

**RANCANG BANGUN APLIKASI *ENTERPRISE RESOURCE PLANNING*
PADA LEMBAGA KEDAULATAN MAHASISWA
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
(LKM PTIIK)**

SKRIPSI

LABORATORIUM SISTEM INFORMASI

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer



Disusun Oleh :

FARIZ IZZAN AGUSTA

NIM. 105060805111002

**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
PROGRAM STUDI INFORMATIKA
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA**

MALANG

2014

LEMBAR PERSETUJUAN

**RANCANG BANGUN APLIKASI *ENTERPRISE RESOURCE PLANNING*
PADA LEMBAGA KEDAULATAN MAHASISWA
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
(LKM PTIIK)**

SKRIPSI

LABORATORIUM SISTEM INFORMASI

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer



Disusun Oleh :

FARIZ IZZAN AGUSTA

NIM. 105060805111002

Telah diperiksa dan disetujui pada 18 Juni 2014

Dosen Pembimbing I

Dosen Pembimbing II

Fajar Pradana, S.ST., M.Eng
NIK. 87112116110371

Aditya Rachmadi, S.ST., M.TI
NIK. 86042116110426

LEMBAR PENGESAHAN

**RANCANG BANGUN APLIKASI *ENTERPRISE RESOURCE PLANNING*
PADA LEMBAGA KEDAULATAN MAHASISWA
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
(LKM PTIIK)**

SKRIPSI

LABORATORIUM SISTEM INFORMASI

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer

Disusun oleh :

FARIZ IZZAN AGUSTA

NIM. 105060805111002

Skripsi ini telah diuji dan dinyatakan lulus pada tanggal 4 Juli 2014

Penguji 1

Suprpto, ST., MT
NIP. 19710727 199603 1 001

Penguji 2

Edy Santoso, S.Si., M.Kom
NIP. 19740414 200312 1 004

Penguji 3

Satrio Agung Wicaksono, S.Kom., M.Kom
NIP. 19860521 201212 1 001

Mengetahui

Ketua Program Studi Informatika / Ilmu Komputer

Drs. Marji, M.T.

NIP. 19670801 199203 1 001

**PERNYATAAN
ORISINALITAS SKRIPSI**

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah SKRIPSI ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata didalam naskah SKRIPSI ini dapat dibuktikan terdapat unsur-unsur PLAGIASI, saya bersedia SKRIPSI ini digugurkan dan gelar akademik yang telah saya peroleh (SARJANA) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku. (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).



Malang, Juni 2014
Mahasiswa,

Fariz Izzan Agusta

NIM. 105060805111002

KATA PENGANTAR

Dengan nama Allah SWT Yang Maha Pengasih dan Penyayang. Segala puji bagi Allah SWT karena atas rahmat dan hidayahNya-lah penulis dapat menyelesaikan Tugas Akhir yang berjudul “**Rancang Bangun Aplikasi Enterprise Resource Planning Pada Lembaga Kedaulatan Mahasiswa Program Teknologi Informasi Dan Ilmu Komputer (LKM PTIIK)**”. Shalawat serta salam atas junjungan besar kita Nabi Muhammad S.A.W. beserta keluarga dan para sahabat sekalian. Tugas Akhir ini disusun untuk memenuhi sebagai persyaratan memperoleh gelar Sarjana Komputer di Program Studi Informatika/Ilmu Komputer Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya Malang.

Melalui kesempatan ini, penulis menyampaikan rasa hormat dan terima kasih penulis yang sebesar-besarnya kepada pihak yang telah memberikan bantuan lahir maupun batin selama penulisan tugas akhir ini. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan rasa hormat dan terima kasih penulis kepada :

1. Fajar Pradana, S.ST., M.Eng dan Aditya Rachmadi, S.ST., M.TI selaku dosen pembimbing tugas akhir penulis.
2. Suprpto, ST., MT, Edy Santoso, S.Si., M.Kom, Satrio Agung Wicaksono, S.Kom., M.Kom selaku dosen penguji tugas akhir penulis.
3. Ir. Sutrisno, M.T, Ir. Heru Nurwasito, M.Kom, Himawat Aryadita, S.T, M.Sc, dan Edy Santoso, S.Si., M.Kom selaku Ketua, Wakil Ketua 1, Wakil Ketua 2 dan Wakil Ketua 3 Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
4. Bapak Drs. Marji, MT dan Bapak Issa Arwani, S.Kom., M.Sc selaku Ketua dan Sekretaris Program Studi Informatika serta segenap Bapak / Ibu Dosen, Staff Administrasi dan Perpustakaan Program Studi Informatika Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
5. Seluruh dosen Program Studi Informatika atas kesediaan membaginya ilmunya kepada penulis.
6. Semua Asisten Praktikum, Laboran, dan Kepala Laboratorium pada Program Teknologi Informasi dan Ilmu Komputer yang telah memberikan banyak bantuan dan dukungan dalam menyelesaikan tugas akhir ini.Orang

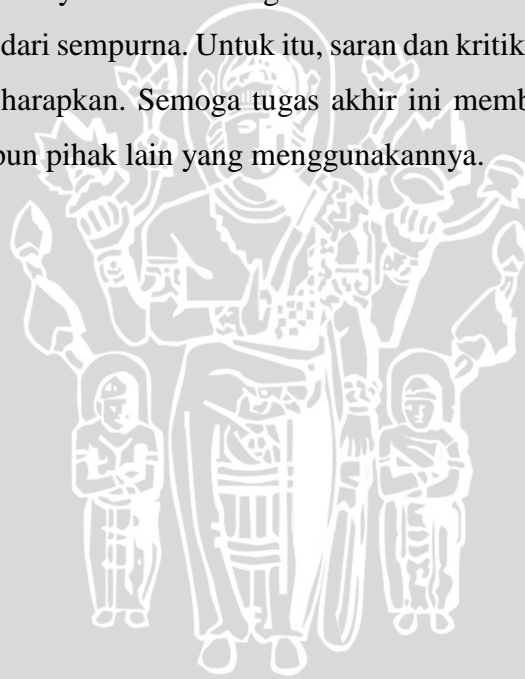
tua penulis dan seluruh keluarga yang senantiasa tiada henti hentinya memberikan doa demi terselesainya tugas akhir ini.

7. Seluruh rekan-rekan dan direksi Cloud Astro yang telah memberikan banyak bantuan dan dukungan dalam menyelesaikan tugas akhir ini.
8. Seluruh rekan-rekan Lembaga Kedaulatan Mahasiswa yang telah bersedia bekerjasama serta memberikan banyak bantuan dan dukungan dalam menyelesaikan tugas akhir ini.
9. Seluruh pihak yang telah membantu kelancaran penulisan tugas akhir yang tidak dapat penulis sebutkan satu per satu.

Penulis menyadari bahwa tugas akhir ini masih banyak kekurangan dan masih jauh dari sempurna. Untuk itu, saran dan kritik yang membangun, sangat penulis harapkan. Semoga tugas akhir ini membawa manfaat bagi penyusun maupun pihak lain yang menggunakannya.

Malang, Juli 2014

Penulis



ABSTRAK

Dalam dunia organisasi, koordinasi merupakan suatu hal yang penting. Koordinasi antar organisasi seringkali menjadi kendala yang memiliki dampak cukup berarti. Penelitian ini akan memberikan solusi untuk meningkatkan koordinasi tersebut dengan menggunakan aplikasi *Enterprise Resource Planning*. *Enterprise Resource Planning* adalah sistem informasi terintegrasi yang digunakan untuk mendukung proses bisnis dan manajemen sumber daya dalam suatu organisasi. Penelitian ini dikhususkan pada Lembaga Kedaulatan Mahasiswa Program Teknologi Informasi dan Ilmu Komputer (LKM PTIIK). Analisa kebutuhan pada penelitian ini dilakukan dengan cara survey dan observasi secara langsung. Aplikasi yang dibuat pada penelitian ini merupakan aplikasi berbasis web. Aplikasi ini dikembangkan menggunakan bahasa pemrograman PHP dengan *framework* Codeigniter. Untuk penyimpanan data, aplikasi ini menggunakan MySQL. Metode pengembangan aplikasi yang digunakan pada penelitian ini adalah *Extreme Programming*. Metode *Extreme Programming* memungkinkan pengembang aplikasi untuk berkomunikasi langsung dengan pengguna ketika pengembang membutuhkan feedback dari pengguna. Sehingga aplikasi dapat dengan cepat diselesaikan dan sesuai dengan yang dibutuhkan. Pengujian yang dilakukan pada penelitian ini menggunakan *functional test*, *usability test* dan *user acceptance test*. Hasil dari *functional test* menyatakan bahwa seluruh kebutuhan fungsional aplikasi telah terpenuhi dan 100% valid. *Usability test* menghasilkan kesimpulan bahwa penggunaan aplikasi ini dapat dimengerti dengan baik. *User acceptance test* menghasilkan kesimpulan bahwa aplikasi ini diterima dan telah sesuai dengan yang dibutuhkan.

Kata Kunci: *Enterprise Resource Planning*, LKM PTIIK, Codeigniter, *Extreme Programming*, integrasi.

ABSTRACT

In the world of organization, coordination is an important thing. Coordination among organizations often become obstacles which have a significant impact. This research will provide solutions to improve the coordination by using Enterprise Resource Planning applications. Enterprise Resource Planning is an integrated information system that is used to support business processes and management of resources in an organization. This study is devoted to Lembaga Kedaulatan Mahasiswa Program Teknologi Informasi dan Ilmu Komputer (LKM PTIIK). Requirements analysis in this study conducted by survey and direct observations. Applications were made in this study is a web-based application. This application was developed using the PHP programming language with CodeIgniter framework. For data storage, the application is using MySQL. Application development methods used in this study is Extreme Programming. Extreme Programming method enables application developers to communicate directly with users when developers need feedback from users. So that the application can be quickly resolved and as required. Tests were conducted in this study using a functional test, usability test and user acceptance test. The results of functional tests said that all the functional requirements have been met and the test 100% valid. The result of usability testing is the use of these applications can be well understood. The result of user acceptance testing is this application has been accepted and as required.

Keywords: *Enterprise Resource Planning, LKM PTIIK, CodeIgniter, Extreme Programming, integration.*

DAFTAR ISI

KATA PENGANTAR..... i

ABSTRAK iii

ABSTRACT iv

DAFTAR ISI..... v

DAFTAR GAMBAR..... viii

DAFTAR TABEL x

BAB I PENDAHULUAN..... 1

 1.1 Latar Belakang 1

 1.2 Rumusan Masalah 2

 1.3 Tujuan Penelitian..... 3

 1.4 Batasan Masalah..... 3

 1.5 Manfaat Penelitian..... 4

 1.6 Sistematika Penulisan..... 5

BAB II KAJIAN PUSTAKA DAN DASAR TEORI..... 7

 2.1 Kajian pustaka 7

 2.1.1 *Enterprise Resource Planning Software Selection For A Small Company Located In Mid-Western Wisconsin*..... 7

 2.1.2 Pelokalan dan Kustomisasi Aplikasi ERP *Open Source Openbravo* ERP Untuk Implementasi pada UKM *Furniture*..... 8

 2.2 Program Teknologi Informasi dan Ilmu Komputer (PTIIK)..... 9

 2.3 Lembaga Kedaulatan Mahasiswa Program Teknologi Informasi dan Ilmu Komputer..... 11

 2.4 HTML..... 15

 2.5 PHP..... 15

 2.6 MySQL..... 16

 2.7 *Framework* 16

 2.8 MVC (*Model, View, Controller*) 17

 2.9 Codeigniter 18

 2.10 *Extreme Programming* 18

 2.11 *Business Process Model and Notation* 19

 2.12 *Rich Picture* 20



2.13	<i>Unified Modeling Language (UML)</i>	20
2.14	<i>Class Diagram</i>	21
2.15	<i>Use Case Diagram</i>	22
2.16	<i>Activity Diagram</i>	25
2.17	<i>Sequence diagram</i>	26
2.18	<i>Functional Test</i>	28
2.19	<i>Usability Test</i>	28
2.20	<i>User Acceptance Test</i>	29
BAB III METODOLOGI PENELITIAN DAN PERANCANGAN		30
3.1	Metodologi Penelitian	30
3.1.1	Studi Literatur	31
3.1.2	Analisis Kebutuhan	31
3.1.3	Perancangan Perangkat Lunak.....	31
3.1.4	Implementasi Perangkat Lunak.....	32
3.1.5	Pengujian Perangkat Lunak.....	32
3.1.6	Pengambilan Kesimpulan.....	32
3.2	Analisis dan Perancangan.....	33
3.2.1	<i>Business Process Model and Notation</i>	33
3.2.2	<i>Rich Picture</i>	33
3.2.3	Kebutuhan Fungsional	36
3.2.4	<i>Use Case Diagram</i>	37
3.2.5	<i>Activity Diagram</i>	39
3.2.6	<i>Class Diagram</i>	43
3.2.7	<i>Sequence Diagram</i>	44
3.2.8	<i>Physical Diagram Database</i>	48
BAB IV IMPLEMENTASI		50
4.1	Batasan Implementasi.....	50
4.2	Implementasi Basis Data	50
4.2.1	Tabel Anggota Lembaga	51
4.2.2	Tabel Jadwal.....	52
4.2.3	Tabel Lembaga.....	53
4.2.4	Tabel M_departemen	53
4.2.5	Tabel M_divisi	54

4.2.6	Tabel M_jabatan.....	54
4.2.7	Tabel M_jenis_lembaga.....	54
4.2.8	Tabel M_previcege.....	55
4.2.9	Tabel Mahasiswa.....	55
4.2.10	Tabel Oprek.....	56
4.2.11	Tabel Pengajuan_lembaga.....	57
4.2.12	Tabel Peserta_oprek.....	58
4.2.13	Tabel Program_kerja.....	60
4.3	Implementasi <i>Class</i>	62
4.4	Implementasi Antar Muka.....	63
4.4.1	Halaman depan aplikasi.....	63
4.4.2	Halaman pengajuan lembaga baru.....	64
4.4.3	Halaman kelola program kerja.....	64
4.4.4	Halaman perancangan program kerja.....	65
4.4.5	Halaman kelola <i>open recruitment</i>	65
4.4.6	Halaman daftar <i>open recruitment</i>	66
4.4.7	Halaman melihat jadwal kegiatan.....	67
BAB V PENGUJIAN DAN ANALISIS.....		68
5.1	Hasil <i>Functional Test</i>	68
5.2	Hasil dan Analisa <i>Usability Test</i>	68
5.3	Hasil dan Analisa User Acceptance Test.....	70
BAB VI PENUTUP.....		74
6.1	Kesimpulan.....	74
6.2	Saran.....	74
DAFTAR PUSTAKA.....		DP-1
LAMPIRAN.....		L-1

DAFTAR GAMBAR

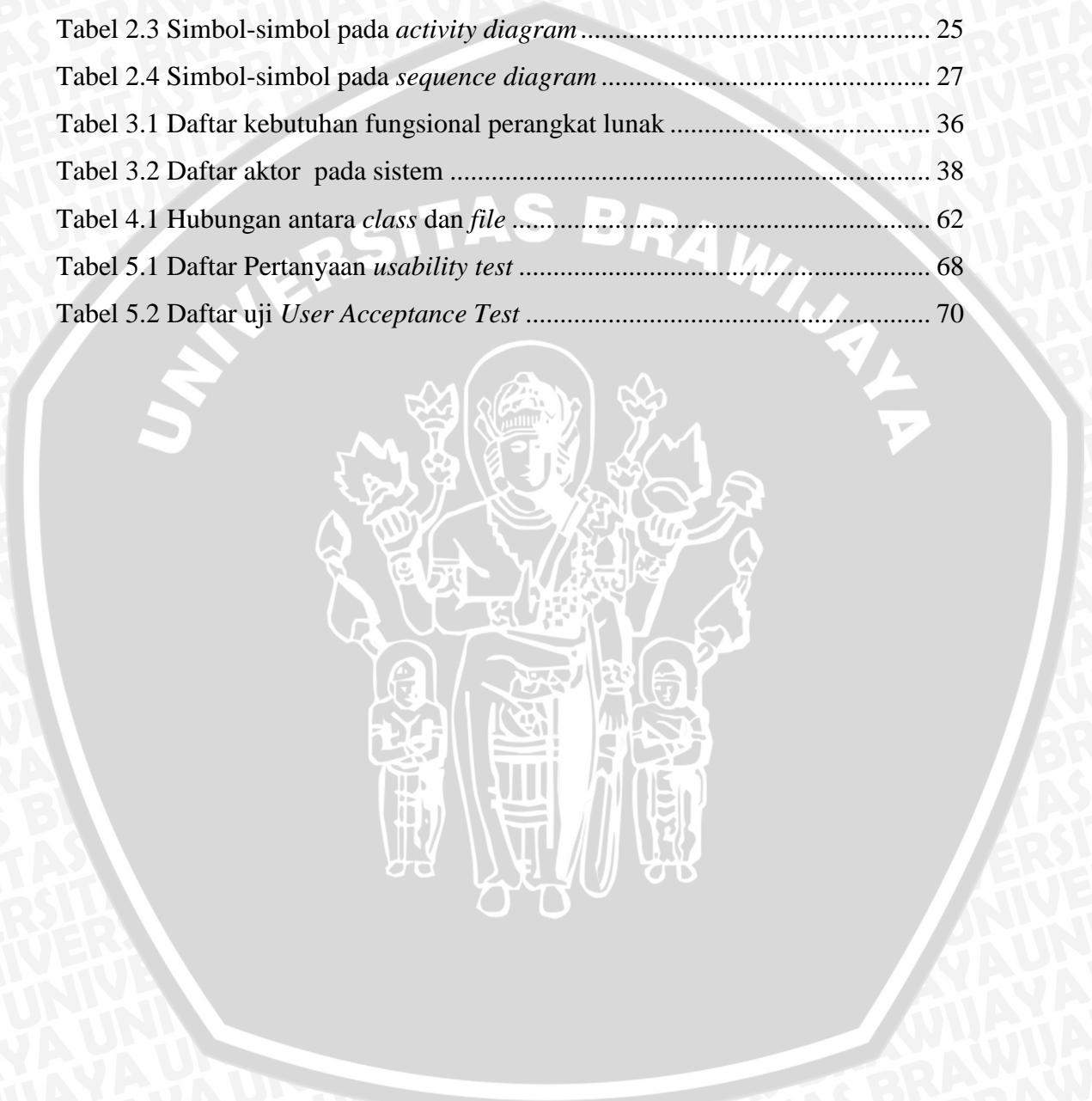
Gambar 2.1 Struktur Organisasi LKM PTIIK.....	11
Gambar 2.2 Model Hubungan MVC.....	17
Gambar 2.3 Model Proses <i>Extreme Programming</i>	18
Gambar 3.1 Diagram Alir Metode Penelitian	30
Gambar 3.2 <i>Rich Picture</i> Pengajuan Lembaga Baru	33
Gambar 3.3 <i>Rich Picture</i> Kelola Program Kerja	34
Gambar 3.4 <i>Rich Picture</i> Kelola <i>Open Recruitement</i>	35
Gambar 3.5 <i>Rich Picture</i> Menentukan Jadwal Program Kerja.....	35
Gambar 3.6 Diagram <i>Use Case</i> Aplikasi ERP LKM PTIIK.....	37
Gambar 3.7 <i>Activity diagram</i> pengajuan lembaga baru	39
Gambar 3.8 <i>Activity diagram</i> pengelolaan program kerja.....	40
Gambar 3.9 <i>Activity diagram</i> pengelolaan open recruitment	41
Gambar 3.10 <i>Activity diagram</i> melihat jadwal kegiatan	42
Gambar 3.11 <i>Class diagram</i> aplikasi ERP LKM PTIIK	43
Gambar 3.12 <i>Sequence diagram</i> mengajukan lembaga baru	44
Gambar 3.13 <i>Sequence diagram</i> mengusulkan program kerja	45
Gambar 3.14 <i>Sequence diagram</i> mengelola <i>open recruitment</i>	46
Gambar 3.15 <i>Sequence diagram</i> melihat jadwal.....	47
Gambar 3.16 <i>Physical diagram database</i> aplikasi ERP LKM PTIIK	48
Gambar 4.1 <i>Screenshot</i> tabel anggota_lembaga	51
Gambar 4.2 <i>Screenshot</i> tabel jadwal.....	52
Gambar 4.3 <i>Screenshot</i> tabel lembaga	53
Gambar 4.4 <i>Screenshot</i> tabel m_departemen.....	53
Gambar 4.5 <i>Screenshot</i> tabel m_divisi	54
Gambar 4. 6 <i>Screenshot</i> tabel m_jabatan	54
Gambar 4.7 <i>Screenshot</i> tabel m_jenis_lembaga	54
Gambar 4.8 <i>Screenshot</i> tabel m_privilege.....	55
Gambar 4.9 <i>Screenshot</i> tabel mahasiswa	55
Gambar 4.10 <i>Screenshot</i> tabel oprek	56

Gambar 4.11 <i>Screenshot</i> tabel pengajuan_lembaga.....	57
Gambar 4.12 <i>Screenshot</i> tabel peserta_oprek	58
Gambar 4.13 <i>Screenshot</i> tabel program_kerja.....	60
Gambar 4.14 Halaman depan aplikasi	63
Gambar 4.15 Halaman pengajuan lembaga baru	64
Gambar 4.16 Halaman kelola program kerja	64
Gambar 4.17 Halaman perancangan program kerja.....	65
Gambar 4.18 Halaman kelola <i>open recruitment</i>	65
Gambar 4.19 Halaman daftar <i>open recruitment</i>	66
Gambar 4.20 Halaman melihat jadwal kegiatan	67
Gambar 5.1 <i>Chart</i> Hasil <i>Usability Test</i>	70
Gambar 5.2 <i>Chart</i> Hasil <i>User Acceptance Test</i>	72



DAFTAR TABEL

Tabel 2.1 Simbol-simbol pada <i>class diagram</i>	21
Tabel 2.2 Simbol-simbol pada <i>use case</i>	23
Tabel 2.3 Simbol-simbol pada <i>activity diagram</i>	25
Tabel 2.4 Simbol-simbol pada <i>sequence diagram</i>	27
Tabel 3.1 Daftar kebutuhan fungsional perangkat lunak.....	36
Tabel 3.2 Daftar aktor pada sistem.....	38
Tabel 4.1 Hubungan antara <i>class</i> dan <i>file</i>	62
Tabel 5.1 Daftar Pertanyaan <i>usability test</i>	68
Tabel 5.2 Daftar uji <i>User Acceptance Test</i>	70



BAB I PENDAHULUAN

1.1 Latar Belakang

Program Teknologi Informasi dan Ilmu Komputer (PTIHK) merupakan sebuah program setingkat fakultas di Universitas Brawijaya yang memiliki mahasiswa sebanyak 919 mahasiswa dengan rincian 192 mahasiswa program studi sistem informasi, 156 mahasiswa program studi sistem informasi, dan 571 mahasiswa program studi informatika/ilmu komputer [JMP-14]. Setiap mahasiswa tentu saja memiliki kewajiban untuk mengembangkan kemampuannya baik kemampuan *soft skill* maupun *hard skill*. Pengembangan kemampuan *hard skill* dapat dilakukan dengan belajar, mengikuti kuliah, dan mengikuti pelatihan-pelatihan yang berkaitan dengan bidang yang diambil. Sedangkan pengembangan kemampuan *soft skill* dapat dilakukan salah satunya dengan aktif mengikuti kegiatan organisasi di lembaga mahasiswa yang ada di PTIHK.

Lembaga Kedaulatan Mahasiswa Program Teknologi Informasi dan Ilmu Komputer (LKM PTIHK) merupakan lembaga mahasiswa tingkat fakultas yang memiliki fungsi sebagai wadah mahasiswa dalam mengimplementasikan peran dan fungsi mahasiswa untuk memberikan kontribusi kepada fakultas, universitas, bangsa, dan negara. LKM PTIHK terdiri dari beberapa lembaga yaitu Dewan Perwakilan Mahasiswa, Badan Eksekutif Mahasiswa, Himpunan Mahasiswa Program Studi, dan Lembaga Semi Otonom. Masing-masing dari lembaga tersebut memiliki struktur, pengurus, dan program kerja masing-masing.

Sampai saat ini, setiap lembaga dalam melaksanakan program kerjanya sering mengalami masalah yang sama antara lain berupa waktu pelaksanaan yang sama antar lembaga, rangkap panitia maupun pengurus, dan kesulitan dalam mengetahui keaktifan dari seorang mahasiswa. Beberapa hal tersebut terjadi karena kurangnya koordinasi antar lembaga, antar mahasiswa, antara mahasiswa dengan lembaga, serta jumlah mahasiswa yang sangat banyak.

Beberapa permasalahan tersebut dapat diselesaikan dengan adanya sebuah sistem yang saling terintegrasi. *Enterprise Resource Planning* (ERP) merupakan sistem informasi terintegrasi yang digunakan untuk mendukung proses bisnis dan

manajemen sumber daya dalam suatu organisasi [DGR-10]. Dengan adanya aplikasi ERP ini diharapkan seluruh kegiatan di LKM PTIIK dapat terintegrasi dengan baik sehingga permasalahan yang sering terjadi di LKM PTIIK dapat berkurang. Aplikasi ERP ini nantinya akan memiliki 4 modul yang saling berhubungan, yaitu modul pembentukan lembaga mahasiswa, modul program kerja, modul penjadwalan program kerja, dan modul *open recruitment*.

Keempat modul tersebut diharapkan mampu mengurangi permasalahan yang ada. Modul pembentukan lembaga berfungsi untuk mengajukan pembentukan lembaga mahasiswa. Modul program kerja berfungsi untuk melakukan manajemen dan monitoring program kerja yang ada pada setiap lembaga mahasiswa PTIIK dari awal hingga program kerja selesai. Modul penjadwalan program kerja berfungsi untuk mengintegrasikan seluruh jadwal program kerja dari setiap lembaga mahasiswa PTIIK. Modul *open recruitment* berfungsi untuk melakukan seleksi dan pendataan terhadap keaktifan organisasi setiap mahasiswa PTIIK.

Perancangan aplikasi ERP ini menggunakan metode *Extreme Programming*. *Extreme Programming* merupakan modifikasi dari metode *Rapid Application Development* (RAD), pada metode ini peneliti diijinkan untuk berkomunikasi langsung dengan pengguna atau *user* ketika peneliti membutuhkan *feedback* dari *user* [SLM-13]. Sehingga diharapkan masalah proses bisnis yang timbul ketika aplikasi dikembangkan dapat segera diatasi dan aplikasi dapat diselesaikan dengan cepat. Aplikasi ERP ini nantinya akan diuji dengan menggunakan metode *functional test*, *usability test* dan *user acceptance test*.

1.2 Rumusan Masalah

Berdasarkan pada permasalahan yang telah diuraikan pada latar belakang sebelumnya, maka rumusan masalah dalam penelitian ini dapat disusun, sebagai berikut:

1. Bagaimana melakukan analisis proses bisnis pada proses pembentukan lembaga mahasiswa, perancangan program kerja, penjadwalan program kerja dan *open recruitment* di LKM PTIIK?

2. Bagaimana merancang dan mengimplementasikan aplikasi *Enterprise Resource Planning* untuk LKM PTIIK?
3. Bagaimana menguji aplikasi *Enterprise Resource Planning* pada LKM PTIIK?

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut:

1. Mengetahui proses bisnis pada proses pembentukan lembaga mahasiswa, perancangan program kerja, penjadwalan program kerja dan *open recruitment* di LKM PTIIK.
2. Melakukan dan mengambil kesimpulan hasil perancangan dan implementasi aplikasi *Enterprise Resource Planning* untuk LKM PTIIK.
3. Mengetahui hasil pengujian aplikasi *Enterprise Resource Planning* pada LKM PTIIK.

