



TUGAS AKHIR - KS09 1336

PEMBUATAN SISTEM INFORMASI PENERIMAAN BEASISWA INTERNAL DIREKTORAT JENDERAL PERBENDAHARAAN DENGAN TEKNOLOGI JAVASERVER PAGES DAN MYSQL

FITRIA YULIANTO
NRP 5211 105 708

Dosen Pembimbing
Dr.Eng. Febriliyan Samopa, S.Kom. M.Kom

JURUSAN SISTEM INFORMASI
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2015



FINAL PROJECT - KS09 1336

DEVELOPMENT OF DIRECTORATE GENERAL OF TREASURY INTERNAL SCHOLARSHIP ADMISSION INFORMATION SYSTEM USING JAVASERVER PAGES AND MYSQL TECHNOLOGY

**FITRIA YULIANTO
NRP 5211 105 708**

**Supervisor
Dr.Eng. Febriliyan Samopa, S.Kom., M.Kom**

**INFORMATION SYSTEM DEPARTMENT
Information Technology Faculty
Institut Teknologi Sepuluh Nopember
Surabaya 2015**

**PEMBUATAN SISTEM INFORMASI PENERIMAAN
BEASISWA INTERNAL DIREKTORAT JENDERAL
PERBENDAHARAAN DENGAN TEKNOLOGI
JAVASERVER PAGES DAN MYSQL**

TUGAS AKHIR

**Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada**

**Jurusan Sistem Informasi
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember**

Oleh:

FITRIA YULIANTO
5211 105 708

Surabaya, 2015

**KETUA
JURUSAN SISTEM INFORMASI**

Dr.Eng. Febriliyan Samopa, S.Kom. M.Kom
NIP. 197302191998021001

**PEMBUATAN SISTEM INFORMASI PENERIMAAN
BEASISWA INTERNAL DIREKTORAT JENDERAL
PERBENDAHARAAN DENGAN TEKNOLOGI
JAVASERVER PAGES DAN MYSQL**

TUGAS AKHIR

Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Jurusan Sistem Informasi
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh :

FITRIA YULIATO

5211 105 708

Disetujui Tim Penguji :

Tanggal Ujian : 16 Januari 2015

Periode Wisuda : Maret 2015

Dr. Eng. Febriliyan Samopa, S.Kom. M.Kom (Pembimbing I)

Edwin Riksakomara, S.Kom. M.T.

(Penguji 1)

Faizal Johan Atletiko, S.Kom. M.T.

(Penguji 2)

DEVELOPMENT OF DIRECTORATE GENERAL OF TREASURY INTERNAL SCHOLARSHIP ADMISSION INFORMATION SYSTEM USING JAVASERVER PAGES AND MYSQL TECHNOLOGY

Name : **Fitria Yulianto**
NRP : **5211 105 708**
Department : **Information System FTIf-ITS**
Supervisor : **Dr. Eng. Febriliyan Samopa, S.Kom,
M.Kom**

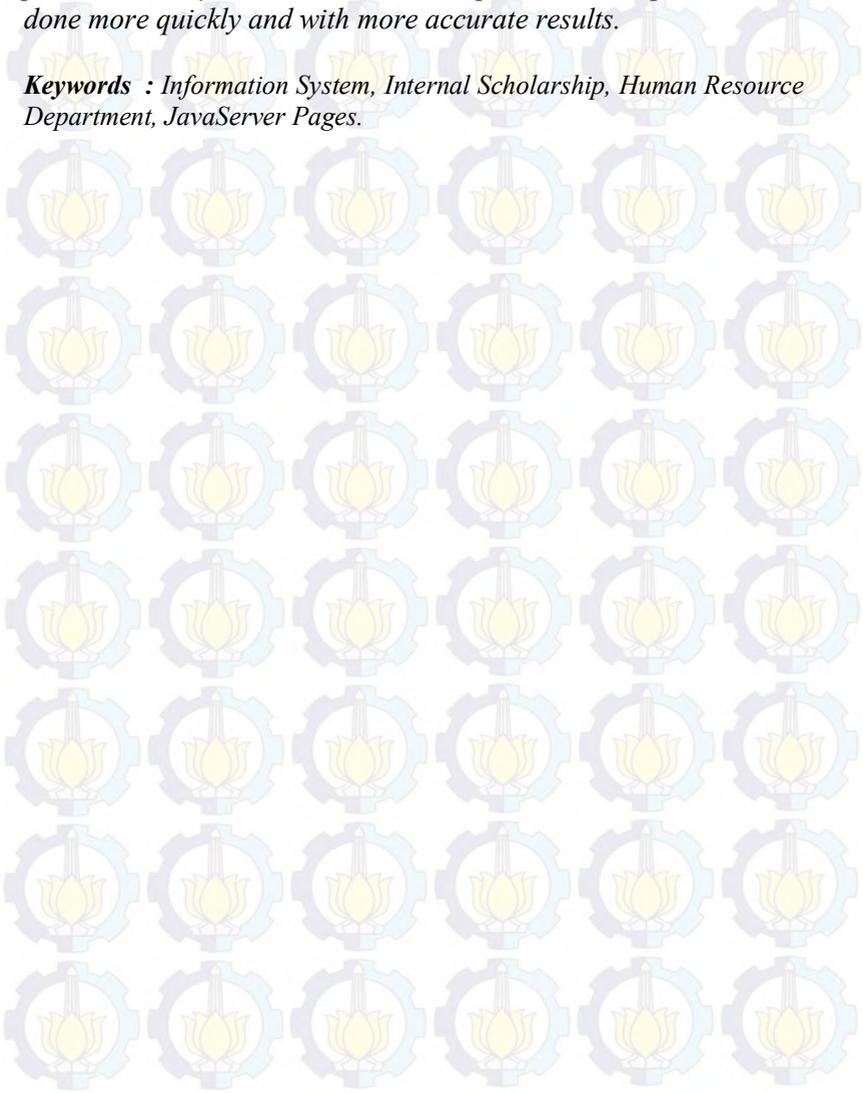
ABSTRACT

One of the function run by Secretariat Directorate General of Treasury (Human Resource Department) is planning requirement, selecting participant, and evaluating training and education program. In order to perform these function, Human Resource Department held internal scholarship selection for employees of Directorate General of Trasury every year. The scholarship program held for bachelor,s degree, master,s degree, and doctoral degree with a collaboration involving several universities both domestic and abroad.

To obtain the internal scholarship, employees must go through three stages of selection i.e administrative selection, written test, and interview. Administrative selection process done manually by Human Resource Department so it takes quite a long time and have a risk of error checking. In addition, Human Resource Department also requires a mechanism to prevent double scholarship nomination to the same employee. To overcome this an information system need to be made that can enhance the Internal Scholarship selection process. Through this final project the author will develop Internal Scholarship Admission Information System using JavaServer Pages technology and MySQL.

With this Information System, the Scholarship selection process held by Human Resource Department is expected to be done more quickly and with more accurate results.

Keywords : *Information System, Internal Scholarship, Human Resource Department, JavaServer Pages.*



PEMBUATAN SISTEM INFORMASI PENERIMAAN BEASISWA INTERNAL DIREKTORAT JENDERAL PERBENDAHARAAN DENGAN TEKNOLOGI JAVASERVER PAGES DAN MYSQL

Nama Mahasiswa : Fitria Yulianto
NRP : 5211 105 708
Jurusan : Sistem Informasi FTIf-ITS
Dosen Pembimbing : Dr. Eng. Febriliyan Samopa, S.Kom,
M.Kom

ABSTRAK

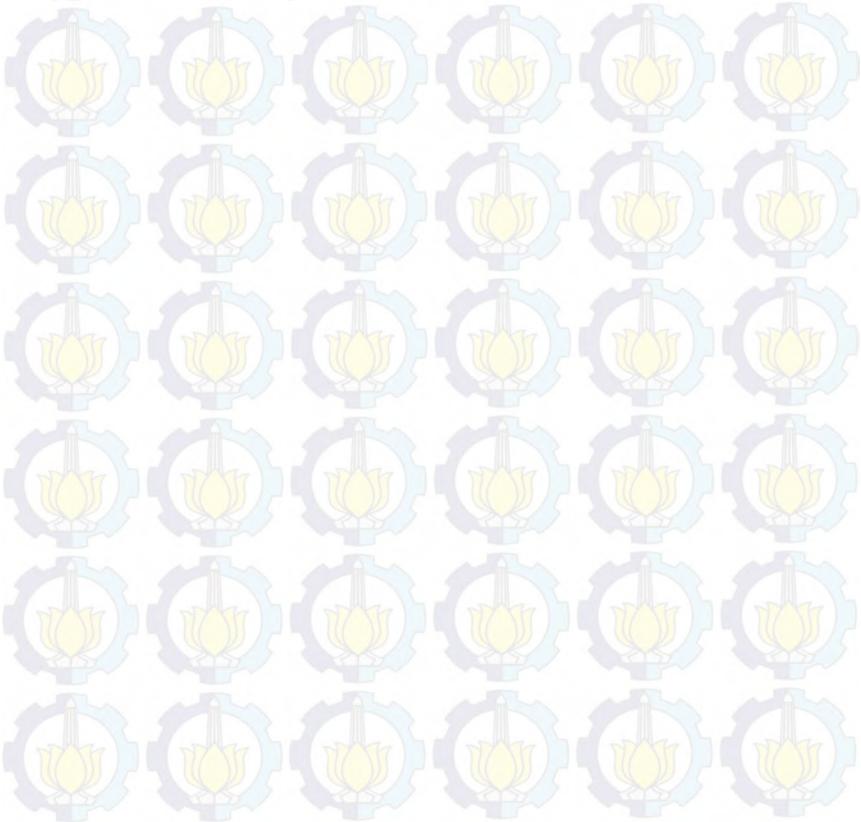
Salah satu fungsi yang dijalankan Sekretariat Ditjen Perbendaharaan (Bagian Pengembangan Pegawai) adalah merencanakan kebutuhan, menyaring calon peserta dan mengevaluasi penyelenggaraan pendidikan dan pelatihan. Dalam rangka menjalankan fungsi tersebut, Bagian Pengembangan Pegawai hampir setiap tahun menyelenggarakan seleksi untuk program beasiswa internal bagi pegawai di lingkungan Direktorat Jenderal Perbendaharaan. Program beasiswa diselenggarakan untuk jenjang Strata 1, Strata 2, maupun Strata 3 dengan kerjasama yang melibatkan beberapa Perguruan Tinggi baik dalam negeri maupun luar negeri.

Untuk mendapatkan beasiswa internal tersebut, pegawai harus melalui 3 tahap seleksi/penyaringan yaitu seleksi administratif, tes tertulis, dan tes wawancara. Selama ini proses seleksi administratif dilakukan secara manual oleh Bagian Pengembangan Pegawai sehingga memerlukan waktu yang cukup lama dan memiliki resiko kesalahan pemeriksaan. Disamping itu Bagian Pengembangan Pegawai juga memerlukan suatu mekanisme pencegahan terjadinya pengusulan beasiswa ganda terhadap satu pegawai. Untuk mengatasi hal tersebut perlu dibuat suatu Sistem Informasi yang bisa membantu proses seleksi

penerimaan beasiswa internal. Melalui tugas akhir ini penulis akan membuat Sistem Informasi Penerimaan Beasiswa Internal (SIPERBEN) dengan teknologi JavaServer Pages dan MySQL.

Dengan adanya aplikasi tersebut diharapkan proses seleksi penerimaan beasiswa internal yang dilakukan oleh Bagian Pengembangan Pegawai dapat dilakukan dengan lebih cepat dan dengan hasil yang lebih akurat.

Kata kunci: *Sistem Informasi, Beasiswa Internal, Pengembangan Pegawai, JavaServer Pages.*



KATA PENGANTAR

Segala puji dan syukur penulis ucapkan pada Allah SWT yang telah memberikan kekuatan bagi penulis sehingga dapat menyelesaikan laporan tugas akhir dengan judul “PEMBUATAN SISTEM INFORMASI PENERIMAAN BEASISWA INTERNAL DIREKTORAT JENDERAL PERBENDAHARAAN DENGAN TEKNOLOGI JAVASERVER PAGES DAN MYSQL”.

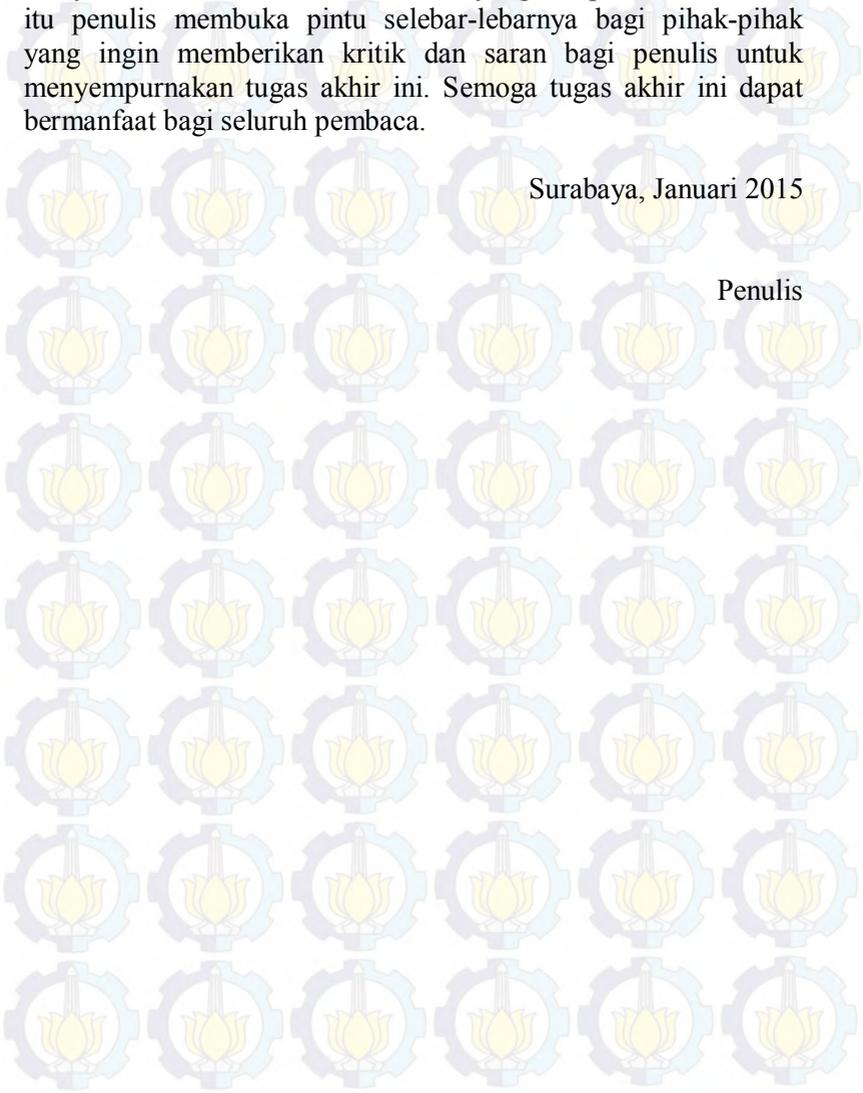
Tugas akhir ini dapat terselesaikan atas dukungan dan bantuan dari berbagai pihak yang telah meluangkan waktu dan pikirannya bagi penulis. Semoga karya tulis ini dapat memberikan manfaat bagi para pembaca dan memberikan kontribusi pada berbagai pihak, khususnya Direktorat Jenderal Perbendaharaan. Pada kesempatan ini penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada :

- Bapak, ibu, serta keluarga yang selalu memberikan dukungan, doa dan kepercayaan kepada penulis.
- Bapak Febriliyan Samopa, selaku dosen pembimbing yang telah meluangkan waktu dan pikiran untuk memberikan bimbingan dan memberi arahan kepada penulis dalam mengerjakan tugas akhir ini.
- Novi Mujilestari dan Safira Putri Kirana, atas segala doa dan dukungan yang telah diberikan selama ini.
- Segenap keluarga seperjuangan Tugas Belajar angkatan 2011.
- Direktorat Jenderal Perbendaharaan yang telah memberikan kesempatan dan kepercayaan untuk menjalankan tugas belajar selama ini.
- Jurusan Sistem Informasi ITS Surabaya yang selalu memberikan dukungan serta pembelajaran.
- Pihak-pihak lain yang telah mendukung adanya tugas akhir ini.

Seperti kata pepatah “tiada gading yang tak retak”, penulis menyadari bahwa tidak ada manusia yang sempurna, oleh sebab itu penulis membuka pintu selebar-lebarnya bagi pihak-pihak yang ingin memberikan kritik dan saran bagi penulis untuk menyempurnakan tugas akhir ini. Semoga tugas akhir ini dapat bermanfaat bagi seluruh pembaca.

Surabaya, Januari 2015

Penulis



DAFTAR ISI

Abstrak.....	iii
Abstract.....	v
Kata Pengantar.....	vii
Daftar Isi.....	ix
Daftar Gambar.....	xvi
Daftar Tabel.....	xviii
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Perumusan Masalah.....	3
1.3. Batasan Masalah.....	3
1.4. Tujuan.....	3
1.5. Manfaat.....	4
1.6. Sistematika Penulisan.....	4
BAB II LANDASAN TEORI.....	7
1.1. Bagian Pengembangan Pegawai.....	7
1.2. Tugas Belajar.....	8
1.3. JavaServer Pages (JSP).....	9
1.4. MySQL.....	10
1.5. CAPTCHA.....	11
BAB III metode penelitian.....	13
3.1. Studi Literatur.....	13
3.2. Identifikasi Kebutuhan.....	15
3.3. Desain Aplikasi.....	15
3.4. Pembuatan Aplikasi.....	15
3.5. Uji Coba Aplikasi.....	15
3.6. Pembuatan Laporan.....	16
BAB IV analisis dan desain sistem.....	17
4.1. Analisis Sistem.....	17

4.1.1. Analisis Proses	18
4.1.2. Analisis Pengguna Sistem.....	21
4.1.3. Analisis Kebutuhan Sistem.....	22
4.2. Desain Sistem	23
4.2.1. Arsitektur Teknologi	23
4.2.2. GUI Storyboard.....	24
4.2.3. Domain Model	40
4.2.4. Use Case	42
4.2.5. Robustness Diagram.....	47
4.2.6. Sequence Diagram.....	47
4.2.7. Class Diagram.....	50
4.2.8. Test Case.....	50
BAB V implementasi dan uji coba	51
5.1. Lingkungan Pengembangan Aplikasi.....	51
5.2. Konfigurasi.....	51
5.2.1. Pembuatan Struktur Direktori	52
5.2.2. Konfigurasi File	54
5.3. Pembuatan Kelas.....	55
5.3.1. Package ‘controller’.....	56
5.3.2. Package ‘entity’.....	57
5.3.3. Package ‘session’	59
5.4. Pembuatan Antarmuka	60
5.5. Pembuatan Fungsionalitas Aplikasi	62
5.6. Uji Coba	67
BAB VI Kesimpulan dan saran.....	69
6.1. Kesimpulan.....	69

6.2. Saran.....	69
Daftar Pustaka	71
7 Lampiran A deskripsi use case	1
A.1. Deskripsi Use Case Login.....	2
A.2. Deskripsi Use Case Logout.....	3
A.3. Deskripsi Use Case Meminta Reset Password.....	3
A.4. Deskripsi Use Case Melihat Pengumuman.....	3
A.5. Deskripsi Use Case Rekam User.....	4
A.6. Deskripsi Use Case Ubah User	4
A.7. Deskripsi Use Case Hapus User.....	5
A.8. Deskripsi Use Case Reset Password User.....	6
A.9. Deskripsi Use Case Rekam Universitas.....	6
A.10. Deskripsi Use Case Ubah Universitas	7
A.11. Deskripsi Use Case Hapus Universitas.....	7
A.12. Deskripsi Use Case Rekam Jurusan	8
A.13. Deskripsi Use Case Ubah Jurusan	8
A.14. Deskripsi Use Case Hapus Jurusan.....	9
A.15. Deskripsi Use Case Membuka Pendaftaran	9
A.16. Deskripsi Use Case Tambah Detail Pembukaan Pendaftaran.....	10
A.17. Deskripsi Use Case Hapus Detail Pembukaan Pendaftaran.....	11
A.18. Deskripsi Use Case Rekam Surat Pembukaan Pendaftaran.....	12
A.19. Deskripsi Use Case Rekam Surat Panggilan Tes Tertulis 13	
A.20. Deskripsi Use Case Rekam Hasil Tes Tertulis.....	14
A.21. Deskripsi Use Case Rekam Surat Panggilan Tes Wawancara	15
A.22. Deskripsi Use Case Rekam Hasil Tes Wawancara.....	16
A.23. Deskripsi Use Case Rekam Surat Pengumuman	16
A.24. Deskripsi Use Case Mendaftar	17
A.25. Deskripsi Use Case Membatalkan Pendaftaran.....	18
A.26. Deskripsi Use Case Melihat Riwayat Pendaftaran	19

A.27. Deskripsi Use Case Rekam Permintaan Rekomendasi..	19
A.28. Deskripsi Use Case Ubah Permintaan Rekomendasi	20
A.29. Deskripsi Use Case Hapus Permintaan Rekomendasi.....	21
A.30. Deskripsi Use Case Melihat Permintaan Rekomendasi.....	22
A.31. Deskripsi Use Case Rekam Rekomendasi	22
A.32. Deskripsi Use Case Ubah Rekomendasi.....	23
A.33. Deskripsi Use Case Hapus Rekomendasi	24
A.34. Deskripsi Use Case Tolak Rekomendasi	24
8 Lampiran B DIAGRAM ROBUSTNESS ANALYSIS	1
B.1. Diagram Login Robustness	2
B.2. Diagram Logout Robustness	2
B.3. Diagram Meminta Reset Password Robustness	3
B.4. Diagram Melihat Pengumuman Robustness	3
B.5. Diagram Rekam User Robustness	4
B.6. Diagram Ubah User Robustness.....	4
B.7. Diagram Hapus User Robustness	5
B.8. Diagram Reset Password User Robustness.....	5
B.9. Diagram Rekam Universitas Robustness	6
B.10. Diagram Ubah Universitas Robustness	6
B.11. Diagram Hapus Universitas Robustness	7
B.12. Diagram Rekam Jurusan Robustness.....	7
B.13. Diagram Ubah Jurusan Robustness	8
B.14. Diagram Hapus Jurusan Robustness.....	8
B.15. Diagram Membuka Pendaftaran Robustness	9
B.16. Diagram Tambah Detail Pembukaan Pendaftaran Robustness.....	9
B.17. Diagram Hapus Detail Pembukaan Pendaftaran Robustness.....	10
B.18. Diagram Rekam Surat Pembukaan Pendaftaran Robustness.....	10
B.19. Diagram Rekam Surat Panggilan Tes Tertulis Robustness	11
B.20. Diagram Rekam Hasil Tes Tertulis Robustness.....	11
B.21. Diagram Rekam Surat Panggilan Tes Wawancara Robustness.....	12

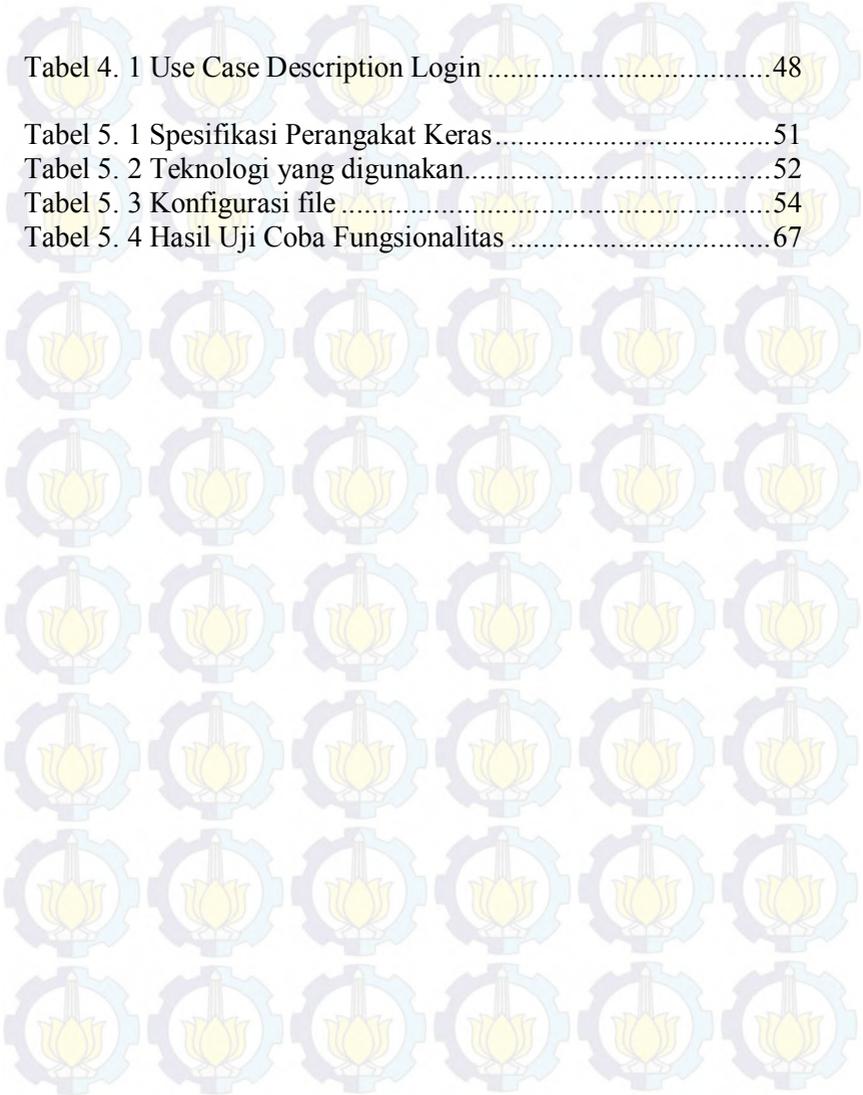
B.22. Diagram Rekam Hasil Tes Wawancara Robustness	12
B.23. Diagram Rekam Surat Pengumuman Robustness.....	13
B.24. Diagram Mendaftar Robustness	13
B.25. Diagram Membatalkan Pendaftaran Robustness	14
B.26. Diagram Melihat Riwayat Pendaftaran Robustness.....	14
B.27. Diagram Rekam Permintaan Rekomendasi Robustness	15
B.28. Diagram Ubah Permintaan Rekomendasi Robustness..	15
B.29. Diagram Hapus Permintaan Rekomendasi Robustness	16
B.30. Diagram Melihat Permintaan Rekomendasi Robustness	16
B.31. Diagram Rekam Rekomendasi Robustness.....	17
B.32. Diagram Ubah Rekomendasi Robustness	17
B.33. Diagram Hapus Rekomendasi Robustness.....	18
B.34. Diagram Tolak Rekomendasi Robustness.....	18
9 Lampiran C Sequence Diagram.....	1
C.1. Diagram Login Sequence.....	2
C.2. Diagram Logout Sequence.....	2
C.3. Diagram Meminta Reset Password Sequence	3
C.4. Diagram Melihat Pengumuman Sequence	3
C.5. Diagram Rekam User Sequence	4
C.6. Diagram Ubah User Sequence.....	4
C.7. Diagram Hapus User Sequence	5
C.8. Diagram Reset Password User Sequence.....	5
C.9. Diagram Rekam Universitas Sequence.....	6
C.10. Diagram Ubah Universitas Sequence	6
C.11. Diagram Hapus Universitas Sequence.....	7
C.12. Diagram Rekam Jurusan Sequence.....	7
C.13. Diagram Ubah Jurusan Sequence	8
C.14. Diagram Hapus Jurusan Sequence.....	8
C.15. Diagram Membuka Pendaftaran Sequence	9
C.16. Diagram Tambah Detail Pembukaan Pendaftaran Sequence.....	9
C.17. Diagram Hapus Detail Pembukaan Pendaftaran Sequence	10

C.18. Diagram Rekam Surat Pembukaan Pendaftaran Sequence	10
C.19. Diagram Rekam Surat Panggilan Tes Tertulis Sequence	11
C.20. Diagram Rekam Hasil Tes Tertulis Sequence.....	11
C.21. Diagram Rekam Surat Panggilan Tes Wawancara Sequence	12
C.22. Diagram Rekam Hasil Tes Wawancara Sequence.....	12
C.23. Diagram Rekam Surat Pengumuman Sequence	13
C.24. Diagram Mendaftar Sequence	13
C.25. Diagram Membatalkan Pendaftaran Sequence.....	14
C.26. Diagram Melihat Riwayat Pendaftaran Sequence	14
C.27. Diagram Rekam Permintaan Rekomendasi Sequence ...	15
C.28. Diagram Ubah Permintaan Rekomendasi Sequence	15
C.29. Diagram Hapus Permintaan Rekomendasi Sequence	16
C.30. Diagram Melihat Permintaan Rekomendasi Sequence..	16
C.31. Diagram Rekam Rekomendasi Sequence	17
C.32. Diagram Ubah Rekomendasi Sequence.....	17
C.33. Diagram Hapus Rekomendasi Sequence	18
C.34. Diagram Tolak Rekomendasi Sequence	18
10 Lampiran D	1
Test Case.....	1
D.1. Test Case Login	2
D.2. Test Case Logout	2
D.3. Test Case Meminta Reset Password.....	3
D.4. Test Case Melihat Pengumuman.....	3
D.5. Test Case Rekam User.....	4
D.6. Test Case Ubah User	4
D.7. Test Case Hapus User.....	5
D.8. Test Case Reset Password User	6
D.9. Test Case Rekam Universitas	6
D.10. Test Case Ubah Universitas.....	7
D.11. Test Case Hapus Universitas.....	8
D.12. Test Case Rekam Jurusan	8

D.13. Test Case Ubah Jurusan.....	9
D.14. Test Case Hapus Jurusan	10
D.15. Test Case Membuka Pendaftaran	10
D.16. Test Case Tambah Detail Pembukaan Pendaftaran	11
D.17. Test Case Hapus Detail Pembukaan Pendaftaran.....	12
D.18. Test Case Rekam Surat Pembukaan Pendaftaran.....	12
D.19. Test Case Rekam Surat Panggilan Tes Tertulis	13
D.20. Test Case Rekam Hasil Tes Tertulis.....	14
D.21. Test Case Rekam Surat Panggilan Tes Wawancara	15
D.22. Test Case Rekam Hasil Tes Wawancara.....	15
D.23. Test Case Rekam Surat Pengumuman	16
D.24. Test Case Mendaftar.....	17
D.25. Test Case Membatalkan Pendaftaran.....	17
D.26. Test Case Melihat Riwayat Pendaftaran	18
D.27. Test Case Rekam Permintaan Rekomendasi	18
D.28. Test Case Ubah Permintaan Rekomendasi	19
D.29. Test Case Hapus Permintaan Rekomendasi	20
D.30. Test Case Melihat Permintaan Rekomendasi.....	20
D.31. Test Case Rekam Rekomendasi	21
D.32. Test Case Ubah Rekomendasi.....	21
D.33. Test Case Hapus Rekomendasi	22
D.34. Test Case Tolak Rekomendasi	23
11 Biodata Penulis	25

DAFTAR TABEL

Tabel 4. 1 Use Case Description Login.....	48
Tabel 5. 1 Spesifikasi Perangkat Keras.....	51
Tabel 5. 2 Teknologi yang digunakan.....	52
Tabel 5. 3 Konfigurasi file	54
Tabel 5. 4 Hasil Uji Coba Fungsionalitas	67



DAFTAR GAMBAR

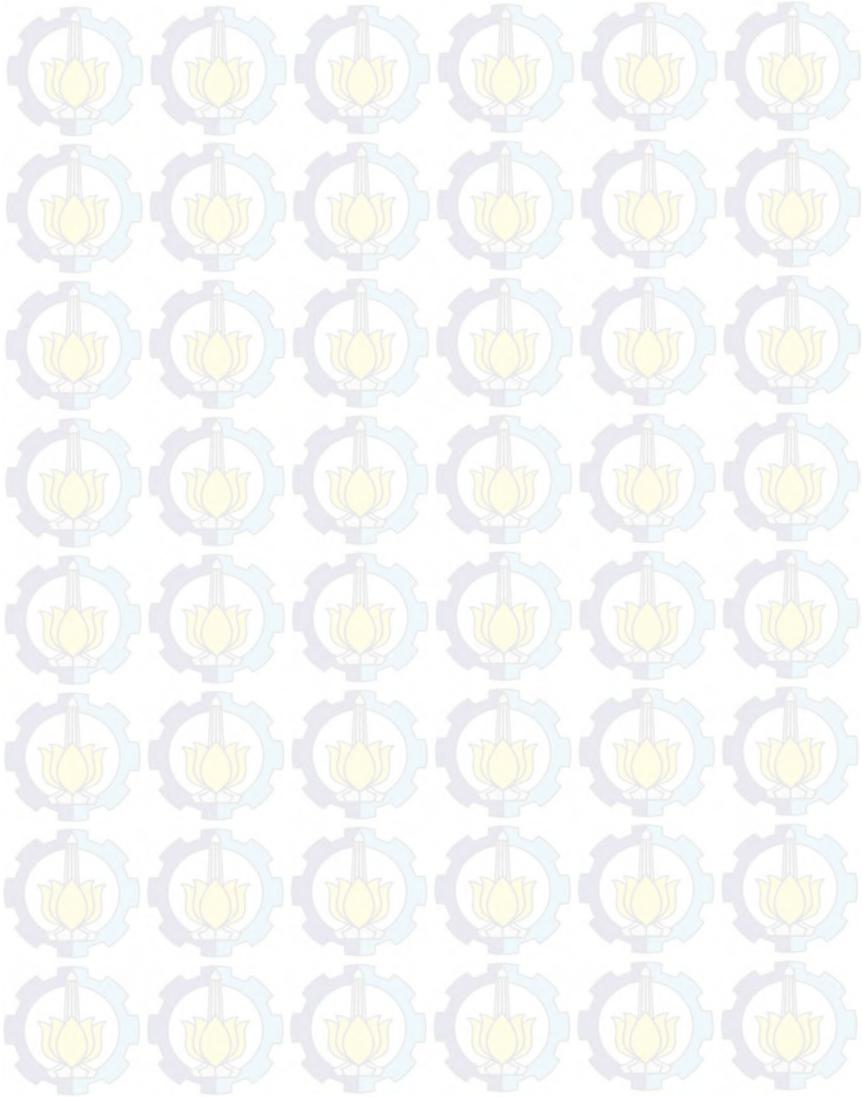
Gambar 2. 1 Struktur Organisasi Setditjen.....	7
Gambar 3. 1 Metode Penelitian.....	14
Gambar 4. 1 ICONIX Process.....	17
Gambar 4. 2 Alur Proses Penerimaan Beasiswa.....	19
Gambar 4. 3 Alur Proses Penerimaan Beasiswa (lanjutan).....	20
Gambar 4. 4 Arsitektur MVC.....	24
Gambar 4. 5 Halaman Utama.....	25
Gambar 4. 6 Form Login.....	26
Gambar 4. 7 Pesan Error Tidak Boleh Kosong.....	26
Gambar 4. 8 Pesan Error CAPTCHA Salah.....	27
Gambar 4. 9 Pesan Error Username dan Password Salah.....	27
Gambar 4. 10 Halaman Utama Admin.....	28
Gambar 4. 11 Halaman Utama Pegawai.....	28
Gambar 4. 12 Halaman Utama Atasan Langsung.....	29
Gambar 4. 13 Halaman User.....	29
Gambar 4. 14 Form Rekam User.....	30
Gambar 4. 15 Konfirmasi Penghapusan.....	30
Gambar 4. 16 Halaman Pendaftaran.....	31
Gambar 4. 17 Halaman Permintaan Rekomendasi.....	31
Gambar 4. 18 Halaman Rekomendasi.....	32
Gambar 4. 19 Form Rekam Rekomendasi.....	32
Gambar 4. 20 Halaman Seleksi Administrasi.....	33
Gambar 4. 21 Form Pembukaan Pendaftaran.....	33
Gambar 4. 22 Detail Pembukaan Pendaftaran.....	34
Gambar 4. 23 List Seleksi Administratif.....	35
Gambar 4. 24 Halaman Tes Tertulis.....	36
Gambar 4. 25 Form Rekam Surat Panggilan Tes.....	36
Gambar 4. 26 Form Rekam Hasil Tes.....	37
Gambar 4. 27 Halaman Tes Wawancara.....	37
Gambar 4. 28 Halaman Universitas.....	38
Gambar 4. 29 Form Rekam Universitas.....	39

Gambar 4. 30 Halaman Jurusan.....	39
Gambar 4. 31 Form Rekam Jurusan.....	40
Gambar 4. 32 Domain Model.....	41
Gambar 4. 33 Aktor	44
Gambar 4. 34 Usecase Diagram Pengguna	45
Gambar 4. 35 Use Case Diagram Admin	45
Gambar 4. 36 Use Case Diagram Pegawai.....	46
Gambar 4. 37 Use Case Diagram Atasan Langsung.....	46
Gambar 4. 38 Robustness Diagram Login	49
Gambar 4. 39 Sequence Diagram Login	49
Gambar 4. 40 Class Diagram.....	50
Gambar 5. 1 Struktur Direktori Aplikasi.....	53
Gambar 5. 2 Konfigurasi File glassfish-resource.xml.....	55
Gambar 5. 3 Konfigurasi File psersistence.xml.....	55
Gambar 5. 4 Konfigurasi File web.xml.....	55
Gambar 5. 5 File dalam Package controller	56
Gambar 5. 6 Kode Program file IndexServlet.java.....	56
Gambar 5. 7 Kode Program AdminServlet.java	57
Gambar 5. 8 class java yang terdapat dalam packages entity.....	58
Gambar 5. 9 Kode Program pada Role.java	58
Gambar 5. 10 Class java yang terdapat dalam Package Session	59
Gambar 5. 11 Kode Program pada AbstractFacade.java.....	60
Gambar 5. 12 Kode Program foreach pada Index.jsp	60
Gambar 5. 13 Tampilan Halaman Utama.....	61
Gambar 5. 14 Kode Program file headerAdmin.jspf	61
Gambar 5. 15 Tampilan yang dihasilkan headerAdmin.jspf.....	62
Gambar 5. 16 Kode Program pada file beasiswa.css	62
Gambar 5. 17 Kode Program proses validasi pengguna	63
Gambar 5. 18 Kode Program proses menampilkan surat.....	64
Gambar 5. 19 Kode Program proses Pembukaan Pendaftaran ...	65
Gambar 5. 20 Kode Program proses Rekam Universitas	66
Gambar 5. 21 Kode Program proses Hapus Universitas	67

DAFTAR PUSTAKA

- Bergsten, H. (2002). *JavaServer Pages*. Sebastopol: O'Reilly & Associates, Inc.
- DuBois, P. (2009). *MySQL*. New Jersey: Pearson Education, Inc.
- Hall, M. (2000). *Core Servlets and JavaServer Pages*. Upper Saddle River: Prentice-Hall, Inc.
- Menteri Keuangan. (2009). *Peraturan Menteri Keuangan Nomor 18/PMK.01/2009 tentang Tugas Belajar di Lingkungan Departemen Keuangan*. Menteri Keuangan.
- Menteri Keuangan. (2010). *Peraturan Menteri Keuangan Nomor 184/PMK.01/2010 tentang Organisasi dan Tata Kerja Kementerian Keuangan*. Menteri Keuangan.
- Myers, G. J. (2004). *The Art of Software Testing*. New Jersey: John Wiley & Sons, Inc.
- Perry, B. W. (2004). *Java Servlet and JSP Cookbook*. Sebastopol: O'Reilly Media, Inc.
- Pressman, R. (2006). *Software Engineering A Practitioner's Approach Sixth Edition*. Mc Graw-Hill Companies.
- Ratzmann, M., & De Young, C. (2003). *Galileo Computing Software Testing and Internationalization*. Salt Lake City: Lemoine International, inc.
- Rosenberg, D., & Stephens, M. (2007). *User Case Driven Object Modelling UML Theory and Practice*. Apress.
- Seshadri, G. (1999). *Server-side Java: Understanding JavaServer Pages Model 2 architecture*. Retrieved from Javaworld: <http://www.javaworld.com/javaworld/jw-12-1999/jw-12-ssj-jspmvc.html>
- Van Haecke, B. (2002). *JDBC 3.0 Java Database Connectivity*. New York: Hungry Minds, Inc.
- Wikipedia.org. (n.d.). *CAPTCHA*. Retrieved December 20, 2014, from [en.wikipedia.org: http://en.wikipedia.org/wiki/CAPTCHA](http://en.wikipedia.org/wiki/CAPTCHA)

Halaman ini sengaja dikosongkan



BIODATA PENULIS



Penulis dilahirkan di Gunung Kidul, pada tanggal 29 Juli 1982. Penulis telah menempuh pendidikan formal di SDN Banyubening I, lalu melanjutkan di SMPN I Karangmojo, dilanjutkan di SMK I Wonosari.

