

SISTEM INFORMASI PENGELOLAAN BIMBINGAN TUGAS AKHIR MAHASISWA BERBASIS WEB DI STMIK JENDERAL ACHMAD YANI YOGYAKARTA MENGUNAKAN CODEIGNITER DAN JQUERY

Saifullah Al Amin¹, Choerun Asnawi²

Program Studi Manajemen Informatika
STMIK Jenderal Achmad Yani Yogyakarta

¹saipel45@gmail.com, ²c.asnawi@gmail.com

Abstrak

Sekolah Tinggi Manajemen Informatika dan Komputer Jenderal Achmad Yani (STMIK Jend. A. Yani) adalah salah satu perguruan tinggi swasta di Yogyakarta. Sebagaimana perguruan tinggi lain, mahasiswa tingkat akhir tiap tahunnya mengerjakan proyek yang memiliki daya guna, proyek ini di kalangan mahasiswa disebut sebagai tugas akhir. Dalam pengelolaannya, lembaga memiliki sistem informasi akademik, akan tetapi belum sepenuhnya dipergunakan karena sistem yang dimiliki di desain untuk perguruan tinggi selevel universitas. Saat ini lembaga belum mempunyai sistem yang menangani hal tersebut sesuai dengan kondisi riil, apalagi pada tahun ini, tahun 2013, lembaga meluluskan bukan hanya diploma tiga namun juga meluluskan strata satu. Dengan adanya kebutuhan yang sedemikian penting dan demi kemudahan dalam pengelolaan, maka sudah saatnya dibuat sistem yang menangani hal tersebut.

Sistem Informasi Pengelolaan Bimbingan Tugas Akhir Mahasiswa atau disebut SILABITAMA merupakan hasil dari usaha untuk memenuhi kebutuhan yang ada. Dengan didahului pemahaman masalah dan dilanjutkan dengan menganalisa permasalahan dan solusi yang memungkinkan, maka penulis merancang dan mengimplementasikan sistem yang ditujukan untuk menangani pengelolaan bimbingan tugas akhir setiap tahunnya. Dengan adanya SILABITAMA berbasis web, proses pengelolaan dan pemantauan tugas akhir bisa dilakukan dengan cepat, akurat dan mudah.

Dengan demikian, masalah yang ada dapat diatasi sesuai kondisi riil kampus STMIK Jend. A. Yani begitu juga dengan kompleksitas masalah yang selalu berkembang seiring bertambahnya jumlah mahasiswa dan program studi.

Kata Kunci: Pengelolaan Tugas Akhir, Kompleksitas, Pemantauan, Sistem Informasi, SILABITAMA.

1. Pendahuluan

1.1 Latar Belakang Masalah

Bimbingan tugas akhir merupakan suatu forum konsultasi mahasiswa kepada dosen pembimbing tugas akhirnya, selama mahasiswa mengerjakan tugas akhir. Proses bimbingan dapat dilakukan dengan cara tatap muka secara langsung, atau dengan memanfaatkan teknologi yang memungkinkan (*e-mail, chat, dll*). Kegiatan pembimbingan tugas akhir utamanya bertujuan untuk mengontrol pengerjaan tugas akhir mahasiswa agar tidak melenceng dari tujuan awal atau melakukan kesalahan yang akan memperlambat proses penyelesaian

tugas akhirnya. Selain itu diharapkan dengan adanya bimbingan ini maka dapat menghasilkan karya yang berkualitas tinggi.

Kampus Sekolah Tinggi Manajemen Informatika dan Komputer Jenderal Achmad Yani Yogyakarta (STMIK Jend. A. Yani) saat ini sudah memiliki Sistem Informasi Akademik. Masalahnya adalah bahwa sistem yang dimaksud ditujukan untuk perguruan tinggi setingkat universitas sehingga ada bagian-bagian tertentu yang jika diterapkan di kampus STMIK Jend. A. Yani menjadi tidak cocok. Salah satunya adalah dalam penanganan tugas akhir mahasiswa.

Sementara itu dengan bertambahnya program studi, masalah tentang penanganan tugas akhir semakin kompleks. Maka penulis mempunyai ide untuk membuat sistem baru yang kemudian digunakan untuk pengelolaan tugas akhir di STMIK Jend. A. Yani. Sistem tersebut akan dirancang agar sesuai dengan kondisi riil yang ada di STMIK Jend. A. Yani sehingga menjadi lebih cocok untuk digunakan daripada sistem yang lama.

1.2 Tujuan dan Manfaat

Tujuan penelitian ini adalah untuk merancang dan mengimplementasikan Sistem Informasi Pengelolaan Bimbingan Tugas Akhir Mahasiswa (SILABITAMA) agar sesuai dengan kondisi riil yang ada di STMIK Jend. A. Yani sehingga mempermudah penanganan tugas akhir mahasiswa yang kini semakin kompleks seiring bertambahnya program studi.

Adapun manfaat yang diharapkan penulis adalah sebagai berikut:

1. Meningkatkan layanan informasi kepada pengguna sistem.
2. Mempermudah dalam mengetahui progres bimbingan tugas akhir sehingga dapat membantu pengambilan keputusan demi terselesaikannya tugas akhir tepat waktu.
3. Meningkatkan kuantitas mahasiswa lulus tepat waktu sehingga menciptakan citra baik kampus.
4. Mengurangi resiko pengerjaan tugas akhir oleh mahasiswa yang biasanya menumpuk di akhir jadwal yang ditentukan, hal ini sangat membebani petugas yang menangani bimbingan tugas akhir dilihat dari pengalaman tahun-tahun sebelumnya.

