

APLIKASI WEB LOGBOOK BIMBINGAN TUGAS AKHIR MAHASISWA STMIK JENDERAL ACHMAD YANI YOGYAKARTA

Dewi Damayanti¹, Ari Cahyono²

Program Studi S1 Teknik Informatika
STMIK Jenderal Achmad Yani Yogyakarta

¹dewidamayanti345@gmail.com, ²arca@arcastudio.co.id

Abstrak

Logbook merupakan catatan harian yang sangat penting dalam sebuah penelitian, tesis, skripsi atau tugas akhir. Proses bimbingan penyelesaian tugas akhir mahasiswa STMIK Jenderal Achmad Yani terdiri dari beberapa tahap yaitu pengajuan proposal, ujian proposal, pembimbingan, pra-pendadaran (seminar tugas akhir akhir), dan pendadaran. Dalam proses pelaksanaannya, bimbingan Tugas Akhir di STMIK Jenderal Achmad Yani memerlukan sebuah sistem yang mampu memfasilitasi proses bimbingan dan meningkatkan kualitas bimbingan sesuai dengan kebutuhan dan prosedur yang berlaku.

Penelitian ini bertujuan untuk merancang dan membangun aplikasi web logbook bimbingan tugas akhir dengan menerapkan metode web engineering untuk menganalisis kebutuhan yang disesuaikan dengan sistem Logbook bimbingan yang ada di STMIK Jend. A. Yani dan diimplementasikan menggunakan bahasa pemrograman PHP dengan Framework CodeIgniter dan pattern HMVC (Hierarchical Mode View Controller).

Dengan adanya aplikasi web Logbook bimbingan TA, diharapkan dapat memudahkan mahasiswa maupun dosen pembimbing dalam melakukan bimbingan kapan dan dimana saja secara online.

Kata Kunci: *Logbook, Bimbingan Tugas Akhir Mahasiswa, Aplikasi Web.*

1. Pendahuluan

Tugas Akhir atau Skripsi adalah tahap akhir yang harus ditempuh oleh mahasiswa yang pada proses pembuatannya tentu membutuhkan pembimbing. Dalam prakteknya, aktifitas bimbingan tugas akhir dilakukan dengan diskusi, tanya jawab, pemberian masukan dan aktifitas lainnya. Selain itu, dalam melakukan aktifitas bimbingan TA mahasiswa maupun dosen pembimbing membutuhkan sebuah catatan kontak (*Logbook*) yang biasanya dalam bentuk kartu bimbingan tugas akhir. Kartu bimbingan tugas akhir diisi pada setiap pertemuan dan hasilnya akan direview pada setiap tahap perkembangan tugas akhir. Pelaksanaan aktifitas bimbingan tugas akhir bertujuan untuk mengontrol pengerjaan tugas akhir mahasiswa mulai dari pengajuan proposal hingga penyelesaian tugas akhir, sehingga diharapkan dengan adanya bimbingan ini maka dapat menghasilkan karya yang berkualitas tinggi.

Namun, terdapat beberapa kendala yang dihadapi dalam pelaksanaan bimbingan di STMIK Jenderal Achmad Yani. Jarak dan jadwal yang berbeda

antara dosen dan mahasiswa kadang kala menciptakan miskomunikasi sehingga proses bimbingan dapat terbengkalai. Selain itu, seringkali pelaksanaan prosedur standar seperti pencatatan formulir bimbingan tidak terorganisasi dengan baik atas faktor ketidaksengajaan sehingga dapat menurunkan kualitas bimbingan.

Aplikasi *Logbook* Bimbingan Tugas Akhir Mahasiswa STMIK Jenderal Achmad Yani dikembangkan guna menjembatani berbagai masalah dalam proses interaksi antara mahasiswa, dosen, dan pengelola program studi. Aplikasi ini dikembangkan dengan menerapkan metode *web engineering*, dan ditujukan untuk memfasilitasi proses bimbingan tugas akhir sebagai alternatif kedua untuk memudahkan mahasiswa dan dosen dalam melakukan bimbingan, kapan dan dimana saja secara *online* sesuai dengan persyaratan yang telah ditentukan.

2. Dasar Teori

2.1 Logbook

Berdasarkan Buku Catatan Penelitian (yang selanjutnya disebut *Logbook*) Lembaga Penelitian Universitas Negeri Gorontalo (2013), mengatakan bahwa Buku Catatan Kegiatan Penelitian (*Logbook*) merupakan buku hasil rekaman atau catatan yang dilakukan oleh peneliti pada setiap kegiatan penelitiannya.

2.2 Bimbingan

Bimbingan merupakan suatu proses yang mengandung pengertian bahwa bimbingan merupakan kegiatan yang berkesinambungan, bukan kegiatan seketika atau kebetulan (Walgito, 2010).

2.3 Web Application (WebApp)

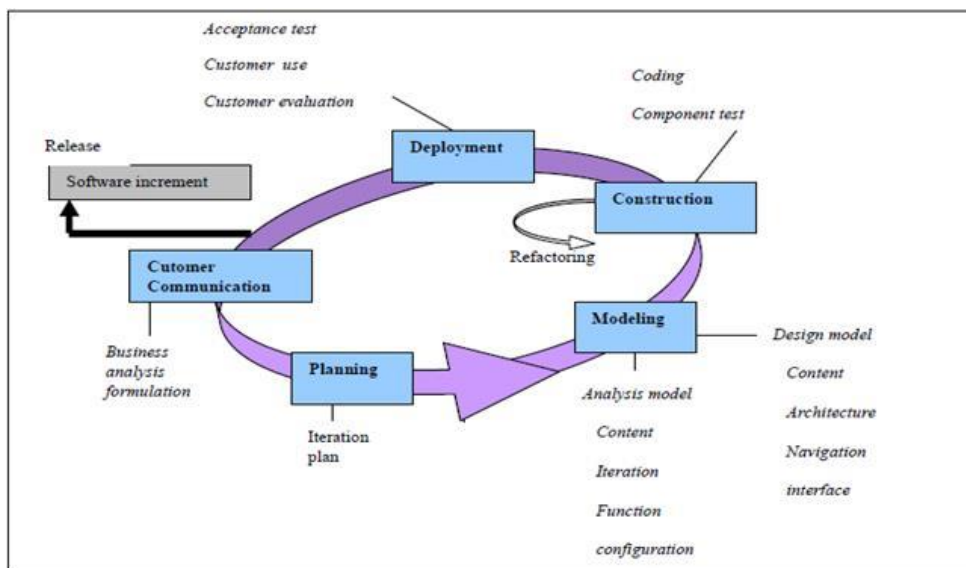
Menurut Pressman dan Lowe (2009), Aplikasi *web* atau *WebApp* adalah kategori perangkat lunak jaringan-sentris yang mencakup beragam aplikasi. *WebApp* dapat menjadi link *hypertext* informatif yang menggunakan teks dan grafis yang terbatas. Namun, dengan munculnya *Web 2.0*, *WebApp* berkembang menjadi lingkungan komputasi canggih yang tidak hanya menyediakan fitur *stand-alone* (berdiri sendiri), fungsi komputasi, dan *content* kepada pengguna akhir, tetapi juga terintegrasi dengan *database* perusahaan dan aplikasi bisnis.

2.4 Web Engineering (Rekayasa Web)

Web engineering (rekayasa *web*) adalah sebuah aplikasi yang menggunakan pendekatan sistematis, disiplin, dan terukur untuk pengembangan, operasi pemeliharaan aplikasi berbasis *web* (*Web-based application*)

(Simarmata, 2010). Dalam membangun sistem *logbook* bimbingan metode yang digunakan adalah metode *web engineering*, karena metode ini memberikan ide bagi pengembang maupun *user* tentang cara sistem akan berfungsi dan yang akan dikembangkan. *Web engineering* menerapkan pola *incremental delivery* yang disebabkan oleh permintaan (*requirement*) yang kerap kali mengalami perubahan hasil evaluasi (Pressman dan Lowe, 2009).

