

APLIKASI PEMINJAMAN RUANG KELAS UNTUK KULIAH PENGANTI JADWAL REGULER (STUDI KASUS: FAKULTAS ILMU TERAPAN)

Berliana Putri Meliani¹, Ir. Ely Rosely, M.B.S.², Elis Hernawati, S.T., M.Kom.³ Program
Studi D3 Sistem Informasi, Fakultas Ilmu Terapan, Universitas Telkom
berlianaputri@student.telkomuniversity.ac.id,
ely.rosely@tass.telkomuniversity.ac.id, elishernawati@tass.telkomuniversity.ac.id

Fakultas Ilmu Terapan merupakan salah satu Fakultas di Universitas Telkom yang mempunyai unit logistik dan Layanan Akademik yang salah satu tugasnya adalah mengelola peminjaman ruangan baik peminjaman ruangan akademik maupun non akademik. Peminjaman ruangan yang dilakukan saat ini masih melakukan proses manual yaitu peminjaman harus menemui langsung pihak pengelola untuk mendapatkan persetujuan peminjaman ruangan yang dibutuhkan.

Aplikasi yang di bangun menggunakan SDLC Model Waterfall sebagai metode pengembangan, yaitu menganalisis kebutuhan pengguna terhadap sistem yang akan di bangun dan dalam pengumpulan data dilakukan 2 hal yaitu observasi dan wawancara, desain sistem yang akan dibuat sesuai dengan kebutuhan pengguna dengan bantuan tools starUML, mengimplementasikan desain dalam bentuk penulisan kode program dengan menggunakan bahasa pemrograman PHP dan memanfaatkan database MySQL untuk penyimpanan datanya, pengujian program dilakukan menggunakan black box texting guna untuk memastikan bahwa program telah sesuai dengan desainnya dan semua fungsi dapat dipergunakan dengan baik.

Kata kunci: Fakultas Ilmu Terapan, Peminjaman ruangan, SDLC Waterfall, PHP, MySQL

School of Applied Science is one of the faculty at the University of Telkom, which has a logistics unit and Academic Services whose job is room management and booking room academic and non-academic. Booking room today still do a manual process that is direct room must meet the manager to get approvals required room. The other hand when you want to see the manager related to non-academic room booking should meet directly related parties are sometimes unable to attend or are

not in place.

Applications that are built using SDLC Waterfall Model as a method of development, namely analysis needs of users of the system to be built, the design of the system being designed according to user requirements, implement the design in the form of writing code program using the programming language PHP and utilizes a MySQL database for data storage, the testing program that was built to ensure that the program complies with the design and all functions can be used properly.

Keywords: Faculty of Applied Science, room booking, SDLC Waterfall, PHP, MySQL

I. PENDAHULUAN

Universitas Telkom merupakan salah satu perguruan tinggi swasta yang mempunyai visi menjadi *world class* university. Mahasiswa di Universitas Telkom dituntut untuk aktif dalam kegiatan akademik maupun non akademik. Jadwal perkuliahan yang padat dan sewaktu-waktu dapat berubah karena beberapa hal mengharuskan mahasiswanya untuk mencari jadwal pengganti agar tetap mendapatkan materi yang seharusnya didapatkan pada perkuliahan yang seharusnya. Unit yang melayani peminjaman ruangan untuk kelas pengganti adalah Layanan Akademik (LA).

Berdasarkan hasil observasi dan wawancara, peminjaman ruang kelas pengganti di LA masih dilakukan secara manual dengan bantuan Ms. Excel dan Mahasiswa mengisi secara manual di lembar kertas yang disediakan. Sehingga mahasiswa atau dosen harus datang langsung ke LA dan meluangkan waktu untuk meminjam ruangan untuk kelas pengganti. Selain itu, jadwal dosen dan mahasiswa yang padat menjadi salah satu hambatan dalam peminjaman

ruangan untuk kelas pengganti.