1.3 Batasan Masalah

Membangun sebuah sistem informasi pengelolaan tugas akhir yang lengkap adalah pekerjaan yang kompleks dan memakan waktu lama atau tenaga yang tidak sedikit. Untuk itu penulis membatasi permasalahan hanya untuk:

1. Proses pendaftaran tugas akhir
2. Proses pembimbingan
 - a. Pemilihan dosen pembimbing tugas akhir
 - b. Progres bimbingan
3. Pencatatan dan pengolahan data terkait ujian pendadaran, seperti: berita acara dan nilai mahasiswa.

Sistem yang dihasilkan melalui penelitian ini tidak menangani hal-hal berikut:

1. Pembagian distribusi pembimbing secara otomatis.
2. Penjadwalan seminar dan pendadaran secara otomatis.
3. Penilaian ujian pendadaran secara otomatis.

2. Dasar Teori

2.1 Konsep Dasar Sistem Informasi

McLeod (2004) menyatakan bahwa sistem informasi merupakan sistem yang mempunyai kemampuan untuk mengumpulkan informasi dari semua sumber dan menggunakan berbagai media untuk menampilkan informasi.

Sistem informasi dapat didefinisikan sebagai suatu sistem di dalam suatu organisasi yang merupakan kombinasi dari orang-orang, fasilitas, teknologi, media prosedur-prosedur dan pengendalian yang ditujukan untuk mendapatkan jalur komunikasi penting, memproses tipe transaksi rutin tertentu, memberi sinyal kepada manajemen dan yang lainnya terhadap kejadian-kejadian internal dan eksternal yang penting dan menyediakan suatu dasar informasi untuk pengambilan keputusan (Jogiyanto, 2005).

Sistem informasi mempunyai enam buah komponen atau disebut juga dengan blok bangunan (*building block*), yaitu (Jogiyanto, 2005):

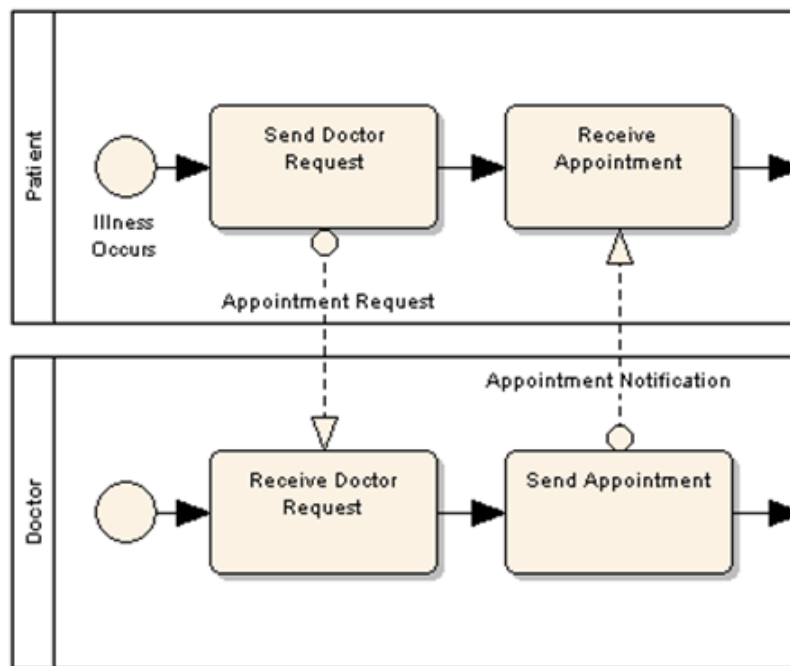
1. Komponen *input* atau komponen masukan
2. Komponen model
3. Komponen *output* atau komponen keluaran
4. Komponen teknologi
5. Komponen basis data
6. Komponen *control* atau komponen pengendalian

Keenam komponen ini harus ada bersama-sama dan membentuk satu kesatuan. Jika satu atau lebih komponen tersebut tidak ada, maka sistem informasi tidak akan dapat melakukan fungsinya, yaitu pengolahan data dan tidak dapat mencapai tujuan, yaitu menghasilkan informasi yang relevan, tepat waktu dan akurat (Jogiyanto, 2005).

2.2 Business Process Modeling Notation (BPMN)

White (2004) menuliskan bahwa BPMN digunakan untuk mendefinisikan suatu diagram proses bisnis atau *Business Process Diagram* (BPD), yang didasarkan pada teknik pembuatan *flow chart* yang disesuaikan untuk model grafik dari suatu operasi proses bisnis. BPMN merupakan notasi yang telah dipersiapkan untuk dapat dipahami oleh semua pengguna bisnis, mulai dari analis bisnis yang membuat *draft* awal dari proses, pengembang teknis yang bertanggung jawab untuk mengimplementasikan teknologi yang akan melakukan proses, hingga pelaku bisnis yang akan mengelola dan mengawasi proses.

Business process dapat digambarkan dalam BPMN secara mudah. Sebagai contoh dapat dilihat pada gambar *business process* sederhana yang tergambar pada Gambar 1.

