



**PENERAPAN ALGORITMA WINNOWER PADA
SISTEM REKOMENDASI PENENTUAN DOSEN PEMBIMBING
SKRIPSI (STUDI KASUS : PRODI SISTEM INFORMASI
UNIVERSITAS JEMBER)**

SKRIPSI

Oleh

Jarwati

1224101027

PROGRAM STUDI SISTEM INFORMASI

UNIVERSITAS JEMBER

2016



**PENERAPAN ALGORITMA WINNOWER PADA
SISTEM REKOMENDASI PENENTUAN DOSEN PEMBIMBING
SKRIPSI (STUDI KASUS : PRODI SISTEM INFORMASI
UNIVERSITAS JEMBER)**

PROPOSAL SKRIPSI

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat untuk menyelesaikan Program Studi Sistem Informasi (SI) dan mendapatkan gelar Sarjana Komputer

Oleh

Jarwati

122410101027

PROGRAM STUDI SISTEM INFORMASI

UNIVERSITAS JEMBER

2016

PERSEMBAHAN

Skripsi ini saya persembahkan untuk :

1. Allah SWT, yang telah memberikan kelancaran dan kemudahan dalam menyelesaikan skripsi ini;
2. Ibunda Darmiasih serta Ayahanda Marlan tercinta atas segala doa, dukungan, kesabaran, keikhlasan, limpahan kasih sayang, dan motivasi yang luar biasa;
3. Kakak saya tercinta Drh. Yayuk Kholifah;
4. Saudara-saudaraku beserta seluruh keluarga besar;
5. Denni Rudiyanto beserta keluarga yang selalu memberikan dukungan;
6. Sahabat kontrakan Puri Bunga Nirwana yaitu Esa, Dellia, Seftya, Ara, Gau, Dwi, Lintang, sahabat seperjuangan yaitu Yulianita, Kurnia, Dita dan kakak angkatan Khoirul Anwar yang selalu memberi dukungan;
7. Teman-teman seperjuangan kuliah di PSSI yang bersama berjuang menuntut ilmu;
8. Guru-guruku taman kanak-kanak hingga perguruan tinggi;
9. Almamater Program Studi Sistem Informasi Universitas Jember

MOTO

“Sesungguhnya bersama kesukaran itu ada keringanan. Karena itu bila kau sudah selesai (mengerjakan yang lain). Dan berharaplah kepada Tuhanmu.”

(Q.S Al Insyirah : 6-8)



PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Jarwati

NIM : 122410101027

menyatakan dengan sesungguhnya bahwa karya tulis ilmiah dengan judul “Penerapan Algoritma *Winnowing* Pada Sistem Rekomendasi Penentuan Dosen Pembimbing Skripsi (Studi Kasus : Prodi Sistem Informasi Universitas Jember)” adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi manapun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenar-benarnya, tanpa ada tekanan dan paksaan dari pihak manapun, serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, 14 Juni 2016

Yang menyatakan,

Jarwati

NIM 122410101027

SKRIPSI

**PENERAPAN ALGORITMA WINNOWER PADA
SISTEM REKOMENDASI PENENTUAN DOSEN PEMBIMBING
SKRIPSI (STUDI KASUS : PRODI SISTEM INFORMASI
UNIVERSITAS JEMBER)**

Oleh

Jarwati

NIM 122410101027

Pembimbing

Dosen Pembimbing Utama : Drs. Antonius Cahya P, M.App., Sc., Ph.D

Dosen Pembimbing Pendamping : Windy Eka Yulia R, S.Kom, MT

PENGESAHAN PEMBIMBING

Skripsi berjudul “Penerapan Algoritma *Winnowing* Pada Sistem Rekomendasi Penentuan Dosen Pembimbing Skripsi (Studi Kasus : Prodi Sistem Informasi Universitas Jember)”, telah diuji dan disahkan pada :

hari, tanggal :

tempat : Program Studi Sistem Informasi Universitas Jember.

Disetujui oleh :

Pembimbing I

Pembimbing II

Drs. Antonius Cahya P, M.App. Sc., Ph.D
NIP 196909281993021001

Windy Eka Yulia R, S.Kom, MT
NIP 198403052010122002

PENGESAHAN

Skripsi berjudul “Penerapan Algoritma *Winnowing* Pada Sistem Rekomendasi Penentuan Dosen Pembimbing Skripsi (Studi Kasus : Prodi Sistem Informasi Universitas Jember)”, telah diuji dan disahkan pada :

hari, tanggal :

tempat : Program Studi Sistem Informasi Universitas Jember.

Disetujui oleh :

Penguji I

Penguji II

(.....)

(.....)

NIP.....

NIP.....

Mengesahkan,
Ketua Program Studi Sistem Informasi
Universitas Jember,

Prof. Drs. Slamin, M. Comp Sc, Ph.D
NIP. 196704201992011001

RINGKASAN

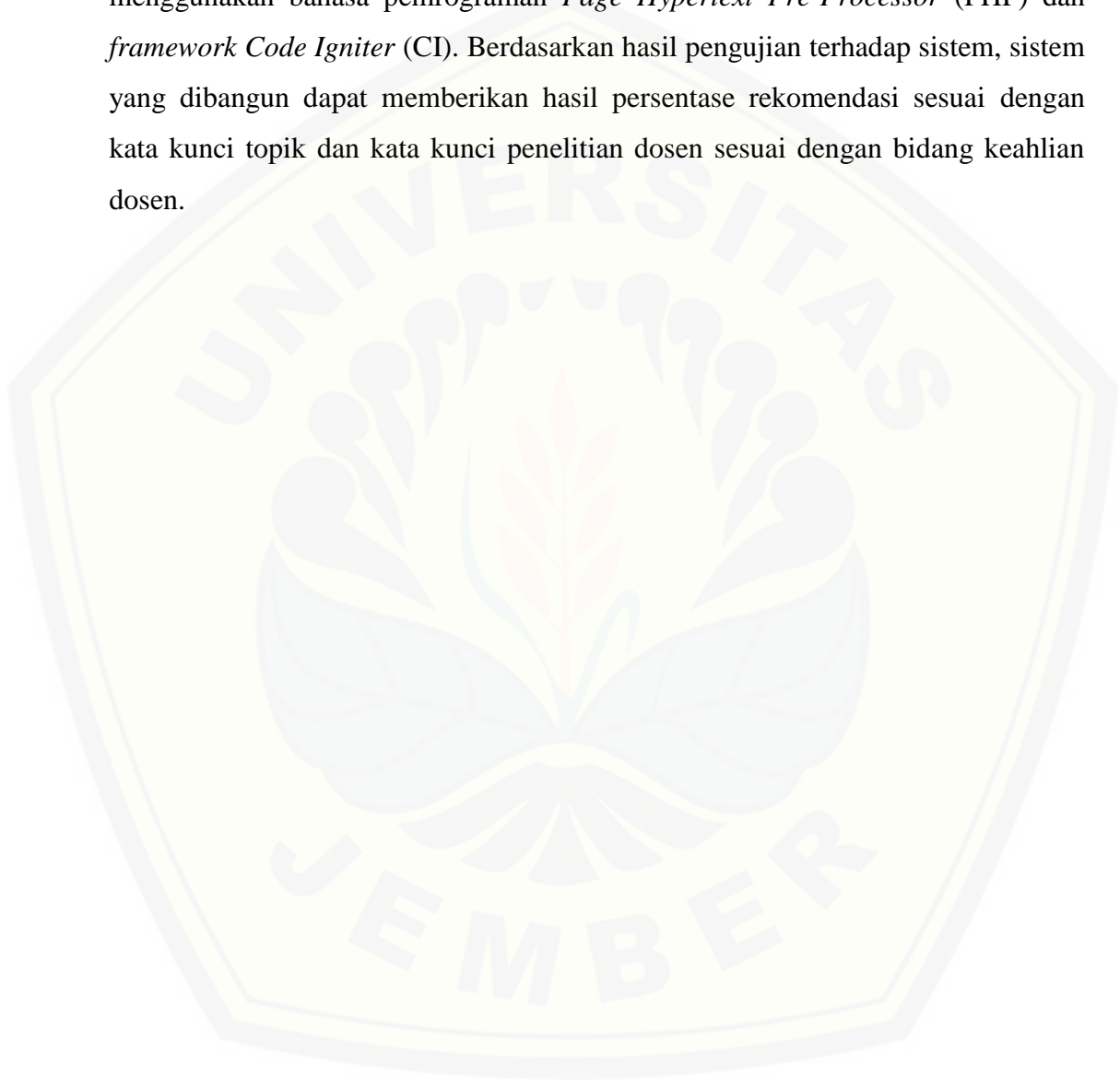
Penerapan Algoritma *Winnowing* Pada Sistem Rekomendasi Penentuan Dosen Pembimbing Skripsi (Studi Kasus : Prodi Sistem Informasi Universitas Jember); Jarwati; 122410101027; 175 halaman; Program Studi Sistem Informasi Universitas Jember

Skripsi merupakan tugas akhir bagi mahasiswa Program Studi (Prodi) Sistem Informasi untuk mendapatkan gelar sarjana sesuai dengan bidang keahlian yang dituliskan dalam bentuk karya tulis ilmiah. Penyusunan karya tulis ilmiah membutuhkan bimbingan dosen pembimbing untuk menghasilkan karya tulis yang berkualitas. Penentuan dosen pembimbing harus dilakukan dengan benar oleh komisi bimbingan sesuai dengan kriteria dan aturan yang berlaku di Prodi Sistem Informasi. Meningkatnya jumlah mahasiswa semester akhir mengakibatkan proses administrasi pendaftaran tugas akhir melambat. Oleh karena itu untuk membantu komisi bimbingan dibutuhkan suatu sistem rekomendasi yang dapat membantu memilih dosen pembimbing skripsi yang tepat sesuai dengan keahlian dosen dan topik yang diambil oleh mahasiswa.

Pada penelitian ini dibangun sistem rekomendasi penentuan dosen pembimbing menggunakan metode *Winnowing*. Sistem dibangun dengan mencocokkan antara keyword topik yang diambil mahasiswa dengan keyword penelitian yang dilakukan oleh dosen pembimbing yang dipilih dengan mempertimbangkan jabatan fungsional dosen. Pencocokan dapat dilakukan dengan menghitung *similarity* dari kata kunci topik mahasiswa dengan kata kunci penelitian dosen. Hasil rekomendasi nama dosen akan ditampilkan dengan persentase rata-rata kemiripan yang diurutkan dari yang terbesar. Sistem rekomendasi ini dibangun berbasis website agar dapat digunakan dengan mudah oleh akademik, kombi dan juga dosen.

Penelitian dilakukan dalam tiga tahap yaitu tahap pengumpulan data, tahap analisis dan tahap pengembangan sistem. Tahap pengumpulan data dilakukan melalui studi literatur dan wawancara kepada narasumber. Tahap analisis dilakukan

dengan menganalisis data yang didapatkan ke dalam metode sistem. Tahap pengembangan sistem dilakukan dengan mengadopsi model *waterfall* yang dimulai dari analisis kebutuhan fungsional dan non-fungsional sistem, pembuatan desain sistem, penulisan kode program dan menguji sistem. Sistem ini dibangun menggunakan bahasa pemrograman *Page Hypertext Pre-Processor* (PHP) dan *framework Code Igniter* (CI). Berdasarkan hasil pengujian terhadap sistem, sistem yang dibangun dapat memberikan hasil persentase rekomendasi sesuai dengan kata kunci topik dan kata kunci penelitian dosen sesuai dengan bidang keahlian dosen.





PRAKATA

Puji syukur kehadiran Allah SWT atas segala rahmat, karunia dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Penerapan Algoritma *Winnowing* Pada Sistem Rekomendasi Penentuan Dosen Pembimbing Skripsi (Studi Kasus : Prodi Sistem Informasi Universitas Jember)”. Skripsi ini disusun untuk melengkapi salah satu syarat menyelesaikan pendidikan Strata 1 (S1) pada Program Studi Sistem Informasi Universitas Jember.

Dalam proses penyusunan skripsi ini tidak terlepas dari bantuan berbagai pihak. Oleh karena itu dalam kesempatan ini penulis ingin mengucapkan terima kasih kepada:

1. Drs. Antonius Cahya P, M.App. Sc., Ph.D sebagai Dosen Pembimbing Utama dan Windi Eka Yulia Retnani, S. Kom., M.T sebagai Dosen Pembimbing Pendamping yang telah meluangkan waktu, pikiran dan perhatian dalam penulisan skripsi ini;
2. Prof. Drs. Slamir, M. Com. Sc., Ph.D selaku Ketua Program Studi Sistem Informasi Universitas Jember;
3. Dr. Saiful Bukhori ST., M.Kom selaku Dosen Pembimbing Akademik yang telah memberikan bimbingan selama penulis menjadi mahasiswa;
4. Seluruh bapak dan ibu dosen beserta staf di Program Studi Sistem Informasi Universitas Jember;
5. Ayah, Ibu, Kakak tercinta Yayuk Kholifah, Adik Tersayang Imam Sholeh yang telah mendukung dalam hal moril dan materil serta selalu memberi memotivasi dan doa yang tanpa putus pada penulis;
6. Sahabat kontrakan Puri Bunga Nirwana yaitu Dellia, Esa, Puput, Ara, Lintang, Ghau, Dwi, Septi dan senior Khoirul Anwar yang selalu memberi semangat dan dukungan;
7. Sahabat Teknik yaitu Denni Rudiyanto, Prasetya, Agus, Anwar, Danard, Yahya, Nanda, Nando, Abdi, Bijak terima kasih atas pertemanan kita selama ini;

8. Keluarga Besar FORMATION semua mahasiswa angkatan 2012 yang telah menjadi keluarga selama menempuh pendidikan S1;
9. Kelompok 130 KKN Desa Rejo Agung Kecamatan Sumberwringin Kabupaten Bondowoso;
10. Semua pihak yang tidak bisa penulis sebutkan satu per satu.

Penulis menyadari penulisan skripsi ini masih jauh dari kesempurnaan, oleh karena itu penulis mengharapkan kritik dan saran dari semua pihak. Akhir kata, penulis berharap semoga skripsi ini dapat memberikan manfaat bagi semua pihak.

Jember, 14 Juni 2016

Penulis

DAFTAR ISI

PROPOSAL SKRIPSI	i
PERSEMBAHAN	ii
MOTO	iii
PERNYATAAN	iv
SKRIPSI	v
PENGESAHAN PEMBIMBING	vi
PENGESAHAN	vii
RINGKASAN	viii
PRAKATA	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xx
DAFTAR LISTING PROGRAM	xxi
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan dan Manfaat.....	3
1.3.1 Tujuan.....	3
1.3.2 Manfaat.....	3
1.4 Batasan Masalah.....	4
1.5 Sistematika Penulisan.....	4
BAB 2. TINJAUAN PUSTAKA	6
2.1 Penelitian Terdahulu.....	6

2.2	Metode String Matching.....	8
2.3	Metode Dokumen <i>Fingerprinting</i>	9
2.4	Algoritma Winnowing.....	9
2.4.1	Definisi Algoritma Winnowing.....	9
2.4.2	Keuntungan Menggunakan Algoritma Winnowing.....	15
2.4.3	Kelemahan Algoritma <i>Winnowing</i>	15
2.5	Persyaratan Administrasi Seorang Dosen Pembimbing.....	16
BAB 3.	METODOLOGI PENELITIAN	18
3.1	Jenis Penelitian.....	18
3.2	Tempat dan Waktu Penelitian	18
3.3	Metode Pengumpulan Data	18
3.4	Metode Analisis Data	19
3.5	Teknik Pengembangan Sistem	20
3.5.1	Analisis Kebutuhan	21
3.5.2	Desain.....	21
3.5.3	Penulisan Kode Program.....	22
3.5.4	Pengujian.....	22
3.5.5	Pemeliharaan	23
BAB 4.	PERANCANGAN SISTEM	24
4.1	Deskripsi Umum Sistem.....	24
4.1.1	SOP (<i>Statement Of Purpose</i>).....	24
4.1.2	Fungsi Sistem.....	24
4.2	Analisis Kebutuhan	25
4.2.1	Kebutuhan Fungsional	26
4.2.2	Kebutuhan Non-Fungsional	26