1.4 Batasan Masalah

Agar permasalahan yang dirumuskan lebih fokus, maka ruang lingkup aspek kajian dalam penelitian ini diberi batasan, sebagai berikut:

1. Aplikasi yang dibuat berupa *Web Application*.
2. Aplikasi dibuat menggunakan bahasa pemrograman PHP versi 5.5.6.
3. Aplikasi dibuat menggunakan *Framework Codeigniter* versi 2.1.4.
4. Aplikasi ini memiliki 4 (empat) modul yaitu modul pembentukan lembaga mahasiswa, modul program kerja, modul penjadwalan program kerja, dan modul *open recruitment*.
5. Fitur-fitur yang terdapat di dalam modul yaitu:
 - a. modul pembentukan lembaga mahasiswa
 - mengajukan lembaga baru
 - melihat detail pengajuan lembaga baru
 - persetujuan lembaga baru
 - b. modul program kerja

- menambah program kerja
 - persetujuan program kerja
 - mengedit program kerja
 - menghapus rancangan program kerja
 - mengubah *progress* program kerja
 - melihat progress program kerja
 - mendelegasikan program kerja
- c. modul penjadwalan program kerja
- menentukan jadwal program kerja
 - mengecek jadwal bentrok
 - mengedit jadwal program kerja
 - melihat jadwal program kerja
- d. modul *open recruitment*
- membuat *open recruitment*
 - mendaftar pada *open recruitment*
 - melakukan seleksi peserta *open recruitment*
 - mengumumkan hasil *open recruitment*
 - melihat hasil *open recruitment*
6. Aplikasi ini tidak dapat melakukan penggabungan 2 (dua) atau lebih lembaga yang berbeda.
7. Aplikasi ini tidak dapat melakukan penghapusan sebuah lembaga yang telah terbentuk.
8. Aplikasi ini khusus untuk studi kasus di Lembaga Kedaulatan Mahasiswa Program Teknologi Informasi dan Ilmu Komputer (LKM PTIIK).

1.5 Manfaat Penelitian

Hasil penelitian ini diharapkan dapat memberikan manfaat bagi semua pihak. Adapun manfaat dari penelitian ini, adalah sebagai berikut:

1. Bagi penulis:

- a. Dapat mengaplikasikan ilmu yang didapat selama mengikuti perkuliahan di Program Studi Informatika Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
 - b. Mendapatkan pemahaman dalam perancangan dan penerapan aplikasi *Enterprise Resource Planning* pada suatu organisasi.
2. Bagi pengguna:
- a. Dapat mempermudah pengguna dalam pembentukan lembaga kemahasiswaan, perancangan program kerja, penjadwalan program kerja, dan *open recruitment* kegiatan mahasiswa.
 - b. Dapat mengetahui kegiatan apa saja yang direncanakan oleh LKM PTIIK serta mahasiswa yang terlibat pada kegiatan tersebut.
 - c. Setiap LKM PTIIK dapat melakukan manajemen dan monitoring program kerja dengan baik.
 - d. Manajemen pembagian sumber daya mahasiswa dapat dilakukan dengan baik.

1.6 Sistematika Penulisan

Skripsi ini disusun berdasarkan sistematika penulisan sebagai berikut:

BAB I Pendahuluan

Memuat latar belakang disusunnya penulisan penelitian ini, rumusan masalah tentang beberapa masalah yang ingin diselesaikan, tujuan dari penelitian, batasan masalah agar penelitian tidak melebar pada bidang lain dan fokus penelitian tetap pada masalah yang ingin diselesaikan, manfaat penelitian, dan sistematika penulisan.

BAB II Kajian Pustaka dan Dasar Teori

Bab ini membahas kajian dari penelitian yang pernah dilakukan sebelumnya serta berisi teori-teori dasar dan teori penunjang untuk menunjang penyelesaian masalah dalam penelitian ini.

BAB III Metode Penelitian Dan Perancangan

Bab ini membahas metode yang digunakan dalam penelitian ini serta analisis kebutuhan dan perancangan aplikasi *Enterprise Resource Planning* di Lembaga Kedaulatan Mahasiswa Program Teknologi Informasi dan Ilmu Komputer.

BAB IV Implementasi

Membahas tentang implementasi sistem.

BAB V Pengujian dan Analisis

Pada bab ini akan dilakukan pengujian perangkat lunak dan analisis perangkat lunak yang dibangun.

BAB VI Penutup

Memuat kesimpulan serta saran yang diperoleh dari pembuatan dan pengujian aplikasi untuk pengembangan lebih lanjut.



BAB II

KAJIAN PUSTAKA DAN DASAR TEORI

Pada bab ini akan diuraikan mengenai kajian pustaka serta teori-teori dasar pembuatan aplikasi *Enterprise Resource Planning* (ERP) pada Lembaga Kedaulatan Mahasiswa Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya Malang.

2.1 Kajian pustaka

Sub bab ini akan menjelaskan mengenai kajian pustaka yang digunakan untuk mendukung penelitian ini. Terdapat 2 (dua) kajian pustaka yang digunakan pada penelitian ini yaitu kajian pustaka dengan judul *Enterprise Resource Planning Software Selection For A Small Company Located In Mid-Western Wisconsin* dan Pelokalan dan Kustomisasi Aplikasi ERP *Open Source* Openbravo ERP Untuk Implementasi pada UKM *Furniture*.

2.1.1 *Enterprise Resource Planning Software Selection For A Small Company Located In Mid-Western Wisconsin*

Penelitian yang dibahas adalah "*Enterprise Resource Planning Software Selection For A Small Company Located In Mid-Western Wisconsin*" oleh Sia Thao. Penelitian ini memiliki tujuan akhir untuk memberikan rekomendasi sistem ERP yang cocok untuk digunakan oleh perusahaan yang berskala kecil. Hal ini dikarenakan sistem ERP tidak hanya diperuntukkan bagi perusahaan besar saja tetapi perusahaan dengan skala yang kecil juga dapat menggunakannya. Nama dari obyek perusahaan disini dirahasiakan dan diganti dengan nama perusahaan ABC. Perusahaan ABC terletak di Mid – Wisconsin Barat dan hanya memiliki 55 orang karyawan. Perusahaan ini merupakan perusahaan kecil yang sudah diwariskan empat generasi berturut-turut. Perusahaan ini sebelumnya menggunakan *Effective Management System* (EMS). Sistem ini dirasa masih kurang efisien karena masih memisahkan dua sistem yang ada, yaitu sistem pada departemen teknik dan departemen sistem informasi sehingga ingin beralih ke sistem ERP. Hal ini disebabkan karena system ERP menekankan pada integrasi sehingga dapat

membuat sistem perusahaan menjadi lebih efisien dan efektif. Kesimpulan dari penelitian ini adalah sistem ERP yang dijual dipasaran tidak ada yang sesuai 100% dengan kondisi perusahaan ABC berdasarkan fitur dan harganya. Selain itu dapat disimpulkan pula bahwa aplikasi Enterprise Resource Planning tidak hanya dibutuhkan oleh perusahaan dengan skala besar. Begitu juga dengan penelitian yang penulis lakukan yaitu mengambil studi kasus di LKM PTIIK, dimana LKM PTIIK merupakan sebuah organisasi yang kecil.

2.1.2 Pelokalan dan Kustomisasi Aplikasi ERP *Open Source* Openbravo ERP Untuk Implementasi pada UKM *Furniture*

Penelitian yang dibahas adalah “Pelokalan dan Kustomisasi Aplikasi ERP *Open Source* Openbravo ERP Untuk Implementasi pada UKM *Furniture*” oleh M. Hanif Fahmi. Penelitian ini membahas mengenai penerapan sebuah aplikasi ERP *Open Source* pada Usaha Kecil Menengah (UKM) *Furniture* untuk meningkatkan efisiensi dan efektifitas UKM agar dapat bersaing secara global dalam menghadapi *China – ASEAN Trade Agreement* (CAFTA). Openbravo ERP adalah aplikasi yang dipilih untuk penerapannya karena aplikasi ini gratis dan dapat dikustomisasi. Pelokalan juga dilakukan pada aplikasi ini supaya mudah untuk dipahami oleh pengguna yang kurang menguasai bahasa inggris. Kesimpulan dari penelitian ini adalah bahwa ERP juga dibutuhkan oleh UKM untuk meningkatkan daya saing, tetapi yang menjadi kendala adalah aplikasi ERP yang ada masih terlalu mahal, berbahasa inggris dan memiliki struktur yang terlalu rumit. Oleh sebab itu untuk mengatasi kendala tersebut dapat digunakan aplikasi ERP yang *open source* seperti Openbravo ERP dengan kustomisasi untuk menyesuaikan dengan kondisi UKM dan pelokalan untuk mengubah bahasa inggris menjadi bahasa Indonesia agar mudah dimengerti. Pada penelitian yang penulis lakukan aplikasi ERP yang dibuat tidak menggunakan aplikasi ERP yang telah ada sebelumnya sehingga diharapkan aplikasi ERP yang dibuat pada penelitian ini akan sesuai dengan kebutuhan pada LKM PTIIK.

2.2 Program Teknologi Informasi dan Ilmu Komputer (PTIIK)

Program Teknologi Informasi dan Ilmu Komputer (PTIIK) merupakan sebuah program setingkat fakultas di Universitas Brawijaya yang berfokus pada bidang teknologi informasi. PTIIK memiliki 3 program studi yaitu Program Studi Informatika/Ilmu Komputer, Program Studi Sistem Komputer, dan Program Studi Sistem Informasi. PTIIK memiliki mahasiswa sebanyak 919 mahasiswa dengan rincian 192 mahasiswa Program Studi Sistem Informasi, 156 mahasiswa Program Studi Sistem Komputer, dan 571 mahasiswa Program Studi Informatika/Ilmu Komputer [JMP-14].

PTIIK memiliki visi menjadi pusat keunggulan (*centre of excellence*) *entrepreneur* pendidikan tinggi dan pengembangan ilmu pengetahuan di bidang teknologi informasi dan komputer di tingkat nasional dan internasional (*world class*) melalui integrasi tri darma perguruan tinggi [VMT-14]. Sedangkan misi dari PTIIK adalah sebagai berikut [VMT-14]:

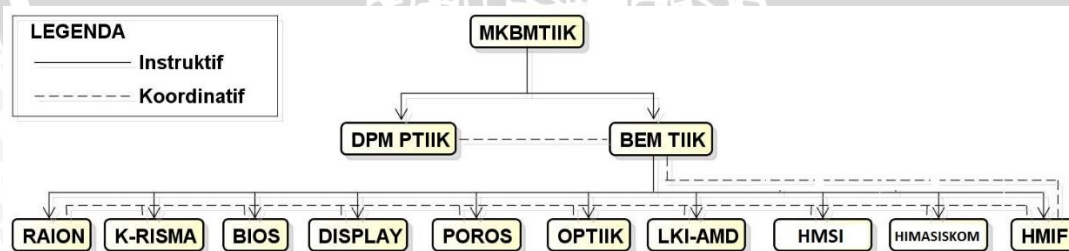
1. Menghasilkan lulusan yang memiliki kompetensi di bidang TIK, berjiwa *entrepreneur* dan dapat dipercaya sehingga mampu bekerjasama dan memberikan kontribusi di tingkat nasional dan internasional (*world class*).
2. Mengembangkan ilmu pengetahuan dan teknologi bidang informatika dan computer melalui integrasi tri darma perguruan tinggi dengan mengedepankan moral dan etika serta didukung oleh pengembangan sumberdaya berkelanjutan.
3. Memberikan pelayanan kepada masyarakat (*stakeholders*) melalui pendidikan, penelitian dan pengabdian masyarakat di tingkat nasional dan internasional.
4. Meningkatkan kontribusi dan kolaborasi (kerjasama) dengan berbagai pihak dengan mengembangkan produk hasil inovasi dan kreasi bidang informatika dan komputer di tingkat nasional maupun internasional.

Tujuan dari PTIIK adalah sebagai berikut [VMT-14]:

1. Menghasilkan lulusan berkualifikasi sebagai berikut:
 - Berjiwa Pancasila, memiliki integritas kepribadian yang tinggi dan berjiwa enterpreneur.
 - Bersifat terbuka, tanggap terhadap perubahan dan kemajuan ilmu dan teknologi serta dinamika perubahan sosial dan kemasyarakatan, khususnya yang berkaitan dengan bidang keahliannya.
 - Mempunyai kemampuan untuk menerapkan pengetahuan serta ketrampilan teknologi yang dimilikinya.
 - Menguasai dasar-dasar ilmiah serta pengetahuan dan metodologi sehingga mampu menemukan, memahami, menjelaskan dan merumuskan cara penyelesaian yang ada di dalam kawasan keahliannya.
 - Menguasai dasar-dasar ilmiah sehingga mampu berpikir, bersikap dan bertindak sebagai ilmuwan.
 - Mampu mengikuti perkembangan ilmu pengetahuan dan ketrampilan sesuai dengan bidangnya.
2. Menghasilkan penelitian yang dapat memperkaya khasanah keilmuan dengan menemukan konsep, model, dan paradigma baru di bidang informatika dan komputer yang berbasis pada moral dan etika dalam rangka mengisi dan menunjang pembangunan regional maupun nasional.
3. Melakukan pengabdian masyarakat dalam bentuk pembinaan, bimbingan dan konsultasi dalam rangka meningkatkan peran serta masyarakat dalam pembangunan serta meningkatkan kerjasama dan kolaborasi dengan berbagai pihak dengan mengembangkan beragam produk hasil inovasi dan kreasi dibidang informatika dan komputer.
4. Melakukan pengembangan sertifikasi kompetensi dibidang informatikan dan komputer ditingkat regional, nasional maupun internasional.

2.3 Lembaga Kedaulatan Mahasiswa Program Teknologi Informasi dan Ilmu Komputer

Lembaga Kedaulatan Mahasiswa (LKM) adalah organisasi mahasiswa yang memiliki fungsi sebagai wadah mahasiswa dalam mengimplementasikan peran dan fungsi mahasiswa untuk memberikan kontribusi kepada jurusan, fakultas, universitas, bangsa, dan negara. Masing-masing fakultas ataupun program setingkat fakultas di setiap universitas pasti memiliki LKM. Begitu pula dengan PTIIK, LKM PTIIK merupakan kumpulan lembaga-lembaga mahasiswa yang ada di PTIIK. Sampai dengan skripsi ini disusun, LKM PTIIK terdiri dari Musyawarah Keluarga Besar Mahasiswa Teknologi Informasi dan Ilmu Komputer (MKBMTIIK), Dewan Perwakilan Mahasiswa PTIIK (DPM PTIIK), Badan Eksekutif Mahasiswa Teknologi Informasi dan Ilmu Komputer (BEM TIIK), Himpunan Mahasiswa Informatika (HMIF), Himpunan Mahasiswa Sistem Informasi (HMSI), Himpunan Mahasiswa Sistem Komputer (HIMASISKOM), Lembaga Pers Mahasiswa Display, Raion Studio, Badan Internal Olahraga dan Seni (BIOS), Kelompok Riset Mahasiswa (K-RISMA), OPTIIK, PTIIK *Organization of Open Source* (POROS), Lembaga Kajian Islam Al-Fatih Muslim Drenalin (LKI-AMD).



Gambar 2.1 Struktur Organisasi LKM PTIIK

Sesuai dengan Gambar 2.1, MKBMTIIK memiliki kedudukan tertinggi pada struktur LKM PTIIK. Musyawarah ini dipimpin oleh tiga orang presidium yang memiliki tugas untuk memastikan musyawarah berjalan dengan tertib dan kondusif. MKBMTIIK dihadiri oleh beberapa perwakilan dari mahasiswa tiap angkatan pada setiap program studi dan perwakilan seluruh lembaga mahasiswa

yang termasuk di dalam LKM PTIIK. MKBMTIIK memiliki jalur instruktif ke seluruh lembaga mahasiswa yang berada dibawahnya, sehingga hasil dari musyawarah ini harus diikuti oleh seluruh lembaga mahasiswa tersebut.

DPM PTIIK merupakan lembaga mahasiswa yang memiliki fungsi sebagai badan legislatif di LKM PTIIK. DPM PTIIK memiliki tugas untuk menampung aspirasi mahasiswa PTIIK dan menyalurkannya sesuai dengan tujuan dari aspirasi tersebut. Selain itu DPM PTIIK juga memiliki tugas untuk melakukan pengawasan terhadap kinerja dari badan eksekutif di LKM PTIIK, yaitu BEM TIIK. DPM PTIIK memiliki jalur koordinatif dengan BEM TIIK, sehingga DPM PTIIK dapat melakukan koordinasi dengan BEM TIIK dalam mengambil keputusan.

BEM TIIK merupakan lembaga mahasiswa yang memiliki fungsi sebagai badan eksekutif di LKM PTIIK. BEM TIIK memiliki tugas untuk melakukan eksekusi terhadap aspirasi dari mahasiswa PTIIK. Selain itu BEM TIIK memiliki tugas untuk mengembangkan kemampuan mahasiswa dalam bidang akademik maupun non akademik. Oleh karena itu, hingga skripsi ini ditulis BEM TIIK menaungi 3 lembaga mahasiswa program studi dan 7 Lembaga Semi Otonom (LSO) yang memiliki tugas lebih spesifik untuk meningkatkan kemampuan mahasiswa. BEM TIIK memiliki jalur instruktif ke seluruh LSO, sehingga BEM TIIK mempunyai wewenang untuk memberikan instruksi kepada seluruh LSO dan seluruh LSO harus menjalankan perintah tersebut. Berbeda dengan BEM TIIK terhadap LSO, BEM TIIK terhadap seluruh Himpunan Mahasiswa Program Studi (HIMAPRODI), BEM TIIK memiliki jalur instruktif koordinatif sehingga BEM TIIK dapat memberikan instruksi kepada HIMAPRODI selama instruksi tersebut tidak berhubungan dengan kedaulatan mahasiswa program studi, sedangkan untuk hal-hal yang berhubungan dengan kedaulatan mahasiswa program studi harus dilakukan koordinasi antara BEM TIIK dengan HIMAPRODI yang terkait.

Lembaga mahasiswa program studi yang ada di PTIIK saat ini adalah Himpunan Mahasiswa Informatika (HMIF), Himpunan Mahasiswa Sistem Informasi (HMSI), dan Himpunan Mahasiswa Sistem Komputer (HIMASISKOM). HMIF terdiri dari EMIF (Eksekutif Mahasiswa Informatika) yang memiliki fungsi sebagai lembaga eksekutif dan BPMIF (Badan Perwakilan Mahasiswa Informatika)

yang memiliki fungsi sebagai lembaga 13ctor13io13ve bagi mahasiswa Program Studi Informatika/Illmu Komputer. HMSI terdiri dari EMSI (Eksekutif Mahasiswa Sistem Informasi) yang memiliki fungsi sebagai lembaga eksekutif dan BPMSI (Badan Perwakilan Mahasiswa Sistem Informasi) yang memiliki fungsi sebagai lembaga legislatif bagi mahasiswa Program Studi Sistem Informasi. HIMASISKOM terdiri dari EMSK (Eksekutif Mahasiswa Sistem Komputer) yang memiliki fungsi sebagai lembaga eksekutif dan FPASK (Forum Perwakilan Angkatan Sistem Komputer) yang memiliki fungsi sebagai lembaga legislatif bagi mahasiswa Program Studi Sistem Komputer.

LKM PTIIK memiliki 7 Lembaga Semi Otonom (LSO) yang berada dibawah naungan BEM TIIK yaitu:

1. Lembaga Pers Mahasiswa Display (LPM Display) memiliki tugas untuk meningkatkan kemampuan mahasiswa di bidang pers dan jurnalistik.
2. Raion Studio bergerak dibidang pengembangan akademik mahasiswa, khususnya dalam bidang *mobile application*.
3. Badan Internal Seni dan Olahraga (BIOS) memiliki tugas untuk mengembangkan minat dan bakat mahasiswa di bidang olahraga dan seni.
4. Kelompok Riset Mahasiswa (K-RISMA) memiliki tugas untuk membantu dan mengembangkan mahasiswa dalam bidang riset dan karya ilmiah.
5. OPTIIK memiliki tugas untuk mengembangkan minat dan bakat mahasiswa di bidang fotografi dan desain.
6. PTIIK Organization of Open Source (POROS) memiliki tugas untuk melakukan sosialisasi serta mengembangkan minat dan bakat mahasiswa terhadap penggunaan *Open Source Software*.
7. Lembaga Kajian Islam Al-Fatih Muslim Drenalin (LKI-AMD) memiliki tugas untuk melakukan dakwah dan pengkajian terhadap ajaran islam.

2.1. Enterprise Resource Planning

Enterprise Resource Planning (ERP) merupakan sistem informasi terintegrasi yang digunakan untuk mendukung proses bisnis dan manajemen sumber daya dalam suatu organisasi [DGR-10]. Berikut ini merupakan sejarah perkembangan ERP, yaitu :

1. Berawal dari *Material Resource Planning* (MRP) pada tahun 1960-an. Saat itu MRP hanya dapat menangani masalah *Inventory Management* dan *Production*.
2. Dari MRP berkembang menjadi *Close Loop MRP* pada tahun 1970-an. Pada *Close Loop MRP* terdapat penambahan fungsional system, yaitu dapat menangani masalah *Engineering*.
3. *Close Loop MRP* kemudain berkembang menjadi *Manufacturing Resource Planning* (MRP II) pada tahun 1980-an. Pada ERP II ditambahkan fungsi untuk menangani masalah *Finance* dan *Human Resource Management*.
4. MRP II berkembang menjadi *Enterprise Resource Planning* (ERP) pada tahun 1990-an. Pada ERP ditambahkan fungsi untuk menangani masalah *Delivery*.
5. Untuk menyesuaikan dengan kebutuhan, ERP berkembang menjadi *Extended ERP* (ERP II) pada tahun 2000-an. Pada ERP II ditambahkan fungsi untuk menangani masalah *Supplier Management*, *Sales & Marketing*, dan *Customer Support*.

Konsep dasar ERP menggunakan *client server computing*. Oleh sebab itu ERP memiliki kaitan yang sangat erat dengan *networking*. Untuk pengolahan data pada ERP menggunakan *relational database*. Pada sistem ERP, peng-input-an sebuah informasi hanya dilakukan sekali pada sistem. Kemudian informasi tersebut diolah dan hasil pengolahannya dapat digunakan oleh seluruh bagian sistem yang membutuhkan. Sebuah *software* ERP terdiri terdiri dari beberapa modul yang saling terintegrasi. Setiap modul pada ERP berfokus pada satu proses bisnis seperti marketing atau sumber daya manusia. Modul-modul pada ERP saling terintegrasi

secara *real time* sehingga data yang ditampilkan adalah data terbaru. Sistem ERP memiliki manfaat ketika diterapkan, antara lain [ODE-00]:

- a. Mampu mengintegrasikan kegiatan pada suatu organisasi
- b. Membuat sebuah standarisasi pada sebuah proses dalam suatu organisasi
- c. Dapat mengurangi informasi yang tidak seimbang
- d. Menyediakan informasi yang *real-time* dan dapat dilihat kapan saja
- e. Dapat memfasilitasi kolaborasi antar organisasi sehingga kolaborasi dapat dilakukan dengan baik dan mudah.

Oleh sebab itu, sistem ERP sangat penting dan dibutuhkan bagi sebuah perusahaan atau organisasi. Sistem ini dapat memberikan dampak yang luar biasa bagi perusahaan terutama dalam melakukan manajemen, pengawasan, dan perencanaan sehingga tercipta perusahaan yang lebih kompetitif.

2.4 HTML

HTML (*Hyper Text Markup Language*) adalah sekumpulan symbol-simbol atau tag-tag yang dituliskan dalam sebuah file yang digunakan untuk menampilkan halaman pada web browser [ANH-10:40]. HTML ditemukan oleh Tim Berners-Lee pada tahun 1989, merupakan dasar framework Internet. Hampir tiap situs Web yang ada menggunakan bahasa HTML untuk menampilkan teks, grafik, suara, dan animasi. Pihak yang menjadi caretaker atas spesifikasi tersebut adalah World Wide Web Consortium (W3C) (<http://www.w3.org>).

2.5 PHP

PHP merupakan singkatan dari PHP: *Hypertext Preprocessor* yaitu bahasa pemrograman web *server-side* yang bersifat *open source* [ANH-10:3]. PHP merupakan script yang terintegrasi dengan HTML dan berada pada server. Bahasa pemrograman ini digunakan untuk membuat halamn web menjadi dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh pengguna web. Mekanisme ini membuat informasi yang ditampilkan selalu informasi yang terbaru.

Beberapa keuntungan penggunaan PHP adalah [ANH-10:3]:

- Mudah dipahami karena memiliki banyak referensi.
- Bersifat open source dan dapat digunakan di berbagai macam system operasi seperti Linux, Unix, Windows, dan Macintosh.
- Web Server yang menggunakan PHP mudah ditemukan dimana-mana.
- PHP dilengkapi dengan berbagai macam pendukung lain seperti dukungan langsung ke berbagai macam database populer.

2.6 MySQL

MySQL (My Structure Query Language) adalah salah satu Database Management System (DBMS) dari sekian banyak DBMS seperti Oracle, MS SQL, Postgre SQL, dan lainnya. MySQL merupakan program yang digunakan untuk mengolah dan mengelola database. Struktur dan prosedur yang dimiliki MySQL dibuat sedemikian rupa sehingga mempermudah dalam menyimpan, mengatur dan menampilkan data yang dibutuhkan [ANH-10:45].

2.7 Framework

Framework secara sederhana dapat diartikan kumpulan perintah atau fungsi fungsi/prosedur-prosedur standar dan *class-class* yang membentuk aturan tertentu dan saling berinteraksi satu sama lain sehingga dalam pembuatan aplikasi harus mengikuti aturan dari *framework* yang digunakan [WRD-10:3]. Penggunaan *framework* bisa lebih mempermudah dan mempercepat pekerjaan seorang programmer, tanpa harus membuat fungsi atau *class* dari awal.

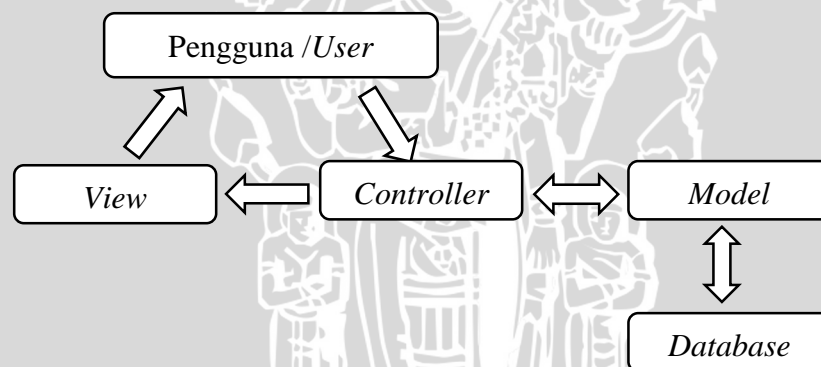
Ada beberapa alasan mengapa menggunakan *Framework* [WRD-10:3]:

- Mempercepat dan mempermudah pembangunan sebuah aplikasi web.
- Kode aplikasi lebih mudah dimengerti oleh *programmer*.
- Relatif memudahkan dalam proses maintenance karena sudah ada pola tertentu dalam sebuah *framework* (dengan syarat programmer mengikuti pola standar yang ada)

- Umumnya *framework* menyediakan fasilitas-fasilitas yang umum dipakai sehingga kita tidak perlu membangun dari awal seperti koneksi *database*, validasi form, dan keamanan.
- *Programmer* lebih fokus ke alur permasalahan aplikasi, tampilan aplikasi, dan layanan yang ada pada aplikasi.
- Ketika aplikasi dikerjakan secara tim akan lebih terarah karena dengan *framework* mengharuskan adanya keteraturan peletakan kode.

2.8 MVC (*Model, View, Controller*)

Model-View-Controller (MVC) adalah sebuah pola pemrograman yang bertujuan memisahkan logika bisnis, logika data, dan logika tampilan [WRD-10:52].



Gambar 2.2 Model Hubungan MVC

Sumber : [WRD-10:53]

Seperti terlihat pada Gambar 2.2 terdapat 3 jenis komponen yang membangun suatu MVC dalam suatu aplikasi yaitu [WRD-10:53]:

View, merupakan komponen dari MVC yang bertugas menampilkan apa yang harus ditampilkan ke pengguna. *View* mengatur bagaimana suatu data yang diperoleh dari controller ditampilkan kepada pengguna/*user*.

Model, merupakan komponen MVC yang bertugas mengambil data dari *database* dan juga memasukkan data ke *database*. Isi utamanya adalah perintah SQL. Hasil dari perintah SQL yang dijalankan dikirimkan ke *controller*.