Secara umum, dalam metode *web engineering* terdapat 5 aktivitas. Kelima aktivitas pengembangan tersebut diulang untuk setiap tahapan pengembangan (*increment*) untuk memudahkan mengelola perubahan (Pressman dan Lowe, 2009) seperti pada Gambar 1.



Gambar 1 Tahapan-tahapan proses *web engineering* (Arifin, 2011)

Tahapan tersebut antara lain:

1. *Customer communication*

Komunikasi dalam hal ini terutama terkonsentrasi pada 2 hal, analisa bisnis dan perumusan. Analisa bisnis akan mendefinisikan hal-hal apa saja yang akan termuat di dalam aplikasi *web*, misalnya pengguna *web* yang akan dibangun, perubahan potensial dalam lingkungan bisnis, integrasi antara *web* yang akan dibangun dengan situasi bisnis perusahaan, maupun *database* perusahaan. Perumusan adalah pengumpulan informasi tentang hal-hal yang akan dimuat dalam *web* yang melibatkan semua calon pengguna.

2. *Planning*

Perencanaan proyek pengembangan aplikasi *web* kemudian ditentukan, perencanaan akan terdiri dari pendefinisian pekerjaan dan target waktu atas pekerjaan maupun sub pekerjaan yang ditentukan tersebut.

3. *Modeling*

Tujuan dari aktivitas ini adalah untuk menjelaskan hal-hal apa saja yang memang diperlukan atau dibutuhkan pada aplikasi yang akan dibangun dan solusi yang ditawarkan yang diharapkan dapat menjawab apa yang tersirat dari hasil-hasil analisa dan pengumpulan data.

4. *Construction*

Pembangunan aplikasi *web* memadukan antara perkembangan teknologi dengan *tools* pengembangan *web* yang telah ada, artinya memilih *tools* yang efektif namun tetap dapat menyesuaikan dengan teknologi yang berkembang saat ini.

5. *Deployment*

Aplikasi *web* diciptakan untuk dapat berguna bagi kebutuhan pekerjaan, dapat dioperasikan oleh *end-user*, dan kemudian dilakukan evaluasi secara berkala, memberi masukan-masukan kepada *team* pengembang dan apabila diperlukan akan dilakukan modifikasi pada aplikasi *web* tersebut.

2.5 Unified Modeling Language (UML)

Menurut Whitten, et al. (2004) dalam bukunya yang berjudul Metode Desain dan Analisis Sistem Edisi 6, dijelaskan bahwa Unified Modeling Language (UML) merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *software* yang terkait dengan objek. UML menawarkan diagram yang dikelompokkan menjadi lima perspektif berbeda untuk memodelkan suatu sistem, seperti satu *set 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*.

UML bukanlah suatu metode atau prosedur melainkan adalah “bahasa” yang digunakan untuk mendefinisikan suatu sistem perangkat lunak, untuk merinci artefak dari sistem untuk mendokumentasikan dan mengkonstruksi. UML digunakan dalam berbagai cara untuk mendukung metodologi pengembangan perangkat lunak, namun sendirinya tidak menentukan metodologi atau proses itu sendiri.

2.6 Entity Relationship Diagram (ERD)

ERD dikembangkan berdasarkan teori himpunan dalam bidang matematis. ERD digunakan untuk pemodelan basis data relasional sehingga jika penyimpanan basis data menggunakan OODBMS maka perancangan basis data

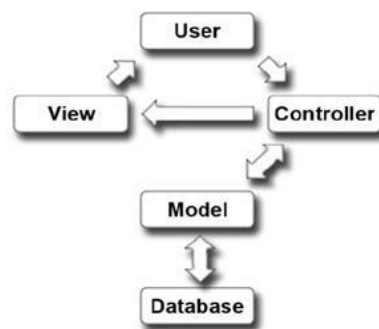
tidak perlu menggunakan ERD. ERD memiliki macam-macam notasi seperti aliran notasi.

Notasi-notasi simbolik ERD yang digunakan meliputi (Sukanto & Shalahuddin, 2011):

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 relasi entitas dimana di kedua ujungnya memiliki *multiplicity* kemungkinan jumlah pemakaian.

2.7 CodeIgniter Framework

CodeIgniter (CI) adalah *framework* pengembangan aplikasi (*Application Development Framework*) dengan menggunakan pemrograman PHP. CodeIgniter memiliki banyak fitur yang membuatnya berbeda dengan *framework* lainnya. Tidak seperti beberapa *framework* PHP lainnya, dokumentasi untuk *framework* ini sangat lengkap, yang mencakup seluruh aspek dalam *framework*. CodeIgniter juga mampu berjalan pada lingkungan *shared hosting* karena memiliki ukuran yang sangat kecil, namun memiliki kinerja yang sangat luar biasa (Sidik, 2012).



Gambar 2 Model-View-Controller

CodeIgniter menggunakan pola desain *Model-View-Controller* (MVC). Model pada Gambar 2 menjelaskan bahwa ketika datang sebuah permintaan dari *user*, maka permintaan tersebut akan ditangani oleh *controller*, kemudian *controller* akan memanggil *model* jika memang diperlukan operasi *database*. Hasil *query* oleh *model* kemudian akan dikembalikan ke *controller*. Selanjutnya *controller* akan memanggil *view* yang tepat dan mengkombinasikannya dengan hasil *query model*. Hasil akhir dari operasi ini akan ditampilkan ke *browser* yang selanjutnya bisa dilihat oleh *user* (Sidik, 2012).

3. Pembahasan

3.1 Analisis Sistem Berjalan

Menurut aturan yang terdapat pada Pedoman Penulisan Tugas Akhir STMIK Jend. A. Yani versi 2.0 (Prodi TI, 2014), proses bimbingan penyelesaian tugas akhir mahasiswa STMIK Jend. A. Yani terdiri dari beberapa tahap yaitu pengajuan proposal, ujian proposal, pembimbingan, pra-pendadaran (seminar tugas akhir), dan pendadaran.

Sekarang ini di STMIK Jendral A. Yani untuk masalah pengajuan dan bimbingan Tugas Akhir (TA) masih dilakukan dengan bertatap muka dengan dosen pembimbing secara langsung. Sedangkan jadwal pertemuan dengan dosen pembimbing terbatas jadwal yang disediakan oleh dosen pembimbing dan kadang harus mengantri, menunggu mahasiswa lain selesai melakukan bimbingan. Dari sudut pandang dosen pembimbing, juga mengalami kesulitan untuk membatasi lama waktu bimbingan seorang mahasiswa dan kadang sulit berkonsentrasi dikarenakan banyaknya materi dan judul yang harus dibahas dalam satu waktu bimbingan.

Secara lengkap, prosedur seorang mahasiswa yang ingin melakukan TA sebagai berikut:

1. Mahasiswa melakukan pengisian Kartu Rencana Studi (KRS) dan mengambil mata kuliah TA.
2. Menemui dosen pembimbing akademik dan/atau kaprodi, untuk konsultasi judul dan pembimbing TA.
3. Kaprodi kemudian akan menghubungi bagian Opsdik untuk memberikan daftar mahasiswa yang melakukan TA dan calon dosen pembimbing TA.
4. Dosen pembimbing TA akan diumumkan di portal akademik oleh bagian Opsdik.
5. Mahasiswa selanjutnya melakukan bimbingan proposal ke dosen pembimbing TA.
6. Jika proposal sudah disetujui, maka mahasiswa dapat mendaftarkan proposal TA ke bagian Opsdik, untuk mendaftarkan ujian proposal TA.
7. Opsdik memberitahukan pada mahasiswa melalui sms atau telepon mengenai jadwal ujian proposal.
8. Setelah mahasiswa melakukan ujian proposal TA dan dianggap layak, maka mahasiswa melanjutkan bimbingan TA sesuai jadwal dosen pembimbing.