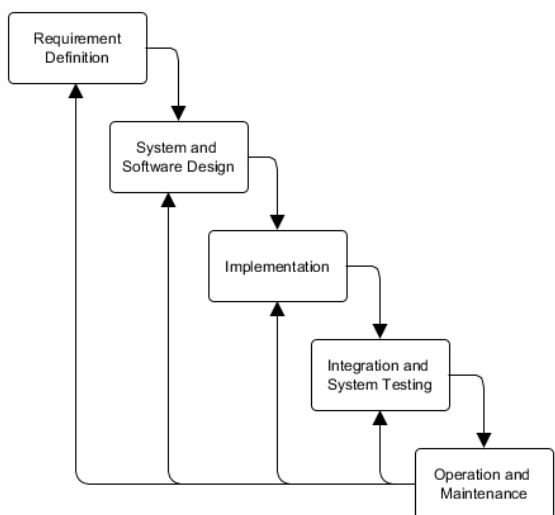
Oleh karena itu dibuatlah aplikasi dengan judul “Peminjaman Ruang Kelas Pengganti di Fakultas Ilmu Terapan” yang berbasis web. Adanya aplikasi ini diharapkan dapat memudahkan mahasiswa, dosen maupun LA dalam peminjaman ruangan untuk kelas pengganti. Dengan adanya aplikasi ini, peminjaman ruangan untuk kelas pengganti dapat dilakukan secara *online* dan terkomputerisasi. Sehingga tidak membutuhkan banyak waktu dan tidak perlu datang langsung ke bagian LA.

II. TINJAUAN PUSTAKA

Berikut tinjauan pustaka yang menjadi referensi pada pengembangan proyek akhir ini.

A. Metode Pengerjaan Aplikasi

Metode pengerjaan pada Aplikasi Peminjaman Ruang Kelas Untuk Kuliah Pengganti (Studi Kasus: Fakultas Ilmu Terapan) menggunakan SDLC *Waterfall Model*. Model *Waterfall* ini berisi rangkaian aktivitas proses seperti analisis kebutuhan, perancangan *design*, pengkodean, dan pengujian-pengujian. Tahapan-tahapan dari metode *waterfall* adalah sebagai berikut :



Gambar 2-1 SDLC Waterfall [1]

1. Requirement Definition

Tahap ini pengembang sistem diperlukan komunikasi yang bertujuan untuk memahami perangkat lunak yang diharapkan oleh pengguna dan batasan perangkat lunak tersebut. Pada tahap ini, dilakukan wawancara dan observasi untuk

mengetahui permasalahan dari peminjaman ruang kelas pengganti untuk menentukan kebutuhan aplikasi yang akan dibangun.

2. System and Software Design

Spesifikasi kebutuhan dari tahap sebelumnya akan dipelajari dalam fase ini dan desain sistem disiapkan. Pada tahap ini dibuat proses bisnis, rancangan *database*, tampilan *User Interface*, berdasarkan hasil observasi dan wawancara.

3. Implementation

Pada tahap ini, sistem pertama kali dikembangkan di program kecil yang disebut *unit*, yang terintegrasi dalam tahap selanjutnya. Pada tahap ini, *tools* yang digunakan adalah Visual Studio Code dengan menggunakan bahasa pemrograman PHP dan *framework* CodeIgniter.

4. Integration & System Testing

Seluruh *unit* yang dikembangkan dalam tahap implementasi diintegrasikan ke dalam sistem setelah pengujian yang dilakukan masing-masing *unit*. Pada tahap ini, dilakukan pengujian secara *Black Box Testing* dan *White Box Testing*.

5. Operation & Maintenance

Operation & Maintenance merupakan tahap akhir dalam model *waterfall*. Dalam pengembangan aplikasi peminjaman ruang kelas pengganti, tahapan ini tidak dilakukan.

B. Perancangan

1. Business Process Modelling Notation (BPMN)

BPMN merupakan teknik yang memungkinkan semua pihak yang terlibat dalam proses berkomunikasi secara jelas, benar dan efisien. Dengan cara ini BPMN mendefinisikan notasi dan *semantic Diagram* Proses Bisnis (BPD). BPD adalah diagram berdasarkan teknik *flowchart*, yang dirancang untuk menyajikan urutan grafis dari semua kegiatan yang terjadi selama proses berlangsung. BPMN menyediakan kemampuan memahami prosedur internal bisnis dalam notasi grafis [2].

2. Use Case Diagram

Use Case atau diagram *Use Case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use Case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu [1].

