

Pengembangan Sistem Manajemen Peminjaman Sarana dan Prasarana (Studi Kasus : Fakultas Hukum Universitas Brawijaya)

Sueddi Sihotang¹, Achmad Arwan², Fitra Abdurrachman Bachtiar³

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹simarsoitsueddi@gmail.com, ²arwan@ub.ac.id, ³fitra.bachtiar@ub.ac.id

Abstrak

Fakultas Hukum Universitas Brawijaya melaksanakan kegiatan perkuliahan setiap harinya namun tidak mempunyai media untuk menyampaikan informasi jadwal kuliah secara *online*. Fakultas Hukum Universitas Brawijaya juga mempunyai kegiatan diluar perkuliahan yang menggunakan ruangan dan peralatan penunjang agar kegiatan dapat berjalan baik. Kebutuhan mahasiswa dalam menggunakan ruangan dan peralatan fakultas sudah terpenuhi selama ini. Mahasiswa dapat melakukan peminjaman ruangan dan barang kepada fakultas yang selama ini dilakukan secara manual. Setiap peminjaman harus divalidasi oleh kasubag akademik, kasubag kemahasiswaan dan kasubag umum. Proses validasi ini terkadang membutuhkan waktu yang lama jika orang-orang tersebut tidak dapat ditemui dalam waktu tertentu. Sistem manajemen peminjaman sarana dan prasarana adalah sebuah sistem yang dapat mengatasi permasalahan diatas. Pengajuan peminjaman dan validasi peminjaman dapat dilakukan secara *online*. Setiap peminjaman akan direkap oleh sistem. Sistem dikembangkan dengan metode *waterfall* dan menggunakan pendekatan berorientasi objek. Pengembang menggunakan *codeigniter* sebagai kerangka kerja untuk melakukan implementasi sistem karena mendukung konsep MVC (*model, view, controller*). Pengujian sistem dilakukan dengan menggunakan *black-box testing* dan *white-box testing* untuk menguji fungsional sistem serta menggunakan *compatibility testing* untuk menguji non fungsional sistem. Berdasarkan pengujian yang dilakukan, sistem mendapat hasil pengujian fungsioanl 100% valid dan mendapat hasil yang baik dalam pengujian *compatibility* karena mampu berjalan baik pada lingkungan browser yang diuji.

Kata kunci: *peminjaman, ruangan, barang, validasi, rekap*

Abstract

Faculty of Law Universitas Brawijaya conducts lecture activities every day but does not have the media to submit lecture information online. The Faculty of Law, Universitas Brawijaya also has activities outside of lectures that use space and supporting equipment so that the activity can run well. The needs of students in using faculty rooms and equipment have been met so far. Students can borrow rooms and goods to the faculty which have been done manually. Every loan must be validated by the head of the academic, the head of the student body and the head of the general. This validation process sometimes takes a long time if these people cannot be found within a certain time. The loan facility and infrastructure management system is a system that can overcome the above problems. Submitting a loan and validating a loan can be done online. Every loan will be recapitulated by the system. The system was developed by the waterfall method and uses an object oriented approach. Developers use codeigniter as a framework for implementing systems because it supports the concept of MVC (model, view, controller). System testing is done by using black-box testing and white-box testing to test the functional system and using compatibility testing to test non-functional systems. Based on the tests conducted, the system gets 100% valid functional test results and gets good results in compatibility testing because it is able to run well in the browser environment under test.

Keywords: *loan, room, goods, validation, recap*

1. PENDAHULUAN

Fakultas Hukum Universitas Brawijaya (FH UB) adalah salah satu fakultas tertua di Universitas Brawijaya yang telah berdiri sejak tahun 1957. FH UB pada awalnya memiliki nama Perguruan Tinggi Hukum dan Pengetahuan Masyarakat (PTHMP). FH UB saat ini memiliki lima program studi yakni program sarjana, program magister kenotariatan, program magister ilmu hukum, 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 teknologi informasi yang sangat cepat mengakibatkan munculnya inovasi yang baru dalam pengolahan dan penyampaian informasi agar dapat tersampaikan dengan 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, hal ini disebabkan karena sulit melakukan pencarian sebab jadwal harus dicek berdasarkan baris dan kolom. Apalagi jumlah kolom dan baris yang ada di jadwal kuliah berjumlah ratusan baris yang membuat sulitnya pencarian data.