9. Setiap melakukan bimbingan, mahasiswa diwajibkan membawa kartu bimbingan dan kemajuan TA.
10. Jika dosen pembimbing menganggap TA mahasiswa layak untuk diseminarkan, maka mahasiswa kemudian mendaftar ujian seminar hasil TA ke bagian Opsdik.
11. Selanjutnya bagian Opsdik akan memberitahukan jadwal seminar kepada mahasiswa melalui SMS dan/atau telepon.
12. Ujian seminar hasil dihadiri oleh dosen pembimbing dan rekan-rekan mahasiswa.
13. Setelah melakukan seminar hasil, mahasiswa kemudian melakukan revisi dan bimbingan lanjutan sampai dinyatakan oleh dosen pembimbing layak untuk ujian pendadaran.
14. Setelah dinyatakan layak untuk pendadaran, mahasiswa membawa naskah rangkap 4 untuk didaftarkan di bagian Opsdik.
15. Opsdik akan memberitahukan jadwal ujian pendadaran kepada mahasiswa melalui SKS dan/atau telepon.
16. Pendadaran dihadiri dua dosen penguji dan satu dosen pembimbing, secara tertutup, tanpa dihadiri rekan-rekan mahasiswa.
17. Setelah pendadaran, mahasiswa melakukan revisi yang disarankan oleh dosen penguji dan dosen pembimbing
18. Setelah revisi selesai, mahasiswa menemui kaprodi untuk persetujuan naskah.
19. Setelah naskah disetujui semua dosen (penguji, pembimbing dan kaprodi) maka naskah kemudian dijilid, kemudian diserahkan ke bagian opsdik.
20. Bagian opsdik, kemudian akan menemui kaprodi untuk penentuan judul TA mana saja yang akan dibuat jurnal dan dipublikasikan.

Dari uraian prosedur tersebut, dapat dilihat beberapa hal yang kurang efisien dalam pelaksanaan TA yang dapat didenifikasikan sebagai berikut:

1. Kaprodi akan bertemu banyak mahasiswa untuk berkonsultasi judul dan pembimbing TA.
2. Bagian Opsdik hanya memberitahukan kepada mahasiswa tentang jadwal ujian melalui SMS dan/atau telepon.
3. Kegiatan bimbingan sering terhambat karena dosen pembimbing berhalangan, sehingga kegiatan bimbingan tidak sesuai dengan jadwal yang telah ditetapkan.

4. Khusus untuk mahasiswa yang tinggal di luar kota atau yang sudah bekerja, proses bimbingan menjadi hal yang kurang disenangi, karena akan menyita waktu dan materi yang tidak sedikit.
5. Sulit memonitor kemajuan proses TA mahasiswa, karena *log* pertemuan bimbingan hanya terdapat pada kartu bimbingan.

Berdasarkan analisis masalah tersebut, diambil kesimpulan berikut:

1. Diperlukan sebuah sistem pendaftaran dan bimbingan secara *online*, sehingga mahasiswa dapat melakukan proses TA, kapan dan dimana saja secara *online* sesuai dengan persyaratan yang telah ditentukan.
2. Diperlukan *database* tersendiri guna menyimpan data yang diperlukan dalam proses TA.
3. Proses TA yang tidak dapat dilakukan secara *online*, seperti penyerahan berkas dan tanda tangan persetujuan tetap dilakukan secara *offline*, namun datanya dapat dimasukkan dan diunggah ke dalam sistem untuk keperluan pengarsipan proses TA.

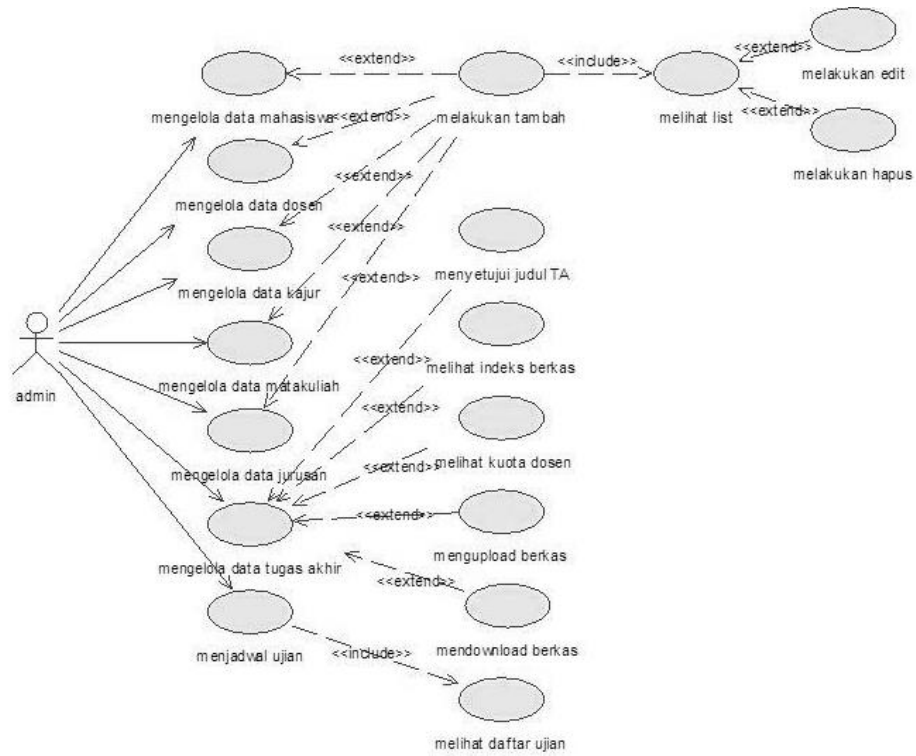
Pembangunan sistem pendaftaran dan bimbingan *online* TA (*Logbook* TA) mahasiswa di STMIK A Yani dirancang untuk mempermudah mahasiswa dalam melakukan proses pendaftaran TA, pendaftaran judul, proses bimbingan dan pendaftaran ujian.

3.2 Perancangan

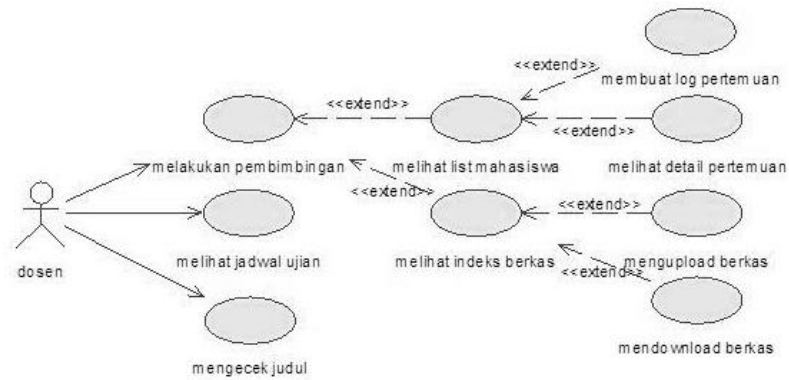
Perancangan dalam membangun Sistem *Logbook* tugas akhir melibatkan 4 (empat) pengguna sistem (aktor), yaitu:

1. Admin/Operasional Pendidikan (Opsdik)
2. Kajar/Kaprodi
3. Dosen Pembimbing
4. Mahasiswa

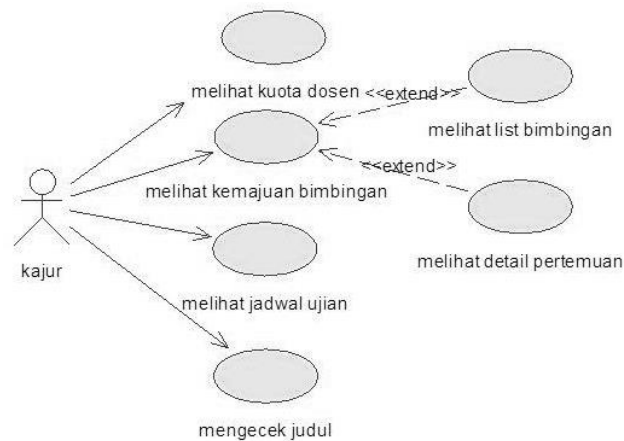
Kegiatan empat aktor tersebut akan dijelaskan melalui *use case diagram* dan *activity diagram*. *Use case diagram* pada aplikasi ini dapat dilihat pada Gambar 3 adalah *use case* admin yang menunjukkan kegiatan yang dilakukan oleh aktor admin, yang dalam sistem merupakan pengelola utama dalam sistem, Gambar 4 adalah *use case* kajar yang menunjukkan kegiatan yang dilakukan oleh aktor kajar, Gambar 5 adalah *use case* dosen pembimbing yang menunjukkan kegiatan yang dilakukan oleh aktor dosen dan Gambar 6 adalah *use case* mahasiswa yang menunjukkan kegiatan yang dilakukan oleh aktor mahasiswa.



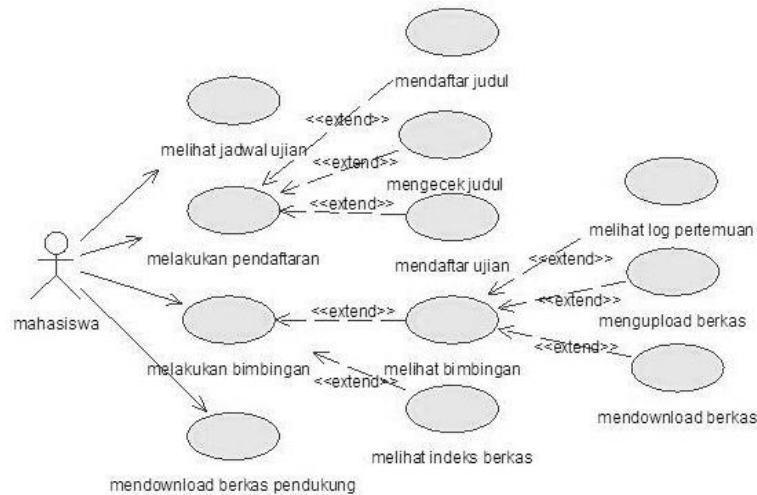
Gambar 3 Use case admin



Gambar 4 Use case dosen

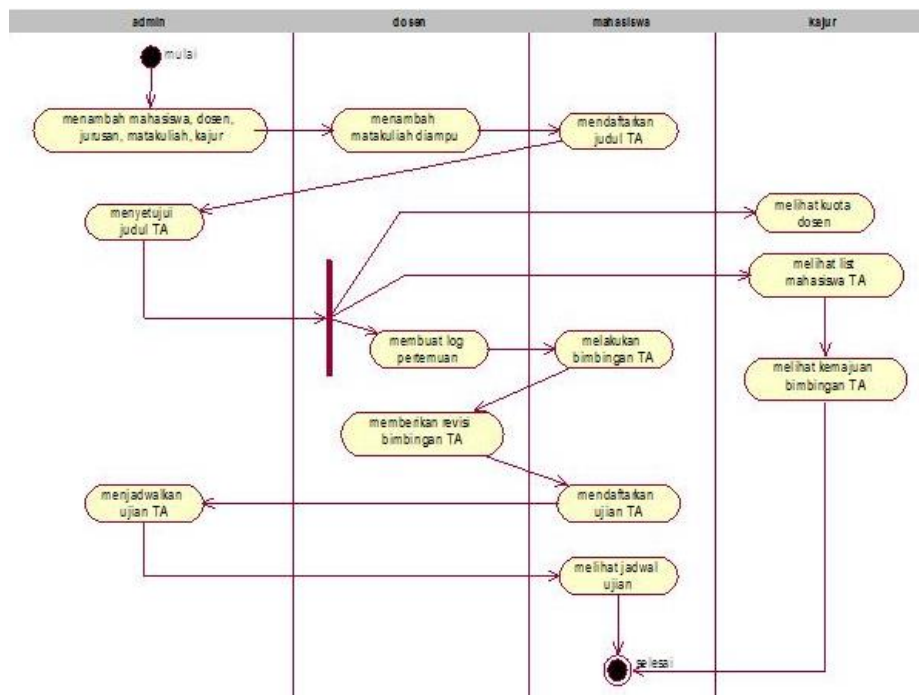


Gambar 5 Use case kajar/kaprodi



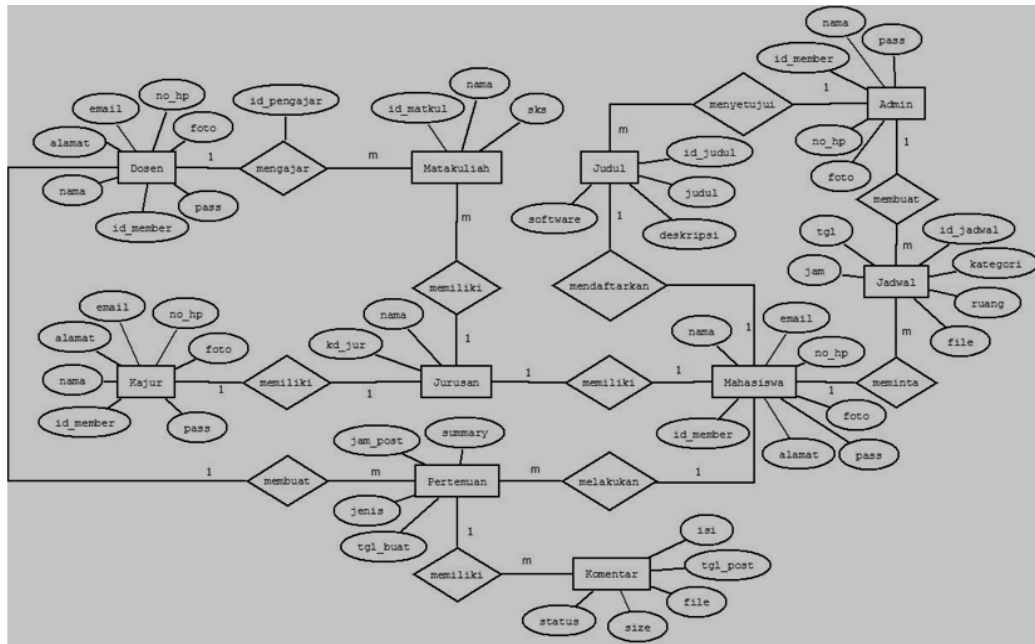
Gambar 6 Use case mahasiswa

Activity diagram disajikan pada Gambar 7 yang mengilustrasikan kegiatan bimbingan TA dan menunjukkan alur kegiatan umum bimbingan TA dari keempat aktor atau pengguna aplikasi tersebut. Kegiatan bimbingan TA merupakan kegiatan utama yang melatarbelakangi dibuatnya aplikasi.