Pada tahun 2000 penulis melanjutkan sekolah ke jenjang yang lebih tinggi di Program Diploma I Sekolah Tinggi Akuntansi Negara (STAN) jurusan Kebendaharaan Negara, setelah lulus pada tahun 2001, penulis ditugaskan di KPPN Biak, Provinsi Papua. Selanjutnya pada tahun 2005 Penulis melanjutkan pendidikan di Program Diploma III dengan kurikulum Khusus pada Sekolah Tinggi Akuntansi Negara (STAN) dan selesai pendidikan pada tahun 2008.

Pada tahun 2011, penulis mengikuti seleksi Program Beasiswa Internal Direktorat Jenderal Perbendaharaan, dan diterima di Jurusan Sistem Informasi, Institut Teknologi Sepuluh Nopember pada tahun 2011 dengan NRP 5211105708. Pengalaman dalam mengikuti seleksi beasiswa inilah yang menginspirasi penulis untuk membangun sistem informasi yang ada pada Tugas Akhir ini.

BAB I

PENDAHULUAN

Pada bab pendahuluan ini akan diuraikan mengenai latar belakang masalah, perumusan masalah, batasan masalah, tujuan tugas akhir, dan manfaat tugas akhir.

1.1.Latar Belakang

Bagian Pengembangan Pegawai adalah salah satu bagian/departemen yang ada pada Sekretariat Ditjen Perbendaharaan. Dalam Peraturan Menteri Keuangan Nomor PMK-184/PMK.01/2010 tentang Organisasi dan Tata Kerja Kementerian Keuangan pasal 836 huruf b disebutkan bahwa salah satu fungsi yang dilaksanakan oleh Bagian Pengembangan Pegawai adalah perencanaan kebutuhan, menyaring peserta, dan mengevaluasi penyelenggaraan pendidikan dan pelatihan (Menteri Keuangan, 2010). Salah satu program yang dijalankan oleh Bagian Pengembangan Pegawai terkait fungsi tersebut adalah program Tugas Belajar.

Tugas Belajar adalah tugas yang diberikan kepada pegawai untuk menuntut ilmu, mendapat pendidikan atau pelatihan keahlian, baik di dalam, maupun di luar negeri, dengan biaya Negara atau dengan biaya oleh instansi pemerintah lainnya (Menteri Keuangan, 2009). Salah satu jenis beasiswa tugas belajar yang dijalankan oleh Bagian Pengembangan Pegawai adalah Beasiswa Internal Direktorat Jenderal Perbendaharaan. Proses seleksi penerimaan beasiswa internal terdiri dari 3 tahapan yaitu :

1. Seleksi Administratif.
2. Test Tertulis.
3. Test Wawancara.

Proses seleksi administratif adalah seleksi awal terhadap usulan pegawai yang berminat mendapatkan beasiswa internal dengan cara memeriksa berkas dan data pegawai tersebut apakah

memenuhi syarat yang telah ditentukan. Seleksi Administratif dilakukan oleh Bagian Pengembangan Pegawai secara manual sehingga membutuhkan waktu yang cukup lama dan memiliki resiko kesalahan pemeriksaan. Untuk memudahkan bagian pengembangan dalam melakukan seleksi administratif tersebut diperlukan suatu sistem informasi yang memungkinkan otomatisasi proses seleksi administratif.

Salah satu syarat untuk mengikuti seleksi beasiswa yang disebutkan dalam pasal 5 ayat 1 huruf g peraturan menteri keuangan nomor 18 tahun 2009 adalah tidak sedang dicalonkan dalam program beasiswa lainnya (Menteri Keuangan, 2009). Untuk menghindari adanya pengusulan ganda, maka Bagian pengembangan Pegawai juga memerlukan suatu mekanisme untuk mencegah terjadinya usulan ganda terhadap peserta seleksi beasiswa.

Saat ini sistem informasi banyak yang dikembangkan dengan teknologi berbasis web salah satu teknologi yang bisa digunakan adalah JavaServer Pages dan database MySQL. JavaServer Pages (JSP) adalah sebuah teknologi untuk mengembangkan halaman web yang berisi konten dinamis (Bergsten, 2002). Sedangkan MySQL adalah sebuah database manajemen system yang mencakup sebuah SQL Server, client program untuk mengakses server, administrative tools, dan programming interface untuk menulis program anda sendiri (DuBois, 2009).

Alasan penulis memilih JavaServerPages dan MySQL untuk mengembangkan sistem informasi pada tugas akhir ini adalah sebagai berikut :

1. JavaServer Pages mampu menciptakan web content yang bersifat dinamis dengan bahasa pemrograman java dengan komunitas pengguna yang cukup banyak.
2. MySQL merupakan sistem manajemen basis data relasional yang populer digunakan untuk aplikasi berbasis web.
3. JavaServer Pages dan MySQL dapat digunakan dengan lisensi *open source*.

1.2. Perumusan Masalah

Permasalahan yang ingin diselesaikan dalam tugas akhir ini adalah bagaimana membuat Sistem Informasi Penerimaan Beasiswa Internal Direktorat Jenderal Perbendaharaan (SIPERBEN) dengan teknologi JavaServer Pages dan MySQL.

1.3. Batasan Masalah

Dari permasalahan yang telah disebutkan diatas, yang menjadi batasan masalah dalam tugas akhir ini adalah sebagai berikut:

1. Data kepegawaian yang digunakan pada aplikasi dalam tugas akhir ini bersifat data *dummy* mengingat data riil kepegawaian yang bersifat rahasia.
2. Proses seleksi penerimaan beasiswa yang dilakukan dalam sistem adalah proses seleksi administratif.
3. Aplikasi dalam Tugas Akhir ini tidak mencakup proses tes tertulis maupun tes wawancara, namun disediakan menu untuk menginput hasil dari tes dimaksud.
4. Pemberian Nomor Surat pada Aplikasi dalam Tugas Akhir ini dilakukan secara manual karena tidak terintegrasi dengan Aplikasi Persuratan.
5. SIPERBEN tidak mencakup proses Administrasi kemahasiswaan selama perkuliahan maupun proses penyaluran pegawai pasca tugas belajar.

1.4. Tujuan

Tujuan dari penulisan tugas akhir ini adalah menghasilkan Sistem Informasi Penerimaan Beasiswa Internal Direktorat Jenderal Perbendaharaan.

1.5. Manfaat

Manfaat yang dapat diberikan dari tugas akhir ini adalah sebagai berikut:

1. Bagi organisasi (Bagian Pengembangan Pegawai)
 - a. Aplikasi dalam tugas akhir ini dapat mempercepat proses seleksi administratif pegawai yang ingin mengikuti beasiswa internal.
 - b. Aplikasi dalam tugas akhir ini dapat membantu meningkatkan akurasi proses seleksi administratif yang sebelumnya dilakukan dengan cara manual.
 - c. Membantu proses penelusuran seleksi Beasiswa.
2. Bagi iptek
Tugas akhir dapat menjadi referensi bagi pengembangan aplikasi yang sama di masa yang akan datang.
3. Bagi penulis
 - a. Sebagai sarana pembelajaran mengenai bagaimana proses membangun aplikasi berbasis web.
 - b. Menerapkan ilmu pengetahuan yang diperoleh selama perkuliahan untuk menghasilkan produk yang bermanfaat pada bidang pekerjaan yang dijalankan oleh penulis.

1.6. Sistematika Penulisan

Sistematika penulisan buku tugas akhir dibagi menjadi 6 bab sebagai berikut.

BAB I PENDAHULUAN

Bab ini berisi latar belakang masalah, rumusan permasalahan, batasan masalah/ruang lingkup, tujuan, manfaat dan sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini menjelaskan teori-teori dari berbagai pustaka yang digunakan sebagai acuan dalam penulisan buku tugas akhir ini.

BAB III METODE PENELITIAN

Bab ini membahas alur dan pengerjaan tugas akhir dari awal sampai selesainya tugas akhir.

BAB IV ANALISIS DAN DESAIN SISTEM

Bab ini menjelaskan rancangan desain aplikasi yang dibuat berdasarkan kebutuhan sistem. desain tersebut digunakan untuk pembangunan aplikasi pada tugas akhir ini.

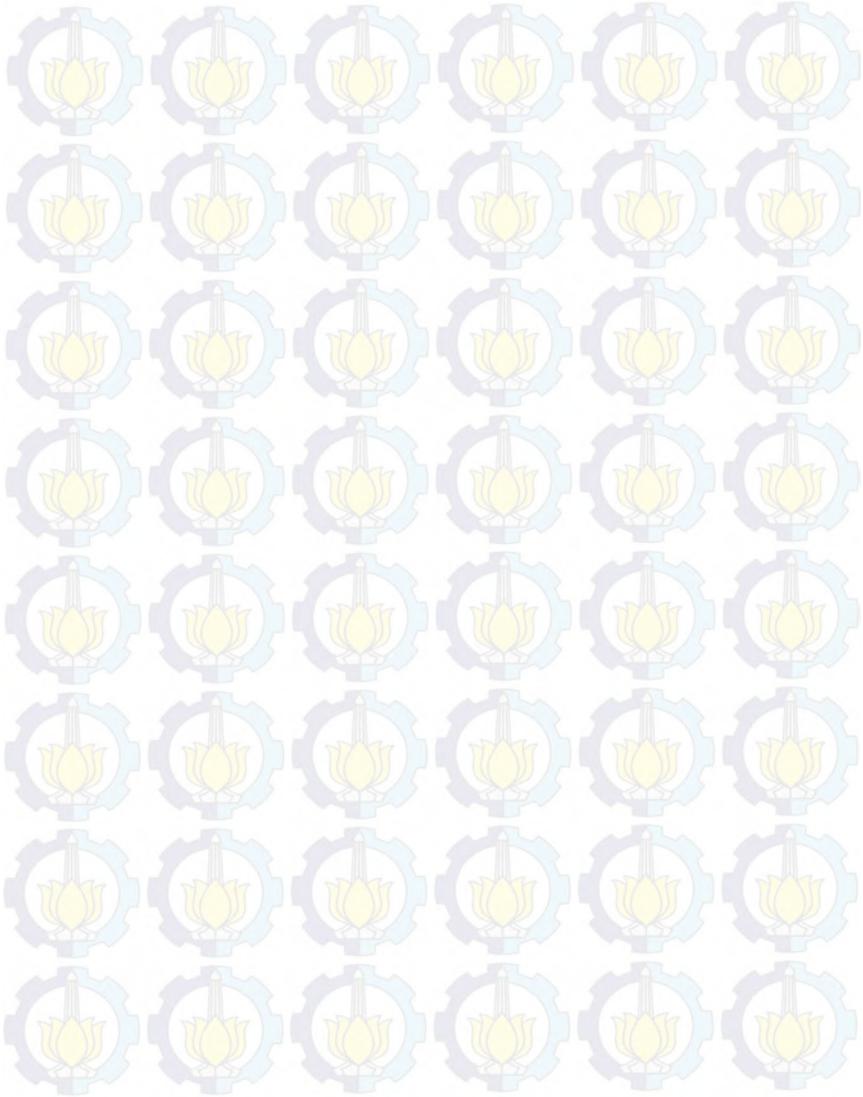
BAB V PEMBUATAN APLIKASI DAN UJI COBA

Bab ini menjelaskan pembangunan aplikasi yang sesuai dengan desain. selain itu, dijelaskan pula uji coba sistem dalam menjaga performa aplikasi.

BAB VI KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan yang dapat diambil dari tugas akhir ini dan saran untuk kelanjutan sistem.

Halaman ini sengaja dikosongkan.



BAB II

LANDASAN TEORI

Pada bab ini dijelaskan tentang teori-teori dari berbagai literatur yang berhubungan dengan permasalahan yang dibahas dalam tugas akhir ini.

1.1. Bagian Pengembangan Pegawai

Bagian Pengembangan Pegawai merupakan salah satu bagian/departemen yang ada pada kantor Sekretariat Ditjen Perbendaharaan. Struktur organisasi dari Sekretariat Ditjen Perbendaharaan dapat dilihat pada Gambar 2.1.



Gambar 2.1 Struktur Organisasi Setditjen

Tugas dari Bagian Pengembangan Pegawai adalah melaksanakan analisis dan evaluasi kebutuhan metode pengembangan pegawai, pendayagunaan kompetensi pegawai, penyaringan pegawai dalam rangka pendidikan dan pelatihan, ujian jabatan, dan mempersiapkan usul penyempurnaan kurikulum pendidikan dan pelatihan pegawai (Menteri Keuangan, 2010). Dalam melaksanakan tugas tersebut Bagian Pengembangan Pegawai menyelenggarakan fungsi sebagai berikut :

1. pelaksanaan identifikasi, evaluasi, pengembangan, dan pendayagunaan kompetensi pegawai;
2. perencanaan kebutuhan, menyaring calon peserta, dan mengevaluasi penyelenggaraan pendidikan dan pelatihan;
3. penyiapan usul penyempurnaan kurikulum pendidikan dan pelatihan pegawai;
4. pengelolaan basis data kompetensi, media informasi dan edukasi, sarana riset, dan sarana pengembangan lainnya.

1.2. Tugas Belajar

Tujuan dari pemberian Tugas Belajar adalah untuk memberikan kesempatan bagi pegawai untuk meningkatkan kompetensi pribadi, yang dapat menunjang tugas dan fungsi Departemen Keuangan sesuai dengan kebutuhan kedinasan dan dapat dipertanggungjawabkan (Menteri Keuangan, 2009). Dalam lingkup Direktorat Jenderal Perbendaharaan, terdapat 2 jenis Tugas Belajar (Beasiswa) yang diselenggarakan yaitu Beasiswa Internal (diselenggarakan oleh DJPB) dan Beasiswa Eksternal (Diselenggarakan lembaga lain seperti ADS, PPSDM, dan Bappenas). Sedangkan apabila dilihat dari jenjang pendidikan, beasiswa yang diselenggarakan terdiri dari 3 jenjang yaitu Strata I (S1), Strata II (S2), dan Strata III (S3).

Persyaratan untuk menjadi calon peserta Beasiswa Internal Direktorat Jenderal Perbendaharaan didasarkan pada Peraturan Menteri Keuangan Nomor 84/PMK.01/2009, persyaratan tersebut antara lain :

Persyaratan Umum :

1. Telah berstatus sebagai Pegawai Negeri Sipil.
2. Masa kerja minimal 2 tahun sejak selesai mengikuti program Tugas Belajar sebelumnya.
3. Tidak sedang dicalonkan dalam program beasiswa lainnya.
4. Memiliki DP3 sekurang-kurangnya bernilai baik dalam 1 tahun terakhir.
5. Tidak sedang menjalani hukuman disiplin.

6. Tidak sedang dalam proses pindah ke luar Direktorat Jenderal Perbendaharaan

Syarat tambahan program S1 :

1. Usia tidak lebih dari 32 tahun.
2. Pangkat/golongan Pengatur (II/c) dengan masa kerja golongan minimal 2 tahun.
3. Memiliki ijazah DIII.

Syarat tambahan program S2 :

1. Usia tidak lebih dari 40 tahun.
2. Pangkat/golongan Penata Muda (III/a) dengan masa kerja golongan minimal 2 tahun
3. Memiliki ijazah DIV atau S1 dengan nilai IPK minimal 2,75.

Syarat tambahan program S3 :

1. Usia tidak lebih dari 42 tahun.
2. Pangkat/golongan Penata Muda Tingkat 1 (III/b) dengan masa kerja golongan minimal 2 tahun
3. Memiliki ijazah S2 dengan nilai IPK minimal 2,75.

1.3. JavaServer Pages (JSP)

JavaServer Pages (JSP) adalah teknologi untuk mengembangkan halaman web dengan memasukkan konten dinamis (Bergsten, 2002). Sedangkan pengertian JSP menurut Marty Hall (Hall, 2000) adalah suatu halaman yang mampu menggabungkan halaman web yang regular, statik dan dinamis yang dihasilkan dari Servlet.

Servlet dan JSP memungkinkan pengembangan aplikasi dengan menggunakan arsitektur *client/server* Model 2 yang memisahkan antara konten dengan logika. Model 2 terdiri dari tiga proses yang terpisah (Seshadri, 1999), yaitu :

1. *Browser*, berada pada sisi *client* yang menampilkan konten dinamis.
2. *Application Server* (Servlet dan JSP).
3. *Database*

Menurut Bergsten (Bergsten, 2002) keunggulan dari penggunaan teknologi JSP adalah:

1. JSP dibangun dari java servlet API seperti JDBC, Remote Method Invocation (RMI) and OMG CORBA support, JNDI (Java Naming and Directory Interface), Enterprise JavaBeans (EJB), JMS (Java Message Service), JTA (Java Transaction API), JAXP (Java API for XML Processing), dan JavaMail
2. JSP mendukung konten dinamis berbasis elemen yang dapat disisipkan secara langsung pada halaman HTML.
3. Halaman JSP dicompile untuk pemrosesan server secara efisien.
4. JSP digunakan untuk menyajikan *user interface* dan dapat dikombinasikan dengan teknologi J2EE lain seperti Java Servlets untuk menangani *input processing*, dan Enterprise JavaBeans untuk menangani *application logic*.

1.4.MySQL

MySQL adalah sebuah SQL client/server relational database management system (RDBMS) yang berasal dari Skandinavia, dimana MySQL merupakan database manajemen system yang mencakup sebuah SQL Server, client program untuk mengakses server, administrative tools, dan programming interface untuk menulis program anda sendiri (DuBois, 2009).

MySQL didistribusikan secara gratis dengan lisensi *General Public License* (GPL). Beberapa keunggulan MySQL dibandingkan dengan sistem manajemen database lainnya antara lain:

1. *Speed*. MySQL adalah sistem database yang cepat;
2. *Ease of use*. MySQL memiliki performa yang tinggi namun relatif sistem database yang simple;

3. *Query language support.* MySQL mendukung SQL (*Structure Query Language*), bahasa standar yang dipilih oleh seluruh sistem database modern;
4. *Capability.* Server MySQL bersifat *multi-threaded*, sehingga banyak *client* dapat terkoneksi pada waktu yang bersamaan;
5. *Connectivity and security.* MySQL terkoneksi dengan jaringan sehingga dapat diakses darimanapun melalui internet. MySQL mendukung koneksi terenkripsi dengan *Secure Sockets Layer* (SSL) protokol untuk keamanan akses database.
6. *Portability.* MySQL dapat berjalan diberbagai macam sistem operasi misalnya Unix dan Linux. MySQL dapat berjalan dalam perangkat keras dari yang *high-end* sampai komputer pribadi;
7. *Small size.* MySQL memiliki ukuran distribusi yang tidak terlalu besar;
8. *Availability and cost.* MySQL adalah sebuah sistem database yang tersedia dalam lisensi *Open Source* dan lisensi komersil;
9. *Open distribution and source code.* MySQL sangat mudah didapatkan bahkan sampai mendapatkan *source code* untuk kepentingan pengembangan.

1.5.CAPTCHA

CAPTCHA merupakan singkatan dari Completely Automated Public Turing test to tell Computers and Humans Apart. Captcha adalah sebuah tes yang berupa challenge-response (pertanyaan-jawaban) yang digunakan untuk menentukan apakah user/pengguna adalah seorang manusia (Wikipedia.org). istilah captcha ditemukan oleh Luis von Ahn, Manuel Blum, Nicholas J. Hopper, dan John Langford pada tahun 2000.

Secara umum captcha digunakan untuk mencegah *bots* atau program otomatis mengakses informasi yang bersifat sensitif/rahasia. Untuk melakukan hal tersebut, captcha didesain agar membutuhkan 3 kemampuan yang hanya dimiliki oleh manusia untuk memecahkannya. Kemampuan tersebut adalah :

1. *Invariant Recognition.*

Invariant recognition adalah kemampuan untuk mengenali huruf/angka dalam berbagai variasi bentuk.

2. *Segmentation.*

Segmentation adalah kemampuan untuk memisahkan antara huruf satu dengan yang lain walaupun ditampilkan secara *crowded* (bertumpuk).

3. *Context*

Context adalah kemampuan untuk mengidentifikasi setiap huruf berdasarkan konteks kata yang ditampilkan dalam captcha.

BAB III

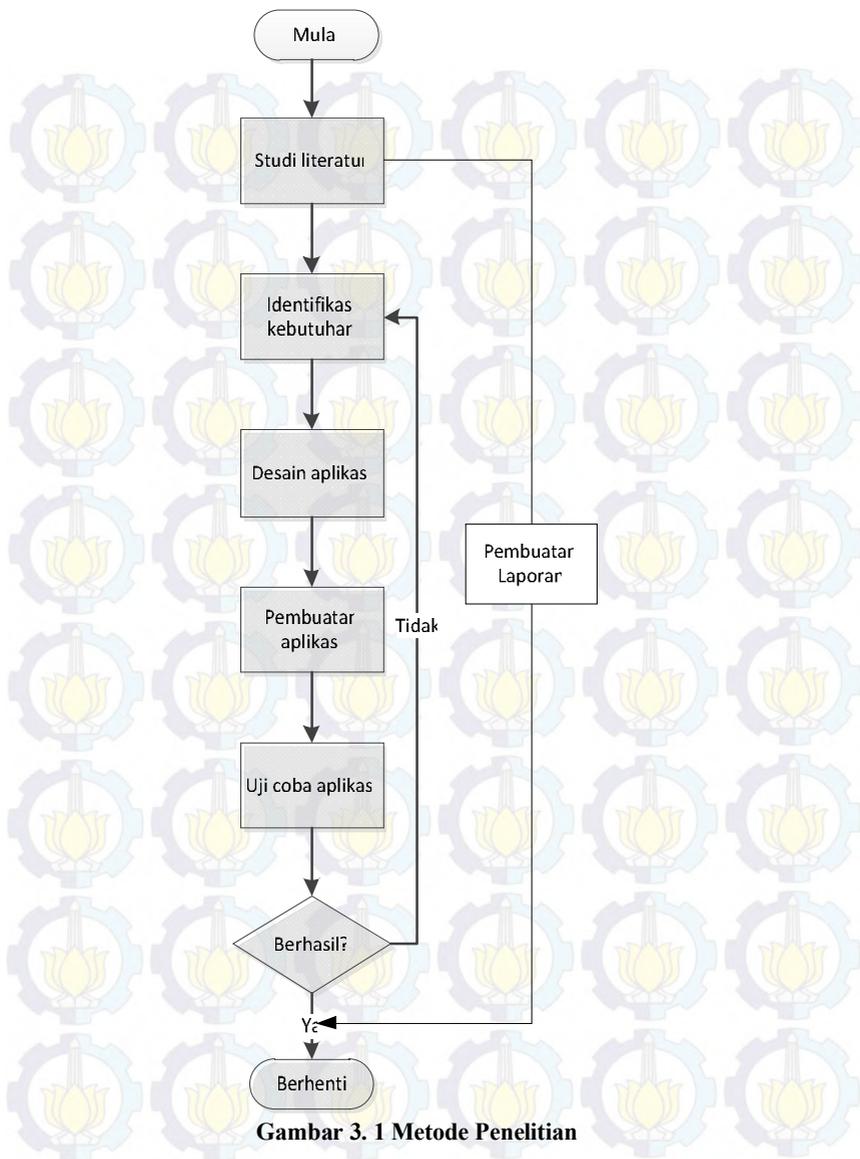
METODE PENELITIAN

Pada bagian ini akan dibahas mengenai langkah-langkah kegiatan penelitian yang akan dilakukan sebagaimana digambarkan pada gambar 3.1. Metode Penelitian dibagi ke dalam beberapa tahap, yaitu studi literatur, identifikasi kebutuhan, desain aplikasi, pembuatan aplikasi, dan ujicoba aplikasi. Dokumentasi dan laporan dibuat selama tahapan tersebut dilakukan.

3.1.Studi Literatur

Studi literatur dilakukan dengan mengkaji literatur yang berkaitan dengan Sistem Informasi yang dibuat pada tugas akhir ini. Literatur bisa berupa buku, jurnal akademis, serta kumpulan peraturan perundangan. Literatur yang menjadi fokus pencarian dalam tugas akhir ini antara lain :

- Tugas Belajar
Studi pustaka mengenai pengertian umum, persyaratan, serta tahapan proses seleksi tugas belajar (beasiswa).
- Java
Studi pustaka mengenai pengertian umum JSP kegunaan serta bagaimana pengaplikasiannya dalam pembuatan aplikasi yang berbasis web.
- MySQL
Studi pustaka mengenai pengertian umum, kelebihan serta bagaimana pengaplikasian MySQL dalam pembuatan aplikasi yang dibuat pada tugas akhir ini.



Gambar 3. 1 Metode Penelitian

3.2. Identifikasi Kebutuhan

Identifikasi kebutuhan yang disebut juga *Requirement Engineering* menyediakan mekanisme untuk mengetahui apa yang diinginkan *customer*, menganalisa kebutuhan, menilai *feasibility*, dan menegosiasikan kemungkinan solusi (Pressman, 2006). Untuk melakukan identifikasi kebutuhan Sistem Informasi dalam tugas akhir ini, beberapa cara yang bisa dilakukan adalah mengkaji peraturan (terkait syarat-syarat penerimaan beasiswa), serta wawancara/interview terhadap pihak yang berkepentingan.

3.3. Desain Aplikasi

Menurut Roger S. Pressman (Pressman, 2006) tujuan dari pembuatan desain adalah untuk menghasilkan model atau representation dari sistem untuk mengetahui *firmness*, *commodity*, dan *delight*. Perancangan dan desain sistem informasi yang ada pada tugas akhir ini dilakukan dengan menggunakan Unified Modeling Language (UML). UML adalah bahasa standar yang digunakan untuk proses dokumentasi, pembuatan spesifikasi dan membangun sebuah sistem perangkat lunak.

3.4. Pembuatan Aplikasi

Setelah proses pembuatan desain selesai, dilanjutkan dengan pembuatan aplikasi berdasarkan desain yang telah dibuat tersebut. Aplikasi dikembangkan dengan menggunakan *platform* JavaServer Pages dan MySQL sebagai basis data.

3.5. Uji Coba Aplikasi

Pada tahap ini akan dilakukan uji coba terhadap aplikasi yang telah dibangun. Uji coba dilakukan untuk melihat kesesuaian antara fungsionalitas sistem informasi yang telah dibuat dengan

kebutuhan aplikasi dan desain aplikasi. Uji coba dilakukan juga untuk mengetahui apakah terdapat kesalahan dalam informasi yang ditampilkan maupun kesalahan dalam penulisan kode program.

Menurut Ratzmann (Ratzmann, et al., 2003) secara tradisional uji fungsional disebut juga sebagai *black box test* dan uji struktural disebut juga sebagai *white box test*. Menurut Myers (Myers, 2004) *black box testing* atau *data-driven testing* atau *input/output testing* adalah pendekatan dalam melakukan pengujian dengan pertimbangan bahwa program merupakan suatu entitas yang komplet dengan tanpa mempertimbangkan struktur internal program. Sementara itu *white box testing* atau *logic-driven testing* adalah pengujian terhadap struktur internal program.

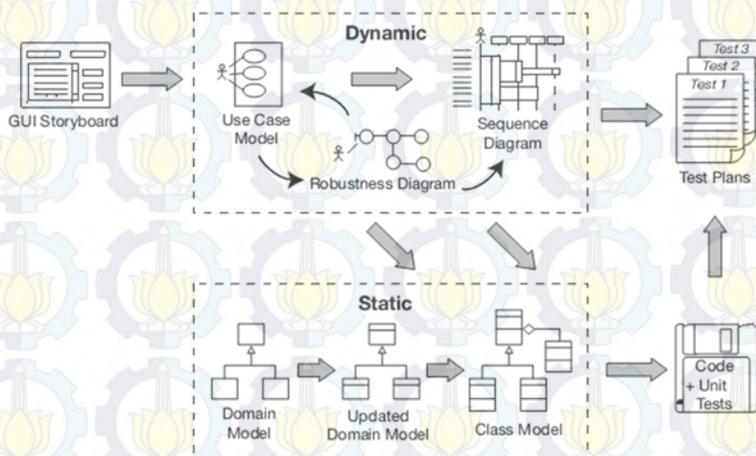
3.6.Pembuatan Laporan

Pembuatan laporan dilakukan sepanjang tahapan pembuatan aplikasi. Hal ini juga dilakukan dalam rangka dokumentasi aplikasi guna memudahkan pemahaman terhadap aplikasi maupun pengembangan aplikasi di masa yang akan datang. Laporan dikerjakan menggunakan format pengerjaan tugas akhir sehingga menghasilkan sebuah buku tugas akhir

BAB IV

ANALISIS DAN DESAIN SISTEM

Pada bab ini akan dijelaskan rancangan sistem dari Sistem Informasi Penerimaan Beasiswa Internal. Perancangan aplikasi ini dilakukan dengan mengacu pada tahapan dalam ICONIX process. Tahapan yang ada pada ICONIX process dapat dilihat pada gambar 4.1. Disamping perancangan berdasarkan ICONIX process, pada bab ini juga akan dibahas mengenai analisis dari Sistem Informasi Penerimaan Beasiswa Internal.



Gambar 4. 1 ICONIX Process

4.1. Analisis Sistem

Analisis diperlukan untuk mengetahui karakteristik dari sistem yang akan dikembangkan. Analisis ini dilakukan dengan melakukan pengamatan serta wawancara dengan narasumber yang memahami tatacara penerimaan beasiswa internal.

4.1.1. Analisis Proses

Proses seleksi penerimaan beasiswa internal terdiri dari tiga tahapan yaitu :

1. Seleksi Administrasi

Seleksi administrasi dilakukan oleh bagian pengembangan dengan cara menyeleksi berkas pendaftaran yang dikirimkan oleh pegawai yang berminat untuk mengikuti beasiswa internal, serta arsip/ dosir pegawai terkait yang disimpan oleh bagian pengembangan pegawai. Seleksi administrasi didasarkan pada Peraturan Menteri Keuangan Nomor 84/PMK.01/2009, yang telah dijelaskan pada Bab II Tugas Akhir ini.

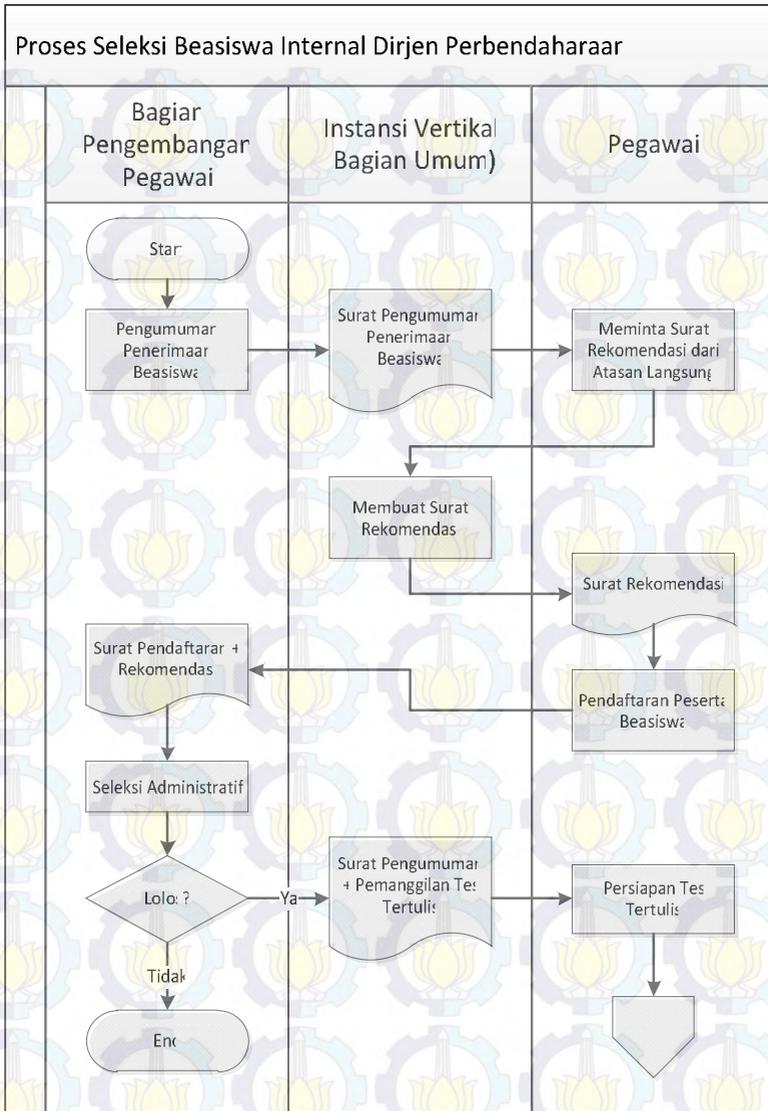
2. Tes Tertulis

Bagi pegawai yang lolos seleksi administrasi kemudian dipanggil untuk mengikuti tes tertulis. Tes tertulis diselenggarakan oleh bagian pengembangan di beberapa kota besar seperti Medan, Jakarta, Surabaya, Makasar, dan Jayapura. Peserta seleksi bebas memilih lokasi tes tertulis, mengingat semua biaya perjalanan serta akomodasi peserta selama tes tertulis ditanggung sendiri oleh masing-masing pegawai.

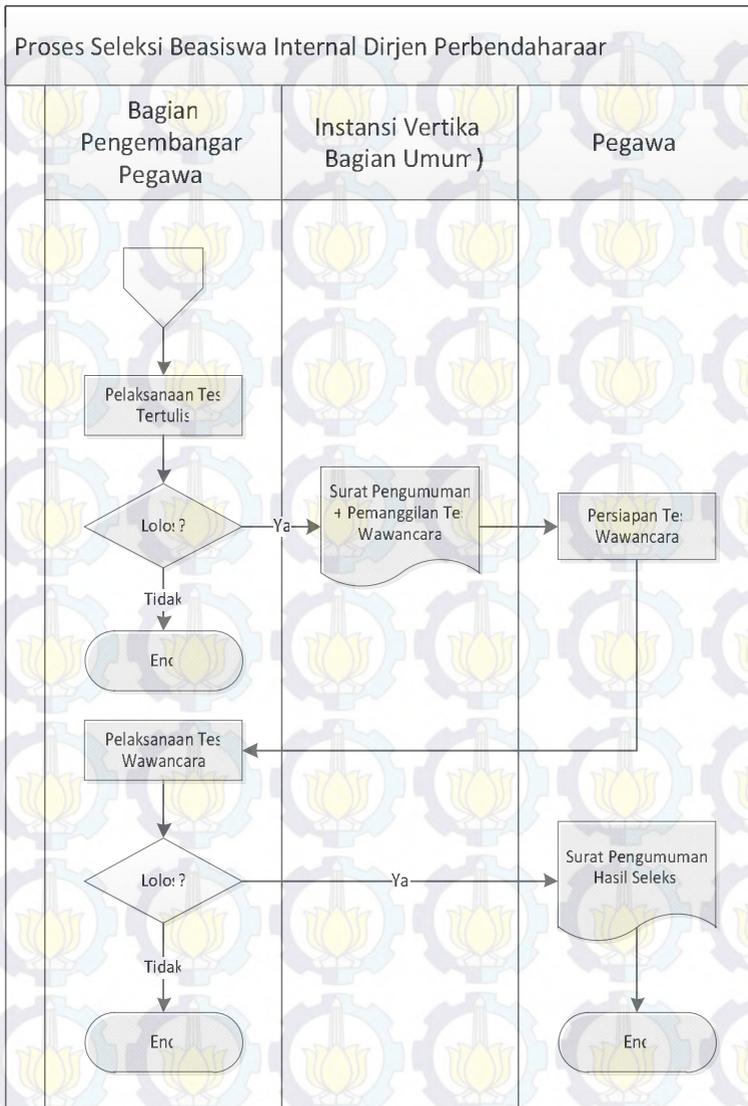
3. Tes Wawancara

Bagi pegawai yang lolos tes tertulis kemudian dipanggil untuk mengikuti tes wawancara. Tes wawancara biasanya berlokasi di beberapa universitas yang menjadi mitra kerjasama beasiswa internal Direktorat Jenderal Perbendaharaan seperti Universitas Indonesia, Institut Teknologi Sepuluh Nopember, dan Universitas Diponegoro.

Adapun alur proses seleksi penerimaan beasiswa dapat dilihat pada gambar 4.2 dan gambar 4.3.



Gambar 4. 2 Alur Proses Penerimaan Beasiswa



Gambar 4. 3 Alur Proses Penerimaan Beasiswa (lanjutan)

4.1.2. Analisis Pengguna Sistem

Setelah melakukan pengamatan serta penggalian informasi terkait proses penerimaan beasiswa internal diketahui bahwa pengguna dari sistem ini adalah sebagai berikut :

1. Administrator

Administrator (Admin) merupakan petugas dari Bagian Pengembangan Pegawai. Administrator bertugas untuk mengelola data pengguna sistem, menata usahakan proses seleksi penerimaan beasiswa, membuat surat panggilan tes dan surat pengumuman hasil tes, serta mengelola data referensi.