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, penggunaan komputer sangatlah membantu seseorang dalam menyelesaikan sebuah permasalahan atau kegiatan yang awalnya harus dilakukan 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.

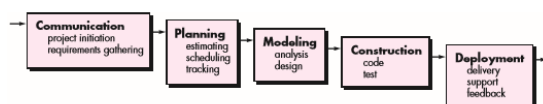
Kehadiran sebuah sistem yang mampu melakukan manajemen peminjaman sarana dan prasarana di FH UB akan sangat berguna. Sistem yang mampu menyajikan informasi jadwal kuliah yang lebih baik. Mampu menyajikan informasi jadwal kuliah yang sudah dipetakan sehingga dapat dilihat lebih lebih cepat dengan tersedianya 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 diatas 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.

2. METODE

2.1 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, dapat dilihat pada Gambar 1.



Gambar 1. *Waterfall Model*

(Sumber : Pressman, 2010)

Pada gambar 1, *Waterfall model* mempunyai

lima tahapan, yakni

1. *Communication*

Pada tahapan ini pengembang aplikasi dan kostumer membuat kesepakatan tentang seperti apa sistem akan dibangun, fitur-fitur apa saja yang harus dilengkapi dan penjelasan terkait spesifikasi sistem.

2. *Planning*

Pada tahap ini dilakukan penyusunan rencana pengerjaan dan perkiraan estimasi pengerjaan sistem dimulai dari tahap *modeling* sampai *deployment*.

3. *Modelling*

Pada tahapan ini pengembang aplikasi melakukan perancangan terkait solusi dari terkait dengan desain algoritma yang sesuai, desain arsitektur perangkat lunak, *database* sistem.

4. *Construction*

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

5. *Deployment*

Pada *deployment* pengembang aplikasi melakukan pengujian sistem. Sistem harus dipastikan agar mampu berjalan dengan baik saat nanti digunakan.

Waterfall model dipilih dalam penelitian karena *waterfall model* adalah metode yang mudah diimplementasikan karena mempunyai model linier. Model linier yang dimiliki *waterfall* membuatnya mudah untuk 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).

2.2 Pendekatan Berorientasi Objek

Pendekatan berorientasi objek terdiri dari tiga bagian yaitu *Object Oriented Analysis* (OOA), *Object Oriented Design* (OOD), *Object Oriented Programming* (OOP). *Object Oriented*

Analisis (OOA) metode analisis kebutuhan yang menggunakan pendekatan berorientasi objek. Analisis kebutuhan yang menggunakan OOA sebaiknya dilakukan oleh orang yang sudah paham tentang objek, hal ini dikarenakan setiap kebutuhan harus dapat dipastikan untuk dapat diimplementasikan dengan berbasis objek. *Object Oriented Design* (OOD) adalah tahapan yang dilakukan untuk membuat model desain rancangan menggunakan pendekatan berorientasi objek (Rosa & Shalahuddin). *Object Oriented Programming* (OOP) adalah metode pengembangan aplikasi dengan melihatnya sebagai sekumpulan objek yang saling terhubung dan melakukan interaksi. Aplikasi dibangun dengan membagi fungsi-fungsi berdasarkan tugas tanggung jawab masing masing (Azis, 2005).

2.3 Pengujian Perangkat Lunak

Pengujian perangkat lunak merupakan tahapan untuk memastikan aplikasi sudah dapat berjalan dengan tanpa mendapat kendala atau *error*. Pengujian ini juga dilakukan agar aplikasi yang dibangun terhindar dari catat atau kesalahan error saat sudah dijalankan pengguna. Pengujian dilakukan dengan menggunakan tiga metode pengujian seperti *white-box testing*, *black-box testing*, dan *compatibility testing* (Sommerville, 2011).

2.3.1 White-Box Testing

White-box testing adalah sebuah metode pengujian berdasarkan penulisan kerangka kerja dari kode program yang ditulis. Serangkaian tes harus dilakukan untuk memastikan bahwa program dapat melewati setiap jalur logis program saat dijalankan. Untuk melakukan pengujian ini, penguji dapat melihat struktur kode program terlebih dahulu, setelah itu penguji dapat mengetahui *flow-graph* dari aliran program yang diuji. Flow-graph dapat memberitahu penguji jumlah jalur independent dari kode program yang diuji. Penguji kemudian dapat memasukkan nilai input yang akan menyebabkan setiap jalur dapat dilewati (Sommerville, 2011).