4.3	Desain Sistem	27
4.3.1	Bussiness Process	27
4.3.2	<i>Use Case</i> Diagram.....	28
4.3.3	<i>Use Case</i> Skenario	32
4.3.4	<i>Activity</i> Diagram.....	39
4.3.5	<i>Sequence</i> Diagram.....	45
4.3.6	<i>Class</i> Diagram	52
4.3.7	Entity Relationship Diagram (ERD)	52
4.4	Implementasi	54
4.5	Pengujian	61
4.5.1	Metode White Box	61
4.5.2	Metode Black Box.....	76
BAB 5.	HASIL DAN PEMBAHASAN	77
5.1	Hasil Penerapan Perhitungan Algoritma Winnowing	77
5.2	Hasil Pembuatan Sistem Rekomendasi Penentuan Dosen Pembimbing Menggunakan Algoritma Winnowing.....	93
5.2.1	Fitur <i>Login</i> sistem	94
5.2.2	Fitur Home Akademik.....	94
5.2.3	Fitur <i>View</i> Pengajuan	95
5.2.4	Fitur Tambah Pengajuan	96
5.2.5	Tampilan Edit Pengajuan Proposal	96
5.2.6	Fitur <i>View</i> Persetujuan	97
5.2.7	Fitur Lengkapi Surat Tugas.....	98
5.2.8	Fitur Rekap Bimbingan Dosen.....	98
5.2.9	Fitur Detail Bimbingan Dosen	99

5.2.10	Fitur <i>View</i> Dosen.....	99
5.2.11	Tampilan Tambah Dosen	100
5.2.12	Fitur Edit Data Dosen.....	101
5.2.13	Fitur Rekomendasi Dosen Pembimbing.....	101
5.2.14	Fitur Detail Bidang Keahlian Dosen	102
5.2.15	Fitur View Keyword Penelitian Dosen	103
5.2.16	Fitur Update Keyword Penelitian.....	103
5.2.17	Fitur <i>View</i> Data Bidang.....	104
5.2.18	Fitur Tambah Bidang Dosen	104
5.2.19	Fitur <i>View</i> Mahasiswa Bimbingan Dosen	105
5.3	Implementasi Algoritma Winnowing Pada Sistem Rekomendasi.....	106
5.4	Pembahasan Pada Sistem Rekomendasi Dosen Pembimbing.....	109
5.4.1	Kelebihan Sistem	110
5.5.2	Keterbatasan Sistem.....	111
BAB 6.	PENUTUP	112
6.1	Kesimpulan.....	112
6.2	Saran.....	113
DAFTAR PUSTAKA	114
LAMPIRAN	115

DAFTAR GAMBAR

Gambar 2.1 <i>Flowchart</i> Algoritma <i>winnowing</i>	10
Gambar 2.2 Perhitungan <i>hash</i> menggunakan <i>rolling hash</i>	12
Gambar 2.3 Penggunaan Irisan Untuk Menghitung Kesamaan Dokumen	14
Gambar 3.1 Diagram Alir Penerapan Algoritma <i>Winnowing</i>	20
Gambar 3.2 <i>Model Waterfall</i> (Sumber: <i>Pressman, 2001</i>).....	20
Gambar 4.1 <i>Bussiness Process</i> Sistem.....	28
Gambar 4.2 <i>Use Case</i> Diagram.....	29
Gambar 4.3 <i>Activity</i> Diagram Menambah Pengajuan Proposal	41
Gambar 4.4 <i>Sequence</i> diagram menambah pengajuan proposal	47
Gambar 4.6 <i>Class</i> Diagram	53
Gambar 4.7 <i>Entity Relationship</i> Diagram	54
Gambar 4.8 Listing <i>program</i> function <i>metodeWinnowing()</i>	63
Gambar 4.9 Listing <i>program</i> function <i>removeSpecialChar()</i>	63
Gambar 4.10 Listing <i>program</i> function <i>split()</i>	63
Gambar 4.11 Listing <i>program</i> function <i>hashSemuaGram()</i>	64
Gambar 4.12 Listing <i>program</i> function <i>hash()</i>	64
Gambar 4.13 Listing <i>program</i> function <i>window()</i>	64
Gambar 4.14 Listing <i>program</i> function <i>windowTerkecil()</i>	64
Gambar 4.15 Diagram alir <i>function</i> <i>metodeWinnowing()</i>	65
Gambar 4.16 Diagram alir <i>function</i> <i>removeSpecialChar()</i>	65
Gambar 4.17 Diagram Alir <i>function</i> <i>split()</i>	65
Gambar 4.18 Diagram Alir <i>function</i> <i>hashSemuaGram()</i>	66
Gambar 4.19 Diagram alir <i>function</i> <i>hash()</i>	66
Gambar 4.20 Diagram alir <i>function</i> <i>window()</i>	66

Gambar 4.21 Diagram alir <i>function</i> windowTerkecil().....	67
Gambar 4.22 Grafik alir <i>function</i> metodeWinnowing().....	67
Gambar 4.23 Grafik alir <i>function</i> removeSpecialChar().....	67
Gambar 4.24 Grafik alir <i>function</i> split()	68
Gambar 4.25 Grafik alir <i>function</i> hashSemuaGram()	68
Gambar 4.26 Grafik alir <i>function</i> hash()	68
Gambar 4.27 Grafik alir <i>function</i> window()	68
Gambar 4.28 Grafik alir <i>function</i> windowTerkecil()	69
Gambar 5.1 Gambaran Proses <i>String Matching</i> pada setiap Keyword.....	79
Gambar 5.2 Tampilan halaman <i>login</i>	94
Gambar 5.3 Tampilan halaman home akademik.....	95
Gambar 5.4 Halaman <i>view</i> pengajuan.....	95
Gambar 5.5 Tampilan fitur tambah pengajuan	96
Gambar 5.6 Tampilan fitur edit pengajuan	97
Gambar 5.7 Fitur halaman persetujuan	97
Gambar 5.8 Fitur lengkapi surat tugas	98
Gambar 5.9 Tampilan rekap bimbingan dosen	99
Gambar 5.10 Tampilan detail bimbingan dosen	99
Gambar 5.11 Tampilan <i>view</i> dosen.....	100
Gambar 5.12 Tampilan tambah dosen	100
Gambar 5.13 Fitur edit data dosen	101
Gambar 5.14 Fitur rekomendasi dosen pembimbing	102
Gambar 5.15 Fitur detail bidang keahlian.....	102
Gambar 5.16 Tampilan <i>view keyword</i> penelitian dosen.....	103
Gambar 5.17 Form <i>update keyword</i> penelitian dosen.....	104
Gambar 5.18 Tampilan View Data Bidang.....	104
Gambar 5.19 Tambah data bidang keahlian dosen	105
Gambar 5.20 Tampilan <i>view</i> mahasiswa bimbingan dosen	106
Gambar 5.21 Kode program proses <i>preprocessing</i>	106
Gambar 5.22 Implementasi kode program pembentukan rangkaian <i>gram</i>	107
Gambar 5.23 kode program menghitung nilai <i>hash</i>	107

Gambar 5.24 Kode program membentuk <i>window</i>	107
Gambar 5.25 Kode program pengambilan nilai fingerprint.....	108
Gambar 5.26 Kode program menghitung persentase keyakinan	109

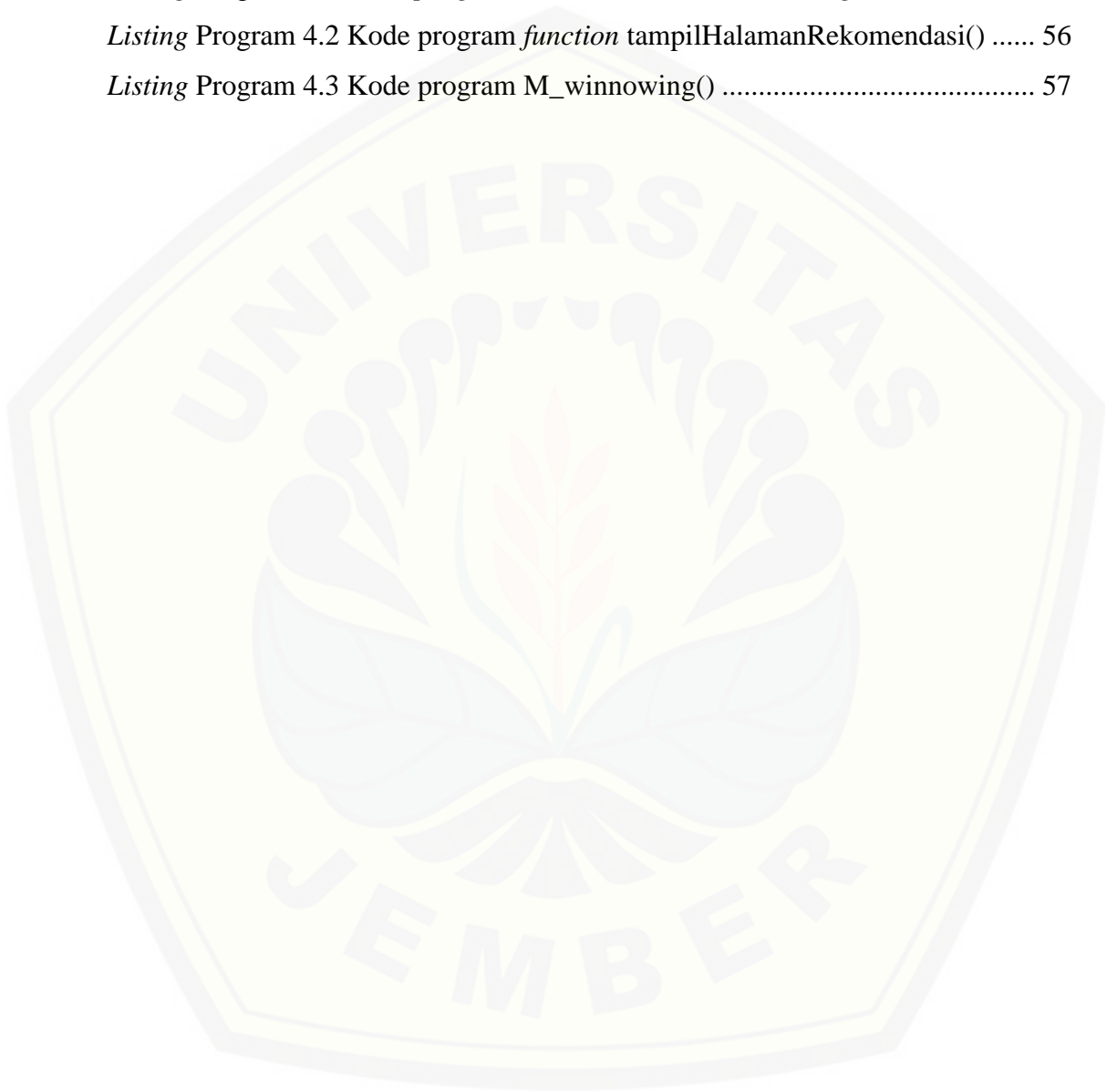


DAFTAR TABEL

Tabel 2.1 Perbandingan Algoritma Rabin-Karp dengan Algoritma Winnowing ...	7
Tabel 2.2 Perhitungan Algoritma <i>winnowing</i> dan <i>jaccard Coeficient</i>	14
Tabel 4.1 Deskripsi pembagian aktor sistem	30
Tabel 4.2 Deskripsi <i>use case</i> sistem.....	30
Tabel 4.3 Skenario menambah pengajuan proposal.....	33
Tabel 4.4 <i>Test case function</i> metode <i>Winnowing()</i>	71
Tabel 4.5 <i>Test case function</i> <i>removeSpecialChar()</i>	71
Tabel 4.6 <i>Test case function</i> <i>split()</i>	72
Tabel 4.7 <i>Test case function</i> <i>hashSemuaGram()</i>	73
Tabel 4.8 <i>Test case function</i> <i>hash()</i>	73
Tabel 4.9 <i>Test case function</i> <i>window()</i>	74
Tabel 4.10 <i>Test case function</i> <i>windowTerkecil()</i>	75
Tabel 5.1 Data pengajuan proposal.....	78
Tabel 5.2 Data Penelitian Dosen	78
Tabel 5.3 <i>Keyword</i> Penelitian Dosen(<i>slamin</i>).....	81
Tabel 5.4 <i>Keyword</i> Penelitian Dosen(<i>slamin</i>).....	82
Tabel 5.5 <i>Keyword</i> Penelitian Dosen (<i>Saiful Bukhori</i>).....	84
Tabel 5.6 Perhitungan Persentase kesamaan “algoritma <i>winnowing</i> ” dengan keyword penelitian dosen.....	85
Tabel 5.7 Perhitungan Persentase kesamaan “sistem rekomendasi” dengan keyword penelitian dosen.....	87
Tabel 5.8 Perhitungan Persentase kesamaan “penentuan dosen pembimbing” dengan keyword penelitian dosen	90

DAFTAR LISTING PROGRAM

Listing Program 4.1 Kode program function metode Winnowing()..... 55
Listing Program 4.2 Kode program function tampilHalamanRekomendasi() 56
Listing Program 4.3 Kode program M_winning() 57



BAB 1. PENDAHULUAN

Bab ini merupakan bab awal dari laporan tugas akhir. Pada bab ini akan dibahas tentang latar belakang, perumusan masalah, tujuan dan manfaat, batasan masalah, dan sistematika penulisan.

1.1 Latar Belakang

Skripsi merupakan tugas akhir bagi mahasiswa untuk mendapatkan gelar sarjana sesuai dengan bidang keahlian yang dituliskan dalam bentuk karya tulis ilmiah. Penulisan karya tulis ilmiah membutuhkan bimbingan dosen pembimbing dari awal penyusunan sampai menghasilkan karya tulis yang berkualitas. Dosen pembimbing berperan penting dalam proses penulisan karya tulis ilmiah sebagai fasilitator, motivator, dan pengarah yang baik sehingga skripsi dapat selesai dan siap untuk diujikan. Penentuan dosen pembimbing harus dilakukan dengan benar sesuai dengan kriteria dan aturan yang berlaku sehingga hasil yang dicapai lebih baik karena penguasaan konsep dan teori yang matang.

Menurut prosedur kerja mengenai tugas akhir di Prodi Sistem Informasi Universitas Jember terdapat tata cara yang harus dilakukan pada proses pendaftaran judul tugas akhir, yaitu mahasiswa harus melampirkan kartu hasil studi, kartu rencana studi, draft proposal tugas akhir kepada bidang akademik (SOP Program Studi Sistem Informasi Universitas Jember, 2015). Data mahasiswa dari bidang akademik kemudian diserahkan pada ketua komisi bimbingan (kombi), yang selanjutnya dilakukan rapat komisi bimbingan untuk menentukan pembimbing tugas akhir berdasarkan kompetensi, beban kerja dan pengajuan mahasiswa. Namun masalah dapat terjadi apabila semakin banyak mahasiswa yang mengajukan judul dan penelitian yang dilakukan tiap dosen, maka kombi harus mencocokkan satu per satu keahlian dosen dengan topik yang diambil oleh mahasiswa.

Saat ini program studi sistem informasi sedang merintis untuk membuka prodi baru yaitu Teknologi Informasi agar dapat berkembang menjadi sebuah

fakultas, maka dapat dipastikan jumlah mahasiswa akan semakin meningkat dan penetapan dosen pembimbing oleh kombi tidak akan sesederhana saat mahasiswa semester akhir masih sedikit. Oleh karena itu dibutuhkan suatu sistem rekomendasi yang dapat membantu kombi memilih dosen pembimbing skripsi yang tepat dengan memasukkan kata kunci topik dan judul yang diajukan mahasiswa.

Sistem rekomendasi pemilihan dosen pembimbing skripsi dibuat untuk memilih dosen pembimbing yang sesuai dengan kompetensi, beban kerja dan topik yang dikerjakan mahasiswa. Berdasarkan hal tersebut, terdapat beberapa kriteria yang digunakan dalam pemilihan dosen pembimbing diantaranya yaitu bidang ilmu atau kompetensi, kata kunci penelitian dosen, dan jabatan. Hasil keputusan diperoleh dengan mencocokkan antara kata kunci topik yang diambil mahasiswa dengan kata kunci penelitian yang dilakukan oleh dosen pembimbing dan bidang ilmu yang kemudian dipilih dengan mempertimbangkan jabatan dosen. Salah satu metode yang tepat untuk diterapkan dalam pencocokan tersebut adalah algoritma *winnowing*. Algoritma *winnowing* menerapkan konsep *string matching* untuk mencocokkan suatu *string* dengan *string* lainnya menggunakan teknik hashing. Pencocokan dapat dilakukan dengan menghitung *similarity* dari kata kunci penelitian dosen dan kata kunci topik yang dibandingkan.

Sistem rekomendasi pemilihan dosen pembimbing dibangun menggunakan algoritma *winnowing*. Algoritma *winnowing* diasumsikan dapat melakukan pencocokan pola string antara kata kunci topik mahasiswa dengan kata kunci penelitian dosen. Implementasi algoritma *winnowing* pada sistem pendukung keputusan ini dapat memberikan rekomendasi yang tepat sesuai dengan aturan yang berlaku pada prodi Sistem Informasi.