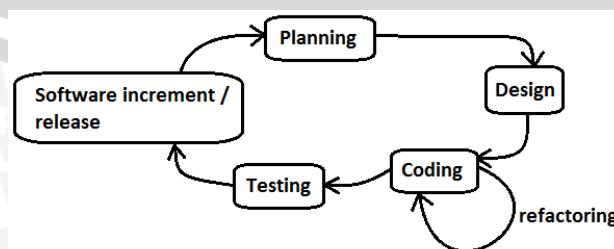
Controller, merupakan komponen dari MVC yang bertugas mengirim perintah ke model untuk mendapatkan data yang diinginkan kemudian mengolahnya dan mengirimkan data tersebut ke *view* untuk ditampilkan ke pengguna/*user*. Controller juga memiliki fungsi untuk mengolah data baik dari inputan pengguna/*user* maupun dari *model* sesuai dengan yang diinginkan.

Dengan menggunakan prinsip MVC suatu aplikasi dapat dikembangkan sesuai dengan kemampuan pengembangnya, yaitu *programmer* yang menangani bagian *model* dan *controller*, sedangkan *designer* yang menangani bagian *view*, sehingga penggunaan arsitektur MVC dapat meningkatkan *maintanability* dan organisasi kode. Walaupun demikian dibutuhkan komunikasi yang baik antara *programmer* dan *designer* dalam menangani variabel-variabel yang akan ditampilkan.

2.9 Codeigniter

Codeigniter adalah aplikasi *open source* berupa *framework* dengan model MVC (*Model, View, Controller*) untuk membangun sebuah aplikasi berbasis web dengan menggunakan bahasa pemrograman PHP [SMG-09]. Codeigniter dapat membantu para pengembang aplikasi untuk membuat aplikasi web dengan cepat tanpa harus membuatnya dari awal. CodeIgniter dirilis pertama kali pada 28 Februari 2006.

2.10 Extreme Programming



Gambar 2.3 Model Proses *Extreme Programming*
Sumber : [RSP-10]

Extreme Programming merupakan modifikasi dari metode *Rapid Application Development* (RAD), pada metode ini peneliti diijinkan untuk berkomunikasi langsung dengan pengguna atau *user* ketika peneliti membutuhkan *feedback* dari *user* [SLM-13:37]. Gambar 2.3 merupakan proses yang terdapat pada *extreme programming*. Proyek pertama yang menggunakan model pengembangan ini dimulai pada tahun 1996. Ciri khas dari model pengembangan ini adalah komunikasi yang dilakukan setiap hari atau setiap ditemukan hal-hal yang kurang jelas dari perangkat lunak yang akan dikembangkan. *Extreme Programming* sudah terbukti sukses di banyak perusahaan dari semua ukuran yang berbeda di seluruh dunia. Keberhasilan *Extreme Programming* disebabkan karena model pengembangan ini menekankan pada kepuasan pelanggan serta komunikasi yang intens antara pengembang dengan pihak terkait begitu ditemukan hal-hal yang kurang jelas [WLD-13]. Beberapa nilai dasar yang terdapat pada *Extreme Programming*, yaitu:

1. Komunikasi (*Communication*), *Extreme Programming* sangat memperhatikan komunikasi antar setiap *stakeholder*, baik antar anggota tim maupun pihak *client*.
2. Kesederhanaan (*Simplicity*), *Extreme Programming* menerapkan prinsip kesederhanaan, terutama pada saat melakukan *coding*.
3. Umpan Balik (*Feedback*), umpan balik sangat diandalkan pada *Extreme Programming* sehingga perbaikan dapat dilakukan dengan secepat mungkin.
4. Keberanian (*Courage*), *Extreme Programming* menerapkan keberanian untuk tetap memiliki integritas pada setiap anggota tim agar aplikasi dapat selesai dengan baik.

2.11 Business Process Model and Notation

Business Process Model and Notation (BPMN) merupakan salah satu standarisasi internasional dalam pemodelan untuk memodelkan sebuah proses bisnis. BPMN dikembangkan oleh *The Business Process Management Initiative*

(BPMI), BPMN 1.0 pertama kali dirilis ke masyarakat pada bulan Mei 2004. Tujuan utama dibuat BPMN adalah untuk menyediakan sebuah notasi standar yang mudah dimengerti oleh semua orang [WSA-04]. Sampai saat ini BPMN udah mencapai versi 2.0. BPMN 2.0 dirilis pada Januari 2011.

2.12 *Rich Picture*

Rich Picture adalah gambar yang merepresentasikan sebuah situasi dan digambarkan dalam grup situasi dimana didalamnya terdapat 20 actor yang terlibat beserta hubungannya [GDC-10]. *Rich Picture* digunakan untuk menggambarkan sebuah proses yang terjadi pada sebuah sistem sehingga dapat dimengerti dengan mudah oleh orang yang melihatnya.

2.13 *Unified Modeling Language (UML)*

Pemodelan adalah gambaran dari realita yang simple dan dituangkan dalam bentuk pemetaan dengan aturan tertentu [SLM-13:135]. Pemodelan dapat menggunakan bentuk yang sama dengan realitas. Pada perkembangan teknik pemrograman berorientasi objek, muncul sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modeling Language (UML)*.

UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari system perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung [SLM-13:137].

Pada UML 2.3 terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori, yaitu [SLM-13:140]:

1. *Structure diagrams*

Structure diagrams yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.

Diagram yang termasuk *structure diagrams* adalah *class diagram*,

object diagram, component diagram, composite structure diagram, package diagram, dan deployment diagram.

2. *Behavior diagrams*

Behavior diagrams yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan system atau rangkaian perubahan yang terjadi pada sebuah sistem. Diagram yang termasuk *behavior diagrams* adalah *use case diagram, activity diagram, dan state machine diagram*

3. *Interaction diagrams*

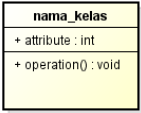
Interaction diagrams yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem. Diagram yang termasuk *interaction diagrams* adalah *sequence diagram, communication diagram, timing diagram, dan interaction overview diagram*

2.14 Class Diagram


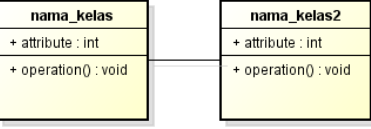
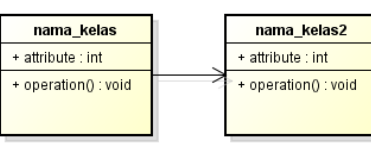
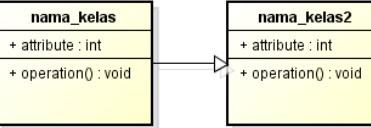
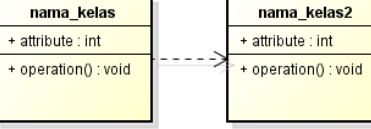
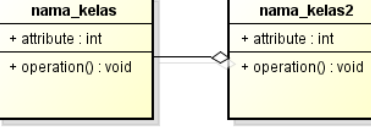
Class diagram merupakan pemodelan yang menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem [SLM-13:141]. Setiap *class* memiliki atribut dan metode. Atribut merupakan variable-variabel yang dimiliki suatu *class*. Metode merupakan fungsi-fungsi yang dimiliki oleh suatu *class*. *Class diagram* dibuat agar pembuat program membuat *class-class* yang sesuai rancangan di dalam *class diagram* agar antara dokumentasi dan perangkat lunak sinkron.

Tabel 2.1 menunjukkan simbol-simbol yang ada pada *class diagram*:

Tabel 2.1 Simbol-simbol pada *class diagram*

Simbol	Deskripsi
	Kelas/ <i>Class</i> pada struktur system
<i>Interface</i>	Sama dengan konsep interface pada pemrograman berorientasi objek



	
<p>Asosiasi/association</p> 	<p>Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i></p>
<p>Asosiasi berarah/directed association</p> 	<p>Relasi antar kelas dengan makna yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i></p>
<p>Generalisasi</p> 	<p>Relasi antar kelas dengan makna generalisasi - spesialisasi (umum-khusus)</p>
<p>Kebergantungan/dependency</p> 	<p>Relasi antar kelas dengan makna kebergantungan antar kelas</p>
<p>Agregasi/aggregation</p> 	<p>Relasi antar kelas dengan makna semua bagian (whole-part)</p>

Sumber : [SLM-13:146]



2.15 Use Case Diagram

Use case diagram merupakan pemodelan untuk memodelkan kelakuan (*behavior*) dari sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah system dan

siapa saja yang berhak menggunakan fungsi-fungsi itu. Syarat penamaan pada *use case* adalah nama didefinisikan sesederhana mungkin dan mudah dipahami. Ada dua hal utama dalam membuat sebuah *use case* yaitu pendefinisian aktor dan *use case*. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem yang akan dibuat di luar sistem yang akan dibuat itu. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor [SLM-13:155].

Tabel 2.2 menunjukkan simbol-simbol yang ada pada diagram *use case* :

Tabel 2.2 Simbol-simbol pada *use case*

Simbol	Deskripsi
	Fungsionalitas yang disediakan unit-unit yang saling bertukar pesan antar unit atau aktor. Biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i> .
	Orang, proses, atau sistem lain yang berinteraksi dengan sistem yang akan dibuat diluar sistem yang akan dibuat itu sendiri. Biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
Asosiasi / <i>association</i>	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use</i>

	<p><i>case</i> atau <i>use case</i> memiliki interaksi dengan actor.</p>
<p>Ekstensi / <i>extend</i></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan berdiri sendiri walau tanpa <i>use case</i> tambahan itu. Biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan. Arah panah mengarah pada <i>use case</i> yang ditambahkan.</p>
<p>Generalisasi / <i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah <i>use case</i> yang dimana fungsi yang satu adalah fungsi yang lebih umum dari yang lainnya. Arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum).</p>
<p><i>Include</i></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankannya <i>use case</i> ini.</p>

Sumber : [SLM-13:156]




2.16 Activity Diagram



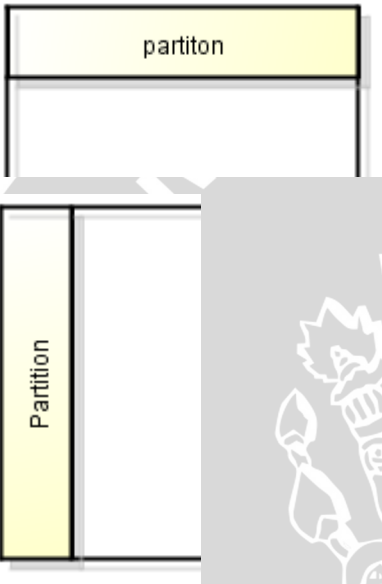
Diagram aktivitas atau *Activity diagram* merupakan salah satu bentuk pemodelan yang menggambarkan aliran kerja (*workflow*) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak bukan menggambarkan apa yang dilakukan oleh actor [SLM-13:161]. Diagram aktivitas biasa digunakan untuk mendefinisikan beberapa hal berikut:

- rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis system yang didefinisikan
- urutan atau pengelompokan tampilan dari sistem / *user interface* dimana setiap aktivitas memiliki sebuah rancangan tampilan antarmuka
- rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya
- rancangan menu yang ditampilkan pada perangkat lunak.

Tabel 2.3 menunjukkan simbol-simbol yang terdapat pada diagram aktivitas:

Tabel 2.3 Simbol-simbol pada *activity diagram*

Simbol	Deskripsi
	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
	Aktivitas yang dilakuakn sistem, aktivitas biasanya diawali dengan kata kerja
	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu

<p>Penggabungan / <i>join</i></p> 	<p>Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu</p>
<p>Status akhir</p> 	<p>Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir</p>
<p>Partition</p> 	<p>Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi</p>



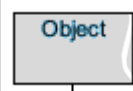

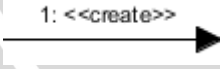
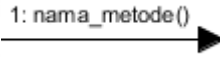
Sumber : [SLM-13:162]

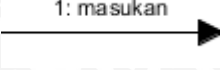
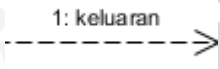
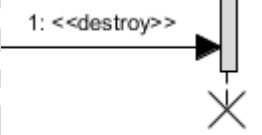
2.17 Sequence diagram

Sequence diagram adalah diagram yang digunakan untuk menggambarkan kelakuan objek dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan serta diterima antar objek [SLM-13]. Diagram ini berasosiasi dengan *use case* diagram. Banyaknya diagram yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri. *Sequence diagram* memperlihatkan tahap demi tahap aliran logika dalam sebuah sistem secara visual. Tabel 2.4 menunjukkan simbol-simbol yang ada pada *sequence diagram*.



Tabel 2.4 Simbol-simbol pada *sequence diagram*

Simbol	Deskripsi
<p>Aktor</p> 	<p>Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem yang akan dibuat. Meskipun memiliki simbol berupa gambar orang, tetapi actor belum tentu merupakan orang. Biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.</p>
<p>Garis hidup / <i>lifeline</i></p> 	<p><i>Lifeline</i> menyatakan kehidupan dari suatu objek</p>
<p>Objek</p> 	<p>Menyatakan objek yang berinteraksi dengan pesan</p>
<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya.</p>
<p>Pesan tipe <i>create</i></p> 	<p>Menyatakan suatu objek membuat yang objek lain, arah panah mengarah pada objek yang dibuat.</p>
<p>Pesan tipe <i>call</i></p> 	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain, arah panah mengarah pada objek yang memiliki operasi atau metode tersebut.</p>

<p>Pesan tipe <i>send</i></p> 	<p>Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dituju.</p>
<p>Pesan tipe <i>return</i></p> 	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi/metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.</p>
<p>Pesan tipe <i>destroy</i></p> 	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri.</p>

Sumber : [SLM-13:165]

2.18 *Functional Test*

Functional test adalah test yang dilakukan berdasarkan fungsi-fungsi yang dibutuhkan dan terdapat pada aplikasi yang sedang dibangun. *Functional test* dilakukan untuk memastikan bahwa seluruh fungsi di dalam aplikasi dapat berjalan sesuai dengan yang diinginkan. Misalnya, fungsi dari tombol simpan, ganti, hapus. *Tester* harus bisa membuktikan bahwa record tersebut benar-benar tersimpan, terganti atau terhapus. Pada *functional test*, struktur dan kode program diabaikan [ATM-10:175].

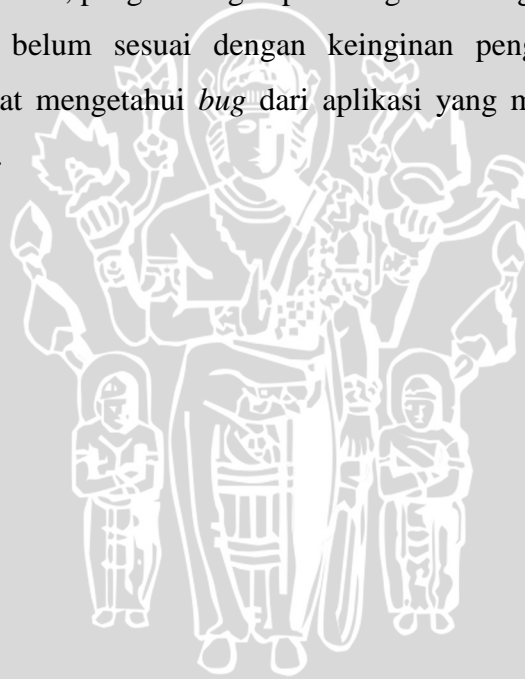
2.19 *Usability Test*

Pengertian usabilitas berdasarkan *International Organization for Standardization (ISO) 9241-11* adalah kelebihan dari sebuah produk yang bisa didapatkan oleh pengguna untuk mencapai sebuah tujuan tertentu yang dapat dinilai berdasarkan efektifitas, efisiensi, dan kepuasan pengguna terhadap produk tersebut [BCM-11:11]. Dalam konteks pengujian usabilitas sebuah aplikasi berarti

pengujian yang dilakukan untuk melihat pengguna ketika menggunakan aplikasi bahwa aplikasi yang dibuat mudah untuk digunakan, dipahami, dan memberikan nilai lebih terhadap pengguna.

2.20 *User Acceptance Test*

User Acceptance Test adalah pengujian yang biasanya dilakukan terakhir kali sebelum aplikasi siap untuk digunakan oleh pengguna. Selain itu pengujian ini juga berfungsi untuk mendapatkan persetujuan dari pengguna. Pada pengujian ini, pengguna diminta untuk menggunakan aplikasi yang telah dibuat, sehingga pengguna dapat merasakan langsung aplikasi yang dibuat apakah telah sesuai atau tidak. Melalui pengujian ini, pengembang dapat mengetahui bagian mana saja dari aplikasi yang masih belum sesuai dengan keinginan pengguna. Selain itu pengembang juga dapat mengetahui *bug* dari aplikasi yang masih terlewat dari pengujian sebelumnya.



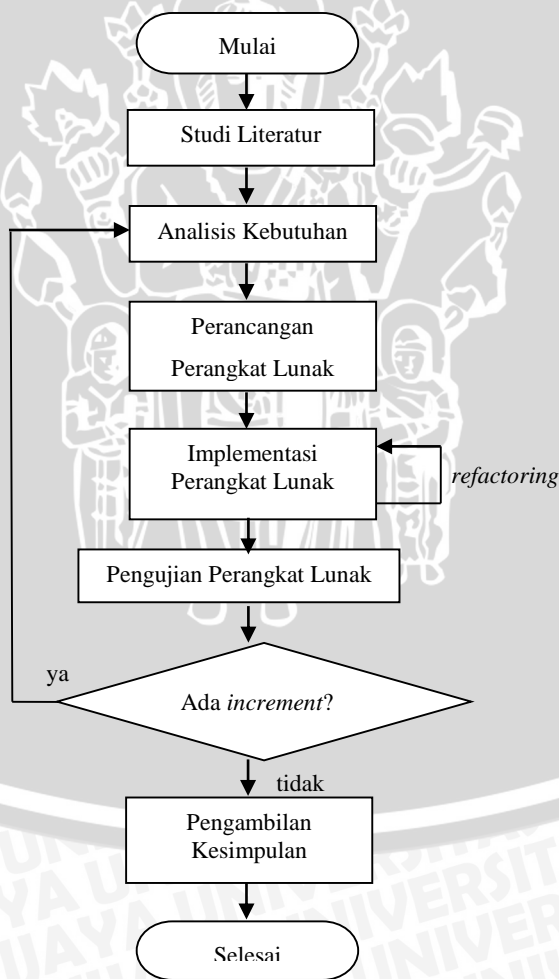
BAB III

METODOLOGI PENELITIAN DAN PERANCANGAN

Bab ini membahas mengenai metode yang digunakan dalam penelitian ini serta analisis kebutuhan dan perancangan aplikasi *Enterprise Resource Planning* di Lembaga Kedaulatan Mahasiswa Program Teknologi Informasi dan Ilmu Komputer.

3.1 Metodologi Penelitian

Langkah-lagkah yang dilakukan dalam penelitian ini yaitu : studi literatur, analisis kebutuhan, perancangan perangkat lunak, implementasi perangkat lunak, pengujian perangkat lunak serta pengambilan kesimpulan dan saran.



Gambar 3.1 Diagram Alir Metode Penelitian

Penjelasan dari masing-masing proses pada Gambar 3.1 diagram alir metode penelitian adalah sebagai berikut :

3.1.1 Studi Literatur

Studi literatur merupakan cara yang digunakan dalam pengumpulan informasi-informasi yang diperlukan dengan sumber yang digunakan berupa data-data dari buku dan website. Dalam studi literatur dikumpulkan data-data dari sumber yang terpercaya yang menjadikan sumber tersebut menjadi dasar dari teori-teori yang akan diimplementasikan lebih lanjut pada aplikasi yang dibuat.

Dalam hal ini studi literatur yang dipergunakan adalah :

1. Lembaga Kemahasiswaan Program Teknologi Informasi dan Ilmu Komputer
2. Codeigniter
3. Enterprise Resource Planning
4. Rekayasa Perangkat Lunak
 - a. Analisis Perangkat Lunak
 - b. Perancangan Perangkat Lunak
 - c. Implementasi
 - d. Pengujian Perangkat Lunak

3.1.2 Analisis Kebutuhan

Analisis kebutuhan dilakukan untuk mengetahui kebutuhan yang diperlukan oleh aplikasi yang akan dibangun sehingga aplikasi dapat berjalan dengan baik dan sesuai dengan yang dibutuhkan. Analisis kebutuhan dilakukan dengan mengidentifikasi seluruh kebutuhan sistem yang direpresentasikan dalam pemodelan *Class Diagram* dan *Use Case Diagram*.

3.1.3 Perancangan Perangkat Lunak

Perancangan perangkat lunak dilakukan setelah semua kebutuhan telah didapatkan dari proses analisis kebutuhan. Perancangan perangkat lunak terdiri dari pembuatan desain database program, pembuatan desain antar muka program, dan pembuatan pembuatan *activity diagram*.

3.1.4 Implementasi Perangkat Lunak

Setelah melakukan perancangan perangkat lunak, maka pembuatan dari aplikasi ini dapat dilakukan. Pembuatan aplikasi ini akan dilakukan dengan mengacu dokumen-dokumen perancangan yang telah dibuat sebelumnya. Pada tahap ini terdapat proses *refactoring* yang merupakan salah satu ciri khas dari metode pengembangan *extreme programming*. *Refactoring* adalah melakukan perubahan pada kode program dari perangkat lunak dengan tujuan meningkatkan kualitas dari struktur program tersebut tanpa mengubah cara program tersebut bekerja. Dengan *refactoring*, pengembang dapat melakukan berbagai usaha untuk meningkatkan kualitas program tanpa kembali mengulang-ulang proses desain [WDP-08].

3.1.5 Pengujian Perangkat Lunak

Untuk memastikan bahwa aplikasi yang dibuat berjalan sesuai dengan yang diinginkan, maka perlu dilakukan suatu pengujian. Pengujian yang dilakukan terhadap aplikasi ini yaitu *functional test* dan *usability test*. *Functional test* dilakukan untuk memastikan bahwa seluruh fungsi aplikasi dapat berjalan dengan baik dan sesuai dengan yang diinginkan. *Usability test* dilakukan untuk mengetahui kemudahan penggunaan aplikasi.

3.1.6 Pengambilan Kesimpulan

Pengambilan kesimpulan dilakukan setelah semua tahapan analisis kebutuhan, perancangan, implementasi dan pengujian sistem aplikasi telah selesai dilakukan. Kesimpulan diambil dari hasil pengujian dan analisis terhadap sistem yang dibangun. Tahap terakhir dari penulisan adalah saran yang dimaksudkan untuk memperbaiki kesalahan-kesalahan yang terjadi dan menyempurnakan penulisan serta untuk memberikan pertimbangan atas pengembangan aplikasi selanjutnya.

3.2 Analisis dan Perancangan

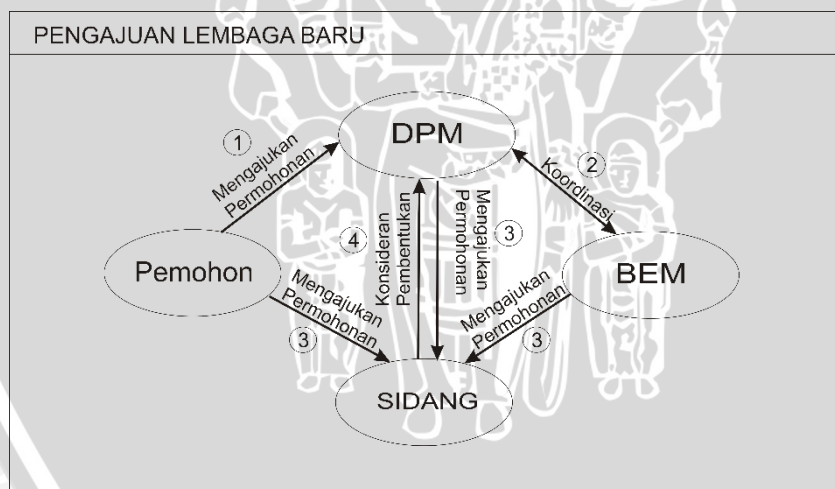
Dalam analisis dan perancangan akan dipaparkan mengenai analisis kebutuhan fungsional serta perancangan aplikasi berupa *business process model and notation*, *activity diagram*, *use case diagram*, *class diagram*, dan *physical diagram database*.

3.2.1 Business Process Model and Notation

Business Process Model and Notation (BPMN) digunakan untuk memodelkan proses bisnis yang ada sekarang dan setelah sistem yang akan dibuat ini ada. Pemodelan dari proses bisnis yang ada sekarang dan setelah ada sistem dapat dilihat pada Lampiran A.

3.2.2 Rich Picture

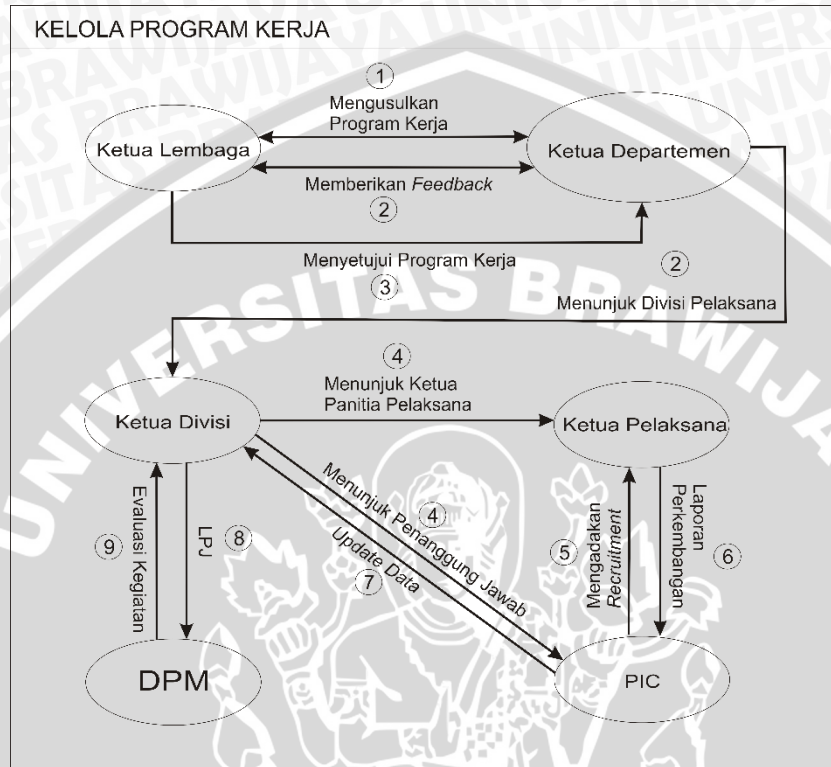
Rich Picture digunakan untuk untuk menggambarkan interaksi yang terjadi di dalam sistem.



Gambar 3.2 Rich Picture Pengajuan Lembaga Baru

Gambar 3.2 merupakan gambaran interaksi yang terjadi pada sistem pada saat pengajuan lembaga baru. Tahap pertama, pemohon harus mengajukan permohonan pembentukan dengan mengisi *form* yang disediakan terlebih dahulu. Tahap kedua, DPM menerima permohonan yang diajukan dan melakukan koordinasi dengan BEM. Hasil dari koordinasi yang dilakukan di-*upload* ke sistem. Tahap ketiga, Pemohon, DPM, dan BEM akan mengajukan permohonan ke

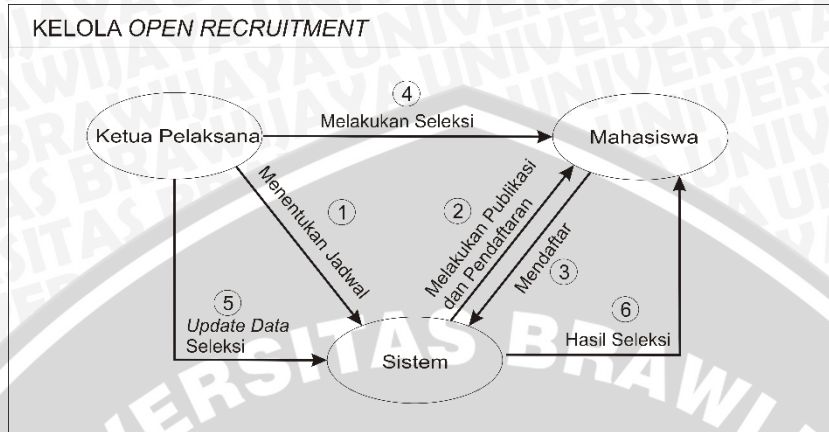
sidang. Terakhir, sidang akan memberikan pertimbangan apabila lembaga yang diajukan telah disetujui untuk dibentuk. Pertimbangan yang dihasilkan diberikan kepada DPM untuk di-*upload* ke sistem.