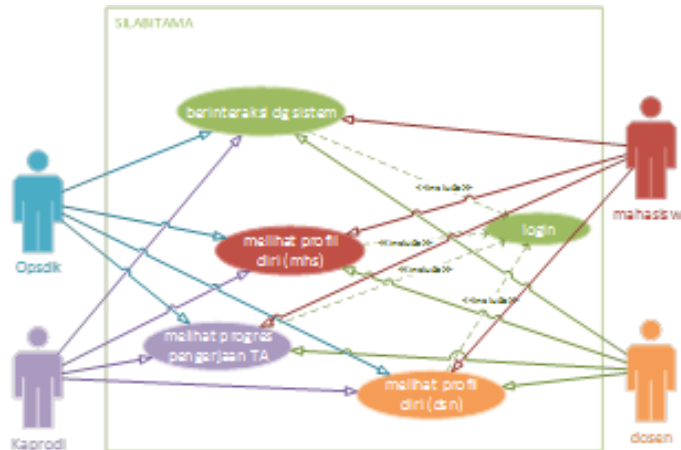


Gambar 1 Contoh BPMN

2.3 Unified Modeling Language (UML)

Berdasarkan Booch, et al. (1999), UML merupakan sebuah bahasa grafis untuk memvisualisasikan, membuat spesifikasi, mengkonstruksi, dan mendokumentasikan artefak dari suatu sistem perangkat lunak. UML menawarkan suatu cara baku untuk menuliskan cetak biru (*blueprint*) dari suatu sistem, dari hal-hal yang bersifat konseptual semacam proses bisnis dan fungsi sistem, hingga hal-hal konkrit seperti *statement* dalam bahasa pemrograman, skema basis data, dan komponen *software* yang *reusable*.

Hal yang penting untuk dipahami adalah bahwa UML bukanlah suatu metode atau prosedur. UML adalah “bahasa” yang digunakan untuk mendefinisikan suatu sistem perangkat lunak, untuk merinci artefak dari sistem, untuk mendokumentasikan dan mengkonstruksi. UML boleh digunakan dalam berbagai cara untuk mendukung metodologi pengembangan perangkat lunak, namun sendirinya tidak menentukan metodologi atau proses itu. Contoh UML *use case* dapat dilihat pada Gambar 2.



Gambar 2 Contoh UML Use Case

2.4 Entity Relationship Diagram (ERD)

ERD merupakan teknik yang digunakan dalam merancang basis data menggunakan bentuk diagram (Chen, 1976). Model *Entity-Relationship* yang berisi komponen-komponen himpunan entitas dan himpunan relasi yang masing-masing dilengkapi dengan atribut-atribut yang merepresentasikan seluruh fakta dari dunia nyata, dapat digambarkan dengan lebih sistematis menggunakan ERD. Notasi-notasi simbolik dalam ERD yang dapat dipergunakan adalah (Fathansyah, 2012):

1. Persegi panjang, menyatakan himpunan entitas.
2. Lingkaran/elips, menyatakan atribut, dengan atribut yang berfungsi sebagai *key* ditandai dengan garis bawah.
3. Belah ketupat, menyatakan himpunan relasi.
4. Garis, sebagai penghubung antara himpunan relasi dengan himpunan entitas serta himpunan entitas dan atributnya.

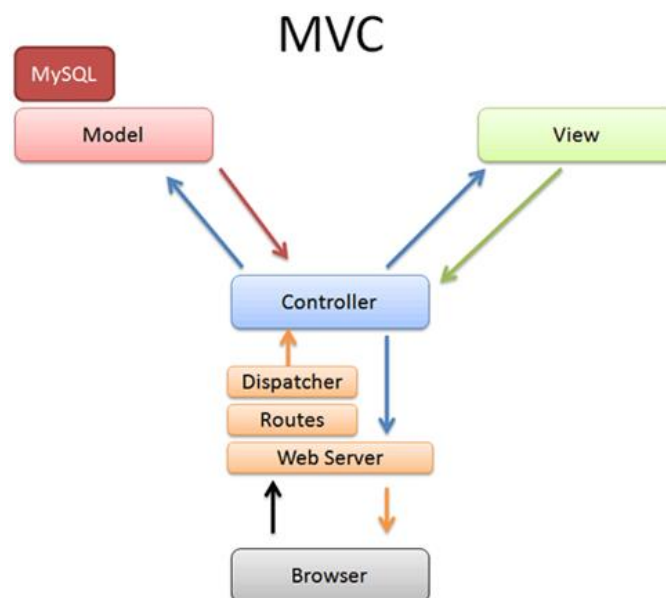
Kardinalitas relasi dapat dinyatakan dengan banyaknya garis cabang atau dengan pemakaian angka. Jika menggunakan angka, maka angka 1 dan 1 untuk menandai relasi satu-ke-satu, angka 1 dan N untuk relasi satu-ke-banyak, serta N dan N untuk relasi banyak-ke-banyak (Fathansyah, 2012).

2.5 Framework

Framework adalah sebuah struktur konseptual dasar yang digunakan untuk memecahkan sebuah permasalahan, bahkan isu-isu kompleks yang ada. Sebuah *framework* telah berisi sekumpulan arsitektur/konsep-konsep yang dapat mempermudah dalam pemecahan sebuah permasalahan. *Framework* bukanlah peralatan (*tools*) untuk memecahkan sebuah masalah, tetapi sebagai alat bantu. *Framework* hanya menjadi sebuah konstruksi dasar yang menopang sebuah konsep atau sistem yang bersifat *essential support* (penting tapi bukan komponen utama) (Daqiqil, 2011).