1.2 Rumusan Masalah

Berdasarkan dari beberapa permasalahan yang telah diuraikan diatas, maka dapat diambil rumusan masalah sebagai berikut :

- a. Bagaimana implementasi algoritma *winnowing* pada pemilihan dosen pembimbing?
- b. Bagaimana merancang dan membangun sistem rekomendasi pemilihan dosen pembimbing skripsi menggunakan algoritma *winnowing*?

1.3 Tujuan dan Manfaat

Tujuan dan manfaat merupakan jawaban dari rumusan masalah yang telah diuraikan di atas serta tujuan yang ingin dicapai pada penelitian ini.

1.3.1 Tujuan

Tujuan yang ingin dicapai dari penelitian ini adalah :

- a. Mengimplementasikan algoritma *winnowing* pada sistem rekomendasi penentuan dosen pembimbing skripsi
- b. Merancang dan membangun sebuah sistem rekomendasi penentuan dosen pembimbing skripsi
- c. Membantu komisi bimbingan melakukan pemilihan dosen pembimbing skripsi sesuai dengan aturan untuk membantu mahasiswa dalam proses penulisan skripsi

1.3.2 Manfaat

Manfaat dari penelitian ini adalah :

- a. Manfaat Akademis

Hasil penelitian ini diharapkan dapat memberikan kontribusi bagi komisi bimbingan berhubungan dengan judul penelitian ini. Selain itu, hasil penelitian ini merupakan suatu upaya untuk meminimalisir kerja kombi dan akademik prodi sistem informasi

1.4 Batasan Masalah

Agar tidak terjadi penyimpangan dalam proses penelitian dan pembangunan sistem rekomendasi dosen pembimbing menggunakan algoritma *Winnowing*, maka ditetapkan beberapa batasan permasalahan. Adapun batasan masalah yang timbul adalah :

- a. Sistem yang dibangun berbasis website.
- b. Sistem ini hanya digunakan untuk memilih dosen pembimbing skripsi pada program studi sistem informasi universitas jember dengan mencocokkan kata kunci menggunakan algoritma *winnowing*.
- c. Sistem yang akan dibangun hanya memberikan rekomendasi dosen sesuai dengan aturan yang berlaku tanpa mempertimbangkan alasan pengajuan dosen oleh mahasiswa.

1.5 Sistematika Penulisan

Adapun Sistematika penulisan tugas akhir ini adalah sebagai berikut:

- a. Pendahuluan

Bab kesatu ini memuat uraian tentang latar belakang, perumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika penulisan skripsi yang masing-masing tertuang secara eksplisit dalam subbab tersendiri.

- b. Tinjauan Pustaka

Bab ini memaparkan tinjauan terhadap hasil-hasil penelitian terdahulu berkaitan dengan masalah yang dibahas, landasan materi dan konsep prediksi, dan kajian teori metode analisis data yang berkaitan dengan masalah dalam penelitian.

- c. Metodologi Penelitian

Bab ini menguraikan tentang tempat dan waktu penelitian, metode penelitian, metode pengumpulan data, metode analisis data, dan teknik pengembangan sistem yang digunakan dalam penelitian.

d. Perancangan Sistem

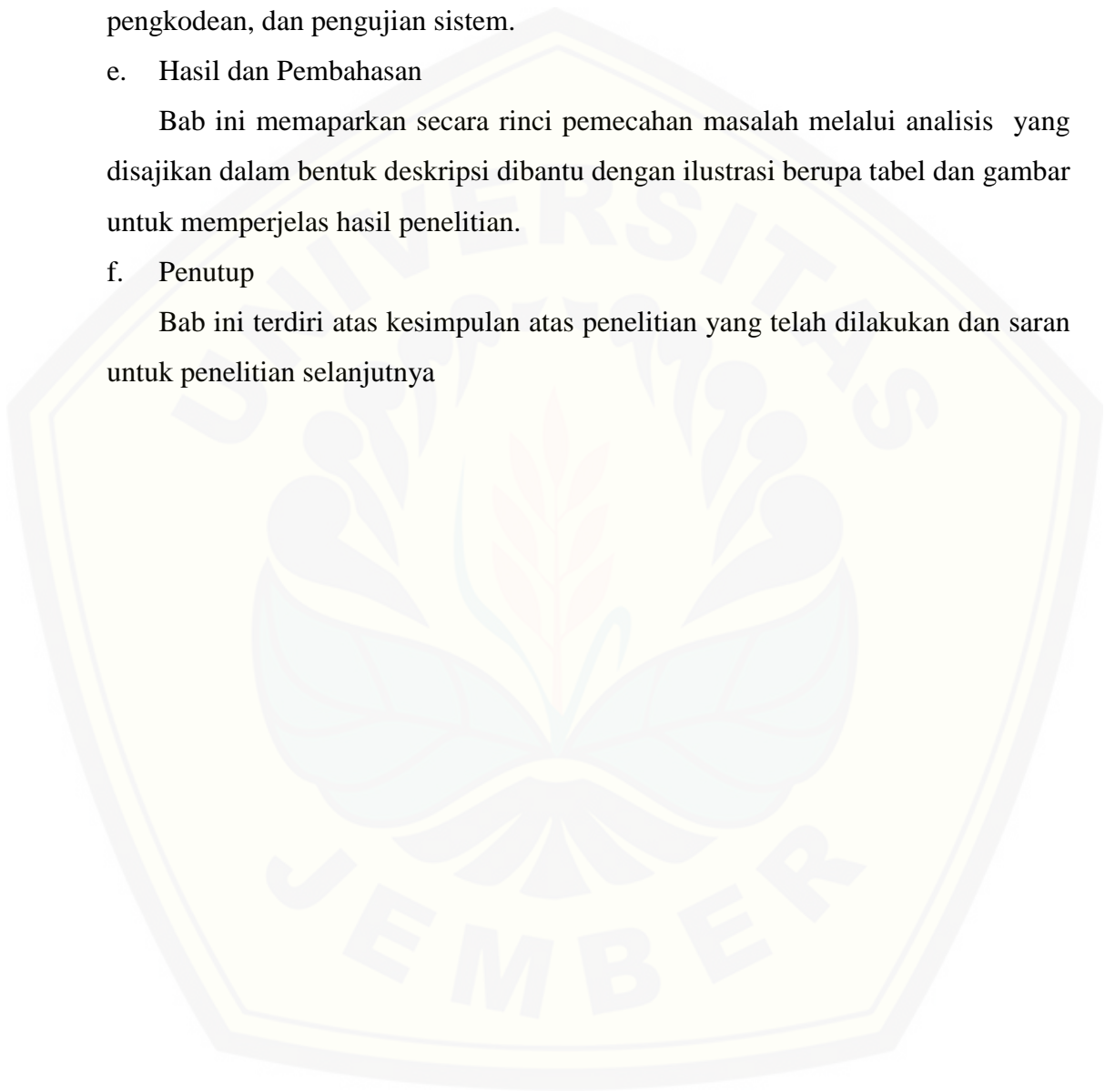
Bab ini berisi uraian tentang langkah-langkah yang ditempuh dalam proses menganalisis dan merancang sistem yang hendak dibangun meliputi desain, pengkodean, dan pengujian sistem.

e. Hasil dan Pembahasan

Bab ini memaparkan secara rinci pemecahan masalah melalui analisis yang disajikan dalam bentuk deskripsi dibantu dengan ilustrasi berupa tabel dan gambar untuk memperjelas hasil penelitian.

f. Penutup

Bab ini terdiri atas kesimpulan atas penelitian yang telah dilakukan dan saran untuk penelitian selanjutnya



BAB 2. TINJAUAN PUSTAKA

Pada bagian ini dipaparkan tinjauan yang berkaitan dengan masalah yang dibahas, kajian teori yang berkaitan dengan masalah, dan juga penelitian-penelitian terdahulu.

2.1 Penelitian Terdahulu

Penelitian pertama dilakukan oleh (Salam, 2015) tentang sistem rekomendasi penentuan dosen pembimbing tugas akhir yang diterapkan menggunakan algoritma Rabin-Karp. Sistem yang dikembangkan bertujuan memberikan rekomendasi kepada mahasiswa mengenai dosen yang sesuai untuk dipilih menjadi pembimbing tugas akhir. Algoritma Rabin-Karp digunakan dalam pencocokan judul yang diambil mahasiswa dengan judul penelitian dosen yang memiliki tingkat variasi beragam. Tingkat keakuratan yang dihasilkan tergantung pada jumlah penelitian yang pernah dilakukan dosen. Semakin banyak judul penelitian, semakin besar peluang menghasilkan rekomendasi yang tepat. Namun penerapan metode Rabin-Karp tanpa modifikasi membutuhkan waktu lama apabila string yang dicocokkan bervariasi. Untuk menutupi kelemahan tersebut, akan lebih baik jika pencocokan dilakukan menggunakan algoritma *winnowing* yang dapat menangani variasi kata kunci yang beragam.

Pada penelitian kedua, algoritma *winnowing* telah digunakan sebelumnya pada sistem deteksi plagiarisme yang dikembangkan oleh (Priantara, 2011). Aplikasi yang dikembangkan bertujuan mendeteksi kesamaan suatu pola kalimat yang diindikasikan sebagai plagiarisme dalam karya ilmiah. Suatu teks dikatakan sama apabila nilai kemiripan pada suatu dokumen sangat tinggi. Pada dasarnya algoritma *winnowing* memproses masukan berupa teks yang diproses sehingga menghasilkan *output* berupa kumpulan nilai-nilai *hash*. Nilai yang dihasilkan akan dimasukkan dalam *window-window* untuk mendapatkan nilai fingerprint yang

akan digunakan dalam perhitungan *similarity* menggunakan *jaccard coefisien*. Metode *winnowing* dipilih karena nilai yang dihasilkan tidak berpengaruh pada bentuk aktif dan pasif suatu kalimat.

Algoritma Rabin-Karp kurang tepat digunakan dalam pencocokan kata kunci topik mahasiswa dengan kata kunci penelitian dosen yang memiliki variasi kalimat beragam. Oleh karena itu dibutuhkan algoritma yang dapat menangani masalah yang terjadi pada algoritma Rabin-Karp. Perbandingan algoritma *winnowing* dan Rabin-Karp dapat dilihat pada Tabel 2.1.

Tabel 2.1 Perbandingan Algoritma Rabin-Karp dengan Algoritma Winnowing

No	Algoritma Winnowing	Algoritma Rabin-Karp
1.	Dapat menangani variasi string yang beragam	Dapat menangani variasi string yang homogen
2.	Hasil keakuratan yang tinggi karena tidak berpengaruh pada variasi kalimat	Hasil keakuratan rendah apabila kalimat bervariasi
3.	Waktu komputasi tidak tergantung pada variasi string	Waktu komputasi tergantung pada variasi string
4.	Nilai hash dipetakan dalam window-window untuk mempermudah pencarian letak fingerprint	Hasil hash tidak dipetakan beraturan sehingga letak fingerprint tidak dapat dicari

Tabel perbandingan diatas menyimpulkan bahwa algoritma *winnowing* memiliki banyak keunggulan dibanding algoritma Rabin-Karp. Oleh karena itu algoritma *winnowing* dipilih dalam penerapan sistem rekomendasi penentuan dosen pembimbing.

2.2 Metode String Matching

String matching adalah algoritma yang digunakan untuk memecahkan masalah pencocokan suatu teks terhadap teks lain. Pencocokan string fokus pada pencarian satu atau lebih kesamaan kata dalam sebuah teks (Pertiwi, 2015). Tahap awal yang dilakukan dalam implementasi *string matching* yaitu ekstraksi informasi menjadi teks terstruktur menggunakan metode *text mining*.

Metode pendeteksi kesamaan *string* dibagi menjadi tiga bagian yaitu metode perbandingan teks lengkap, metode dokumen *fingerprinting*, dan metode kesamaan kata kunci (Kurniawati, 2008). Berikut ini penjelasan dari masing-masing metode dari algoritma pendeteksi kesamaan :

- a. Metode perbandingan teks lengkap diterapkan dengan membandingkan semua isi dokumen. Perbandingan teks lengkap membutuhkan waktu yang lama tetapi cukup efektif, karena kumpulan dokumen yang diperbandingkan adalah dokumen yang disimpan pada penyimpanan lokal. Metode perbandingan teks lengkap tidak dapat diterapkan untuk kumpulan dokumen yang tidak terdapat pada dokumen lokal. Algoritma yang digunakan pada metode ini adalah algoritma *Brute-Force*, algoritma *edit distance*, algoritma *Boyer Moore* dan algoritma *lavenshtein distance*.
- b. Dokumen *fingerprinting* merupakan metode yang digunakan untuk mendeteksi keakuratan salinan antar dokumen, baik semua teks yang terdapat di dalam dokumen atau hanya sebagian teks saja. Prinsip kerja dari metode dokumen *fingerprinting* adalah dengan menggunakan teknik *hashing*. Teknik *hashing* adalah sebuah fungsi yang mengkonversi setiap string menjadi nilai *hash*. Algoritma yang digunakan pada metode ini adalah Rabin-Karp, *Winnowing* dan Manber.
- c. Prinsip metode kesamaan kata kunci adalah mengekstrak kata kunci dari dokumen dan kemudian dibandingkan dengan kata kunci pada dokumen yang lain.

2.3 Metode Dokumen *Fingerprinting*

Prinsip kerja dari metode dokumen *fingerprinting* adalah dengan menggunakan teknik *hashing*. Teknik *hashing* adalah sebuah fungsi yang mengkonversi setiap string menjadi bilangan kemudian menyimpannya dalam sebuah skema atau bagan. Skema digital dokumen *fingerprinting* terdiri dari sejumlah posisi yang diberi tanda di dalam dokumen, algoritma *fingerprinting* yang akan memilih tanda yang akan di tambahkan untuk setiap posisi tergantung pada jumlah salinan.

Secara umum prinsip kerja dari metode dokumen *fingerprinting* adalah dengan tahapan sebagai berikut :

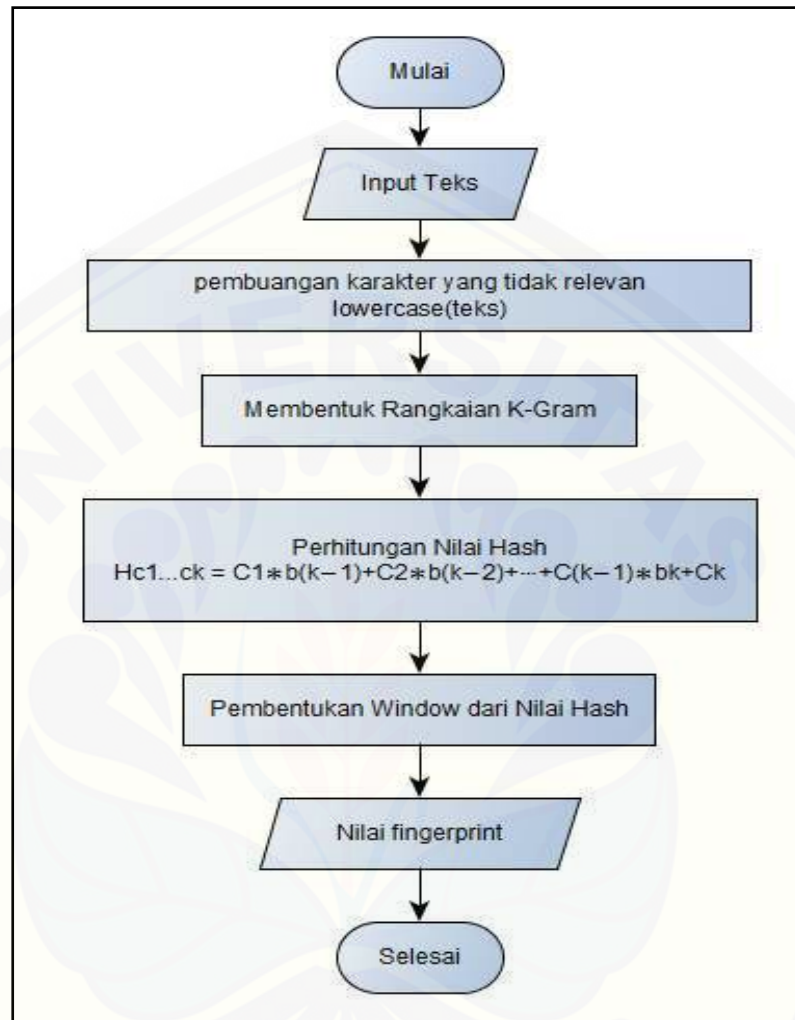
1. Asumsikan teks adalah string s yang panjangnya t .
2. Hilangkan tanda baca dan spasi.
3. Membagi dokumen menjadi k -gram. K -gram digunakan dalam pengambilan potongan karakter huruf sejumlah k dari sebuah kata secara kontinuitas dibaca dari teks sumber hingga akhir dari dokumen.
4. Menghitung nilai *hash* dari setiap gram yang dihasilkan.
5. Memilih beberapa hasil *hash* menjadi dokumen *fingerprinting*.

