

## Pembangunan Sistem Pengelolaan Pengajuan Kegiatan Mahasiswa Universitas Brawijaya

Dainty Resfulany Susanto<sup>1</sup>, Denny Sagita Rusdianto<sup>2</sup>, Tri Astoto Kurniawan<sup>3</sup>

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya  
Email: <sup>1</sup>resfulany@gmail.com, <sup>2</sup>denny.sagita@ub.ac.id, <sup>3</sup>triak@ub.ac.id

### Abstrak

Manual Prosedur (MP) pengajuan kegiatan mahasiswa Universitas Brawijaya masih memiliki kekurangan. Kekurangan tersebut yakni masih perlu dilakukan penindaklanjutan pengajuan secara langsung oleh Lembaga Kedaulatan Mahasiswa (LKM) pada setiap alur prosedurnya. Hal ini dikarenakan tidak adanya status pengajuan yang diinformasikan kepada LKM. Proses penindaklanjutan status pengajuan saat ini sering menemui kendala seperti pegawai rektorat yang akan ditemui sedang tidak berada di tempat dan/atau status pengajuan belum berubah sehingga penindaklanjutan harus dilakukan berkali-kali. Dari sisi kemahasiswaan, juga dibutuhkan suatu sistem yang dapat membantu proses pengelolaan pengajuan kegiatan LKM. Hal tersebut dilakukan dalam rangka *monitoring* dan pencatatan aktivitas LKM. Sebab dalam setiap bulan jumlah pengajuan kegiatan dari seluruh LKM cukup banyak. Untuk mengatasi permasalahan tersebut, diperlukan sistem pengelolaan pengajuan kegiatan mahasiswa yang diharapkan dapat membantu memudahkan proses pengajuan, pengaturan, dan pengelolaan kegiatan mahasiswa. Sistem ini dibuat dalam bentuk web, karena web merupakan media yang dapat diakses dimana saja dan kapan saja. Sistem ini telah berhasil diuji dengan melakukan pengujian unit dan integrasi terhadap 3 sampel fungsi dari beberapa kelas menggunakan metode *white-box testing* dan mendapatkan hasil 100% valid. Selain itu, 23 kebutuhan fungsional telah diuji menggunakan metode *black-box testing* dan mendapatkan hasil 100% valid.

**Kata kunci:** *pengajuan kegiatan mahasiswa, pengajuan proposal kegiatan, pengembangan berorientasi objek, waterfall model, sistem pengelolaan*

### Abstract

*The manual of procedures of student event submission in Brawijaya University (UB) has a disadvantage. The disadvantage is Student Sovereignty Institutions (LKM) need to follow up their submission for each procedure because there is no information about submission status. Currently, the follow-up processes often get some problems such as the absence of UB's student affairs staff and/or their submission status has not been changed so that the follow-up processes need to be repeated. Meanwhile, UB's student affairs also need a management system to manage student event submission because in every month there are so many proposals. These problems can be solved by a system that can help the arrangement and management of submission processes. The system was created in the web-based, so users can use this system anywhere and anytime. This system has been tested through the white-box method for the unit and the integration testing of 3 sample methods from several classes get 100% valid result. Then, 23 functional requirements have been tested by black-box testing for validation testing get 100% valid result.*

**Keywords:** *student event submission, event proposal submission, object oriented development, waterfall model, management system*

### 1. PENDAHULUAN

Universitas Brawijaya (UB) merupakan salah satu lembaga pendidikan tinggi negeri yang terletak di kota Malang. UB saat ini

memiliki ribuan mahasiswa aktif. Tentu hal tersebut merupakan potensi yang luar biasa. Potensi tersebut diantaranya adalah potensi akademik, nonakademik, dan bidang yang lain seperti kepedulian sosial (Bidang Kemahasiswaan, 2017). Dengan jumlah

mahasiswa yang berkisar 62.000 orang, UB mewadahi kegiatan mahasiswa di dalam suatu Lembaga Kedaulatan Mahasiswa (LKM). LKM UB dipetakan berdasarkan jenis kegiatannya menjadi 3 (tiga) yaitu kelompok minat bakat, penalaran, dan kesejahteraan mahasiswa. Setiap tahunnya, LKM menyusun program kerja yang baru (Bidang Kemahasiswaan, 2017). Pengajuan program kerja LKM harus sesuai dengan prosedur yang telah ditetapkan oleh Bidang Kemahasiswaan UB.