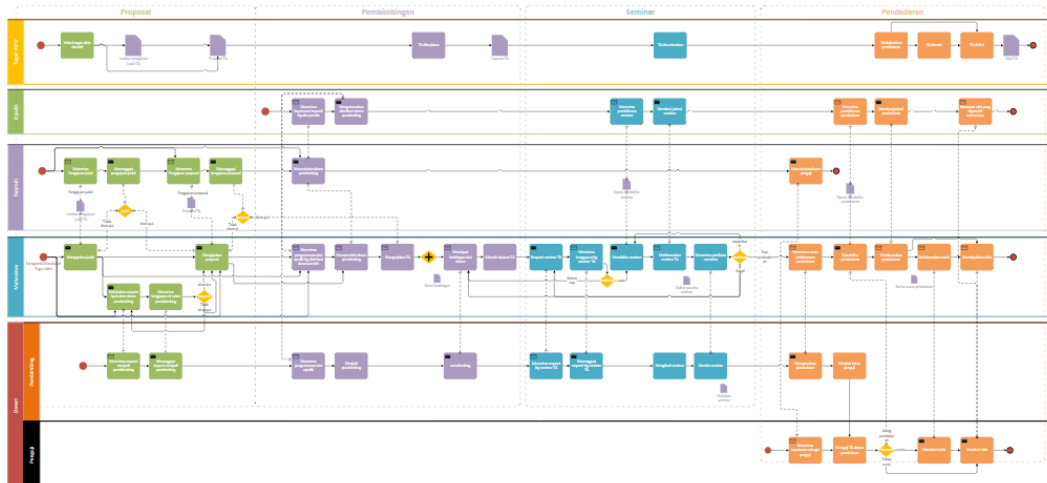
2.6 CodeIgniter

CodeIgniter merupakan *framework* pengembangan aplikasi dan sebuah *toolkit* untuk membangun aplikasi *web* menggunakan bahasa pemrograman PHP. CodeIgniter menyediakan pustaka yang sangat lengkap untuk berbagai *task* yang umum, dengan sekaligus menyertakan antarmuka sederhana dan struktur logis untuk mengakses pustaka yang tersedia itu. Dengan demikian pengembangan aplikasi akan menjadi lebih cepat karena *programmer* tidak perlu menulis kode dari awal lagi (Ellislab, 2013).



Gambar 3 Model-View-Controller (MVC) dalam CodeIgniter

CodeIgniter dibangun menggunakan konsep arsitektur *Model-View-Controller*. CodeIgniter sendiri merupakan salah satu *framework* berbasis PHP yang tercepat. Pada acara frOSCon (Agustus 2008), pembuat PHP Rasmus Lerdorf mengatakan dia menyukai CodeIgniter karena dia lebih ringan dan cepat dibandingkan *framework* lainnya (Ellislab, 2013).



Gambar 5 *Business Process* Pembimbingan Tugas Akhir

Pada Gambar 4 maupun Gambar 5 proses dibagi dalam empat tahap, warna hijau menunjukkan tahap proposal, ungu menunjukkan tahap pembimbingan, biru menunjukkan tahap seminar dan jingga menunjukkan tahap pendadaran. Dari keempat tahap ini dapat dilihat terdapat beberapa dokumen untuk mendokumentasikan peristiwa-peristiwa yang ada, adapun dokumen-dokumen tersebut adalah sebagai berikut:

1. Lembar pengajuan judul tugas akhir
2. Proposal tugas akhir
3. Laporan tugas akhir
4. Kartu bimbingan
5. Daftar peserta seminar
6. Penilaian seminar
7. Berita acara ujian pendadaran
8. Lembar penilaian tugas akhir
9. Nilai ujian tugas akhir
10. Surat keputusan hasil ujian pendadaran

Kartu bimbingan digunakan untuk mencatat pembimbingan dalam mengerjakan tugas akhir. Di dalamnya terdapat informasi siapa pemilik kartu, berapa kali melaksanakan bimbingan dan bagaimana progres pengerjaan tugas akhir mahasiswa yang bersangkutan. Di saat mendaftar seminar terdapat syarat bahwa mahasiswa harus melaksanakan bimbingan sebanyak lima kali. Kartu bimbingan inilah yang digunakan sebagai bukti tentang hal tersebut.

Presensi seminar digunakan untuk mencatat peserta yang hadir ketika tugas akhir diseminarkan, pada lembar ini diberikan penilaian seminar oleh dosen pembimbing.

Lembar berita acara digunakan sebagai bukti bahwa mahasiswa melaksanakan pendadaran, lembar ini dikeluarkan oleh opsdik bersamaan dengan lembar nilai ujian tugas akhir, dan surat keputusan hasil ujian pendadaran untuk diisi oleh dosen penguji ketika pendadaran berlangsung.

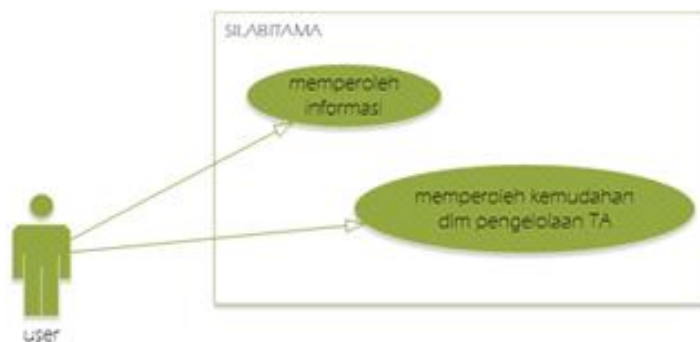
Lembar penilaian tugas akhir dalam pendadaran berjumlah rangkap tiga yang kemudian diisi oleh tiga dosen penguji. Sedangkan untuk lembar nilai ujian tugas akhir diisi oleh ketua tim penguji yaitu dosen pembimbing tugas akhir mahasiswa.