2.4 Algoritma *Winnowing*

2.4.1 Definisi Algoritma *Winnowing*

Algoritma *winnowing* merupakan algoritma yang digunakan dalam deteksi kesamaan menggunakan fungsi *hashing*. *Winnowing* adalah algoritma yang digunakan untuk melakukan proses *document fingerprinting*. Algoritma *winnowing* melakukan penghitungan nilai-nilai *hash* dari setiap k -gram, untuk mencari nilai *hash* selanjutnya digunakan fungsi *rolling hash*. Kemudian dibentuk *window* dari nilai-nilai *hash* tersebut. Dalam setiap *window* dipilih nilai *hash* minimum. Jika ada lebih dari satu *hash* dengan nilai minimum, dipilih nilai *hash* yang paling kanan. Kemudian semua nilai *hash* terpilih disimpan untuk dijadikan *fingerprint* dari suatu dokumen. *Fingerprint* ini yang akan dijadikan dasar

pembandingan kesamaan antara teks yang telah dimasukkan. Langkah perhitungan algoritma *winning* digambarkan pada Gambar 2.1



Gambar 2.1 Flowchart Algoritma *winning*

Gambar 2.1 merupakan langkah yang harus dilakukan pada algoritma *winning* diantaranya:

a. *Text Preprocessing*

Text Preprocessing merupakan tahap awal yang dilakukan pada algoritma *winning*. Tahap *preprocessing* dimulai dari mengubah text menjadi huruf kecil, menghilangkan tanda baca, menghilangkan angka yang ada pada text yang dibandingkan. Berikut contoh proses *text preprocessing* :

Teks : Sistem Informasi 99

Preprocessing : sistem informasi

b. K-Gram

K-gram adalah rangkaian gram dengan panjang k yang diambil secara kontinu. Metode k-gram digunakan dalam pengambilan potongan-potongan karakter huruf sejumlah k dari sebuah kata secara kontinuitas dibaca dari teks sumber hingga akhir dari dokumen. Makin besar k-gram yang digunakan makin kecil persentase kecocokan, sebaliknya apabila makin kecil k-gram yang digunakan maka makin besar juga persentase kecocokan teks. Berikut contoh gram dengan k=5:

Teks : A do run run run, a do run run

Preprocessing : Adorunrunrunadorunrun

Gram K=5 : adoru dorun orunr runru unrun nrunr runru unrun nruna runad unado nador adoru dorun orunr runru unrun (Kurniawati, 2008).

c. Hashing

Hashing adalah suatu cara untuk mentransformasi sebuah string menjadi suatu nilai yang unik dengan panjang tertentu (*fixed-length*) yang berfungsi sebagai penanda pada string tersebut. Fungsi untuk menghasilkan nilai disebut fungsi hash, sedangkan nilai yang dihasilkan adalah nilai hash. Pada perhitungan nilai hash dibutuhkan nilai ASCII karakter. ASCII (*American Standard Code for Information Interchange*) merupakan suatu standart internasional dalam kode huruf dan simbol yang bersifat universal. Algoritma *winnowing* didasarkan pada fakta jika dua buah string sama maka dapat dipastikan memiliki hash value yang sama. Cara untuk menghitung nilai hash pada kalimat disebut *rolling hash*.

Hashing mempermudah pencarian pada database. Apabila string tidak di-hash, maka pencarian akan dilakukan karakter per karakter pada kalimat kalimat yang panjangnya bervariasi dan ada 26 kemungkinan pada setiap karakter. Namun pencarian akan lebih efisien jika di-hash karena hanya akan membandingkan empat digit angka dengan 10 kemungkinan pada setiap angka. Fungsi hash $H(c_1...c_k)$ didefinisikan sebagaimana pada rumus :

$$H(c_1...c_k) = c_1 * b_{(k-1)} + c_2 * b_{(k-2)} + \dots + c_{(k-1)} * b_k + c_k \quad (2.1)$$

Keterangan :

c : nilai ASCII karakter

b : basis (bilangan prima)

k : banyak karakter

Berikut ini merupakan contoh penghitungan *rolling hash* dengan k-gram = 5 dan basis = 3:

Diketahui teks : theclassicproblem

ASCII : t = 116, h = 104, e = 101, c = 99, l = 108, a = 97,

s = 115, i = 105, p = 112, r = 114, o = 111, b = 98, m = 109

Perhitungan nilai hash dapat dilihat pada Gambar 2.2 yaitu :

	t	h	e	c	l	a	s	s
h[0]	$t \cdot 3^3 + h \cdot 3^2 + e \cdot 3^1 + c \cdot 3^0$							
h[1]		$h \cdot 3^3 + e \cdot 3^2 + c \cdot 3^1 + l \cdot 3^0$						
h[2]			$e \cdot 3^3 + c \cdot 3^2 + l \cdot 3^1 + a \cdot 3^0$					
h[3]				$c \cdot 3^3 + l \cdot 3^2 + a \cdot 3^1 + s \cdot 3^0$				
h[4]					$l \cdot 3^3 + a \cdot 3^2 + s \cdot 3^1 + s \cdot 3^0$			

Gambar 2.2 Perhitungan *hash* menggunakan *rolling hash*

Hasil perhitungan nilai *hash* dari Gambar 2.2 dengan memasukkan nilai ASCII dan basis hash yaitu :

1850198	1670359	1624760	1606324
1725620	1588411	1850686	1836795
1683991	1613644	1806143	1829962
1769877			

d. Membentuk *Window* dari Hash yang Didapat

Besarnya nilai *window* berpengaruh pada fingerprint yang dihasilkan. *Window* yang besar akan menghasilkan jumlah fingerprint yang kecil dan sebaliknya. Langkah selanjutnya dari algoritma *winnowing* yaitu pembentukan *window-window* dari nilai *hash* yang telah terbentuk. Pada tiap *window* yang

terbentuk dilakukan pemilihan nilai *hash* terkecil untuk dijadikan *fingerprint*. Apabila terdapat dua nilai *hash* terkecil yang sama pada dua atau lebih *window*, maka akan dipilih *window* yang paling kanan untuk dijadikan *fingerprint*. Contoh dari pembentukan *window-window* dengan nilai *window* = 4 yaitu :

1850198 1670359 1624760 1606324
 1725620 1588411 1850686 1836795
 1683991 1613644 1806143 1829962
 1769877

Pembentukan window-window

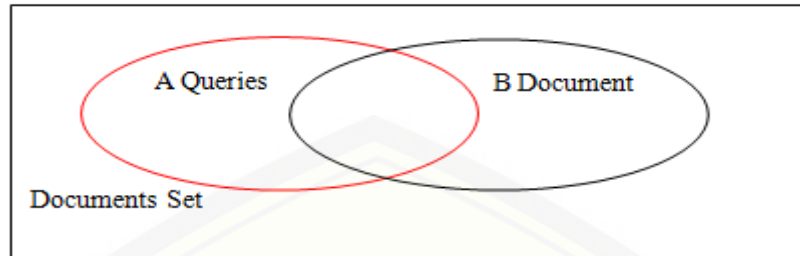
{ 1850198 1670359 1624760 **1606324** }
 { 1670359 1624760 **1606324** 1725620 }
 { 1624760 1606324 1725620 **1588411** }
 { 1606324 1725620 **1588411** 1850686 }
 { 1725620 **1588411** 1850686 1836795 }
 { **1588411** 1850686 1836795 1683991 }
 { 1850686 1836795 1683991 **1613644** }
 { 1836795 1683991 **1613644** 1806143 }
 { 1683991 **1613644** 1806143 1829962 }
 { **1613644** 1806143 1829962 1769877 }

Nilai *fingerprint* yang didapat pada pembentukan setiap *window* diatas yaitu [1606324], [1588411] dan [1613644].

e. Pengukuran Nilai Kemiripan (*Similarity Value*)

Mengukur *similarity* dan jarak antara dua entitas informasi merupakan syarat utama pada semua kasus penemuan informasi, seperti pada Information retrieval dan data mining yang kemudian dikembangkan dalam bentuk perangkat lunak salah satunya sistem deteksi kesamaan (Salmuasih, 2013). Perhitungan *similarity* dalam pemecahan masalah penemuan informasi tidak hanya meningkatkan kualitas pilihan informasi tetapi juga membantu mengurangi biaya dan waktu proses. Untuk mendapatkan tingkat presentase *similarity* sebuah dokumen dengan

dokumen lain dapat menggunakan Persamaan *Jaccard Coefficient* yang ditunjukkan pada gambar 2.3 dan persamaan sebagai berikut :



Gambar 2.3 Penggunaan Irisan Untuk Menghitung Kesamaan Dokumen

$$Sims(q, dj) = J(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{2.2}$$

Dimana Sims adalah nilai *Similarity* , sedangkan A dan B adalah jumlah *fingerprint* yang dihasilkan teks A dan teks B. Berikut contoh perhitungan *similarity* dengan membandingkan teks A dan teks B :

1. Diketahui teks A : theclassicproblem, teks B = theclassicstyle
2. Dengan K = 4 maka didapat nilai kesamaan pada Tabel 2.2 berikut :

Tabel 2.2 Perhitungan Algoritma *winnowing* dan *jaccard Coeficient*

Proses	Teks A				Teks B			
Rangkaian Gram	thecl lassi icpro oblem	hecla assic cprob	eclas ssicp probl	class sicpr roble	thecl lassi icsty	hecla assic cstyl	eclas ssics style	class sicst
Nilai Hash	1850198 1606324 1850686 1613644 1769877	1670359 1725620 1836795 1806143	1624760 1588411 1683991 1829962	1850198 1606324 1850689 1617999	1670359 1725620 1836830 1854041	1624760 1588411 1684386		

Dilanjutkan..

Lanjutan..

Keterangan	Teks A	Teks B
Window	{ 1850198 1670359 1624760 1606324 } { 1670359 1624760 1606324 1725620 } { 1624760 1606324 1725620 1588411 } { 1606324 1725620 1588411 1850686 } { 1725620 1588411 1850686 1836795 } { 1588411 1850686 1836795 1683991 } { 1850686 1836795 1683991 1613644 } { 1836795 1683991 1613644 1806143 } { 1683991 1613644 1806143 1829962 } { 1613644 1806143 1829962 1769877 }	{ 1850198 1670359 1624760 1606324 } { 1670359 1624760 1606324 1725620 } { 1624760 1606324 1725620 1588411 } { 1606324 1725620 1588411 1850689 } { 1725620 1588411 1850689 1836830 } { 1588411 1850689 1836830 1684386 } { 1850689 1836830 1684386 1617999 } { 1836830 1684386 1617999 1854041 }
Fingerprint	[1606324] [1588411] [1613644]	[1606324] [1588411] [1617999]
Kesamaan	$(2/4) * 100 = 50\%$	

2.4.2 Keuntungan Menggunakan Algoritma *Winnowing*

Ada beberapa alasan mengapa orang menggunakan algoritma *winnowing* untuk pencocokan *string* antara lain :

- Metode *winnowing* memiliki tingkat keakuratan yang cukup tinggi
- Kecepatan komputasi metode *winnowing* tidak tergantung pada kompleksitas kalimat yang di kerjakan.
- Metode *winnowing* cocok digunakan pada pencocokan *string multiple pattern*
- Memiliki kemampuan untuk menentukan letak *fingerprint* yang dihasilkan
- Dapat dimodifikasi untuk menambah nilai keakuratan
- Menghemat waktu dalam pengambilan keputusan

2.4.3 Kelemahan Algoritma *Winnowing*

Disamping memiliki beberapa keuntungan, algoritma *winnowing* juga memiliki beberapa kelemahan diantaranya:

- a. Tidak dapat menangani kata yang jumlah huruf kurang dari tiga dengan nilai gram yang besar.
- b. Persentase yang dihasilkan tidak 100% benar karena persentase tergantung pada nilai gram dan nilai *window*.

2.5 Persyaratan Administrasi Seorang Dosen Pembimbing

Persyaratan administrasi juga menjadi bahan pertimbangan pada sistem rekomendasi yang akan dibangun. Hal ini dimaksudkan supaya tidak hanya hasil pencocokan yang menjadi pertimbangan, tetapi harus sesuai dengan syarat akademik. Penentuan dosen pembimbing mempunyai aturan atau *rules* yang tidak boleh dilanggar untuk memperoleh hasil tugas akhir yang maksimal.

Pembimbing skripsi adalah dosen yang diberi tugas oleh Dekan atau Pembantu Dekan I untuk memberikan bimbingan skripsi berdasarkan usulan Komisi Bimbingan (Kombi) dengan memperhatikan jabatan fungsional dosen. Pergantian pembimbing skripsi dapat dilakukan dengan pertimbangan tertentu berdasarkan rapat Kombi dan disetujui oleh Dekan atau Pembantu Dekan I. Pembimbing Skripsi terdiri dari Pembimbing I (utama) dan Pembimbing II (Anggota) dimana Pembimbing I (Utama) memiliki jabatan fungsional lebih tinggi atau minimal sama dengan Pembimbing II (Anggota). Pembimbing I (Utama) adalah pembimbing yang bertanggung jawab atas substansi dan konsistensi penulisan skripsi serta pengambilan keputusan terhadap segala hal yang terkait dengan penulisan skripsi mahasiswa bimbingan (Khoiri, 2015). Pedoman yang diterapkan di FKM Universitas Jember untuk menentukan dosen pembimbing juga dapat diterapkan pada prodi Sistem Informasi Universitas Jember. Adapun syarat sebagai pembimbing utama adalah sebagai berikut:

1. Kualifikasi pendidikan minimal S-2
2. Jabatan fungsional minimal asisten ahli (bagi dosen tetap)
3. Mempunyai kompetensi yang sesuai dengan judul/tema skripsi yang diajukan oleh mahasiswa

4. Berstatus dosen tetap di lingkungan prodi Sistem Informasi Universitas Jember atau dosen luar biasa yang mengajar di prodi Sistem Informasi Universitas Jember.
5. Bagi dosen luar biasa, selain memenuhi syarat (1, 3, dan 4), juga harus memiliki golongan ruang minimal III-c atau menjabat pada posisi *middle management* yang setara dengan jabatan kepala seksi.

Pembimbing II (Anggota) adalah pembimbing yang bertanggung jawab atas sistematika penulisan skripsi yang sesuai pedoman penulisan skripsi serta membantu pembimbing utama dalam pengambilan keputusan terhadap segala hal yang terkait dengan penulisan skripsi mahasiswa (Khoiri, 2015). Adapun syarat sebagai pembimbing anggota adalah sebagai berikut :

1. Kualifikasi pendidikan minimal S2
2. Jabatan fungsional minimal asisten ahli (bagi dosen tetap)
3. Mempunyai kompetensi yang sesuai dengan judul/tema skripsi yang diajukan oleh mahasiswa
4. Pembimbing II tidak harus berstatus dosen tetap atau dosen luar biasa di Prodi Sistem Informasi tetapi bisa didasarkan pada pertimbangan keilmuan atau pengalamannya yang akan dipertimbangkan lebih lanjut oleh Kombi (Komisi Bimbingan) Prodi Sistem Informasi Universitas Jember.

BAB 3. METODOLOGI PENELITIAN

Bab ini menjelaskan tentang metode yang digunakan dalam merancang dan membangun sistem rekomendasi penentuan dosen pembimbing menggunakan algoritma *winnowing*.

3.1 Jenis Penelitian

Jenis penelitian yang digunakan pada penelitian ini merupakan jenis penelitian pengembangan, karena tujuan penelitian adalah untuk membangun sebuah sistem rekomendasi. Penelitian pengembangan bertujuan untuk membuat dan mengembangkan suatu produk yang efektif untuk digunakan. Penelitian ini bukan jenis penelitian yang ditunjukkan untuk menemukan teori atau menguji kebenaran dari suatu teori dalam bentuk eksperimentasi.

3.2 Tempat dan Waktu Penelitian

Penelitian dilakukan di Program Studi Sistem Informasi Universitas Jember. Waktu penelitian dilakukan selama 4 (empat) bulan, dimulai pada bulan Maret 2016 sampai dengan bulan Juni 2016.