Syarat penamaan pada *Use Case* adalah nama di definisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *Use Case* untuk pendefinisian apa yang disebut aktor dan *Use Case* [1] yaitu :

- a. Aktor merupakan orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
- b. *Use Case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

3. ER Diagram

Pemodelan awal basis data yang paling banyak digunakan adalah menggunakan *Entity-Relationship Diagram* (ERD). ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional. Sehingga jika penyimpanan basis data menggunakan OODBMS maka perancangan basis data tidak perlu menggunakan ERD. ERD memiliki beberapa aliran notasi seperti notasi Chen (dikembangkan oleh Peter Chen), Barker (dikembangkan oleh Richard Barker, Ian Palmer, Harry Ellis), notasi *Crow's Foot*, dan beberapa notasi lain. Namun yang banyak digunakan adalah notasi dari Chen [1].

Entity Relationship Diagram (ERD) merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi [1].

4. Class Diagram

Class diagram, merupakan diagram yang digunakan untuk merepresentasikan kelas, komponen-komponen kelas dan hubungan antara masing-masing kelas. Selain itu, *class diagram* mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat diantara mereka. *Class diagram* juga menunjukkan *property* dan operasi sebuah kelas serta batasan-

batasan yang terdapat dalam hubungan-hubungan objek tersebut [3].

C. Pemrograman

1. *Hyper Markup Language* (HTML)

Hyper Markup Language yaitu sebuah bahasa *scripting* yang dapat menghasilkan halaman *website* pada dasarnya HTML merupakan bahasa *script* dasar yang berjalan bersama berbagi bahasa pemrograman lainnya. Kode HTML tidak dapat di jadikan sebagai *file executabel program*, karena HTML hanyalah sebuah bahasa *Scripting* yang dapat berjalan apabila di jalankan di dalam browser seperti Internet Explorer, Netscape Navigator, Opera, Mozilla dan lain-lain [4].

2. *Cascading Style Sheets* (CSS)

Cascading Style Sheets (CSS) adalah suatu Bahasa *stylesheet* yang digunakan untuk mengatur tampilan suatu *website*, baik tata letaknya, jenis huruf, warna, dan semua yang berhubungan dengan tampilan. Pada umumnya CSS digunakan untuk memformat halaman web yang ditulis dengan HTML atau XHTML. Ada dua cara yang bisa diterapkan untuk menggunakan CSS pada web. Cara yang pertama dengan membuat CSS langsung di dalam satu *file* HTML (*internal/inline style sheet*). Cara yang kedua dengan memanggil CSS tersebut dari *file* CSS tersendiri (*external style sheet*) [5].

3. CodeIgniter

CodeIgniter adalah sebuah *framework* yang digunakan untuk membuat sebuah aplikasi berbasis web yang disusun menggunakan Bahasa PHP. Di dalam CI ini terdapat beberapa macam kelas yang berbentuk *library* dan *helper* yang berfungsi untuk membantu pemrogram dalam mengembangkan aplikasinya. CI sangat mudah dipelajari oleh seorang pemula sekalipun. Alasannya, karena CI mempunyai *file* dokumentasi yang sangat memadai untuk menjelaskan setiap fungsi yang ada pada *library* dan *helper*. *File* dokumentasi ini disertakan secara langsung pada saat mengunduh paket *framework* CI [6].

4. *My Structured Query Language* (MySQL)

MySQL adalah sebuah sistem manajemen *database* relasi (*Relational Database Management System*) yang bersifat "terbuka" *open source*. Terbuka maksudnya 24 adalah MySQL boleh di *download*

oleh siapa saja, baik versi kode program aslinya maupun versi binernya dan bisa digunakan secara gratis baik untuk dimodifikasi sesuai dengan kebutuhan seseorang maupun sebagai suatu program aplikasi komputer [7].

D. Pengujian

1. *Black Box Testing*

Pengujian *black box testing* merupakan pengujian yang berfokus pada persyaratan fungsional perangkat lunak. Pengujian *black box testing* disebut juga pengujian *behavioral* atau pengujian partisi. Pengujian *black box* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian *input* sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program [8].