3.2 Perancangan

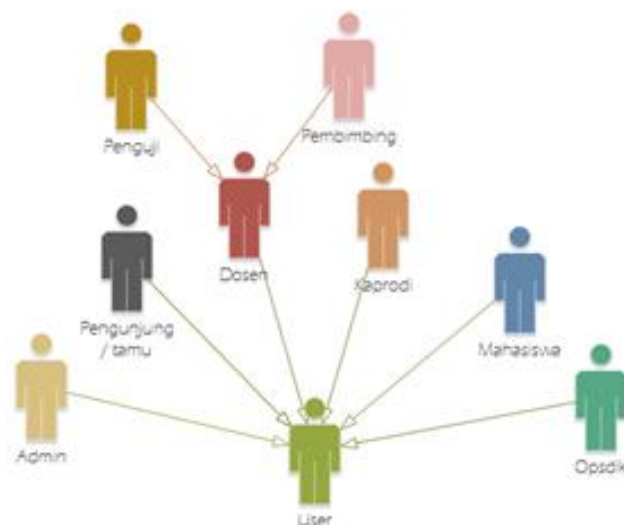
3.2.1 Rancangan Proses

Rancangan proses merupakan gambaran secara keseluruhan dari sistem yang dikembangkan sehingga dapat menggambarkan setiap proses yang dilakukan. Rancangan proses SILABITAMA digambarkan dengan *use case diagram* dan *activity diagram*.

Berikut penulis paparkan SILABITAMA dilihat dari sudut pandang user.