Gambar 3.3 Rich Picture Kelola Program Kerja

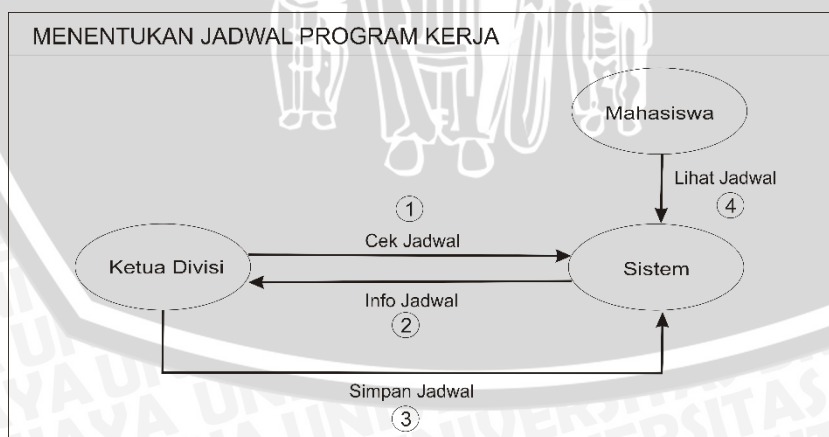
Gambar 3.3 merupakan gambaran interaksi yang terjadi pada sistem pada saat pengelolaan program kerja. Tahap pertama, ketua lembaga atau ketua departemen mengusulkan sebuah program kerja. Tahap kedua, ketua lembaga atau ketua departemen dapat memberikan *feedback* dan ketua departemen menunjuk divisi yang bertanggung jawab atas program kerja tersebut. Tahap ketiga, ketua lembaga menyetujui program kerja yang diusulkan. Tahap keempat, ketua divisi yang bertanggung jawab menunjuk seorang ketua pelaksana dan penanggung jawab atau PIC dari program kerja tersebut. Tahap kelima, PIC meminta ketua pelaksana untuk mengadakan *recruitment*. Tahap keenam, ketua pelaksana melaporkan perkembangan program kerja kepada PIC. Tahap ketujuh PIC melaporkan perkembangan tersebut ke ketua divisi untuk dilakukan *update data* pada sistem. Tahap kedelapan, setelah program kerja selesai dilaksanakan, ketua divisi

menyerahkan Laporan Pertanggung Jawaban (LPJ) kepada DPM melalui sistem. Tahap terakhir, DPM akan menyerahkan hasil evaluasi kegiatan.



Gambar 3.4 Rich Picture Kelola Open Recruitement

Gambar 3.4 merupakan gambaran interaksi yang terjadi pada sistem pada saat pengelolaan *open recruitment*. Tahap pertama, ketua pelaksana menentukan jadwal publikasi dan pendaftaran *open recruitment* pada sistem. Tahap kedua, sistem akan memublikasikan *open recruitment* yang telah dibuat. Tahap ketiga, mahasiswa dapat melakukan pendaftaran melalui sistem. Tahap keempat ketua pelaksana melakukan seleksi terhadap mahasiswa yang telah terdaftar. Tahap kelima, ketua pelaksana melakukan *update* hasil seleksi ke sistem. Tahap keenam, sistem akan mengumumkan hasil seleksi ke mahasiswa.



Gambar 3.5 Rich Picture Menentukan Jadwal Program Kerja

Gambar 3.5 merupakan gambaran interaksi yang terjadi pada sistem pada saat menentukan jadwal program kerja. Tahap pertama, ketua divisi akan

melakukan pengecekan jadwal yang diinginkan pada sistem. Tahap kedua, sistem akan memberikan info jadwal yang diinginkan. Tahap ketiga, ketua divisi menyimpan jadwal yang diinginkan. Tahap keempat, mahasiswa dapat melihat seluruh jadwal program kerja yang ada.

3.2.3 Kebutuhan Fungsional

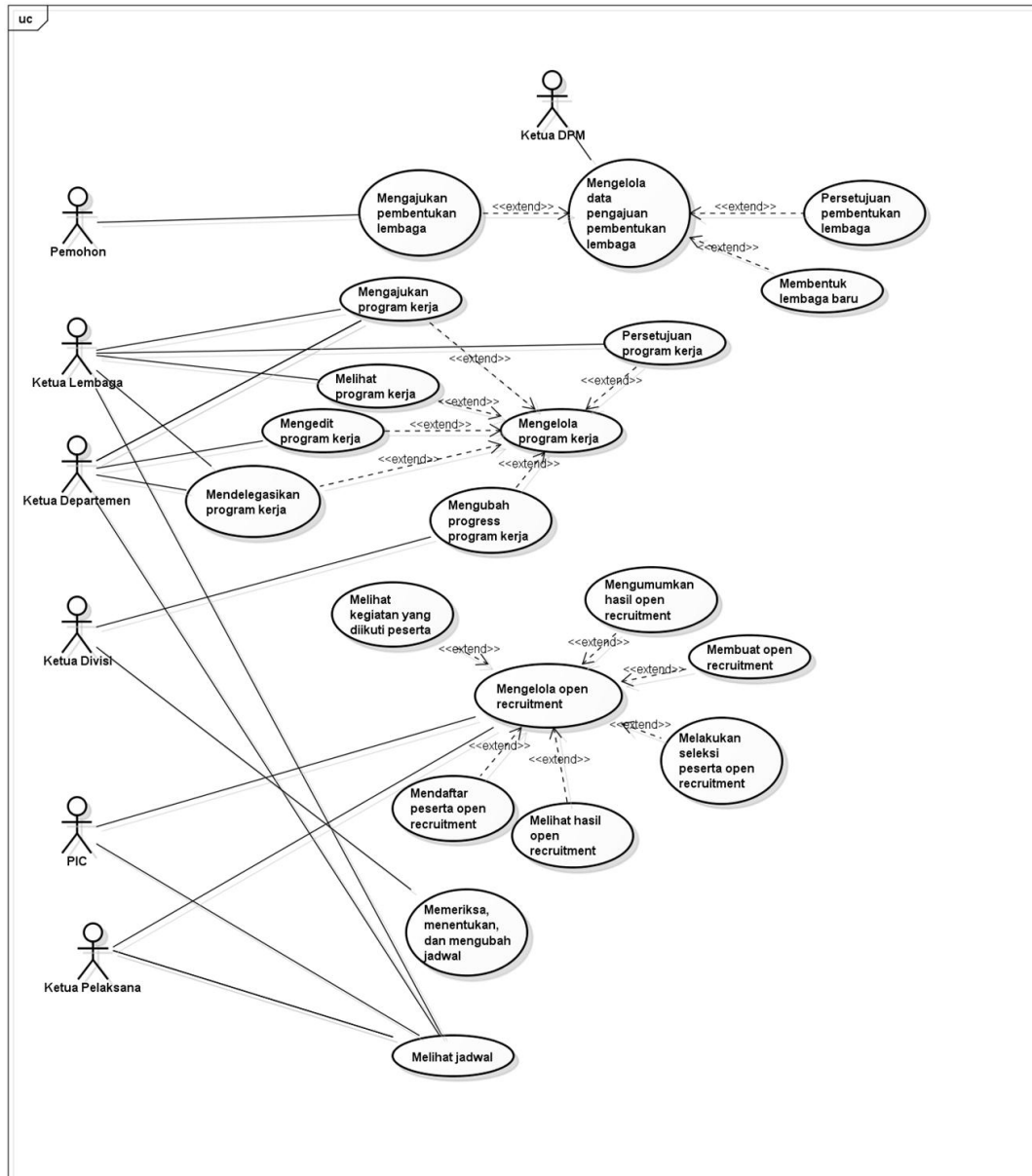
Kebutuhan fungsional merupakan kebutuhan utama yang dibutuhkan di dalam aplikasi ini. Daftar kebutuhan fungsional yang dibutuhkan oleh pengguna aplikasi ini dapat dilihat pada Tabel 3.1 di bawah ini.

Tabel 3.1 Daftar kebutuhan fungsional perangkat lunak

No SRS	Kebutuhan	Use Case
SRS_001_01	Mahasiswa dapat mengajukan pembentukan lembaga mahasiswa baru.	Mengisi form pengajuan pembentukan lembaga
SRS_001_02	Pengguna dapat mengelola pembentukan lembaga mahasiswa baru.	Mengelola data pengajuan pembentukan lembaga
SRS_001_03	Pengguna dapat menyetujui atau menolak pengajuan pembentukan lembaga dari mahasiswa.	Persetujuan pembentukan lembaga
SRS_002_01	Pengguna dapat mengelola program kerja.	Mengelola program kerja
SRS_002_02	Pengguna dapat melihat rincian program kerja.	Melihat program kerja
SRS_002_03	Pengguna dapat mengubah progress dari setiap program kerja.	Mengubah progress program kerja
SRS_002_04	Pengguna mendapatkan peringatan apabila jadwal program kerja yang diajukan bentrok dengan jadwal lain.	Memeriksa jadwal
SRS_003_01	Mahasiswa dapat melakukan pendaftaran <i>open recruitment</i> .	Mendaftar peserta <i>open recruitment</i>
SRS_003_02	Pengguna dapat melakukan seleksi terhadap peserta <i>open recruitment</i> .	Melakukan seleksi peserta <i>open recruitment</i>
SRS_003_03	Pengguna dapat melihat keaktifan peserta <i>recruitment</i> saat ini.	Melihat kegiatan yang diikuti peserta
SRS_003_04	Pengguna dapat melihat hasil <i>open recruitment</i>	Melihat hasil <i>open recruitment</i>
SRS_004_01	Mahasiswa dapat melihat jadwal kegiatan pada LKM PTIHK	Melihat jadwal

3.2.4 Use Case Diagram

Diagram *use case* digunakan untuk memodelkan perilaku (*behavior*) dari sistem yang akan dibuat. Diagram *use case* pada sistem ini dapat dilihat pada Gambar 3.6 berikut ini.



powered by Astah

Gambar 3.6 Diagram Use Case Aplikasi ERP LKM PTIIK



Tabel 3.2 berikut ini adalah penjelasan aktor dari *use case* yang terdapat pada Gambar 3.6 Sedangkan aktor dari *use case* pada gambar 3.6 terdapat pada Lampiran B.

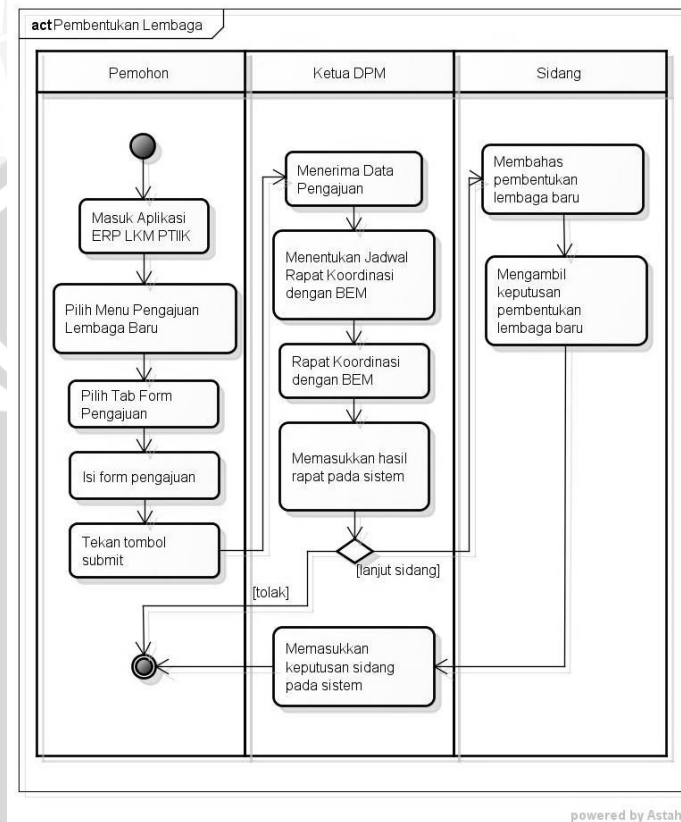
Tabel 3.2 Daftar aktor pada sistem

No	Aktor	Deskripsi
1	Presiden BEM	Mahasiswa yang mendapat amanah untuk memimpin Badan Eksekutif Mahasiswa Teknologi Informasi dan Ilmu Komputer
2	Ketua DPM	Mahasiswa yang mendapat amanah untuk memimpin Dewan Perwakilan Mahasiswa Program Teknologi Informasi dan Ilmu Komputer
3	Pemohon	Mahasiswa yang ingin mengajukan permohonan pembentukan lembaga baru
4	Ketua Lembaga	Mahasiswa yang mendapat amanah untuk memimpin sebuah lembaga di LKM PTIIK, misalnya adalah ketua DPM, presiden BEM, ketua himpunan, ketua umum BIOS, ketua LKI-AMD, pimpinan umum DISPLAY, ketua OPTIIK, ketua RAION dan ketua POROS.
5	Ketua departemen	Mahasiswa yang mendapat amanah untuk memimpin departemen dalam sebuah lembaga mahasiswa atau jabatannya berada satu tingkat di bawah ketua dan wakil ketua lembaga mahasiswa, misalnya kementerian advokesma pada BEM.
6	Ketua divisi	Mahasiswa yang mendapat amanah untuk memimpin sebuah divisi dalam sebuah departemen atau jabatannya berada dua tingkat di bawah ketua dan wakil ketua lembaga mahasiswa, misalnya ketua divisi advokasi pada kementerian Advokesma.
7	PIC	<i>Person in Charge</i> adalah anggota lembaga yang bertanggung jawab terhadap suatu proker
8	Ketua Pelaksana	Mahasiswa yang mendapatkan amanah untuk menjalankan sebuah kegiatan atau program kerja

3.2.5 Activity Diagram

Pada penelitian ini, *activity diagram* digunakan untuk memodelkan aktifitas pengguna pada sistem ini.

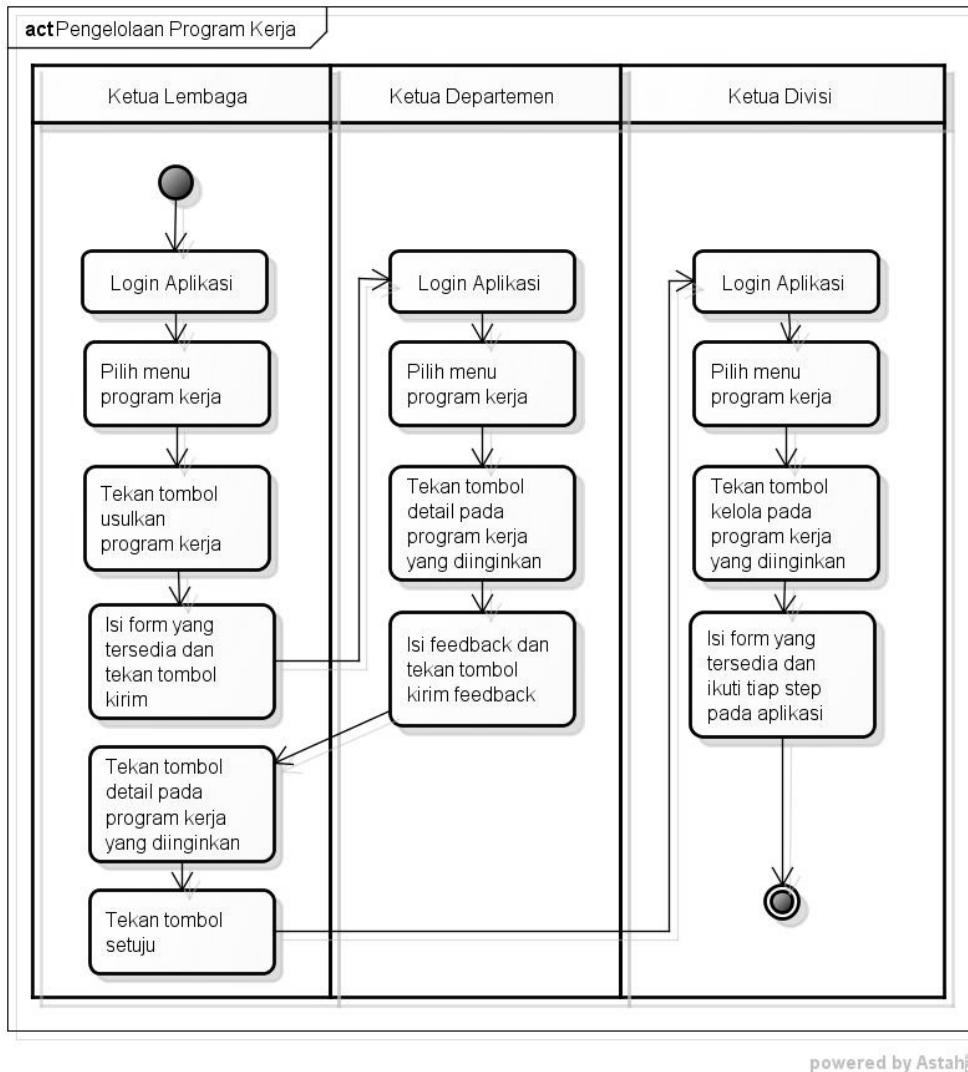
a. Pembentukan lembaga



Gambar 3.7 Activity diagram pengajuan lembaga baru

Gambar 3.7 menjelaskan aktifitas untuk pengajuan lembaga baru melalui aplikasi ERP LKM PTIIK. Seperti terlihat pada gambar 3.7 pemohon terlebih dahulu masuk aplikasi, kemudian mengisi form yang telah disediakan beserta persyaratan yang dibutuhkan. Selanjutnya DPM dan BEM yang akan mengolah pengajuan tersebut.

b. Pengelolaan Program Kerja



Gambar 3.8 Activity diagram pengelolaan program kerja

Gambar 3.8 menjelaskan aktifitas pengelolaan program kerja melalui aplikasi ERP LKM PTIIK. Pengelolaan program kerja dapat dilakukan oleh ketua divisi. Sebelum bisa mengelola program kerja, ketua lembaga atau ketua departemen harus membuat sebuah program kerja terlebih dahulu. Kemudian ketua departemen memberikan program kerja tersebut ke divisi yang sesuai. Selanjutnya ketua divisi dapat melakukan pengelolaan program kerja.

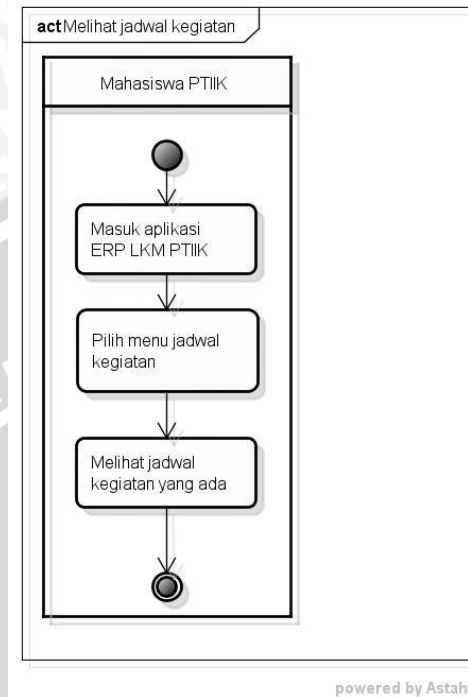
c. Pengelolaan *Open Recruitment*



Gambar 3.9 Activity diagram pengelolaan open recruitment

Gambar 3.9 menjelaskan aktifitas pegelolaan *open recruitment* melalui aplikasi ERP LKM PTIIK. Pengelolaan *open recruitment* dilakukan oleh ketua pelaksana. Ketua pelaksana harus mengisialisasi terlebih dahulu *open recruitment* yang akan dilakukan. Setelah selesai dan pendaftaran telah dibuka, mahasiswa PTIIK dapat melakukan pendaftaran *open recruitment* melalui aplikasi ini.

d. Melihat seluruh jadwal kegiatan

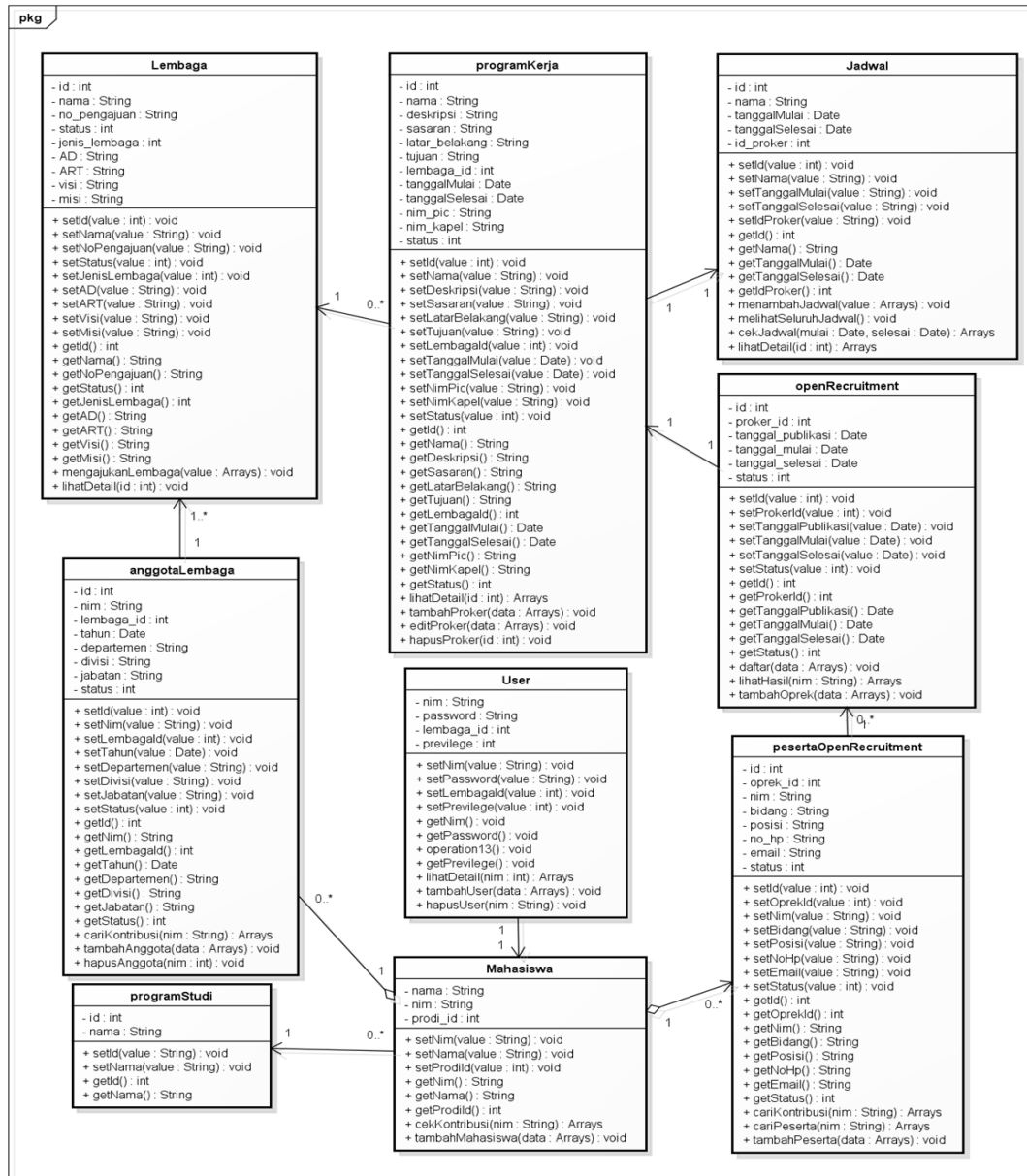


Gambar 3.10 Activity diagram melihat jadwal kegiatan

Gambar 3.10 menjelaskan aktifitas untuk melihat jadwal kegiatan melalui aplikasi ERP LKM PTIIK. Seluruh mahasiswa PTIIK dapat melihat jadwal kegiatan yang ada di PTIIK. Mahasiswa cukup mengakses aplikasi, kemudian memilih menu jadwal kegiatan untuk melihat seluruh jadwal kegiatan yang ada.

3.2.6 Class Diagram

Class diagram digunakan untuk memodelkan struktur sistem yang akan dibuat dilihat dari kelas-kelas yang ada.

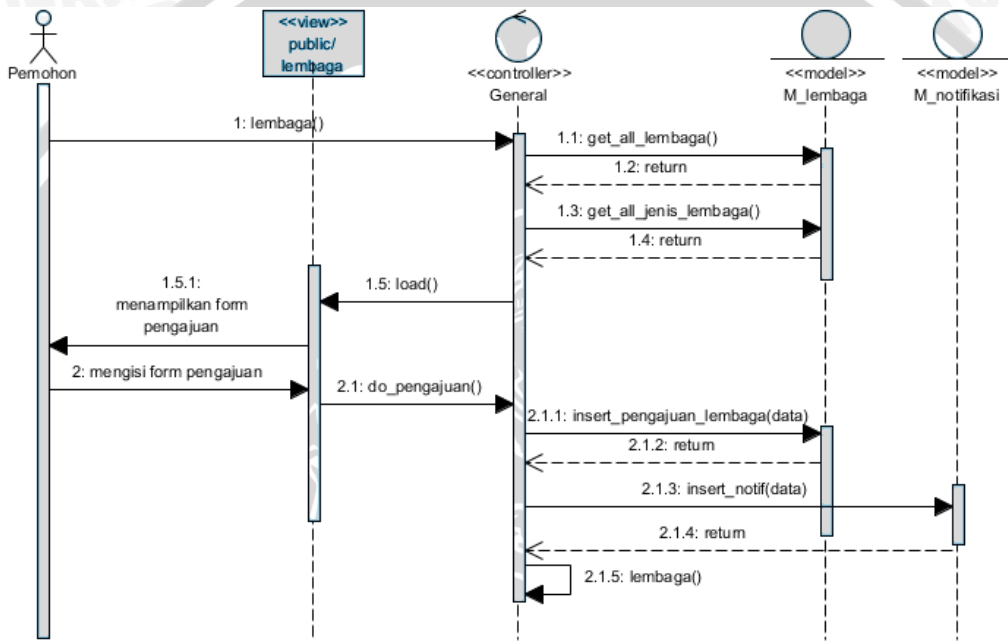


Gambar 3.11 Class diagram aplikasi ERP LKM PTIIK

Gambar 3.11 menunjukkan rancangan class diagram secara logical pada aplikasi ERP LKM PTIIK. Class diagram pada gambar 3.11 akan berbeda ketika disesuaikan dengan struktur pada framework Codeigniter. Class Diagram yang telah disesuaikan dengan framework Codeigniter dapat dilihat pada Lampiran C.

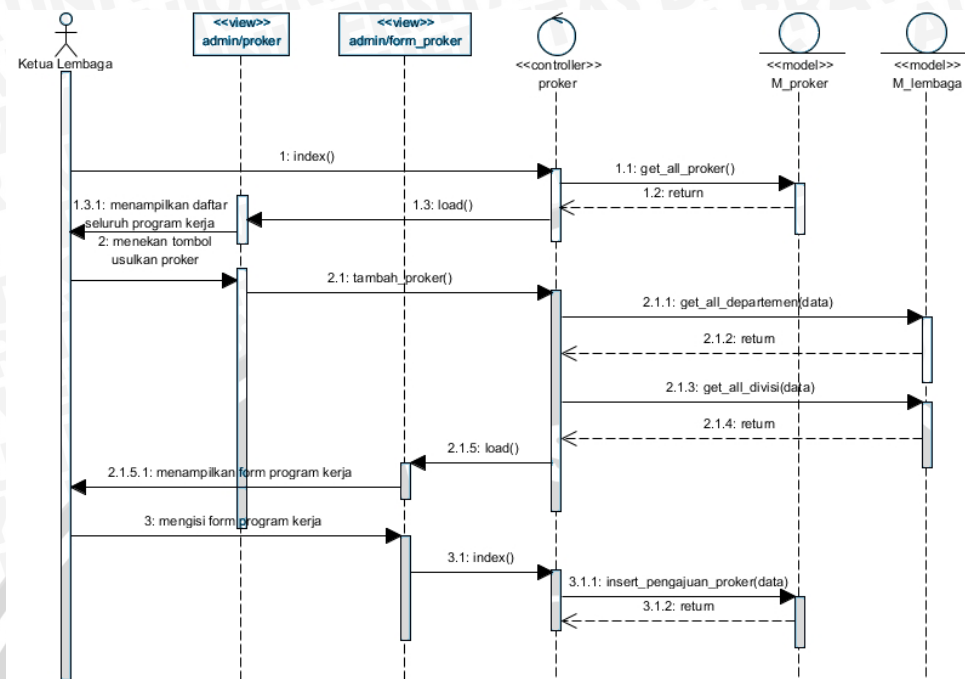
3.2.7 Sequence Diagram

Sequence diagram digunakan untuk menggambarkan perilaku dalam sebuah sistem. Terdapat beberapa *sequence diagram* yang akan dijelaskan pada bagian ini, yaitu *sequence diagram* pengajuan lembaga baru, *sequence diagram* mengusulkan program kerja, *sequence diagram* mengelola *open recruitment*, dan *sequence diagram* melihat jadwal. *Sequence diagram* lainnya terdapat pada Lampiran C.