Secara garis besar, Manual Prosedur (MP) pengajuan kegiatan LKM adalah dengan menggunakan media berupa proposal kegiatan. Namun, sebelum LKM dapat mengajukan kegiatan melalui suatu proposal, LKM harus terlebih dahulu mengajukan daftar program kerja yang akan dilaksanakan selama satu periode kepengurusan di awal tahun. Setelah program kerja diterima, maka proposal kegiatan dapat diajukan. Proses pengajuan proposal kegiatan dimulai dengan menyerahkan proposal yang telah ditandatangani oleh Pembina LKM dan Presiden EM kepada Wakil Rektor III. Setelah proposal disetujui, maka proposal akan didisposisi kepada Kepala Biro (Kabiro) Akademik dan Kemahasiswaan untuk ditelaah tingkat kegiatannya. Kemudian proposal didisposisi kepada Kepala Bagian (Kabag) Kemahasiswaan untuk ditentukan besaran dana yang disetujui. Setelah itu, proposal didisposisi kepada Kepala Subbagian (Kasubag) yang bersangkutan dan diserahkan kepada Pemegang Uang Muka Kegiatan (PUMK) untuk proses pencairan dana kegiatan. Dana kegiatan yang diberikan adalah sebesar 70% dari total dana yang disetujui. Sisa dana yakni sebesar 30% dari total dana akan diberikan kepada LKM setelah LKM menyerahkan Laporan Pertanggungjawaban (LPJ) kegiatan.

MP pengajuan kegiatan mahasiswa masih memiliki kekurangan. Kekurangan tersebut yakni masih perlu dilakukan penindaklanjutan pengajuan secara langsung oleh LKM pada setiap alur prosedurnya. Hal ini dikarenakan tidak adanya status pengajuan yang diinformasikan kepada LKM. Proses penindaklanjutan status pengajuan saat ini sering menemui kendala seperti pegawai rektorat yang akan ditemui sedang tidak berada di tempat dan/atau status pengajuan belum berubah sehingga penindaklanjutan harus dilakukan berkali-kali. Dari sisi kemahasiswaan, juga dibutuhkan suatu sistem yang dapat membantu proses pengelolaan pengajuan kegiatan LKM.

Hal tersebut dilakukan dalam rangka monitoring dan pencatatan aktivitas LKM. Sebab dalam setiap bulan jumlah pengajuan kegiatan dari seluruh LKM cukup banyak.

Untuk mengatasi permasalahan tersebut diperlukan pembangunan sistem pengelolaan pengajuan kegiatan mahasiswa Universitas Brawijaya. Di dalam pengembangan sistem ini terdapat beberapa rumusan masalah, yang pertama adalah kebutuhan apa saja yang harus ada pada sistem yang akan dikembangkan, yang kedua bagaimana perancangan dan implementasi dari sistem ini, dan yang terakhir adalah bagaimana pengujian dari sistem ini. Sehingga hasil dari pembangunan sistem ini diharapkan dapat membantu LKM dalam mengajukan dan mengetahui status pengajuan kegiatan. Selain itu sistem ini diharapkan dapat membantu proses pengelolaan pengajuan kegiatan. Sistem ini akan dibuat dalam bentuk web sehingga dapat diakses dimana saja dan kapan saja.

Pembangunan sistem pengelolaan pengajuan kegiatan mahasiswa ini menggunakan model *Software Development Life Cycle (SDLC) Waterfall*. Model tersebut dipilih karena kebutuhan sistem ini telah terdefinisi pada tahap awal dan kebutuhan tidak berubah-ubah. Dalam proses pengelolaan data, sistem ini menggunakan bahasa pemrograman PHP dan *framework CodeIgniter*. *CodeIgniter* dibangun dengan menggunakan konsep pengembangan *Model-View-Controller* (Daqiqil, 2011). Sedangkan untuk proses pengelolaan basis data, sistem ini menggunakan *Database Management System (DBMS)* berupa MySQL.

## 2. METODOLOGI PENELITIAN

Pada tahap metodologi penelitian terdapat beberapa tahapan yang digunakan dalam membangun Sistem Pengelolaan Pengajuan Kegiatan Mahasiswa. Metodologi yang dibuat mengacu pada model *SDLC Waterfall*. Alur dalam penelitian ini dapat dilihat pada Gambar 1.