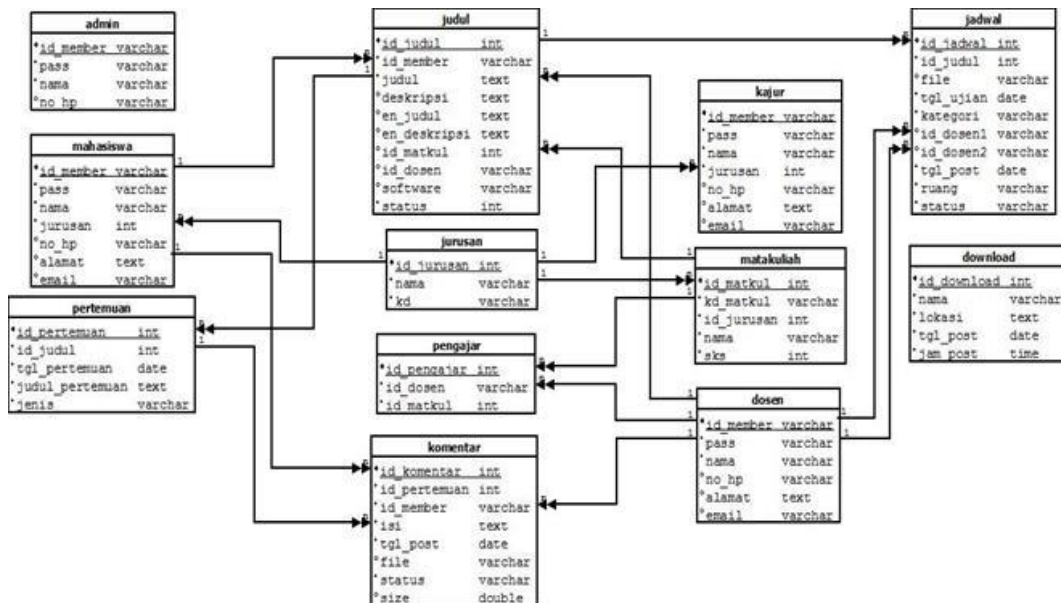
Gambar 7 Activity diagram kegiatan bimbingan TA

Setelah desain perancangan keempat pengguna sistem tersebut, berikutnya adalah menggambarkan *Entity Relationship Diagram* (ERD) yang bertujuan untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan relasi yang terdapat dalam aplikasi *Logbook Tugas Akhir* yang dapat dilihat pada Gambar 8.



Gambar 8 Entity Relationship Diagram Logbook

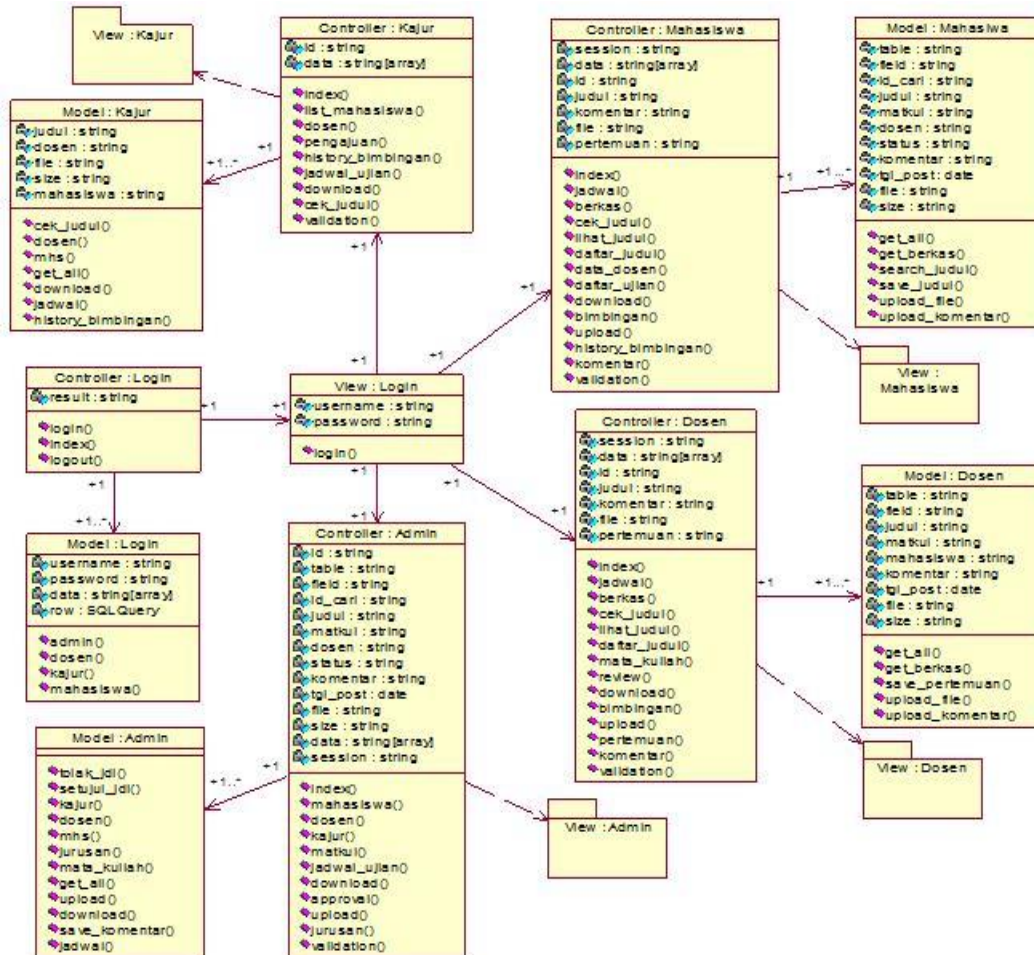
Perancangan selanjutnya dilanjutkan dengan perancangan database dalam bentuk model relasional. Model ini menunjukkan cara mengelola/mengorganisasi data secara fisik yang terdapat pada sistem Logbook TA yang dapat dilihat pada Gambar 9.



Gambar 9 Model Relasional Logbook

Aplikasi Logbook TA memiliki 12 tabel, yaitu: tabel admin, tabel dosen, tabel download, tabel jadwal, tabel judul, tabel jurusan, tabel kajur, tabel komentar, tabel mahasiswa, tabel matakuliah, tabel pengajar, dan tabel pertemuan.

Berdasarkan model relasional *logbook* tersebut dibuatlah *class diagram* untuk menggambarkan relasi antar kelas dalam aplikasi serta atribut-atribut dan fungsi-fungsi yang melekat pada kelas-kelas di dalam sistem. *Class diagram* pada Gambar 10 menggambarkan rancangan MVC yang digunakan dalam aplikasi.



Gambar 10 Class diagram Logbook

Aplikasi *Logbook* dirancang menggunakan teknik HMVC (*Hierarchy Model View Controller*) sehingga pada bagian utama aplikasi terdapat modul-modul yang menggambarkan aktor dalam aplikasi dan pada *view* pada tiap modul terdapat banyak objek-objek tampilan. Objek-objek tersebut disatukan dalam *package view* dalam setiap modul.

Fungsi dari kelas *controller* sebagai pengatur alur data dari perintah pengguna, kelas model dan kelas *view*, sedangkan kelas model berisi *query-query* SQL dalam bentuk sintaks CodeIgniter dan kelas *view* berisi kode-kode CSS, HTML, JavaScript, jQuery, Ajax dan kode PHP yang berfungsi untuk menampilkan *interface* aplikasi kepada pengguna.

3.3 Implementasi

Implementasi pada sistem *Logbook* TA menggunakan bahasa pemrograman PHP dan *framework* CodeIgniter dengan teknik modular HMVC. Teknik HMVC digunakan karena teknik modular dapat mengatasi kompleksitas pemrograman dalam mengimplementasikan perancangan ke dalam bentuk pemrograman PHP.

Akses masuk ke sistem *Logbook* TA diawali dengan melakukan *login* sesuai hak akses pengguna sistem. Tabel 1 menunjukkan fungsi-fungsi modul *controller login* yang berfungsi sebagai pengatur dan pengolah aliran data dari masukan pengguna, penyimpanan *database* dan tampilan.