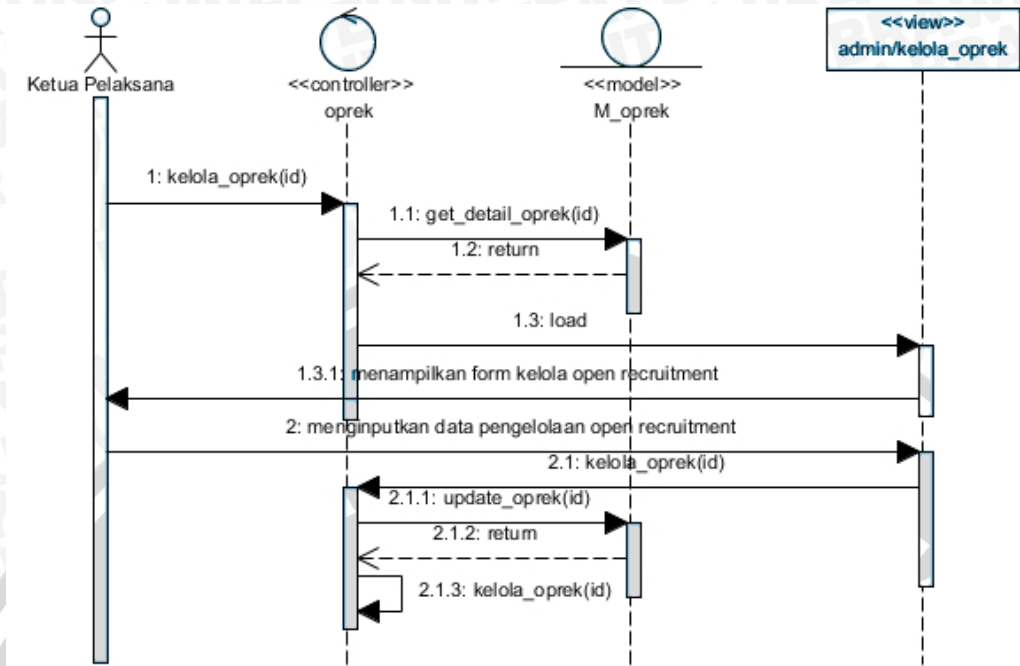
Gambar 3.12 *Sequence diagram* mengajukan lembaga baru

Gambar 3.12 merupakan *sequence diagram* dari proses mengajukan lembaga baru. Terdapat satu aktor dan empat objek pada proses ini. Dalam mengajukan sebuah lembaga baru, pemohon akan memanggil *method* lembaga yang terdapat pada *controller* General. *Method* tersebut akan mengambil data seluruh lembaga yang ada dan jenis lembaga yang ada pada *database* melalui model M_lembaga. Kemudian *controller* General akan menampilkan form pengajuan yang harus diisi. Setelah pemohon melakukan *submit form*, *controller* General akan menjalankan *method* do_pengajuan serta menyimpan data pengajuan melalui model M_lembaga dan menyimpan notifikasi baru melalui model M_notifikasi. Kemudian *controller* General akan memanggil *method* lembaga lagi.



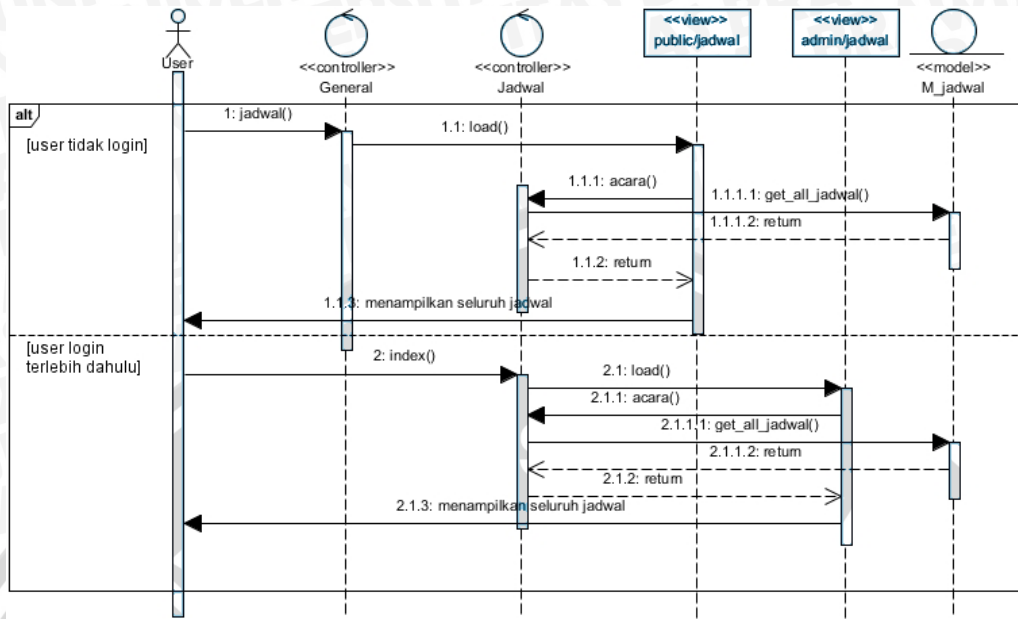
Gambar 3.13 *Sequence diagram* mengusulkan program kerja

Gambar 3.13 merupakan *sequence diagram* dari proses mengusulkan program kerja. Terdapat satu aktor dan empat objek pada proses ini. Dalam mengusulkan sebuah program kerja, ketua lembaga akan memanggil fungsi index pada *controller* proker. Kemudian aplikasi akan menampilkan daftar seluruh program kerja yang ada. Ketika ketua lembaga menekan tombol usulkan proker, aplikasi akan memanggil fungsi tambah_proker pada *controller* proker, dan ketua lembaga akan melihat *form* untuk mengusulkan program kerja. Ketika *form* tersebut di *submit* aplikasi akan memanggil fungsi index kembali dan menyimpan data yang telah diisi pada *database* melalui model M_proker.



Gambar 3.14 Sequence diagram mengelola open recruitment

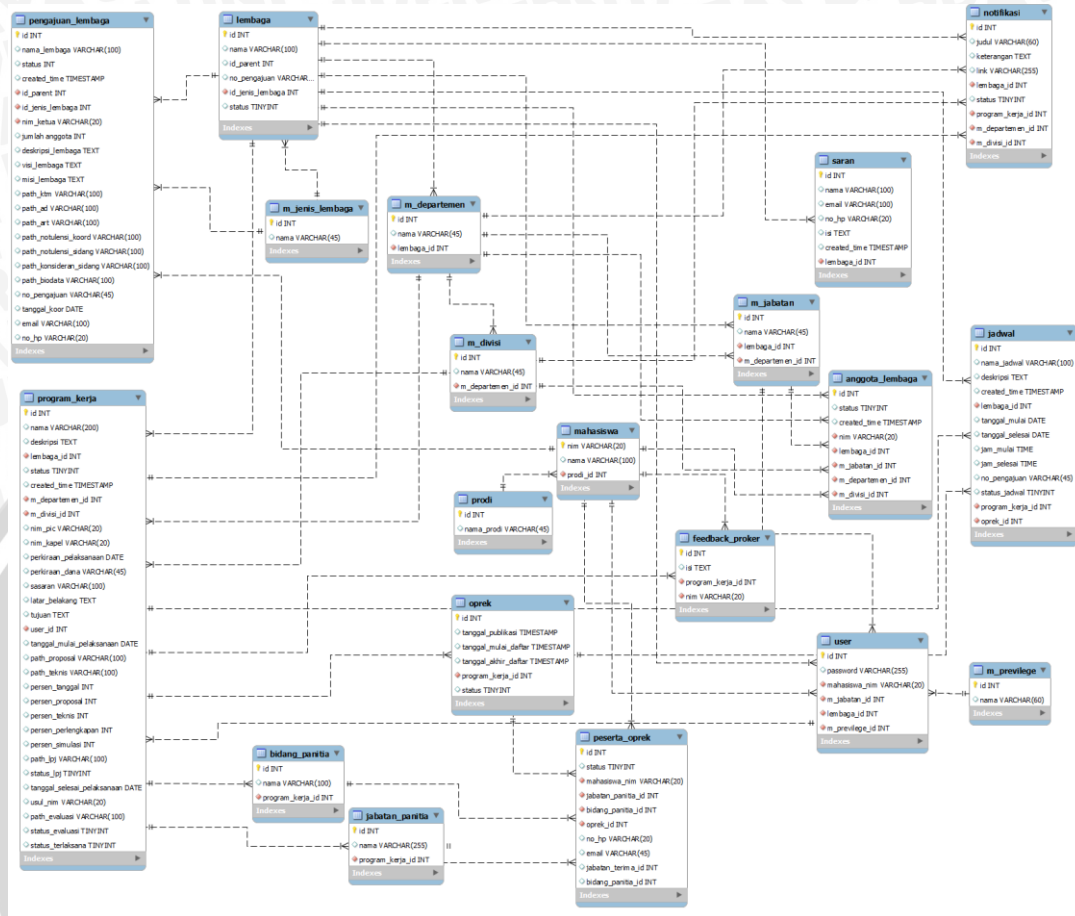
Gambar 3.14 merupakan *sequence diagram* dari proses mengelola *open recruitment*. Terdapat satu aktor dan empat objek yang saling berinteraksi pada proses ini. Untuk mengelola *open recruitment* ketua pelaksana akan memanggil fungsi *kelola_oprek* pada *controller* *oprek*. Kemudian aplikasi akan menampilkan form *kelola open recruitment* yang harus diisi. Terdapat beberapa *form* yang ditampilkan, tergantung pada *progress open recruitment* tersebut. Ketua pelaksana akan mengisi seluruh *form* tersebut. Data dari *form* tersebut akan dikirim ke *controller* *kelola_oprek*. Kemudian *controller* akan memperbarui data pada database melalui *model* *m_oprek*. Setelah itu *controller* akan kembali memanggil fungsi *kelola_oprek*.



Gambar 3.15 Sequence diagram melihat jadwal

Gambar 3.15 merupakan *sequence diagram* dari proses melihat jadwal seluruh kegiatan yang ada. Aktor user pada proses ini dibedakan menjadi dua, yaitu user yang melakukan *login* ke aplikasi dan yang tidak melakukan *login*. User yang tidak melakukan *login* pada aplikasi akan memanggil fungsi jadwal pada *controller* general. Kemudian *controller* general akan menampilkan seluruh jadwal yang ada melalui *view* public/jadwal. User yang melakukan *login* pada aplikasi akan memanggil fungsi *index* pada *controller* jadwal. Kemudian *controller* jadwal akan menampilkan seluruh jadwal yang ada melalui *view* admin/jadwal.

3.2.8 Physical Diagram Database

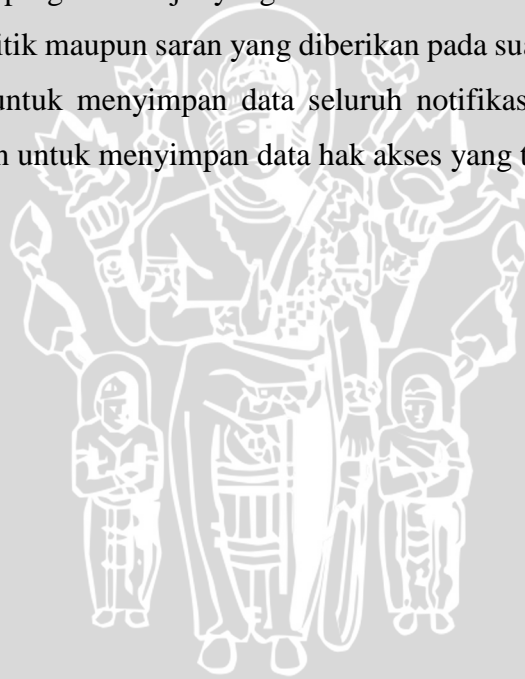


Gambar 3.16 Physical diagram database aplikasi ERP LKM PTIIK

Gambar 3.12 merupakan rancangan database aplikasi ERP LKM PTIIK. Rancangan tersebut kemudian dapat dijadikan sebuah script SQL untuk diimplementasikan. Terdapat 20 (dua puluh) tabel yang saling berhubungan pada sistem ini. Tabel pengajuan_lembaga digunakan untuk menyimpan seluruh data pengajuan lembaga yang ada. Tabel program_kerja digunakan untuk menyimpan data seluruh program kerja yang ada. Tabel lembaga digunakan untuk menyimpan data seluruh lembaga yang ada. Tabel anggota_lembaga digunakan untuk menyimpan data anggota dari setiap lembaga. Tabel jadwal digunakan untuk menyimpan seluruh jadwal yang ada. Tabel oprek digunakan untuk menyimpan seluruh *open recruitment* yang ada. Tabel peserta_oprek digunakan untuk menyimpan data seluruh peserta *open recruitment*. Tabel user digunakan untuk menyimpan data user yang dapat mengakses aplikasi ini. Tabel mahasiswa



digunakan untuk menyimpan data seluruh mahasiswa. Tabel `m_jenis_lembaga` merupakan tabel yang digunakan untuk menyimpan data master jenis lembaga yang ada. Tabel `m_departemen` digunakan untuk menyimpan data seluruh departemen yang ada. Tabel `m_divisi` digunakan untuk menyimpan data seluruh divisi yang ada. Tabel `m_jabatan` digunakan untuk menyimpan data seluruh jabatan yang ada pada suatu lembaga. Tabel `prodi` digunakan untuk menyimpan data seluruh program studi yang ada. Tabel `bidang_panitia` digunakan untuk menyimpan data bidang panitia apa saja yang dibuka pada suatu *open recruitment*. Tabel `jabatan_panitia` digunakan untuk menyimpan data jabatan panitia apa saja yang dibuka pada suatu *open recruitment*. Tabel `feedback_proker` digunakan untuk menyimpan data *feedback* pada suatu program kerja yang ada. Tabel `saran` digunakan untuk menyimpan seluruh kritik maupun saran yang diberikan pada suatu lembaga. Tabel `notifikasi` digunakan untuk menyimpan data seluruh notifikasi yang ada. Tabel `m_previlege` digunakan untuk menyimpan data hak akses yang tersedia.



BAB IV

IMPLEMENTASI

Bab ini membahas mengenai implementasi dari aplikasi *Enterprise Resource Planning* LKM PTIHK berdasarkan perancangan dan analisis kebutuhan yang telah dilakukan sebelumnya. Pembahasan terdiri atas batasan dalam implementasi, implementasi basis data, implementasi tiap *class* yang ada pada perancangan, dan implementasi antar muka.

4.1 Batasan Implementasi

Beberapa batasan dalam mengimplementasikan aplikasi *Enterprise Resource Planning* pada LKM PTIHK adalah sebagai berikut:

- a. *Database Mangement System* yang digunakan adalah *MySQL*
- b. Versi PHP yang digunakan adalah versi 5.5.6
- c. Versi *framework* Codeigniter yang digunakan adalah versi 2.1.4
- d. Implementasi dilakukan pada server lokal dengan menggunakan XAMPP versi 3.2.1
- e. Data mahasiswa yang digunakan pada saat implementasi adalah data *dummy* yang berarti bukan data mahasiswa yang sebenarnya.

4.2 Implementasi Basis Data

Implementasi basis data dilakukan dengan Database Management System *MySQL*. Hasil dari implementasi basis data berupa *script* SQL. Berikut ini merupakan tabel hasil implementasi yang telah dilakukan.

4.2.1 Tabel Anggota Lembaga

Name	Type	Length	Decimals	Not null	
▶ id	int	11	0	<input checked="" type="checkbox"/>	🔑 1
status	int	11	0	<input type="checkbox"/>	
nim	varchar	20	0	<input checked="" type="checkbox"/>	
lembaga_id	int	11	0	<input checked="" type="checkbox"/>	
m_jabatan_id	int	11	0	<input checked="" type="checkbox"/>	
m_departemen_id	int	11	0	<input checked="" type="checkbox"/>	
m_divisi_id	int	11	0	<input checked="" type="checkbox"/>	
created_time	timestamp	0	0	<input type="checkbox"/>	

Gambar 4.1 Screenshot tabel anggota_lembaga

Gambar 4.1 merupakan atribut dari tabel anggota_lembaga. Tabel ini digunakan untuk menyimpan data seluruh anggota lembaga pada LKM PTIIK. Kolom id merupakan *primary key* dari tabel ini. Kolom status digunakan untuk menyimpan status anggota, apakah masih aktif atau tidak. Kolom nim merupakan Nomor Induk Mahasiswa dari anggota lembaga. Kolom nim juga merupakan *foreign key* dari tabel mahasiswa. Kolom lembaga_id merupakan *foreign key* dari tabel lembaga. Kolom m_jabatan_id merupakan *foreign key* dari tabel m_jabatan. Kolom m_departemen_id merupakan *foreign key* dari tabel m_departemen. Kolom m_divisi_id merupakan *foreign key* dari tabel m_divisi. Kolom created_time merupakan kolom untuk menyimpan waktu pembuatan tiap data pada tabel anggota_lembaga.

4.2.2 Tabel Jadwal

Name	Type	Length	Decimals	Not null	
▶ id	int	11	0	<input checked="" type="checkbox"/>	🔑 1
nama_jadwal	varchar	100	0	<input type="checkbox"/>	
deskripsi	text	0	0	<input type="checkbox"/>	
created_time	timestamp	0	0	<input type="checkbox"/>	
lembaga_id	int	11	0	<input checked="" type="checkbox"/>	
tanggal_mulai	date	0	0	<input type="checkbox"/>	
tanggal_selesai	date	0	0	<input type="checkbox"/>	
jam_mulai	time	0	0	<input type="checkbox"/>	
jam_selesai	time	0	0	<input type="checkbox"/>	
program_kerja_id	int	11	0	<input type="checkbox"/>	
oprek_id	int	11	0	<input type="checkbox"/>	
no_pengajuan	varchar	45	0	<input type="checkbox"/>	
status_jadwal	tinyint	4	0	<input type="checkbox"/>	

Gambar 4.2 Screenshot tabel jadwal

Gambar 4.2 merupakan atribut dari tabel jadwal. Tabel ini digunakan untuk menyimpan data seluruh seluruh jadwal kegiatan pada LKM PTIIK. Kolom id merupakan *primary key* dari tabel jadwal. Kolom nama_jadwal merupakan kolom untuk menyimpan nama jadwal. Kolom deskripsi merupakan kolom untuk menyimpan deskripsi jadwal yang disimpan. Kolom created_time merupakan kolom untuk menyimpan waktu pembuatan tiap data pada tabel jadwal. Kolom lembaga_id merupakan *foreign key* dari tabel lembaga. Kolom tanggal mulai merupakan kolom untuk menyimpan tanggal mulai kegiatan pada jadwal. Kolom tanggal selesai merupakan kolom untuk menyimpan tanggal selesai kegiatan pada jadwal. Kolom jam mulai merupakan kolom untuk menyimpan jam mulai kegiatan pada jadwal. Kolom jam selesai merupakan kolom untuk menyimpan jam selesai kegiatan pada jadwal. Kolom program_kerja_id merupakan kolom untuk menyimpan primary key dari tabel program kerja apabila jadwal yang disimpan berhubungan dengan sebuah program kerja. Kolom oprek_id merupakan kolom untuk menyimpan primary key dari tabel oprek apabila jadwal yang disimpan berhubungan dengan sebuah *open recruitment*. Kolom no_pengajuan merupakan kolom untuk menyimpan nomor pengajuan lembaga baru apabila jadwal yang disimpan berhubungan dengan sebuah pengajuan lembaga baru.

4.2.3 Tabel Lembaga

Name	Type	Length	Decimals	Not null	
▶ id	int	11	0	<input checked="" type="checkbox"/>	🔑 1
nama	varchar	100	0	<input type="checkbox"/>	
id_parent	int	11	0	<input type="checkbox"/>	
id_jenis_lembaga	int	11	0	<input checked="" type="checkbox"/>	
no_pengajuan	varchar	45	0	<input type="checkbox"/>	
status	tinyint	4	0	<input type="checkbox"/>	

Gambar 4.3 Screenshot tabel lembaga

Gambar 4.3 merupakan atribut dari tabel lembaga. Tabel ini digunakan untuk menyimpan data seluruh lembaga yang ada pada LKM PTIIK. Kolom id merupakan *primary key* dari tabel lembaga. Kolom nama merupakan kolom untuk menyimpan nama lembaga. Kolom id_parent merupakan kolom untuk menyimpan id lembaga lain yang secara struktur organisasi berada diatas lembaga tersebut. Kolom id_jenis lembaga merupakan *foreign key* dari tabel m_jenis_lembaga. Kolom no_pengajuan merupakan kolom untuk menyimpan nomor pengajuan dari lembaga tersebut. Kolom status merupakan kolom untuk menyimpan status dari sebuah lembaga.

4.2.4 Tabel M_departemen

Name	Type	Length	Decimals	Not null	
▶ id	int	11	0	<input checked="" type="checkbox"/>	🔑 1
nama	varchar	45	0	<input type="checkbox"/>	
lembaga_id	int	11	0	<input checked="" type="checkbox"/>	

Gambar 4.4 Screenshot tabel m_departemen

Gambar 4.4 merupakan atribut dari tabel m_departemen. Tabel ini digunakan untuk menyimpan data master departemen dari setiap lembaga yang ada di LKM PTIIK. Kolom id merupakan *primary key* dari tabel m_departemen. Kolom nama merupakan kolom untuk menyimpan nama departemen. Kolom lembaga_id merupakan *foreign key* dari dari tabel lembaga.

4.2.5 Tabel M_divisi

Name	Type	Length	Decimals	Not null	
▶ id	int	11	0	<input checked="" type="checkbox"/>	🔑 1
nama	varchar	45	0	<input type="checkbox"/>	
m_departemen_id	int	11	0	<input checked="" type="checkbox"/>	

Gambar 4.5 Screenshot tabel m_divisi

Gambar 4.5 merupakan atribut dari tabel m_divisi. Tabel ini digunakan untuk menyimpan data master divisi dari setiap departemen yang ada di LKM PTIIK. Kolom id merupakan *primary key* dari tabel m_divisi. Kolom nama merupakan kolom untuk menyimpan nama divisi. Kolom m_departemen_id merupakan *foreign key* dari dari tabel m_departemen.

4.2.6 Tabel M_jabatan

Name	Type	Length	Decimals	Not null	
▶ id	int	11	0	<input checked="" type="checkbox"/>	🔑 1
nama	varchar	45	0	<input type="checkbox"/>	
lembaga_id	int	11	0	<input checked="" type="checkbox"/>	
m_departemen_id	int	11	0	<input checked="" type="checkbox"/>	
m_divisi_id	int	11	0	<input checked="" type="checkbox"/>	

Gambar 4. 6 Screenshot tabel m_jabatan

Gambar 4.6 merupakan atribut dari tabel m_jabatan. Tabel ini digunakan untuk menyimpan data master jabatan dari setiap lembaga yang ada di LKM PTIIK. Kolom id merupakan *primary key* dari tabel m_jabatan. Kolom nama merupakan kolom untuk menyimpan nama jabatan. Kolom lembaga_id merupakan *foreign key* dari dari tabel lembaga. Kolom m_departemen_id merupakan *foreign key* dari tabel m_departemen. Kolom m_divisi_id merupakan *foreign key* dari tabel m_divisi.

4.2.7 Tabel M_jenis_lembaga

Name	Type	Length	Decimals	Not null	
▶ id	int	11	0	<input checked="" type="checkbox"/>	🔑 1
nama	varchar	45	0	<input type="checkbox"/>	

Gambar 4.7 Screenshot tabel m_jenis_lembaga

Gambar 4.7 merupakan atribut dari tabel `m_jenis_lembaga`. Tabel ini digunakan untuk menyimpan data master jenis lembaga yang ada di LKM PTIIK. Kolom `id` merupakan *primary key* dari tabel `m_jenis_lembaga`. Kolom `nama` merupakan kolom untuk menyimpan nama jenis lembaga.

4.2.8 Tabel M_privilege

Name	Type	Length	Decimals	Not null	
▶ id	int	11	0	<input checked="" type="checkbox"/>	🔑 1
nama	varchar	60	0	<input type="checkbox"/>	

Gambar 4.8 Screenshot tabel `m_privilege`

Gambar 4.8 merupakan atribut dari tabel `m_privilege`. Tabel ini digunakan untuk menyimpan data master hak akses yang ada pada aplikasi ERP LKM PTIIK. Kolom `id` merupakan *primary key* dari tabel `m_privilege`. Kolom `nama` merupakan kolom untuk menyimpan nama *privilege*.

4.2.9 Tabel Mahasiswa

Name	Type	Length	Decimals	Not null	
▶ nim	varchar	20	0	<input checked="" type="checkbox"/>	🔑 1
nama	varchar	100	0	<input type="checkbox"/>	
prodi_id	int	11	0	<input checked="" type="checkbox"/>	

Gambar 4.9 Screenshot tabel mahasiswa

Gambar 4.9 merupakan atribut dari tabel mahasiswa. Tabel ini digunakan untuk menyimpan data seluruh mahasiswa PTIIK. Kolom `nim` merupakan *primary key* dari tabel mahasiswa. Kolom `nama` merupakan kolom untuk menyimpan nama mahasiswa. Kolom `prodi_id` merupakan *foreign key* dari tabel `prodi`.

4.2.10 Tabel Oprek

Name	Type	Length	Decimals	Not null	
▶ id	int	11	0	<input checked="" type="checkbox"/>	🔑 1
tanggal_publicasi	timestamp	0	0	<input type="checkbox"/>	
tanggal_mulai_daftar	timestamp	0	0	<input type="checkbox"/>	
tanggal_akhir_daftar	timestamp	0	0	<input type="checkbox"/>	
program_kerja_id	int	11	0	<input checked="" type="checkbox"/>	
status	tinyint	4	0	<input type="checkbox"/>	

Gambar 4.10 Screenshot tabel oprek

Gambar 4.10 merupakan atribut dari tabel oprek. Tabel ini digunakan untuk menyimpan data seluruh *open recruitment* yang ada pada LKM PTIIK. Kolom id merupakan *primary key* dari tabel oprek. Kolom tanggal publikasi merupakan kolom untuk menyimpan tanggal publikasi dari *open recruitment*. Kolom tanggal_mulai_daftar merupakan kolom untuk menyimpan tanggal mulai pendaftaran. Kolom tanggal_selesai_daftar merupakan kolom untuk menyimpan tanggal selesai pendaftaran. Kolom program_kerja_id merupakan *foreign key* dari tabel program_kerja. Kolom status untuk menyimpan status dari *open recruitment*.