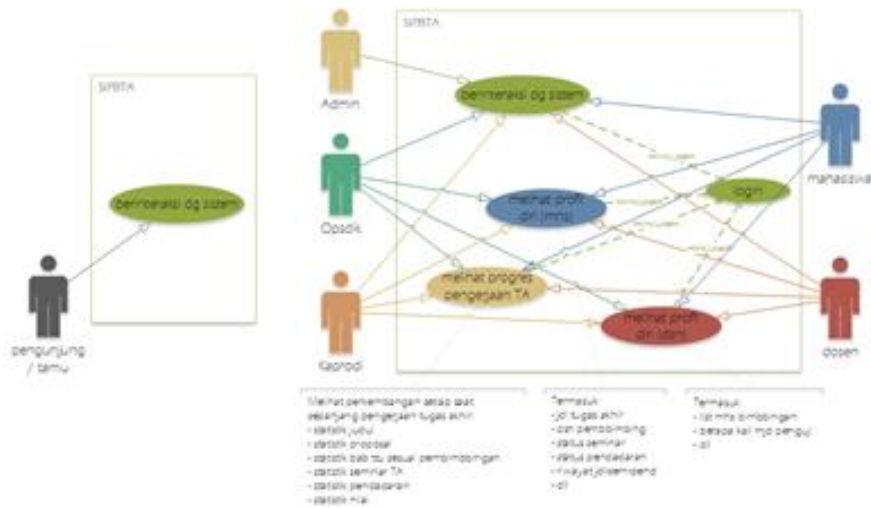
Gambar 6 Top Use Case



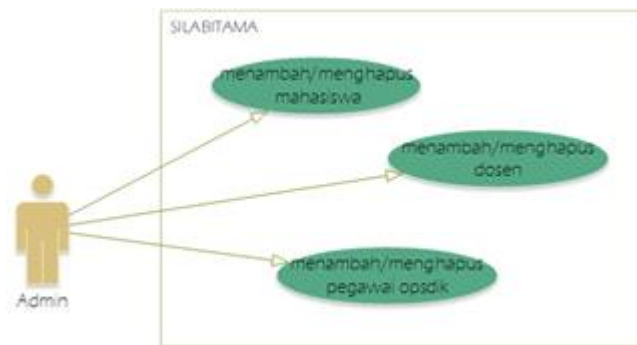
Gambar 7 Actors



Gambar 8 System - Subsystem

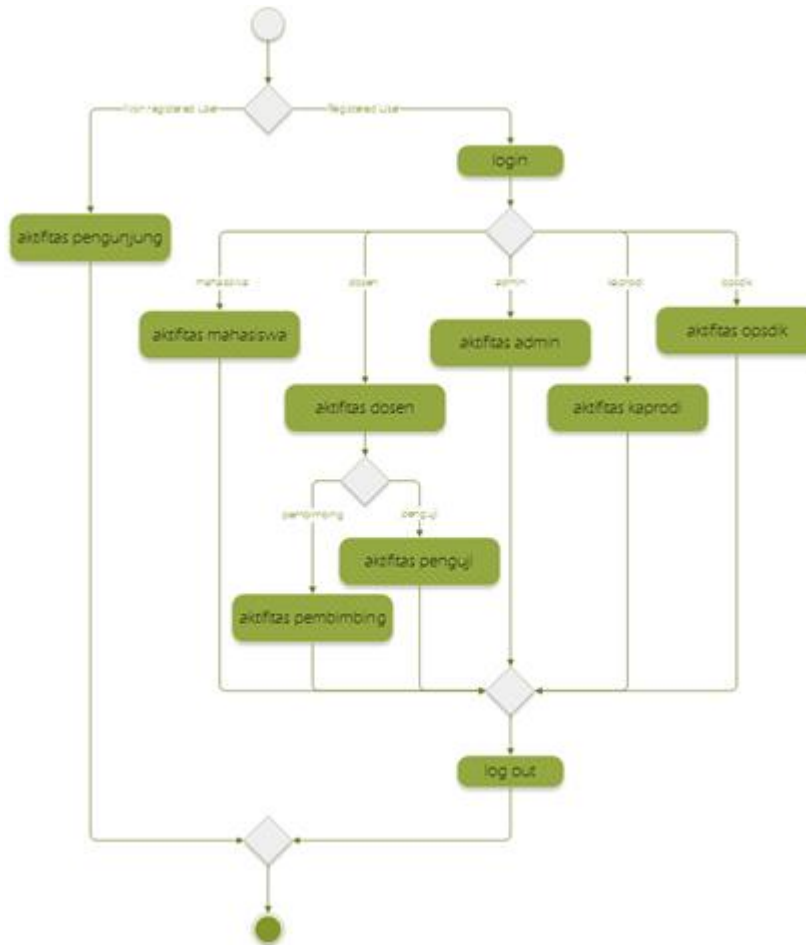


Gambar 9 Use Case



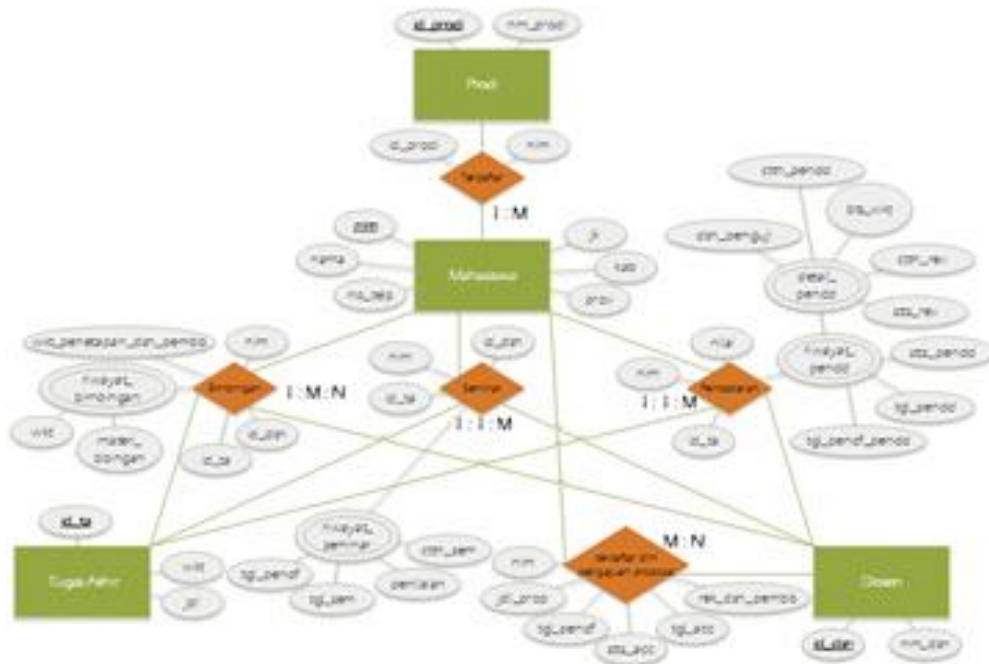
Gambar 10 Use Case untuk Admin

Fungsi *admin* di sini masih sangat sederhana yaitu hanya menambah dan menghapus pengguna sistem, tidak menutup kemungkinan untuk menambah dan mengurangi tugas *admin* sesuai kebutuhan lembaga di kemudian hari.



Gambar 11 Activity Diagram

3.2.2 Rancangan Basis Data



Gambar 12 Entity Relationship Diagram

Gambar 14 Form Penyetujuan Judul Tugas Akhir

Form pembagian pembimbing dipergunakan oleh Kaprodi untuk menentukan distribusi pembimbing bagi mahasiswa yang belum mengajukan proposal maupun yang belum merekomendasikan pembimbing tugas akhir.

Rancangan awal *form* terlihat pada Gambar 15. Di situ terdapat pencarian mahasiswa berdasarkan NIM dan dosen pembimbing. Jika *form input* dikosongkan maka semua mahasiswa yang belum memiliki pembimbing akan tampil.

Gambar 15 Form Pembagian Dosen Pembimbing

Form revisi dan catatan pendaran ditunjukkan pada Gambar 16. *Form* ini digunakan bagi dosen penguji untuk memberikan revisi dan catatan pengingat bagi penguji sendiri terkait dengan revisi yang diberikan kepada mahasiswa.

Adapun poin-poin yang dimasukkan dalam *form* ini ada tiga, yaitu revisi, catatan revisi dan batas waktu revisi. Catatan revisi dan batas waktu tidak dapat dikosongkan, akan tetapi catatan pendaran dapat dikosongkan.

The image shows a web form with the following elements:

- Title: **Revisi dan catatan pendadaran**
- Fields: NIM, Nama, judul TA, catatan pendadaran (text area), catatan revisi (text area), batas waktu revisi (text input with date format (dd/mm/yyyy)).
- Button: simpan

Gambar 16 Form Revisi dan Catatan Pendadaran

3.2.4 Rancangan *Output*

SILABITAMA memiliki banyak *user* dengan tingkatan yang berbeda-beda. Berikut adalah rancangan *output* sistem masing-masing *user*.

3.2.4.1 Dosen Pembimbing Tugas Akhir

1. Menampilkan daftar mahasiswa yang dibimbing.
2. Menampilkan judul tugas akhir mahasiswa yang disetujui oleh kaprodi.
3. Menampilkan progres mahasiswa yang dibimbing:
 - a. Menampilkan banyaknya bimbingan yang telah dilaksanakan oleh masing-masing mahasiswa.
 - b. Menampilkan progres mahasiswa sampai bab tertentu sesuai bimbingan Tugas Akhir yang sudah dilaksanakan.
 - c. Menampilkan daftar mahasiswa yang sudah atau belum seminar.
 - d. Menampilkan daftar mahasiswa yang sudah atau belum ujian pendadaran.
 - e. Menampilkan daftar mahasiswa yang sudah atau belum mendapatkan nilai.

3.2.4.2 Dosen penguji dalam pendadaran Tugas Akhir

1. Menampilkan jadwal pendadaran mahasiswa yang di uji.
2. Menampilkan progres revisi tugas akhir mahasiswa setelah pendadaran.
3. Menampilkan nilai mahasiswa yang diuji.