Tabel 1 Fungsi-fungsi *Controller Login*


Fungsi	Sintaks & Keterangan	
__construct	<code>\$this->load->Model('db_login');</code>	Konstruktor kelas login untuk memuat model <i>database</i> bernama 'db_login'
index	Fungsi index adalah fungsi yang akan dijalankan pertama kali ketika kelas login dipanggil	
	<code>\$this->load->View('login/View');</code>	Memuat <i>view</i> tampilan pada kelas login/View.php
logout	Fungsi logout adalah fungsi keluar dari aplikasi	
	<code>\$this->session ->unset_userdata('login');</code>	Menghapus data sesi <i>login</i>
	<code>redirect('login', 'refresh');</code>	Alihkan/jalankan fungsi login
m_login	Fungsi yang akan dijalankan jika pengguna melakukan <i>login</i>	
	<pre>\$result = \$this->db_login ->mahasiswa(); if (!\$result) { \$result = \$this->db_login ->dosen(); if (!\$result) { \$result = \$this->db_login ->kajur(); if (!\$result) { \$result = \$this->db_login ->admin(); if (!\$result) { redirect('login/error'); } else redirect('admin'); } else redirect('kajur'); } else redirect('dosen'); } else redirect('mahasiswa');</pre>	Jalankan fungsi mahasiswa pada kelas Model, jika hasilnya false jalankan fungsi dosen, jika hasilnya false jalankan fungsi kajur, jika hasilnya false jalankan fungsi admin, jika semua false jalankan fungsi error(). Jika hasilnya true pada salah satu hasil result maka jalankan fungsi sesuai modul pengguna
error	Fungsi yang akan dijalankan jika terjadi kesalahan <i>login</i>	
	<code>\$this->load ->View('login/error');</code>	Memuat <i>view</i> tampilan pada kelas login/error.php

Fungsi pada kelas *controller login* akan diteruskan ke fungsi kelas model *login* yang terdapat pada Tabel 2 untuk koneksi *database*, sedangkan masukan dari kelas *view login* akan menjalankan fungsi validasi dari kelas *controller login* dan hasilnya akan diteruskan ke kelas *view* admin, *view* dosen, *view* kajur atau

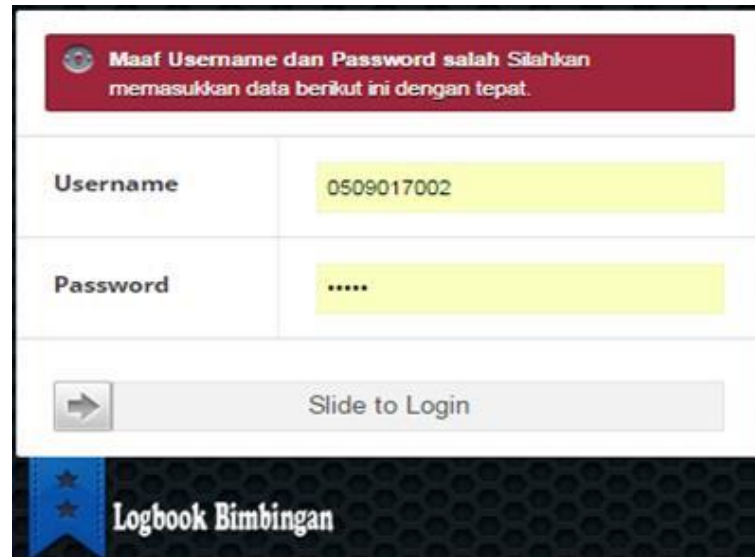
view mahasiswa tergantung dari *username* dan *password* yang dimasukkan pengguna. View *input login* dapat dilihat pada Gambar 11 dan *view error login* terdapat pada Gambar 12 jika pengguna aplikasi melakukan kesalahan dalam melakukan login.

Tabel 2 Fungsi-fungsi Model *Login*

Fungsi	Sintaks & Keterangan
mahasiswa, dosen, kajur, admin	<pre>// simpan nilai dari form post ke variabel echo \$username = \$this->security ->xss_clean(\$this->input->post('username')); echo \$password = \$this->security ->xss_clean(\$this->input->post('password')); // query ambil data mahasiswa dengan syarat \$username & \$password \$this->db->where('id_member', \$username); \$this->db->where('pass', \$password); \$query = \$this->db->get('mahasiswa'); // Cek apakah user tersebut ada?? if (\$query->num_rows != 0) { // Jika user ada maka buat sesi yang berisi // id_member, nama, foto dan validated = true \$row = \$query->row(); \$data = array('id_member' => \$row->id_member, 'nama' => \$row->nama, 'foto' => \$row->foto, 'validated' => true); \$this->session->set_userdata("login", \$data); return true; } // Jika tidak maka hasil = false return false;</pre>

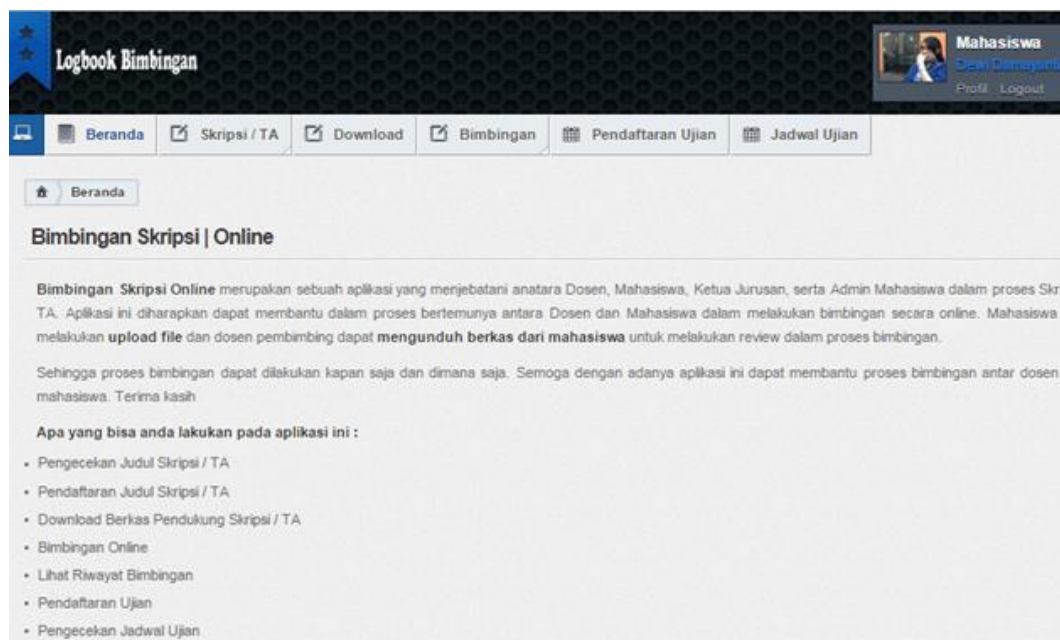


Gambar 11 Tampilan *View Login*



Gambar 12 Tampilan *View Login Error*

Halaman beranda Mahasiswa adalah halaman pertama ketika mahasiswa berhasil melakukan *login*. Pada halaman ini terdapat beberapa menu yang dapat dilakukan oleh Mahasiswa yaitu skripsi/TA, *download*, pendaftaran ujian, dan jadwal ujian. Tampilan hasil implementasi halaman beranda Mahasiswa dapat dilihat pada Gambar 13.



Gambar 13 Tampilan Beranda Mahasiswa

Pertama kali yang dilakukan mahasiswa pada saat *login* adalah mengklik halaman pendaftaran judul TA untuk pendaftaran judul TA mahasiswa untuk di *approve* oleh admin. Tampilan hasil implementasi halaman pendaftaran judul TA dapat dilihat pada Gambar 14.