2. Pegawai

Pegawai merupakan pengguna dari sistem yang berminat untuk mengikuti seleksi penerimaan beasiswa internal. Pengguna ini membutuhkan sistem untuk melakukan pendaftaran sebagai peserta seleksi penerimaan beasiswa internal.

3. Atasan Langsung

Atasan Langsung adalah pegawai yang memiliki jabatan sebagai atasan langsung dari pegawai yang akan mendaftar sebagai peserta seleksi penerimaan beasiswa internal. Atasan langsung bertanggung jawab untuk memberikan rekomendasi atas permintaan pegawai yang menjadi bawahannya.

Dari ketiga pengguna tersebut terdapat beberapa hak dan kewenangan yang sama seperti login, logout, dan melihat pengumuman, sehingga dapat kita tentukan bahwa dalam sistem informasi penerimaan beasiswa internal terdapat 4 aktor yaitu Administrator (Admin), Pegawai, Atasan Langsung, dan Pengguna (yang merupakan generalisasi dari ketiga aktor tersebut)

4.1.3. Analisis Kebutuhan Sistem

Berdasarkan analisa proses serta pengguna diatas, maka cakupan kebutuhan utama yang terdapat dalam sistem informasi penerimaan beasiswa internal antara lain :

1. Penayangan data surat
2. Pengelolaan pengguna sistem
3. Pengelolaan proses pendaftaran
4. Pengelolaan proses seleksi administratif
5. Pengelolaan proses tes tertulis
6. Pengelolaan proses tes wawancara
7. Pengelolaan data referensi

Dari kebutuhan utama tersebut kemudian dirinci lebih lanjut sehingga terbentuk kebutuhan fungsional dan kebutuhan non fungsional dari sistem informasi penerimaan beasiswa internal.

4.1.3.1. Kebutuhan Fungsional

Kebutuhan Fungsional merupakan fungsionalitas sistem yang harus ada dalam aplikasi karena merupakan kegunaan utama aplikasi sebagai alat bantu dalam proses penerimaan beasiswa internal. Kebutuhan Fungsional dari sistem ini antara lain :

1. Proses otentifikasi pengguna sistem
2. Proses Rekam/Ubah/Hapus (RUH) data pengguna
3. Proses RUH data universitas
4. Proses RUH data jurusan
5. Proses perekaman surat
6. Proses perekaman hasil seleksi
7. Proses penayangan surat
8. Proses pembukaan pendaftaran
9. Proses pendaftaran
10. Proses pembatalan pendaftaran
11. Proses penayangan riwayat pendaftaran
12. Proses permintaan rekomendasi
13. Proses penayangan permintaan rekomendasi

14. Proses RUH dan tolak rekomendasi

4.1.3.2. Kebutuhan Non Fungsional

Kebutuhan Non Fungsional dalam sistem informasi penerimaan beasiswa internal adalah sebagai berikut :

1. Keamanan
Sistem hanya dapat diakses oleh pengguna sesuai dengan perannya.
2. Kompatibilitas *Browser*
Sistem dapat dijalankan di berbagai *browser*.

4.2. Desain Sistem

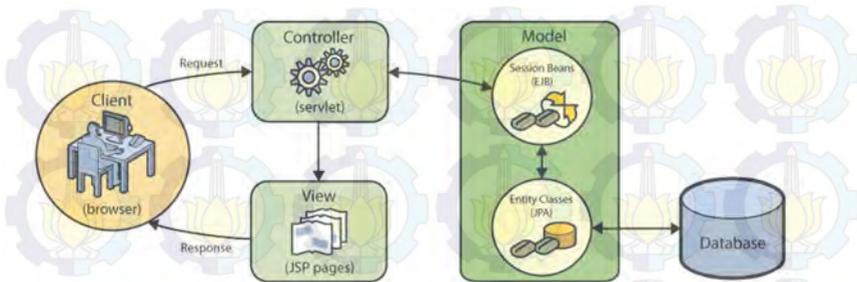
Setelah melakukan analisa baik terhadap alur proses, pengguna, maupun kebutuhan sistem, penulis kemudian melakukan proses desain sistem sesuai dengan ICONIX proses.

4.2.1. Arsitektur Teknologi

Arsitektur teknologi yang digunakan dalam pembuatan SIPERBENI adalah *Model, View, Controller* (MVC), dimana aplikasi dibagi kedalam 3 komponen yaitu :

1. Model
Model merupakan representasi dari *business logic* yang mengatur akses dan perubahan data.
2. View
View mengatur bagaimana data ditampilkan. View mengambil data dari model kemudian menampilkannya, view juga meneruskan input dari user kepada controller
3. Controller
Controller menentukan bagaimana *behaviour* dari sistem. Controller menginterpretasikan input dari user dan memetakan pada tindakan yang harus dilakukan dalam Model.

Arsitektur MVC dapat dilihat pada gambar 4.4.



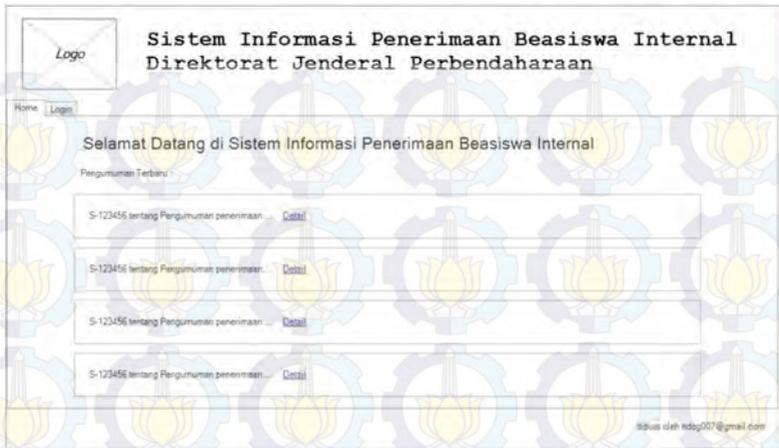
Gambar 4. 4 Arsitektur MVC

4.2.2.GUI Storyboard

GUI Storyboard merupakan gambaran awal dari aplikasi yang akan dibuat. Untuk memudahkan pemahaman, GUI Storyboard dikelompokkan berdasarkan alur proses serta kebutuhan utama yang telah disebutkan diatas.

4.2.2.1.Penayangan Data Surat

Pada saat sistem diakses, yang pertama kali ditampilkan adalah data surat terbaru terkait proses penerimaan beasiswa baik surat pengumuman maupun surat panggilan tes. Penayangan daftar surat diurutkan berdasarkan tanggal surat. Tampilan utama sistem informasi penerimaan beasiswa internal dapat dilihat pada gambar 4.5, halaman ini dapat diakses oleh pengguna tanpa memerlukan proses login terlebih dahulu.



Gambar 4. 5 Halaman Utama

4.2.2.2. Pengelolaan Pengguna Sistem

Salah satu bagian dari pengelolaan pengguna adalah otentifikasi pengguna sistem. Otentifikasi pengguna sistem ditujukan agar saat seorang pengguna mengakses sistem, maka menu yang disediakan sesuai dengan level dari pengguna tersebut. Proses otentifikasi dimulai saat pengguna memilih menu login pada halaman utama. Form login dapat dilihat pada gambar 4.6 berikut.

Untuk melakukan login pengguna mengisi username serta password kemudian menekan tombol login, apabila pengguna ingin membatalkan proses login maka pengguna menekan tombol batal. Apabila pengguna lupa dengan password nya maka pengguna tersebut bisa meminta admin untuk mereset password dengan menekan link lupa password.



Masukkan Username dan Password untuk Login

Username

Password

[Lupa Password ?](#)

 Tulis ulang teks :

CAPTCHA

Gambar 4. 6 Form Login

Setelah pengguna memasukkan username dan password kemudian menekan tombol login, proses otentifikasi dijalankan dengan mencocokkan input dari user dengan data yang terdapat dalam database. Apabila field username dan atau password tidak diisi pada saat menekan tombol login maka sistem akan menampilkan pesan error seperti Gambar 4. 7.



Masukkan Username dan Password untuk Login

Username dan Password tidak boleh kosong

Username

Password

[Lupa Password ?](#)

 Tulis ulang teks :

CAPTCHA

Gambar 4. 7 Pesan Error Tidak Boleh Kosong

Apabila pengguna salah memasukkan username dan atau password maka sistem akan menampilkan pesan error seperti



Masukkan Username dan Password untuk Login

text CAPTCHA yang anda masukkan salah

Username

Password

[Lupa Password ?](#)

 Tulis ulang teks :
CAPTCHA

Gambar 4. 8 Pesan Error CAPTCHA Salah

Apabila pengguna salah memasukkan username dan atau password maka sistem akan menampilkan pesan error seperti Gambar 4. 9.



Masukkan Username dan Password untuk Login

Username dan atau Password yang anda masukkan salah

Username

Password

[Lupa Password ?](#)

 Tulis ulang teks :
CAPTCHA

Gambar 4. 9 Pesan Error Username dan Password Salah

Apabila pengguna berhasil login, sistem akan menampilkan halaman yang disesuaikan dengan level pengguna yaitu Admin, Pegawai, atau Atasan Langsung.



Gambar 4. 10 Halaman Utama Admin



Gambar 4. 11 Halaman Utama Pegawai



Gambar 4. 12 Halaman Utama Atasan Langsung

Termasuk dalam pengelolaan pengguna sistem adalah proses penambahan, pengubahan, serta penghapusan pengguna. Ketiga proses tersebut hanya bisa dilakukan oleh Admin. Proses penambahan dilakukan admin dengan mengakses halaman User seperti gambar 4.10. kemudian menekan tombol rekam



Gambar 4. 13 Halaman User

Admin kemudian mengisi form rekam user seperti gambar 4.11



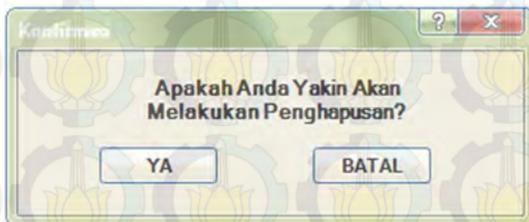
The image shows a web form titled "Form Rekam User". It contains several input fields and two buttons. The fields are labeled as follows:

Label	Value
NIP	1234567890
Username	username
Password	password
Konfirmasi Password	password
Nama Ibu Kandung	Ibu

At the bottom of the form, there are two buttons: "Simpan" (Save) and "Batal" (Cancel).

Gambar 4. 14 Form Rekam User

Untuk melakukan perubahan pada data pengguna, admin menekan link ubah pada halaman user, form untuk melakukan perubahan data user sama dengan form rekam user. Untuk melakukan penghapusan data pengguna, admin menekan link hapus pada halaman user. Sistem akan menampilkan konfirmasi apakah yakin akan melakukan penghapusan data seperti Gambar 4. 15



Gambar 4. 15 Konfirmasi Penghapusan

4.2.2.3. Pengelolaan Proses Pendaftaran

Proses pendaftaran dimulai saat pegawai memilih menu pendaftaran, halaman pendaftaran dapat dilihat pada Gambar 4. 16. Pendaftaran dilakukan dengan memilih universitas, jurusan, dan jenjang yang diminati kemudian menekan tombol daftar.

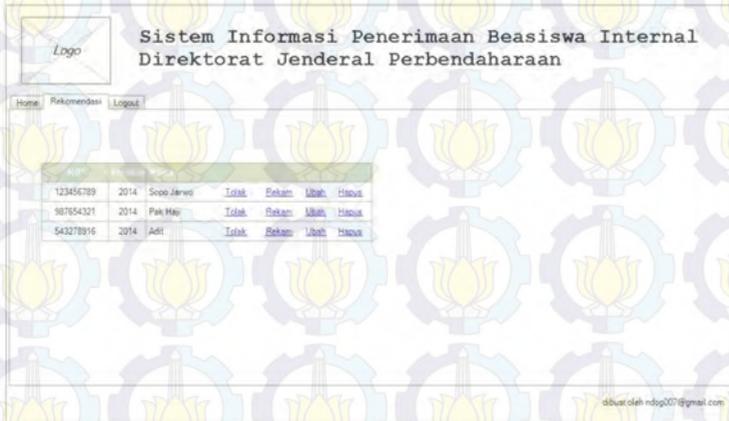
Gambar 4. 16 Halaman Pendaftaran

Salah satu syarat untuk melakukan pendaftaran adalah rekomendasi dari atasan langsung. Untuk meminta rekomendasi, Pegawai memilih menu Permintaan Rekomendasi seperti Gambar 4. 17

Tahun	No	Tipe	Status	Aksi
2013	123456789	Tersedia	Ulang	Batalan
2014	987654321	Menunggu	Ulang	Batalan

Gambar 4. 17 Halaman Permintaan Rekomendasi

Perekaman, Perubahan, dan penghapusan rekomendasi dilakukan oleh atasan langsung dengan mengakses halaman Rekomendasi seperti Gambar 4. 18 dan memilih link yang sesuai.



Gambar 4. 18 Halaman Rekomendasi

Form untuk merekam rekomendasi terlihat pada Gambar 4. 19

Form Rekam Rekomendasi

NIP: 198207292001121001

Nama: Fitria Yulianto

Integritas	<input type="checkbox"/>				
Intelektual	<input type="checkbox"/>				
Kreatifitas	<input type="checkbox"/>				
Motivasi	<input type="checkbox"/>				
Kedewasaan	<input type="checkbox"/>				
Administratif	<input type="checkbox"/>				
Kerjasama	<input type="checkbox"/>				

Simpan Batal

Gambar 4. 19 Form Rekam Rekomendasi

4.2.2.4. Pengelolaan Proses Seleksi Administratif

Proses Seleksi Administrasi dimulai dengan dibukanya pendaftaran untuk mengikuti seleksi beasiswa. Proses ini dilakukan oleh Admin dengan mengakses halaman Seleksi Administrasi seperti Gambar 4. 20

Gambar 4. 20 Halaman Seleksi Administrasi

Untuk melakukan pembukaan pendaftaran, Admin menekan tombol Pembukaan Pendaftaran kemudian mengisi form seperti Gambar 4. 21

Gambar 4. 21 Form Pembukaan Pendaftaran

Link lihat pada kolom detail digunakan untuk menampilkan rincian universitas dan jurusan yang telah dibuka untuk periode yang dipilih.

Form Detail Pembukaan Pendaftaran

Periode: 2014

Universitas	Jurusan	Detail
Universitas Diponegoro	S1 Akuntansi	Hapus
Universitas Diponegoro	S2 Hukum	Hapus
Universitas Brawijaya	S1 Ekonomi Kuantitatif	Hapus
ITS	S1 Sistem Informasi	Hapus

Tambah Kembali

Gambar 4. 22 Detail Pembukaan Pendaftaran

Untuk menambahkan detail universitas atau jurusan yang dibuka untuk periode tersebut, admin menekan tombol Tambah. Form yang digunakan untuk menambah detail pembukaan pendaftaran sama dengan form pembukaan pendaftaran.

Untuk menghapus detail universitas atau jurusan yang telah dibuka, admin menekan link hapus. Pada saat ditekan, sistem akan menampilkan konfirmasi penghapusan.

Seleksi Administratif dilakukan secara otomatis oleh sistem pada saat pegawai melakukan pendaftaran. Saat pegawai menekan tombol daftar, sistem akan memeriksa atribut dari data pegawai yang bersangkutan apakah sudah memenuhi kriteria yang ditentukan, kemudian menampilkan list hasil pemeriksaan atribut seperti gambar Gambar 4. 23.

List Hasil Seleksi Administratif

Jenjang yang dipilih :

Aspek Administrasi	Hasil
Berstatus PNS	✓
Masa Kerja sebagai PNS minimal 5 tahun	✓
Masa Kerja minimal 2 tahun sejak pendidikan terakhir	✓
Tidak sedang dicalonkan dalam program beasiswa lain	✓
Tidak sedang menjalani hukuman disiplin	✓
Tidak dalam proses mutasi ke direktorat lain	✓
Rekomendasi dari Atasan	✓
Usia Maksimal 32 tahun	✗
Pengatur (II/c) dengan masa kerja 2 tahun	✓
memiliki ijazah D III	✗
IPK minimal 2,75	✓

Anda TIDAK LOLOS seleksi administratif

Gambar 4. 23 List Seleksi Administratif

4.2.2.5. Pengelolaan Proses Tes Tertulis

Penanganan proses tes tertulis dilakukan oleh admin. Proses tes tertulis dimulai dengan perekaman surat panggilan tes tertulis yang dilakukan dengan mengakses halaman tes tertulis seperti Gambar 4. 24 dan menekan tombol rekam surat panggilan.



Sistem Informasi Penerimaan Beasiswa Internal
Direktorat Jenderal Perbendaharaan

Home | User | Universitas | Jurusan | Seleksi Administrasi | Tes Tertulis | Tes Wawancara | Pengumuman | Logout

Rekam Surat Panggilan Rekam Hasil Tes

SPT001	2011	01 Mei 2011	Cetak
SPT002	2012	10 April 2012	Cetak
SPT003	2013	28 April 2013	Cetak

di buatkan oleh: nbaq007@gmail.com

Gambar 4. 24 Halaman Tes Tertulis

Form untuk melakukan perekaman surat panggilan tes tertulis terlihat seperti Gambar 4. 25



Form Rekam Surat Panggilan Tes

Periode / tahun: 2014

Tanggal Surat: 10 Mei 2014

Nomor Surat: SPT001

Jenis Surat: 1

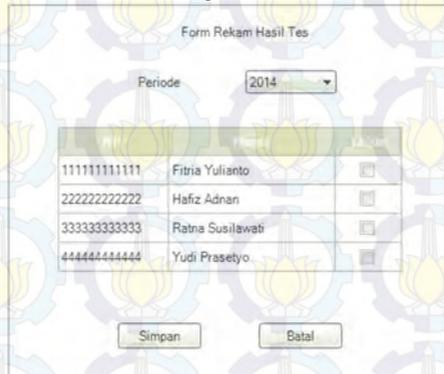
Perihal: Panggilan Tes Tertulis Periode 2014

NIP Penandatanganan: 1234567890

Kirim Batal

Gambar 4. 25 Form Rekam Surat Panggilan Tes

Untuk melakukan perekaman hasil tes tertulis, admin menekan tombol hasil tes pada halaman tes tertulis, form untuk melakukan perekaman terlihat pada Gambar 4. 26.



No	Nama	Jenis Tes	Status
11111111111111	Fitria Yulianto		<input type="checkbox"/>
22222222222222	Hafiz Adnan		<input type="checkbox"/>
33333333333333	Ratna Susilawati		<input type="checkbox"/>
44444444444444	Yudi Prasetyo		<input type="checkbox"/>

Gambar 4. 26 Form Rekam Hasil Tes

4.2.2.6. Pengelolaan Proses Tes Wawancara

Penanganan proses tes wawancara dilakukan oleh admin. Proses tes wawancara dimulai dengan perekaman surat panggilan tes wawancara yang dilakukan dengan mengakses halaman tes wawancara seperti Gambar 4. 27.



ID	Tahun	Tgl Tes	Status
SPW001	2011	26 Jun 2011	Ceklis
SPW002	2012	28 Jun 2012	Ceklis
SPW003	2013	27 Jun 2013	Ceklis

Gambar 4. 27 Halaman Tes Wawancara

Untuk melakukan perekaman hasil tes wawancara, admin menekan tombol rekam hasil tes pada halaman tes wawancara, form untuk melakukan perekaman hasil tes wawancara sama dengan form untuk melakukan perekaman hasil tes tertulis seperti terlihat pada Gambar 4. 26.

4.2.2.7. Pengelolaan Data Referensi

Pengelolaan data referensi dilakukan oleh admin. Data referensi meliputi data Universitas dan data Jurusan. Untuk melakukan pengelolaan data universitas, admin mengakses halaman universitas seperti terlihat pada Gambar 4. 28.



Gambar 4. 28 Halaman Universitas

Untuk melakukan perekaman data universitas, admin menekan tombol rekam pada halaman universitas kemudian mengisi form rekam universitas seperti terlihat pada Gambar 4. 29

Form Rekam Universitas

Kode Universitas

Nama Universitas

Alamat

Keterangan

Gambar 4. 29 Form Rekam Universitas

Untuk melakukan perubahan data universitas, admin menekan link ubah pada halaman universitas, form untuk melakukan perubahan data universitas sama dengan form rekam universitas. Untuk melakukan penghapusan data universitas, admin menekan link hapus pada halaman universitas, sistem akan menampilkan konfirmasi penghapusan.

Pengelolaan data jurusan dilakukan admin dengan mengakses halaman jurusan seperti terlihat pada Gambar 4. 30.


**Sistem Informasi Penerimaan Beasiswa Internal
Direktorat Jenderal Perbendaharaan**

[Home](#) | [User](#) | [Universitas](#) | [Jurusan](#) | [Salaku Administrasi](#) | [Tes Tertulis](#) | [Tes Wawancara](#) | [Pengumuman](#) | [Logout](#)

Pilih Universitas

J001	Sistem Informasi	Ubah	Hapus
J002	Ekonomi Kuantitatif	Ubah	Hapus

dibuat oleh rotop007@gmail.com

Gambar 4. 30 Halaman Jurusan

Untuk melakukan perekaman data jurusan pada suatu universitas, admin memilih universitas yang diinginkan kemudian menekan tombol rekam pada halaman jurusan. Form untuk melakukan perekaman data jurusan terlihat pada Gambar 4. 31.

The image shows a web form titled "Form Rekam Jurusan". It contains the following fields and values:

- Kode Jurusan: J001
- Nama Jurusan: Sistem Informasi
- Fakultas: Teknologi Informasi
- Keterangan: (empty)

At the bottom of the form are two buttons: "Simpan" (Save) and "Batal" (Cancel).

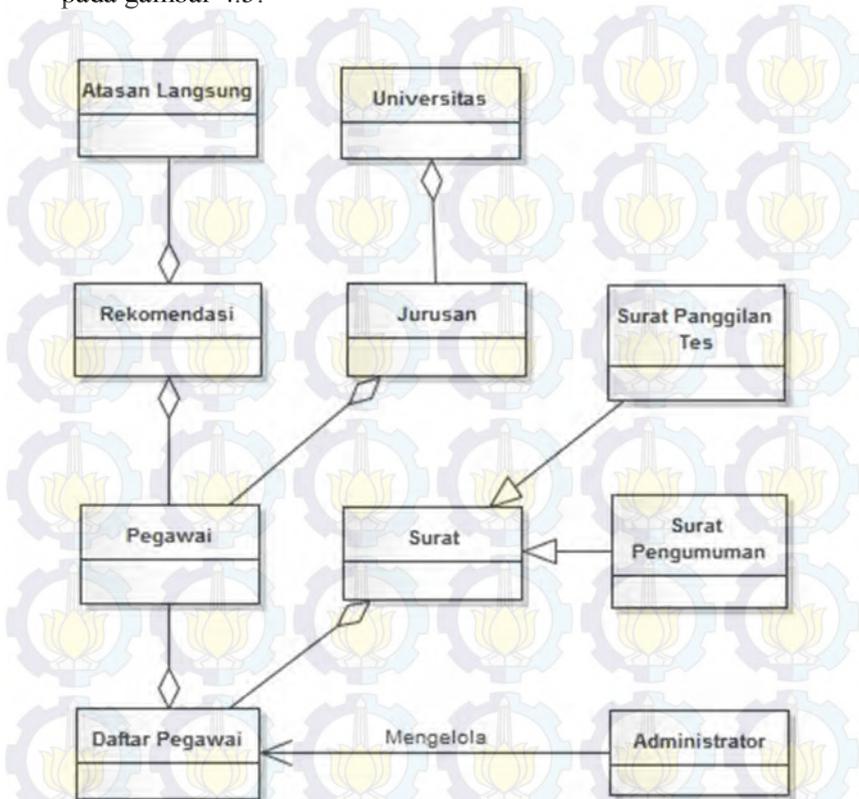
Gambar 4. 31 Form Rekam Jurusan

Untuk melakukan perubahan data jurusan, admin menekan link ubah dibelakang data jurusan yang diinginkan pada halaman jurusan. Form untuk melakukan perubahan data sama dengan form untuk melakukan perekaman data jurusan. Untuk melakukan penghapusan data jurusan, admin menekan link hapus dibelakang data jurusan yang ingin dihapus, sistem akan menampilkan konfirmasi penghapusan data.

4.2.3.Domain Model

Domain model dalam ICONIX Process berisi istilah (*glossary*) yang digambarkan dalam bentuk domain object, dimana antar domain tersebut memiliki hubungan agregasi dan generalisasi (relasi *has-a* dan *is-a*) (Rosenberg & Stephens, 2007). Berdasarkan analisa yang dilakukan oleh penulis, domain model

dari Sistem Informasi Penerimaan Beasiswa Internal dapat dilihat pada gambar 4.5.



Gambar 4. 32 Domain Model

Dari domain model tersebut diketahui terdapat beberapa class yang di inialisasi antara lain :

1. Pegawai

Merupakan seluruh pegawai di lingkup direktorat jenderal perbendaharaan yang terkait dengan proses penerimaan beasiswa internal.

2. Universitas
Merupakan universitas yang menjadi mitra kerja sama beasiswa dengan Dirjen Perbendaharaan.
3. Jurusan
Merupakan jurusan dalam universitas mitra kerjasama yang dijadikan pilihan program beasiswa internal.
4. Rekomendasi
Merupakan penilaian dari atasan langsung terhadap pegawai yang berminat untuk mengikuti seleksi penerimaan beasiswa internal.
5. Surat Panggilan Tes
Merupakan dokumen yang ditandatangani oleh sekretaris ditjen perbendaharaan yang berisi pengumuman hasil seleksi tahap tertentu serta panggilan untuk mengikuti seleksi tahap berikutnya.
6. Surat Pengumuman
Merupakan dokumen yang ditandatangani oleh sekretaris ditjen perbendaharaan yang berisi pengumuman pegawai yang berhak memperoleh beasiswa (lolos seluruh tahapan seleksi)

4.2.4. Use Case

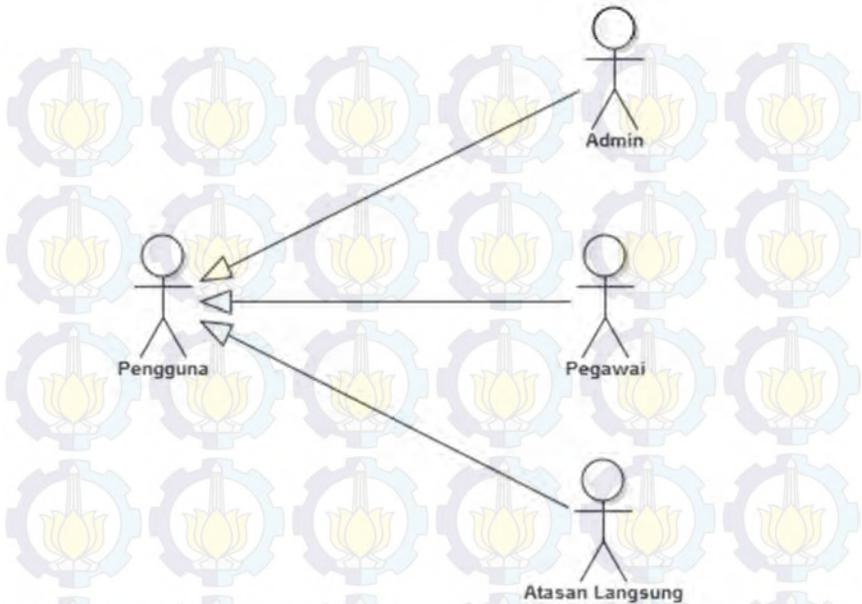
4.2.4.1. Use Case Diagram

Use Case Diagram merupakan penggambaran interaksi yang bisa dilakukan oleh pengguna (*actor*) dengan sistem. Daftar usecase yang terdapat dalam sistem informasi penerimaan beasiswa internal antara lain :

1. Login (UC01)
2. Logout (UC02)
3. Meminta Reset Password (UC03)
4. Melihat Pengumuman (UC04)
5. Rekam User (UC05)
6. Ubah User (UC06)

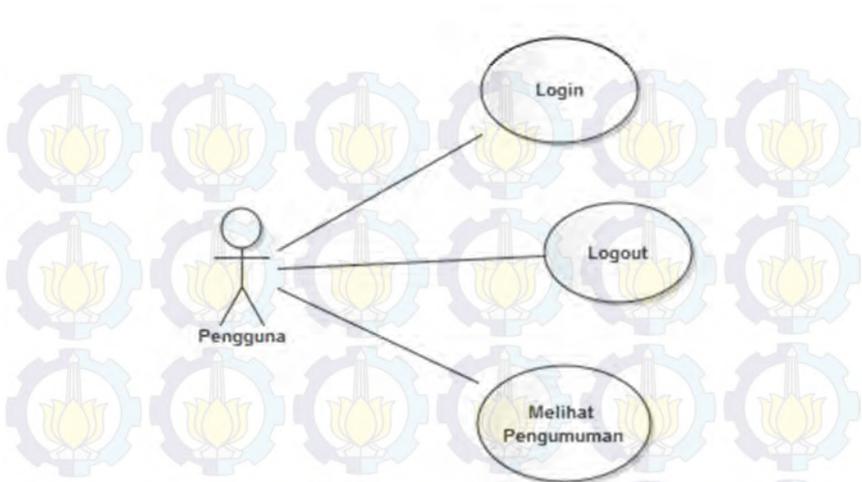
7. Hapus User (UC07)
8. Reset Password User (UC08)
9. Rekam Universitas (UC09)
10. Ubah Universitas (UC10)
11. Hapus Universitas (UC11)
12. Rekam Jurusan (UC12)
13. Ubah Jurusan (UC13)
14. Hapus Jurusan (UC14)
15. Membuka Pendaftaran (UC15)
16. Tambah Detail Pembukaan Pendaftaran (UC16)
17. Hapus Detail Pembukaan Pendaftaran (UC17)
18. Rekam Surat Pembukaan Pendaftaran (UC18)
19. Rekam Surat Panggilan Tes Tertulis (UC19)
20. Rekam Hasil Tes Tertulis (UC20)
21. Rekam Surat Panggilan Tes Wawancara (UC21)
22. Rekam Hasil Tes Wawancara (UC22)
23. Rekam Surat Pengumuman (UC23)
24. Mendaftar (UC24)
25. Membatalkan Pendaftaran (UC25)
26. Melihat Riwayat Pendaftaran (UC26)
27. Rekam Permintaan Rekomendasi (UC27)
28. Ubah Permintaan Rekomendasi (UC28)
29. Hapus Permintaan Rekomendasi (UC29)
30. Melihat Permintaan Rekomendasi (UC30)
31. Rekam Rekomendasi (UC31)
32. Ubah Rekomendasi (UC32)
33. Hapus Rekomendasi (UC33)
34. Tolak Rekomendasi (UC34)

Terdapat 4 Aktor dalam sistem informasi penerimaan beasiswa antara lain, Administrator, Pegawai, Atasan Langsung, serta Pengguna (yang merupakan generalisasi dari ketiga aktor tersebut). Relasi dari keempat aktor dapat dilihat pada Gambar 4.33.

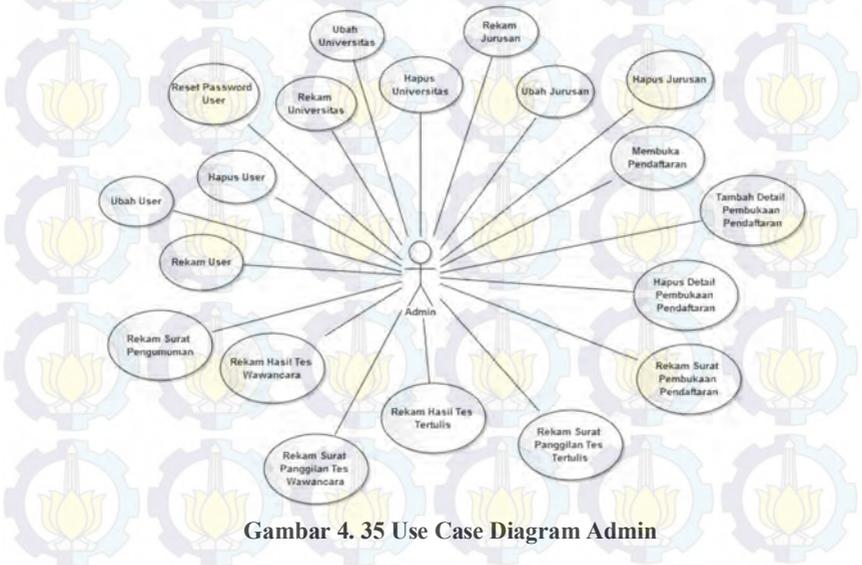


Gambar 4. 33 Aktor

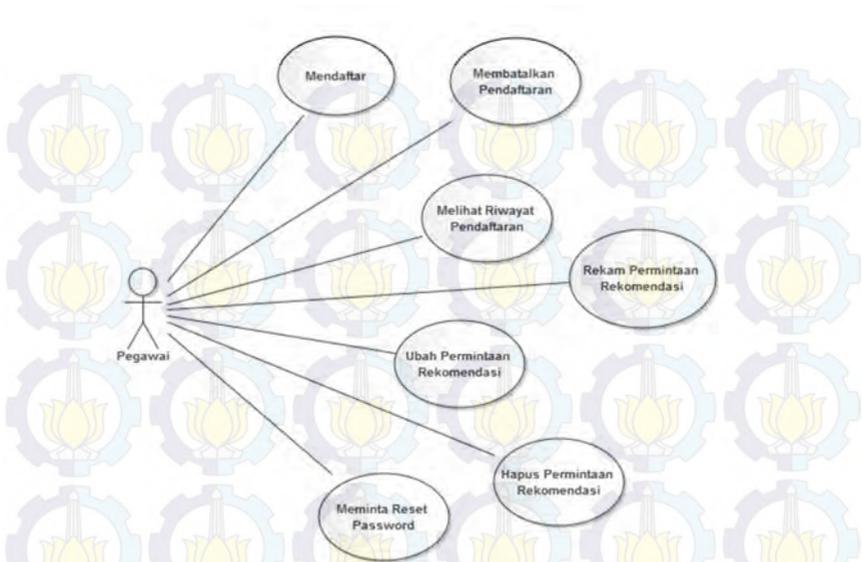
Setiap Aktor memiliki hak dan kewenangan yang berbeda sehingga usecase yang ada terbagi untuk masing-masing aktor tersebut. Usecase diagram untuk Pengguna bisa dilihat pada Gambar 4. 34, usecase diagram untuk Administrator bisa dilihat pada Gambar 4. 35, usecase diagram untuk Pegawai bisa dilihat pada Gambar 4. 36, sedangkan usecase diagram untuk Atasan Langsung bisa dilihat pada Gambar 4. 37.



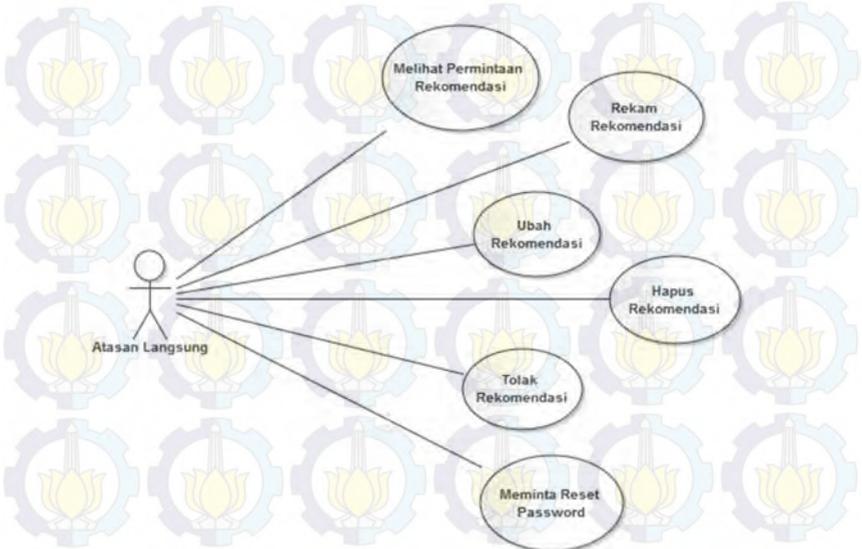
Gambar 4. 34 Usecase Diagram Pengguna



Gambar 4. 35 Use Case Diagram Admin



Gambar 4. 36 Use Case Diagram Pegawai



Gambar 4. 37 Use Case Diagram Atasan Langsung

4.2.4.2. Use Case Description

Use Case Description merupakan narasi dari setiap use case yang terdapat dalam sistem. *Use case description* biasanya ditulis dengan menggunakan kalimat aktif serta *menggunakan event-response flow* yang menggambarkan aktivitas/ dialog antara aktor dengan sistem.

Contoh dari use case description yang ada pada sistem informasi penerimaan beasiswa internal dapat dilihat pada tabel 4.1. Use case description secara lengkap untuk semua use case dapat dilihat pada lampiran A.

4.2.5. Robustness Diagram

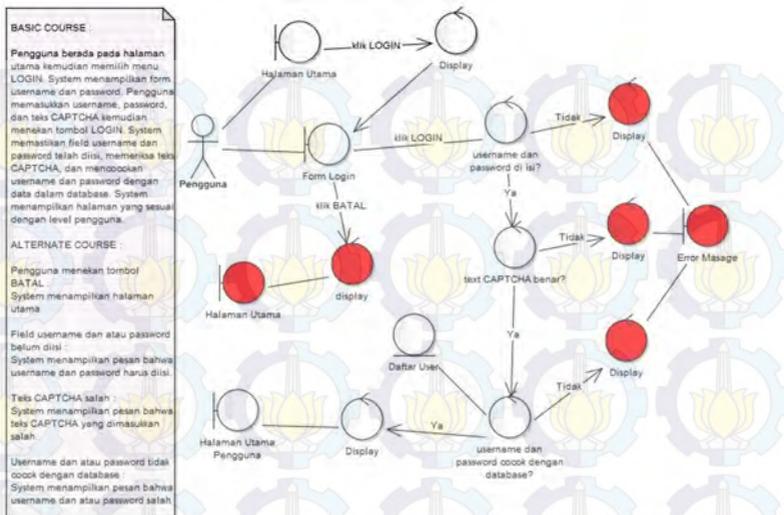
Dalam bukunya, Rosenberg dan Stephens (Rosenberg & Stephens, 2007) mengatakan bahwa robustness diagram adalah gambaran object dari sebuah use case. Robustness diagram merupakan perpaduan antara class diagram dan activity diagram yang menggambarkan behaviour yang disebutkan dalam use case. Gambar 4. 38. memperlihatkan Robustness Diagram Login, Detail dari Robustness Diagram untuk semua Use Case pada aplikasi ini dapat dilihat dalam lampiran B.