Tahap pertama pada metodologi penelitian adalah studi literatur. Studi literatur dilakukan untuk mengumpulkan teori yang berasal dari sumber.

Tahap kedua merupakan tahap rekayasa kebutuhan. Di dalam proses rekayasa kebutuhan terdapat proses analisis kebutuhan. Pada tahap analisis kebutuhan dilakukan proses elisitasi kebutuhan untuk menghasilkan spesifikasi

kebutuhan yang dikategorikan menjadi 2 jenis kebutuhan yakni kebutuhan fungsional dan nonfungsional dan identifikasi aktor. Kemudian dibuat pemodelan kebutuhan berupa *Use Case Diagram*, *Use Case Scenario*, dan *Class Diagram*.

Tahap ketiga merupakan tahap perancangan sistem. Pada tahap ini dibuat pemodelan *Sequence Diagram* dari hasil pemodelan kebutuhan. Hasil pemodelan *Sequence Diagram* kemudian dimodelkan menjadi *Class Diagram* dan membuat algoritme pada masing-masing fungsi di setiap klas. Kemudian membuat model *Entity Relationship Diagram (ERD)*, *Physical Data Model (PDM)*, dan rancangan antarmuka.

Tahap keempat adalah tahap implementasi. Implementasi sistem dilakukan berdasarkan rancangan *class diagram*, algoritme, *PDM*, dan antarmuka.

Tahap kelima merupakan tahap pengujian dan analisis. Pada tahap ini, dilakukan pengujian unit, integrasi, dan validasi serta menganalisis hasil dari masing-masing pengujian.