3.3 Metode Pengumpulan Data

Pengumpulan data dilakukan untuk memperoleh informasi yang dibutuhkan dalam mencapai tujuan penelitian. Pengumpulan data pada penelitian ini dilakukan melalui beberapa teknik pengumpulan data yaitu:

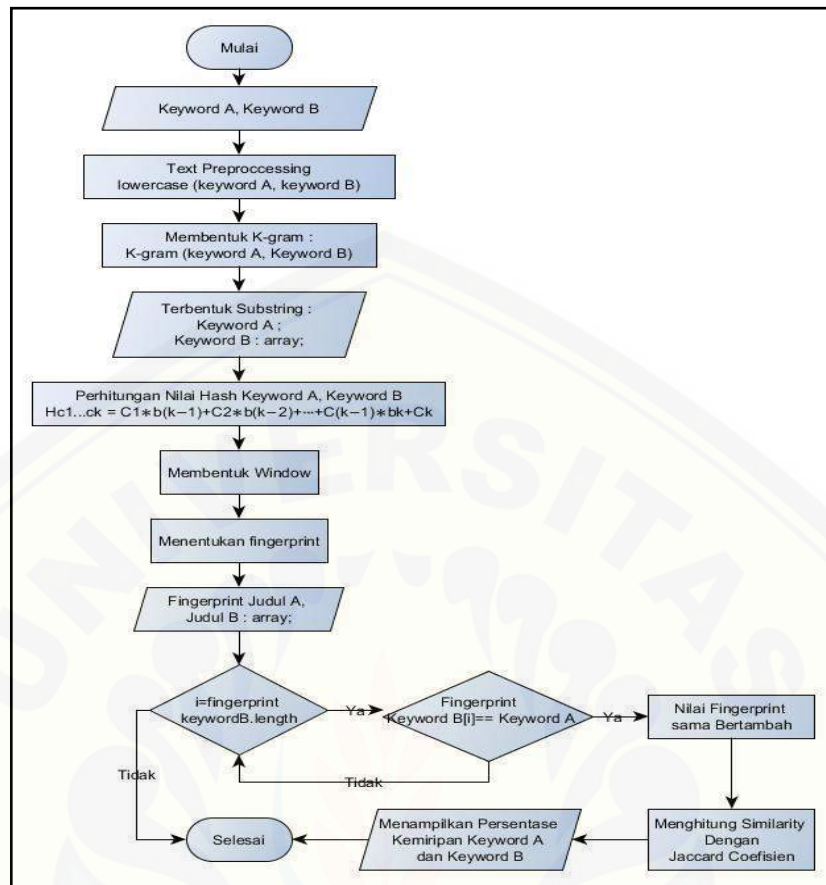
- a. Melakukan wawancara kepada pihak akademik dan komisi bimbingan untuk mendapatkan data mengenai penelitian dosen dan pengajuan proposal.

- b. Melakukan studi literature, jurnal, media, maupun internet mengenai algoritma *winnowing*.

Jenis data yang digunakan dalam penelitian adalah data primer dan data sekunder. Data primer merupakan data yang diperoleh melalui metode wawancara. Data sekunder merupakan data yang diperoleh dari studi literatur. Pengumpulan data dan informasi yang dilakukan antara lain data dosen, data penelitian dosen, data aturan penetapan dosen pembimbing, dan data pengajuan tugas akhir.

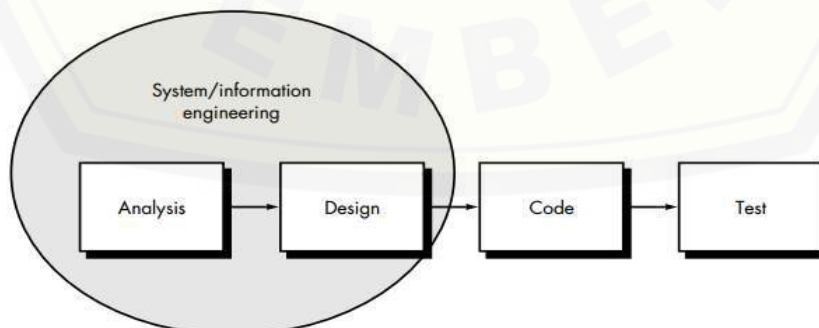
3.4 Metode Analisis Data

Tahap analisis dimulai dengan menelaah data secara keseluruhan yang telah diperoleh dari tahap pengumpulan data. Langkah selanjutnya adalah menganalisa data dengan menggunakan algoritma *winnowing*. Penerapan algoritma *winnowing* pada sistem rekomendasi penentuan dosen pembimbing dapat dilihat pada Gambar 3.1

Gambar 3.1 Diagram Alir Penerapan Algoritma *Wining*

3.5 Teknik Pengembangan Sistem

Pengembangan sistem rekomendasi dosen pembimbing menggunakan algoritma *wining* mengadopsi dari model *waterfall* seperti pada Gambar 3.2



Gambar 3.2 Model Waterfall (Sumber: Pressman, 2001)

3.5.1 Analisis Kebutuhan

Analisis kebutuhan merupakan tahap untuk mengumpulkan data, informasi, serta mencari kebutuhan fungsional dan non fungsional sistem. Pada tahap ini, peneliti mencari permasalahan yang ada untuk dapat dianalisis kebutuhan yang diperlukan, sebagai solusi dari permasalahan yang muncul. Data-data yang telah didapat kemudian dikelompokkan menjadi kebutuhan fungsional dan non-fungsional sistem.

3.5.2 Desain

Tahap ini meliputi tahap desain sistem dengan pembuatan diagram menggunakan tools *UML Visual Paradigm*. Diagram-diagram tersebut akan digunakan sebagai acuan pembuatan sistem pada tahap implementasi. Diagram-diagram yang digunakan yang digunakan sebagai berikut :

a. *Business Process*

Business process merupakan diagram yang menggambarkan proses dari sebuah sistem yang meliputi *input*, *output*, dan *goal* yang merupakan tujuan dari sebuah sistem yang dibangun.

b. *Use Case Diagram*

Use Case Diagram menggambarkan fungsionalitas dari sebuah sistem dan interaksi antara pengguna dengan sistem untuk melakukan suatu pekerjaan tertentu. *Use Case Diagram* juga menggambarkan hak akses dari pengguna.

c. *Scenario*

Scenario digunakan untuk menjelaskan fitur sistem yang terdapat pada *Use Case Diagram*.

d. *Sequence Diagram*

Sequence Diagram digunakan untuk menunjukkan rangkaian pesan yang dikirim antar object juga interaksi antar object.

e. *Class Diagram*

Class Diagram digunakan untuk menggambarkan struktur statis class dalam sistem.

f. *Entity Realtionship Diagram*

Entity Realtionship Diagram digunakan untuk menunjukkan relasi antar object.

3.5.3 Penulisan Kode Program

Setelah desain sistem telah selesai dilakukan maka selanjutnya pada tahap ini akan dilakukan pembuatan sistem dan implementasiannya. Pembuatan sistem meliputi penulisan kode program, dan pembuatan basis data. Penulisan kode program dilakukan menggunakan tools *Sublime Text 3* sebagai editor dengan bahasa pemrograman *PHP* sedangkan untuk manajemen basis data menggunakan tools *Xampp for windows*.

3.5.4 Pengujian

Pengujian digunakan untuk mengetahui sejauh mana sistem ini dapat berjalan. Testing berfungsi untuk mengetahui apakah sistem ini dapat berfungsi dengan baik sesuai dengan yang diharapkan. Serta untuk mengetahui letak kekurangan yang ada pada sistem ini. Terdapat dua metode yang digunakan untuk pengujian ini yakni :

a. *White box testing*

White box testing merupakan pengujian perangkat lunak yang dilakukan dengan cara melihat modul yang telah dibuat dengan program yang ada. Pengujian *white box* ini menggunakan *Cyclomatic Complexity (CC)*. *CC* merupakan alat pengukuran untuk mengindikasikan kekompleksitasan program dengan menelusuri jalur-jalur yang ada melalui *source code* nya. Pengujian *white box* dilakukan oleh penulis dengan menggunakan rumus *CC* yang dapat dilihat pada persamaan 3.1.

$$CC = (Edge - Node) + 2 \quad (3.1)$$

b. *Black box testing*

Black box testing merupakan pengujian perangkat lunak dengan cara memeriksa fungsionalitas dari aplikasi yang berkaitan dengan struktur internal atau kerja saat program dijalankan. Pengujian dilakukan dengan menjalankan program secara langsung dan menganalisis *input* serta *output* yang dihasilkan aplikasi. *Black box testing* dilakukan oleh *user* yang menjalankan aplikasi. Yang terakhir dilakukan dalam tahapan model *waterfall* ini adalah tahapan pemeliharaan. Sistem yang sudah selesai akan mengalami perubahan. Perubahan biasanya berupa *error* sehingga diperlukan perbaikan dan pemeliharaan pada sistem.

3.5.5 Pemeliharaan

Pemeliharaan merupakan proses perawatan sistem setelah sistem digunakan oleh pengguna. Pemeliharaan dilakukan dengan mengecek kinerja sistem secara berkala. Pengecekan dilakukan apakah kinerja sistem masih berjalan dengan baik dan memperbaiki apakah terdapat kerusakan atau tidak.

BAB 4. PERANCANGAN SISTEM

Bab ini menguraikan mengenai analisis kebutuhan, desain, implementasi, dan pengujian sistem yang digunakan dalam proses pengembangan sistem rekomendasi dosen pembimbing menggunakan algoritma *winning*. Dimana tahapan analisis hingga pengujian yang dilakukan sesuai dengan metode pengembangan *waterfall*.

4.1 Deskripsi Umum Sistem

Deskripsi umum dari sistem rekomendasi penentuan dosen pembimbing skripsi yang dibangun dalam penelitian ini akan dijelaskan lebih detail pada SOP (statement of purpose) sistem dan fungsi sistem.

4.1.1 SOP (*Statement Of Purpose*)

Sistem Rekomendasi penentuan dosen pembimbing menggunakan algoritma *winning* ini berisikan kata kunci penelitian dosen serta kata kunci topik mahasiswa yang akan di dihitung kesamaan untuk menghasilkan rekomendasi dosen pembimbing yang sesuai. Hasil perhitungan kesamaan akan ditampilkan berupa saran dosen dengan rata-rata persentase kemiripan kata kunci penelitian yang dimiliki. Sistem rekomendasi ini dibangun berbasis website agar dapat digunakan dengan mudah oleh akademik, dosen, dan komisi bimbingan.

4.1.2 Fungsi Sistem

Fungsi utama dari sistem yang dibangun dalam penelitian ini terletak pada fitur login yang dapat menentukan hak akses dari setiap pengguna sistem. Ketika

pengguna melakukan login, maka sistem akan melakukan autentifikasi *username* dan *password* dari pengguna. Selanjutnya sistem akan menyajikan tampilan sistem yang sesuai dengan hak akses dari pengguna yang meliputi:

a. Akademik

Akademik merupakan pengguna sistem yang telah terdaftar, dan ketika login berhasil maka sistem akan menampilkan halaman dashboard dan halaman-halaman yang sesuai dengan level akademik. Akademik dapat mengakses halaman data pegajuan, data persetujuan, data bimbingan, data dosen, data keyword, dan data bidang keahlian dosen. Akademik bertanggung jawab atas data pengajuan, data keyword dosen dan data surat tugas yang digunakan dalam sistem rekomendasi penentuan dosen pembimbing.

b. Komisi Bimbingan

Komisi Bimbingan (Kombi) merupakan pengguna sistem yang telah terdaftar, dan ketika login berhasil maka sistem akan menampilkan halaman dashboard dan halaman-halaman yang sesuai dengan level kombi. Kombi dapat melakukan rekomendasi dosen pembimbing dan memilih dosen pembimbing yang tepat untuk setiap mahasiswa yang mengajukan skripsi. Kombi juga dapat melihat data kelulusan mahasiswa, data mahasiswa aktif semester akhir, dan data pengajuan proposal skripsi.

c. Dosen

Dosen merupakan pengguna yang telah terdaftar apabila biodata ada pada daftar dosen Program Studi Sistem Informasi, dan ketika login berhasil maka sistem akan menampilkan halaman dashboard dosen yang sesuai dengan level dosen. Dosen dapat melihat jumlah bimbingan yang sedang diampu dan melihat rekap mahasiswa yang dibimbing.

4.2 Analisis Kebutuhan

Berdasarkan metode pengembangan sistem model *waterfall*, tahapan awal yang dilakukan adalah tahapan analisis. Tahapan analisis ini dilakukan terhadap objek penelitian untuk memperoleh kebutuhan-kebutuhan dari sistem yang

dibangun, baik berupa kebutuhan fungsional maupun kebutuhan nonfungsional. Hasil analisa tersebut sangat mempengaruhi fungsionalitas sistem yang dibangun untuk dapat digunakan sesuai dengan fungsi dan kebutuhan pengguna.

4.2.1 Kebutuhan Fungsional

Kebutuhan fungsional sistem berisi fitur-fitur inti yang harus dipenuhi dalam sistem agar sistem mampu difungsikan sesuai dengan tujuan dan kebutuhan pengguna terhadap sistem itu sendiri. Kebutuhan fungsional dari sistem rekomendasi penentuan dosen menggunakan algoritma *winnowing* yaitu:

- a. Sistem mampu memberikan rekomendasi dosen pembimbing kepada kombi sesuai dengan topik dan kompetensi.
- b. Sistem mampu mengelola data pengajuan (input, view)
- c. Sistem mampu mengelola data persetujuan (edit, view)
- d. Sistem mampu mengelola cetak surat tugas (edit, print)
- e. Sistem mampu menampilkan jumlah bimbingan setiap dosen berikut dengan detail mahasiswa yang dibimbing
- f. Sistem mampu memberi rekomendasi dosen pembimbing sesuai dengan bidang yang dipilih
- g. Sistem mampu mengelola data keyword penelitian dosen (input, view)
- h. Sistem mampu mengelola data mahasiswa yang telah selesai studi
- i. Sistem mampu menampilkan total pengajuan, mahasiswa aktif, jumlah dosen, dan jumlah mahasiswa lulus studi
- j. Sistem mampu memberikan informasi kepada dosen pembimbing mengenai jumlah mahasiswa aktif bimbingan tugas akhir.

4.2.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional merupakan fitur-fitur yang dimiliki untuk mendukung sistem dalam memenuhi fungsionalitasnya untuk dapat memenuhi

kebutuhan dari pengguna. Kebutuhan non-fungsional sistem rekomendasi penentuan dosen menggunakan algoritma *winnowing* yaitu:

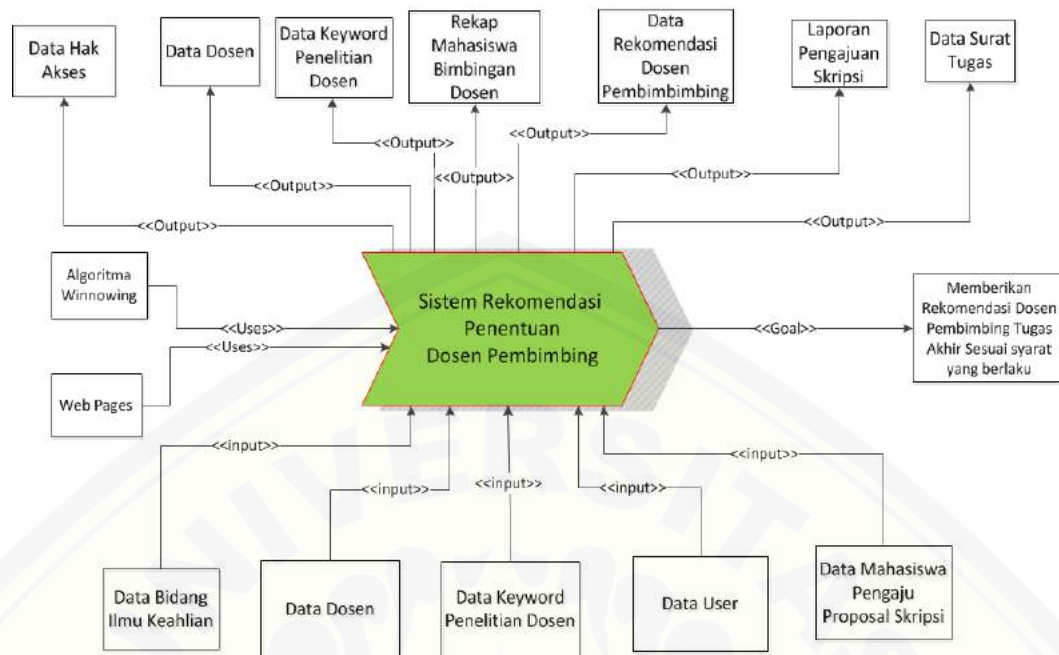
- a. Sistem dapat digunakan oleh banyak pengguna pada saat yang bersamaan dan pada media yang berbeda.
- b. Tampilan dan bahasa komunikasi sistem mudah dimengerti oleh pengguna sehingga memberikan kenyamanan pemakaian dan memudahkan pengoperasian.
- c. Sistem menggunakan *username* dan *password* untuk autentifikasi akses pengguna terhadap sistem.