4.2.11 Tabel Pengajuan_ lembaga

Name	Type	Length	Decimals	Not null	
▶ id	int	11	0	<input checked="" type="checkbox"/>	🔑 1
nama_lembaga	varchar	100	0	<input type="checkbox"/>	
status	int	11	0	<input type="checkbox"/>	
created_time	timestamp	0	0	<input type="checkbox"/>	
id_parent	int	11	0	<input checked="" type="checkbox"/>	
id_jenis_lembaga	int	11	0	<input checked="" type="checkbox"/>	
jumlah_anggota	int	11	0	<input type="checkbox"/>	
deskripsi_lembaga	text	0	0	<input type="checkbox"/>	
visi_lembaga	text	0	0	<input type="checkbox"/>	
misi_lembaga	text	0	0	<input type="checkbox"/>	
path_ktm	varchar	100	0	<input type="checkbox"/>	
path_biodata	varchar	100	0	<input type="checkbox"/>	
path_ad	varchar	100	0	<input type="checkbox"/>	
path_art	varchar	100	0	<input type="checkbox"/>	
path_notulensi_koord	varchar	100	0	<input type="checkbox"/>	
path_notulensi_sidang	varchar	100	0	<input type="checkbox"/>	
path_konsideran_sidang	varchar	100	0	<input type="checkbox"/>	
no_pengajuan	varchar	45	0	<input type="checkbox"/>	
tanggal_koor	varchar	25	0	<input type="checkbox"/>	
nim_ketua	varchar	20	0	<input type="checkbox"/>	
email	varchar	100	0	<input type="checkbox"/>	
no_hp	varchar	20	0	<input type="checkbox"/>	

Gambar 4.11 Screenshot tabel pengajuan_lembaga

Gambar 4.11 merupakan atribut dari tabel pengajuan_lembaga. Tabel ini digunakan untuk menyimpan data seluruh pengajuan lembaga baru pada LKM PTIIK. Kolom id merupakan *primary key* dari tabel pengajuan_lembaga. Kolom nama_lembaga merupakan kolom untuk menyimpan nama lembaga yang diajukan. Kolom status merupakan kolom untuk menyimpan status pengajuan lembaga. Kolom created_time merupakan kolom untuk menyimpan waktu pembuatan dari tiap data yang dibuat. Kolom id_parent merupakan kolom untuk menyimpan *id* lembaga lain yang secara struktur organisasi berada diatas lembaga tersebut. Kolom id_jenis lembaga merupakan *foreign key* dari tabel m_jenis_lembaga. Kolom jumlah_anggota merupakan kolom untuk menyimpan jumlah anggota lembaga. Kolom deskripsi_lembaga merupakan kolom untuk menyimpan deskripsi lembaga yang diajukan. Kolom visi_lembaga merupakan kolom untuk menyimpan visi lembaga yang diajukan. Kolom misi_lembaga merupakan kolom untuk menyimpan misi lembaga yang diajukan. Kolom path_ktm merupakan kolom untuk menyimpan letak berkas ktm ketua lembaga yang diajukan. Kolom path_biodata merupakan

kolom untuk menyimpan letak berkas biodata anggota lembaga yang diajukan. Kolom `path_ad` merupakan kolom untuk menyimpan letak berkas anggaran dasar lembaga yang diajukan. Kolom `path_art` merupakan kolom untuk menyimpan letak berkas anggaran rumah tangga lembaga yang diajukan. Kolom `path_notulensi_koord` merupakan kolom untuk menyimpan letak berkas notulensi rapat koordinasi antara BEM dan DPM yang membahas pembentukan lembaga baru. Kolom `path_notulensi_sidang` merupakan kolom untuk menyimpan letak berkas notulensi sidang Musyawarah Keluarga Besar Mahasiswa Teknologi Informasi dan Ilmu Komputer (MKBMTIHK) yang membahas pembentukan lembaga baru. Kolom `path_konsideran_sidang` merupakan kolom untuk menyimpan letak berkas konsideran sidang MKBMTIHK yang membahas pembentukan lembaga baru. Kolom `no_pengajuan` merupakan kolom untuk menyimpan nomor pengajuan dari lembaga tersebut. Kolom `tanggal_koor` merupakan kolom untuk menyimpan tanggal pelaksanaan rapat koordinasi antar BEM dan DPM. Kolom `nim_ketua` merupakan kolom untuk menyimpan nim dari ketua lembaga yang diajukan. Kolom `email` merupakan kolom untuk menyimpan email dari ketua lembaga yang diajukan. Kolom `no_hp` merupakan kolom untuk menyimpan nomor *handphone* dari ketua lembaga yang diajukan.

4.2.12 Tabel Peserta_oprek

Name	Type	Length	Decimals	Not null	
▶ id	int	11	0	<input checked="" type="checkbox"/>	🔑 1
status	tinyint	4	0	<input type="checkbox"/>	
mahasiswa_nim	varchar	20	0	<input checked="" type="checkbox"/>	
jabatan_panitia_id	int	11	0	<input checked="" type="checkbox"/>	
bidang_panitia_id	int	11	0	<input checked="" type="checkbox"/>	
jabatan_terima_id	int	11	0	<input type="checkbox"/>	
bidang_terima_id	int	11	0	<input type="checkbox"/>	
oprek_id	int	11	0	<input checked="" type="checkbox"/>	
no_hp	varchar	20	0	<input type="checkbox"/>	
email	varchar	45	0	<input type="checkbox"/>	

Gambar 4.12 Screenshot tabel peserta_oprek

Gambar 4.12 merupakan atribut dari tabel peserta_oprek. Tabel ini digunakan untuk menyimpan data seluruh peserta *open recruitment* yang ada pada LKM PTIHK. Kolom `id` merupakan *primary key* dari tabel peserta_oprek. Kolom

status merupakan kolom untuk menyimpan status peserta *open recruitment*. Kolom mahasiswa_nim merupakan *foreign key* dari tabel mahasiswa. Kolom jabatan_panitia_id merupakan *foreign key* dari tabel jabatan_panitia. Kolom bidang_panitia_id merupakan *foreign key* dari tabel bidang_panitia. Kolom jabatan_terima_id merupakan id dari tabel jabatan_panitia setelah dinyatakan diterima. Kolom bidang_terima_id merupakan id dari tabel bidang_panitia setelah dinyatakan diterima. Kolom oprek_id merupakan *foreign key* dari tabel oprek. Kolom no_hp merupakan kolom untuk menyimpan nomor *handphone* dari peserta *open recruitment*. Kolom email merupakan kolom untuk menyimpan email dari peserta *open recruitment*.



4.2.13 Tabel Program_kerja

Name	Type	Length	Decimals	Not null	
▶ id	int	11	0	<input checked="" type="checkbox"/>	🔑 1
nama	varchar	200	0	<input type="checkbox"/>	
deskripsi	text	0	0	<input type="checkbox"/>	
lembaga_id	int	11	0	<input checked="" type="checkbox"/>	
status	tinyint	4	0	<input type="checkbox"/>	
created_time	timestamp	0	0	<input type="checkbox"/>	
m_departemen_id	int	11	0	<input checked="" type="checkbox"/>	
m_divisi_id	int	11	0	<input checked="" type="checkbox"/>	
nim_pic	varchar	20	0	<input type="checkbox"/>	
nim_kapel	varchar	20	0	<input type="checkbox"/>	
perkiraan_pelaksanaan	date	0	0	<input type="checkbox"/>	
perkiraan_dana	varchar	45	0	<input type="checkbox"/>	
sasaran	varchar	100	0	<input type="checkbox"/>	
latar_belakang	text	0	0	<input type="checkbox"/>	
tujuan	text	0	0	<input type="checkbox"/>	
user_id	int	11	0	<input checked="" type="checkbox"/>	
tanggal_mulai_pelaksanaan	date	0	0	<input type="checkbox"/>	
tanggal_selesai_pelaksanaan	date	0	0	<input type="checkbox"/>	
path_proposal	varchar	100	0	<input type="checkbox"/>	
path_teknis	varchar	100	0	<input type="checkbox"/>	
persen_tanggal	int	11	0	<input type="checkbox"/>	
persen_proposal	int	11	0	<input type="checkbox"/>	
persen_teknis	int	11	0	<input type="checkbox"/>	
persen_perlengkapan	int	11	0	<input type="checkbox"/>	
persen_simulasi	int	11	0	<input type="checkbox"/>	
path_lpj	varchar	100	0	<input type="checkbox"/>	
status_lpj	tinyint	4	0	<input type="checkbox"/>	
usul_nim	varchar	20	0	<input checked="" type="checkbox"/>	
path_evaluasi	varchar	100	0	<input type="checkbox"/>	

Gambar 4.13 Screenshot tabel program_kerja

Gambar 4.13 merupakan atribut dari tabel program_kerja. Tabel ini digunakan untuk menyimpan data seluruh program kerja yang ada pada LKM PTIIK. Kolom id merupakan *primary key* dari tabel program_kerja. Kolom nama merupakan kolom untuk menyimpan nama program kerja. Kolom deskripsi merupakan kolom untuk menyimpan deskripsi program kerja. Kolom lembaga_id merupakan *foreign key* dari tabel lembaga. Kolom status merupakan kolom untuk menyimpan status dari program kerja. Kolom created_time merupakan kolom untuk menyimpan waktu pembuatan tiap data pada tabel program kerja. Kolom m_departemen_id merupakan *foreign key* dari tabel m_departemen. Kolom m_divisi_id merupakan *foreign key* dari tabel m_divisi. Kolom nim_pic merupakan

kolom untuk menyimpan NIM dari mahasiswa yang ditunjuk sebagai *Person In Charge* pada tiap program kerja. Kolom *nim_kapel* merupakan kolom untuk menyimpan NIM dari mahasiswa yang ditunjuk sebagai ketua pelaksana pada tiap program kerja. Kolom *perkiraan_pelaksanaan* merupakan kolom untuk menyimpan tanggal perkiraan pelaksanaan program kerja. Kolom *perkiraan_dana* merupakan kolom untuk menyimpan data perkiraan dana penyelenggaraan dari tiap program kerja. Kolom *sasaran* merupakan kolom untuk menyimpan data sasaran tiap program kerja. Kolom *latar_belakang* merupakan kolom untuk menyimpan data latar belakang program kerja. Kolom *tujuan* merupakan kolom untuk menyimpan data tujuan program kerja. Kolom *user_id* merupakan *foreign key* dari dari tabel *user*. Kolom *tanggal_mulai_pelaksanaan* merupakan kolom untuk menyimpan tanggal mulai pelaksanaan tiap program kerja. Kolom *tanggal_selesai_pelaksanaan* merupakan kolom untuk menyimpan tanggal selesai pelaksanaan tiap program kerja. Kolom *path_proposal* merupakan kolom untuk menyimpan letak berkas proposal program kerja. Kolom *path_teknis* merupakan kolom untuk menyimpan letak berkas teknis pelaksanaan program kerja. Kolom *persen_tanggal* merupakan kolom untuk menyimpan persentase dari penetapan tanggal. Kolom *persen_proposal* merupakan kolom untuk menyimpan persentase dari proposal. Kolom *persen_teknis* merupakan kolom untuk menyimpan persentase dari teknis pelaksanaan. Kolom *persen_perlengkapan* merupakan kolom untuk menyimpan persentase dari persiapan perlengkapan pelaksanaan program kerja. Kolom *persen_simulasi* merupakan kolom untuk menyimpan persentase dari simulasi persiapan pelaksanaan program kerja. Kolom *path_lpj* merupakan kolom untuk menyimpan letak berkas LPJ program kerja. Kolom *status_lpj* merupakan kolom untuk menyimpan status LPJ pelaksanaan program kerja. Kolom *usul_nim* merupakan kolom untuk menyimpan nomor induk mahasiswa yang mengusulkan program kerja. Kolom *path_evaluasi* merupakan kolom untuk menyimpan letak berkas evaluasi pelaksanaan program kerja. Kolom *status_evaluasi* merupakan kolom untuk menyimpan status evaluasi pelaksanaan program kerja. Kolom *status_terlaksana* merupakan kolom untuk menyimpan status pelaksanaan program kerja.

4.3 Implementasi Class

Setiap *class* yang telah dirancang pada proses perancangan direalisasikan pada sebuah file dengan ekstensi *.php*. Pada Tabel 4.1 berikut ini merupakan hubungan antara *class* yang dirancang dengan *file* yang dibuat.

Tabel 4.1 Hubungan antara *class* dan *file*

No	Package	Nama Class	Lokasi File	Nama File
1	Controller	CI_Controller	/system/core/	Controller.php
2	Controller	Dashboard	/application/controllers/	dashboard.php
3	Controller	General	/application/controllers/	general.php
4	Controller	Jadwal	/application/controllers/	jadwal.php
5	Controller	Lembaga	/application/controllers/	lembaga.php
6	Controller	User	/application/controllers/	user.php
7	Controller	Settings	/application/controllers/	settings.php
8	Controller	Notifikasi	/application/controllers/	notifikasi.php
9	Controller	Mahasiswa	/application/controllers/	mahasiswa.php
10	Controller	Proker	/application/controllers/	proker.php
11	Controller	Oprek	/application/controllers/	oprek.php
12	Model	CI_Model	/system/core/	Model.php
13	Model	M_Feedback	/application/models/	m_feedback.php
14	Model	M_Jadwal	/application/models/	m_jadwal.php
15	Model	M_Lembaga	/application/models/	m_lembaga.php
16	Model	M_Mahasiswa	/application/models/	m_mahasiswa.php
17	Model	M_Notifikasi	/application/models/	m_notifikasi.php
18	Model	M_Oprek	/application/models/	m_oprek.php
19	Model	M_Proker	/application/models/	m_proker.php
20	Model	M_Saran	/application/models/	m_saran.php
21	Model	M_Setting	/application/models/	m_setting.php
22	Model	M_User	/application/models/	m_user.php

4.4 Implementasi Antar Muka

Antar muka aplikasi ERP LKM PTIIK merupakan tampilan aplikasi yang dilihat langsung oleh pengguna dan merupakan sarana bagi pengguna untuk berkomunikasi dengan aplikasi. Beberapa antar muka aplikasi yang akan dibahas antara lain antar muka halaman depan aplikasi, halaman pengajuan lembaga baru, halaman kelola program kerja, halaman perancangan program kerja, halaman kelola *open recruitment*, halaman daftar *open recruitment*, dan halaman melihat jadwal kegiatan.

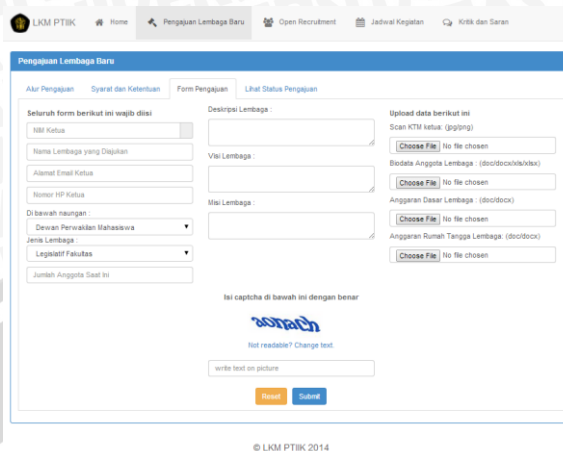
4.4.1 Halaman depan aplikasi



Gambar 4.14 Halaman depan aplikasi

Gambar 4.14 merupakan hasil implementasi antar muka pada halaman depan aplikasi. Terdapat 4 (empat) menu yang dapat dipilih oleh pengguna. Menu pengajuan lembaga baru adalah menu apabila pengguna ingin mengajukan lembaga baru pada LKM PTIIK. Menu *open recruitment* digunakan apabila pengguna ingin mengikuti *open recruitment* yang ada. Menu jadwal kegiatan digunakan untuk melihat jadwal kegiatan yang ada di LKM PTIIK. Menu kritik dan saran digunakan untuk memberikan kritik atau saran kepada lembaga mahasiswa yang ada di LKM PTIIK.

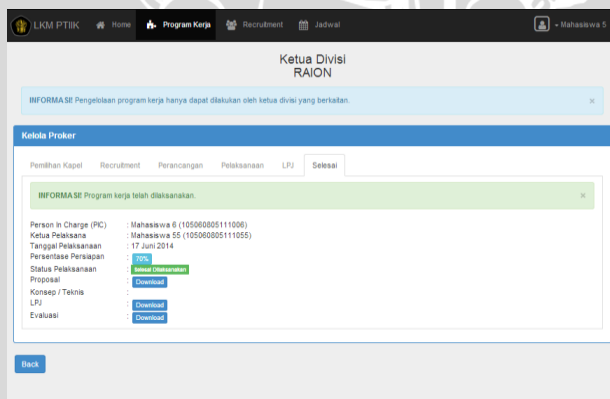
4.4.2 Halaman pengajuan lembaga baru



Gambar 4.15 Halaman pengajuan lembaga baru

Gambar 4.15 merupakan hasil implementasi antar muka untuk pengajuan lembaga baru. Pengguna cukup mempersiapkan syarat yang dibutuhkan dan mengisi form yang tersedia untuk mengajukan permohonan pembentukan lembaga baru.

4.4.3 Halaman kelola program kerja



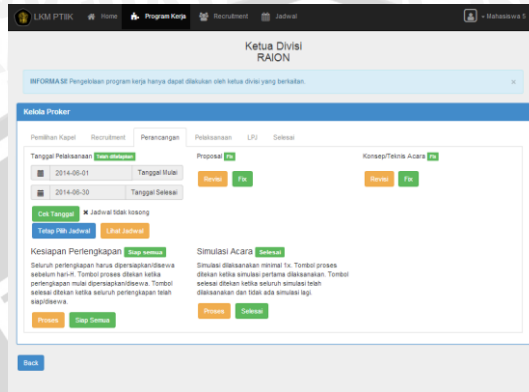
Gambar 4.16 Halaman kelola program kerja

Gambar 4.16 merupakan hasil implementasi antar muka untuk halaman pengelolaan program kerja. Terdapat 5 (lima) tahap yang harus dilalui pada pengelolaan program kerja. Tahap pertama adalah tahap pemilihan ketua pelaksana. Tahap kedua adalah tahap perekrutan panitia untuk melaksanakan program kerja tersebut. Tahap ketiga adalah perancangan program kerja secara detail. Tahap



keempat adalah pelaksanaan program kerja. Tahap terakhir adalah tahap pembuatan laporan pertanggung jawaban.

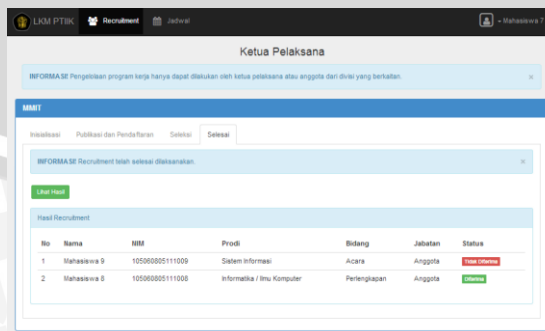
4.4.4 Halaman perancangan program kerja



Gambar 4.17 Halaman perancangan program kerja

Gambar 4.17 merupakan hasil implementasi antar muka untuk halaman perancangan program kerja. Terdapat 5 (lima) detail yang harus dipersiapkan untuk menjalankan program kerja, yaitu penentuan tanggal pelaksanaan, proposal kegiatan, konsep acara, kesiapan perlengkapan, dan simulasi sebelum program kerja dijalankan. Pada halaman ini terdapat fitur untuk melakukan pemeriksaan jadwal program kerja yang akan digunakan. Ketika jadwal yang akan digunakan sudah terdapat jadwal lain, aplikasi akan memberikan peringatan bahwa jadwal bentrok.

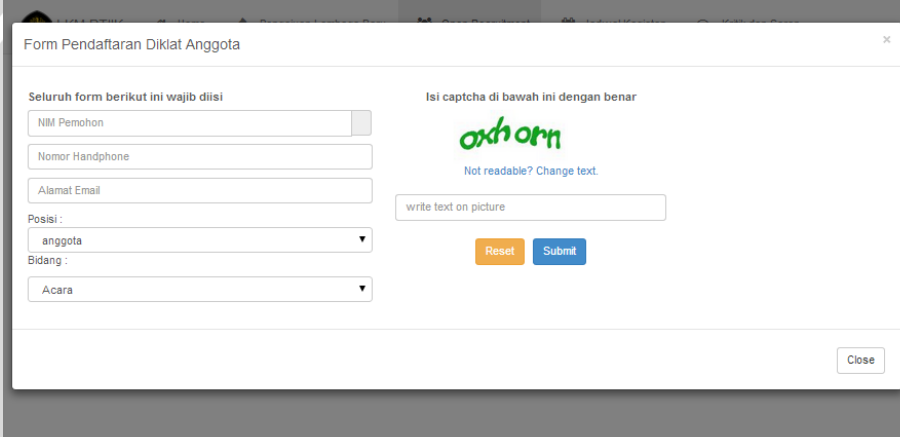
4.4.5 Halaman kelola *open recruitment*



Gambar 4.18 Halaman kelola *open recruitment*

Gambar 4.18 merupakan hasil implementasi antar muka untuk halaman pengelolaan *open recruitment*. Terdapat 3 (tiga) tahap yang harus dilalui pada pengelolaan *open recruitment*. Tahap pertama adalah tahap inisialisasi. Tahap kedua adalah tahap publikasi dan pendaftaran. Tahap ketiga adalah tahap seleksi. Pada tahap seleksi terdapat fitur aplikasi untuk melihat kontribusi peserta *recruitment*. Sehingga pengambil keputusan dapat mengetahui apakah peserta *recruitment* sedang mengikuti kegiatan lain atau tidak.

4.4.6 Halaman daftar *open recruitment*



Form Pendaftaran Diklat Anggota

Seluruh form berikut ini wajib diisi

NIM Pemohon

Nomor Handphone

Alamat Email

Posisi :

anggota

Bidang :

Acara

Isi captcha di bawah ini dengan benar

oxhorn

Not readable? Change text.

write text on picture

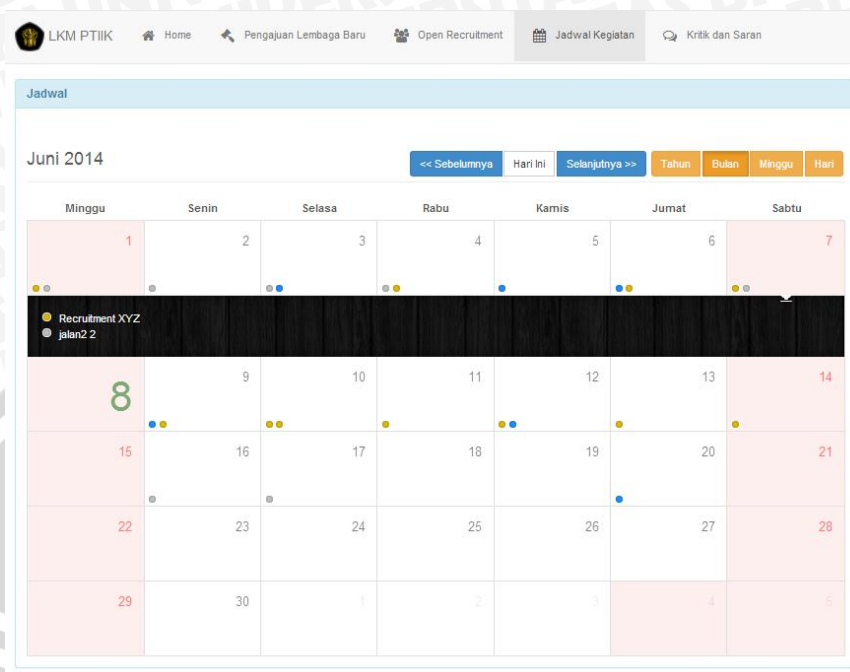
Reset Submit

Close

Gambar 4.19 Halaman daftar *open recruitment*

Gambar 4.19 merupakan hasil implementasi antar muka untuk halaman pendaftaran *open recruitment*. Mahasiswa cukup mengisi form yang disediakan untuk mendaftarkan dirinya pada sebuah *open recruitment*.

4.4.7 Halaman melihat jadwal kegiatan



Gambar 4.20 Halaman melihat jadwal kegiatan

Gambar 4.20 merupakan hasil implementasi antar muka untuk halaman jadwal kegiatan. Halaman ini menampilkan seluruh jadwal kegiatan yang akan dilakukan oleh setiap lembaga yang ada di LKM PTIIK.

BAB V

PENGUJIAN DAN ANALISIS

Bab ini membahas mengenai pengujian dan analisa yang dilakukan terhadap aplikasi ERP LKM PTIIK. Pengujian pada aplikasi ini dilakukan menggunakan metode *Funtionast Test*, *Usability Test* dan *User Acceptance Test*. *Usability Test* dan *User Acceptance Test* dilakukan oleh pengguna aplikasi ERP LKM PTIIK dengan mencoba menggunakan aplikasi secara langsung dan mengisi kuisisioner yang disediakan.

5.1 Hasil *Functional Test*

Functional test dilakukan untuk mengetahui bahwa aplikasi dapat berfungsi dengan baik dan sesuai dengan kebutuhan yang ada. *Functional test* dilakukan berdasarkan kebutuhan fungsional yang telah ditentukan pada analisa kebutuhan. Hasil dari *functional test* yang telah dilakukan dinyatakan bahwa seluruh kebutuhan fungsional aplikasi telah terpenuhi dan 100% valid. Hasil rinci dari *functional test* yang dilakukan dapat dilihat pada Lampiran E.

5.2 Hasil dan Analisa *Usability Test*

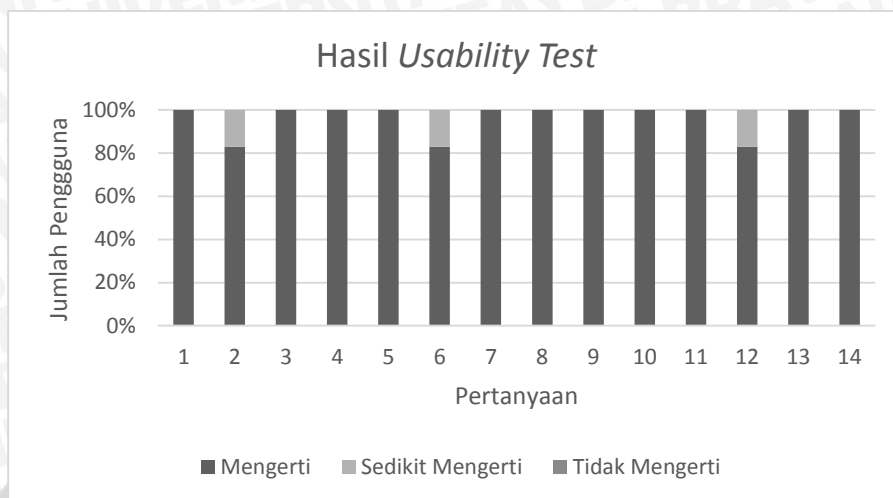
Usability test dilakukan untuk mengetahui kemudahan penggunaan aplikasi oleh pengguna. Tabel 5.1 berikut ini merupakan daftar pertanyaan yang digunakan pada *usability test* yang telah dilakukan.

Tabel 5.1 Daftar Pertanyaan *usability test*

No	Pertanyaan
1	Apakah Anda mengerti bagaimana cara mengajukan lembaga baru melalui aplikasi ini?
2	Apakah Anda mengerti bagaimana cara memproses pengajuan lembaga yang ada melalui aplikasi ini?
3	Apakah Anda mengerti bagaimana cara menambah program kerja pada aplikasi ini?

4	Apakah Anda mengerti bagaimana cara melihat program kerja yang ada melalui aplikasi ini?
5	Apakah Anda mengerti bagaimana cara mengetahui <i>progress</i> program kerja yang ada?
6	Apakah Anda mengerti bagaimana cara mengelola program kerja melalui aplikasi ini?
7	Apakah Anda mengerti bagaimana cara melakukan <i>recruitment</i> melalui aplikasi ini?
8	Apakah Anda mengerti bagaimana cara melakukan pendaftaran <i>recruitment</i> melalui aplikasi ini?
9	Apakah Anda mengerti bagaimana cara mengetahui keaktifan peserta <i>recruitment</i> melalui aplikasi ini?
10	Apakah Anda mengerti bagaimana cara melakukan seleksi peserta <i>recruitment</i> melalui aplikasi ini?
11	Apakah Anda mengerti bagaimana cara melihat hasil <i>recruitment</i> yang telah diikuti?
12	Apakah Anda mengerti bagaimana cara mengetahui jadwal program kerja yang diinginkan bentrok dengan program kerja lain atau tidak?
13	Apakah Anda mengerti bagaimana cara melihat seluruh jadwal kegiatan lembaga yang ada melalui aplikasi ini?
14	Secara keseluruhan, apakah Anda mengerti cara menggunakan aplikasi ini?

Pengguna aplikasi pada pengujian ini adalah ketua lembaga dari DPM, BEM, HMSI, HMIF, POROS, dan Display. Hasil yang didapatkan dari pengujian ini dapat dilihat pada gambar 5.1 berikut ini.



Gambar 5.1 Chart Hasil Usability Test

Gambar 5.1 merupakan hasil dari *usability test* yang telah dilakukan. Seperti terlihat pada gambar 5.1 terdapat 3 (tiga) pertanyaan mengenai bagian aplikasi yang sedikit dimengerti oleh 17% pengguna yaitu pada pertanyaan 2, 6, dan 12. Pengguna tidak memberikan penjelasan rinci mengenai bagian yang belum dimengerti. Sehingga peneliti kesulitan untuk mengidentifikasinya. Tetapi secara keseluruhan, pengguna yang menggunakan aplikasi menyatakan mengerti penggunaan aplikasi ini dengan baik.