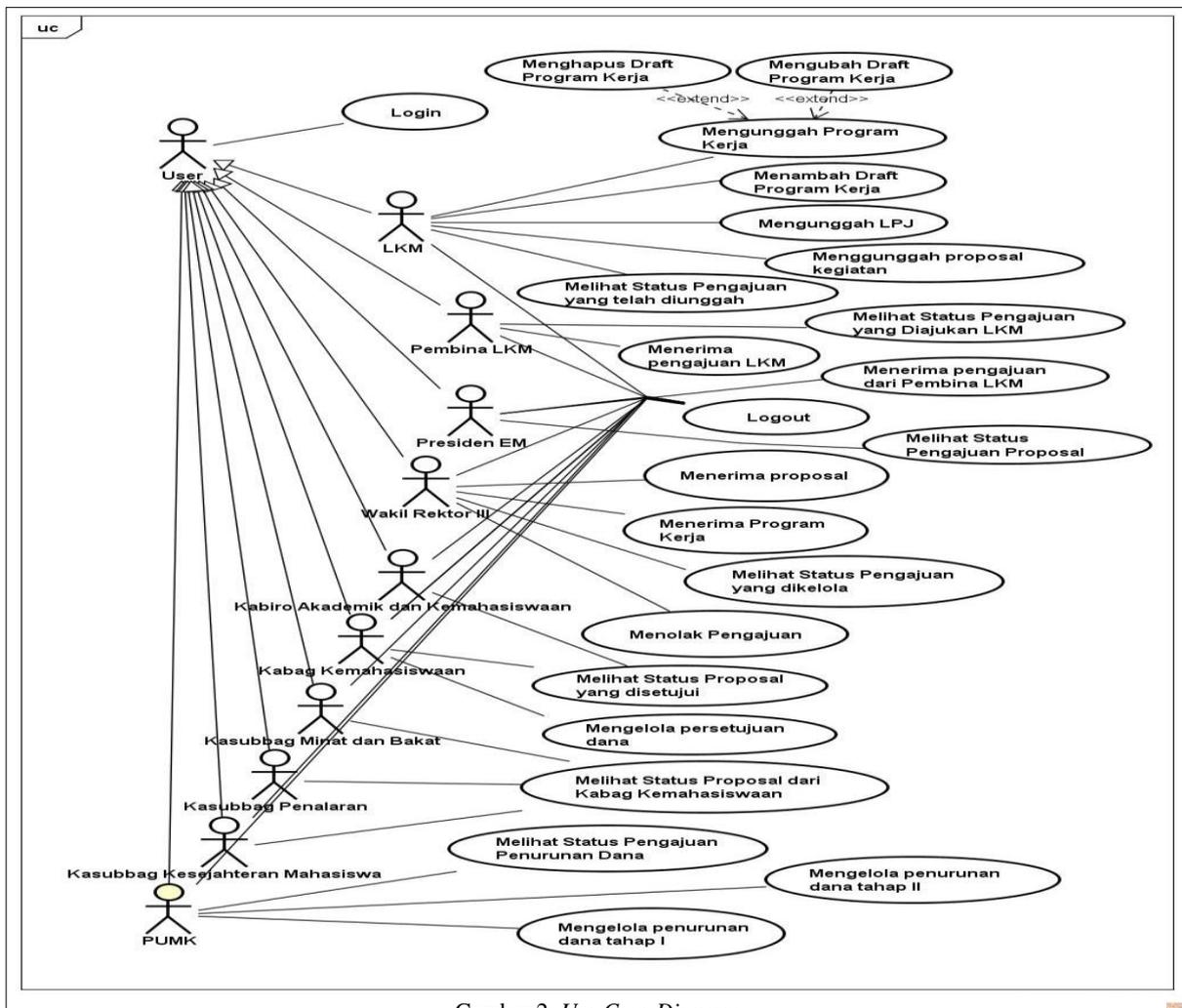
Tahap yang terakhir adalah tahap

pengambilan kesimpulan dan saran. Kesimpulan dan saran diperoleh setelah semua tahapan dari studi literatur hingga pengujian telah dilakukan.



Gambar 1. Diagram Alur Metodologi Penelitian

### 3. REKAYASA KEBUTUHAN



Gambar 2. Use Case Diagram

Rekayasa kebutuhan adalah tahap awal dalam pembangunan sistem ini. Tahap ini bertujuan untuk menggali kebutuhan apa saja yang harus dimiliki sistem sehingga dapat digunakan dalam perancangan sistem menggunakan pendekatan berbasis objek.

Tahap pertama yang dilakukan adalah elisitasi kebutuhan, kemudian mengidentifikasi aktor. Kemudian mendefinisikan dan menspesifikasikan kebutuhan sistem. Setelah itu dilakukan pemodelan *use case diagram* dari seluruh kebutuhan fungsional yang telah didefinisikan. *Use Case Diagram* tertera pada Gambar 2. Setelah membuat *use case diagram*, kemudian dibuat *use case scenario* dan *class diagram*.

Sistem pengelolaan pengajuan kegiatan mahasiswa ini memiliki dua puluh tiga kebutuhan fungsional yang dapat membantu proses pengelolaan pengajuan kegiatan mahasiswa. Dan sistem pengelolaan pengajuan kegiatan mahasiswa ini juga mempunyai parameter kebutuhan nonfungsional *compatibility*, yaitu sistem harus dapat dibuka pada berbagai *browser* seperti *Google Chrome* dan *Mozilla Firefox*.

#### 4. PERANCANGAN SISTEM

Tahap selanjutnya adalah tahap perancangan. Perancangan yang dilakukan menggunakan pendekatan berorientasi objek dengan diagram UML (*Unified Modelling Language*). Perancangan ini bertujuan untuk mengidentifikasi objek dan hubungan antar objek berdasarkan spesifikasi kebutuhan dan *use case scenario* yang telah dibuat pada tahap rekayasa kebutuhan. Terdapat 4 (empat) tahapan yang dilakukan pada perancangan yakni pemodelan *sequence diagram*, *class diagram*, *database*, dan antarmuka.

Pemodelan *sequence diagram* dilakukan untuk menggambarkan interaksi antar objek. Pemodelan *class diagram* dilakukan untuk mendefinisikan struktur kelas dan hubungan antar kelas berdasarkan hasil pemodelan *class diagram* dari tahap rekayasa kebutuhan dan *sequence diagram*. Kemudian dilakukan perancangan algoritme pada masing-masing fungsi yang ada di dalam kelas pada *class diagram*. Hasil pemodelan *class diagram* dapat dilihat pada Gambar 3.

Proses selanjutnya adalah merancang basis data yang dilakukan secara *conceptual* dan *physical*. Perancangan basis data secara

*conceptual* yaitu dengan menggunakan *Entity Relationship Diagram*, sedangkan perancangan basis data secara *physical* menggunakan *Physical Data Model*. Perancangan basis data secara *conceptual* dapat dilihat pada Gambar 4.

Tahap yang terakhir dalam perancangan sistem ini adalah merancang antarmuka. Tahap perancangan antarmuka merupakan tahap membuat rancangan tampilan sistem yang akan dibangun.