3.2.4.3 Mahasiswa

1. Menampilkan progres dalam pengerjaan tugas akhir meliputi:
 - a. Status disetujuinya judul tugas akhir.
 - b. Berapa kali melaksanakan bimbingan.
 - c. Status sudah seminar atau belum.
 - d. Status sudah ujian pendadaran atau belum.
 - e. Status sudah yudisium atau belum.
2. Menampilkan apa yang direvisi dalam pengerjaan tugas akhir (berkaitan dengan dosen pembimbing tugas akhir).
3. Menampilkan apa yang direvisi dalam pengerjaan tugas akhir (berkaitan dengan dosen penguji saat pendadaran).
4. Menampilkan jadwal seminar.
5. Menampilkan jadwal ujian pendadaran.
6. Menampilkan nilai.

3.2.4.4 Opsdik

1. Menampilkan daftar judul tugas akhir yang sudah disetujui oleh kaprodi.
2. Menampilkan daftar judul tugas akhir yang sudah diujikan dalam ujian pendadaran.
3. Menampilkan jadwal seminar.
4. Menampilkan jadwal ujian pendadaran.
5. Menampilkan nilai hasil ujian pendadaran yang diperoleh mahasiswa.

3.2.4.5 Kaprodi

1. Menampilkan daftar judul tugas akhir yang sudah disetujui oleh kaprodi.
2. Menampilkan daftar judul tugas akhir yang sudah diujikan dalam ujian pendadaran.
3. Progres mahasiswa pada program studinya secara keseluruhan:
 - a. Menampilkan progres mahasiswa sampai bab tertentu sesuai bimbingan tugas akhir yang sudah dilaksanakan.
 - b. Menampilkan daftar mahasiswa yang sudah atau belum seminar.
 - c. Menampilkan daftar mahasiswa yang sudah atau belum ujian pendadaran.
 - d. Menampilkan daftar mahasiswa yang sudah atau belum mendapatkan nilai.

3.2.5 Rancangan Kendali

3.2.5.1 Kendali *Login*

Adanya kendali ketika *login* dimaksudkan untuk pembatasan akses terhadap sistem. Pengguna SILABITAMA memiliki *username* dan *password* yang digunakan untuk *login* ke sistem. Setiap pengguna sistem memiliki otoritas yang berbeda-beda terkait dengan pengaksesan yang dilakukan.

Adapun *username* dan *password* yang digunakan untuk *login* ke sistem sama seperti yang digunakan untuk *login* ke portal akademik STMIK Jend. A. Yani karena pada dasarnya *user* tidak bisa masuk ke sistem sebelum memperoleh ijin portal akademik, hal ini dimaksudkan untuk memudahkan *user* tanpa harus menghafal banyak *username* dan *password* hanya untuk masuk ke sistem yang sebenarnya ada dalam satu naungan lembaga. Selain itu, dengan proses *login* seperti ini sistem memiliki proteksi yang kuat layaknya sistem informasi akademik yang saat ini dipergunakan.

3.2.5.2 Kendali *Input* dan *Pemrosesan Data*

Adanya kendali pada saat memasukkan data digunakan untuk menjamin keakuratan data, kelengkapan data dan validasi terhadap masukan. Validasi masukan data pada sistem dibuat menggunakan JavaScript. Apabila pengguna sistem belum memasukkan data secara lengkap maupun melakukan kesalahan dalam *input* data, maka akan muncul pesan kesalahan dan pemrosesan data tidak dapat dilakukan sebelum data yang dimasukkan lengkap dan sesuai yang dikehendaki sistem.

3.3 Implementasi

3.3.1 Koneksi *Database* dan Sistem

Script yang digunakan sistem untuk menyimpan konfigurasi koneksi ke *database* dapat dilihat pada Kode Program 1. Kode tersebut merupakan bagian dari konfigurasi yang sudah tersedia di dalam CodeIgniter. *Programmer* hanya tinggal mengubah nilai-nilai yang ada agar sesuai dengan konfigurasi *server* basis data yang digunakannya.

Kode Program 1 Kode yang berisi konfigurasi koneksi ke *database*

```
$active_group = 'default';
$active_record = TRUE;
$db['default']['hostname'] = 'localhost';
$db['default']['username'] = 'root';
$db['default']['password'] = '*****';
```



```

$db['default']['database'] = 'silabitama';
$db['default']['dbdriver'] = 'mysql';
$db['default']['dbprefix'] = '';
$db['default']['pconnect'] = TRUE;
$db['default']['db_debug'] = TRUE;
$db['default']['cache_on'] = FALSE;
$db['default']['cachedir'] = '';
$db['default']['char_set'] = 'utf8';
$db['default']['dbcollat'] = 'utf8_general_ci';
$db['default']['swap_pre'] = '';
$db['default']['autoinit'] = TRUE;
$db['default']['stricton'] = FALSE;

```

3.3.2 Implementasi *Input*

Gambar 17 Implementasi *Input* Edit Profil Dosen

Gambar 17 merupakan salah satu *form input* sistem, masih banyak *form input* lain yang seirama dengan *form* tersebut; bentuknya kurang lebih sama. Pada *form* tersebut, kode program yang digunakan untuk menyimpan *input* profil dosen dapat dilihat pada Kode Program 2. Dalam proses penyimpanan data yang dimasukkan sistem mempunyai *controller* dan model yang berinteraksi. *Method* *edit_profil_proses* merupakan *controller* dan *method* *edit_profil* merupakan model.