4.3 Desain Sistem

Tahapan yang dilakukan setelah melakukan analisis kebutuhan sistem yaitu tahap perencanaan pembangunan sistem yang dapat digambarkan dengan desain sistem. Desain sistem sistem rekomendasi penentuan dosen menggunakan algoritma *winnowing* ini meliputi *bussiness process*, *use case diagram*, *scenario*, *activity diagram*, *sequence diagram*, *class diagram*, dan *entity relationship diagram*.

4.3.1 Bussiness Process

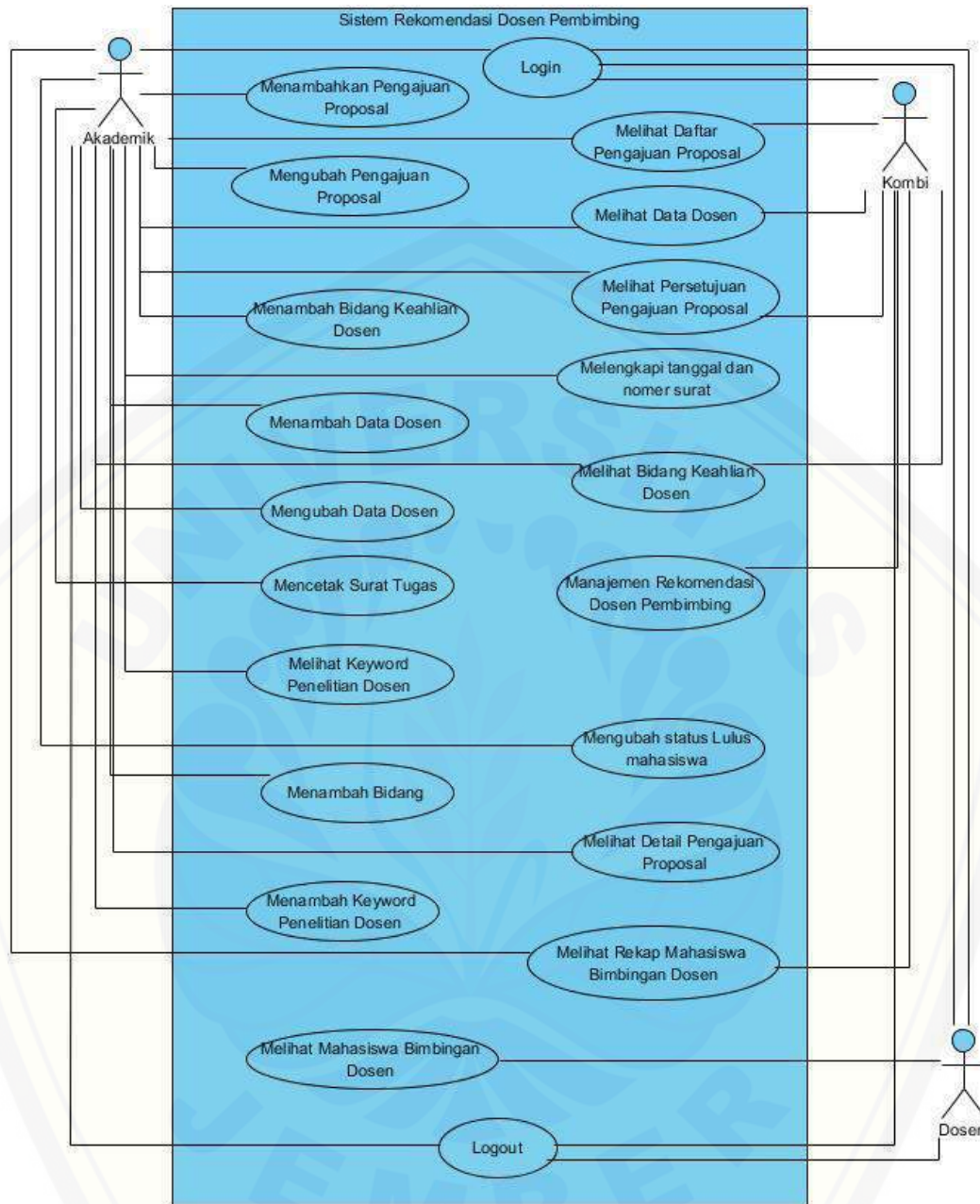
Selain dapat dideskripsikan dalam sebuah SOP (Statement of Purpouse), gambaran umum sistem rekomendasi penentuan dosen menggunakan algoritma *winnowing* dapat digambarkan melalui sebuah *business process*. Seperti yang dapat kita lihat pada Gambar 4.1 yang menggambarkan data-data yang digunakan sebagai masukan, data keluaran, uses sistem yang dibangun, hingga goal dari sistem itu sendiri.



Gambar 4.1 Business Process Sistem

4.3.2 Use Case Diagram

Use case diagram merupakan pemodelan yang dibuat untuk dapat menggambarkan interaksi antara aktor dengan sistem rekomendasi penentuan dosen menggunakan algoritma *winnowing* yang akan dibangun. Melalui use case diagram dapat diketahui interaksi yang dapat dilakukan setiap aktor terhadap sistem sesuai dengan hak akses yang dimiliki oleh masing-masing aktor atau pengguna. Pada Gambar 4.2 digambarkan use case diagram yang terdiri atas tiga aktor dengan dua puluh satu *use case*.



Gambar 4.2 Use Case Diagram

Berdasarkan *use case* diagram pada Gambar 4.2 terdapat tiga pengguna, yaitu Akademik, komisi bimbingan, dan dosen. Adapun deskripsi dari masing-masing aktor dapat dilihat pada Tabel 4.1.

Tabel 4.1 Deskripsi pembagian aktor sistem

No	Aktor	Deskripsi
1.	Akademik	Aktor yang memiliki hak penuh terhadap data pengajuan, data persetujuan, data bidang, data dosen, data surat tugas dosen, dan data keyword penelitian dosen.
2.	Komisi Bimbingan	Komisi bimbingan yang memiliki hak memilih hasil rekomendasi yang disajikan sistem, melihat rekap data mahasiswa semester akhir, data dosen, data pengajuan, data jumlah bimbingan dosen dan data persetujuan.
3.	Dosen	Dosen yang memiliki hak melihat biodata dan jumlah mahasiswa yang dibimbing.

Selain memiliki tiga pengguna, dalam *use case* diagram juga terdapat dua puluh dua satu *use case*. Deskripsi umum dari *use case* tersebut dapat dilihat pada Tabel 4.2.

Tabel 4.2 Deskripsi *use case* sistem

No	Use Case	Deskripsi
1.	Login	Fitur yang berfungsi melakukan autentifikasi hak akses semua aktor dalam menggunakan sistem.
2.	Menambah pengajuan proposal	Fitur yang dapat diakses oleh akademik untuk menambahkan data pengajuan dosen pembimbing oleh mahasiswa.
3.	Mengubah pengajuan proposal	Fitur yang dapat diakses oleh akademik untuk mengubah data pengajuan dosen pembimbing oleh mahasiswa.

Dilanjutkan..

Lanjutan..

No	Use Case	Deskripsi
4.	Melihat daftar pengajuan proposal	Fitur yang dapat diakses oleh akademik dan komisi bimbingan untuk melihat daftar pengajuan dosen pembimbing oleh mahasiswa.
5.	Melihat data dosen	Fitur yang dapat diakses oleh akademik dan komisi bimbingan untuk melihat daftar dosen program studi sistem informasi.
6.	Melihat persetujuan pengajuan proposal	Fitur yang dapat diakses oleh akademik dan komisi bimbingan untuk melihat daftar persetujuan pengajuan proposal tugas akhir.
7.	Menambah bidang keahlian dosen	Fitur yang dapat diakses oleh akademik untuk menambahkan bidang keahlian setiap dosen.
8.	Melengkapi cetak surat tugas	Fitur yang dapat diakses oleh akademik untuk mencetak surat tugas dari pengajuan proposal tugas akhir yang telah disetujui.
9.	Menambah data dosen	Fitur yang dapat diakses oleh akademik untuk menambahkan data dosen baru.
10.	Melihat bidang keahlian dosen	Fitur yang dapat diakses oleh akademik dan komisi bimbingan untuk melihat data bidang keahlian setiap dosen.
11.	Mengubah data dosen	Fitur yang dapat diakses oleh akademik untuk mengubah data dosen dalam sistem.
12.	Mencetak surat tugas	Fitur yang dapat diakses oleh akademik untuk mencetak surat tugas, hasil dari persetujuan pengajuan yang dilakukan oleh komisi bimbingan.
13.	Manajemen rekomendasi dosen pembimbing	Fitur yang dapat diakses oleh komisi bimbingan untuk memilih dosen dari hasil rekomendasi sistem.

Dilanjutkan...

Lanjutan..

No	Use Case	Deskripsi
14.	Melihat keyword penelitian dosen	Fitur yang dapat diakses oleh akademik untuk melihat keyword arsip penelitian yang pernah dilakukan dosen.
15.	Mengubah status lulus mahasiswa	Fitur yang dapat diakses oleh akademik untuk mengupdate status aktif mahasiswa menjadi lulus yang ada pada daftar persetujuan.
16.	Menambah bidang	Fitur yang dapat diakses oleh akademik untuk menambahkan bidang ilmu yang dikuasai dosen.
17.	Menambah keyword penelitian dosen	Fitur yang dapat diakses oleh akademik untuk menambahkan kata kunci penelitian sebelumnya yang dilakukan dosen.
18.	Melihat rekap mahasiswa bimbingan dosen	Fitur yang dapat diakses oleh akademik dan komisi bimbingan untuk melihat rekap jumlah bimbingan setiap dosen dan daftar mahasiswa yang dibimbing oleh tiap dosen.
19.	Melihat mahasiswa bimbingan dosen	Fitur yang dapat diakses oleh dosen untuk melihat jumlah bimbingan dosen dan daftar mahasiswa yang dibimbing oleh dosen.
20.	Melihat detail pengajuan	Fitur yang dapat diakses oleh akademik untuk melihat detail pengajuan proposal tugas akhir.
21.	Logout	Fitur yang berfungsi melakukan proses keluar dari sistem bagi semua actor

4.3.3 Use Case Skenario

Usecase skenario adalah dokumentasi terhadap kebutuhan fungsional sistem. *Usecase* skenario sistem rekomendasi dosen pembimbing adalah sebagai berikut:

A. Skenario *Login*

Skenario *login* digunakan untuk menggambarkan proses *login* mulai dari memasukkan username dan password hingga menampilkan halaman *home*. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *use case* skenario login dapat dilihat Lampiran A (*Usecase Skenario*).

B. Skenario Menambah Pengajuan Proposal

Skenario menambah pengajuan proposal digunakan untuk menggambarkan proses tambah pengajuan proposal mulai dari memasukkan biodata pengajuan sampai sistem berhasil menyimpan data pengajuan. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario menambah pengajuan proposal dapat dilihat pada tabel 4.4

Tabel 4.3 Skenario menambah pengajuan proposal

ID	USC-2
Name	Menambah Pengajuan Proposal
Participating Actor	Akademik
Entry Condition	Akademik akan menambah pengajuan proposal tugas akhir
Exit Condition	Akademik akan menambah pengajuan proposal tugas akhir

SKENARIO UTAMA

“Menambah Pengajuan Proposal”

Aksi Aktor	Reaksi Sistem
1. Klik menu pengajuan	2. Menampilkan data pengajuan proposal

Dilanjutkan..

Lanjutan..

Aksi Aktor	Reaksi Sistem
3. Klik tombol tambah	
	4. Menampilkan halaman form pengajuan baru
5. Mengisi form pengajuan meliputi : a. Nim b. Nama c. Judul Proposal d. Keyword 1 e. Keyword 2 f. Keyword 3 g. Dosen Utama h. Dosen Pendamping	
6. Klik tombol save	
	7. Menampilkan halaman data pengajuan proposal.

SKENARIO ALTERNATIF

“kolom belum diisi ”

Aksi Aktor	Reaksi Sistem
5. Mengisi form pengajuan meliputi : a. Nim b. Nama c. Judul Proposal d. Keyword 1 e. Keyword 2 f. Keyword 3 g. Dosen Utama h. Dosen Pendamping	
6. Klik tombol save	
	7. Menampilkan pesan dengan menunjuk pada kolom yang belum diisi.

Dilanjutkan..

Lanjutan..

SKENARIO ALTERNATIF	
“Terdapat pilihan pembimbing utama dan pendamping sama”	
Aksi Aktor	Reaksi Sistem
5. Mengisi form pengajuan meliputi : <ol style="list-style-type: none"> a. Nim b. Nama c. Judul Proposal d. <i>Keyword 1</i> e. <i>Keyword 2</i> f. <i>Keyword 3</i> g. Dosen Utama h. Dosen Pendamping 6. Klik tombol save	7. Menampilkan pesan untuk memilih dosen pembimbing yang berbeda

C. Skenario Mengubah Pengajuan Proposal

Skenario mengubah pengajuan proposal menjelaskan mengenai aktifitas sistem dan aktor dalam mengubah pengajuan proposal mulai dari nama, judul, *keyword* penelitian. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mengubah pengajuan proposal dapat dilihat pada Lampiran A (*Usecase* Skenario).

D. Skenario Melihat Daftar Pengajuan Proposal

Skenario mengubah pengajuan proposal menjelaskan mengenai aktifitas aktor dan sistem dalam melihat data pengajuan proposal dengan memilih submenu pengajuan. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario melihat daftar pengajuan proposal dapat dilihat pada Lampiran A (*Usecase* Skenario).

E. Skenario Melihat Data Dosen

Skenario mengubah pengajuan proposal menjelaskan mengenai aktifitas aktor dan sistem dalam melihat biodata dosen dengan memilih menu dosen. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario melihat data dosen dapat dilihat pada Lampiran A (*Usecase* Skenario)

F. Skenario Melihat Persetujuan Pengajuan Proposal

Skenario melihat pengajuan proposal menjelaskan mengenai aktifitas aktor dan sistem dalam melihat persetujuan pengajuan proposal dengan memilih menu persetujuan. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario melihat persetujuan pengajuan proposal dapat dilihat pada Lampiran A (*Usecase* Skenario).

G. Skenario Menambah Bidang Keahlian Dosen

Skenario menambah bidang keahlian dosen menjelaskan mengenai aktifitas aktor dan sistem dalam menambah bidang keahlian dosen ke dalam sistem. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario menambah bidang keahlian dosen dapat dilihat pada Lampiran A (*Usecase* Skenario).

H. Skenario Melengkapi Cetak Surat Tugas

Skenario melengkapi form cetak surat tugas menjelaskan mengenai aktifitas aktor dan sistem dalam melengkapi form tanggal dan nomer surat sebelum surat tugas dicetak. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario melengkapi cetak surat tugas dapat dilihat pada Lampiran A (*Usecase* Skenario).

I. Skenario Menambah Data Dosen

Skenario menambah data dosen menjelaskan mengenai aktifitas aktor dan sistem dalam menambah data dosen baru dengan memilih menu tambah yang

akan menampilkan form isian seperti nama dosen, nip, tanggal lahir, pangkat dan jabatan fungsional. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario menambah data dosen dapat dilihat pada Lampiran A (*Usecase* Skenario).

J. Skenario Melihat Bidang Keahlian Dosen

Skenario melihat bidang keahlian dosen menjelaskan mengenai aktifitas aktor dan sistem dalam melihat bidang keahlian atau kompetensi dosen. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario melihat bidang keahlian dosen dapat dilihat pada Lampiran A (*Usecase* Skenario).

K. Skenario Mengubah Data Dosen

Skenario mengubah data dosen menjelaskan mengenai aktifitas aktor dan sistem dalam mengubah biodata dosen apabila terjadi kesalahan dalam inputan. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mengubah data dosen dapat dilihat pada Lampiran A (*Usecase* Skenario).

L. Skenario Mencetak Surat Tugas

Skenario mencetak surat tugas menjelaskan mengenai aktifitas aktor dan sistem dalam mencetak surat tugas dengan memilih menu cetak. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mencetak surat tugas dapat dilihat pada Lampiran A (*Usecase* Skenario).