#### 5. IMPLEMENTASI

Implementasi dilakukan berdasarkan hasil yang didapatkan dari rekayasa kebutuhan dan perancangan. Pada tahap ini, rancangan *class diagram*, algoritme, basis data dan antarmuka diimplementasikan. Rancangan *class diagram* dan algoritme dibuat menjadi kode program.

Berikut adalah beberapa hasil implementasi antarmuka diantaranya antarmuka mengunggah proposal pada Gambar 5, antarmuka mengunggah laporan pertanggungjawaban pada Gambar 6, dan antarmuka mengelola persetujuan dana pada Gambar 7.

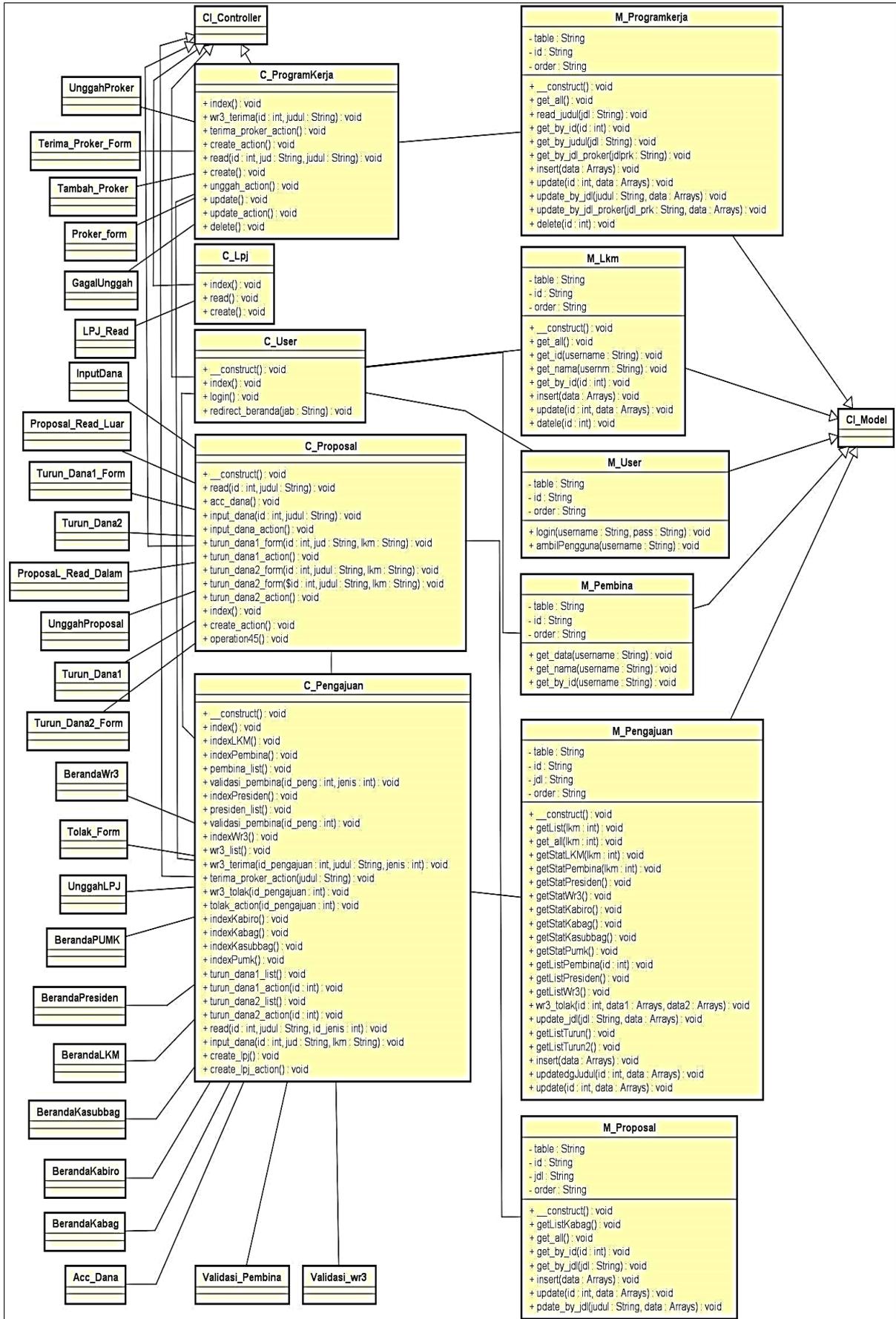
#### 6. PENGUJIAN

Tahap ini dilakukan setelah tahap implementasi selesai. Pengujian dilakukan untuk memeriksa apakah hasil implementasi sesuai dengan rekayasa kebutuhan dan perancangan atau tidak. Pengujian yang akan dilakukan adalah pengujian unit, pengujian integrasi, dan pengujian validasi.