2. *User Acceptance Testing*

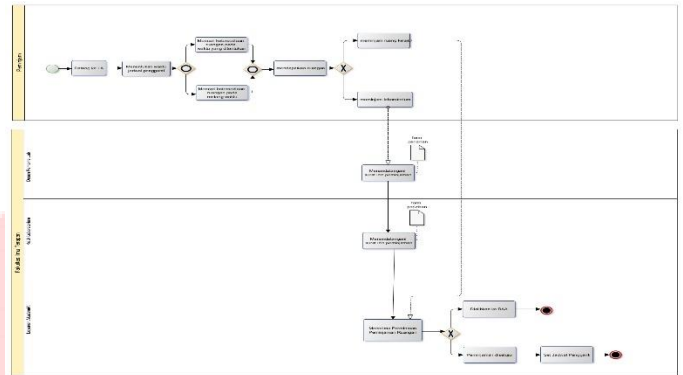
Pada jenis pengujian UAT, perangkat lunak akan diserahkan kepada pengguna untuk mengetahui apakah perangkat lunak memenuhi harapan pengguna dan bekerja seperti yang diharapkan. Pada pengembangan perangkat lunak, *user acceptance testing* (UAT), juga disebut juga pengujian beta (*beta testing*), pengujian aplikasi (*application testing*), dan pengujian pengguna akhir (*end user testing*) adalah tahapan pengembangan perangkat lunak ketika perangkat lunak diuji pada dunia nyata yang dimaksudkan oleh pengguna. Hasil pengujian dari pengguna akan dikembalikan kepada pengembang untuk selanjutnya dilakukan perubahan sebelum perangkat lunak dilepaskan secara komersil [9].

III. ANALISIS DAN PERANCANGAN

A. Gambaran Sistem Saat Ini

Proses peminjaman ruang kelas pengganti saat ini, peminjam (Mahasiswa dan Dosen) masih mengalami kesulitan dalam mencari ruang kelas untuk jadwal pengganti dikarenakan peminjaman ruangan saat ini masih dilakukan secara manual dengan datang langsung ke pihak penyedia ruangan (Layanan Akademik). Untuk mengetahui ketersediaan ruangan yang dapat digunakan untuk jadwal kuliah pengganti, peminjam (Mahasiswa dan Dosen) harus datang ke LA. Apabila ada ruangan yang tersedia, peminjam harus mengisi form peminjaman secara manual dan menunggu pihak LA mengatur jadwal pengganti di

sistem.



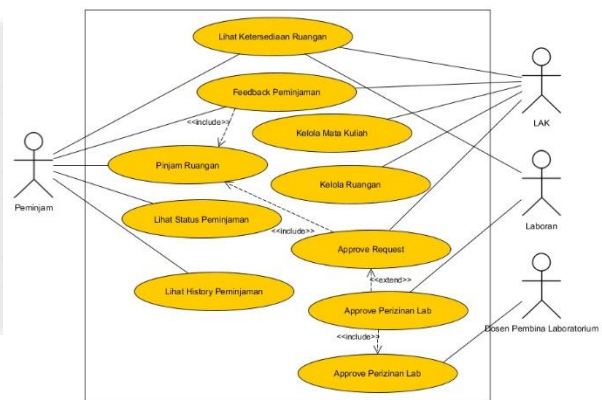
Gambar 2 Gambaran Sistem Saat Ini

B. Gambaran Sistem Usulan

Pada proses bisnis sistem usulan, *user* (mahasiswa dan dosen) dapat melihat ketersediaan ruang kelas yang dapat dipinjam secara langsung melalui aplikasi. Setelah *user* mendapatkan ruangan yang akan dipinjam, *user* dapat langsung masuk ke menu pinjam ruangan dan mengisi form yang ditampilkan dengan lengkap. Setelah itu pihak Layanan Akademik akan melakukan *approval* terhadap peminjaman yang dilakukan. Apabila pihak Layanan Akademik sudah melakukan *approval*, *user* akan mendapatkan pemberitahuan melalui *e-mail*.

C. Use Case Diagram

Berikut merupakan hasil analisis kebutuhan sistem yang digambarkan dalam *Use Case Diagram* yang digunakan pada Aplikasi Peminjaman Ruang Kelas Pengganti.