[Beranda](#) > [Skripsi](#) > [Pendaftaran Judul](#)

Riwayat Pendaftaran Judul

Nim : 11.0259
 Nama : Dewi Damayanti
 Judul : Aplikasi Web Logbook Bimbingan Tugas Akhir Mahasiswa STMIK Jenderal Achmad Yani

Deskripsi : Sistem Logbook Tugas akhir (selanjutnya disebut 'Logbook') adalah sistem informasi yang akan mencatat kemajuan Tugas Akhir mahasiswa, mulai dari pengajuan proposal hingga penyelesaian Tugas Akhir. Dosen pembimbing dan mahasiswa diwajibkan melengkapi isian pada Logbook, serta mencatat setiap pertemuan yang berkaitan dengan kemajuan penelitian. Catatan dalam sistem logbook adalah catatan resmi dari kontak antara mahasiswa dan dosen pembimbing. Data yang tersedia pada Logbook harus tersedia untuk di-review pada setiap tahap perkembangan Tugas Akhir.

Title : The web application guidance logbook final project STMIK Jend. Yani

Description : The final task Logbook System (hereinafter referred to as 'Logbook') is an information system that will record the progress of the students final project, ranging from proposals to REMEDY Final. Lecturers and students are required to complete the fields in the Logbook and record any meeting relating to the progress of research. Note in the logbook system is the official record of contact between students and lecturers. Data available on the Logbook should be available for review at any stage of development Final.

Software : XAMPP, Notepad++
 Dosen Pembimbing : Ari Cahyono, S.Si., M.T.
 Status : **Disetujui**

Keterangan Status :

- **Belum Disetujui**, Judul dan Dosen Pembimbing belum direview oleh Ketua Jurusan
- **Ditolak**, Judul atau Dosen Pembimbing di tolak oleh Ketua Jurusan. Silahkan mengganti Judul / Dosen pembimbing yang lain
- **Disetujui**, Judul dan Dosen Pembimbing disetujui oleh Ketua Jurusan. Silahkan melanjutkan ke tahap bimbingan
- **Error**, Sistem error. Silahkan laporkan ke admin


Gambar 14 Tampilan Riwayat Pendaftaran Judul


Bimbingan Skripsi / TA


Berikut ini adalah riwayat Bimbingan Skripsi / TA anda. Silahkan melihat serta melakukan review hasil bimbingan dengan dosen anda. Terimakasih


Pertemuan ke 1 — 2015-06-25 | [Proposal](#) | [online](#)

Review/ Tanggapan

 Bagaimana proses bisnis dari proposalnya bisa dijelaskan.
 Ari Cahyono,
 S.Si., M.T.
 2015-06-24

 oiya pak tentang logbook, ini bab 1 nya
 Dewi Damayanti
 2015-06-24

 oh ya mbak lngsung daftar ujian proposalnya aja
 Ari Cahyono,
 S.Si., M.T.
 2015-06-24

 Glop Pak
 Dewi Damayanti
 2015-06-24

Bab_1_Proposal.docx
 | 210.64 Kb

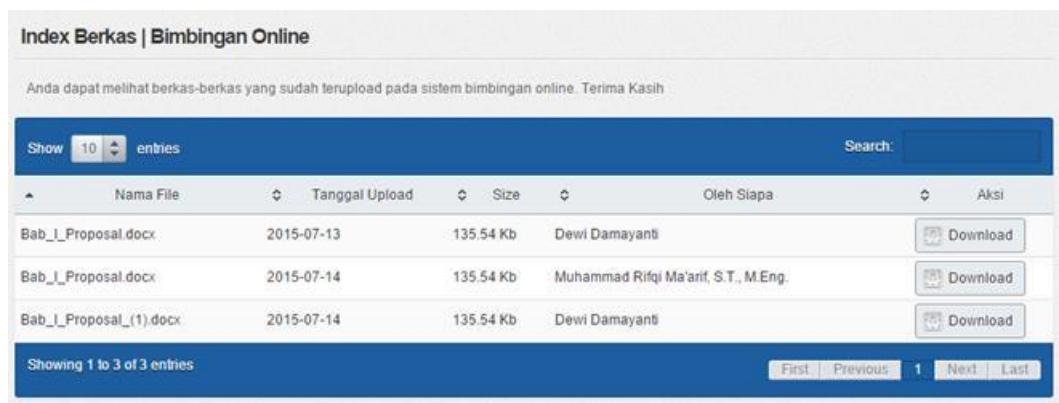
Komentar

Berkas

No file selected

Gambar 15 Tampilan Bimbingan TA

Jika judul TA telah di-*approve* oleh admin, maka mahasiswa dapat secara otomatis masuk ke dalam daftar kuota bimbingan dosen pembimbing dan dapat melakukan bimbingan TA secara *online* dengan dosen pembimbing TA, memberikan tanggapan atau komentar, meng-*upload* berkas bimbingan dari mahasiswa dan men-*download* berkas revisi seperti yang terlihat pada Gambar 15. Catatan kontak (*logbook*) antara mahasiswa dan dosen pembimbing akan tersimpan pada indeks berkas bimbingan TA seperti yang terlihat pada Gambar 16, sebagai tempat penyimpanan data-data yang di-*upload* oleh mahasiswa maupun dosen selama melakukan bimbingan TA, dan kemudian data tersebut akan di-*review* pada setiap tahap perkembangan tugas akhir.



Index Berkas | Bimbingan Online

Anda dapat melihat berkas-berkas yang sudah terupload pada sistem bimbingan online. Terima Kasih

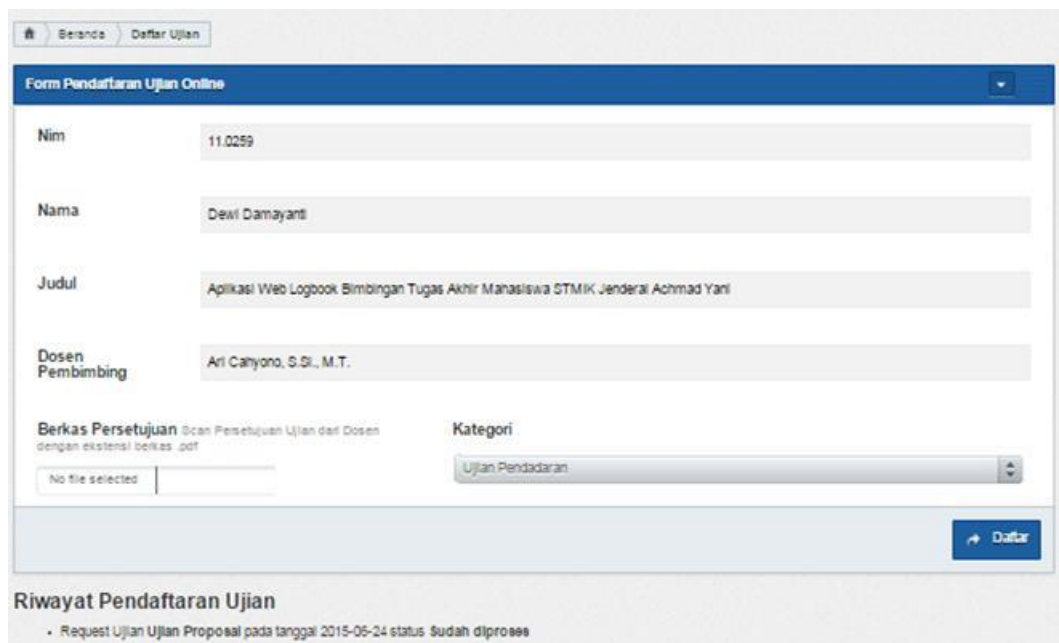
Show 10 entries Search:

Nama File	Tanggal Upload	Size	Oleh Siapa	Aksi
Bab_I_Proposal.docx	2015-07-13	135.54 Kb	Dewi Damayanti	Download
Bab_I_Proposal.docx	2015-07-14	135.54 Kb	Muhammad Rifqi Ma'arif, S.T., M.Eng.	Download
Bab_I_Proposal_(1).docx	2015-07-14	135.54 Kb	Dewi Damayanti	Download

Showing 1 to 3 of 3 entries

First Previous 1 Next Last

Gambar 16 Tampilan Indeks Berkas TA



Beranda Daftar Ujian

Form Pendaftaran Ujian Online

Nim: 11.0259

Nama: Dewi Damayanti

Judul: Aplikasi Web Logbook Bimbingan Tugas Akhir Mahasiswa STMIK Jenderal Achmad Yani

Dosen Pembimbing: Ari Cahyono, S.Si., M.T.