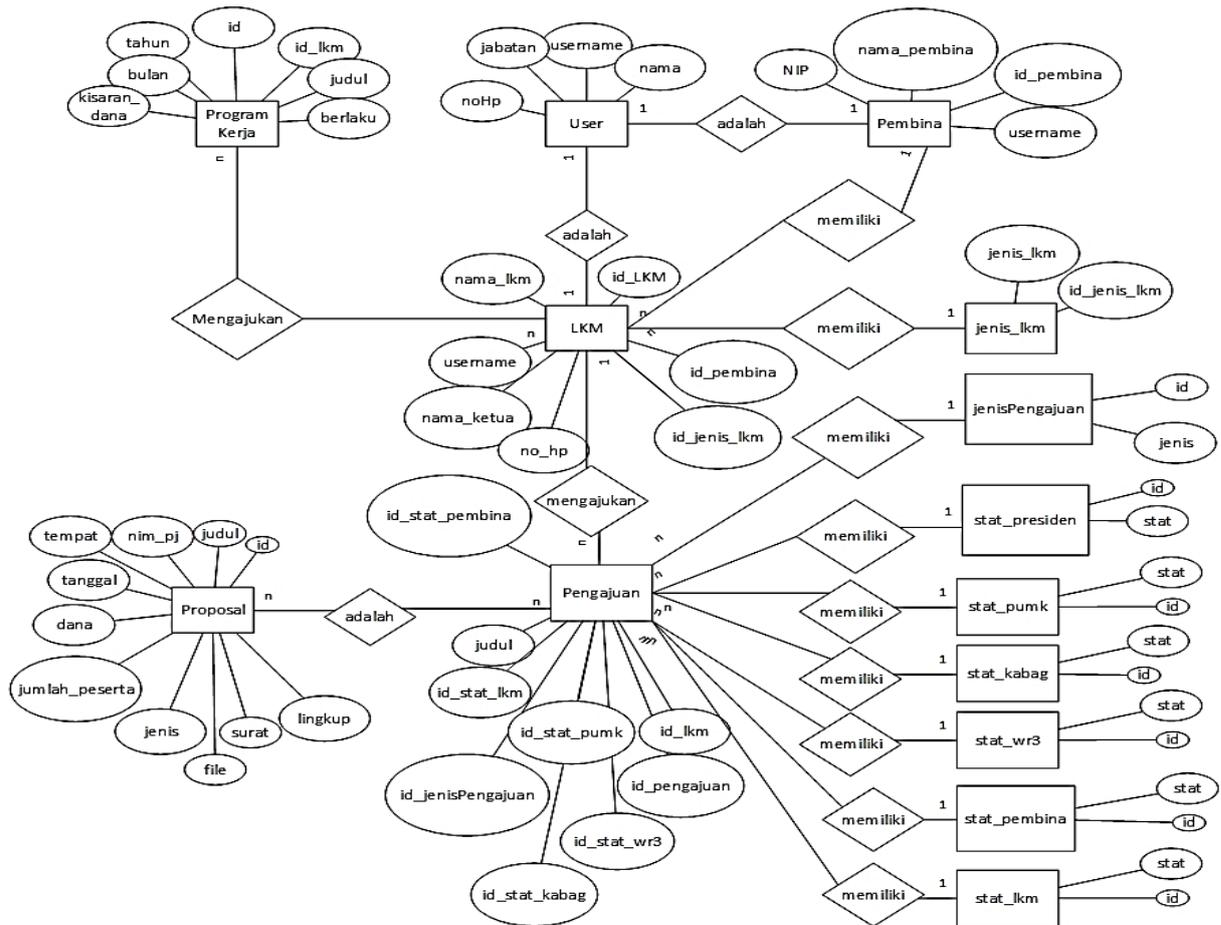
Pengujian unit adalah pengujian yang dilakukan terhadap setiap operasi dari setiap kelas. Pengujian unit dilakukan dengan menggunakan *basis path testing* yang merupakan salah satu jenis pengujian dari *white box testing*. Langkah-langkah *white-box testing* adalah membuat *flow graph* dari hasil perancangan algoritme, menghitung *cyclomatic complexity* dari hasil *flow graph*, menentukan jalur independen dari hasil *flow graph*, dan membuat kasus uji berdasarkan jalur independen.