5.3 Hasil dan Analisa User Acceptance Test

Usability test dilakukan untuk mengetahui kemudahan penggunaan aplikasi oleh pengguna. Tabel 5.2 berikut ini merupakan hasil dari *user acceptance test* yang telah dilakukan.

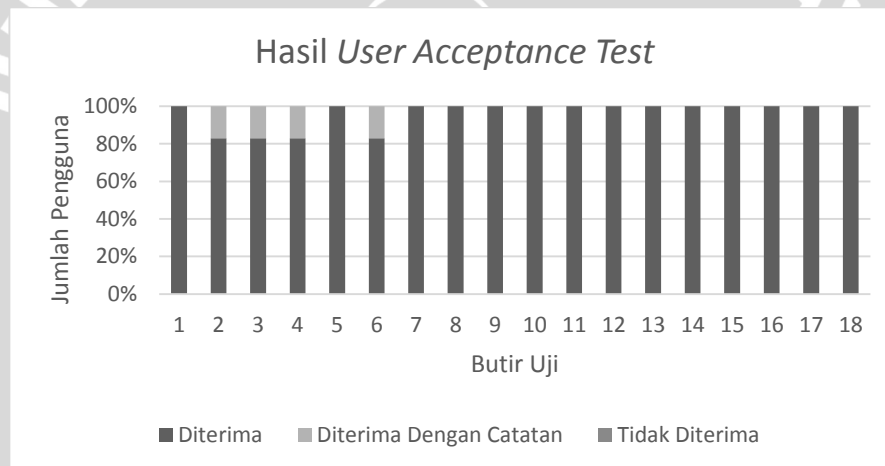
Tabel 5.2 Daftar uji *User Acceptance Test*

No	Butir Uji	Deskripsi
1	Pengajuan Lembaga Baru	Menguji apakah aplikasi dapat menerima pengajuan lembaga baru
2	Form input pengajuan lembaga baru	Menguji apakah form pengajuan lembaga baru telah sesuai dengan yang dibutuhkan

3	Pengambilan keputusan pengajuan lembaga baru	Menguji apakah aplikasi memberikan opsi untuk menerima atau menolak pengajuan yang dilakukan
4	Alur pengajuan lembaga	Menguji apakah alur pengajuan lembaga pada aplikasi sudah sesuai
5	Membuat Program Kerja	Menguji apakah aplikasi dapat membuat program kerja sesuai dengan yang diinginkan
6	Form input penambahan program kerja	Menguji apakah form penambahan program kerja telah sesuai dengan yang dibutuhkan
7	Pengawasan program kerja	Apakah aplikasi dapat memberikan info terbaru mengenai program kerja yang ada sehingga mempermudah pengawasan program kerja yang ada
8	Alur program kerja	Alur program kerja dari awal hingga selesai pada aplikasi sesuai dengan yang ada
9	Membuat recruitment	Apakah aplikasi dapat membuat recruitment sesuai yang diinginkan
10	Penentuan jadwal recruitment	Apakah aplikasi dapat menentukan jadwal recruitment yang ada
12	Form recruitment	Apakah form recruitment pada aplikasi telah sesuai dengan yang dibutuhkan
13	Daftar recruitment	Apakah mahasiswa dapat mendaftar recruitment yang ada melalui aplikasi ini
14	Kepanitiaan ganda atau lebih	Apakah aplikasi dapat menampilkan peserta yang mengikuti kepanitiaan ganda atau lebih
15	Seleksi peserta	Pengujian untuk mengetahui apakah aplikasi dapat melakukan seleksi peserta recruitment
16	Hasil recruitment	Pengujian untuk menguji apakah aplikasi dapat menampilkan hasil recruitment yang sesuai
17	Menampilkan jadwal bentrok ketika memilih	Pengujian yang dilakukan untuk menguji apakah aplikasi dapat memberi peringatan dan

	tanggal untuk pelaksanaan program kerja	menampilkan kegiatan yang ada ketika jadwal yang dipilih tidak kosong
18	Menampilkan jadwal kegiatan	Pengujian dilakukan untuk menguji apakah aplikasi dapat menampilkan jadwal kegiatan dari seluruh lembaga yang ada

Pengguna aplikasi pada pengujian ini adalah ketua lembaga dari DPM, BEM, HMSI, HMIF, POROS, dan Display. Hasil yang didapatkan dari pengujian ini dapat dilihat pada gambar 5.2 berikut ini.



Gambar 5.2 Chart Hasil *User Acceptance Test*

Gambar 5.2 merupakan hasil dari *user acceptance test* yang telah dilakukan. Seperti terlihat pada gambar 5.2 terdapat 4 (empat) butir uji yang diterima dengan catatan oleh 17% pengguna yaitu pada butir uji 2, 3, 4, dan 6. Pada butir uji 2, salah satu pengguna memberikan catatan bahwa pilihan lembaga yang menaungi lembaga baru pada *form* pengajuan lembaga untuk ditentukan saja. Hal ini disebabkan karena sampai saat ini lembaga yang dapat dibentuk pada LKM PTIIK hanya dapat berada dibawah naungan BEM TIIK dengan jenis LSO. Pada butir uji 3, salah satu pengguna memberikan catatan untuk memberikan keterangan ketika pengajuan lembaga baru yang ada ditolak. Sehingga pemohon dapat mengetahui alasan mengapa lembaga baru yang diajukan tidak diterima. Pada butir uji 4, salah

satu pengguna memberikan catatan agar aplikasi dapat menerima banyak *input* data pengajuan. Catatan tersebut timbul karena pengguna menilai bahwa aplikasi hanya dapat menerima satu pengajuan lembaga. Pada butir uji 6 pengguna memberikan catatan berupa saran agar aplikasi dapat melakukan pengecekan *detail* proker secara otomatis. Hal tersebut karena setiap lembaga terkadang memiliki program kerja yang sama. Saran yang diberikan pengguna dimaksudkan agar aplikasi dapat memberikan peringatan ketika program kerja yang akan dibuat sama dengan program kerja lainnya. Catatan yang diberikan oleh pengguna seluruhnya berupa saran agar aplikasi dapat bekerja dengan baik. Tetapi secara keseluruhan, aplikasi ini telah sesuai dengan yang dibutuhkan dan dapat diterima oleh pengguna untuk diimplementasikan.



BAB VI PENUTUP

6.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi dan pengujian yang dilakukan pada aplikasi, maka dapat diambil kesimpulan sebagai berikut :

1. Proses bisnis pembentukan lembaga mahasiswa, perancangan program kerja, penjadwalan program kerja, dan *open recruitment* di LKM PTIIK dapat dilihat pada BPMN hasil analisa (Lampiran A).
2. Perancangan aplikasi ERP dilakukan dengan melakukan analisa terlebih dahulu, kemudian hasil analisa digunakan sebagai dasar dalam perancangan sistem.
3. Implementasi aplikasi ERP dilakukan berdasarkan hasil analisa dan perancangan serta menggunakan bahasa pemrograman PHP dengan menggunakan *framework Codeigniter*, hal ini sesuai dengan sifat aplikasi ERP yang selalu membutuhkan komunikasi antara *server* dan *client*.
4. Berdasarkan hasil *functional testing* dapat disimpulkan bahwa seluruh kebutuhan fungsional aplikasi telah terpenuhi dan 100% valid.
5. Berdasarkan hasil *usability testing* dapat disimpulkan bahwa penggunaan aplikasi ini dapat dimengerti dengan baik.
6. Berdasarkan hasil *user acceptance testing* dapat disimpulkan bahwa aplikasi ini diterima dan telah sesuai dengan yang dibutuhkan.

6.2 Saran

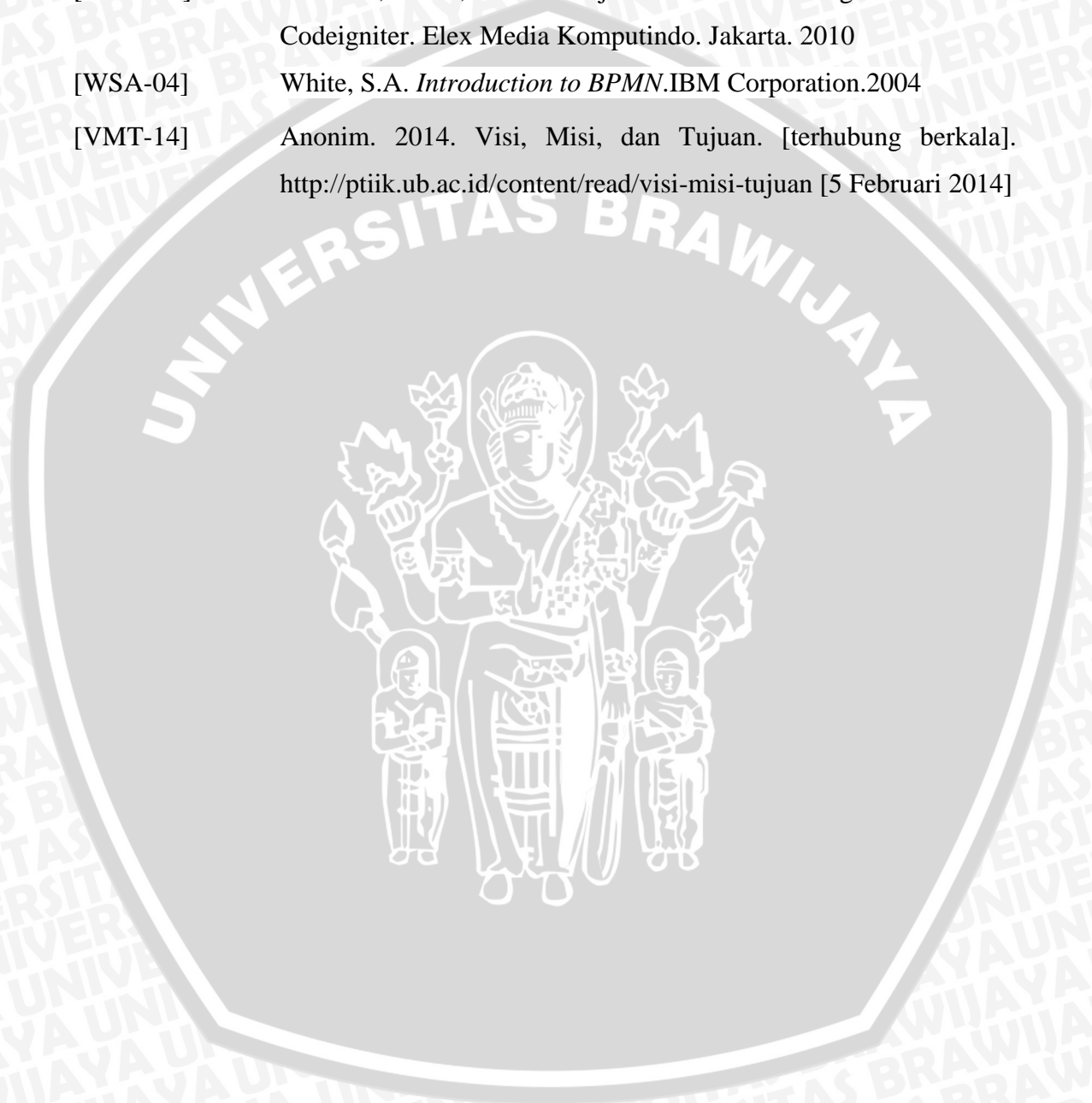
Saran yang dapat diberikan untuk pengembangan aplikasi ini antara lain :

1. Pada pengembangan selanjutnya dapat ditambahkan fitur-fitur untuk menunjang kegiatan organisasi lainnya, seperti fitur untuk dokumentasi surat dan fitur seleksi peserta *recruitment* secara otomatis menggunakan metode tertentu.
2. Data mahasiswa yang digunakan pada aplikasi ini dapat menggunakan *web service* yang disediakan oleh Universitas Brawijaya.
3. Aplikasi dapat dikembangkan sehingga dapat mengakomodasi struktur organisasi dan hak akses yang dinamis.

DAFTAR PUSTAKA

- [ANH-10] Anhar, ST. Panduan Menguasai PHP & MySQL Secara Otodidak. Mediakita. Jakarta. 2010
- [ATM-10] B. B. Agarwal, S. P. Tayal, Mahesh Gupta. *Software Engineering and Testing*. Jones & Bartlett Learning. United States of America. 2010
- [BCM-11] Barnum, Carol M. *Usability Testing Essentials : Ready, Set, Test*. Elsevier. United States of America. 2011
- [DGR-10] Dantes, G.R., Hasibuan, Z.A. *Measurements of Key Success Factors (KSFs) on Enterprise Resource Planning (ERP) Adoption*. IBIMA Business Review Journal. 2010.
- [GDC-10] Gunn, Robert., Durkin, Chris. *Social Entrepreneurship: A Skills Approach*. Policy Press. United Kingdom. 2010
- [JMP-14] Anonim. 2014. Jumlah Mahasiswa Pertahun. [terhubung berkala]. <http://ptiik.ub.ac.id/content/read/jumlah-mahasiswa-pertahun> [4 Februari 2014]
- [ODE-00] O'Leary ,D. E. *Enterprise Resource Planning Systems*. 2000
- [RSP-10] R. S. Pressman, *Software Engineering: A Practitioner's Approach*. McGraw-Hill. 2010.
- [SLM-13] Salahuddin, M. dan S, Rosa A. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Informatika. 2013.
- [SMG-09] McGary, Sean. 2009. *Introduction to the Codeigniter Framework*. [terhubung berkala]. <http://people.rit.edu/~spm8975/539/project2/post/Introduction-to-the-Codeigniter-PHP-Framework> [9 Februari 2014]
- [WDP-08] Widhiartha, Putu. *Extreme Programming – Melakukan Pengembangan Perangkat Lunak dengan Lebih Sederhana*. IlmuKomputer.com. 2008

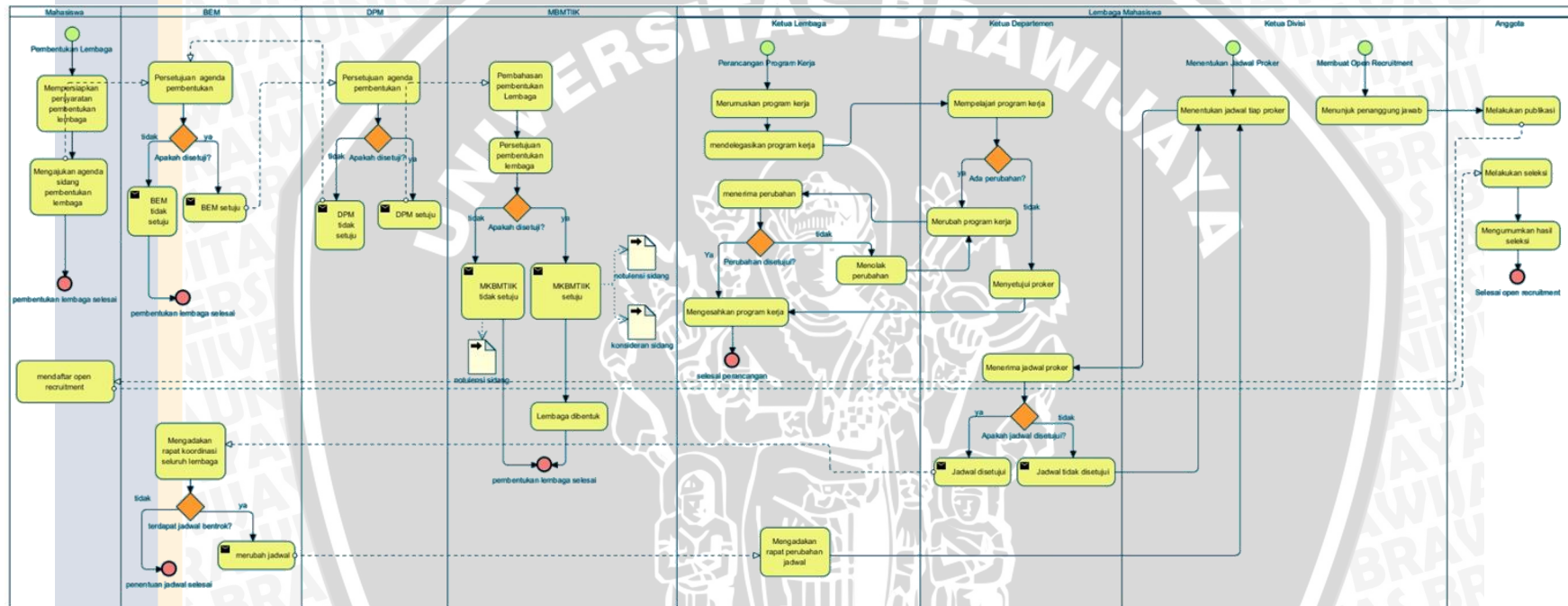
- [WLD-13] Wells, Don. 2013. *Extreme Programming: A Gentle Introduction*. [terhubung berkala]. <http://www.extremeprogramming.org/> [11 Februari 2014]
- [WRD-10] Wardana, S.Hut, M.Si. *Menjadi Master PHP dengan Framework Codeigniter*. Elex Media Komputindo. Jakarta. 2010
- [WSA-04] White, S.A. *Introduction to BPMN*. IBM Corporation. 2004
- [VMT-14] Anonim. 2014. *Visi, Misi, dan Tujuan*. [terhubung berkala]. <http://ptiik.ub.ac.id/content/read/visi-misi-tujuan> [5 Februari 2014]



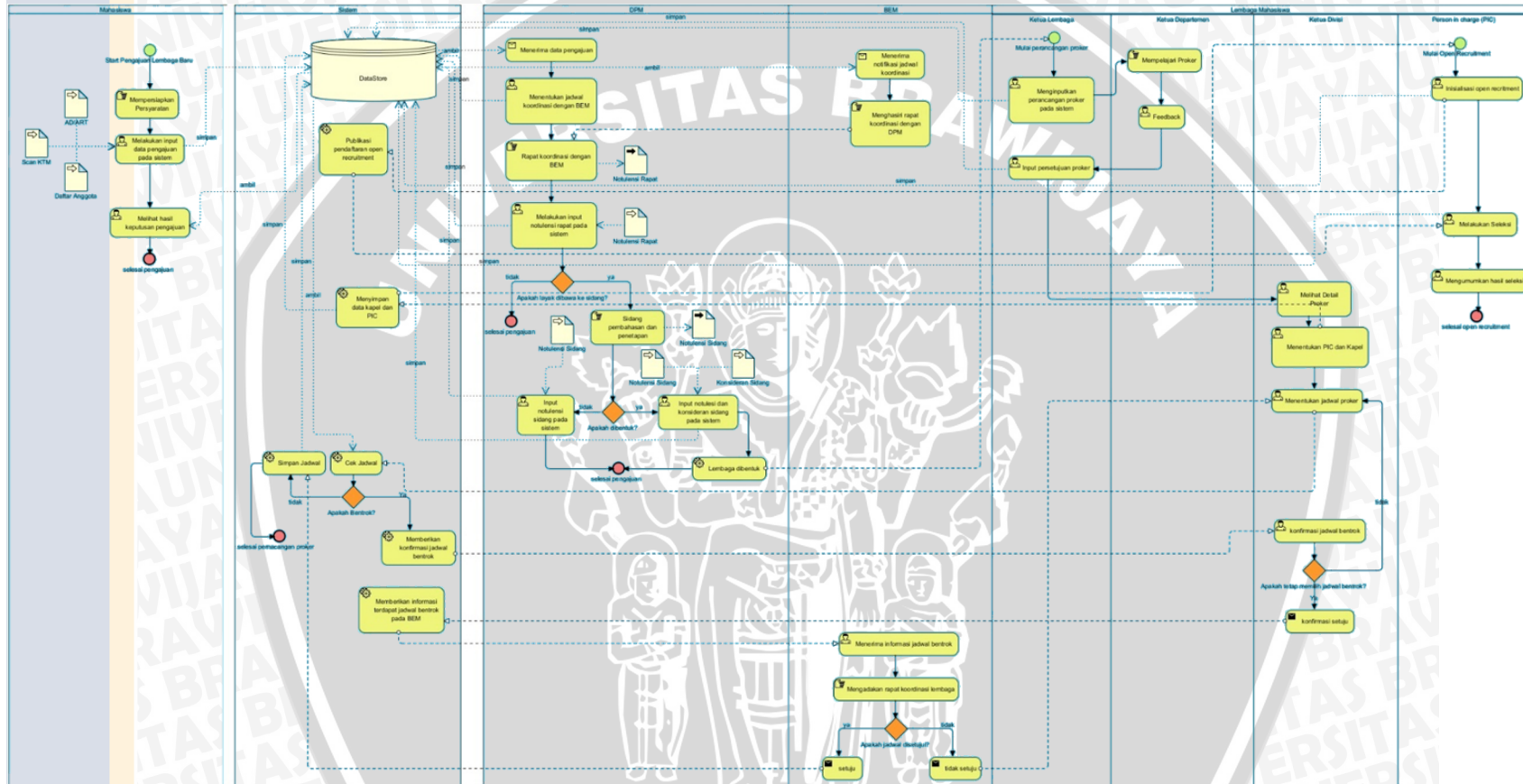
LAMPIRAN

LAMPIRAN A

1. BPMN Sebelum Ada Sistem



2. BPMN Setelah Ada Sistem



LAMPIRAN B

1. Skenario *Use Case* Mengajukan pembentukan lembaga

<i>Use Case Name</i>	<i>Use Case</i> mengajukan pembentukan lembaga
<i>Short Description</i>	Aplikasi menampilkan form untuk mengajukan lembaga baru.
<i>Aktor</i>	Pemohon
<i>Pre Condition</i>	Pemohon masuk ke halaman depan aplikasi dan memilih <i>menu</i> pengajuan pembentukan lembaga mahasiswa baru.
<i>Post Condition</i>	Sistem menampilkan pesan yang berisi informasi nomor pengajuan dan pesan pengajuan sudah diterima.
<i>Normal Flow</i>	Pemohon mengisi form pengajuan yang disediakan secara lengkap kemudian menekan tombol <i>submit</i> .
<i>Alternate Flow</i>	-

2. Skenario *Use Case* Mengelola data pengajuan pembentukan lembaga

<i>Use Case Name</i>	<i>Use Case</i> mengelola data pengajuan pembentukan lembaga
<i>Short Description</i>	Aplikasi menampilkan menu pengelolaan data pengajuan pembentukan lembaga mahasiswa.
<i>Aktor</i>	Ketua DPM
<i>Pre Condition</i>	Memilih menu Kelembagaan
<i>Post Condition</i>	Berada di halaman utama kelembagaan.
<i>Normal Flow</i>	Melakukan pengelolaan data pengajuan seperti melakukan perbaikan terhadap <i>detail</i> pengajuan.
<i>Alternate Flow</i>	-

3. Skenario *Use Case* Mengajukan program kerja

<i>Use Case Name</i>	<i>Use Case</i> mengajukan program kerja
<i>Short Description</i>	Aplikasi menampilkan <i>form</i> untuk mengusulkan program kerja baru.
<i>Aktor</i>	Ketua Lembaga atau Ketua Departemen
<i>Pre Condition</i>	Menekan tombol usulkan program kerja pada halaman program kerja
<i>Post Condition</i>	Program kerja diusulkan
<i>Normal Flow</i>	<ol style="list-style-type: none"> 1. Menampilkan <i>form</i> program kerja. 2. Mengisi seluruh <i>field</i> yang terdapat pada <i>form</i>. 3. Menekan tombol kirim.
<i>Alternate Flow</i>	<ol style="list-style-type: none"> 1. Menekan tombol close 2. Tidak jadi mengusulkan program kerja

4. Skenario *Use Case* Menolak pengajuan program kerja

<i>Use Case Name</i>	<i>Use Case</i> menolak pengajuan program kerja
<i>Short Description</i>	Aplikasi menampilkan <i>list</i> program kerja yang diajukan, pada setiap baris pengajuan program kerja terdapat tombol tolak.
<i>Aktor</i>	Ketua Lembaga
<i>Pre Condition</i>	Memilih menu program kerja
<i>Post Condition</i>	Pengajuan ditolak
<i>Normal Flow</i>	<ol style="list-style-type: none"> 4. Menekan tombol tolak pada <i>list</i> pengajuan program kerja. 5. Menampilkan pesan konfirmasi penghapusan. 6. Menekan tombol ok pada pesan konfirmasi.
<i>Alternate Flow</i>	<ol style="list-style-type: none"> 3. Menekan tombol cancel pada pesan konfirmasi 4. Pengajuan program kerja batal ditolak

5. Skenario Use Case Mengubah *progress* program kerja

<i>Use Case Name</i>	<i>Use Case</i> mengubah <i>progress</i> program kerja
<i>Short Description</i>	Aplikasi menampilkan program kerja yang sedang berjalan. Pengguna dapat mengubah <i>progress</i> dari program kerja tersebut.
<i>Aktor</i>	Ketua Divisi
<i>Pre Condition</i>	Memilih menu <i>progress</i> program kerja
<i>Post Condition</i>	<i>Progress</i> program kerja berubah
<i>Normal Flow</i>	<ol style="list-style-type: none"> 1. Menampilkan list program kerja yang sedang berjalan. 2. Memilih salah satu program kerja 3. Mengubah <i>progress</i> program kerja 4. Menekan tombol simpan
<i>Alternate Flow</i>	-

6. Skenario Use Case Mengelola *open recruitment*

<i>Use Case Name</i>	<i>Use Case</i> mengelola <i>open recruitment</i>
<i>Short Description</i>	Aplikasi menampilkan menu pengelolaan <i>open recruitment</i> .
<i>Aktor</i>	PIC dan Ketua Pelaksana
<i>Pre Condition</i>	Memilih menu Open Recruitment
<i>Post Condition</i>	Masuk ke halaman utama pengelolaan <i>open recruitment</i>
<i>Normal Flow</i>	Melakukan pengelolaan <i>open recruitment</i> seperti menambah, menghapus, dan mengubah <i>open recruitment</i> .
<i>Alternate Flow</i>	-

7. Skenario *Use Case* Mendaftar peserta *open recruitment*

<i>Use Case Name</i>	<i>Use Case</i> mendaftar peserta <i>open recruitment</i>
<i>Short Description</i>	Aplikasi menampilkan form pendaftaran pada <i>open recruitment</i> . Pengguna cukup mengisi <i>form</i> dengan lengkap dan menekan tombol daftar untuk melakukan pendaftaran.
<i>Aktor</i>	Mahasiswa PTIIK
<i>Pre Condition</i>	Masuk ke halaman pendaftaran <i>open recruitment</i>
<i>Post Condition</i>	Menampilkan pesan berhasil mendaftar
<i>Normal Flow</i>	<ol style="list-style-type: none"> 1. Mengisi <i>form</i> dengan lengkap 2. Menekan tombol daftar
<i>Alternate Flow</i>	-

8. Skenario *Use Case* Melakukan seleksi peserta *open recruitment*

<i>Use Case Name</i>	<i>Use Case</i> melakukan seleksi peserta <i>open recruitment</i>
<i>Short Description</i>	Aplikasi menampilkan list peserta yang mendaftar disertai dengan <i>checkbox</i> diterima pada setiap peserta.
<i>Aktor</i>	PIC dan Ketua Pelaksana
<i>Pre Condition</i>	Telah melewati batas akhir pendaftaran
<i>Post Condition</i>	Menampilkan list peserta <i>open recruitment</i>
<i>Normal Flow</i>	<ol style="list-style-type: none"> 1. Menampilkan list peserta <i>open recruitment</i> 2. Centang <i>checkbox</i> terima pada list peserta
<i>Alternate Flow</i>	<i>checkbox</i> tidak dicentang

9. Skenario Use Case Melihat kegiatan yang diikuti peserta

<i>Use Case Name</i>	Use Case melihat kegiatan yang diikuti peserta
<i>Short Description</i>	Aplikasi menampilkan kegiatan yang pernah diikuti peserta baik yang sedang berlangsung maupun yang telah selesai
<i>Aktor</i>	PIC dan Ketua Pelaksana
<i>Pre Condition</i>	Telah melewati batas akhir pendaftaran
<i>Post Condition</i>	Menampilkan list kegiatan yang pernah diikuti oleh peserta
<i>Normal Flow</i>	<ol style="list-style-type: none"> 1. Menampilkan list peserta <i>open recruitment</i> 2. Menekan tombol kontribusi yang terdapat pada tiap peserta
<i>Alternate Flow</i>	-