4.2.6. Sequence Diagram

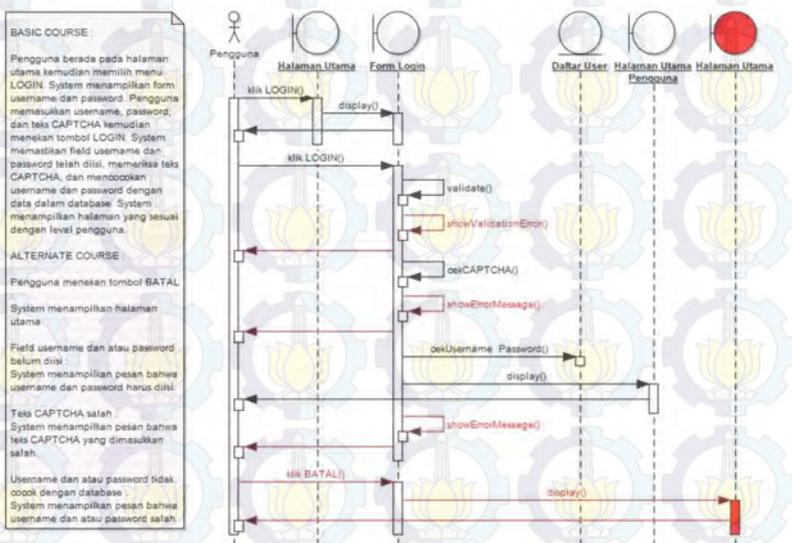
Setelah membuat Robustness Diagram, langkah selanjutnya adalah membuat desain lebih rinci dengan membuat Sequence Diagram. Sequence Diagram menunjukkan bagaimana suatu objek, method, dan class berinteraksi antara satu sama lain dan bagaimana urutan pesan dipertukarkan dalam sistem tersebut. Gambar 4. 39. memperlihatkan Sequence Diagram Login, Detail dari Sequence Diagram dari masing-masing Use Case pada aplikasi ini dapat dilihat dalam lampiran C.

Tabel 4. 1 Use Case Description Login

Usecase	:	Login (UC01)
Aktor	:	Pengguna
Pre-condition		Pengguna berada pada halaman utama
Trigger		Pengguna menklik menu LOGIN
Basic Course	:	System menampilkan form username dan password. Pengguna memasukkan username, password, dan teks CAPTCHA kemudian menekan tombol LOGIN. System memastikan field username dan password telah diisi, memeriksa teks CAPTCHA, dan mencocokkan username dan password dengan data dalam database.
Post-condition		System menampilkan halaman utama sesuai dengan level pengguna.
Alternate Course	:	<p>Pengguna menekan tombol BATAL : System menampilkan halaman utama</p> <p>Field username dan atau password belum diisi : System menampilkan pesan bahwa username dan password harus diisi.</p> <p>Teks CAPTCHA salah : System menampilkan pesan bahwa teks CAPTCHA yang dimasukkan salah.</p> <p>Username dan atau password tidak cocok dengan database : System menampilkan pesan bahwa username dan atau password salah</p>



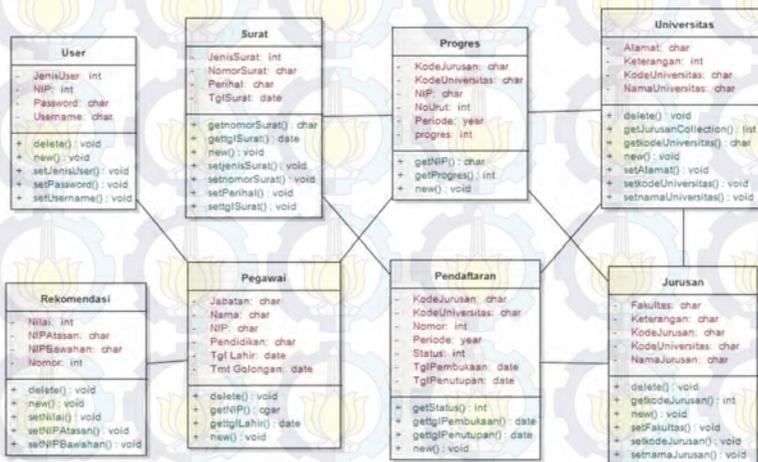
Gambar 4. 38 Robustness Diagram Login



Gambar 4. 39 Sequence Diagram Login

4.2.7. Class Diagram

Dari Domain Model yang kemudian diupdate setelah menyusun robustness analysis serta sequence diagram, kita dapat menyusun Class yang terdapat dalam SIPERBEN. Class tersebut digambarkan dalam Class Diagram seperti Gambar 4. 40.



Gambar 4. 40 Class Diagram

4.2.8. Test Case

Test Case merupakan sekumpulan skenario yang dirancang untuk melakukan pengujian terhadap aplikasi, hal ini dilakukan untuk menjamin sebuah aplikasi dapat berjalan dengan benar. Test case dibuat untuk setiap use case untuk menentukan apakah aplikasi memberikan hasil yang sesuai dengan use case yang telah dirancang. Detail mengenai Test Case yang dibuat untuk Sistem Informasi Penerimaan Beasiswa Internal dapat dilihat pada lampiran D.

BAB V

PEMBUATAN APLIKASI DAN UJI COBA

Bab ini akan menjelaskan proses pada tahapan pembuatan aplikasi dan uji coba sistem.

5.1.Lingkungan Pengembangan Aplikasi

Pengembangan aplikasi dilakukan dengan menggunakan laptop yang memiliki spesifikasi dan detail perangkat keras dan perangkat lunak yang dapat dilihat dalam tabel 5.1.

Tabel 5. 1 Spesifikasi Perangkat Keras

Spesifikasi	
Processor	Intel® Core™ i3 CPU 380UM @1.33Ghz
Memori	4GB RAM
Sistem Operasi	Windows 7 Ultimate

Proses pengembangan aplikasi ini menggunakan beberapa teknologi terkait dengan bahasa pemrograman maupun database yang digunakan, teknologi tersebut dapat dilihat pada tabel 5.2.

5.2.Konfigurasi

Pada Tahapan konfigurasi dalam pengembangan aplikasi SIPERBEN langkah yang dilakukan antara lain pembuatan struktur direktori serta pembuatan file konfigurasi.

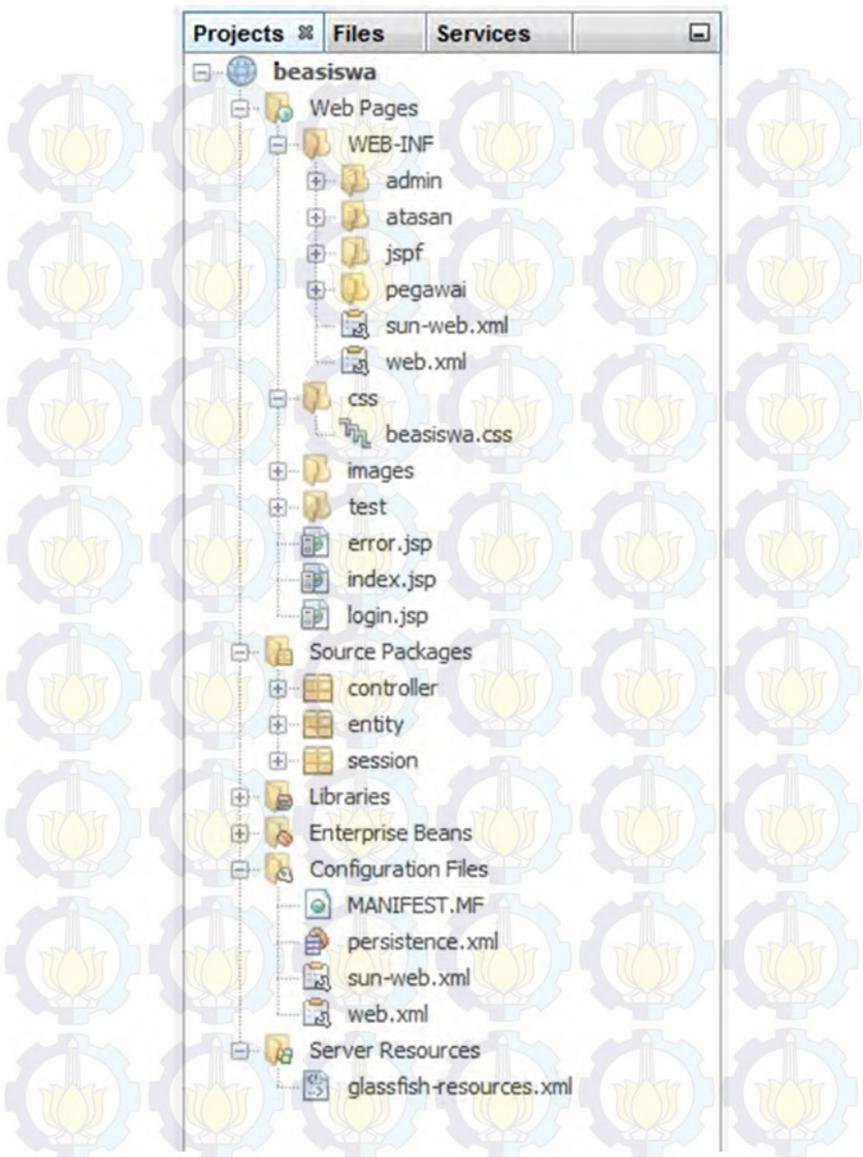
Tabel 5.2 Teknologi yang digunakan

Tools	
Webserver	GlassFish 3.1.2
Bahasa Pemrograman	Java, JSP
Development Kit	JDK 1.7.
Database	MySQL
Database Management	PhpMyAdmin
Editor	Netbeans IDE 7.1.1
Library Pendukung	
Java Persistence	JPA Eclipselink 2.0
Captcha	Simplecaptcha 1.2

5.2.1. Pembuatan Struktur Direktori

Pembuatan struktur direktori aplikasi SIPERBEN mengacu pada standar pembuatan Web Application Java, direktori utama adalah direktori dengan nama beasiswa, kemudian terdapat 2 sub-direktori yaitu 'source package' untuk menampung source file java, dan sub-direktori 'web pages' untuk menampung file-file JSP serta CSS. Gambar 5.1. memperlihatkan tampilan dari struktur direktori yang dibuat dalam aplikasi SIPERBEN.

Untuk memudahkan pengelompokan halaman webpage yang digunakan dalam aplikasi SIPERBEN, maka pada direktori web pages dibuat 3 direktori yaitu admin, atasan, dan pegawai. Disamping itu juga terdapat direktori jspf yang menampung *fragment* halaman yang digunakan.



Gambar 5. 1 Struktur Direktori Aplikasi

Direktori source packages dibagi lagi menjadi tiga packages, yaitu controller, entity, dan session.

1. packages controller berisi file servlet yang mengatur tampilan halaman jsp.
2. packages entity berisi file java yang merupakan *entity class*.
3. packages session berisi kode program untuk proses interaksi dengan setiap *entity class* seperti *create, read, update, delete* (CRUD).

5.2.2. Konfigurasi File

Pengembangan aplikasi berbasis Java Web membutuhkan beberapa file konfigurasi yang memiliki kegunaan masing-masing. Tabel 5.3 menunjukkan beberapa file konfigurasi yang digunakan beserta fungsinya.

Tabel 5.3 Konfigurasi file

Nama File	Fungsi
Glassfish-resource.xml	File yang mengatur resource yang digunakan dalam aplikasi seperti koneksi dengan database, connection pool, serta konfigurasi user dan password untuk koneksi tersebut
web.xml	File XML berisi pengaturan terkait user/pengguna, role, serta pengaturan servlet tambahan seperti captcha servlet.
persistence.xml	File XML yang merupakan standar konfigurasi dari JPA. File ini berisi <i>entity bean</i> pada aplikasi.

Gambar 5.2 sampai dengan 5.4. memperlihatkan tampilan dari konfigurasi file yang digunakan dalam aplikasi SIPERBEN.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <DOCTYPE resources PUBLIC "-//GlassFish.org//DTD GlassFish Application Server 3.1 Resource Definitions//EN
3 <resources>
4 <jdbc-resource enabled="true" jndi-name="jdbc/beasiswa" object-type="user" pool-name="BeasiswaPool">
5 <description/>
6 </jdbc-resource>
7 <jdbc-connection-pool allow-non-component-callers="false" associate-with-thread="false" connection-creati
8 <description>connection to beasiswa database</description>
9 <property name="url" value="jdbc:derby://localhost:3306/beasiswa/">
10 <property name="User" value="saas/">
11 <property name="Password" value="saas/">
12 </jdbc-connection-pool>
13 </resources>
14

```

Gambar 5.2 Konfigurasi File glassfish-resources.xml

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <persistence version="2.0" xmlns="http://java.sun.com/xml/ns/persistence" xmlns:xsi="
3 <persistence-unit name="beasiswaPU" transaction-type="JTA">
4 <jta-data-source>jdbc/beasiswa</jta-data-source>
5 <exclude-unlisted-classes>false</exclude-unlisted-classes>
6 <properties>
7 <property name="eclipselink.logging.level" value="FINEST"/>
8 </properties>
9 </persistence-unit>
10 </persistence>

```

Gambar 5.3 Konfigurasi File persistance.xml

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app version="3.0" xmlns="http://java.sun.com/xml/ns/javaee" xmlns:xsi="http://w
3 <servlet>
4 <servlet-name>SimpleCaptcha</servlet-name>
5 <servlet-class>n1.captcha.servlet.SimpleCaptchaServlet</servlet-class>
6 <init-param>
7 <param-name>captcha-width</param-name>
8 <param-value>250</param-value>
9 </init-param>
10 <init-param>
11 <param-name>captcha-height</param-name>
12 <param-value>75</param-value>

```

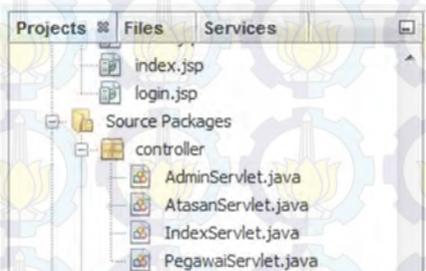
Gambar 5.4 Konfigurasi File web.xml

5.3. Pembuatan Kelas

Seperti dijelaskan pada bagian konfigurasi, untuk memudahkan penyusunan serta penulisan program, source packages terbagi kedalam 3 packages yaitu controller, entity, dan session. Penjelasan terkait penulisan kode program dari masing-masing packages adalah sebagai berikut.

5.3.1.Package ‘controller’

Package ini berisi class java yang berfungsi sebagai controller dalam aplikasi SIPERBEN. Controller ini memiliki fungsi menangani *request* dan mengembalikan *response* kepada client user. Hal utama yang dilakukan controller dalam aplikasi ini adalah membuat, membaca, memperbarui data, serta menghapus data. Dalam package ini terdiri dari 4 file yaitu IndexServlet, AdminServlet, AtasanServlet, serta PegawaiServlet. Gambar 5.5 menunjukkan kelas yang terdapat dalam package controller.



Gambar 5. 5 File dalam Package controller

IndexServlet.java menangani request halaman utama, login, logout, serta validatecapcha. Gambar 5.6. memperlihatkan potongan kode program yang terdapat dalam class IndexServlet.java :

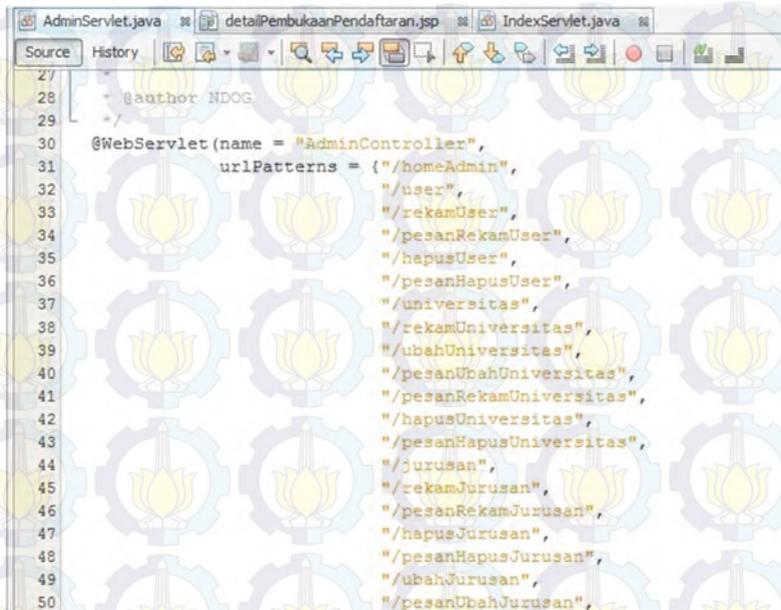
```

24 $@WebServlet(name = "IndexServlet", loadOnStartup = 1,
25             urlPatterns = {"/logout","/login","/validatecapcha"})
26 |
27 public class IndexServlet extends HttpServlet {
28
29     @EJB
30     private SuratFacade suratFacade;
31     private String userPath;
32
33     /**
34      *
35      */
36     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
37         throws ServletException, IOException {
38         HttpSession session = request.getSession(true);
39         //request.getUserPrincipal().getName();
40         userPath = request.getServletPath();
41
42         if (userPath.equals("/login")) {
43             userPath = "/login.jsp";
44         }
45     }
46 }

```

Gambar 5. 6 Kode Program file IndexServlet.java

AdminServlet.java menangani request halaman yang ditangani oleh admin seperti user, universitas, jurusan, seleksi administrasi, tes tertulis, tes wawancara, serta pengumuman. Gambar 5.7. memperlihatkan potongan kode program dari class AdminServlet.java :



```

27
28  * @author NDOG
29  */
30  @WebServlet(name = "AdminController",
31             urlPatterns = {"/homeAdmin",
32                           "/user",
33                           "/rekamUser",
34                           "/pesanRekamUser",
35                           "/hapusUser",
36                           "/pesanHapusUser",
37                           "/universitas",
38                           "/rekamUniversitas",
39                           "/ubahUniversitas",
40                           "/pesanUbahUniversitas",
41                           "/pesanRekamUniversitas",
42                           "/hapusUniversitas",
43                           "/pesanHapusUniversitas",
44                           "/jurusan",
45                           "/rekamJurusan",
46                           "/pesanRekamJurusan",
47                           "/hapusJurusan",
48                           "/pesanHapusJurusan",
49                           "/ubahJurusan",
50                           "/pesanUbahJurusan",

```

Gambar 5. 7 Kode Program AdminServlet.java

AtasanServlet.java menangani request halaman yang digunakan oleh Atasan Langsung yaitu terkait dengan rekomendasi. Sedangkan PegawaiServlet.java menangani request halaman yang digunakan oleh Pegawai seperti pendaftaran, serta permintaan rekomendasi.

5.3.2.Package 'entity'

Package ini berisi class-class java yang digunakan untuk merepresentasikan seluruh data yang digunakan dalam aplikasi

SIPERBEN. Class java dalam package ini disebut entity karena merepresentasikan data yang digunakan dalam bentuk object. Gambar 5.8. memperlihatkan class java yang terdapat dalam package entity.



Gambar 5. 8 class java yang terdapat dalam packages entity

Class java dalam package entity berisi deklarasi object serta atributnya disertai dengan method *get* dan *set* untuk setiap atribut yang dimiliki. Gambar 5.9 menunjukkan potongan kode program dari Role.java

```

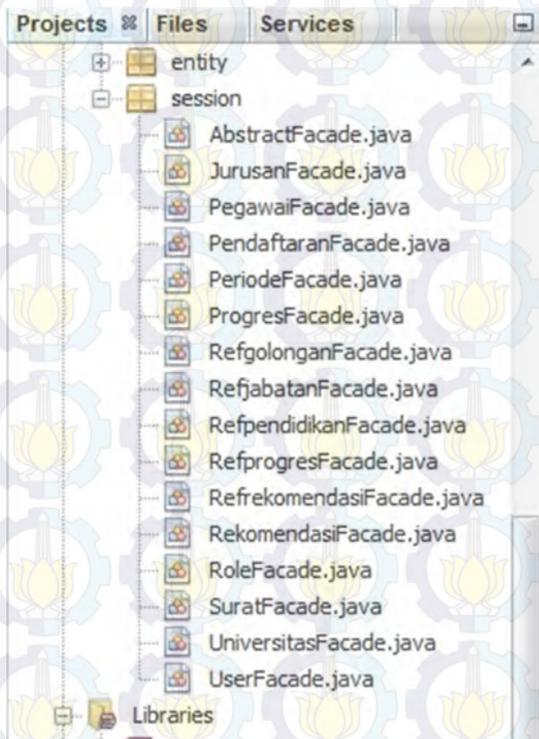
19 @Entity
20 @Table(name = "role")
21 @XmlRootElement
22 @NamedQueries({
23     @NamedQuery(name = "Role.findAll", query = "SELECT r FROM Role r"),
24     @NamedQuery(name = "Role.findById", query = "SELECT r FROM Role r WHERE r.id=:id"),
25     @NamedQuery(name = "Role.findByidUser", query = "SELECT r FROM Role r WHERE r.idUser=:idUser")})
26 public class Role implements Serializable {
27     private static final long serialVersionUID = 1L;
28     @Id
29     @GeneratedValue(strategy = GenerationType.IDENTITY)
30     @Basic(optional = false)
31     @NotNull
32     @Column(name = "idRole")
33     private Integer idRole;
34     @Column(name = "idUser")
35     private String idUser;
36     @JoinTable(name = "user_role", joinColumns = {
37         @JoinColumn(name = "idRole", referencedColumnName = "idRole")}, inverseJoinColumns = {
38         @JoinColumn(name = "idUser", referencedColumnName = "idUser")})
39     @ManyToMany
40     private Collection<User> userCollection;
41

```

Gambar 5. 9 Kode Program pada Role.java

5.3.3. Package 'session'

Package ini berisi class java yang digunakan sebagai penghubung antara class java yang terdapat dalam package 'entity' dengan controller yang berada dalam package 'controller'. Gambar 5.10. memperlihatkan class java yang terdapat dalam package session.



Gambar 5. 10 Class java yang terdapat dalam Package Session

Class java dalam package ini berisi method-method yang digunakan untuk memanggil data dari class yang terdapat dalam entity untuk diproses dalam class controller sehingga dapat

ditampilkan ke dalam view. Gambar 5.11 menunjukkan contoh kode program pada class AbstractFacade.java



```

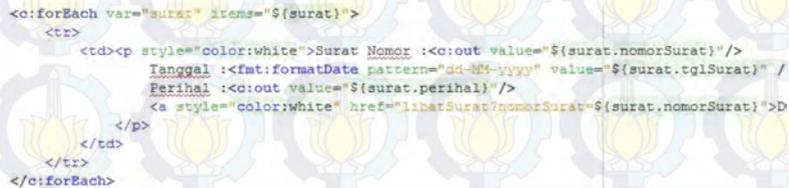
Source History
15 public abstract class AbstractFacade<T> {
16     private Class<T> entityClass;
17
18     public AbstractFacade(Class<T> entityClass) {
19         this.entityClass = entityClass;
20     }
21
22     protected abstract EntityManager getEntityManager();
23
24     public void create(T entity) {
25         getEntityManager().persist(entity);
26     }
27
28     public void edit(T entity) {
29         getEntityManager().merge(entity);
30     }
31
32     public void remove(T entity) {
33         getEntityManager().remove(getEntityManager().merge(entity));
34     }
35 }

```

Gambar 5. 11 Kode Program pada AbstractFacade.java

5.4.Pembuatan Antarmuka

Pembuatan antarmuka untuk aplikasi ini menggunakan Java Server Pages (JSP) yang kemudian dipercantik menggunakan CSS. Pembuatan antarmuka dilakukan sesuai dengan desain pada bab 4. Halaman utama aplikasi menampilkan list pengumuman terbaru. Untuk menampilkan pengumuman tersebut digunakan method *foreach* yang melakukan iterasi list data surat dan ditampilkan pada halaman website. Potongan kode program pada halaman index.jsp dapat dilihat pada gambar 5.12.



```

<c:forEach var="surat" items="${surat}">
  <tr>
    <td><p style="color:white">Surat Nomor :<out value="${surat.nomorSurat}"/>
    Tanggal :<fmt:formatDate pattern="dd-MM-yyyy" value="${surat.tglSurat}" />
    Perihal :<out value="${surat.perihal}" />
    <a style="color:white" href="lihatSurat?nomorSurat=${surat.nomorSurat}">D
  </td>
</tr>
</c:forEach>

```

Gambar 5. 12 Kode Program foreach pada Index.jsp

Tampilan halaman utama dari SIPERBEN yang dihasilkan oleh file `Index.jsp` dapat dilihat pada gambar 5.13.



Gambar 5.13 Tampilan Halaman Utama

Untuk memudahkan pembuatan antarmuka, maka penulis membuat *fragment* yang digunakan sebagai *header* maupun *footer* pada halaman aplikasi. Gambar 5.14. memperlihatkan contoh kode program untuk *fragment* pada `headerAdmin.jspf`.

```

<div id="inner">
  <div id="content">
    <p>Pengumuman terbaru terkait Beasiswa Internal Direktorat Jenderal Perbendaharaan </p>
    <table id="pengumuman">
      <tbody>
        <tr>
          <td><p style="color:white">Surat Nomor :<out value fore="surat.nomorSurat" />
            Tanggal :<fmt:formatDate pattern="dd-MM-yyyy" value fore="surat.tglSurat" />
            Perihal :<out value fore="surat.perihal" />
            <a style="color:white" href fore="lihatSurat?nomorSurat fore="surat.nomorSurat" />
          </td>
        </tr>
      </tbody>
    </table>
  </div>
</div>

```

Gambar 5.14 Kode Program file `headerAdmin.jspf`

File headerAdmin.jspf digunakan untuk menampilkan header serta menu utama pada halaman yang digunakan oleh Administrator. Tampilan header dari headerAdmin.jspf dapat dilihat pada Gambar 5.15.



Gambar 5. 15 Tampilan yang dihasilkan headerAdmin.jspf

Pengaturan tampilan seperti warna, gambar latar, jenis huruf serta pengaturan tampilan lainnya dilakukan dengan menggunakan file css (Cascading Style Sheets). Potongan kode program pada file beasiswa.css dapat dilihat pada gambar 5.16.

```
#header {
    background: #000 url(../images/header.png) no-repeat top left;
    min-height: 130px;
    margin: 0 auto;
    overflow: hidden;
    width: 960px;
}

#title-area {
    float: left;
    overflow: hidden;
    padding: 40px 0 0 30px;
    width: 840px;
}
```

Gambar 5. 16 Kode Program pada file beasiswa.css

5.5.Pembuatan Fungsionalitas Aplikasi

Fungsionalitas aplikasi dibuat berdasarkan kebutuhan fungsional yang telah didefinisikan pada bab sebelumnya. Sebagian besar fungsionalitas aplikasi dibuat pada class java yang terdapat dalam packages controller. Berikut ini adalah beberapa contoh pembuatan fungsionalitas aplikasi SIPERBEN.

1. Validasi pengguna sistem

Untuk melakukan validasi terhadap pengguna yang mengakses aplikasi digunakan method `captcha.isCorrect(answer)` dimana method tersebut memeriksa apakah teks yang dimasukkan oleh pengguna sudah sesuai dengan captcha, apabila salah maka sistem akan menampilkan pesan error. Apabila validasi captcha sukses, maka selanjutnya dilakukan validasi terhadap username dan password yang dimasukkan oleh pengguna dengan menggunakan method `request.login(username, password)`. Method tersebut akan memeriksa username dan password yang dimasukkan oleh pengguna dengan data yang tersimpan pada tabel user dan table role. Potongan kode program yang digunakan untuk proses validasi pengguna dapat dilihat pada gambar 5.17.

```

if (userPath.equals("/login")) {
    userPath = "/login.jsp";
}

if (userPath.equals("/validatecaptcha")) {
    Captcha captcha = (Captcha) session.getAttribute(Captcha.NAME);
    String answer = request.getParameter("answer");
    if (captcha.isCorrect(answer)) {
        String username = request.getParameter("username");
        String password = request.getParameter("password");
        try {
            request.login(username, password);
            request.getSession().removeAttribute("errorMessage");
            response.sendRedirect("/beasiswa");
        } catch (ServletException e) {
            request.getSession().setAttribute("errorMessage", "Username dan atau Password Salah");
            response.sendRedirect("/beasiswa/login.jsp");
        }
    } else {
        request.getSession().setAttribute("errorMessage", "text CAPTCHA Salah silahkan coba lagi!");
        response.sendRedirect("/beasiswa/login.jsp");
        return;
    }
}
}

```

Gambar 5. 17 Kode Program proses validasi pengguna

2. Menampilkan pengumuman terbaru

Untuk menampilkan pengumuman terbaru, maka controller akan memanggil method `suratFacade.findAllDesc()`. method ini akan membuat list yang berisi daftar object surat yang diurutkan berdasarkan tanggal surat mulai dari tanggal yang terakhir. List ini kemudian dimasukkan ke dalam atribut surat yang nantinya

akan ditampilkan dengan menggunakan method foreach pada halaman antarmuka. Potongan kode program untuk menampilkan daftar pengumuman terbaru dapat dilihat pada gambar 5.18.

```
if (userPath.equals("/homeAdmin")) {  
    userPath = "/homeAdmin";  
    List surat = suratFacade.findAllDesc();  
    if (surat.size() > 5) {  
        List suratpotong = surat.subList(0, 5);  
        getContext().setAttribute("surat", suratpotong);  
    } else {  
        getContext().setAttribute("surat", surat);  
    }  
}
```

Gambar 5. 18 Kode Program proses menampilkan surat

3. Pembukaan pendaftaran

Untuk melakukan pembukaan pendaftaran terlebih dahulu input dari Admin yang berupa tanggal diparse menjadi data tipe date dengan menggunakan method `format.parse(string)` dengan terlebih dahulu mendefinisikan format tanggal yang sesuai dengan menggunakan `SimpleDateFormat`.

Setelah input dari Admin sesuai dengan tipe data yang dibutuhkan, selanjutnya dibuat object periode. Setelah object periode dibuat kemudian atribut dari object periode tersebut dimasukkan dengan menggunakan method `set`.

Setelah object periode dibuat maka object tersebut dibuat ke dalam entity dengan menggunakan method `periodeFacade.create(newPeriode)`. Gambar 5.19 menunjukkan potongan kode program untuk melakukan pembukaan pendaftaran.

```

} else if (userPath.equals("/pesanPembukaanPendaftaran")) {

    userPath = "/pesanPembukaanPendaftaran";
    int periode = Integer.parseInt(request.getParameter("periode"));
    SimpleDateFormat format = new SimpleDateFormat("dd-MM-yyyy");
    Date pembukaan = null;
    Date penutupan = null;
    try {
        pembukaan = format.parse(request.getParameter("pembukaan"));
        try {
            penutupan = format.parse(request.getParameter("penutupan"));
        } catch (Exception e) {
        }
    } catch (Exception e) {
    }

    Periode newPeriode = new Periode();
    newPeriode.setPeriode(periode);
    newPeriode.setPembukaan(pembukaan);
    newPeriode.setPenutupan(penutupan);
    newPeriode.setStatus(1);

    periodeFacade.create(newPeriode);

    getServletContext().setAttribute("periode", periode);
    getServletContext().setAttribute("pembukaan", pembukaan);
    getServletContext().setAttribute("penutupan", penutupan);
}

```

Gambar 5. 19 Kode Program proses Pembukaan Pendaftaran

4. Rekam, ubah, dan hapus universitas

Untuk fungsionalitas rekam universitas, terlebih dahulu ditampilkan form perekaman universitas, input dari Admin kemudian disimpan dalam atribut. Setelah itu, dilakukan pembuatan object universitas dengan menggunakan method Universitas `newUniversitas = new Universitas()`. Kemudian atribut dimasukkan ke dalam object dengan menggunakan method `set`.

Setelah object selesai dibuat kemudian dibuat ke dalam Entity dengan menggunakan method `universitasFacade.create(newUniversitas)`. Potongan kode program yang menangani proses rekam universitas dapat dilihat pada gambar 5.20

```

} else if (userPath.equals("/pesanRekamUniversitas")) {

    userPath = "/pesanRekamUniversitas";

    String kodeUniversitas = request.getParameter("kodeUniversitas");
    String namaUniversitas = request.getParameter("namaUniversitas");
    String alamat = request.getParameter("alamat");
    String keterangan = request.getParameter("keterangan");

    Universitas newUniversitas = new Universitas();

    newUniversitas.setKodeUniversitas(kodeUniversitas);
    newUniversitas.setNamaUniversitas(namaUniversitas);
    newUniversitas.setAlamat(alamat);
    newUniversitas.setKeterangan(keterangan);

    universitasFacade.create(newUniversitas);

    getServletContext().setAttribute("kodeUniversitas", kodeUniversitas);
    getServletContext().setAttribute("namaUniversitas", namaUniversitas);
    getServletContext().setAttribute("alamat", alamat);
    getServletContext().setAttribute("keterangan", keterangan);
}

```

Gambar 5. 20 Kode Program proses Rekam Universitas

Untuk fungsionalitas ubah universitas langkah yang dilakukan hampir sama dengan rekam universitas, namun object universitas tidak dibuat instance baru namun dicari dengan menggunakan method Universitas univ = universitasFacade.find (kodeUniversitas). Setelah ditemukan universitas tersebut dirubah atributnya sesuai dengan input dari Admin dengan menggunakan method set. Setelah atribut dilakukan perubahan kemudian dilakukan update terhadap object dalam Entity dengan menggunakan method universitasFacade.edit(univ).

Untuk fungsionalitas hapus universitas terlebih dahulu dilakukan pencarian terhadap object universitas yang ingin dihapus dengan menggunakan method Universitas univ = universitasFacade.find (kodeUniv). Setelah itu, dilakukan penghapusan object dari Entity dengan menggunakan method universitasFacade.remove(univ). Gambar 5.21 merupakan potongan kode program yang menangani proses hapus universitas.