2.4.2 Black-Box Testing

Black-box testing adalah sebuah metode pengujian yang memperhatikan spesifikasi *requirement* dari sistem. Pengujian ini dilakukan melihat kembali spesifikasi kebutuhan yang telah disepakati dan memastikan setiap

kebutuhan sudah diimplementasi dan mampu dijalankan dengan baik. Penguji dapat langsung memasukkan nilai input dan melihat bagaimana hasil keluarannya, berdasarkan hasil keluaran penguji menyimpulkan valid atau tidaknya kasus uji yang dilakukan (Sommerville, 2011).

2.4.3 Compatibility Testing

Compatibility testing adalah pengujian untuk kebutuhan non fungsional sistem. Hal ini berkaitan dengan bagaimana sebuah sistem agar mampu berjalan pada lingkungan akses yang berbeda-beda (Lonetti & Marchetti, 2017). *Sortsite* adalah aplikasi yang digunakan untuk pengujian ini. Aplikasi ini mampu memberikan informasi apakah sistem sudah mampu dijalankan dengan baik saat diakses pada jenis perangkat yang berbeda-beda atau tidak dengan memberikan status valid dan *issue* jika terdapat masalah.

3. METODOLOGI PENELITIAN

Metodologi penelitian adalah urutan tahapan yang dilakukan selama penelitian. Tahapan yang dilakukan pada metodologi penelitian ini dapat dilihat pada Gambar 2.



Gambar 2. Diagram Alir Metodologi Penelitian

Tahapan pertama yang dilakukan adalah studi literatur yang dilakukan dengan mengumpulkan informasi-informasi terkait dengan manajemen peminjaman sarana dan prasarana. Informasi dapat diperoleh dari berbagai sumber terpercaya seperti jurnal, laporan penelitian terdahulu, buku, artikel *online*, dan laporan penelitian. Penelitian

terlebih dahulu dapat digunakan penulis sebagai pedoman acuan dalam penelitian. Peneliti dapat mengambil informasi terkait kelebihan dan kekurangan dari penelitian terdahulu.

Tahapan kedua yang dilakukan adalah analisis kebutuhan untuk mengetahui bagaimana dan seperti apa gambaran sistem manajemen peminjaman sarana dan prasarana. Penulis wawancarai setiap pihak yang terlibat dalam kegiatan manajemen dan peminjaman ruangan dan barang. Penulis melakukan analisis kebutuhan dalam beberapa tahapan, yakni tahapan proses bisnis, elisistasi kebutuhan dan pemodelan kebutuhan.

Tahapan ketiga yang dilakukan adalah perancangan dengan menggunakan perancangan arsitektur dan perancangan komponen. Perancangan arsitektur dilakukan dengan memperhatikan kembali hasil pemodelan kebutuhan yang telah diperoleh dan kemudian dapat dimodelkan kedalam sebuah diagram. Sedangkan perancangan komponen dilakukan dengan menggunakan perancangan algoritma dan perancangan antarmuka. Perancangan algoritma dilakukan dengan membuat rancangan urutan logika yang diterapkan di sistem. Sedangkan perancangan antarmuka, penulis membuat rancangan tampilan halaman sistem. Rancangan tersebut berupa gambar yang memuat bentuk formulir, halaman, tabel, tombol, ikon.

Tahapan keempat yang dilakukan adalah implementasi. Implementasi dilakukan mengubah hasil rancangan menjadi sebuah sistem menggunakan bahasa pemrograman. Implementasi dimulai dengan membangun antarmuka atau *front end sistem* kemudian dilanjut dengan pemberian fungsi pada sistem. Sistem diimplementasikan menggunakan bahasa pemrograman PHP, HTML, CSS, JS dan MySql untuk untuk menjadi *database* sistem. Untuk memudahkan implementasi maka penulis juga menggunakan *framework bootstrap* untuk implementasi tampilan sistem dan *codeignitier* untuk *back end*.

Tahapan kelima yang dilakukan adalah pengujian sistem. Pengujian dilakukan untuk memastikan apakah sistem perangkat lunak yang dikembangkan sudah memenuhi semua persyaratan atau tidak. Dalam penelitian ini penulis menggunakan tiga teknik pengujian, *white-box testing*, *black-box-testing* dan *compatibility-testing*.