Gambar 3 Use Case Diagram

1. Deskripsi Aktor

Pada aplikasi peminjaman ruang kelas pengganti,

terdapat beberapa aktor yang terlibat dalam sistem. Adapun deskripsi aktor yang terlibat adalah sebagai berikut:

Tabel 1 Deskripsi Aktor

Nama Aktor	Deskripsi Aktor
Peminjam (Dosen, Mahasiswa)	Aktor yang dapat melakukan peminjaman ruangan.
LAK	Aktor yang dapat menyetujui data peminjaman ruangan
Dosen Pembina Laboratorium	Aktor yang menyetujui permintaan peminjaman laboratorium yang dibina.
Laboran	Aktor yang memiliki hak untuk melakukan approval perizinan peminjaman laboratorium setelah dilakukan approval oleh dosen Pembina laboratoium.

2. Deskripsi Use Case

Pada pembangunan aplikasi peminjaman ruangan, terdapat beberapa fungsionalitas yang digambarkan menggunakan Use Case diagram. Adapun Deskripsi Use Case Aplikasi Peminjaman Ruang Kelas Pengganti adalah sebagai berikut:

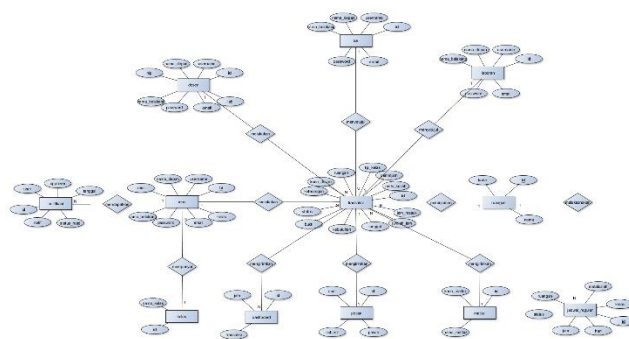
Tabel 2 Deskripsi Use Case

Nama Use Case	Deskripsi Use Case
Lihat Ketersediaan Ruangan	Use Case untuk melihat ketersediaan ruangan yang dapat dipinjam untuk kelas pengganti.
Pinjam Ruangan	Use Case untuk meminjam ruangan untuk jadwal kelas pengganti.
Lihat Status Peminjaman	Use Case untuk melihat status peminjaman ruangan yang akan digunakan untuk kelas pengganti.
Approve Request	Use Case untuk menyetujui permintaan peminjaman ruangan yang akan digunakan untuk kelas pengganti.
Feedback Peminjaman	Use Case untuk bertanya ataupun memberikan <i>feedback</i> /masukan mengenai peminjaman ruangan yang dilakukan oleh peminjam.
Kelola Matakuliah	Use case untuk mengelola data Matakuliah yang dapat dilakukan kuliah pengganti

Nama Use Case	Deskripsi Use Case
Kelola Ruangan	Use case untuk mengelola data ruangan yang ada di Fakultas Ilmu Terapan yang dapat digunakan untuk kuliah pengganti

D. Entity Relationship Diagram

ER Diagram digunakan untuk menjelaskan hubungan antar tabel yang terdapat pada basis data yang dibutuhkan oleh sistem berdasarkan data utama yang mempunyai relasi dengan tabel yang lain. Berikut ini adalah gambaran ER Diagram pada aplikasi peminjaman ruang kelas untuk jadwal kuliah pengganti jadwal reguler.