```

} else if (userPath.equals("/pesanHapusUniversitas")) {

    userPath = "/pesanHapusUniversitas";
    String kodeUniv = request.getParameter("kodeUniversitas");
    Universitas univ = universitasFacade.find(kodeUniv);

    getServletContext().setAttribute("kodeUniversitas", univ.getKodeUniversitas());
    getServletContext().setAttribute("namaUniversitas", univ.getNamaUniversitas());
    getServletContext().setAttribute("alamat", univ.getAlamat());

    universitasFacade.remove(univ);
}

```

Gambar 5. 21 Kode Program proses Hapus Universitas

5.6. Uji Coba

Uji coba yang dilakukan pada aplikasi ini terbatas pada uji coba fungsionalitas mengingat aplikasi yang bersifat prototype dan masih berjalan dalam lingkungan localhost. Uji coba fungsional dilakukan dengan mengacu pada test case yang telah dibuat pada tahap desain dan melihat apakah aplikasi yang dikembangkan ini telah memenuhi test case.

Pada tabel 5.4 dan 5.5 ditampilkan daftar test case beserta hasil uji coba yang telah dilakukan. Untuk detail dari setiap test case dapat dilihat pada lampiran D.

Hasil uji coba fungsional menunjukkan bahwa aplikasi ini telah memenuhi test case yang telah dibuat pada tahap desain. Hal ini berarti bahwa secara fungsional aplikasi SIPERBEN telah berjalan dengan baik.

Tabel 5. 4 Hasil Uji Coba Fungsionalitas

No.	Test Case	Status
1	Login (UC01)	Terpenuhi
2	Logout (UC02)	Terpenuhi
3	Meminta Reset Password (UC03)	Terpenuhi
4	Melihat Pengumuman (UC04)	Terpenuhi
5	Rekam User (UC05)	Terpenuhi
6	Ubah User (UC06)	Terpenuhi
7	Hapus User (UC07)	Terpenuhi
8	Reset Password User (UC08)	Terpenuhi
9	Rekam Universitas (UC09)	Terpenuhi

Tabel 5. 5 Hasil Uji Coba Fungsionalitas (lanjutan)

No.	Test Case	Status
10	Ubah Universitas (UC10)	Terpenuhi
11	Hapus Universitas (UC11)	Terpenuhi
12	Rekam Jurusan (UC12)	Terpenuhi
13	Ubah Jurusan (UC13)	Terpenuhi
14	Hapus Jurusan (UC14)	Terpenuhi
15	Membuka Pendaftaran (UC15)	Terpenuhi
16	Tambah Detail Pembukaan Pendaftaran (UC16)	Terpenuhi
17	Hapus Detail Pembukaan Pendaftaran (UC17)	Terpenuhi
18	Rekam Surat Pembukaan Pendaftaran (UC18)	Terpenuhi
19	Rekam Surat Panggilan Tes Tertulis (UC19)	Terpenuhi
20	Rekam Hasil Tes Tertulis (UC20)	Terpenuhi
21	Rekam Surat Panggilan Tes Wawancara (UC21)	Terpenuhi
22	Rekam Hasil Tes Wawancara (UC22)	Terpenuhi
23	Rekam Surat Pengumuman (UC23)	Terpenuhi
24	Mendaftar (UC24)	Terpenuhi
25	Membatalkan Pendaftaran (UC25)	Terpenuhi
26	Melihat Riwayat Pendaftaran (UC26)	Terpenuhi
27	Rekam Permintaan Rekomendasi (UC27)	Terpenuhi
28	Ubah Permintaan Rekomendasi (UC28)	Terpenuhi
29	Hapus Permintaan Rekomendasi (UC29)	Terpenuhi
30	Melihat Permintaan Rekomendasi (UC30)	Terpenuhi
31	Rekam Rekomendasi (UC31)	Terpenuhi
32	Ubah Rekomendasi (UC32)	Terpenuhi
33	Hapus Rekomendasi (UC33)	Terpenuhi
34	Tolak Rekomendasi (UC34)	Terpenuhi

BAB VI

KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dari seluruh proses pengerjaan tugas akhir beserta saran untuk pengembangan selanjutnya.

6.1. Kesimpulan

Setelah melewati proses pengerjaan aplikasi SIPERBEN, penulis dapat mengambil kesimpulan antara lain :

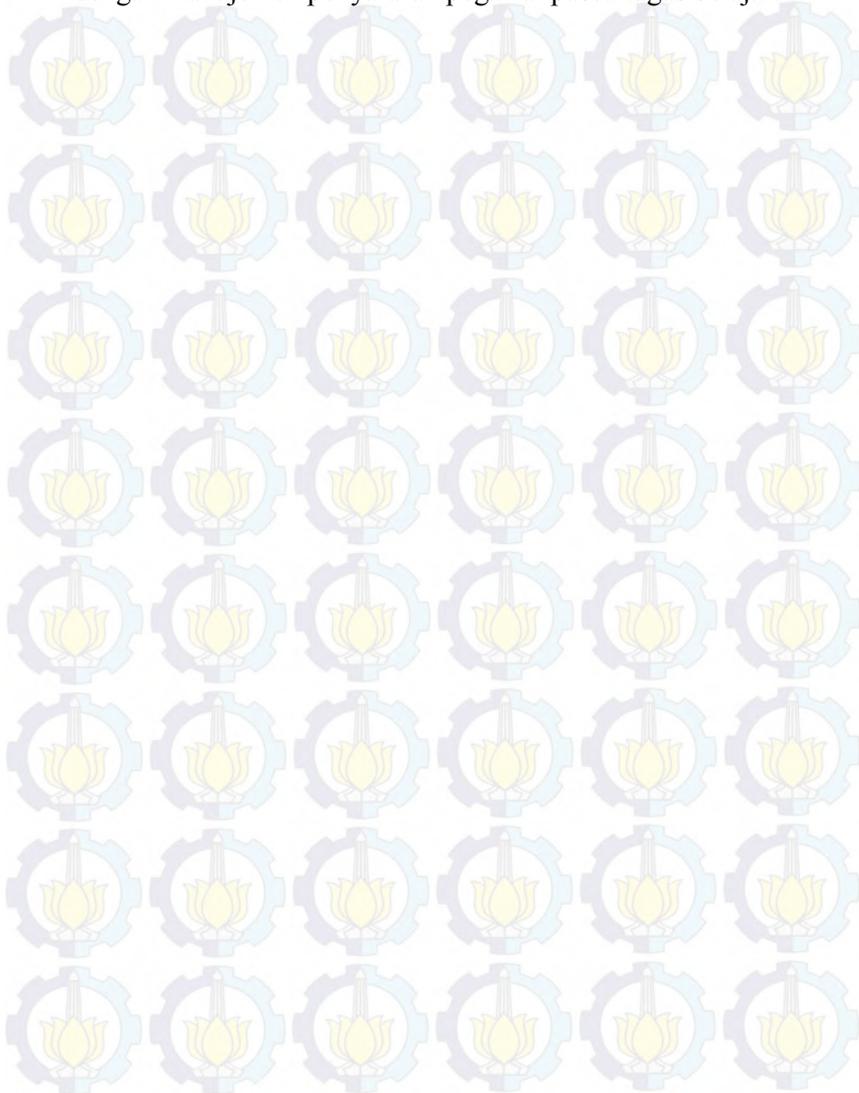
1. Teknologi Java Server Pages dan MySQL dapat diimplementasikan dengan baik dalam pengembangan aplikasi SIPERBEN.
2. Aplikasi SIPERBEN telah berhasil dibuat dan dapat berjalan dengan baik, seluruh test case yang telah dibuat pada tahap desain dapat dijalankan.
3. Dengan adanya aplikasi SIPERBEN, otomasi proses seleksi administrasi dapat dijalankan.

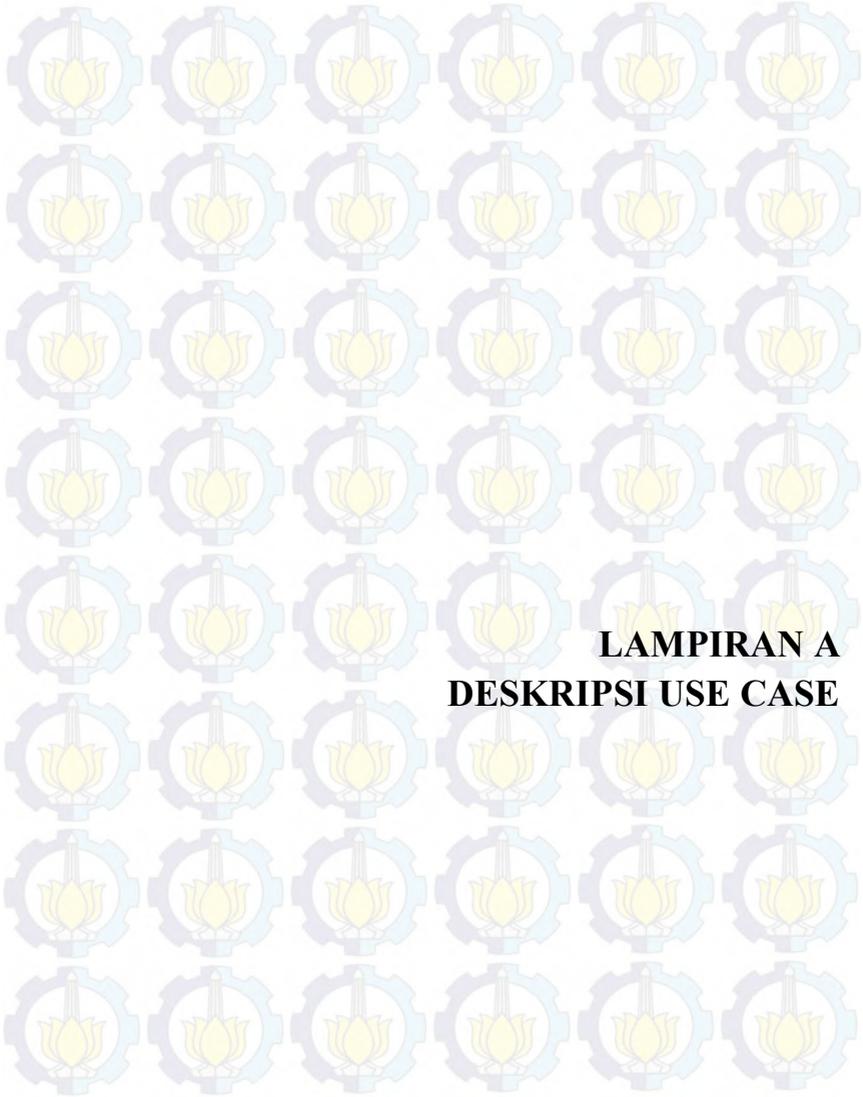
6.2. Saran

Beberapa saran dari penulis terkait pengembangan dari Sistem Informasi yang telah dibuat dalam Tugas Akhir ini adalah sebagai berikut :

1. Integrasi dengan aplikasi persuratan yang akan memungkinkan proses penomoran surat yang baku dan dilakukan secara otomatis.
2. kemungkinan penambahan fitur tes tertulis yang dilakukan secara online, sehingga hasil tes tertulis dapat diketahui secara langsung.
3. Integrasi dengan sistem informasi beasiswa yang telah dibuat sehingga menjadi sistem informasi yang mencakup seluruh proses terkait beasiswa internal mulai dari seleksi

penerimaan, pengelolaan administrasi kemahasiswaan, sampai dengan manajemen penyaluran pegawai pasca tugas belajar.





LAMPIRAN A
DESKRIPSI USE CASE

A.1. Deskripsi Use Case Login

Usecase	:	Login (UC01)
Aktor	:	Pengguna
Pre-condition	:	Pengguna berada pada halaman utama
Trigger	:	Pengguna menklik menu LOGIN
Basic Course	:	System menampilkan form username dan password. Pengguna memasukkan username, password, dan teks CAPTCHA kemudian menekan tombol LOGIN. System memastikan field username dan password telah diisi, memeriksa teks CAPTCHA, dan mencocokkan username dan password dengan data dalam database.
Post-condition	:	System menampilkan halaman utama sesuai dengan level pengguna.
Alternate Course	:	<p>Pengguna menekan tombol BATAL : System menampilkan halaman utama</p> <p>Field username dan atau password belum diisi : System menampilkan pesan bahwa username dan password harus diisi.</p> <p>Teks CAPTCHA salah : System menampilkan pesan bahwa teks CAPTCHA yang dimasukkan salah.</p> <p>Username dan atau password tidak cocok dengan database : System menampilkan pesan bahwa username dan atau password salah</p>

A.2. Deskripsi Use Case Logout

Usecase	:	Logout (UC02)
Aktor	:	Pengguna
Pre-condition	:	Pengguna berada pada halaman utama.
Trigger	:	Pengguna menekan tombol Logout.
Basic Course	:	System menampilkan konfirmasi logout. Pengguna menekan tombol YA.
Post-condition	:	System menampilkan halaman utama.
Alternate Course	:	User menekan tombol TIDAK : System menampilkan halaman utama pengguna

A.3. Deskripsi Use Case Meminta Reset Password

Usecase	:	Meminta Reset Password (UC03)
Aktor	:	Pegawai, Atasan Langsung
Pre-condition	:	Pengguna berada pada form username dan password
Trigger	:	Pengguna menekan link lupa password
Basic Course	:	System menampilkan form permintaan reset password. Pengguna mengisi field yang disediakan kemudian menekan tombol KIRIM.
Post-condition	:	System menyimpan permintaan dalam database dan menampilkan halaman utama
Alternate Course	:	User menekan tombol BATAL : System menampilkan halaman utama

A.4. Deskripsi Use Case Melihat Pengumuman

Usecase	:	Melihat Pengumuman (UC04)
Aktor	:	Pengguna
Pre-condition	:	--

Trigger	:	Pengguna mengakses halaman utama.
Basic Course	:	System menampilkan daftar pengumuman terbaru pada halaman utama. Pengguna menekan link DETAIL pada surat yang ingin dilihat.
Post-condition	:	System menampilkan surat yang dipilih.
Alternate Course	:	

A.5. Deskripsi Use Case Rekam User

Usecase	:	Rekam User (UC05)
Aktor	:	Admin
Pre-condition	:	Admin berada pada halaman user
Trigger	:	Admin menekan tombol REKAM.
Basic Course	:	System menampilkan form rekam user. Admin mengisi field yang diperlukan kemudian menekan tombol SIMPAN. System memastikan semua field terisi
Post-condition	:	System menyimpan data user kedalam database dan menampilkan pesan bahwa user telah ditambahkan.
Alternate Course	:	Admin menekan tombol BATAL: System menampilkan halaman user. Terdapat field yang belum diisi pada saat Admin menekan tombol SIMPAN : System menampilkan pesan terdapat field yang belum diisi.

A.6. Deskripsi Use Case Ubah User

Usecase	:	Ubah User (UC06)
Aktor	:	Admin
Pre-condition	:	Admin berada pada halaman user

Trigger	:	Admin menekan link UBAH.
Basic Course	:	System menampilkan form perubahan data user. Admin melakukan perubahan terhadap field yang diperlukan kemudian menekan tombol SIMPAN. System memastikan semua field telah di isi
Post-condition	:	System menyimpan perubahan data user dan menampilkan pesan bahwa data user telah diubah.
Alternate Course	:	Admin menekan tombol BATAL: System menampilkan halaman user. terdapat field yang kosong saat menekan tombol SIMPAN : System menampilkan pesan bahwa terdapat field yang kosong

A.7. Deskripsi Use Case Hapus User

Usecase	:	Hapus User (UC07)
Aktor	:	Admin
Pre-condition	:	Admin berada pada halaman user
Trigger	:	Admin menekan link HAPUS.
Basic Course	:	System menampilkan konfirmasi penghapusan user. Admin menekan tombol YA.
Post-condition	:	System menghapus user dimaksud kemudian menampilkan pesan bahwa user telah dihapus
Alternate Course	:	Admin menekan tombol TIDAK: System menampilkan halaman user.

A.8. Deskripsi Use Case Reset Password User

Usecase	:	Reset Password User (UC08)
Aktor	:	Admin
Pre-condition	:	Admin berada pada halaman user
Trigger	:	Admin menekan tombol PERMINTAAN RESET PASSWORD.
Basic Course	:	System menampilkan daftar permintaan reset password. Admin menekan link RESET. System menampilkan konfirmasi apakah akan mereset password. Admin menekan tombol YA.
Post-condition	:	System mengubah password menjadi default (nama ibu kandung) kemudian menampilkan daftar perminataan reset password
Alternate Course	:	Admin menekan tombol TIDAK: System menampilkan daftar permintaan reset password.

A.9. Deskripsi Use Case Rekam Universitas

Usecase	:	Rekam Universitas (UC09)
Aktor	:	Admin
Pre-condition	:	Admin berada pada halaman universitas
Trigger	:	Admin menekan tombol REKAM.
Basic Course	:	System menampilkan form rekam universitas. Admin mengisi field yang diperlukan kemudian menekan tombol SIMPAN. System memastikan semua field terisi
Post-condition	:	System menyimpan data universitas ke dalam database dan menampilkan pesan bahwa universitas telah ditambahkan.
Alternate Course	:	Admin menekan tombol BATAL: System menampilkan halaman universitas.

	Terdapat field yang belum diisi pada saat Admin menekan tombol SIMPAN : System menampilkan pesan terdapat field yang belum diisi.
--	--

A.10. Deskripsi Use Case Ubah Universitas

Usecase	:	Ubah Universitas (UC10)
Aktor	:	Admin
Pre-condition	:	Admin berada pada halaman universitas
Trigger	:	Admin menekan link UBAH.
Basic Course	:	System menampilkan form perubahan data universitas. Admin melakukan perubahan terhadap field yang diperlukan kemudian menekan tombol SIMPAN. System memastikan semua field telah terisi
Post-condition	:	System menyimpan perubahan data dan menampilkan pesan bahwa data universitas telah diubah.
Alternate Course	:	Admin menekan tombol BATAL: System menampilkan halaman universitas.

A.11. Deskripsi Use Case Hapus Universitas

Usecase	:	Hapus Universitas (UC11)
Aktor	:	Admin
Pre-condition	:	Admin berada pada halaman universitas
Trigger	:	Admin menekan link HAPUS.
Basic Course	:	System menampilkan konfirmasi penghapusan universitas. Admin menekan tombol YA.
Post-condition	:	System menghapus data universitas kemudian menampilkan pesan bahwa universitas telah dihapus
Alternate	:	Admin menekan tombol TIDAK:

Course	System menampilkan halaman universitas.
--------	---

A.12. Deskripsi Use Case Rekam Jurusan

Usecase	:	Rekam Jurusan (UC12)
Aktor	:	Admin
Pre-condition	:	Admin berada pada halaman jurusan kemudian memilih universitas
Trigger	:	Admin menekan tombol REKAM.
Basic Course	:	System menampilkan form rekam jurusan. Admin mengisi field yang diperlukan kemudian menekan tombol SIMPAN. System memastikan semua field telah terisi
Post-condition	:	System menyimpan data jurusan dan menampilkan pesan bahwa jurusan telah ditambahkan.
Alternate Course	:	Admin belum memilih universitas : System menampilkan pesan bahwa universitas belum dipilih Admin menekan tombol BATAL: System menampilkan halaman jurusan. Terdapat field yang belum diisi pada saat Admin menekan tombol SIMPAN : System menampilkan pesan terdapat field yang belum diisi.

A.13. Deskripsi Use Case Ubah Jurusan

Usecase	:	Ubah Jurusan (UC13)
Aktor	:	Admin
Pre-condition	:	Admin berada pada halaman Jurusan kemudian memilih universitas.
Trigger	:	Admin menekan link UBAH.
Basic Course	:	System menampilkan form perubahan data

		jurusan. Admin melakukan perubahan terhadap field yang diperlukan kemudian menekan tombol SIMPAN. System memastikan semua field telah terisi.
Post-condition	:	System menyimpan perubahan data dan menampilkan pesan bahwa data jurusan telah diubah
Alternate Course	:	Admin menekan tombol BATAL: System menampilkan halaman jurusan. Terdapat field yang belum diisi pada saat Admin menekan tombol SIMPAN : System menampilkan pesan terdapat field yang belum diisi.

A.14. Deskripsi Use Case Hapus Jurusan

Usecase	:	Hapus Jurusan (UC14)
Aktor	:	Admin
Pre-condition	:	Admin berada pada halaman jurusan, memilih universitas
Trigger		Admin menekan link HAPUS.
Basic Course	:	System menampilkan konfirmasi penghapusan jurusan. Admin menekan tombol YA.
Post-condition	:	System menghapus data jurusan kemudian menampilkan pesan bahwa jurusan telah dihapus.
Alternate Course	:	Admin menekan tombol TIDAK: System menampilkan halaman jurusan.

A.15. Deskripsi Use Case Membuka Pendaftaran

Usecase	:	Membuka Pendaftaran (UC15)
Aktor	:	Admin

Pre-condition	:	Admin berada pada halaman Seleksi Administrasi
Trigger	:	Admin menekan tombol PEMBUKAAN PENDAFTARAN.
Basic Course	:	System menampilkan form pembukaan pendaftaran. Admin mengisi field yang disediakan kemudian menekan tombol SIMPAN. System memastikan periode yang diisikan Admin belum pernah dilakukan pembukaan pendaftaran, memastikan semua field telah terisi
Post-condition	:	System menyimpan data dalam database dan menampilkan pesan bahwa pendaftaran telah dibuka.
Alternate Course	:	Admin menekan tombol BATAL: System menampilkan halaman seleksi administrasi. Periode yang diisikan Admin sudah pernah dilakukan pembukaan pendaftaran : System menampilkan pesan bahwa periode yang diisikan sudah dibuka. Terdapat field yang belum diisi : System menampilkan pesan bahwa semua field harus terisi.

A.16. Deskripsi Use Case Tambah Detail Pembukaan Pendaftaran

Usecase	:	Tambah Detail Pembukaan Pendaftaran (UC16)
Aktor	:	Admin
Pre-condition	:	Admin berada pada halaman Seleksi Administrasi

Trigger	:	Admin menekan link DETAIL
Basic Course	:	System menampilkan daftar detail pembukaan pendaftaran. Admin menekan tombol TAMBAH. System menampilkan form pembukaan pendaftaran. Admin mengisi field yang disediakan kemudian menekan tombol SIMPAN. System memastikan jurusan belum pernah ditambahkan pada periode tersebut, memastikan semua field telah terisi
Post-condition	:	System menyimpan data ke dalam database dan menampilkan pesan detail telah ditambahkan.