Berkas Persetujuan: Scan Persetujuan Ujian dari Dosen dengan ekstensi berkas .pdf
No file selected

Kategori: Ujian Pendadaran

Daftar

Riwayat Pendaftaran Ujian

- Request Ujian Ujian Proposal pada tanggal 2015-06-24 status: sudah diproses

Gambar 17 Tampilan Pendaftaran Ujian

Halaman pendaftaran ujian adalah halaman untuk pendaftaran ujian mahasiswa. Pada halaman ini mahasiswa dapat meminta kepada admin untuk

dilakukan penjadwalan ujian. Mahasiswa diwajibkan untuk meng-*upload* berkas persetujuan ujian dan memilih jenis ujian yang diajukan. Tampilan hasil implementasi halaman pendaftaran ujian dapat dilihat pada Gambar 17.

3.4 Pengujian

3.4.1 Uji Coba *Black box* (*Black box Testing*)

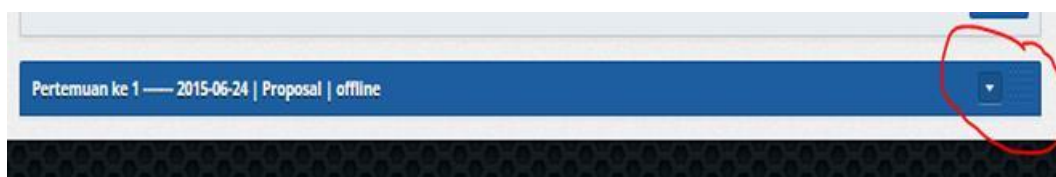
Pengujian *black box* bertujuan untuk menunjukkan fungsi perangkat lunak tentang cara beroperasinya. Apakah proses memasukkan data telah berjalan sebagaimana mestinya dan apakah informasi yang tersimpan dapat dijaga keamanannya. Tabel 3 menunjukkan metode pengujian *black box* pada beberapa *form* utama aplikasi.

Tabel 3 Hasil pengujian *Black box*

No	Interface	Keterangan
1	Form Login	Sudah berjalan dengan baik, dan akan memunculkan pesan kesalahan jika terjadi kesalahan masukan
2	Form Penambahan, Edit dan Hapus	Sudah berjalan dengan baik dan pesan kesalahan sudah muncul
3	Form Pendaftaran	Sudah berjalan dengan baik dan diberi penanda This Field is required.
4	Form Upload	Sudah bekerja baik dan diberi petunjuk berkas yang dapat di- <i>upload</i>
5	Tampilan Profil	Sudah bekerja baik, muncul <i>icon</i> awal jika gambar tidak tersedia

3.4.2 Uji Coba *White box* (*White box Testing*)

White box testing adalah metode pengujian desain *test case* yang menggunakan struktur kontrol desain secara prosedural untuk memperoleh *test case*. Perbaikan pada hasil pengujian *white box*, dapat dilakukan selama pembuatan aplikasi atau ketika ada penambahan fitur yang menyebabkan perubahan pada struktur aplikasi. Misal pada aplikasi *Logbook* bimbingan ini, diperlukan penggantian *template view* untuk dapat membuat mahasiswa dan dosen tidak perlu membuka ulang *toggle bar* setiap memberikan komentar seperti ditunjukkan pada Gambar 18.



Gambar 18 *Toogle Bar* Pertemuan Bimbingan

Pengembangan aplikasi juga perlu dilakukan agar aplikasi dapat menampilkan notifikasi pada halaman beranda, jika ada *review* atau komentar baru yang masuk. Selain itu, pola manajemen tugas akhir seperti fungsi *approve* judul TA yang sebelumnya dilakukan oleh *user* dengan *role* admin dapat dengan mudah dipindahkan ke *user* dengan *role* kajar atau kaprodi.

Database pada aplikasi belum mendukung *transaction*, sehingga menimbulkan potensi masalah saat terjadi *concurrent access* pada tabel yang sama, misalkan penetapan dosen pembimbing TA oleh beberapa kaprodi secara bersamaan.

4. Penutup

Kompleksitas dalam pemrograman aplikasi *Logbook* bimbingan tugas akhir dapat diatasi dengan menggunakan teknik pemrograman *framework* CodeIgniter *Hierarchycal Mode View Controller* (CI HMVC), yang memudahkan *programmer* dalam melakukan perawatan serta pengembangan aplikasi.

Perancangan sistem *Logbook* tugas akhir dengan menggunakan metode *web engineering* merupakan solusi untuk membangun sebuah sistem yang bisa dilakukan secara cepat dan memungkinkan untuk merubah kembali perangkat lunak agar sesuai dengan kebutuhan sistem *Logbook* bimbingan yang ada di STMIK Jend. A. Yani.

Perbaikan kesalahan yang ditemukan pada pengujian *black box* dilakukan selama implementasi dan pemrograman aplikasi, sedangkan perbaikan pada kesalahan pengujian *white box* dilakukan setelah aplikasi diunggah secara *online* dan diberikan pengujian *stress testing*.

Daftar Pustaka

- Arifin, M., 2011. Sistem Informasi Support Ticket Sebagai Layanan Bantuan Untuk Pelanggan Di Perusahaan Web Hosting. *Skripsi*. Bandung: Unikom.
- Kadir, A., 2008. *Dasar Pemrograman Web Dinamis Menggunakan PHP*. Yogyakarta: Penerbit Andi.
- Kadir, A., 2013. *Pemrograman Database MySQL Untuk Pemula*. Yogyakarta: Media Kom.
- Lembaga Penelitian Universitas Negeri Gorontalo, 2013. *Buku Catatan Kegiatan Penelitian (Logbook)*. Gorontalo: Lembaga Penelitian Universitas Negeri Gorontalo.

- Pressman, R.S. & Lowe, D., 2009. *Web Engineering A Practitioner's Approach*. New York: The McGraw-Hill Higher Companies, Inc.
- Prodi TI, 2014. *Pedoman Penulisan Tugas Akhir versi 2.0*. Yogyakarta: Program Studi Teknik Informatika STMIK Jenderal Achmad Yani.
- Sidik, B., 2012. *Pemrograman PHP Dengan Menggunakan Framework CodeIgniter 2*. Bandung: Informatika.
- Simarmata, J., 2010. *Rekayasa Web*. Yogyakarta: Penerbit Andi.
- Sukamto, R.A & Shalahudin, M., 2011. *Rekayasa Perangkat Lunak*, Edisi 1. Bandung: Modula.
- Walgito, B., 2010. *Bimbingan dan Konseling (Studi & Karier)*. Yogyakarta: Penerbit Andi.
- Whitten, J.L., Bentley, L.D. & Dittman, K.C., 2004. *Metode Desain dan Analisis Sistem*, Edisi 6. Yogyakarta: Penerbit Andi.
- Winkel, W. & Hastuti, M.S., 2010. *Bimbingan dan Konseling di Institusi Pendidikan*. Yogyakarta: Media Abadi.