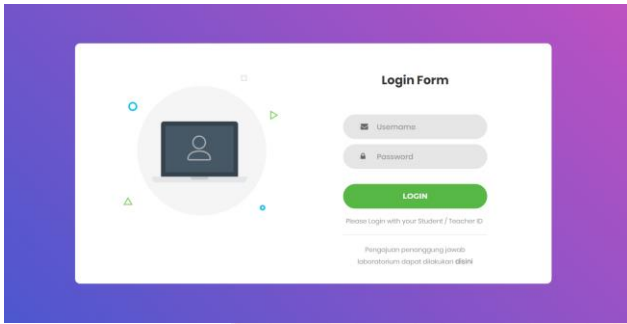
Gambar 4 ER Diagram

IV. IMPLEMENTASI DAN PENGUJIAN

A. IMPLEMENTASI

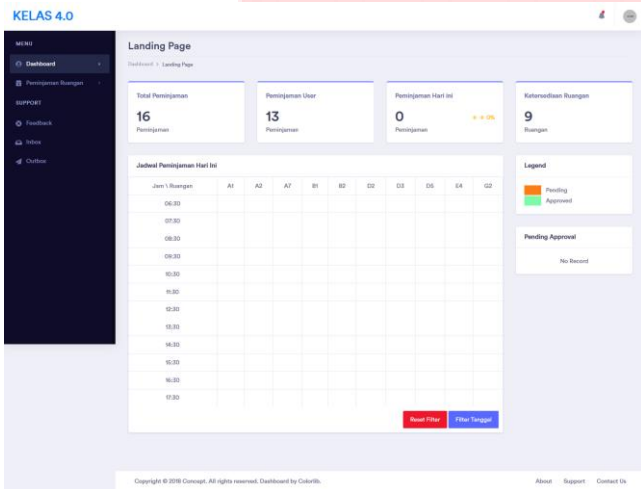
Berikut merupakan implementasi dari rancangan tampilan antarmuka dari aplikasi proyek akhir ini. Implementasi aplikasi yang telah dibuat yaitu halaman login, lihat ketersediaan ruangan, pinjam ruangan, approval, lihat status dan history peminjaman,

a. Halaman Login



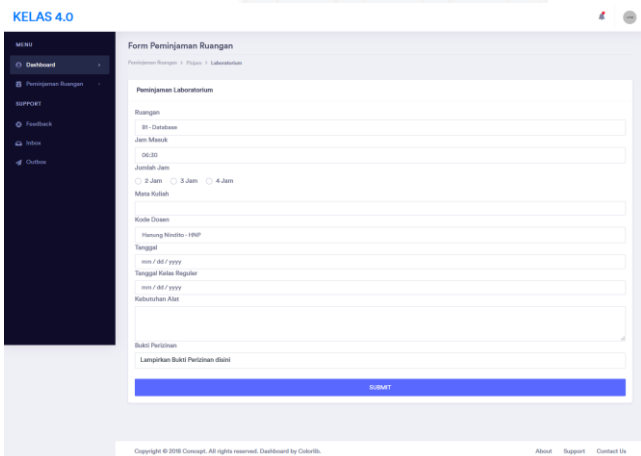
Gambar 5 Halaman Login

b. Halaman Lihat Ketersediaan Ruang



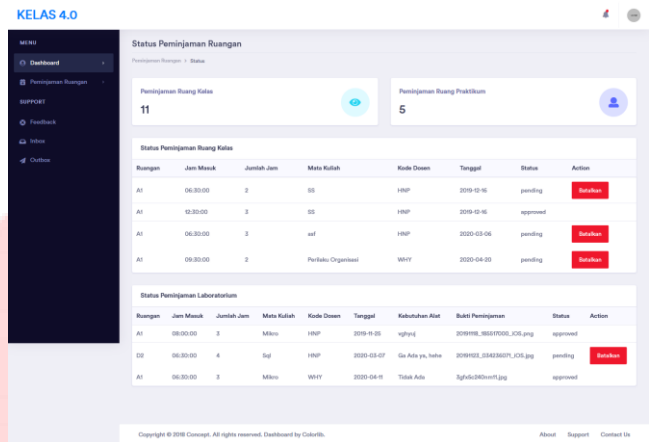
Gambar 6 Halaman Lihat Ketersediaan Ruang