4. ANALISIS KEBUTUHAN

4.1. Gambaran Umum Sistem

Sistem manajemen peminjaman sarana dan prasarana melingkupi pekerjaan sebagai berikut.

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

4.2 Identifikasi Aktor

Aktor merupakan orang maupun sistem yang ikut ambil bagian dalam menjalankan sebuah proses bisnis. Aktor yang terlibat dalam sistem manajemen sarana prasarana ini terdiri dari lima aktor yang memiliki peran dari masing-masing aktor. Aktor yang terlibat dapat dilihat pada Tabel 1.

Tabel 1. Identifikasi Aktor

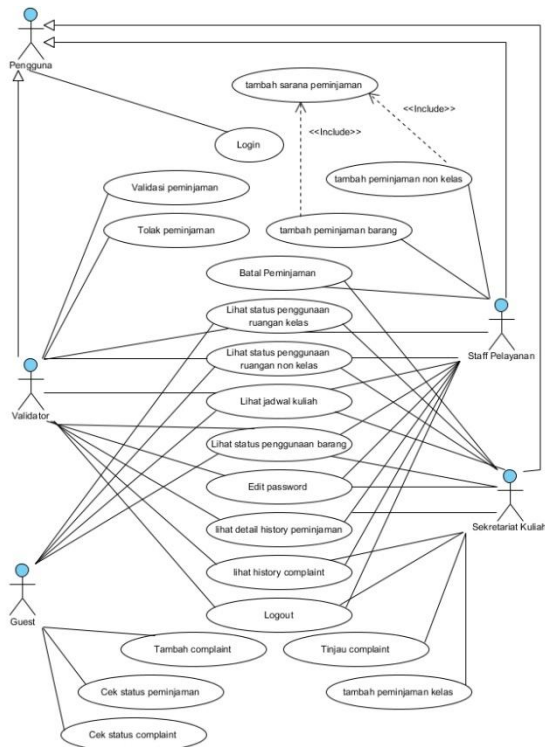
Aktor	Deskripsi
Guest	Guest adalah aktor untuk mahasiswa atau dosen. Yang melihat informasi di sistem
Sekretariat Perkuliahan	Aktor yang melakukan peminjaman ruangan kelas
Staf Pelayanan	Aktor yang melakukan penambahan peminjaman ruangan untuk kegiatan diluar perkuliahan dan peminjaman barang
Validator	Aktor yang melakukan validasi peminjaman, terdiri dari tiga jenis, kasubag akademik, kasubag kemahasiswaan, kasubag umum
Admin	Aktor yang mengatur sistem secara menyeluruh.

4.3. Kebutuhan Sistem

Kebutuhan sistem daftar kebutuhan yang harus dipenuhi oleh sistem. Daftar kebutuhan sistem terdiri dari kebutuhan fungsional dan kebutuhan non fungsional. Kebutuhan fungsional merupakan kemampuan atau layanan yang harus dimiliki oleh sistem sehingga sistem mampu berjalan dengan baik. Sistem manajemen peminjaman sarana dan prasarana mempunyai 59 kebutuhan fungsional dan 1 kebutuhan non fungsional.

4.4. Use Case Diagram

Use case diagram adalah diagram yang digunakan dalam memodelkan aktor yang terlibat dan apa yang saja yang dapat dilakukan. Use case diagram sistem manajemen peminjaman sarana dan prasarana digambarkan pada Gambar 3. Pada Gambar 3 dapat dilihat apa saja yang dapat dilakukan oleh aktor *guest*, sekretariat kuliah, staf pelayanan, dan validator.



Gambar 2. Use Case Diagram Guest, Staf Pelayanan, Sekretaris Kuliah, dan Validator

5. PERANCANGAN DAN IMPLEMENTASI

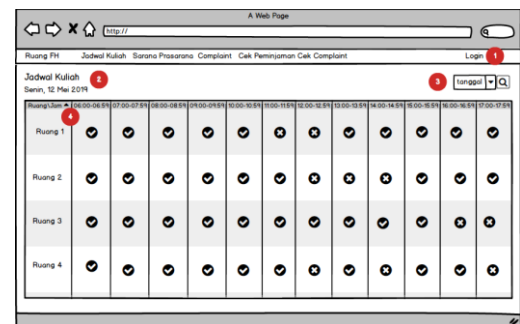
5.1. Class Diagram

Class diagram sistem manajemen peminjaman ruangan dan sarana prasarana dibuat dengan mengikuti konsep MVC dari *framework codeigniter*. Class diagram dirancang dengan membagi class ke dalam *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 peminjaman sarana dan prasarana yang dirancang mempunyai 14 *class*.