Alternate Course	:	Admin menekan tombol BATAL: System menampilkan halaman seleksi administrasi Jurusan sudah pernah ditambahkan : System menampilkan pesan bahwa jurusan sudah dibuka Terdapat field yang kosong : System menampilkan pesan bahwa semua field harus terisi.

A.17. Deskripsi Use Case Hapus Detail Pembukaan Pendaftaran

Usecase	:	Hapus Detail Pembukaan Pendaftaran (UC17)
Aktor	:	Admin
Pre-condition	:	Admin berada pada halaman Seleksi Administrasi
Trigger	:	Admin menekan link DETAIL
Basic Course	:	System menampilkan daftar detail

		pembukaan pendaftaran. Admin menekan link HAPUS. System menampilkan konfirmasi penghapusan. Admin menekan tombol YA.
Post-condition		System menghapus data dari database kemudian menampilkan pesan detail telah dihapus.
Alternate Course	:	Admin menekan tombol TIDAK: System menampilkan halaman seleksi administrasi.

A.18. Deskripsi Use Case Rekam Surat Pembukaan Pendaftaran

Usecase	:	Rekam Surat Pembukaan Pendaftaran (UC18)
Aktor	:	Admin
Pre-condition	:	Admin berada pada halaman seleksi administrasi
Trigger	:	Admin menekan tombol SURAT PEMBUKAAN PENDAFTARAN.
Basic Course	:	System menampilkan halaman surat pembukaan pendaftaran. Admin menekan tombol REKAM SURAT. System menampilkan form Rekam Surat Pembukaan Pendaftaran. Admin memasukkan tahun/periode, tanggal surat, nomor surat, jenis surat, dan pejabat penandatanganan surat kemudian menekan tombol KIRIM. System menampilkan preview surat pembukaan pendaftaran. Admin menekan tombol SIMPAN. System memastikan periode/tahun belum pernah dilakukan perekaman
Post-condition	:	System menyimpan data surat pembukaan pendaftaran tersebut dalam database dan

		menampilkan pesan surat telah tersimpan.
Alternate Course	:	Tahun/periode sudah pernah dilakukan perekaman : System menampilkan pesan bahwa periode yang dimasukkan telah dilakukan perekaman surat. Admin menekan tombol BATAL : System menampilkan halaman seleksi administrasi

A.19. Deskripsi Use Case Rekam Surat Panggilan Tes Tertulis

Usecase	:	Rekam Surat Panggilan Tes Tertulis (UC19)
Aktor	:	Admin
Pre-condition	:	Admin berada pada halaman tes tertulis
Trigger	:	Admin menekan tombol REKAM SURAT PANGGILAN.
Basic Course	:	System menampilkan form Surat Panggilan Tes Tertulis. Admin memasukkan tahun/periode tes, tanggal surat, nomor surat, dan pejabat penandatanganan surat kemudian menekan tombol KIRIM. System memuat daftar pegawai yang lolos seleksi administratif berdasarkan rule yang telah ditentukan kemudian menampilkan preview surat panggilan tes tertulis. Admin menekan tombol SIMPAN.
Post-condition	:	System menyimpan surat panggilan tes tertulis tersebut dalam database dan menampilkan pesan bahwa surat telah tersimpan.
Alternate Course	:	Tahun/periode tes sudah pernah dilakukan perekaman :

	System menampilkan pesan bahwa periode yang dimasukkan telah dilakukan perekaman surat. Admin menekan tombol BATAL : System menampilkan halaman tes tertulis
--	--

A.20. Deskripsi Use Case Rekam Hasil Tes Tertulis

Usecase	:	Rekam Hasil Tes Tertulis (UC20)
Aktor	:	Admin
Pre-condition	:	Admin berada pada halaman tes tertulis
Trigger	:	Admin menekan tombol REKAM HASIL TES.
Basic Course	:	System menampilkan form rekam hasil tes. Admin mengisi periode/ tahun. Sistem menampilkan list pegawai yang menjadi peserta tes periode tersebut. Admin memasukkan hasil tes tertulis yaitu Lulus atau Tidak kemudian menekan tombol SIMPAN.
Post-condition	:	System menyimpan hasil tes tertulis kedalam database kemudian menampilkan pesan bahwa hasil tes tertulis telah disimpan.
Alternate Course	:	list pegawai tidak ditemukan dalam database : system menampilkan pesan bahwa daftar peserta tes untuk periode yang dimasukkan tidak ditemukan. Admin menekan tombol BATAL : System menampilkan halaman tes tertulis

A.21. Deskripsi Use Case Rekam Surat Panggilan Tes Wawancara

Usecase	:	Rekam Surat Panggilan Tes Wawancara (UC21)
Aktor	:	Admin
Pre-condition	:	Admin berada pada halaman tes wawancara
Trigger	:	Admin menekan tombol REKAM SURAT PANGGILAN.
Basic Course	:	System menampilkan form Surat Panggilan Tes Wawancara. Admin memasukkan tahun/periode tes, tanggal surat, dan pejabat penandatanganan surat kemudian menekan tombol KIRIM. System memuat daftar pegawai yang lolos tes tertulis, kemudian menampilkan preview surat panggilan tes wawancara. Admin menekan tombol SIMPAN.
Post-condition	:	System menyimpan surat panggilan tes wawancara tersebut dalam database dan menampilkan pesan bahwa surat telah tersimpan.
Alternate Course	:	Tahun/periode tes sudah pernah dilakukan perekaman : System menampilkan pesan bahwa periode yang dimasukkan telah dilakukan perekaman surat. Admin menekan tombol BATAL : System menampilkan halaman tes wawancara

A.22. Deskripsi Use Case Rekam Hasil Tes Wawancara

Usecase	:	Rekam Hasil Tes Wawancara (UC22)
Aktor	:	Admin
Pre-condition	:	Admin berada pada halaman tes wawancara
Trigger	:	Admin menekan tombol REKAM HASIL TES.
Basic Course	:	System menampilkan form rekam hasil tes. Admin mengisi periode/ tahun. Sistem menampilkan list pegawai yang menjadi peserta tes. Admin memasukkan hasil tes wawancara yaitu Lulus atau Tidak kemudian menekan tombol SIMPAN.
Post-condition	:	System menyimpan hasil tes tertulis kedalam database kemudian menampilkan pesan bahwa hasil tes tertulis telah disimpan.
Alternate Course	:	list pegawai tidak ditemukan dalam database : system menampilkan pesan bahwa daftar peserta tes untuk periode yang dimasukkan tidak ditemukan. Admin menekan tombol BATAL : System menampilkan halaman tes wawancara

A.23. Deskripsi Use Case Rekam Surat Pengumuman

Usecase	:	Rekam Surat Pengumuman (UC23)
Aktor	:	Admin
Pre-condition	:	Admin berada pada halaman pengumuman
Trigger	:	Admin menekan tombol REKAM SURAT PENGUMUMAN.
Basic Course	:	System menampilkan Form isian Surat Pengumuman. Admin memasukkan

		periode/tahun, tanggal surat, nomor surat, dan pejabat penandatanganan surat kemudian menekan tombol KIRIM. System memuat daftar pegawai yang lulus tes wawancara kemudian menampilkan preview surat pengumuman. Admin menekan tombol SIMPAN.
Post-condition	:	System menyimpan surat pengumuman dalam database dan menampilkan pesan bahwa surat pengumuman telah disimpan.
Alternate Course	:	Tahun/periode sudah pernah dilakukan perekaman : System menampilkan pesan bahwa periode yang dimasukkan telah dilakukan perekaman pengumuman. Admin menekan tombol BATAL : System menampilkan halaman pengumuman

A.24. Deskripsi Use Case Mendaftar

Usecase	:	Mendaftar (UC24)
Aktor	:	Pegawai
Pre-condition	:	Pegawai berada pada halaman pendaftaran, memilih periode/tahun, universitas, dan jurusan yang diminati
Trigger	:	Pegawai menekan tombol DAFTAR.
Basic Course	:	System memastikan pegawai belum terdaftar, memastikan tanggal pendaftaran tidak melewati batas waktu, memastikan pegawai telah mendapatkan rekomendasi dari atasan, memuat data dan memeriksa administrasi pegawai, menandai flag terdaftar pada data pegawai,.
Post-condition	:	System menambahkan pegawai kedalam

		daftar dalam database, kemudian menampilkan pesan bahwa pendaftaran telah berhasil dilakukan
Alternate Course	:	<p>Pegawai telah melakukan pendaftaran : System menampilkan pesan bahwa pegawai tersebut telah terdaftar.</p> <p>melewati batas tanggal pendaftaran : System menampilkan pesan bahwa pendaftaran sudah melewati batas waktu.</p> <p>Pegawai belum mendapatkan rekomendasi : System menampilkan pesan bahwa pegawai belum mendapatkan rekomendasi dari atasan.</p> <p>Pegawai tidak lolos administrasi : System menampilkan pesan bahwa pegawai tidak memenuhi syarat administrasi yang ditentukan</p>

A.25. Deskripsi Use Case Membatalkan Pendaftaran

Usecase	:	Membatalkan Pendaftaran (UC25)
Aktor	:	Pegawai
Pre-condition	:	Pegawai berada pada halaman pendaftaran
Trigger	:	Pegawai menekan tombol PEMBATALAN.
Basic Course	:	System memastikan bahwa pegawai telah terdaftar kemudian menampilkan konfirmasi pembatalan pendaftaran. Pegawai menekan tombol YA. System mengubah status pegawai.
Post-condition	:	System menghapus pegawai dari daftar pegawai, kemudian menampilkan pesan bahwa pendaftaran telah dibatalkan.

Alternate Course	:	Pegawai belum mendaftar : System menampilkan pesan bahwa pegawai belum melakukan pendaftaran. Pegawai menekan tombol BATAL : System menampilkan halaman pendaftaran
------------------	---	--

A.26. Deskripsi Use Case Melihat Riwayat Pendaftaran

Usecase	:	Melihat Riwayat Pendaftaran (UC26)
Aktor	:	Pegawai
Pre-condition	:	Pegawai berada pada halaman pendaftaran
Trigger	:	Pegawai menekan tombol RIWAYAT PENDAFTARAN.
Basic Course	:	System memuat data riwayat pendaftaran dari daftar pegawai
Post-condition	:	System menampilkan halaman riwayat pendaftaran.
Alternate Course	:	Data tidak ditemukan : System menampilkan pesan bahwa riwayat pendaftaran tidak ditemukan

A.27. Deskripsi Use Case Rekam Permintaan Rekomendasi

Usecase	:	Rekam Permintaan Rekomendasi (UC27)
Aktor	:	Pegawai
Pre-condition	:	Pegawai berada pada halaman permintaan rekomendasi
Trigger	:	Pegawai menekan tombol REKAM.
Basic Course	:	System menampilkan form permintaan rekomendasi. Pegawai memasukkan periode dan NIP atasannya kemudian menekan tombol KIRIM. System memastikan permintaan rekomendasi pada periode

		tersebut belum pernah dilakukan
Post-condition	:	System menyimpan permintaan rekomendasi dalam database dan menampilkan pesan permintaan rekomendasi telah dikirim.
Alternate Course	:	Pegawai menekan tombol BATAL : System menampilkan halaman permintaan rekomendasi Permintaan Rekomendasi sudah pernah dilakukan : System menampilkan pesan bahwa permintaan rekomendasi telah dilakukan

A.28. Deskripsi Use Case Ubah Permintaan Rekomendasi

Usecase	:	Ubah Permintaan Rekomendasi (UC28)
Aktor	:	Pegawai
Pre-condition	:	Pegawai berada pada halaman permintaan rekomendasi
Trigger	:	Pegawai menekan link UBAH.
Basic Course	:	System memastikan rekomendasi belum dilakukan perekaman oleh atasan kemudian menampilkan form permintaan rekomendasi. Pegawai melakukan perubahan yang diperlukan kemudian menekan tombol KIRIM.
Post-condition	:	System menyimpan perubahan permintaan rekomendasi dalam database dan menampilkan pesan perubahan telah disimpan.
Alternate Course	:	Pegawai menekan tombol BATAL : System menampilkan halaman permintaan rekomendasi

	Rekomendasi telah dilakukan perekaman oleh atasan : System menampilkan pesan bahwa rekomendasi telah direkam oleh atasan.
--	--

A.29. Deskripsi Use Case Hapus Permintaan Rekomendasi

Usecase	:	Hapus Permintaan Rekomendasi (UC29)
Aktor	:	Pegawai
Pre-condition	:	Pegawai berada pada halaman permintaan rekomendasi
Trigger	:	Pegawai menekan link HAPUS.
Basic Course	:	System memastikan rekomendasi belum dilakukan perekaman oleh atasan kemudian menampilkan konfirmasi penghapusan permintaan rekomendasi. Pegawai menekan tombol YA.
Post-condition	:	System menghapus data permintaan rekomendasi dalam database dan menampilkan pesan permintaan rekomendasi telah dihapus.
Alternate Course	:	Pegawai menekan tombol BATAL : System menampilkan halaman permintaan rekomendasi Rekomendasi telah dilakukan perekaman oleh atasan : System menampilkan pesan bahwa rekomendasi telah direkam oleh atasan.

A.30. Deskripsi Use Case Melihat Permintaan Rekomendasi

Usecase	:	Melihat Permintaan Rekomendasi (UC30)
Aktor	:	Atasan Langsung
Pre-condition	:	Atasan Langsung berada pada halaman atasan langsung
Trigger	:	Atasan memilih menu REKOMENDASI.
Basic Course	:	System memuat daftar pegawai yang meminta rekomendasi dari atasan tersebut
Post-condition	:	System menampilkan daftar permintaan rekomendasi pada halaman rekomendasi
Alternate Course	:	Tidak ada yang meminta rekomendasi dari atasan tersebut : System menampilkan pesan bahwa tidak ada pegawai yang meminta rekomendasi dari atasan tersebut.

A.31. Deskripsi Use Case Rekam Rekomendasi

Usecase	:	Rekam Rekomendasi (UC31)
Aktor	:	Atasan Langsung
Pre-condition	:	Atasan Langsung berada pada halaman rekomendasi
Trigger	:	Atasan menekan link REKAM
Basic Course	:	untuk pegawai yang akan diberi rekomendasi. System menampilkan form rekomendasi. Atasan Langsung mengisi field yang diperlukan kemudian menekan tombol SIMPAN. System memastikan semua field terisi
Post-condition	:	System menyimpan data dalam database, kemudian menampilkan pesan bahwa rekomendasi telah tersimpan.
Alternate	:	Atasan Langsung menekan tombol BATAL:

Course	System menampilkan halaman rekomendasi terdapat field yang belum diisi : System menampilkan pesan bahwa semua field harus diisi.
--------	--

A.32. Deskripsi Use Case Ubah Rekomendasi

Usecase	: Ubah Rekomendasi (UC32)
Aktor	: Atasan Langsung
Pre-condition	: Atasan Langsung berada pada halaman rekomendasi
Trigger	: Atasan menekan link UBAH.
Basic Course	: System memastikan bahwa pegawai yang dipilih telah dilakukan perekaman rekomendasi kemudian menampilkan form rekomendasi. Atasan Langsung melakukan perubahan yang diperlukan kemudian menekan tombol SIMPAN. System memastikan semua field telah terisi
Post-condition	: System menyimpan perubahan ke dalam database dan menampilkan pesan bahwa perubahan telah tersimpan.
Alternate Course	: Pegawai yang dipilih belum dilakukan perekaman rekomendasi : System menampilkan pesan bahwa pegawai tersebut belum dilakukan perekaman rekomendasi. Atasan Langsung menekan tombol BATAL : System menampilkan halaman rekomendasi. terdapat field yang kosong : System menampilkan pesan bahwa semua field harus diisi.

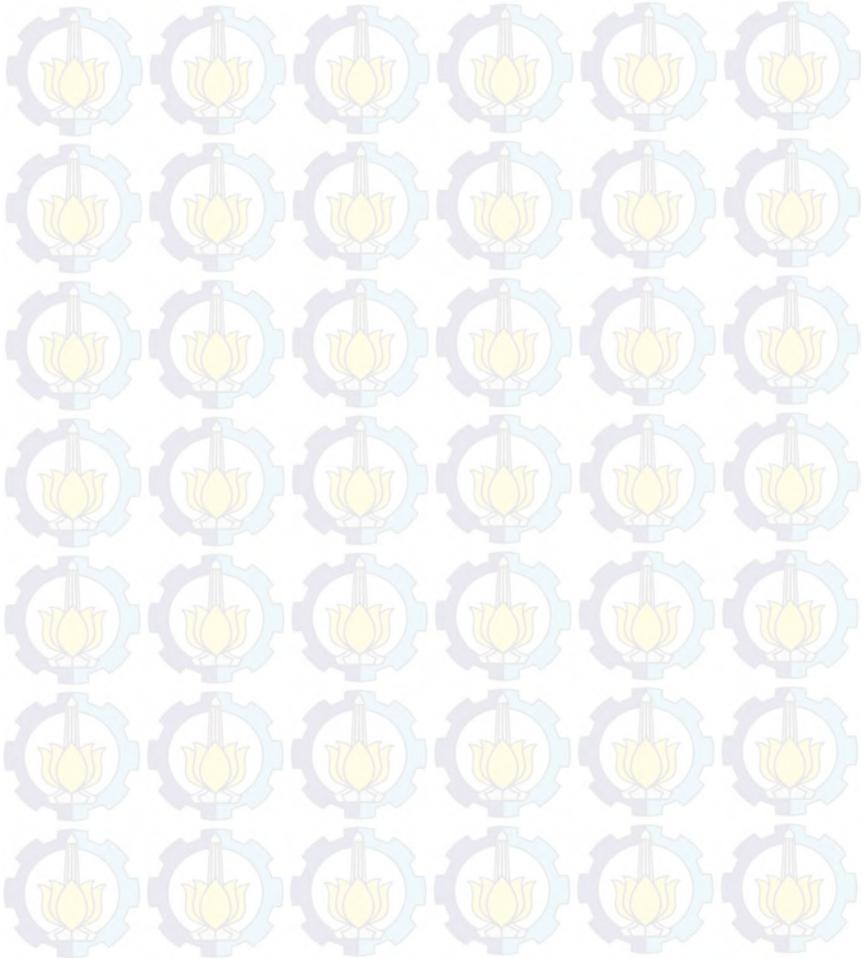
A.33. Deskripsi Use Case Hapus Rekomendasi

Usecase	:	Hapus Rekomendasi (UC33)
Aktor	:	Atasan Langsung
Pre-condition	:	Atasan langsung berada pada halaman rekomendasi
Trigger	:	Atasan menekan link HAPUS.
Basic Course	:	System memastikan bahwa pegawai yang dipilih telah dilakukan perekaman rekomendasi kemudian menampilkan konfirmasi penghapusan rekomendasi. Atasan Langsung menekan tombol YA.
Post-condition	:	System menghapus rekomendasi dari database dan menampilkan pesan rekomendasi telah terhapus.
Alternate Course	:	Pegawai yang dipilih belum dilakukan perekaman rekomendasi : System menampilkan pesan bahwa pegawai tersebut belum dilakukan perekaman rekomendasi. Atasan Langsung menekan tombol TIDAK : System menampilkan halaman rekomendasi.

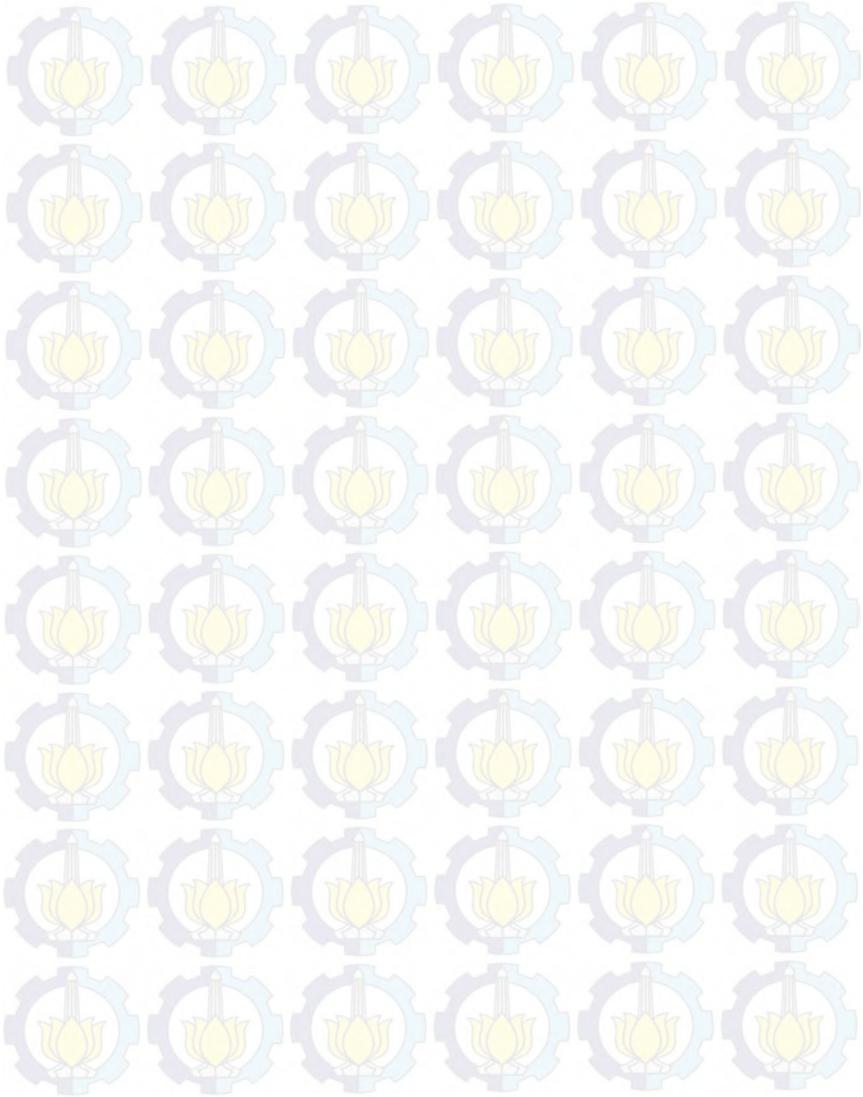