10. Skenario Use Case Melihat jadwal

<i>Use Case Name</i>	Use Case melihat jadwal
<i>Short Description</i>	Aplikasi menampilkan seluruh jadwal kegiatan yang akan dilaksanakan.
<i>Aktor</i>	Seluruh aktor
<i>Pre Condition</i>	Masuk ke menu jadwal kegiatan
<i>Post Condition</i>	Menampilkan <i>list</i> jadwal kegiatan
<i>Normal Flow</i>	Menampilkan <i>list</i> jadwal kegiatan
<i>Alternate Flow</i>	-

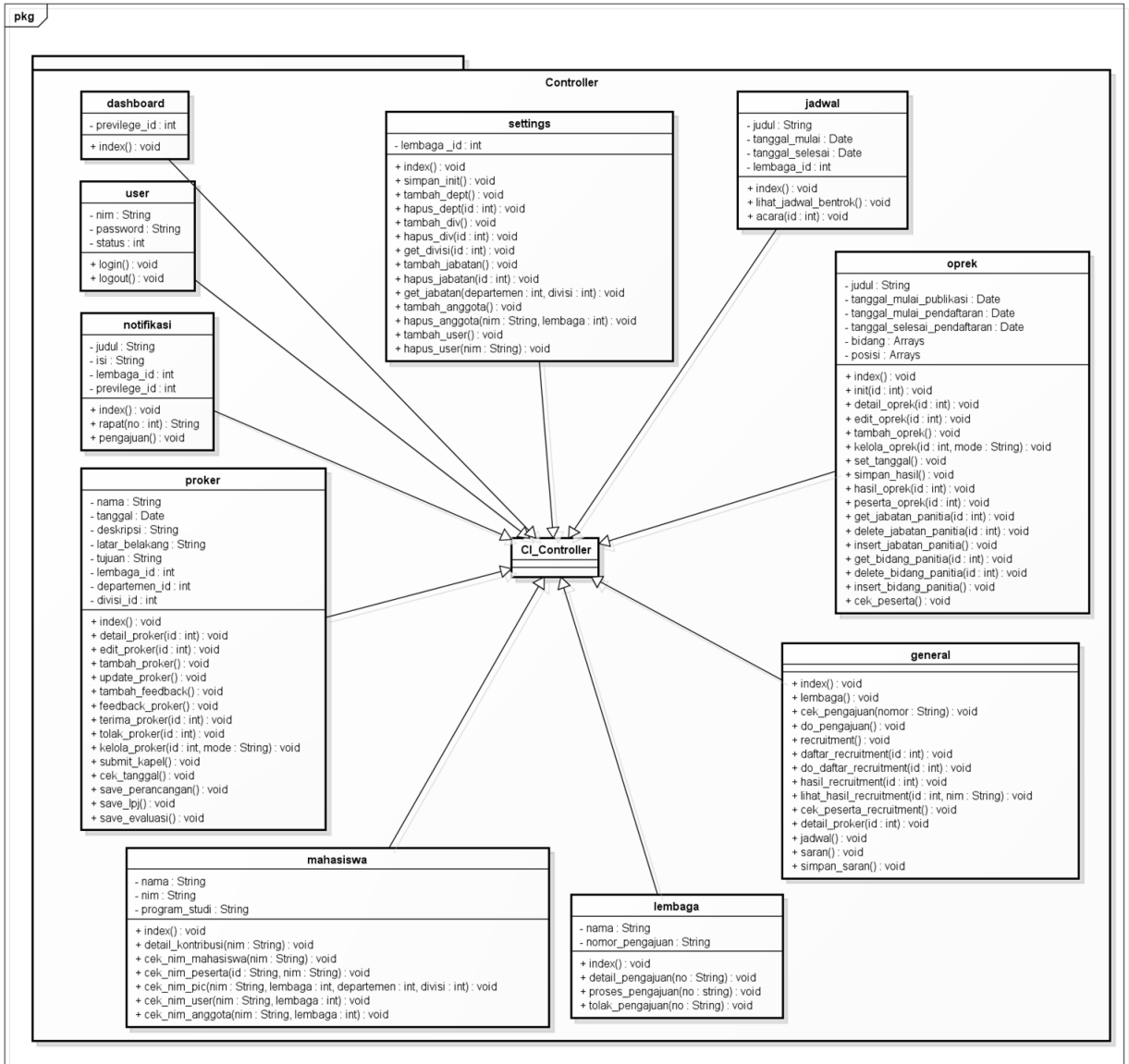
II. Skenario Use Case Memeriksa jadwal

<i>Use Case Name</i>	Use Case memeriksa jadwal
<i>Short Description</i>	Aplikasi memeriksa jadwal yang akan digunakan.
<i>Aktor</i>	Ketua divisi
<i>Pre Condition</i>	Masuk ke menu kelola program kerja dan program kerja pada tahap perancangan
<i>Post Condition</i>	Menampilkan hasil pemeriksaan jadwal, apakah kosong atau tidak.
<i>Normal Flow</i>	<ol style="list-style-type: none">1. Memilih tanggal mulai dan tanggal selesai2. Tekan tombol cek jadwal
<i>Alternate Flow</i>	-



LAMPIRAN C

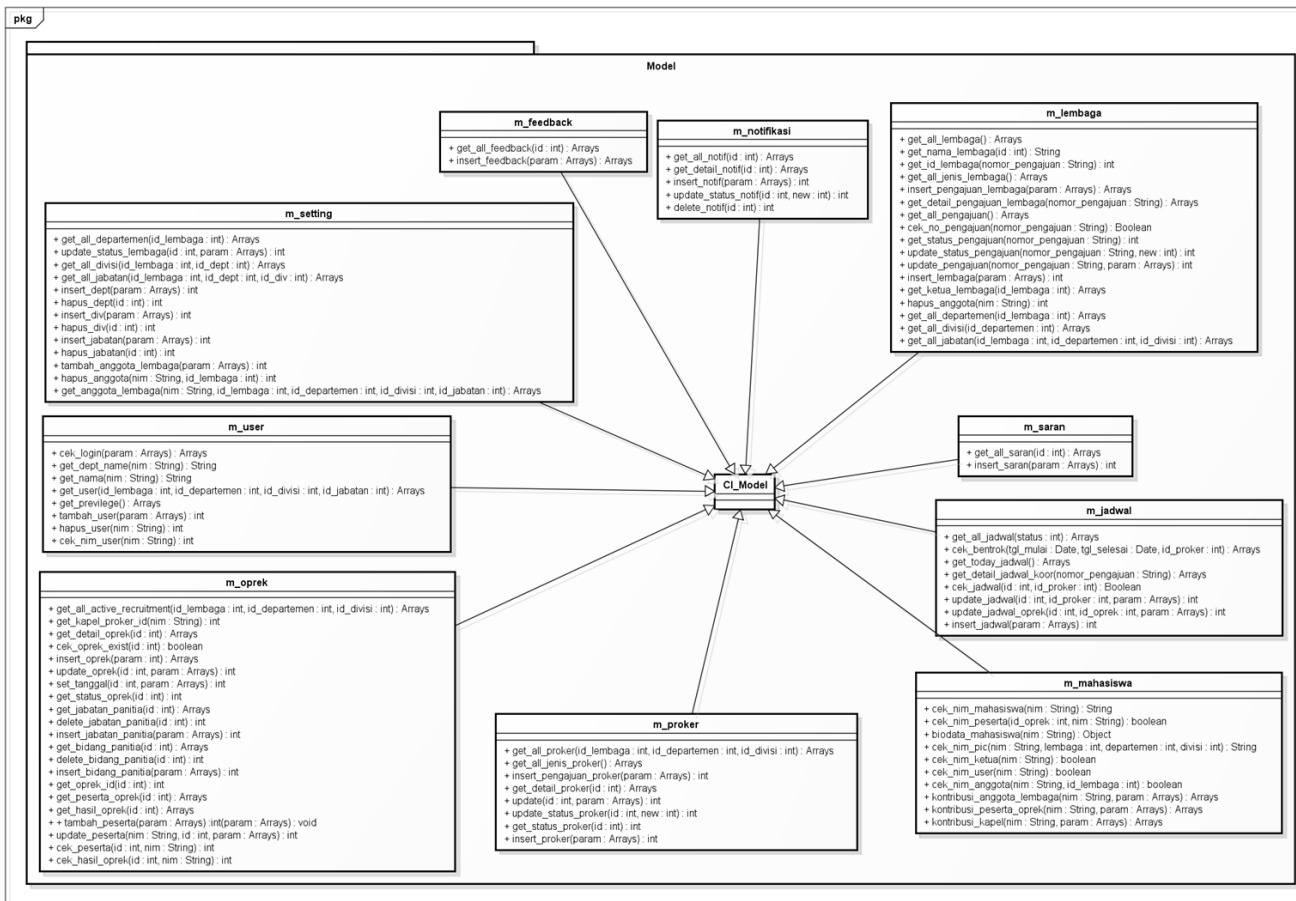
1. Class Diagram Controller



powered by Astah



2. Class Diagram Model

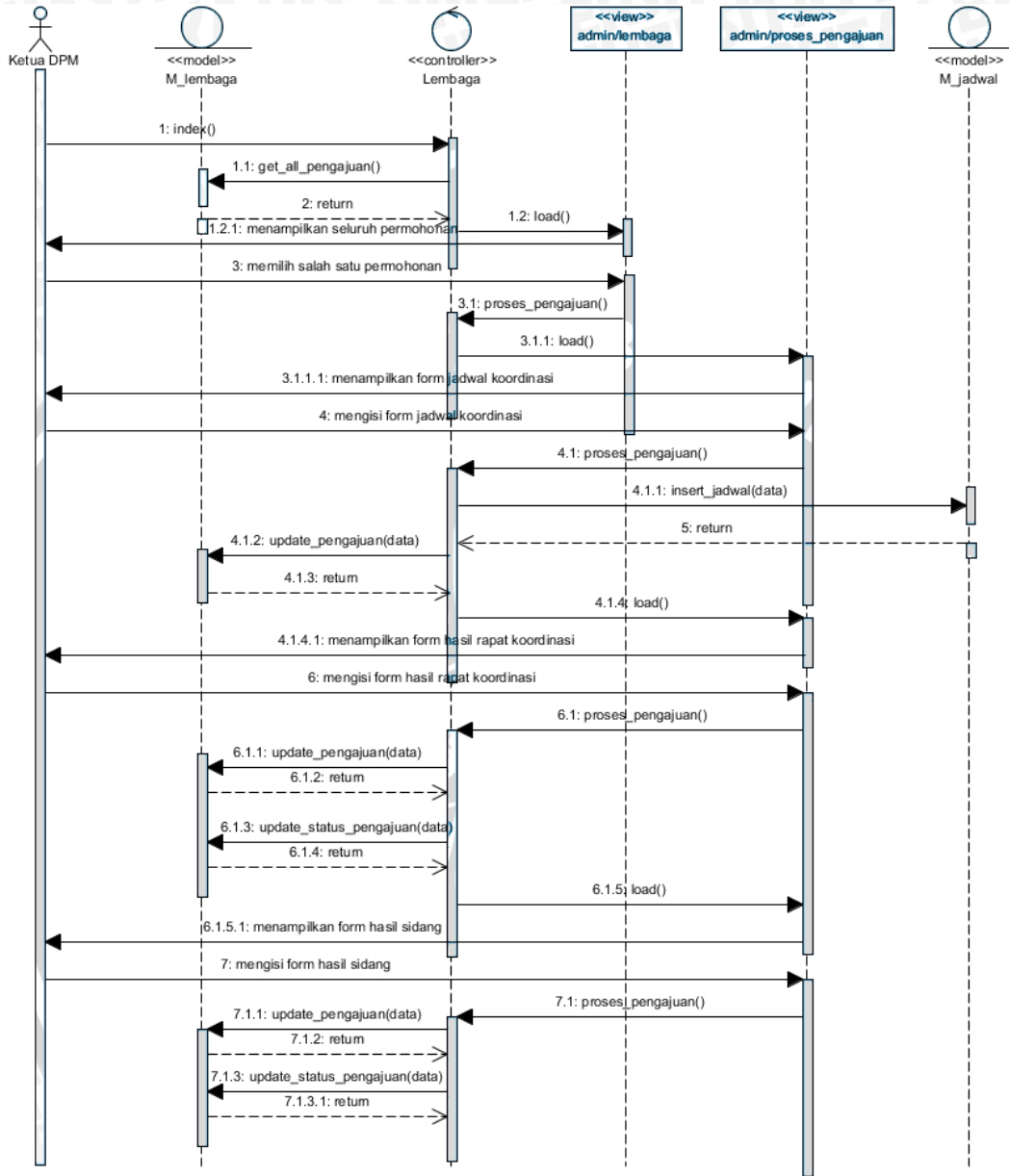


powered by Astah

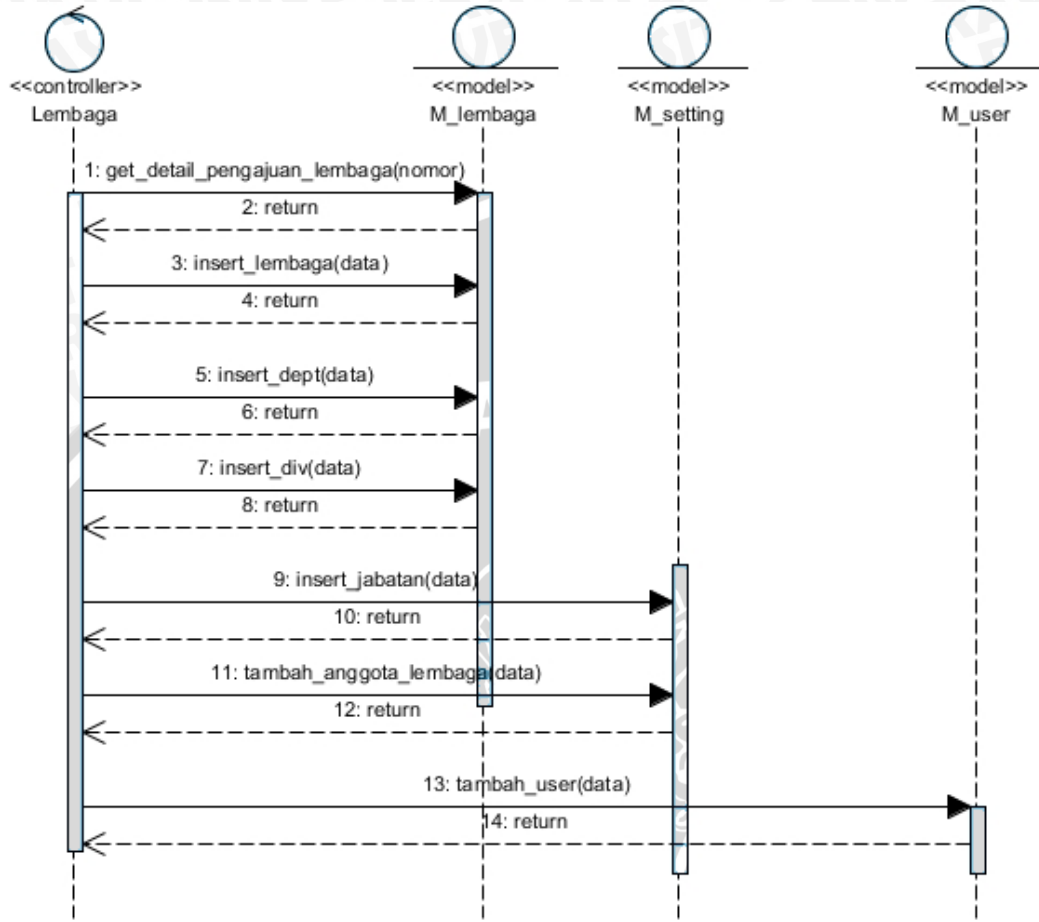


LAMPIRAN D

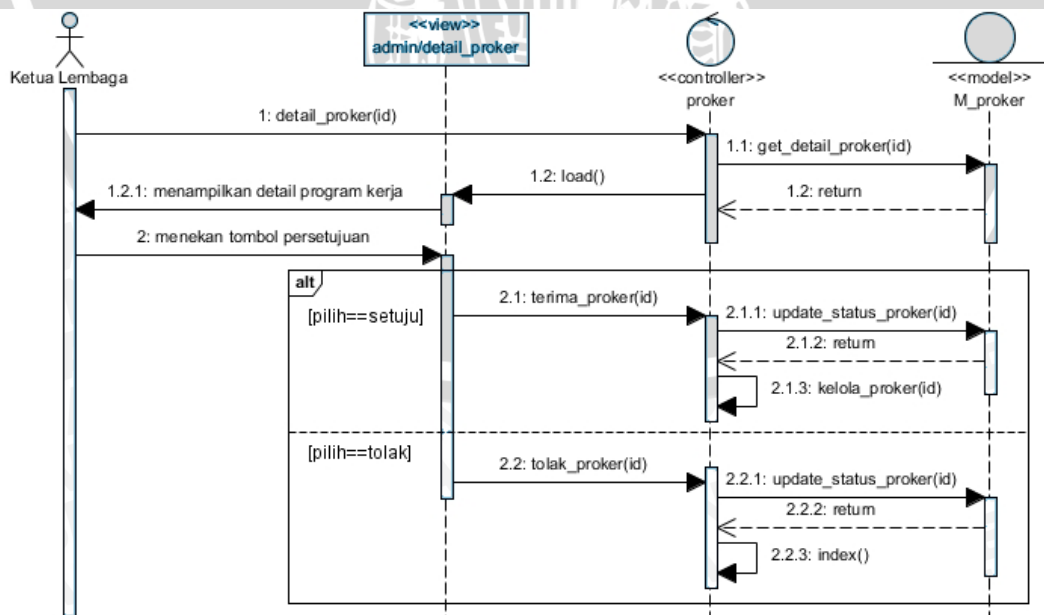
1. Sequence Diagram Persetujuan Pembentukan Lembaga Baru



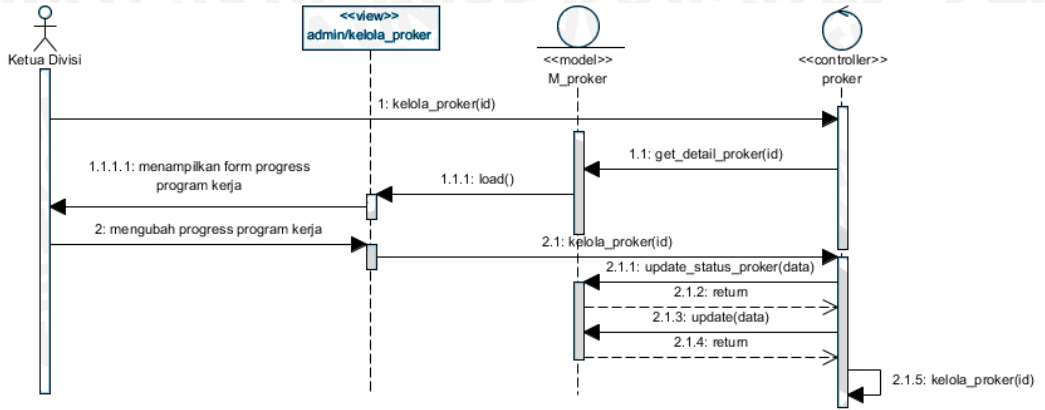
2. Sequence Diagram Membentuk Lembaga Baru



3. Sequence Diagram Persetujuan Program Kerja

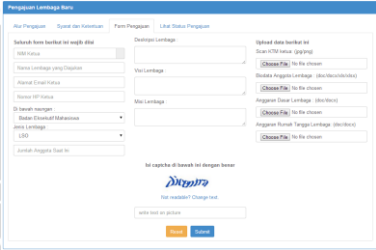
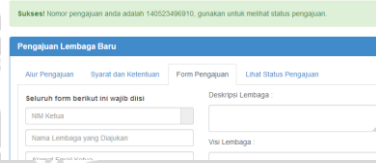
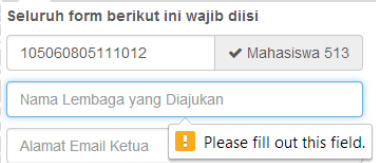



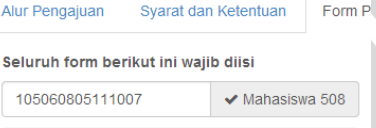
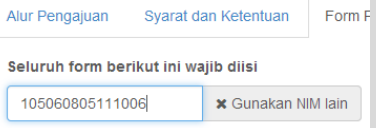
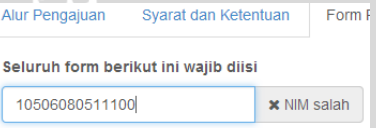

4. Sequence Diagram Mengubah Progress Program Kerja

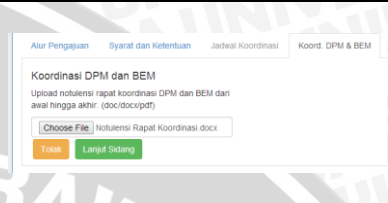
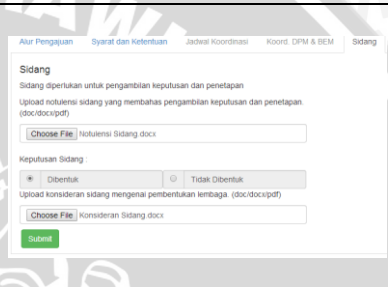
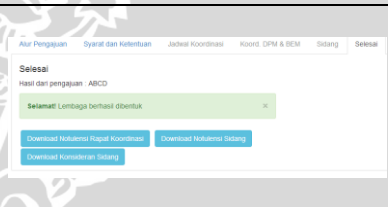
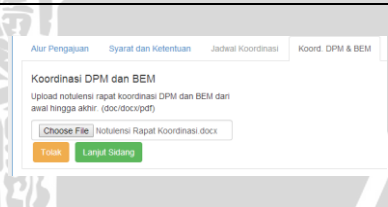


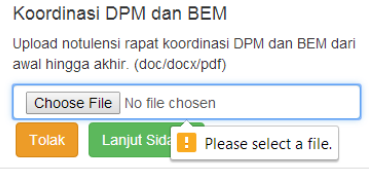
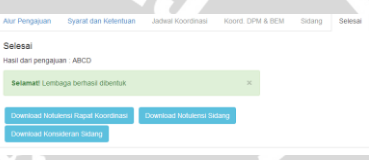
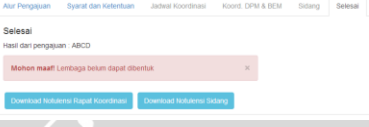
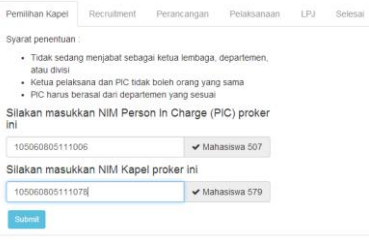
LAMPIRAN E

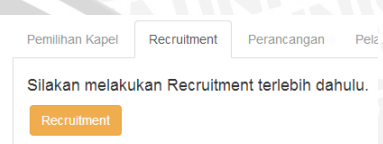
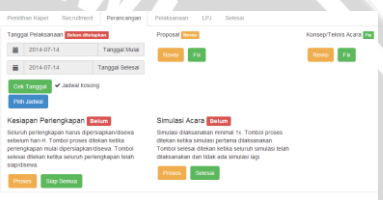


Hasil Functional Test

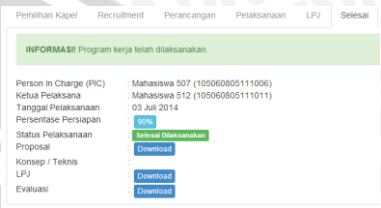
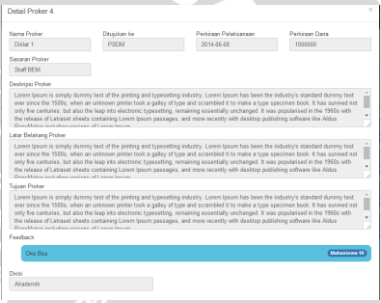
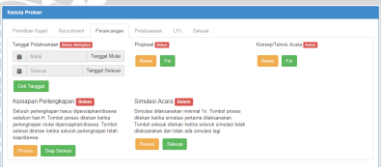
No SRS	Kebutuhan	Skenario	Tampilan	Hasil
SRS_001_01	Mahasiswa dapat mengajukan pembentukan lembaga mahasiswa baru.	tampilan awal		Valid
		Seluruh form diisi dengan benar		Valid
		form tidak diisi dengan lengkap		Valid




		Format file tidak sesuai		Valid
		NIM diisi dengan benar		Valid
		NIM telah terdaftar sebagai pengguna pada sistem		Valid
		NIM tidak valid		Valid
SRS_001_02	Pengguna dapat mengelola pembentukan lembaga mahasiswa baru.	Proses mencapai penentuan jadwal koordinasi		Valid

		Proses mencapai hasil rapat koordinasi		Valid
		Proses mencapai hasil sidang		Valid
		Proses pembentukan selesai		Valid
SRS_001_03	Pengguna dapat menyetujui atau menolak pengajuan pembentukan lembaga dari mahasiswa.	Tampilan awal		Valid

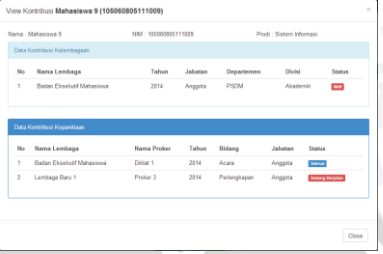

		File tidak diisi		Valid
		Menyetujui pengajuan pembentukan lembaga		Valid
		Menolak pengajuan pembentukan lembaga		Valid
SRS_002_01	Pengguna dapat mengelola program kerja.	Proses mencapai penentuan PIC dan ketua pelaksana		Valid

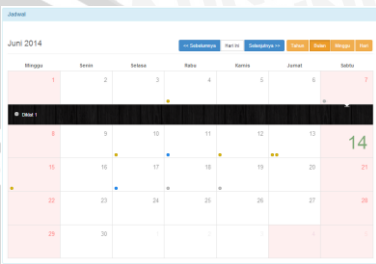
		<p>Proses mencapai <i>open recruitment</i> anggota panitia</p>		<p>Valid</p>
		<p>Proses mencapai perancangan pelaksanaan</p>		<p>Valid</p>
		<p>Proses mencapai pelaksanaan program kerja</p>		<p>Valid</p>
		<p>Proses mencapai LPJ pelaksanaan</p>		<p>Valid</p>

		<p>Program kerja selesai dilaksanakan</p>		<p>Valid</p>
<p>SRS_002_02</p>	<p>Pengguna dapat melihat rincian program kerja.</p>	<p>Tampilan rinci program kerja</p>		<p>Valid</p>
<p>SRS_002_03</p>	<p>Pengguna dapat mengubah progress dari setiap program kerja.</p>	<p>Tampilan awal</p>		<p>Valid</p>

		<p>Proposal masih revisi, konsep acara sudah selesai, perlengkapan belum siap, dan belum melakukan simulasi</p>		<p>Valid</p>
<p>SRS_002_04</p>	<p>Pengguna mendapatkan peringatan apabila jadwal program kerja yang diajukan bentrok dengan jadwal lain.</p>	<p>Jadwal bentrok</p>		<p>Valid</p>
		<p>Tanggal mulai kurang dari hari ini</p>		<p>Valid</p>

		Tanggal selesai kurang dari tanggal mulai		Valid
SRS_003_01	Mahasiswa dapat melakukan pendaftaran <i>open recruitment</i> .	Form pendaftaran <i>open recruitment</i>		Valid
SRS_003_02	Pengguna dapat melakukan seleksi terhadap peserta <i>open recruitment</i> .	Melakukan seleksi peserta <i>open recruitment</i>		Valid

<p>SRS_003_03</p>	<p>Pengguna dapat melihat keaktifan peserta <i>recruitment</i> saat ini.</p>	<p>Melihat keaktifan peserta <i>open recruitment</i></p>		<p>Valid</p>
<p>SRS_003_04</p>	<p>Pengguna dapat melihat hasil <i>open recruitment</i></p>	<p>Tampilan awal</p> <p>Peserta diterima</p> <p>Peserta tidak diterima</p> <p>Peserta tidak terdaftar</p>		<p>Valid</p> <p>Valid</p> <p>Valid</p> <p>Valid</p>

SRS_004_01	Mahasiswa dapat melihat jadwal kegiatan pada LKM PTIIK			Valid
------------	--	--	---	-------