7 *class* diantaranya adalah *class controller* dan 7 lagi adalah *class model*. *Class controller* terdiri dari *class* peminjaman, auth, user, sarana_prasarana, jadwal_kuliah, complaint dan rekap. Sedangkan *class model* terdiri dari *class* M_peminjaman, M_sarana_prasarana, M_auth, M_user, M_jadwal_kuliah, M_complaint dan M_Rekap.

5.2 Perancangan Antarmuka

Perancangan antar muka adalah perancangan yang dilakukan untuk membuat rancangan halaman antarmuka tampilan sistem yang dibangun. Tampilan harus dirancang terlebih dahulu sebelum diimplementasikan dengan menggunakan bahasa pemrograman. Salah satu rancangan antarmuka dapat dilihat pada Gambar 3 dan penjelasan dari Gambar 3 dapat dilihat pada Tabel 2.



Gambar 3. Rancangan Antarmuka Halaman Peta Ruang Non Kelas

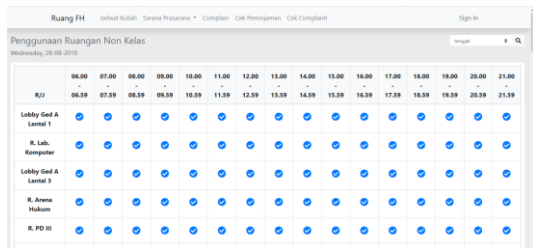
Tabel 2. Penjelasan Rancangan Antarmuka Halaman Peta Ruang Non Kelas

No	Nama Objek	Keterangan
1	Navbar	Pilihan menu pada sistem Ruang FH
2	Judul Halaman	Judul halaman yang sedang diakses
3	Tombol Filter	Tombol untuk melakukan filter jadwal kuliah
4	Tabel Jadwal Kuliah	Tabel yang berisi informasi status penggunaan ruangan

3.2. Implementasi Antarmuka

Implementasi antarmuka adalah hasil implementasi dari hasil rancangan tampilan halaman antarmuka sistem. Antarmuka atau tampilan sistem diimplementasi dengan menggunakan *framework bootstrap*. Gambar 4

menunjukkan implementasi dari antarmuka halaman peta ruangan non kelas.



Gambar 4. Impementasi Antarmuka Halaman Peta Ruangnon Kelas

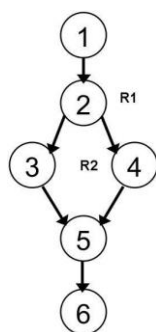
6. PENGUJIAN

6.1 Pengujian White-Box

Pengujian *white-box* dilakukan untuk melakukan menguji alur program sistem. Pengujian dilakukan pada *method* sampel tambahJadwalKuliah().

Tabel 3. *Pseudocode method* tambahJadwalKuliah()

Mulai	1
Get data input_jadwal_kuliah[]	2
if cek_jadwal_kuliah() = true	
Output = jadwal kuliah gagal ditambahkan	3
Redirect halaman tambah jadwal kuliah	
else	4
data = data_input_jadwal_kuliah[]	5
tambahJadwalKuliah(data)	
Output = jadwal kuliah berhasil ditambahkan	
Redirect halaman jadwal kuliah	
end If	6
Selesai	7



Gambar 5. Flow Graph Method tambahJadwalKuliah()

Cyclomatic Complexity

- $V(G) = R = 2$
- $V(G) = E - N + 2 = 6 - 6 + 2 = 2$
- $V(G) = P + 1 = 1 + 1 = 2$

Independent Path

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

- Jalur 2 = 1-2-4-5-6

6.2 Pengujian Black-Box

Pengujian *Black-Box* dilakukan untuk memastikan semua *requirement* telah diimplementasi dengan baik. Hasil pengujian ini mendapatkan hasil pengujian 100% valid. Berdasarkan hal tersebut maka sistem manajemen peminjaman sarana dan prasarana telah mampu dijalankan dengan baik sesuai dengan hasil yang diharapkan.