M. Skenario Manajemen Rekomendasi Dosen Pembimbing

Skenario manajemen rekomendasi dosen pembimbing menjelaskan mengenai aktifitas aktor dan sistem dalam melakukan proses persetujuan pengajuan yaitu mencari rekomendasi dosen pembimbing utama dan pembimbing pendamping.

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario manajemen rekomendasi dosen pembimbing dapat dilihat pada Lampiran A (*Usecase* Skenario).

N. Skenario Melihat *Keyword* Penelitian Dosen

Skenario manajemen rekomendasi dosen pembimbing menjelaskan mengenai aktifitas aktor dan sistem dalam melihat *keyword* penelitian dosen dengan memilih sub menu *keyword*. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario melihat *keyword* penelitian dosen dapat dilihat pada Lampiran A (*Usecase* Skenario).

O. Skenario Mengubah Status Lulus Mahasiswa

Skenario mengubah status lulus mahasiswa menjelaskan mengenai aktifitas aktor dan sistem dalam proses mengubah status lulus mahasiswa dengan memilih tombol ubah status mahasiswa. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mengubah status lulus mahasiswa dapat dilihat pada Lampiran A (*Usecase* Skenario).

P. Skenario Menambah Bidang

Skenario menambah bidang menjelaskan mengenai aktifitas aktor dan sistem dalam proses menambah bidang ilmu dengan memilih sub menu bidang dan memilih menu tambah. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario menambah bidang dapat dilihat pada Lampiran A (*Usecase* Skenario).

Q. Skenario Menambah *Keyword* Penelitian Dosen

Skenario menambah *keyword* penelitian dosen menjelaskan mengenai aktifitas aktor dan sistem dalam proses menambah *keyword* penelitian dosen dengan menampilkan formulir tambah *keyword*. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario

menambah keyword penelitian dosen dapat dilihat pada Lampiran A (*Usecase Skenario*).

R. Skenario Melihat Rekap Mahasiswa Bimbingan Dosen

Skenario melihat rekap mahasiswa bimbingan dosen menjelaskan mengenai aktifitas aktor dan sistem dalam proses melihat rekap mahasiswa bimbingan masing-masing dosen. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario rekap mahasiswa bimbingan dosen dapat dilihat pada Lampiran A (*Usecase Skenario*).

S. Skenario Melihat Detail Pengajuan Proposal

Skenario melihat detail pengajuan proposal menjelaskan mengenai aktifitas aktor dan sistem dalam proses melihat detail pengajuan proposal dengan memilih menu detail pada kolom *action*. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario melihat detail pengajuan proposal dapat dilihat pada Lampiran A (*Usecase Skenario*).

T. Skenario Logout

Skenario logout menjelaskan mengenai aktifitas aktor dan sistem dalam proses keluar dari sistem. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario *logout* dapat dilihat pada Lampiran A (*Usecase Skenario*).

4.3.4 Activity Diagram

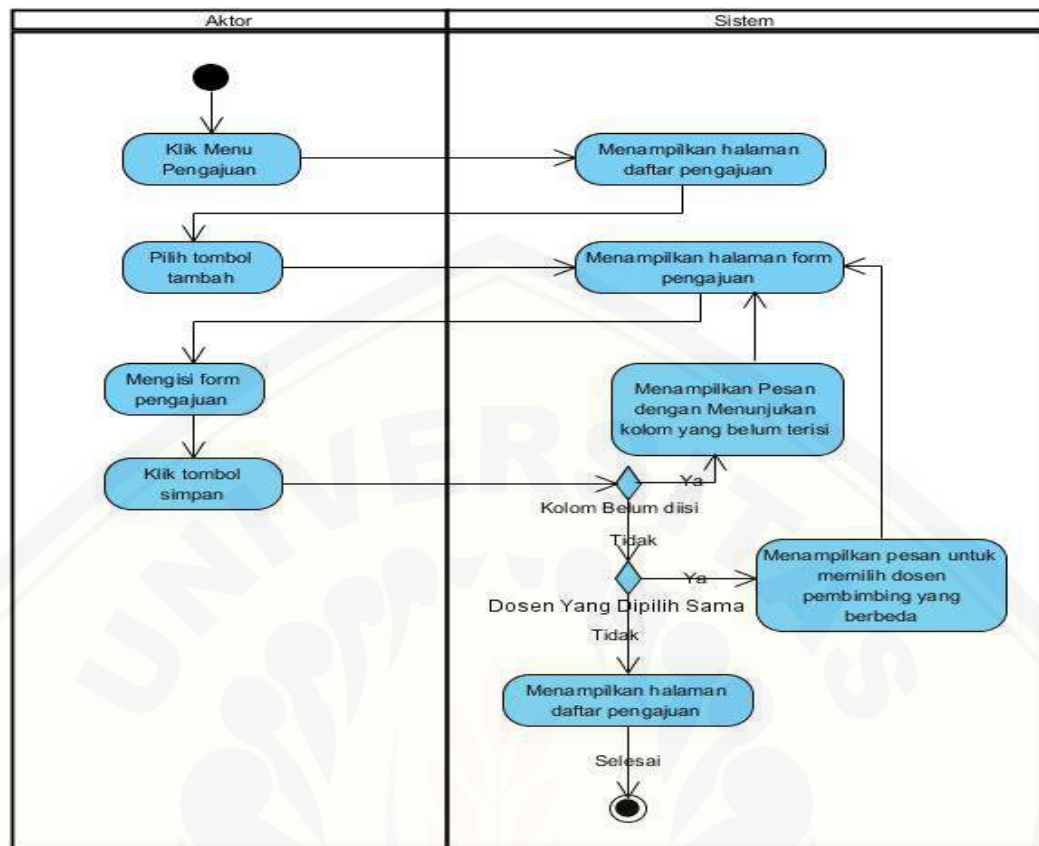
Activity Diagram menggambarkan aliran aktivitas dalam sistem rekomendasi penentuan dosen pembimbing menggunakan algoritma *winnowing* yang akan dibangun. Sistem rekomendasi penentuan dosen pembimbing menggunakan algoritma *winnowing* ini memiliki dua puluh satu *activity diagram* yaitu sebagai berikut:

A. *Activity Diagram Login*

Activity diagram login menjelaskan tentang bagaimana sistem dapat menjalankan fungsi melakukan autentifikasi hak akses semua aktor dalam menggunakan sistem. *Activity diagram login* dapat dilihat pada Lampiran B (*Activity Diagram*).

B. *Activity Diagram Menambah pengajuan proposal*

Activity diagram menambah pengajuan proposal dilakukan oleh akademik. *Activity diagram menambah pengajuan proposal* menjelaskan bagaimana akademik dapat menambah pengajuan proposal yang belum terdaftar dalam sistem. *Activity diagram menambah pengajuan proposal* dapat dilihat pada Gambar 4.3



Gambar 4.3 Activity Diagram Menambah Pengajuan Proposal

C. Activity Diagram Mengubah Pengajuan Proposal

Activity diagram mengubah pengajuan proposal dapat dilakukan oleh akademik. Activity diagram mengubah pengajuan proposal menjelaskan tentang bagaimana akademik dapat mengubah pengajuan proposal pada sistem. Activity diagram Mengubah pengajuan proposal dapat dilihat pada Lampiran B (Activity Diagram).

D. Activity Diagram Melihat Daftar Pengajuan Proposal

Activity diagram melihat daftar pengajuan proposal dapat dilakukan oleh akademik dan kombi. Activity diagram Melihat daftar pengajuan proposal menjelaskan tentang bagaimana akademik dan kombi dapat menampilkan daftar pengajuan proposal tugas akhir pada sistem. Activity diagram melihat daftar pengajuan proposal dapat dilihat pada Lampiran B (Activity Diagram).

E. *Activity Diagram* Melihat Data Dosen

Activity diagram melihat data dosen dapat dilakukan oleh komisi bimbingan dan akademik. *Activity diagram* melihat data dosen menjelaskan tentang bagaimana pengguna dapat menampilkan data dosen program studi dan banyaknya bimbingan dosen pada sistem. *Activity diagram* melihat data dosen dapat dilihat pada Lampiran B (*Activity Diagram*).

F. *Activity Diagram* Melihat Persetujuan Pengajuan Proposal

Activity diagram melihat persetujuan pengajuan proposal dapat dilakukan oleh pengguna yaitu akademik dan komisi bimbingan. *Activity diagram* melihat persetujuan pengajuan proposal menjelaskan tentang bagaimana pengguna dapat menampilkan daftar persetujuan pengajuan proposal. *Activity diagram* melihat persetujuan pengajuan proposal dapat dilihat pada Lampiran B (*Activity Diagram*).

G. *Activity Diagram* Menambah Bidang Keahlian Dosen

Activity diagram menambah bidang keahlian dosen dapat dilakukan oleh akademik. *Activity diagram* menambah bidang keahlian dosen menjelaskan tentang bagaimana akademik dapat menambahkan bidang keahlian dosen pada sistem. *Activity diagram* menambah bidang keahlian dosen dapat dilihat pada Lampiran B (*Activity Diagram*).

H. *Activity Diagram* Melengkapi Cetak Surat Tugas

Activity diagram melengkapi cetak surat tugas dapat dilakukan akademik. *Activity diagram* melengkapi cetak surat tugas menjelaskan tentang bagaimana akademik dapat melengkapi form surat tugas sebelum dicetak pada sistem. *Activity diagram* melengkapi cetak surat tugas dapat dilihat pada Lampiran B (*Activity Diagram*).

I. *Activity Diagram* Menambah Data Dosen

Activity diagram menambah data dosen dapat dilakukan oleh akademik. *Activity diagram* menambah data dosen menjelaskan tentang bagaimana akademik

dapat menambahkan data dosen baru pada sistem. *Activity* diagram menambah data dosen dapat dilihat pada Lampiran B (*Activity* Diagram).

J. *Activity* Diagram Melihat Bidang Keahlian Dosen

Activity diagram Melihat bidang keahlian dosen dapat dilakukan oleh dua pengguna pengguna yaitu akademik dan kombi. *Activity* diagram melihat bidang keahlian dosen menjelaskan tentang bagaimana pengguna dapat menampilkan semua data bidang keahlian setiap dosen. *Activity* diagram melihat bidang keahlian dosen dapat dilihat pada Lampiran B (*Activity* Diagram).

K. *Activity* Diagram Mengubah Data Dosen

Activity diagram mengubah data dosen dapat dilakukan oleh akademik. *Activity* diagram Mengubah data dosen menjelaskan tentang bagaimana akademik dapat mengubah semua data dosen yang dipilih. *Activity* diagram mengubah data dosen dapat dilihat pada Lampiran B (*Activity* Diagram).

L. *Activity* Diagram Mencetak Surat Tugas

Activity diagram mencetak surat tugas dapat dilakukan oleh akademik. *Activity* diagram Mencetak surat tugas menjelaskan tentang bagaimana akademik dapat mencetak surat tugas sesuai dengan nama mahasiswa yang dipilih dalam sistem. *Activity* diagram mencetak surat tugas dapat dilihat pada Lampiran B (*Activity* Diagram).

M. *Activity* Diagram Manajemen Rekomendasi Dosen Pembimbing

Activity diagram manajemen rekomendasi dosen pembimbing dapat dilakukan oleh kombi. *Activity* diagram manajemen rekomendasi dosen pembimbing menjelaskan tentang bagaimana kombi dapat menampilkan hasil rekomendasi dan memilih rekomendasi berdasarkan pada besarnya persentase dan beban bimbingan

dosen pada sistem. *Activity* diagram manajemen rekomendasi dosen pembimbing dapat dilihat pada Lampiran B (*Activity* Diagram).

N. *Activity* Diagram Melihat *Keyword* Penelitian Dosen

Activity diagram melihat *keyword* penelitian dosen dapat dilakukan oleh akademik. *Activity* diagram melihat *keyword* penelitian dosen menjelaskan tentang bagaimana akademik dapat menampilkan data *keyword* penelitian dosen terdahulu. *Activity* diagram melihat *keyword* penelitian dosen dapat dilihat pada Lampiran B (*Activity* Diagram).

O. *Activity* Diagram Mengubah Status Lulus Mahasiswa

Activity diagram mengubah status lulus mahasiswa dapat dilakukan oleh akademik. *Activity* diagram mengubah status lulus mahasiswa menjelaskan tentang bagaimana akademik dapat mengubah status aktif mahasiswa menjadi lulus. *Activity* diagram mengubah status lulus mahasiswa dapat dilihat pada Lampiran B (*Activity* Diagram).

P. *Activity* Diagram Menambah Bidang

Activity diagram menambah bidang dapat dilakukan oleh akademik. *Activity* diagram menambah bidang menjelaskan tentang bagaimana akademik dapat menambahkan jenis bidang ilmu ke dalam sistem. *Activity* diagram menambah bidang dapat dilihat pada Lampiran B (*Activity* Diagram).

Q. *Activity* Diagram Menambah *Keyword* Penelitian Dosen

Activity diagram menambah *keyword* penelitian dosen dapat dilakukan oleh akademik. *Activity* diagram menambah *keyword* penelitian menjelaskan tentang bagaimana akademik dapat menambahkan *keyword* penelitian setiap dosen ke dalam sistem. *Activity* diagram menambah *keyword* penelitian dapat dilihat pada Lampiran B (*Activity* Diagram).

R. *Activity Diagram* Melihat Rekap Mahasiswa Bimbingan Dosen

Activity diagram melihat rekap mahasiswa bimbingan dosen dapat dilakukan oleh dua pengguna yaitu kombi dan akademik. *Activity diagram* melihat rekap mahasiswa bimbingan dosen menjelaskan tentang bagaimana pengguna dapat melihat rekap bimbingan dosen yang berisi jumlah mahasiswa bimbingan setiap dosen dalam sistem. *Activity diagram* melihat rekap mahasiswa bimbingan dosen dapat dilihat pada Lampiran B (*Activity Diagram*).

S. *Activity Diagram* Melihat Detail Pengajuan Proposal

Activity diagram melihat detail pengajuan proposal dapat dilakukan oleh akademik. *Activity diagram* melihat detail pengajuan proposal menjelaskan tentang bagaimana pengguna dapat melihat setiap pengajuan lengkap yang ada dalam sistem. *Activity diagram* melihat detail pengajuan proposal dapat dilihat pada Lampiran B (*Activity Diagram*).

T. *Activity Diagram Logout*

Activity diagram logout menjelaskan tentang bagaimana sistem dapat menjalankan fungsi melakukan proses keluar dari sistem. *Activity diagram logout* dapat dilihat pada Lampiran B (*Activity Diagram*).

4.3.5 *Sequence Diagram*

Sequence diagram adalah dokumentasi suatu diagram terurut yang menampilkan interaksi - interaksi antar objek di dalam sistem. *Sequence diagram* digunakan untuk menggambarkan skenario dan memodelkan aliran logika dalam sistem dengan cara *visual*. *Sequence diagram* dari sistem rekomendasi dosen

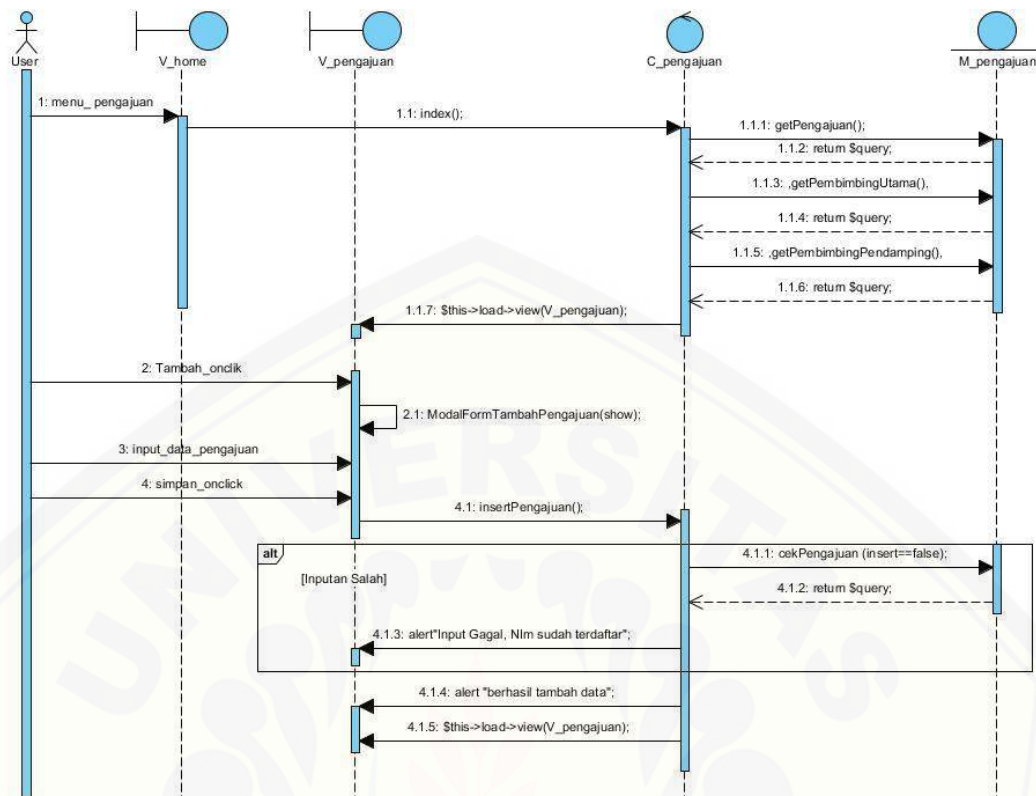
pembimbing skripsi yang menggunakan algoritma *winnowing* adalah sebagai berikut:

A. *Sequence Diagram Login*

Penggambaran *sequence diagram login* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case login*. *Class* yang berperan dalam *sequence diagram login* adalah *view V_dashboard*, *view V_login*, *controller c_login* dan *model M_username*, *M_session*. *Sequence diagram login* lebih lengkap dijelaskan pada Lampiran C (*Sequence Diagram*).