A.34. Deskripsi Use Case Tolak Rekomendasi

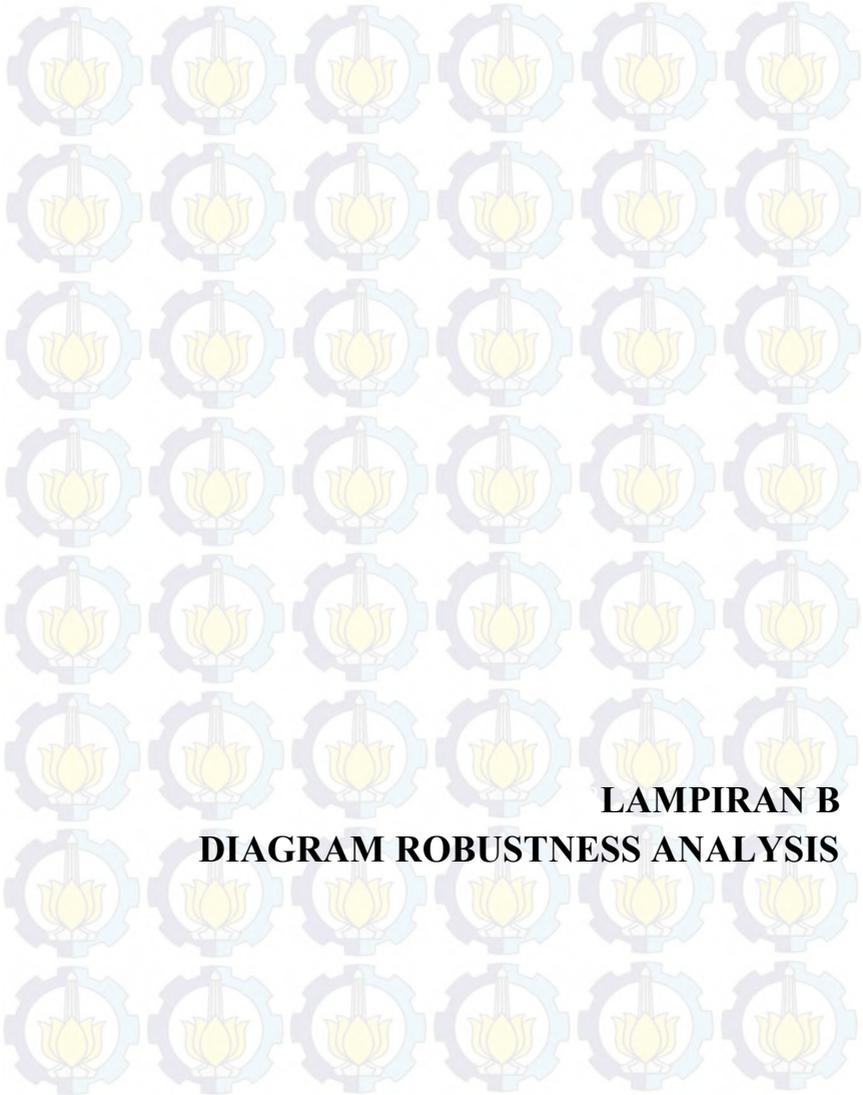
Usecase	:	Tolak Rekomendasi (UC34)
Aktor	:	Atasan Langsung
Pre-condition	:	Atasan langsung berada pada halaman rekomendasi
Trigger	:	Atasan menekan link TOLAK.
Basic Course	:	System menampilkan konfirmasi penolakan rekomendasi. Atasan Langsung menekan tombol YA. System menandai permintaan

		rekomendasi dalam database
Post-condition	:	System menampilkan pesan permintaan rekomendasi telah ditolak
Alternate Course	:	Atasan Langsung menekan tombol TIDAK : System menampilkan halaman rekomendasi.



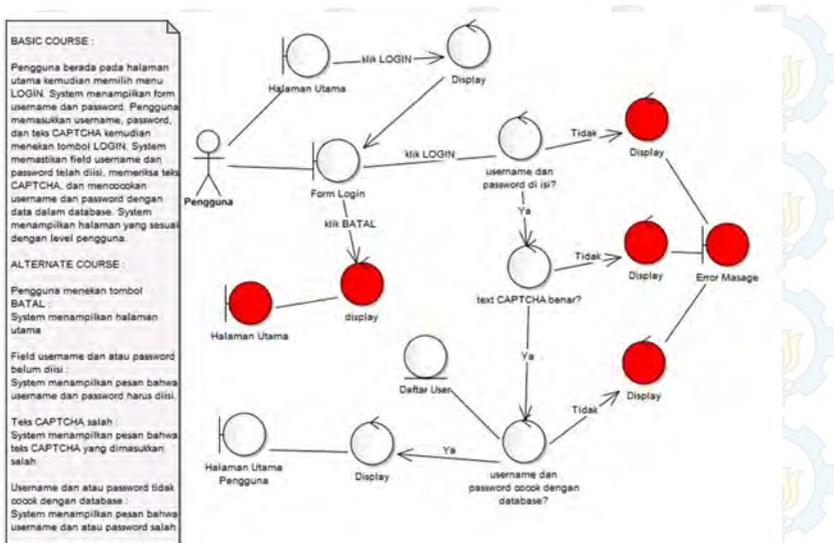
Halaman ini sengaja dikosongkan



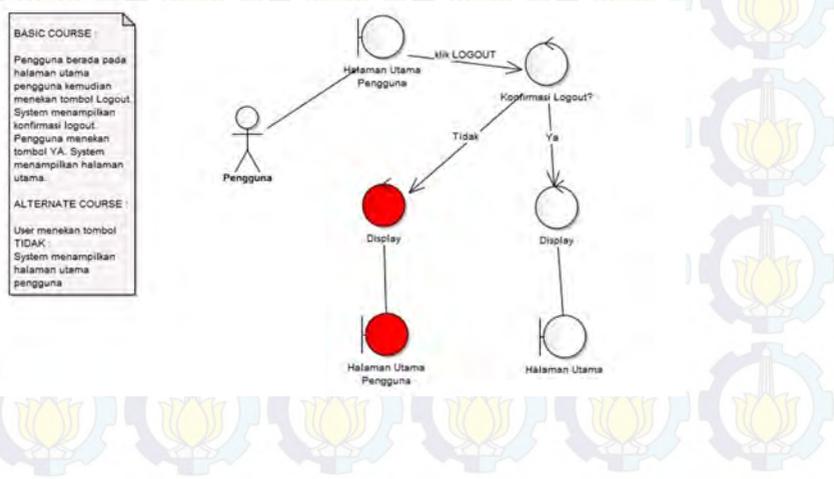


LAMPIRAN B
DIAGRAM ROBUSTNESS ANALYSIS

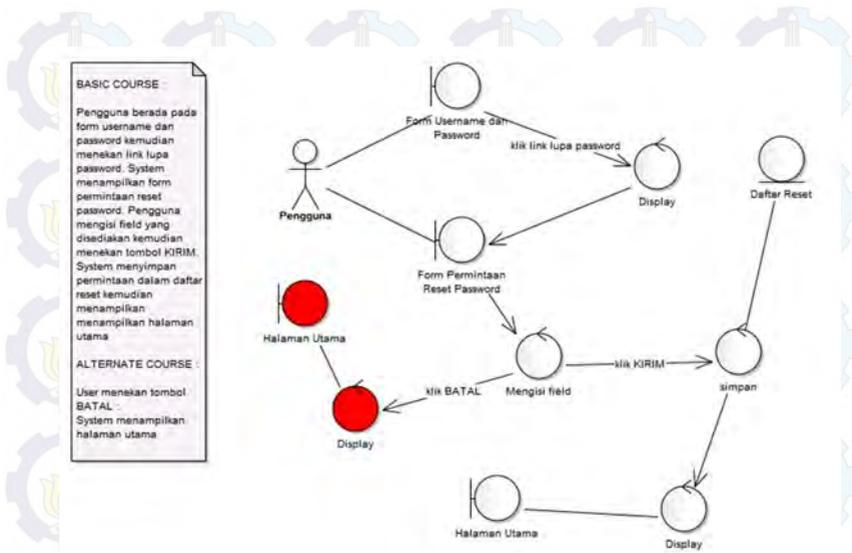
B.1. Diagram Login Robustness



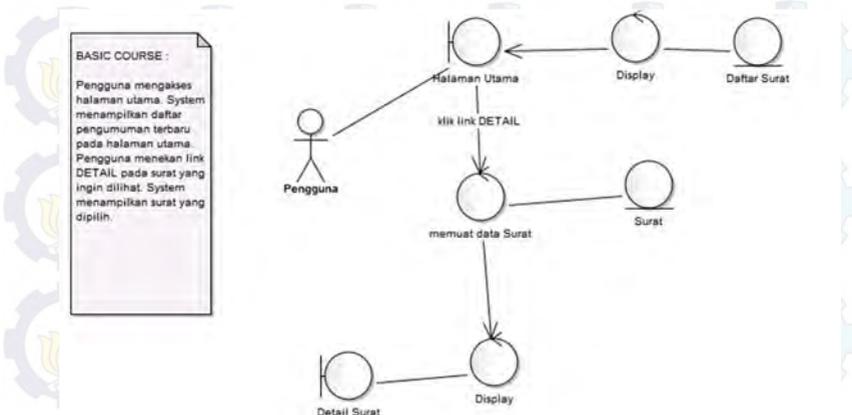
B.2. Diagram Logout Robustness



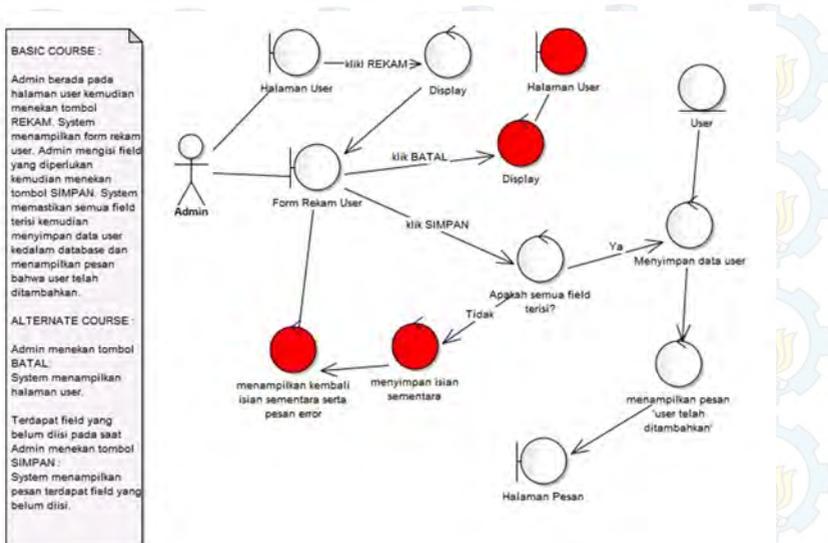
B.3. Diagram Meminta Reset Password Robustness



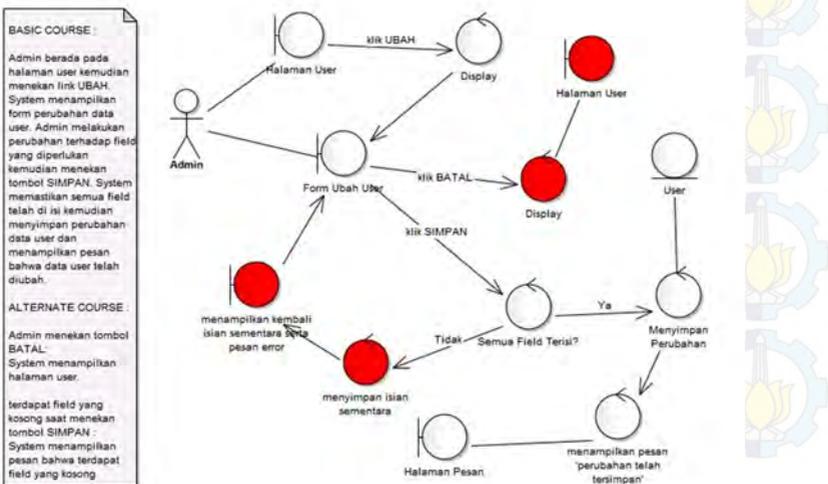
B.4. Diagram Melihat Pengumuman Robustness



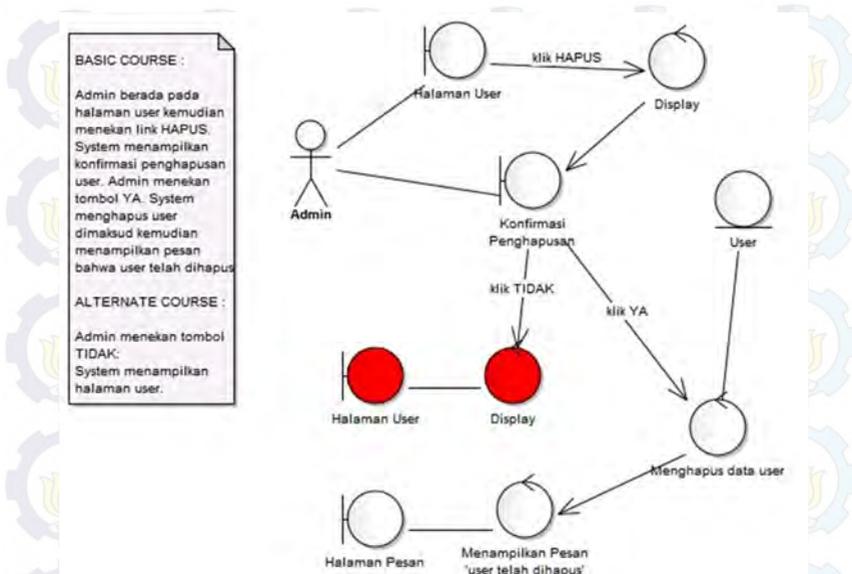
B.5. Diagram Rekam User Robustness



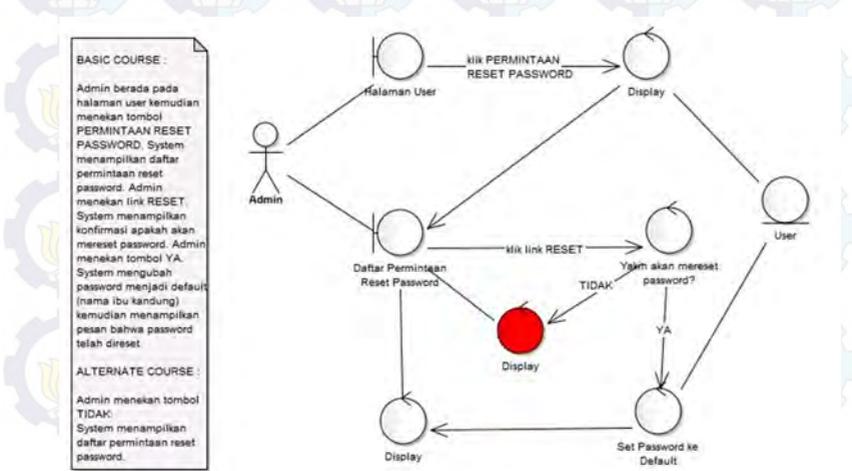
B.6. Diagram Ubah User Robustness



B.7. Diagram Hapus User Robustness



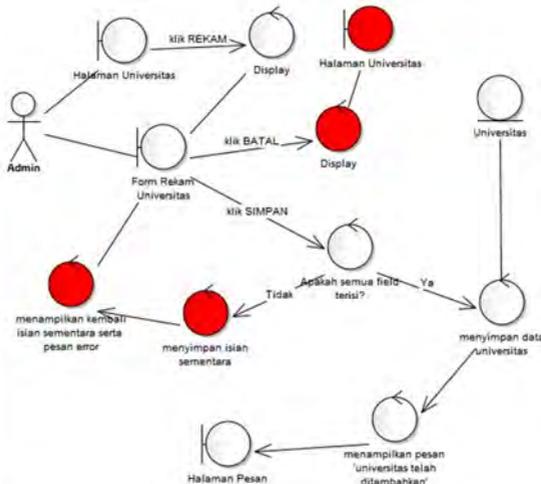
B.8. Diagram Reset Password User Robustness



B.9. Diagram Rekam Universitas Robustness

BASIC COURSE :
 Admin berada pada halaman universitas kemudian menekan tombol REKAM. System menampilkan form rekam universitas. Admin mengisi field yang diperlukan kemudian menekan tombol SIMPAN. System memastikan semua field terisi kemudian menyimpan data universitas ke dalam database dan menampilkan pesan bahwa universitas telah ditambahkan.

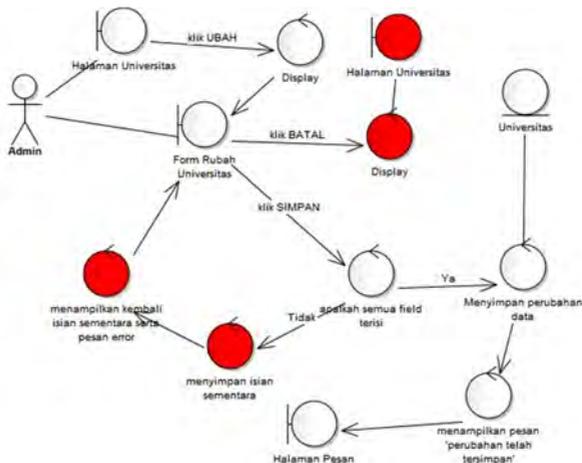
ALTERNATE COURSE :
 Admin menekan tombol BATAL. System menampilkan halaman universitas.
 Terdapat field yang belum diisi pada saat Admin menekan tombol SIMPAN. System menampilkan pesan terdapat field yang belum diisi.



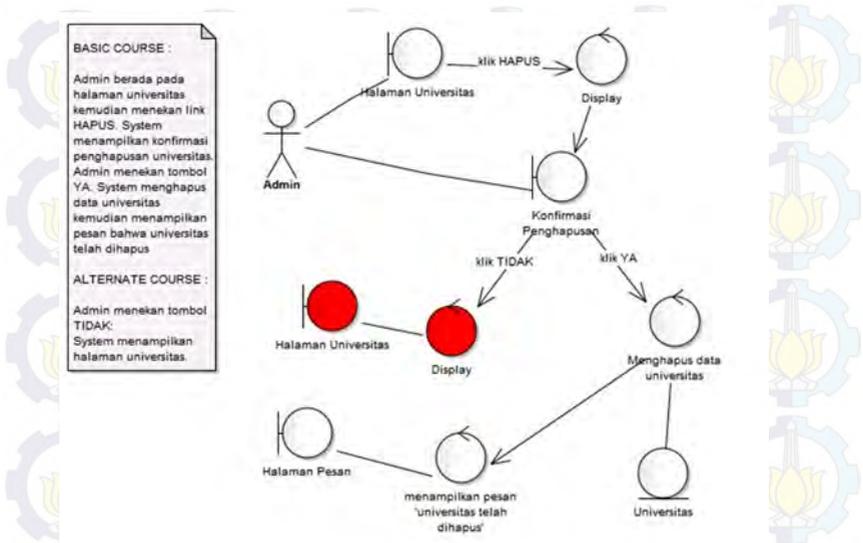
B.10. Diagram Ubah Universitas Robustness

BASIC COURSE :
 Admin berada pada halaman universitas kemudian menekan link UBAH. System menampilkan form perubahan data universitas. Admin melakukan perubahan terhadap field yang diperlukan kemudian menekan tombol SIMPAN. System memastikan semua field terisi kemudian menyimpan perubahan data dan menampilkan pesan bahwa data universitas telah diubah.

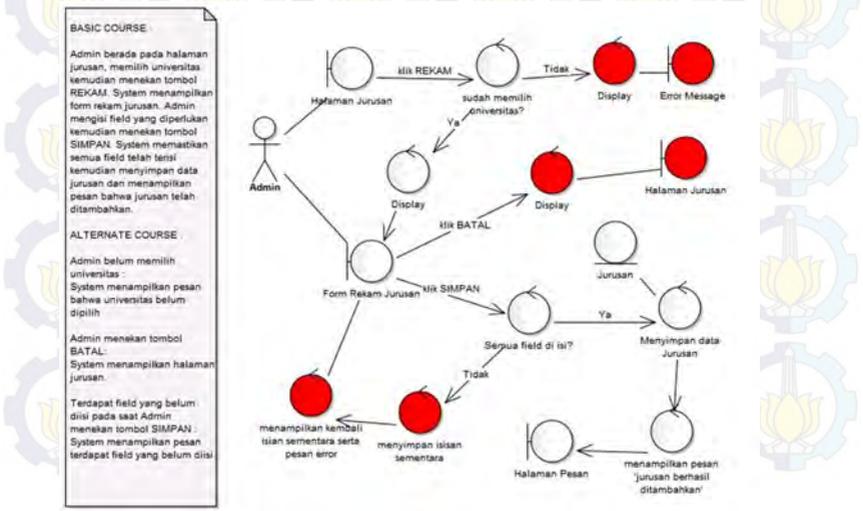
ALTERNATE COURSE :
 Admin menekan tombol BATAL. System menampilkan halaman universitas.



B.11. Diagram Hapus Universitas Robustness



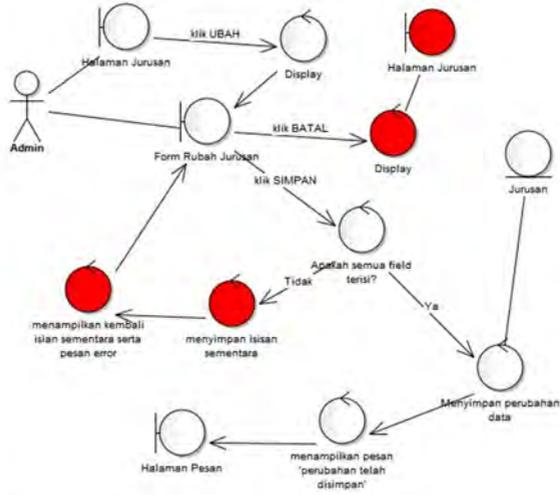
B.12. Diagram Rekam Jurusan Robustness



B.13. Diagram Ubah Jurusan Robustness

BASIC COURSE :
 Admin berada pada halaman Jurusan, memilih universitas, kemudian menekan link UBAH. System menampilkan form perubahan data jurusan. Admin melakukan perubahan terhadap field yang diperlukan kemudian menekan tombol SIMPAN. System memastikan semua field telah terisi kemudian menyimpan perubahan data dan menampilkan pesan bahwa data jurusan telah diubah.

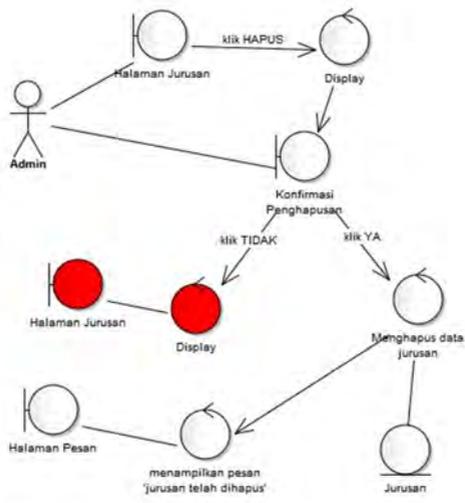
ALTERNATE COURSE :
 Admin menekan tombol BATAL. System menampilkan halaman jurusan.
 Terdapat field yang belum diisi pada saat Admin menekan tombol SIMPAN : System menampilkan pesan terdapat field yang belum diisi.



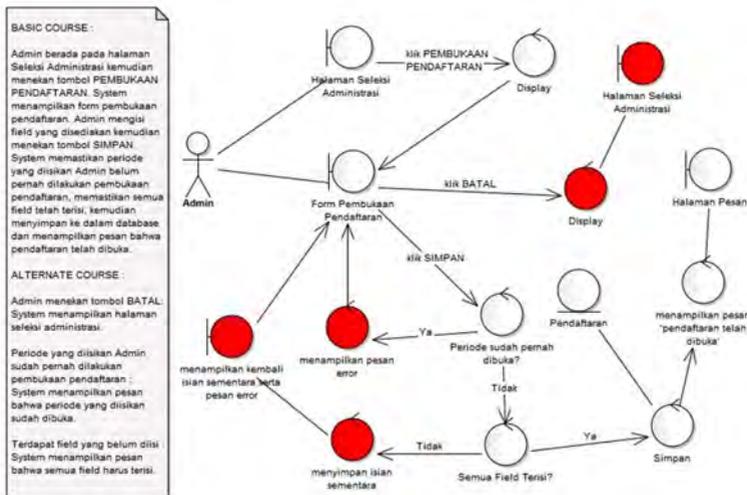
B.14. Diagram Hapus Jurusan Robustness

BASIC COURSE :
 Admin berada pada halaman jurusan, memilih universitas kemudian menekan link HAPUS. System menampilkan konfirmasi penghapusan jurusan. Admin menekan tombol YA. System menghapus data jurusan kemudian menampilkan pesan bahwa jurusan telah dihapus.

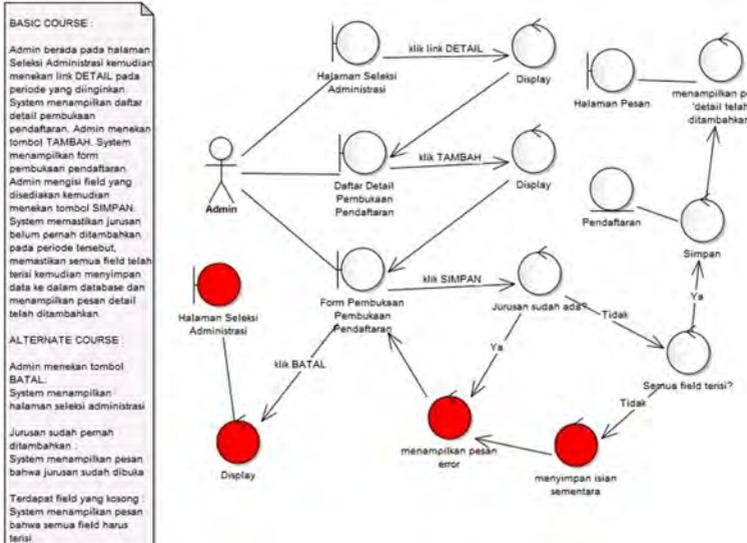
ALTERNATE COURSE :
 Admin menekan tombol TIDAK. System menampilkan halaman jurusan.



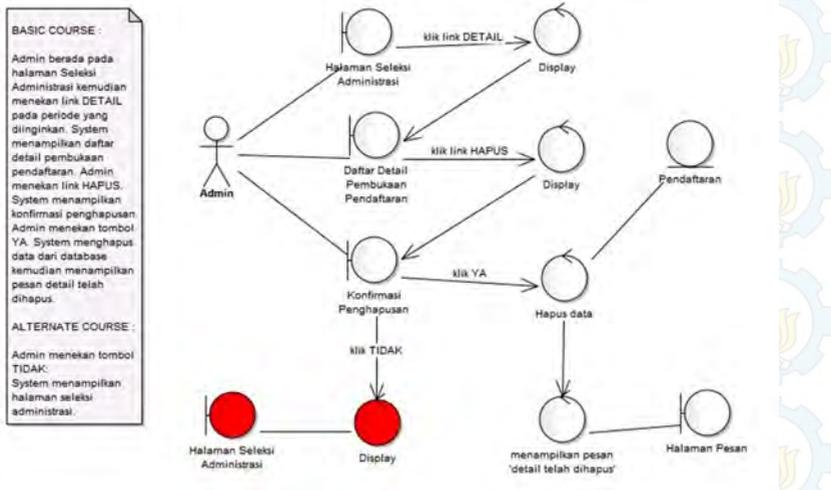
B.15. Diagram Membuka Pendaftaran Robustness



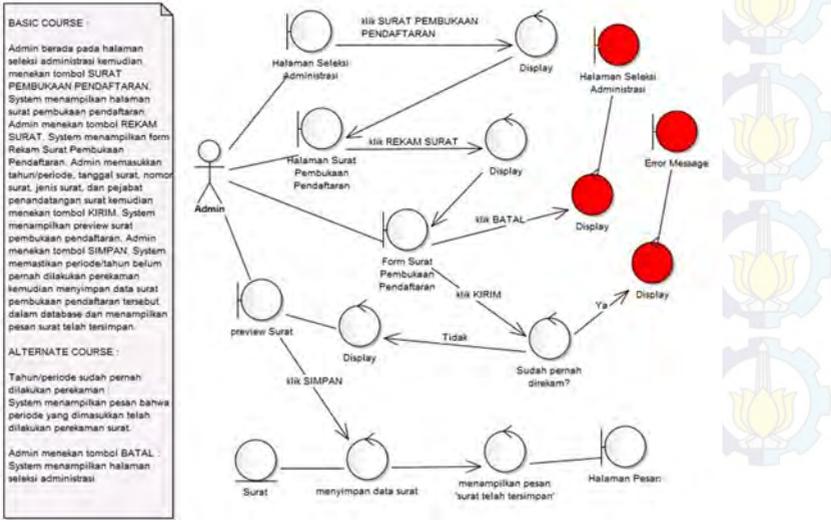
B.16. Diagram Tambah Detail Pembukaan Pendaftaran Robustness



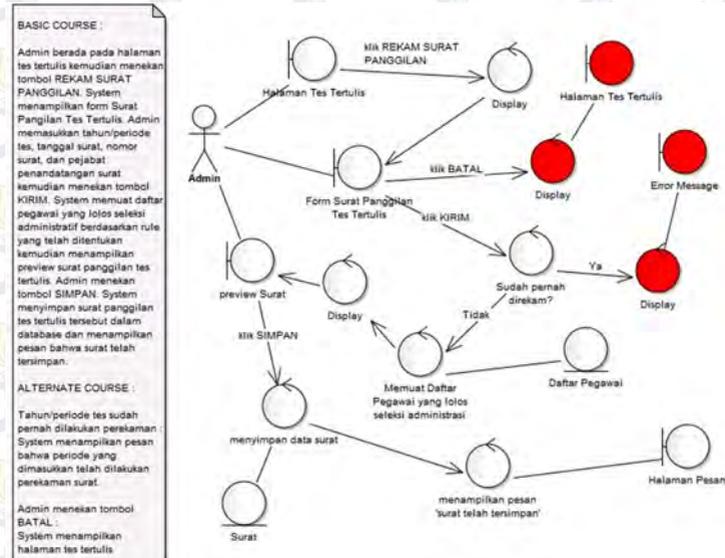
B.17. Diagram Hapus Detail Pembukaan Pendaftaran Robustness



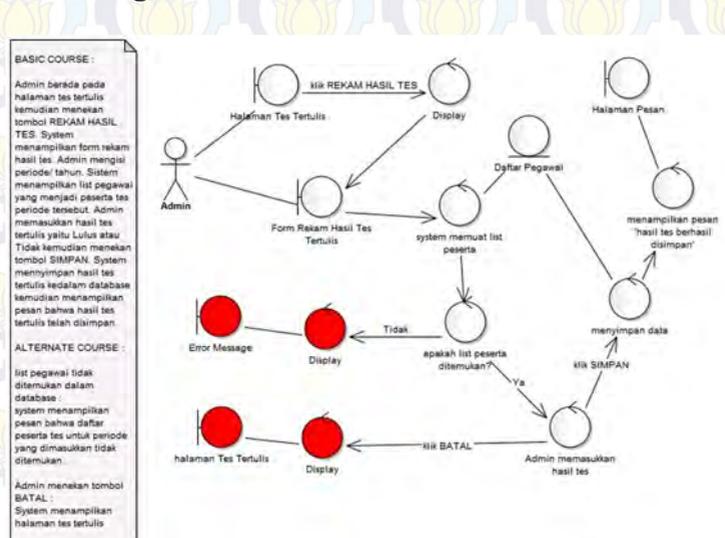
B.18. Diagram Rekam Surat Pembukaan Pendaftaran Robustness



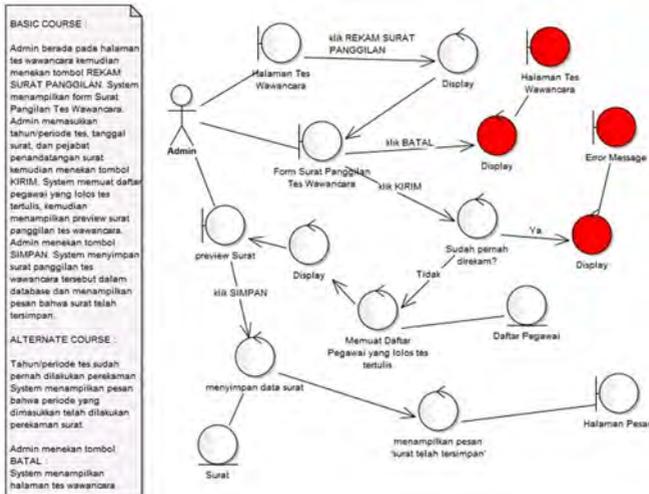
B.19. Diagram Rekam Surat Panggilan Tes Tertulis Robustness



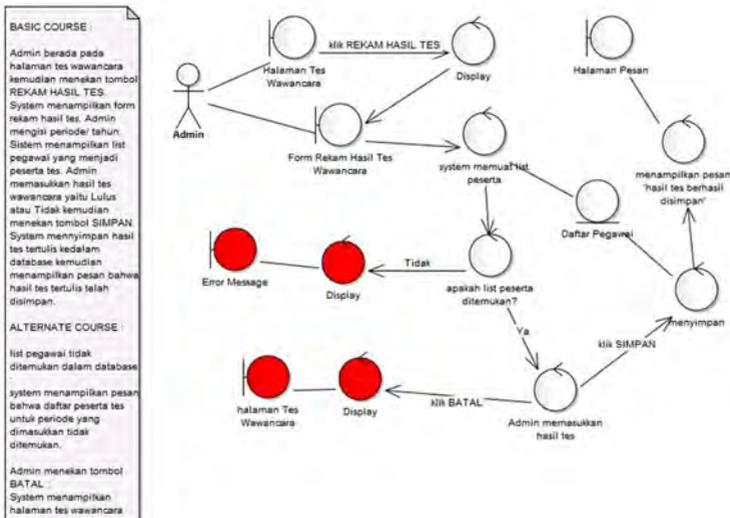
B.20. Diagram Rekam Hasil Tes Tertulis Robustness



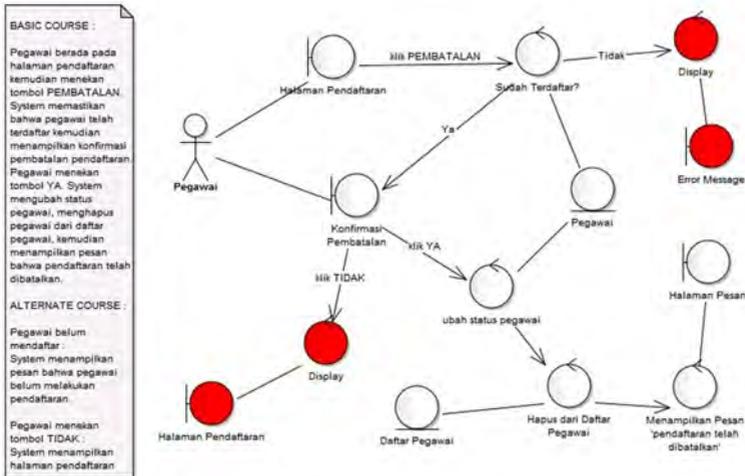
B.21. Diagram Rekam Surat Panggilan Tes Wawancara Robustness



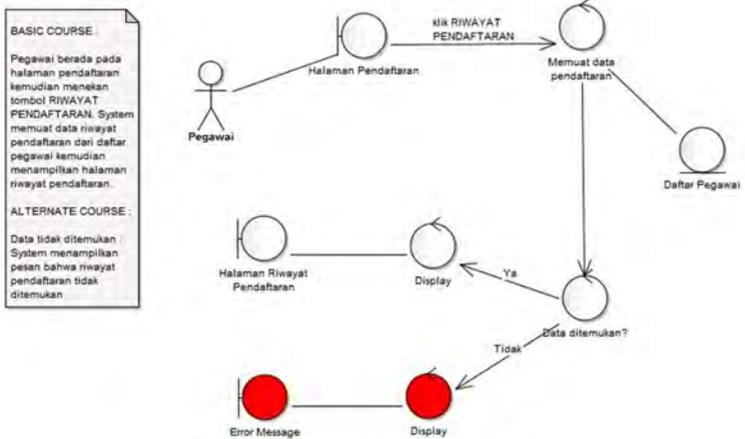
B.22. Diagram Rekam Hasil Tes Wawancara Robustness



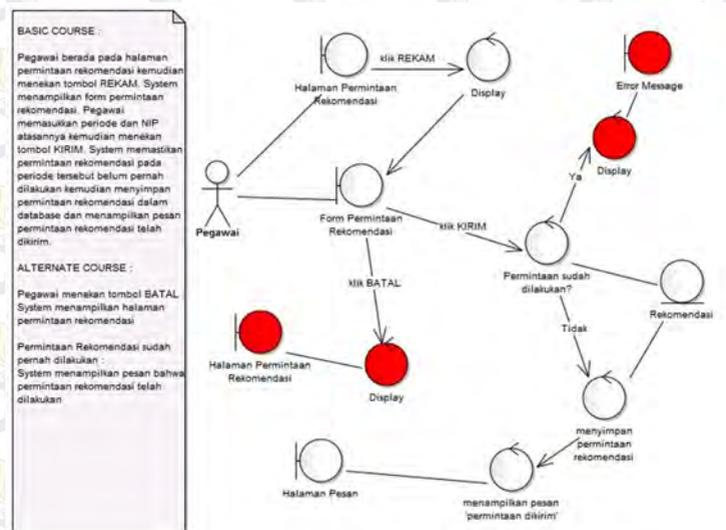
B.25. Diagram Membatalkan Pendaftaran Robustness



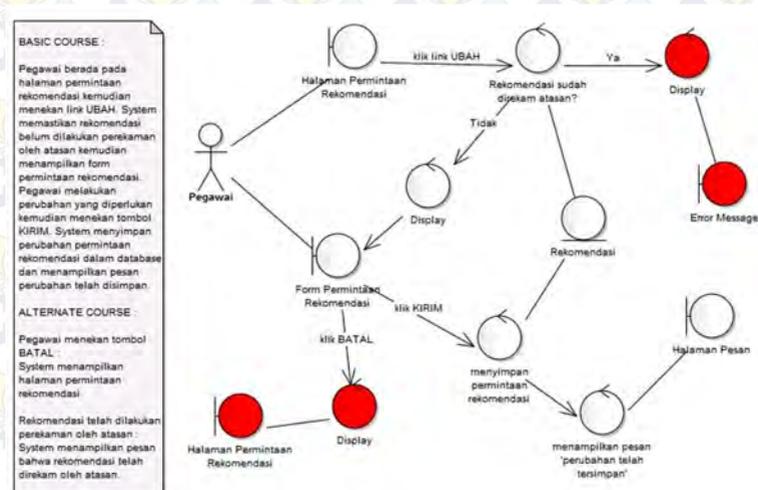
B.26. Diagram Melihat Riwayat Pendaftaran Robustness



B.27. Diagram Rekam Permintaan Rekomendasi Robustness



B.28. Diagram Ubah Permintaan Rekomendasi Robustness

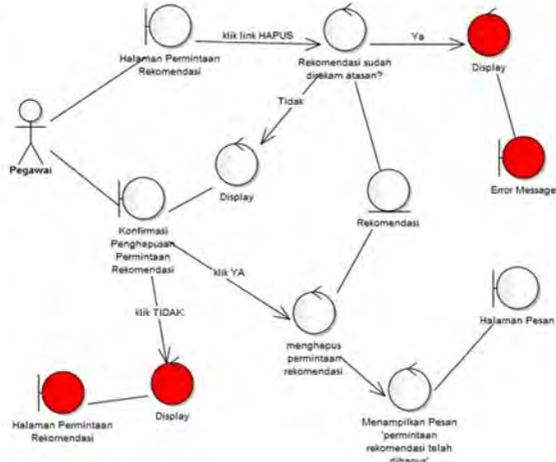


B.29. Diagram Hapus Permintaan Rekomendasi Robustness

BASIC COURSE :
 Pegawai berada pada halaman permintaan rekomendasi kemudian menekan link HAPUS. System memastikan rekomendasi belum dilakukan perekaman oleh atasan kemudian menampilkan konfirmasi penghapusan permintaan rekomendasi. Pegawai menekan tombol YA. System menghapus data permintaan rekomendasi dalam database dan menampilkan pesan permintaan rekomendasi telah dihapus.

ALTERNATE COURSE :
 Pegawai menekan tombol TIDAK. System menampilkan halaman permintaan rekomendasi

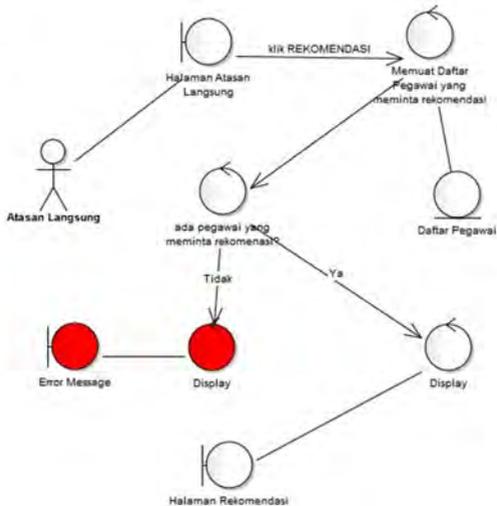
Rekomendasi telah dilakukan perekaman oleh atasan : System menampilkan pesan bahwa rekomendasi telah direkam oleh atasan.



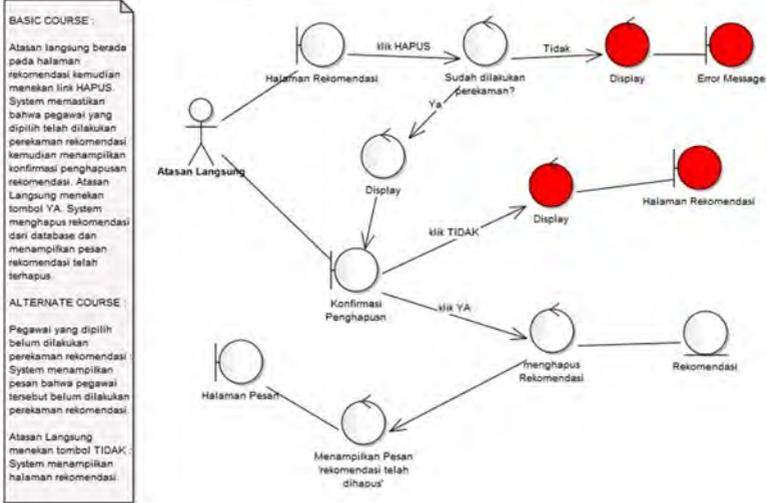
B.30. Diagram Melihat Permintaan Rekomendasi Robustness

BASIC COURSE :
 Atasan Langsung berada pada halaman atasan langsung kemudian memilih menu REKOMENDASI. System memuat daftar pegawai yang meminta rekomendasi dari atasan tersebut kemudian menampilkan nya pada halaman rekomendasi

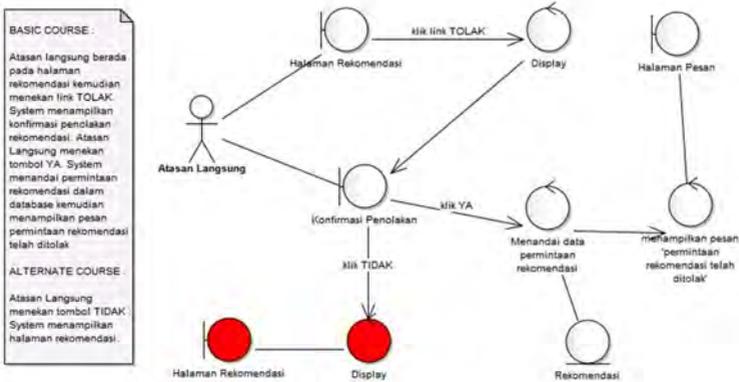
ALTERNATE COURSE :
 Tidak ada yang meminta rekomendasi dari atasan tersebut : System menampilkan pesan bahwa tidak ada pegawai yang meminta rekomendasi dari atasan tersebut.

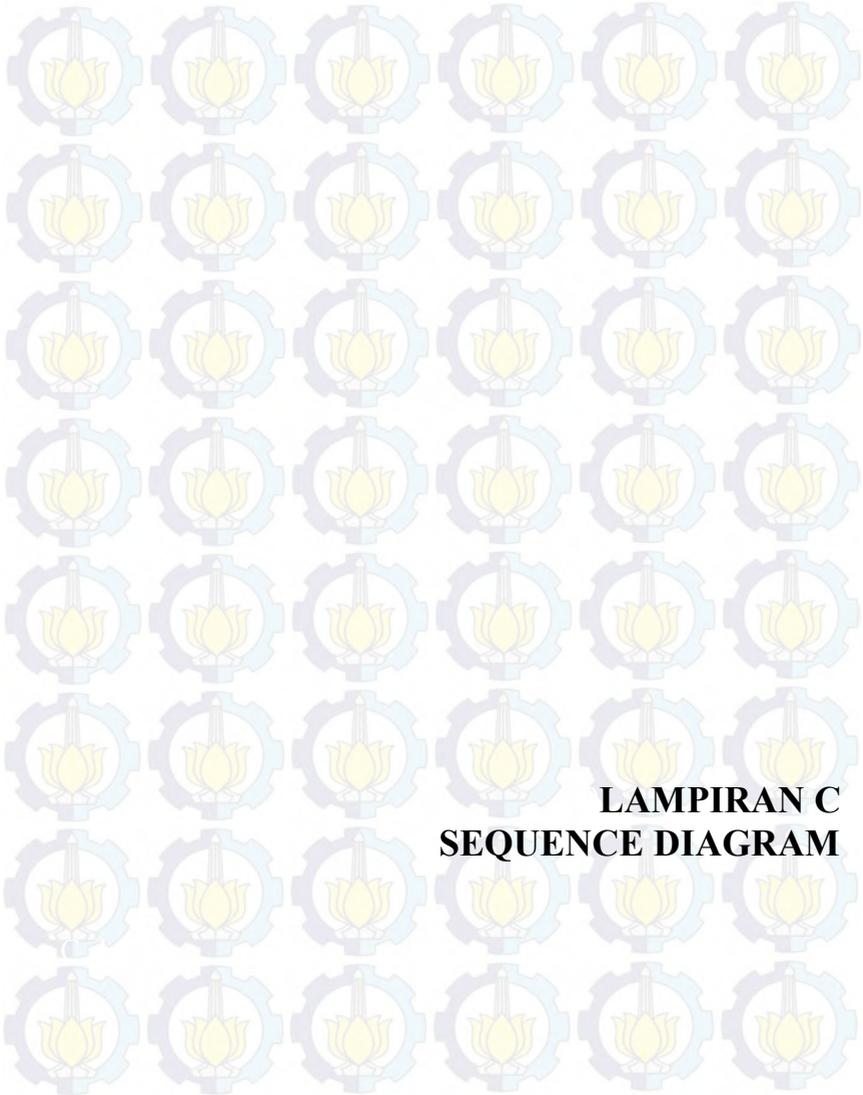


B.33. Diagram Hapus Rekomendasi Robustness



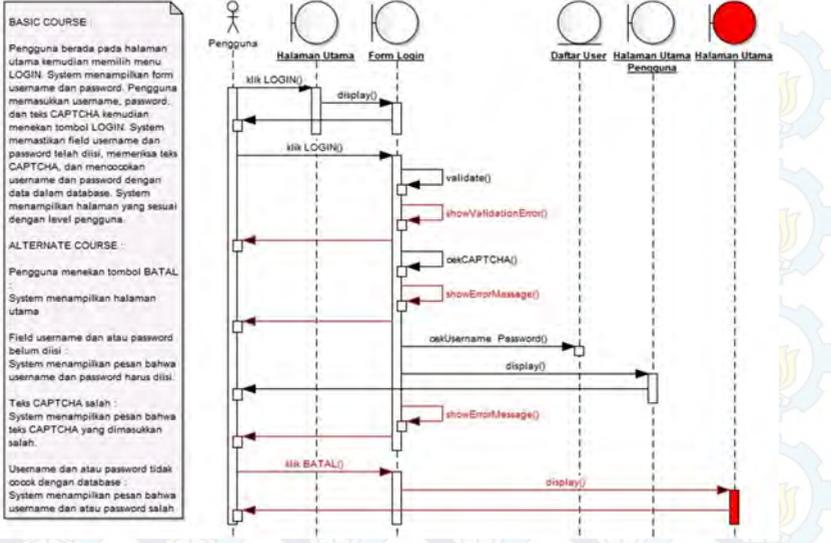
B.34. Diagram Tolak Rekomendasi Robustness



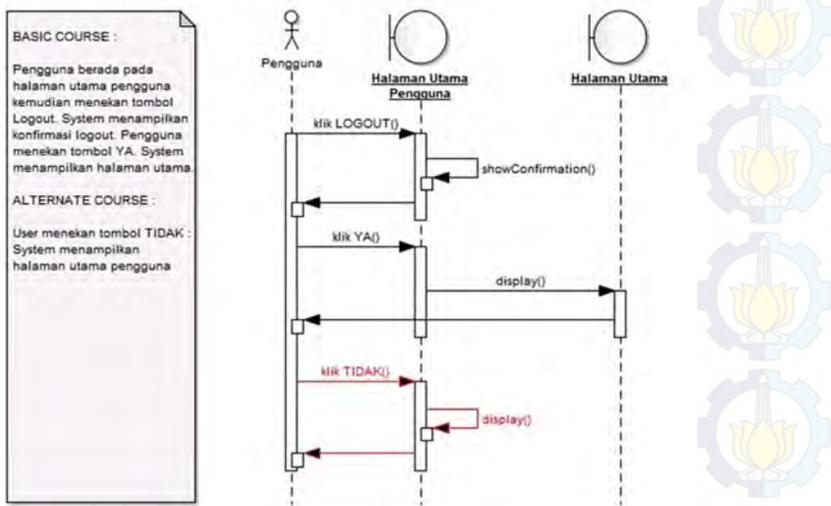


LAMPIRAN C
SEQUENCE DIAGRAM

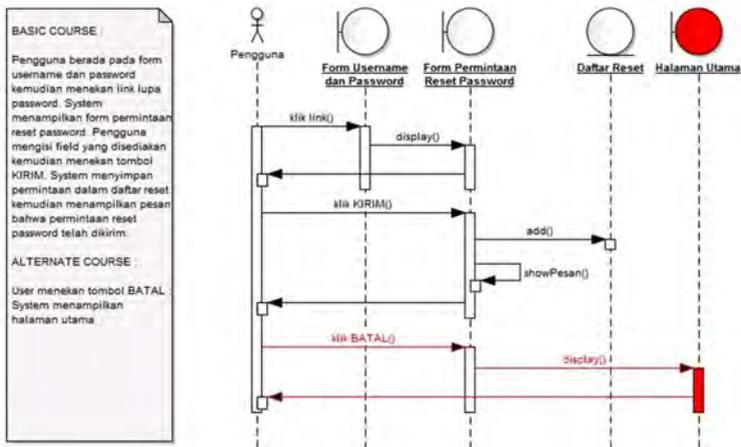
C.1. Diagram Login Sequence



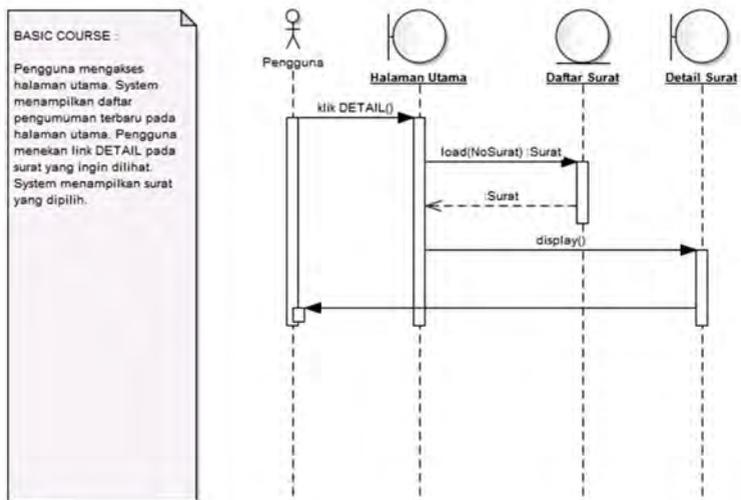
C.2. Diagram Logout Sequence



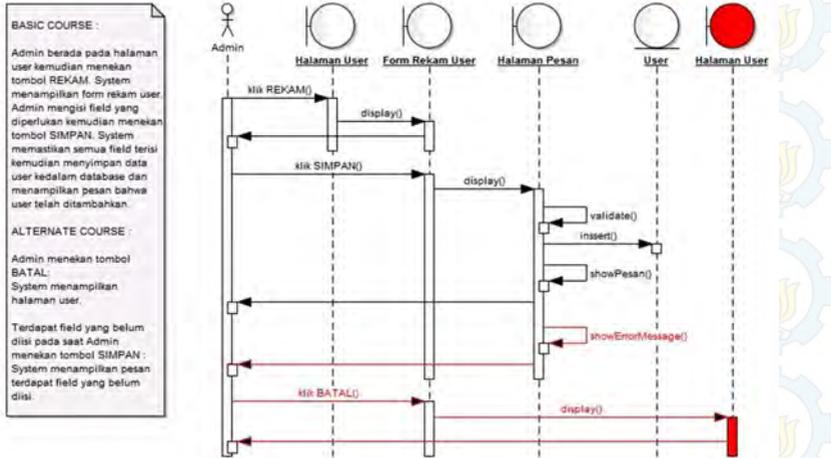
C.3. Diagram Meminta Reset Password Sequence



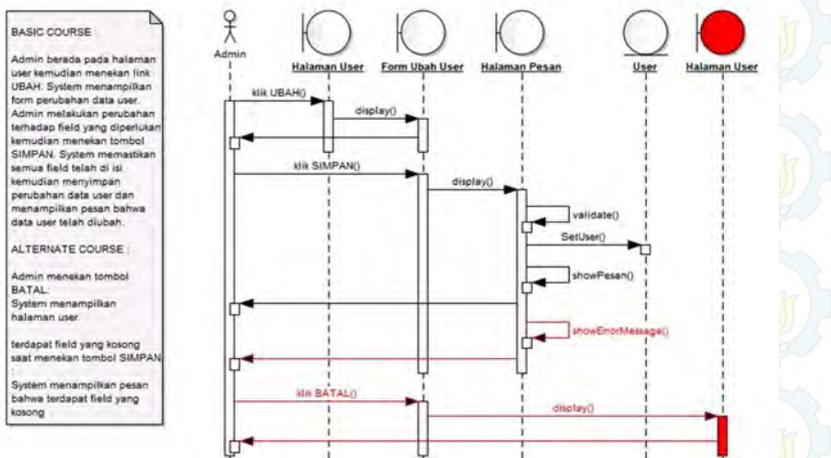
C.4. Diagram Melihat Pengumuman Sequence



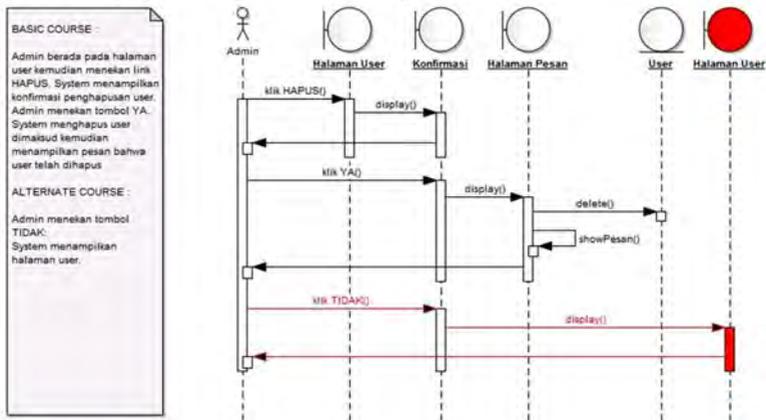
C.5. Diagram Rekam User Sequence



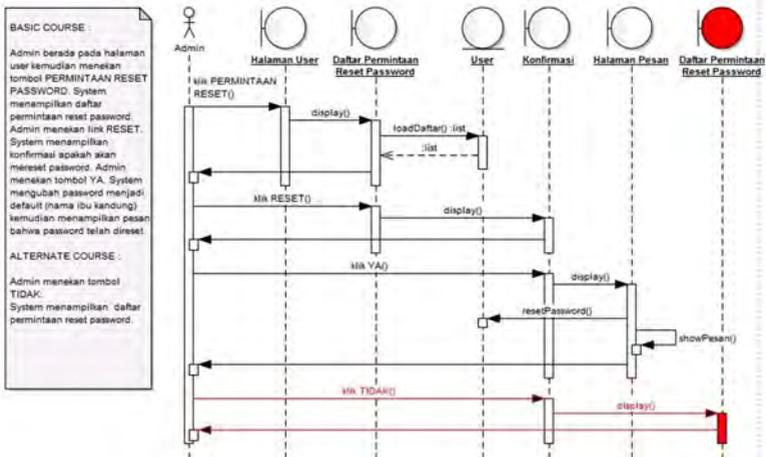
C.6. Diagram Ubah User Sequence



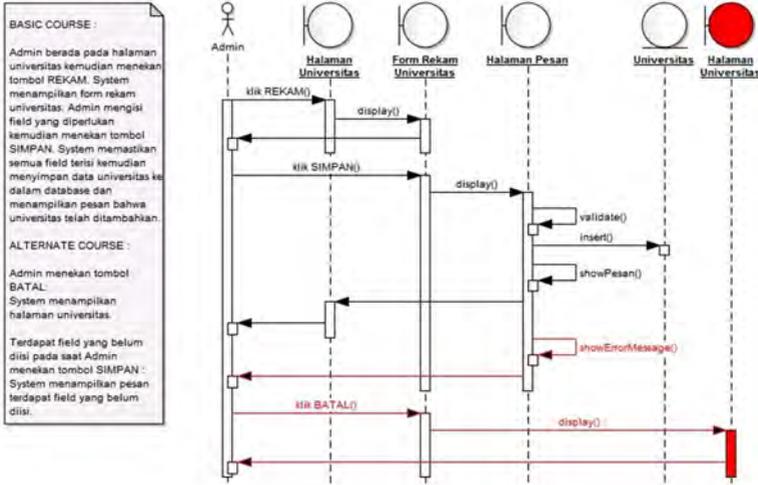
C.7. Diagram Hapus User Sequence



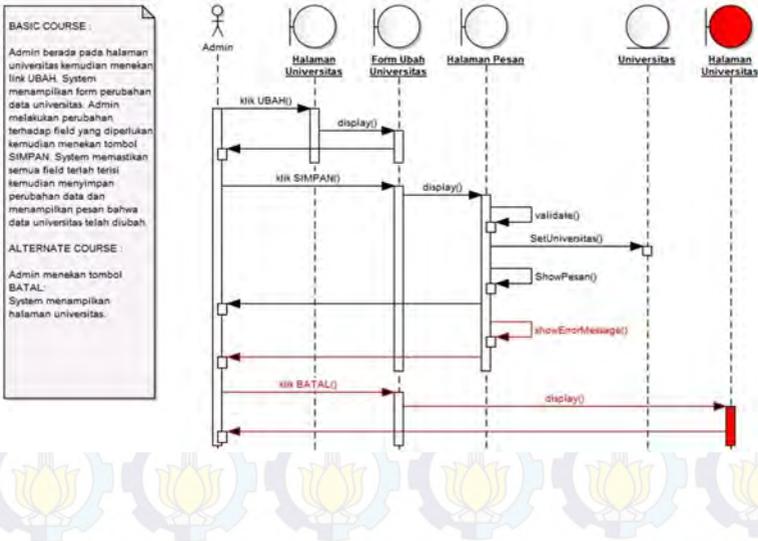
C.8. Diagram Reset Password User Sequence



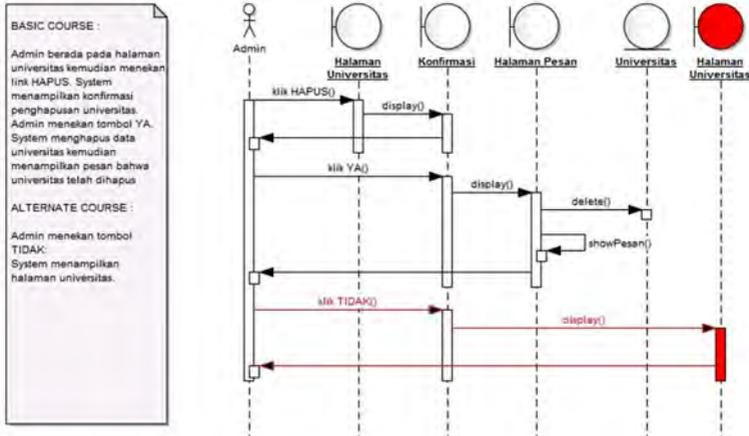
C.9. Diagram Rekam Universitas Sequence



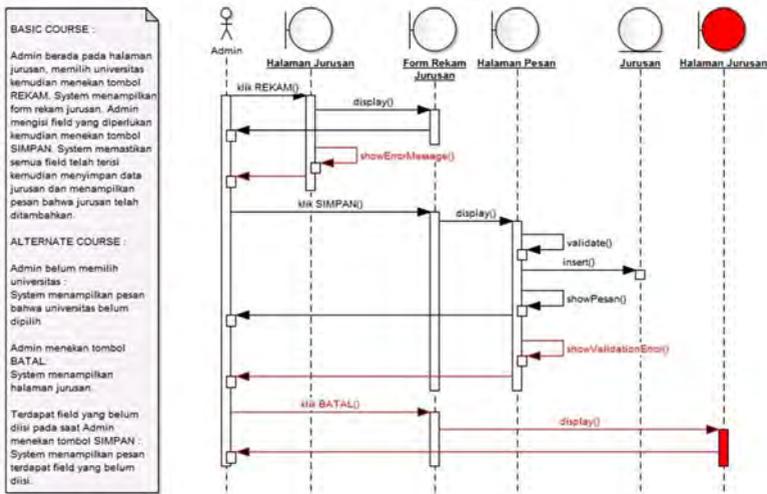
C.10. Diagram Ubah Universitas Sequence



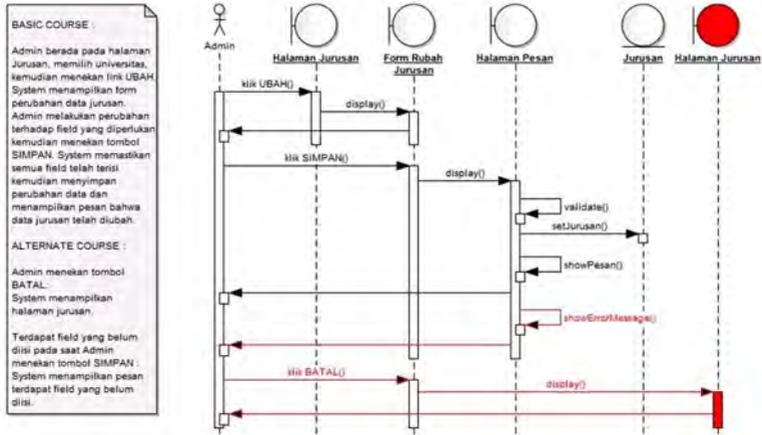
C.11. Diagram Hapus Universitas Sequence



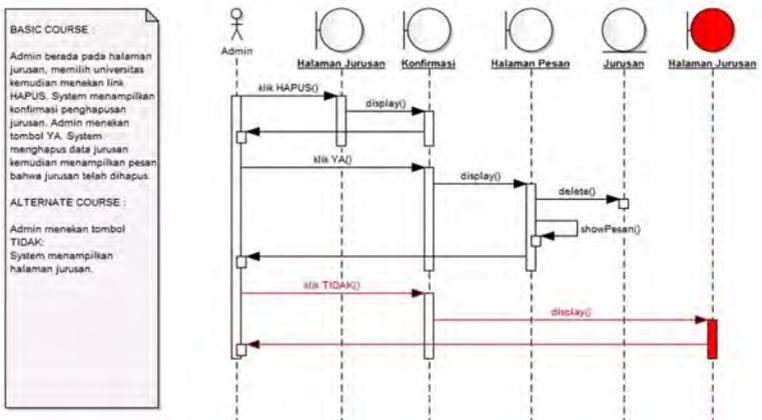
C.12. Diagram Rekam Jurusan Sequence



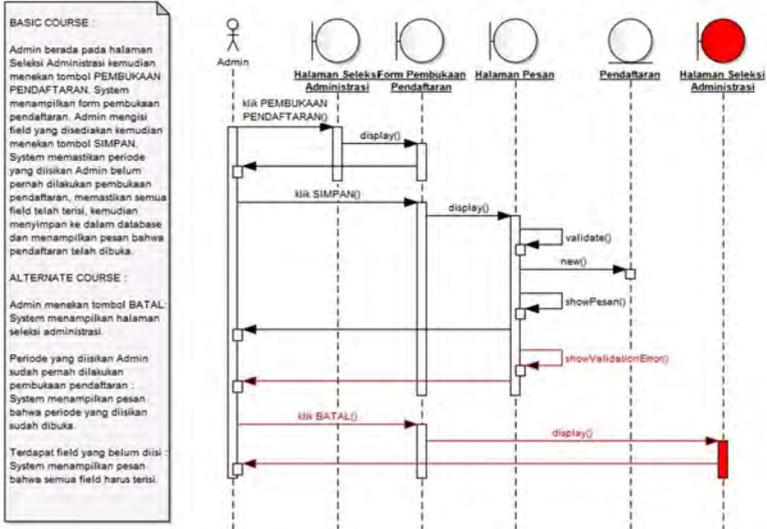
C.13. Diagram Ubah Jurusan Sequence



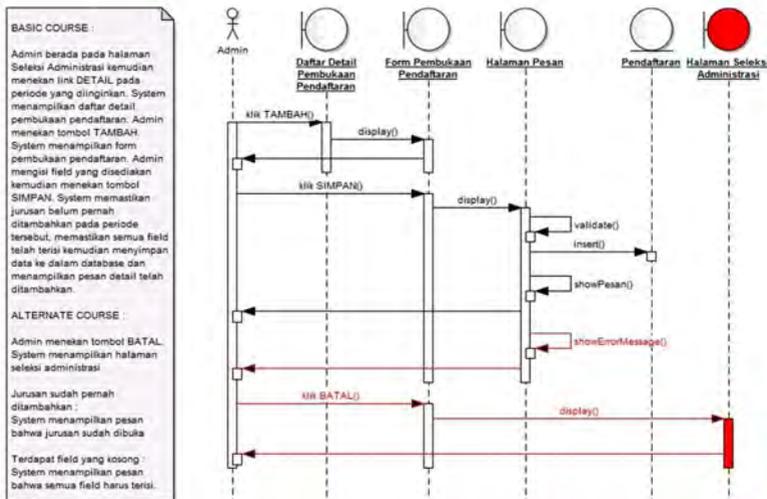
C.14. Diagram Hapus Jurusan Sequence



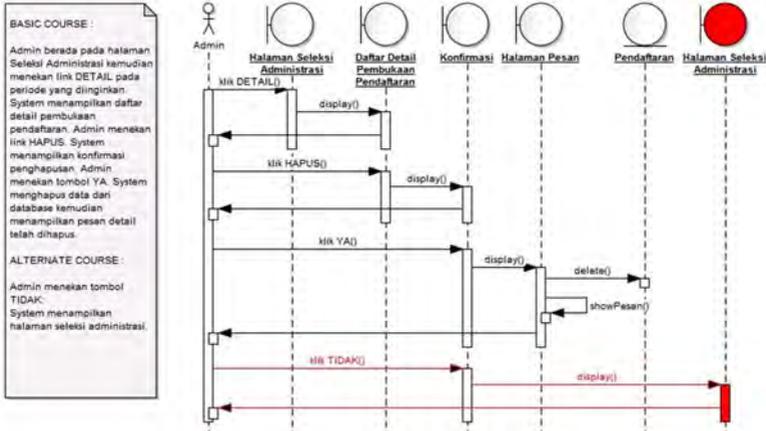
C.15. Diagram Membuka Pendaftaran Sequence



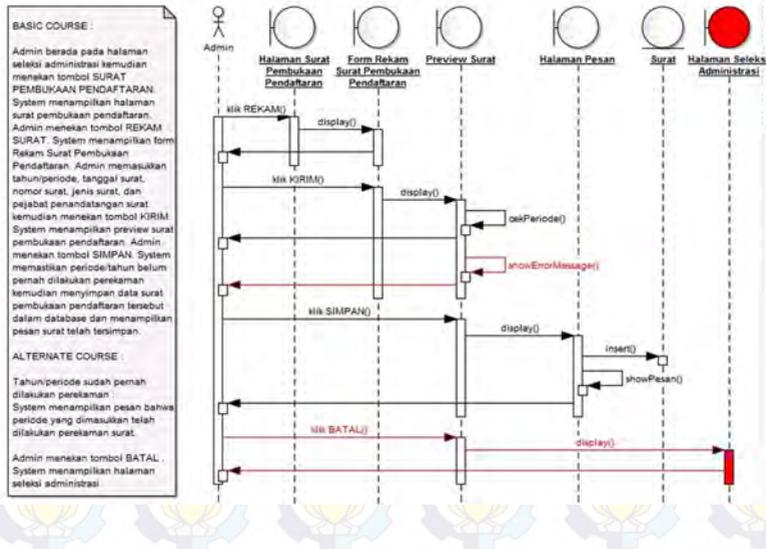
C.16. Diagram Tambah Detail Pembukaan Pendaftaran Sequence



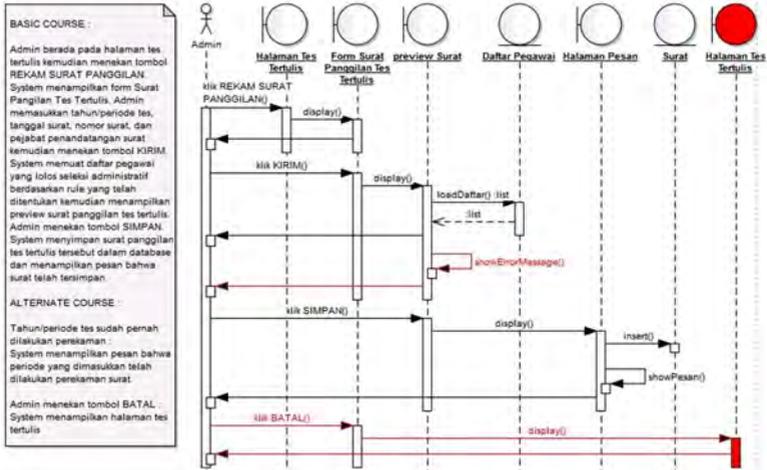
C.17. Diagram Hapus Detail Pembukaan Pendaftaran Sequence



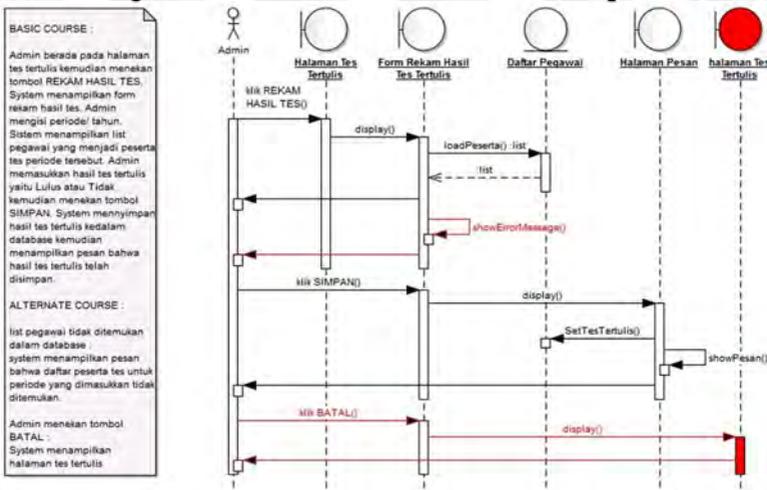
C.18. Diagram Rekam Surat Pembukaan Pendaftaran Sequence



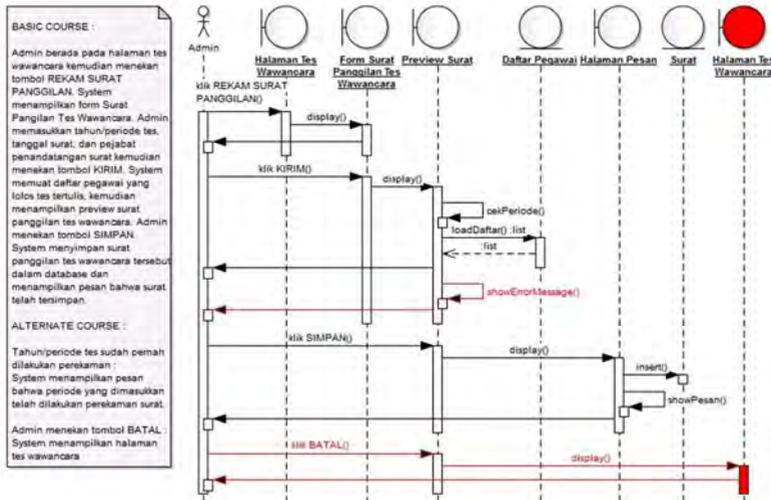
C.19. Diagram Rekam Surat Panggilan Tes Tertulis Sequence



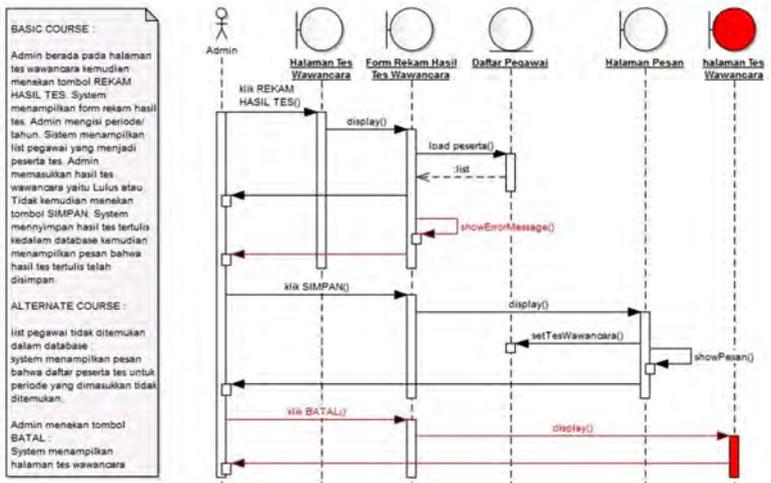
C.20. Diagram Rekam Hasil Tes Tertulis Sequence



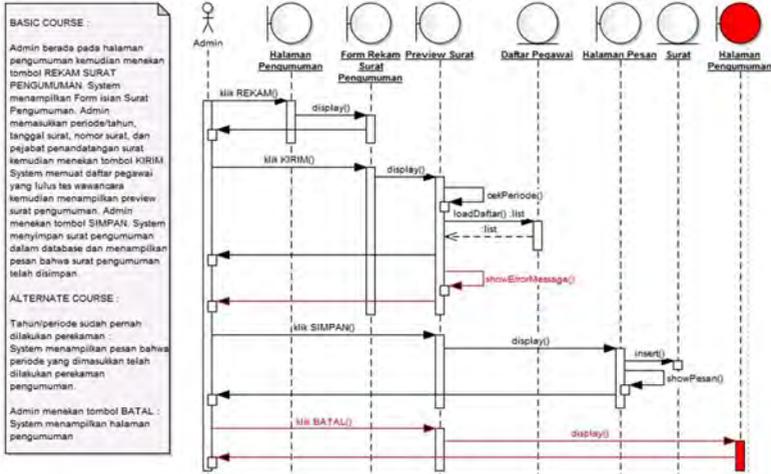
C.21. Diagram Rekam Surat Panggilan Tes Wawancara Sequence



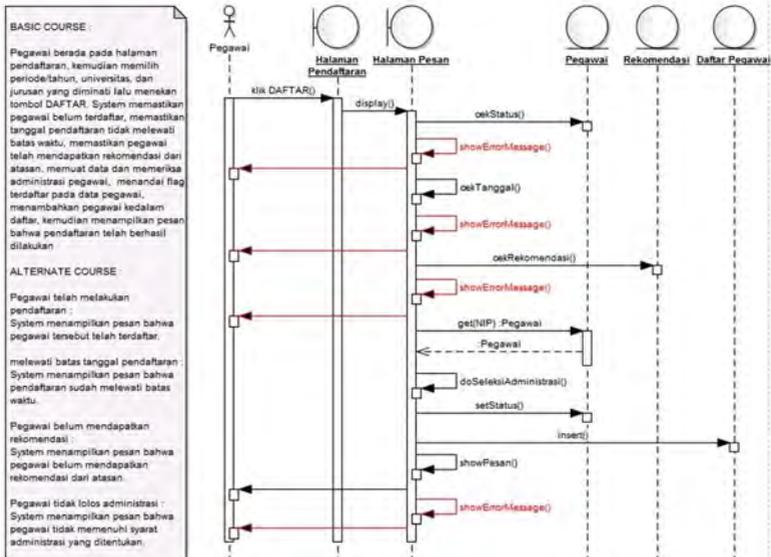
C.22. Diagram Rekam Hasil Tes Wawancara Sequence



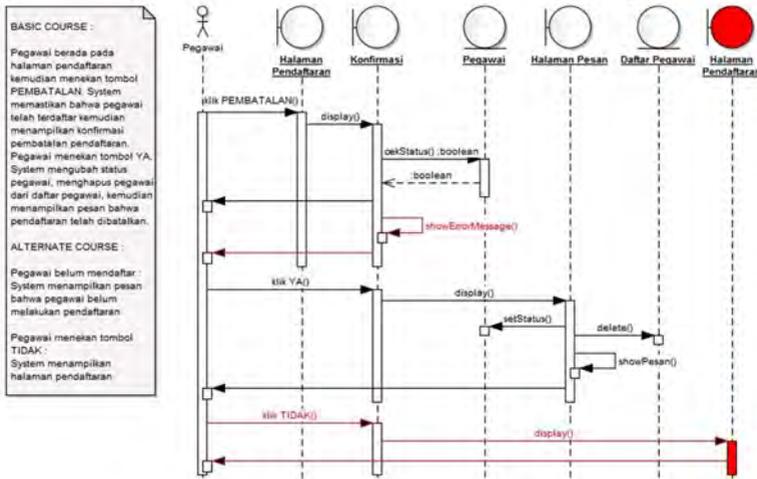
C.23. Diagram Rekam Surat Pengumuman Sequence



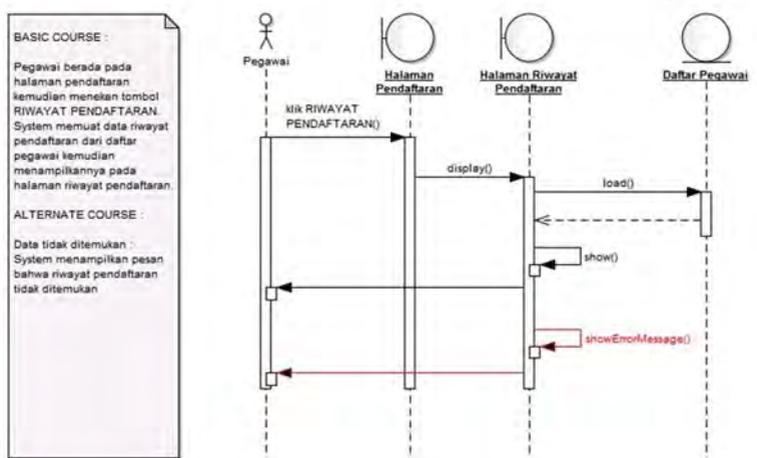
C.24. Diagram Mendaftar Sequence



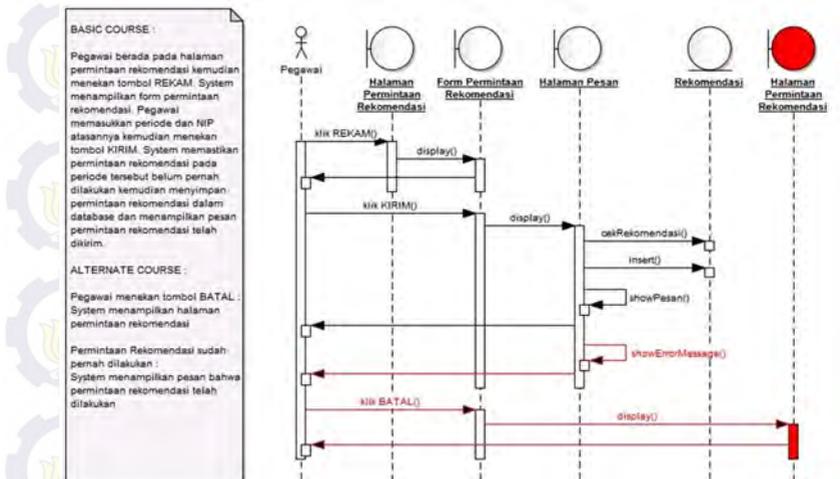
C.25. Diagram Membatalkan Pendaftaran Sequence



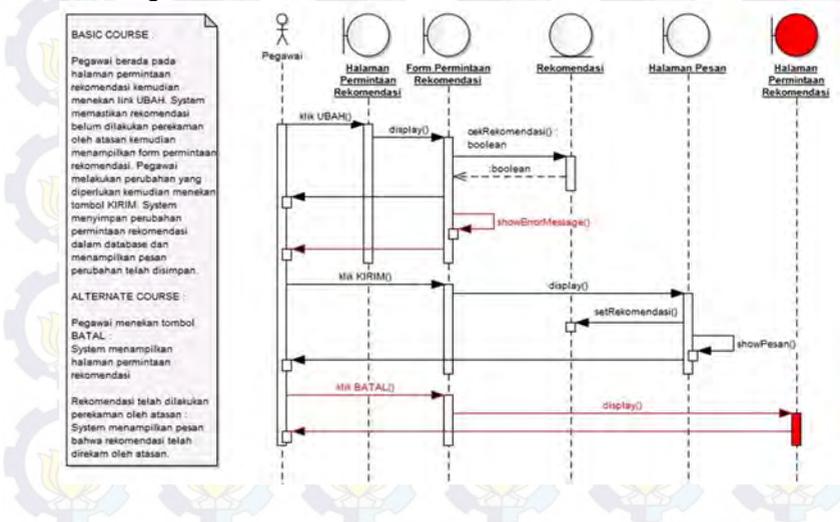
C.26. Diagram Melihat Riwayat Pendaftaran Sequence



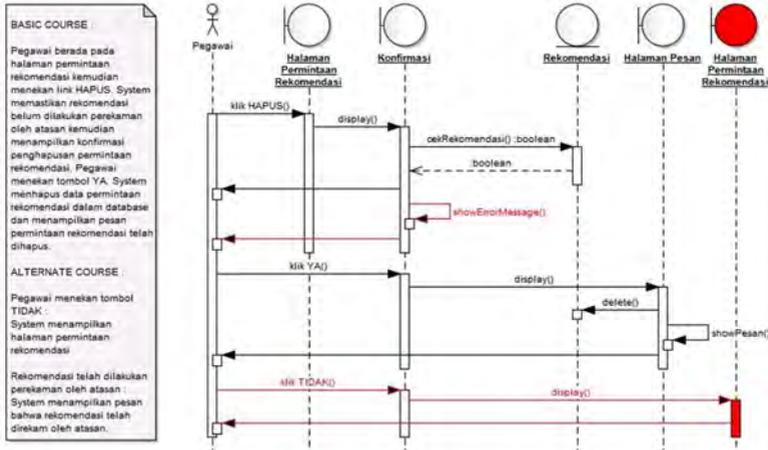
C.27. Diagram Rekam Permintaan Rekomendasi Sequence



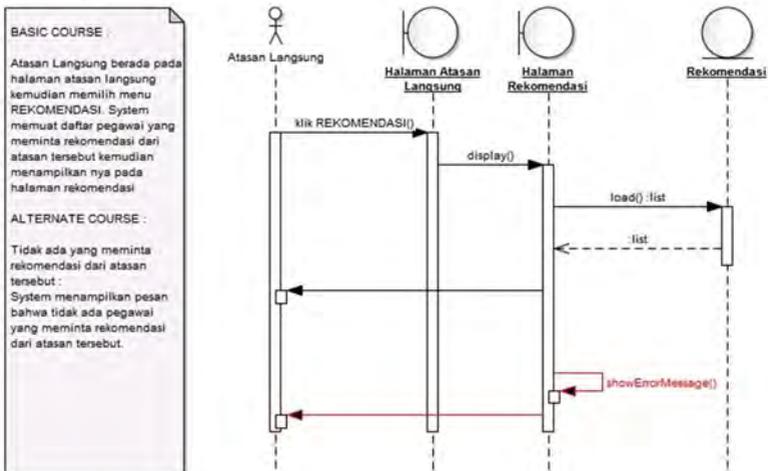
C.28. Diagram Ubah Permintaan Rekomendasi Sequence



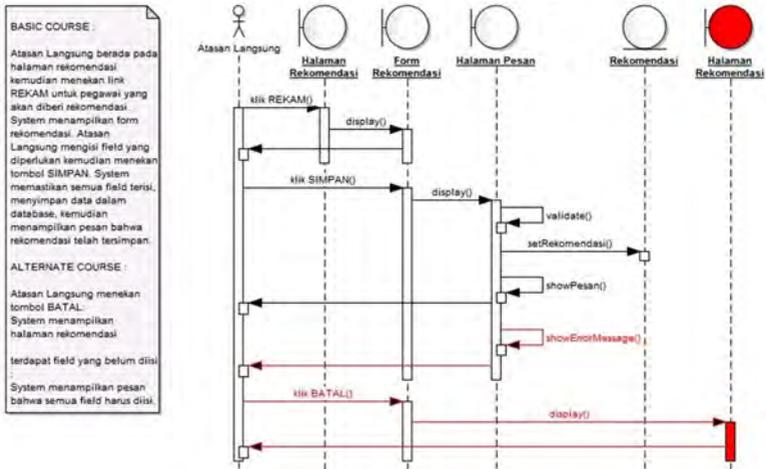
C.29. Diagram Hapus Permintaan Rekomendasi Sequence



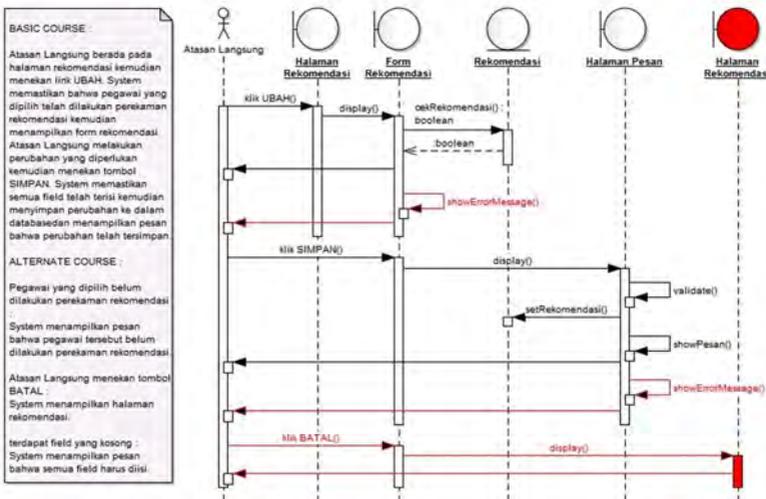
C.30. Diagram Melihat Permintaan Rekomendasi Sequence



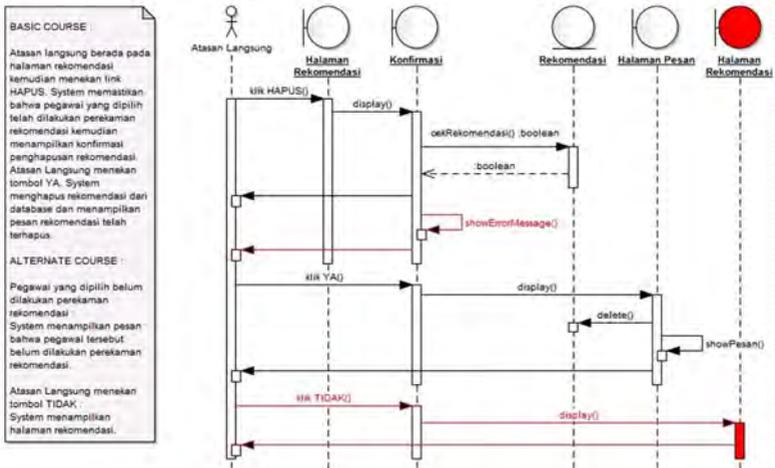
C.31. Diagram Rekam Rekomendasi Sequence



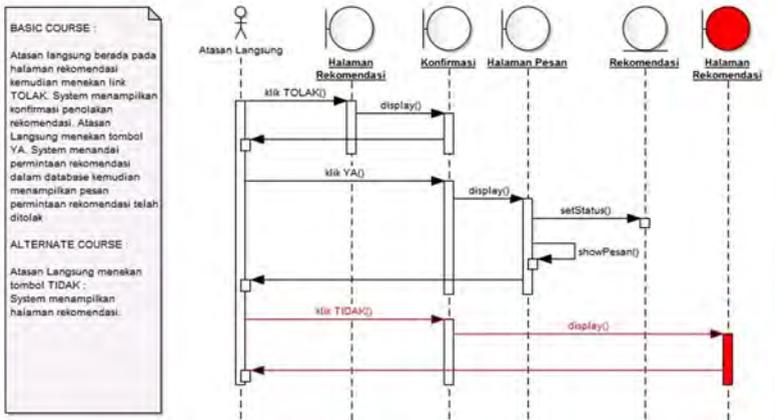
C.32. Diagram Ubah Rekomendasi Sequence

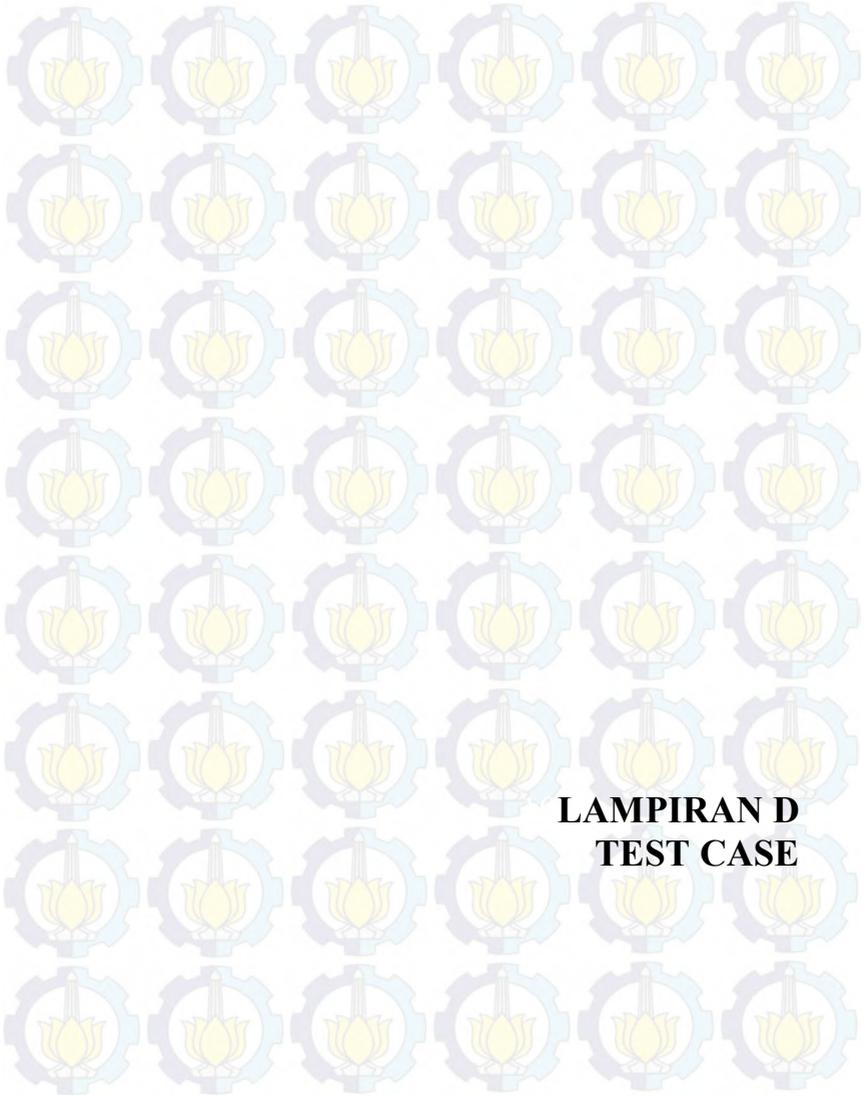


C.33. Diagram Hapus Rekomendasi Sequence



C.34. Diagram Tolak Rekomendasi Sequence





LAMPIRAN D
TEST CASE

D-2

D.1. Test Case Login

No	Skenario	Aksi	Hasil	Sukses / Gagal
1	Pengguna berada pada halaman utama	Klik “login”	Menampilkan form login	Sukses
2	Salah satu field tidak diisi (kosong)	Klik “login”	Muncul pesan error	Sukses
3	Username dan atau password salah	Klik “login”	Muncul pesan error	Sukses
4	Text CAPTCHA salah	Klik “login”	Muncul pesan error	Sukses
5	Pengguna berada pada form login kemudian membatalkan login	Klik “batal”	Kembali ke halaman Utama	Sukses

D.2. Test Case Logout

No	Skenario	Aksi	Hasil	Sukses / Gagal
1	Pengguna berada pada halaman utama pengguna	Klik “logout”	Menampilkan konfirmasi logout	Sukses
2	Pengguna berada pada konfirmasi logout dan melanjutkan proses logout	Klik “ya”	Menampilkan halaman utama	Sukses
3	Pengguna berada pada konfirmasi logout dan membatalkan proses logout	Klik “tidak”	Kembali ke halaman utama pengguna	Sukses

D.3. Test Case Meminta Reset Password

No	Skenario	Aksi	Hasil	Sukses / Gagal
1	Pengguna berada pada form Login	Klik “lupa password”	Menampilkan form permintaan reset password	Sukses
2	Pengguna mengisi form permintaan reset pasword kemudian melanjutkan proses permintaan	Klik “kirim”	Menyimpan isian dan menampilkan halaman utama	Sukses
3	Pengguna berada pada form permintaan reset password kemudian membatalkan permintaan.	Klik “batal”	Kembali ke halaman utama	Sukses

D.4. Test Case Melihat Pengumuman

No	Skenario	Aksi	Hasil	Sukses / Gagal
1	Pengguna mengakses halaman utama	Ketik URL	Menampilkan daftar surat pengumuman pada halaman utama	Sukses
2	Pada halaman utama, pengguna menekan link salah satu daftar surat	Klik “detail”	Menampilkan detail surat yang dipilih	Sukses

D-4

D.5. Test Case Rekam User

No	Skenario	Aksi	Hasil	Sukses / Gagal
1	Admin berada pada halaman user	Klik “rekam”	Menampilkan form rekam user	Sukses
2	Pengguna mengisi form rekam kemudian melanjutkan proses perekaman user	Klik “simpan”	Menyimpan isian dan menampilkan pesan user telah ditambahkan	Sukses
3	Pengguna berada pada form rekam user namun membatalkan proses perekaman	Klik “batal”	Kembali ke halaman user	Sukses
4	Terdapat field yang belum diisi oleh pengguna pada form rekam user	Klik “simpan”	Menampilkan pesan error	Sukses

D.6. Test Case Ubah User

No	Skenario	Aksi	Hasil	Sukses / Gagal
1	Admin berada pada halaman user	Klik “rekam”	Menampilkan form ubah user	Sukses
2	Admin mengisi form rekam user dengan benar kemudian melanjutkan proses ubah	Klik “simpan”	Menyimpan isian dan menampilkan pesan user	Sukses

			telah ditambahkan	
3	Admin berada pada form ubah user namun membatalkan proses ubah	Klik “batal”	Kembali ke halaman user	Sukses
4	Terdapat field yang belum diisi oleh Admin pada form ubah user	Klik “simpan”	Menampilkan pesan error	Sukses

D.7. Test Case Hapus User

No	Skenario	Aksi	Hasil	Sukses / Gagal
1	Admin berada pada halaman user	Klik “hapus”	Menampilkan konfirmasi hapus user	Sukses
2	Admin berada pada halaman konfirmasi kemudian melanjutkan proses penghapusan	Klik “ya”	Menghapus user dan menampilkan pesan penghapusan berhasil	Sukses
3	Admin berada pada halaman konfirmasi kemudian membatalkan proses penghapusan	Klik “tidak”	Kembali ke halaman user	Sukses

D-6

D.8. Test Case Reset Password User

No	Skenario	Aksi	Hasil	Sukses / Gagal
1	Admin berada pada halaman user	Klik “permintaan reset password”	Menampilkan daftar user yang meminta reset password	Sukses
2	Admin memilih salah satu dari daftar user yang meminta reset password	Klik “reset”	Menampilkan konfirmasi	Sukses
3	Admin berada pada halaman konfirmasi kemudian melanjutkan proses reset	Klik “ya”	Merest password dan kembali ke daftar permintaan reset	Sukses
4	Admin berada pada halaman konfirmasi namun membatalkan proses reset	Klik “tidak”	Kembali ke daftar permintaan reset	Sukses

D.9. Test Case Rekam Universitas

No	Skenario	Aksi	Hasil	Sukses / Gagal
1	Admin berada pada halaman universitas	Klik “rekam”	Menampilkan form rekam universitas	Sukses
2	Admin mengisi form rekam universitas dengan	Klik “simpan”	Menyimpan data dan	Sukses

	benar kemudian melanjutkan proses rekam		menampilkan pesan perekaman berhasil	
3	Admin berada pada form rekam universitas namun membatalkan proses rekam	Klik “batal”	Kembali ke halaman universitas	Sukses
4	Terdapat field yang kosong oleh Admin pada form rekam universitas	Klik “simpan”	Menampilkan pesan error	Sukses

D.10. Test Case Ubah Universitas

No	Skenario	Aksi	Hasil	Sukses / Gagal
1	Admin berada pada halaman universitas	Klik “ubah”	Menampilkan form perubahan data universitas	Sukses
2	Admin mengisi form perubahan kemudian melanjutkan proses ubah	Klik “simpan”	Menyimpan data dan menampilkan pesan perubahan berhasil	Sukses
3	Admin berada pada form perubahan data namun membatalkan proses ubah	Klik “batal”	Kembali ke halaman universitas	Sukses