Pengujian unit dilakukan pada operasi *validasi\_pembina()* pada kelas *C\_Pengajuan*, *create\_action()* pada kelas *C\_Proposal*, dan operasi *wr3\_terima* pada kelas *C\_Pengajuan*. Hasil dari pengujian unit ini adalah 100% semua kasus uji valid pada ketiga operasi tersebut.

Pengujian integrasi merupakan pengujian yang berfokus pada hubungan antar kelas. Pengujian integrasi dilakukan menggunakan



Gambar 3. Class Diagram



Gambar 4. Entity Relationship Diagram (ERD)

Gambar 5. Implementasi Antarmuka Halaman Mengunggah Proposal

Gambar 6. Implementasi Antarmuka Halaman Mengunggah Laporan Pertanggungjawaban

Gambar 7. Implementasi Antarmuka Halaman Mengelola Perseetujuan Dana

*basis path testing*. Pengujian integrasi dilakukan pada operasi `input_dana_action()` dari kelas `C_Proposal` yang memanggil operasi `index()` pada kelas `C_Pengajuan`, operasi `wr3_terima()` dari kelas `C_Pengajuan` yang memanggil operasi `wr3_terima()` pada kelas `C_ProgramKerja`, dan operasi `create_action` dari kelas `C_Proposal` yang memanggil operasi `index()` pada kelas `C_Pengajuan`. Hasil pengujian integrasi ini adalah 100% semua kasus uji valid pada ketiga operasi tersebut.

Pengujian validasi adalah pengujian yang dilakukan untuk memastikan bahwa semua kebutuhan terpenuhi pada sistem dan untuk memastikan bahwa tidak terdapat kesalahan pada saat menjalankan kebutuhan tersebut. Pengujian ini menggunakan *equivalence partitioning testing* dan *scenario-based testing* yang merupakan jenis pengujian dari metode *black-box testing*. Hasil dari pengujian validasi ini adalah 100% semua kasus uji valid pada 23 kebutuhan fungsional.

## 7. KESIMPULAN

Berdasarkan hasil penelitian mengenai “Pembangunan Sistem Pengelolaan Pengajuan Kegiatan Mahasiswa Universitas Brawijaya” yang telah dilakukan maka dapat ditarik kesimpulan dari seluruh tahapan yang telah dilakukan.

Berdasarkan hasil analisis kebutuhan, terdapat 11 aktor dan 23 kebutuhan fungsional yang dapat membantu proses pengelolaan pengajuan kegiatan mahasiswa.

Berdasarkan hasil perancangan terdapat 44 kelas yang terbagi dalam komponen *Model*, *View*, dan *Controller* yang dijelaskan dalam bentuk algoritme pada masing-masing operasi dari

setiap kelas, 14 tabel dari hasil perancangan basis data, dan antarmuka yang dijadikan acuan pada tahap implementasi. Hasil implementasi sistem ini menghasilkan fitur yang dapat membantu melakukan pengelolaan pengajuan kegiatan mahasiswa.

Berdasarkan hasil pengujian yang telah dilakukan, 100% pengujian yang dilakukan bernilai valid. Pengujian dilakukan dengan menggunakan teknik pengujian *white-box testing* dan *black-box testing*. Pengujian *white-box testing* dilakukan pada pengujian unit dan integrasi. Pengujian validasi menggunakan *black-box testing* dilakukan pada seluruh kebutuhan fungsional.

## DAFTAR PUSTAKA

- Bidang Kemahasiswaan, 2017. *Buku Panduan LKM Universitas Brawijaya*, Malang: Universitas Brawijaya.
- Bidang Kemahasiswaan, 2017. *Buku Panduan Pengajuan dan Pelaporan Kegiatan*, Malang: Universitas Brawijaya.
- Daqiqil, I., 2011. *Framework CodeIgniter Sebuah Panduan dan Best Practice*. CodeIgniter Lovers Community ed. Pekanbaru: Pekanbaru.