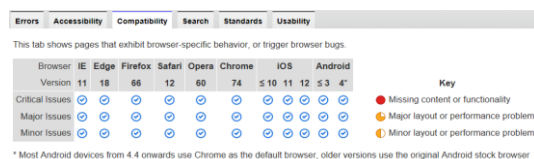
6.3 Pengujian Compatibility

Pengujian *compatibility* dilakukan untuk menguji perangkat lunak apakah mampu berjalan pada lingkungan akses yang berbeda-beda atau tidak. Pengujian *compatibility* dilakukan dengan menggunakan aplikasi Sortsite. Sortsite yang digunakan adalah versi trial 5.33.384.0. Untuk menguji *compatibility*, sortsite menggunakan tiga kategori penilaian yang dapat dilihat pada Tabel 4.

Tabel 4. Kategori Penilaian *Compatibility*

Simbol	Nama Kategori	Keterangan
	Critical issues	Terdapat masalah pada fungsional sistem, kemungkinan disebabkan karena hilangnya konten pada browser tertentu
	Major issues	Terdapat masalah performa browser seperti masalah mayor pada <i>layout</i> sistem
	Minor issues	Terdapat masalah performa browser seperti masalah minor pada <i>layout</i> sistem

Gambar 5 menunjukkan hasil pengujian yang dilakukan dengan mendapat hasil yang baik. Dari ketiga kategori permasalahan yang ada, tidak ada satu *issues* yang menjadi masalah pada sistem. Hal ini kemudian dapat disimpulkan bahwasanya sistem sudah mampu berjalan baik jika diakses pada lingkungan yang berbeda-beda. Hasil pengujian menunjukkan, sxistem mampu berjalan baik pada perangkat seperti *internet explorer*, *microsoft edge*, *safari*, *opera*, *chrome*, dan *android*.



Gambar 5. Hasil Pengujian *Compatibility*

7. KESIMPULAN DAN SARAN

7.1 Kesimpulan

Analisis kebutuhan mendapatkan 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. Perancangan menggunakan pendekatan *object oriented design* OOD, perancangan yang telah dilakukan mendapatkan hasil rancangan arsitektur dan rancangan komponen. Rancangan komponen menghasilkan *sequence* diagram dan *class* diagram. Rancangan komponen dibuat dengan menggunakan tiga sampel dan semua sampel dapat dibuatkan rancangannya. Sedangkan hasil rancangan class diagram menghasilkan 14 *class*.

Sistem diimplementasikan 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.

Pengujian dilakukan pada sistem dengan menggunakan metode pengujian *white-box*, pengujian *black-box* dan pengujian *compatibility*. Hasil pengujian *white-box* mengambil tiga sampel uji sebagai kasus uji dan mendapatkan hasil 100% valid. Pengujian *black-box* dilakukan pada semua fungsional yang telah ditetapkan dan menunjukkan hasil 100 % valid. Sedangkan pengujian *compatibility* yang telah dilakukan menunjukkan sistem sudah mampu dioperasikan dari lingkungan akses yang berbeda-beda.

7.2 Saran

Saran yang dapat diberikan oleh penulis setelah melakukan penelitian ini 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.

8. DAFTAR PUSTAKA

- Azis, M. F. 2005. Object Oriented Programming dengan PHP 5. PT Elex Media Komputindo, Jakarta.
- Balaji, S., & Murugaiyan, S. 2012. Waterfall vs V-Model vs Agile: A Comparative Study On SDLC. *International Journal of Computer Applications*, 2(1), 26-30.
- Chandra, V. 2015. Comparison between Various Software Development Methodologies. *International Journal of Computer Applications*, 131(9), 7-10.
- Lonetti, F., & Marchetti, E. 2017. Emerging Software Testing Technologies. Academic Press, Pisa.
- Nugraha, M., & Nugraha, Y. 2012. Selamat Tinggal Bentro! Efisiensi Pengelolaan Jadwal Kuliah dengan SIM-DraW (Sistem Informasi Manajemen Dosen, Ruang dan Waktu). *Foundation of Informatics Journal*, 3(1), 24-29.
- Nurwasinto, H. 2009. Sistem Informasi Jadwal Perkuliahan dengan Metode Sistem Pakar. *Jurnal EECCIS*, 3(1), 57-61.
- Pressman, R. S. 2010. *Software Engineering A Practitioner's Approach*. 7th ed. McGraw-Hill, New York.
- Rosa, A. S., & Shalahuddin, M. 2015. Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek. Informatika Bandung, Bandung.
- Sommerville, I. 2011. *Software Engineering*. 9th ed. Addison-Wesley, Boston.