4	Terdapat field yang kosong pada form perubahan data universitas	Klik “simpan”	Menampilkan pesan error	Sukses

D.11. Test Case Hapus Universitas

No	Skenario	Aksi	Hasil	Sukses / Gagal
1	Admin berada pada halaman universitas	Klik “hapus”	Menampilkan konfirmasi	Sukses
2	Admin berada pada halaman konfirmasi kemudian melanjutkan proses penghapusan	Klik “ya”	Menghapus data dan menampilkan pesan penghapusan berhasil	Sukses
3	Admin berada pada halaman konfirmasi namun membatalkan proses penghapusan	Klik “tidak”	Kembali ke halaman universitas	Sukses

D.12. Test Case Rekam Jurusan

No	Skenario	Aksi	Hasil	Sukses / Gagal
1	Admin berada pada halaman jurusan	Memilih universitas	Menampilkan daftar jurusan	Sukses
2	Admin melihat daftar jurusan kemudian melanjutkan proses rekam jurusan	Klik “rekam”	Menampilkan form rekam jurusan	Sukses
3	Admin mengisi form rekam jurusan dengan benar kemudian melanjutkan proses rekam	Klik “simpan”	Menyimpan data dan menampilkan pesan perekaman berhasil	Sukses

4	Admin berada pada form rekam jurusan namun membatalkan proses rekam	Klik “batal”	Kembali ke halaman universitas	Sukses
5	Terdapat field yang kosong pada form rekam jurusan	Klik “simpan”	Menampilkan pesan error	Sukses
6	Admin mengakses halaman universitas tetapi belum memilih universitas		Menampilkan pesan untuk memilih universitas	Sukses

D.13. Test Case Ubah Jurusan

No	Skenario	Aksi	Hasil	Sukses / Gagal
1	Admin berada pada halaman jurusan	Klik “ubah”	Menampilkan form perubahan data universitas	Sukses
2	Admin mengisi form perubahan dengan benar kemudian melanjutkan proses ubah	Klik “simpan”	Menyimpan data dan menampilkan pesan perubahan berhasil	Sukses
3	Admin berada pada form perubahan data namun membatalkan proses ubah	Klik “batal”	Kembali ke halaman jurusan	Sukses
4	Terdapat field yang kosong pada form perubahan data universitas	Klik “simpan”	Menampilkan pesan error	Sukses

D.14. Test Case Hapus Jurusan

No	Skenario	Aksi	Hasil	Sukses / Gagal
1	Admin berada pada halaman jurusan	Klik “hapus”	Menampilkan konfirmasi	Sukses
2	Admin berada pada halaman konfirmasi kemudian melanjutkan proses penghapusan	Klik “ya”	Menghapus data dan menampilkan pesan penghapusan berhasil	Sukses
3	Admin berada pada halaman konfirmasi namun membatalkan proses penghapusan	Klik “tidak”	Kembali ke halaman jurusan	Sukses

D.15. Test Case Membuka Pendaftaran

No	Skenario	Aksi	Hasil	Sukses / Gagal
1	Admin berada pada halaman seleksi administrasi	Klik “pembukaan pendaftaran”	Menampilkan form pembukaan pendaftaran	Sukses
2	Admin mengisi form pembukaan pendaftaran dengan benar kemudian melanjutkan proses pembukaan pendaftaran	Klik “simpan”	Menyimpan data dan menampilkan pesan perekaman berhasil	Sukses
3	Admin berada pada form pembukaan	Klik “batal”	Kembali ke halaman	Sukses

	pendaftaran namun membatalkan proses		seleksi administrasi	
4	Terdapat field yang kosong pada form pembukaan pendaftaran	Klik “simpan”	Menampilkan pesan error	Sukses
5	Periode sudah pernah dibuka	Klik “simpan”	Menampilkan pesan error	Sukses

D.16. Test Case Tambah Detail Pembukaan Pendaftaran

No	Skenario	Aksi	Hasil	Sukses / Gagal
1	Admin berada pada halaman seleksi administrasi	Klik “detail”	Menampilkan daftar detail pembukaan pendaftaran	Sukses
2	Admin melihat daftar detail pembukaan pendaftaran kemudian melanjutkan proses penambahan detail	Klik “tambah”	Menampilkan form detail pembukaan pendaftaran	Sukses
3	Admin mengisi form detail pembukaan pendaftaran dengan benar kemudian melanjutkan proses penambahan	Klik “simpan”	Menyimpan data dan menampilkan pesan perekaman berhasil	Sukses
4	Admin berada pada form detail pembukaan pendaftaran namun membatalkan proses	Klik “batal”	Kembali ke halaman seleksi administrasi	Sukses
5	Terdapat field yang kosong pada form detail pembukaan pendaftaran	Klik “simpan”	Menampilkan pesan error	Sukses

D.17. Test Case Hapus Detail Pembukaan Pendaftaran

No	Skenario	Aksi	Hasil	Sukses / Gagal
1	Admin berada pada halaman seleksi administrasi	Klik “detail”	Menampilkan daftar detail pembukaan pendaftaran	Sukses
2	Admin melihat daftar detail pendaftaran kemudian melanjutkan proses hapus	Klik “hapus”	Menampilkan konfirmasi	Sukses
3	Admin berada pada halaman konfirmasi kemudian melanjutkan proses penghapusan detail	Klik “ya”	Menghapus data dan menampilkan pesan penghapusan berhasil	Sukses
4	Admin berada pada halaman konfirmasi namun membatalkan proses penghapusan	Klik “tidak”	Kembali ke daftar detail pembukaan pendaftaran	Sukses

D.18. Test Case Rekam Surat Pembukaan Pendaftaran

No	Skenario	Aksi	Hasil	Sukses / Gagal
1	Admin berada pada halaman seleksi administrasi	Klik “Surat pembukaan pendaftaran”	Menampilkan halaman surat pembukaan pendaftaran	Sukses
2	Admin melihat daftar surat kemudian	Klik “rekam”	Menampilkan form rekam	Sukses

	melanjutkan proses perekaman		surat pembukaan pendaftaran	
3	Admin mengisi form rekam surat pembukaan dengan benar kemudian melanjutkan perekaman	Klik “kirim”	Menampilkan preview surat	Sukses
4	Admin melihat preview surat kemudian melanjutkan proses perekaman	Klik “simpan”	Menyimpan data dan menampilkan pesan perekaman berhasil	Sukses
5	Admin berada pada form rekam surat namun membatalkan proses perekaman	Klik “batal”	Kembali ke halaman seleksi administrasi	Sukses
6	Periode sudah pernah dilakukan perekaman	Klik “kirim”	Menampilkan pesan error	Sukses

D.19. Test Case Rekam Surat Panggilan Tes Tertulis

No	Skenario	Aksi	Hasil	Sukses / Gagal
1	Admin berada pada halaman tes tertulis	Klik “rekam surat panggilan”	Menampilkan form rekam surat panggilan tes	Sukses
2	Admin mengisi form rekam surat dengan benar kemudian melanjutkan proses perekaman	Klik “kirim”	Menampilkan preview surat	Sukses
3	Admin melihat preview surat kemudian melanjutkan proses perekaman	Klik “simpan”	Menyimpan data dan menampilkan pesan	Sukses

			perekaman berhasil	
4	Admin berada pada form rekam surat namun membatalkan proses perekaman	Klik “batal”	Kembali ke halaman tes tertulis	Sukses
5	Periode sudah pernah dilakukan perekaman	Klik “kirim”	Menampilkan pesan error	Sukses

D.20. Test Case Rekam Hasil Tes Tertulis

No	Skenario	Aksi	Hasil	Sukses / Gagal
1	Admin berada pada halaman tes tertulis	Klik “rekam hasil tes”	Menampilkan form rekam hasil tes	Sukses
2	Admin mengisi form rekam hasil tes dengan benar kemudian melanjutkan proses perekaman	Klik “simpan”	Menyimpan data dan menampilkan pesan perekaman berhasil	Sukses
3	Admin berada pada form rekam hasil tes namun membatalkan proses perekaman	Klik “batal”	Kembali ke halaman tes tertulis	Sukses
4	List pegawai tidak ditemukan	Klik “rekam hasil tes”	Menampilkan pesan error	Sukses

D.21. Test Case Rekam Surat Panggilan Tes Wawancara

No	Skenario	Aksi	Hasil	Sukses / Gagal
1	Admin berada pada halaman tes wawancara	Klik “rekam surat panggilan”	Menampilkan form rekam surat panggilan tes	Sukses
2	Admin mengisi form rekam surat dengan benar kemudian melanjutkan proses perekaman	Klik “kirim”	Menampilkan preview surat	Sukses
3	Admin melihat preview surat kemudian melanjutkan proses perekaman	Klik “simpan”	Menyimpan data dan menampilkan pesan perekaman berhasil	Sukses
4	Admin berada pada form rekam surat namun membatalkan proses perekaman	Klik “batal”	Kembali ke halaman tes wawancara	Sukses
5	Periode sudah pernah dilakukan perekaman	Klik “kirim”	Menampilkan pesan error	Sukses

D.22. Test Case Rekam Hasil Tes Wawancara

No	Skenario	Aksi	Hasil	Sukses / Gagal
1	Admin berada pada halaman tes wawancara	Klik “rekam hasil tes”	Menampilkan form rekam hasil tes	Sukses
2	Admin mengisi form rekam hasil tes dengan	Klik “simpan”	Menyimpan data dan	Sukses

	benar dan melanjutkan proses perekaman		menampilkan pesan perekaman berhasil	
3	Admin berada pada form rekam hasil tes namun membatalkan proses perekaman	Klik “batal”	Kembali ke halaman tes wawancara	Sukses
4	List pegawai tidak ditemukan	Klik “rekam hasil tes”	Menampilkan pesan error	Sukses

D.23. Test Case Rekam Surat Pengumuman

No	Skenario	Aksi	Hasil	Sukses / Gagal
1	Admin berada pada halaman pengumuman	Klik “rekam surat pengumuman”	Menampilkan form rekam surat pengumuman	Sukses
2	Admin mengisi form rekam surat dengan benar kemudian melanjutkan proses perekaman	Klik “kirim”	Menampilkan preview surat	Sukses
3	Admin melihat preview surat kemudian melanjutkan proses perekaman	Klik “simpan”	Menyimpan data dan menampilkan pesan perekaman berhasil	Sukses
4	Admin berada pada form rekam surat namun membatalkan proses perekaman	Klik “batal”	Kembali ke halaman pengumuman	Sukses
5	Periode sudah pernah dilakukan perekaman	Klik “kirim”	Menampilkan pesan error	Sukses

D.24. Test Case Mendaftar

No	Skenario	Aksi	Hasil	Sukses / Gagal
1	Pegawai berada pada halaman pendaftaran	Klik “daftar”	Menampilkan list hasil seleksi administrasi dan pesan lolos	Sukses
2	Pegawai telah terdaftar	Klik “daftar”	Menampilkan pesan error	Sukses
3	Melewati batas waktu pendafratan	Klik “daftar”	Menampilkan pesan error	Sukses
4	Pegawai belum meminta rekomendasi dari atasan	Klik “daftar”	Menampilkan pesan error	Sukses
5	Administrasi pegawai tidak memenuhi syarat	Klik “daftar”	Menampilkan pesan tidak lolos	Sukses

D.25. Test Case Membatalkan Pendaftaran

No	Skenario	Aksi	Hasil	Sukses / Gagal
1	Pegawai berada pada halaman pendaftaran	Klik “pembatalan”	Menampilkan konfirmasi	Sukses
2	Pegawai berada pada halaman konfirmasi kemudian melanjutkan proses pembatalan	Klik “ya”	Menghapus pegawai dari daftar dan menampilkan	Sukses

			pesan pembatalan berhasil	
3	Pegawai belum terdaftar	Klik “pembatalan”	Menampilkan pesan error	Sukses
4	Pegawai berada pada halaman konfirmasi namun membatalkan proses pembatalan	Klik “batal”	Kembali ke halaman pendaftaran	Sukses

D.26. Test Case Melihat Riwayat Pendaftaran

No	Skenario	Aksi	Hasil	Sukses / Gagal
1	Pegawai berada pada halaman pendaftaran	Klik “riwayat pendaftaran”	Menampilkan riwayat pendaftaran	Sukses
2	Data riwayat tidak ditemukan	Klik “riwayat pendaftaran”	Menampilkan pesan error	Sukses

D.27. Test Case Rekam Permintaan Rekomendasi

No	Skenario	Aksi	Hasil	Sukses / Gagal
1	Pegawai berada pada halaman permintaan rekomendasi	Klik “rekam”	Menampilkan form permintaan rekomendasi	Sukses
2	Pegawai mengisi form permintaan	Klik “kirin”	Menyimpan data dan	Sukses

	rekomendasi dengan benar kemudian melanjutkan proses permintaan		menampilkan pesan perekaman berhasil	
3	Permintaan rekomendasi sudah pernah dilakukan	Klik “kirim”	Menampilkan pesan error	Sukses
4	Pegawai berada pada form permintaan rekomendasi namun membatalkan permintaan	Klik “batal”	Kembali ke halaman permintaan rekomendasi	Sukses

D.28. Test Case Ubah Permintaan Rekomendasi

No	Skenario	Aksi	Hasil	Sukses / Gagal
1	Pegawai berada pada halaman permintaan rekomendasi	Klik “ubah”	Menampilkan form perubahan permintaan rekomendasi	Sukses
2	Pegawai mengisi form perubahan dengan benar kemudian melanjutkan proses ubah	Klik “kirim”	Menyimpan data dan menampilkan pesan perubahan berhasil	Sukses
3	Permintaan rekomendasi sudah diproses atasan	Klik “kirim”	Menampilkan pesan error	Sukses
4	Pegawai berada pada form perubahan namun membatalkan proses ubah	Klik “batal”	Kembali ke halaman permintaan rekomendasi	Sukses

D.29. Test Case Hapus Permintaan Rekomendasi

No	Skenario	Aksi	Hasil	Sukses / Gagal
1	Pegawai berada pada halaman permintaan rekomendasi	Klik “hapus”	Menampilkan konfirmasi	Sukses
2	Pegawai berada pada halaman konfirmasi kemudian melanjutkan proses penghapusan	Klik “ya”	Menghapus data dan menampilkan pesan penghapusan berhasil	Sukses
3	Permintaan rekomendasi sudah diproses atasan	Klik “kirim”	Menampilkan pesan error	Sukses
4	Pegawai berada pada halaman konfirmasi namun membatalkan proses penghapusan	Klik “tidak”	Kembali ke halaman permintaan rekomendasi	Sukses

D.30. Test Case Melihat Permintaan Rekomendasi

No	Skenario	Aksi	Hasil	Sukses / Gagal
1	Atasan Langsung berada pada halaman utama atasan langsung	Klik “rekomendasi”	Menampilkan daftar pegawai pada halaman rekomendasi	Sukses
2	Tidak ada permintaan rekomendasi	Klik “rekomendasi”	Menampilkan pesan error	Sukses

D.31. Test Case Rekam Rekomendasi

No	Skenario	Aksi	Hasil	Sukses / Gagal
1	Atasan langsung berada pada halaman rekomendasi	Klik “rekam”	Menampilkan form rekam rekomendasi	Sukses
2	Atasan mengisi form rekam rekomendasi dengan benar kemudian melanjutkan proses perekaman	Klik “simpan”	Menyimpan data dan menampilkan pesan perekaman berhasil	Sukses
3	Terdapat field yang kosong pada form rekam rekomendasi	Klik “simpan”	Menampilkan pesan error	Sukses
4	Atasan Langsung berada pada form rekam rekomendasi namun membatalkan proses rekam	Klik “batal”	Kembali ke halaman rekomendasi	Sukses

D.32. Test Case Ubah Rekomendasi

No	Skenario	Aksi	Hasil	Sukses / Gagal
1	Atasan langsung berada pada halaman rekomendasi	Klik “ubah”	Menampilkan form ubah rekomendasi	Sukses
2	Atasan mengisi form ubah rekomendasi	Klik “simpan”	Menyimpan data dan	Sukses

	dengan benar kemudian melanjutkan perubahan		menampilkan pesan perubahan berhasil	
3	Terdapat field yang kosong pada form ubah rekomendasi	Klik “simpan”	Menampilkan pesan error	Sukses
4	Atasan langsung berada pada form ubah rekomendasi namun membatalkan proses ubah	Klik “batal”	Kembali ke halaman rekomendasi	Sukses

D.33. Test Case Hapus Rekomendasi

No	Skenario	Aksi	Hasil	Sukses / Gagal
1	Atasan langsung berada pada halaman rekomendasi	Klik “hapus”	Menampilkan konfirmasi	Sukses
2	Atasan berada pada halaman konfirmasi kemudian melanjutkan proses penghapusan	Klik “ya”	Menghapus data dan menampilkan pesan penghapusan berhasil	Sukses
3	Belum dilakukan perekaman rekomendasi	Klik “ya”	Menampilkan pesan error	Sukses
4	Atasan berada pada halaman konfirmasi namun membatalkan proses penghapusan	Klik “tidak”	Kembali ke halaman rekomendasi	Sukses

D.34. Test Case Tolak Rekomendasi

No	Skenario	Aksi	Hasil	Sukses / Gagal
1	Atasan langsung berada pada halaman rekomendasi	Klik “tolak”	Menampilkan konfirmasi	Sukses
2	Atasan langsung berada pada halaman konfirmasi dan melanjutkan proses penolakan	Klik “ya”	Merubah status dan menampilkan pesan penolakan berhasil	Sukses
3	Atasan langsung berada pada halaman konfirmasi dan membatalkan proses penolakan	Klik “tidak”	Kembali ke halaman rekomendasi	Sukses

D-24

Halaman ini sengaja dikosongkan