Kode Program 2 Kode untuk menangani pengeditan atau *input* profil

```

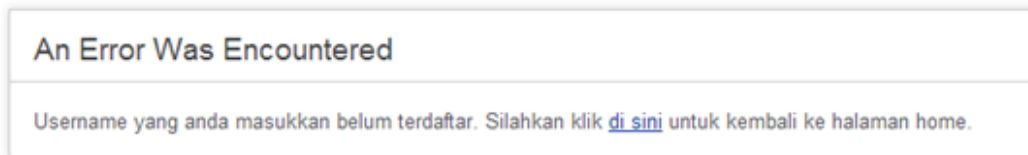
function edit_profil_proses() {
    $id_dsn = $this->session->userdata('id_user');
    $hp = $this->input->post('hp');
    $fb = $this->input->post('fb');
    $kab = $this->input->post('kab');
    $prov = $this->input->post('prov');

    $this->dsn_model->edit_profil(
        $id_dsn, $hp, $fb, $kab, $prov
    );
}

```


3.3.4 Implementasi Kendali

3.3.4.1 Kendali Ketika *Login*



Gambar 19 Implementasi Kendali Ketika *Login*

Gambar 19 memperlihatkan tampilan pesan kesalahan yang muncul jika data *login* yang dimasukkan tidak valid. Kode program yang digunakan untuk mengimplementasikan kendali ketika *login* dapat dilihat pada Kode Program 4.

Kode Program 4 Kode untuk menangani proses *login*

```
function sign_in_proses() {
    $uname = $this->input->post('username');
    $pass = $this->input->post('password');

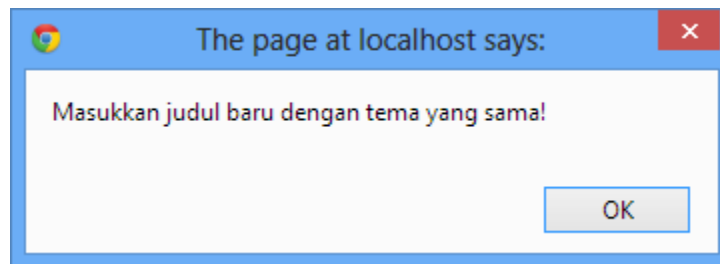
    $user_det = $this->get_user_det($uname, $pass);

    if ($user_det[0]) {
        // cek apakah kaprodi.
        $kap_cek = $this->dsn_model->is_kaprodi($user_det[1])
            ->num_rows();
        if ($kap_cek > 0) $kap = true;
        else $kap = false;
        // update session
        $data_user = array(
            'id_user' => $user_det[1],
            'nama_user' => $user_det[2],
            'user' => $user_det[3],
            'is_logged_in' => $user_det[0],
            'is_kaprodi' => $kap
        );
        $this->session->set_userdata($data_user);
        redirect('sipbta');
    } else {
        $back = anchor('', 'di sini');
        show_error("$user_det[1]. Silahkan klik ".$back.
            " untuk kembali ke halaman home.");
    }
}
```

3.3.4.2 Kendali Ketika Memasukkan Data

Gambar 20 memperlihatkan *form input* perubahan judul tugas akhir. Sistem akan memberikan *alert* (terlihat pada Gambar 21) jika *user* menekan tombol ubah judul tanpa mengisi *field* yang disediakan.

Gambar 20 Form Pengubahan Judul Tugas Akhir



Gambar 21 Implementasi Kendali Ketika Memasukkan Data

Kode program yang digunakan untuk menimplementasikan kendali tersebut dapat dilihat pada Kode Program 5.

Kode Program 5 Contoh kode untuk melakukan validasi *input* di sisi *client*

```
<script language="javascript">
function validateForm()
{
  var x=document.forms["ubah_jdl"]["jdl_baru"].value;
  if (x==null || x==" " || x==" ")
  {
    alert("Masukkan judul baru dengan tema yang sama!");
    return false;
  }
}
</script>
```

4. Penutup

Setelah melalui tahap analisis, perancangan dan implementasi sistem informasi pengelolaan bimbingan tugas akhir mahasiswa berbasis web, maka dapat ditarik kesimpulan sebagai berikut:

1. SILABITAMA telah dikembangkan dan disesuaikan dengan kondisi riil kampus STMIK Jend. A. Yani Yogyakarta dalam menangani pengelolaan bimbingan tugas akhir mahasiswa.
2. SILABITAMA telah menangani kompleksitas penanganan pengelolaan bimbingan tugas akhir pada dua program studi yang jumlah mahasiswanya terus bertambah seiring berjalannya waktu.

Daftar Pustaka

- Booch, G., Rumbaugh, J. & Jacobson, I., 1999. *The Unified Modeling Language User Guide*. Reading, Massachusetts, US: Addison Wesley.
- Chen, P. P.-S., 1976. The Entity-Relationship Model—Toward a Unified View of Data. *ACM Transactions on Database Systems (TODS)*, 1(1), pp. 9-36.
- Daqiqil, 2011. *Framework Codeigniter – Sebuah Panduan dan Best Practice*. [Online] Available at: <ftp://jran.undip.ac.id/pustaka/Framework%20Codeigniter%202.pdf> [Accessed 01/06/2013].
- Ellislab, 2013. *Welcome to CodeIgniter: CodeIgniter User Guide*. [Online] Available at: <https://ellislab.com/codeigniter/user-guide/index.html> [Accessed 01/06/2013].
- Fathansyah, 2012. *Basis Data*, Edisi Revisi. Bandung: Penerbit Informatika.
- Highcharts, 2013. *Highcharts – Highcharts product*. [Online] Available at: <http://www.highcharts.com/products/highcharts> [Accessed 01/06/2013].
- Jogiyanto, 2005. *Analisis dan Desain Sistem Informasi*. Yogyakarta: Penerbit Andi.
- jQuery, 2013. *jQuery*. [Online] Available at: <http://jquery.com> [Accessed 01/06/2013].
- McLeod, Jr. R., 2004. *Management Information Systems*, 9th Edition. London: Pearson.
- White, S. A., 2004. *Introduction to BPMN*. [Online] Available at: http://www.omg.org/bpmn/Documents/Introduction_to_BPMN.pdf [Accessed 01/06/2014].