c. Halaman Pinjam Ruang



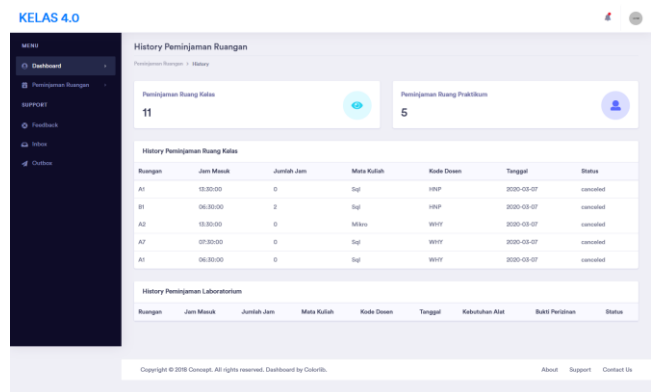
Gambar 7 Halaman Pinjam Ruang

d. Halaman Status Peminjaman



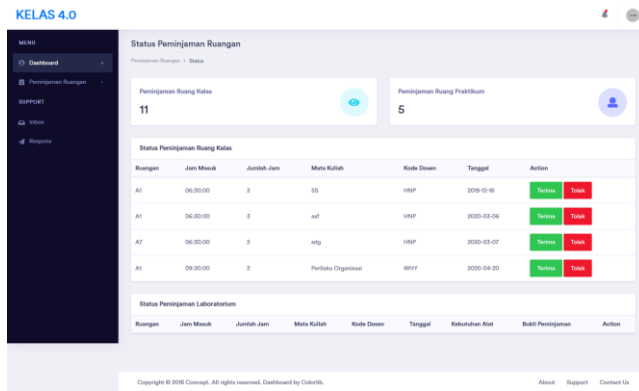
Gambar 8 Halaman Status Peminjaman

e. Halaman History Peminjaman



Gambar 9 Halaman History Peminjamane

f. Halaman Approval



Gambar 10 Halaman Approval

B. PENGUJIAN

Berdasarkan hasil skenario pengujian menggunakan test case dapat disimpulkan bahwa aplikasi dapat memenuhi hasil uji seperti yang diinginkan. Sehingga memenuhi tujuan yang ingin dicapai dalam pembuatan Aplikasi Peminjaman Ruang Kelas Untuk Kuliah Pengganti Jadwal Reguler (Studi Kasus : Fakultas Ilmu Terapan) yang memiliki fitur pinjam ruangan, approval peminjaman, lihat ketersediaan ruangan, status peminjaman dan history peminjaman.

V. KESIMPULAN

Berdasarkan hasil dari melakukan analisis, perancangan, implementasi hingga pengujian Aplikasi Peminjaman Ruang Kelas Untuk Kuliah Pengganti, maka kesimpulan yang dapat diambil adalah telah berhasil dibuat Aplikasi Peminjaman Ruang Kelas Untuk Kuliah Pengganti Jadwal Reguler (Studi Kasus : Fakultas Ilmu Terapan) dengan fitur yang memudahkan proses peminjaman ruangan seperti :

1. Fitur “Pinjam Ruang” yang membantu proses peminjaman ruangan untuk kuliah pengganti secara *online*.
2. Fitur “Lihat Ketersediaan Ruang” yang memperlihatkan data ketersediaan ruangan yang dapat digunakan untuk kuliah pengganti.
3. Fitur “Status Peminjaman” untuk melacak status terakhir peminjaman ruangan yang dilakukan.

4. Fitur “Approval & Notification” untuk menyetujui permintaan peminjaman ruang kelas untuk kuliah pengganti.

VI. DAFTAR PUSTAKA

- [1] R. A. S. dan M. Salahudin, Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek, Bandung: Informatika, 2014.
- [2] Maniah dan D. Hamidin, Analisis dan Perancangan Sistem Informasi: Pembahasan Secara Praktis dengan Contoh Kasus, Yogyakarta: Deepublish, 2017.
- [3] S. Mulyani, Metode Analisis dan Perancangan Sistem, Bandung: Abdi Sistemika, 2016.
- [4] S. B., Pemrograman Web Dengan HTML, Bandung: Informatika, 2012.
- [5] T. Suryana dan K. , Aplikasi Internet Menggunakan HTML, CSS & Javascript, Jakarta: PT Elex Media Komputindo, 2014.
- [6] Wardana, Menjadi Master PHP dengan Framework Codeigniter, Jakarta: PT. Elex Media Komputindo, 2010.
- [7] Arbie, Manajemen Database dengan MySQL, Yogyakarta: ANDI, 2004.
- [8] R. Pressman, Software Engineering: A Practitioner's Approach, McGraw-Hill Education, 2014.
- [9] J. Simarmata, Rekayasa Perangkat Lunak, Yogyakarta: ANDI OFFSET, 2010.