuswantoro <<https://dinus.ac.id>> [Diakses 12 Februari 2019]

Sutabri, T., 2012. *Konsep Sistem Informasi*. [e-book]. Yogyakarta: Andi. Tersedia melalui : Google Books <<https://books.google.co.id>> [Diakses 11 Februari 2019]