B. *Sequence Diagram Menambah Pengajuan Proposal*

Penggambaran *sequence diagram menambah pengajuan proposal* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Class* yang berperan dalam *sequence diagram menambah pengajuan proposal* adalah *view V_dashboard*, *view V_pengajuan*, *view hasil*, *controller c_pengajuan* dan *model M_pengajuan*. Proses dari aliran *sequence diagram* diawali ketika actor melakukan perintah pada suatu *view*, selanjutnya *view* akan mengirim perintah dengan memanggil *method* yang terdapat pada *controller* dimana *controller* akan melanjutkan memanggil *method* pada *class* yang berhubungan baik dengan *class model* jika terdapat *query* yang diambil maupun *class view* jika akan menampilkan *view* selanjutnya. *Sequence diagram menambah pengajuan proposal* lebih lengkap dijelaskan pada Gambar 4.4



Gambar 4.4 *Sequence* diagram menambah pengajuan proposal

C. *Sequence* Diagram Mengubah Pengajuan Proposal

Penggambaran *sequence* diagram mengubah pengajuan proposal digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* mengubah pengajuan proposal. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence* diagram mengubah pengajuan proposal lebih lengkap dijelaskan pada Lampiran C (*Sequence* Diagram).

D. *Sequence* Diagram Melihat Daftar Pengajuan Proposal

Penggambaran *sequence* diagram melihat daftar pengajuan proposal digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* melihat daftar pengajuan proposal. Masing - masing *class* akan ditampilkan secara

visual dengan gambar. *Sequence* diagram melihat daftar pengajuan proposal lebih lengkap dijelaskan pada Lampiran C (*Sequence* Diagram).

E. *Sequence* Diagram Melihat Data Dosen

Penggambaran *sequence* diagram melihat data dosen digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* melihat data dosen. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence* diagram melihat data dosen lebih lengkap dijelaskan pada Lampiran C (*Sequence* Diagram).

F. *Sequence* Diagram Melihat Persetujuan Pengajuan Proposal

Penggambaran *sequence* diagram melihat persetujuan pengajuan proposal digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* melihat persetujuan pengajuan proposal. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence* diagram melihat persetujuan pengajuan proposal lebih lengkap dijelaskan pada Lampiran C (*Sequence* Diagram).

G. *Sequence* Diagram Menambah Bidang Keahlian Dosen

Penggambaran *sequence* diagram Menambah bidang keahlian dosen digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* menambah bidang keahlian dosen. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence* diagram menambah bidang keahlian dosen lebih lengkap dijelaskan pada Lampiran C (*Sequence* Diagram).

H. *Sequence* Diagram Melengkapi Cetak Surat Tugas

Penggambaran *sequence* diagram melengkapi cetak surat tugas digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* melengkapi cetak surat tugas. Masing - masing *class* akan ditampilkan secara

visual dengan gambar. *Sequence* diagram melengkapi cetak surat tugas lebih lengkap dijelaskan pada Lampiran C (*Sequence* Diagram).

I. *Sequence* Diagram Menambah Data Dosen

Penggambaran *sequence* diagram menambah data dosen digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* menambah data dosen. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence* diagram menambah data dosen lebih lengkap dijelaskan pada Lampiran C (*Sequence* Diagram).

J. *Sequence* Diagram Melihat Bidang Keahlian Dosen

Penggambaran *sequence* diagram melihat bidang keahlian dosen digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* melihat bidang keahlian dosen. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence* diagram melihat bidang keahlian dosen lebih lengkap dijelaskan pada Lampiran C (*Sequence* Diagram).

K. *Sequence* Diagram Mengubah Data Dosen

Penggambaran *sequence* diagram mengubah data dosen digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* Mengubah data dosen. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence* diagram mengubah data dosen lebih lengkap dijelaskan pada Lampiran C (*Sequence* Diagram).

L. *Sequence* Diagram Mencetak Surat Tugas

Penggambaran *sequence* diagram Mencetak surat tugas digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* Mencetak surat tugas. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar.

Sequence diagram mencetak surat tugas lebih lengkap dijelaskan pada Lampiran C (*Sequence Diagram*).

M. *Sequence Diagram* Manajemen Rekomendasi Dosen Pembimbing

Penggambaran *sequence* diagram manajemen rekomendasi dosen pembimbing digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* manajemen rekomendasi dosen pembimbing. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence* diagram memasukkan manajemen rekomendasi dosen pembimbing lebih lengkap dijelaskan pada Lampiran C (*Sequence Diagram*).

N. *Sequence Diagram* Melihat *Keyword* Penelitian Dosen

Penggambaran *sequence* diagram melihat *keyword* penelitian dosen digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* melihat *keyword* penelitian dosen. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence* diagram melihat *keyword* penelitian dosen lebih lengkap dijelaskan pada Lampiran C (*Sequence Diagram*).

O. *Sequence Diagram* Mengubah Status Lulus Mahasiswa

Penggambaran *sequence* diagram mengubah status lulus mahasiswa digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* mengubah status lulus mahasiswa. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence* diagram mengubah status lulus mahasiswa lebih lengkap dijelaskan pada Lampiran C (*Sequence Diagram*).

P. *Sequence Diagram* Menambah Bidang

Penggambaran *sequence* diagram menambah bidang digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* menambah bidang. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar.

Sequence diagram menambah bidang lebih lengkap dijelaskan pada Lampiran C (*Sequence Diagram*).

Q. *Sequence Diagram Menambah Keyword Penelitian Dosen*

Penggambaran *sequence* diagram menambah *keyword* penelitian dosen digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* Menambah *keyword* penelitian dosen. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence* diagram menambah *keyword* penelitian dosen lebih lengkap dijelaskan pada Lampiran C (*Sequence Diagram*).

R. *Sequence Diagram Melihat Rekap Mahasiswa Bimbingan Dosen*

Penggambaran *sequence* diagram melihat rekap mahasiswa bimbingan dosen digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* melihat rekap mahasiswa bimbingan dosen. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence* diagram melihat rekap mahasiswa bimbingan dosen dosen lebih lengkap dijelaskan pada Lampiran C (*Sequence Diagram*).

S. *Sequence Diagram Melihat Detail Pengajuan*

Penggambaran *sequence* diagram melihat detail pengajuan bimbingan dosen digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* melihat detail pengajuan. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence* diagram melihat detail pengajuan lebih lengkap dijelaskan pada Lampiran C (*Sequence Diagram*).

T. *Sequence Diagram Logout*

Penggambaran *sequence* diagram *logout* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* *logout*. Masing - masing *class* akan

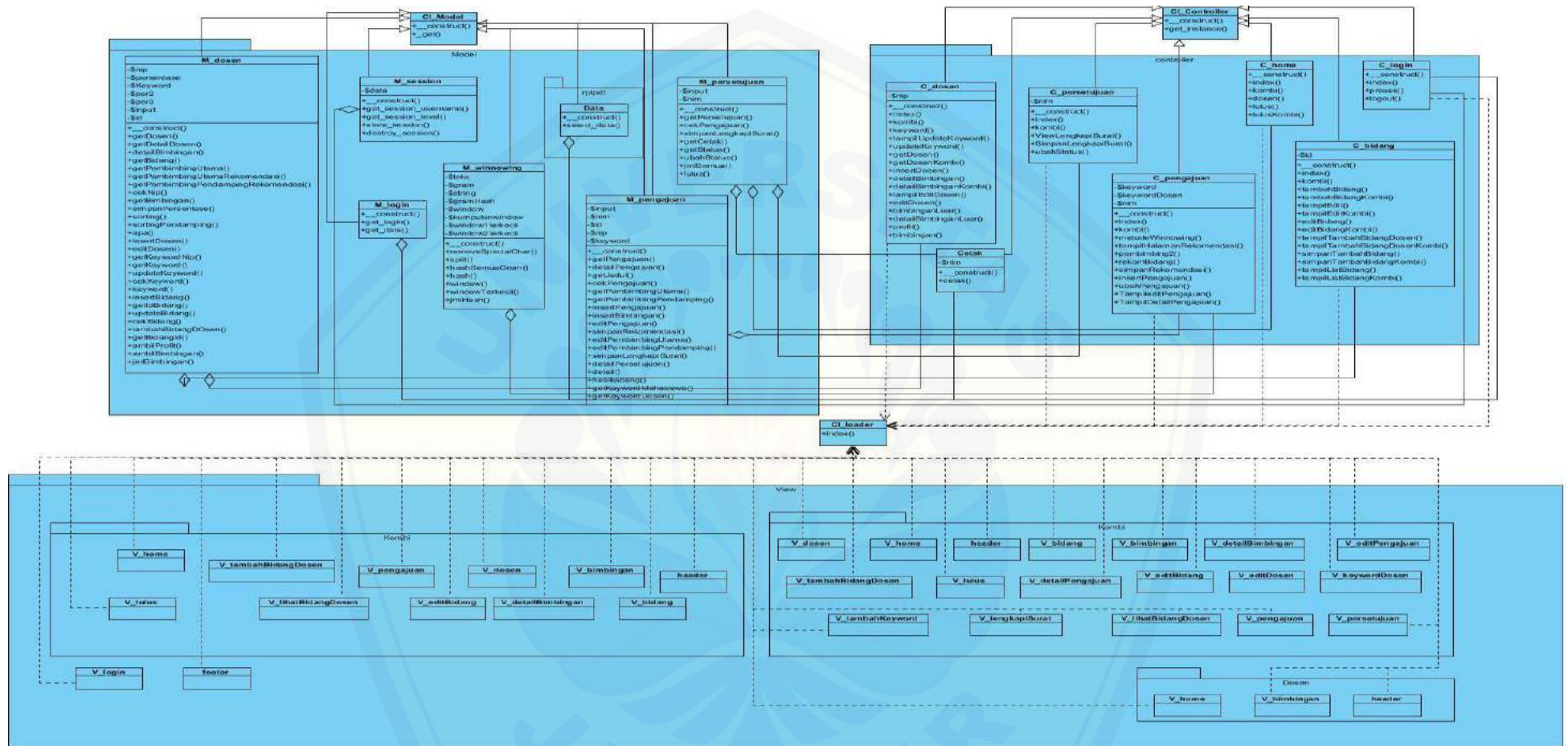
ditampilkan secara *visual* dengan gambar. *Sequence* diagram *logout* lebih lengkap dijelaskan pada Lampiran C (*Sequence* Diagram).

4.3.6 Class Diagram

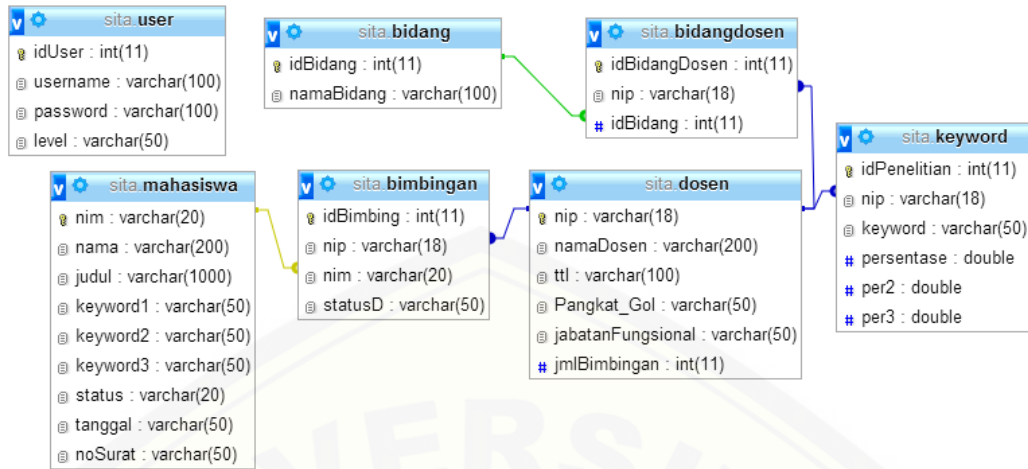
Class diagram menggambarkan hubungan antara kelas yang digunakan untuk membangun suatu sistem. Dalam paradigma OOP (*Object Oriented Programming*) terdapat tiga jenis kelas yaitu *model*, *view* dan *controller*. Berdasarkan *sequence* diagram yang telah dibangun, *class* diagram sistem rekomendasi penentuan dosen pembimbing menggunakan algoritma *winning* terdiri dari 6 kelas *model*, 35 kelas *view* dan 7 kelas *controller*. *Class* diagram sistem dapat dilihat pada Gambar 4.5.

4.3.7 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) sistem rekomendasi penentuan dosen pembimbing menggunakan algoritma *winning* merupakan gambaran komponen dan struktur *database* yang digunakan dalam pembuatan sistem. ERD yang diimplementasikan pada sistem ini terdiri dari 7 entitas sebagaimana terlihat pada Gambar 4.6



Gambar 4.5 Class Diagram



Gambar 4.6 Entity Relationship Diagram

4.4 Implementasi

Desain yang telah dibuat akan diimplementasikan ke dalam kode program. Beberapa hal yang dilakukan dalam tahap implementasi antara lain:

- Penulisan kode program (*coding*) menggunakan bahasa pemrograman *Page Hyper Text Pre-Processor (PHP)* dengan bantuan *framework Code Igniter (CI)*.
- Manajemen basis data menggunakan *DBMS MySQL*.

Kode program perhitungan algoritma *winnowing* terdapat di *class C_pengajuan* pada *package controllers* dan pada *class M_winnowing* pada *package models*. Penulisan kode program perhitungan metode tersebut dapat dilihat pada *Listing Program 4.1*, *Listing Program 4.2*, *Listing Program 4.3* dan *Listing Program 4.4*.

Listing Program 4.1 Kode program function metode *Winnowing()*

No	Kode
36	<code>function metodeWinnowing(\$keyword,\$keywordDosen){</code>
37	<code> \$gram = 3;</code>
38	<code> \$window = 2;</code>
39	<code> \$teks1NonKarakter = \$this->M_winnowing->removeSpecialChar(\$keyword);</code>
40	<code> \$teks2NonKarakter = \$this->M_winnowing->removeSpecialChar(\$keywordDosen);</code>
41.	<code> \$teks1Split = \$this->M_winnowing->split(\$teks1NonKarakter, \$gram);</code>
42	<code> \$teks2Split = \$this->M_winnowing->split(\$teks2NonKarakter, \$gram);</code>
43	<code> \$teks1Hash = \$this->M_winnowing->hashSemuaGram(\$teks1Split);</code>
44	<code> \$teks2Hash = \$this->M_winnowing->hashSemuaGram(\$teks2Split);</code>
45	<code> \$teks1Window = \$this->M_winnowing->window(\$teks1Hash, \$window);</code>
46	<code> \$teks2Window = \$this->M_winnowing->window(\$teks2Hash, \$window);</code>
47	<code> \$window1Terkecil = \$this->M_winnowing->windowTerkecil(\$teks1Window, \$window);</code>
48	<code> \$window2Terkecil = \$this->M_winnowing->windowTerkecil(\$teks2Window, \$window);</code>
49	<code> \$jmlHash = \$this->M_winnowing->jmlHash(\$window1Terkecil, \$window2Terkecil);</code>
50	<code> \$irisanHash = \$jmlHash['n'];</code>
51	<code> \$gabunganHash = \$jmlHash['u'];</code>
52	<code> if(\$irisanHash==null){</code>
53	<code> return \$p=0;</code>
54	<code> }</code>
55	<code> else{</code>
56	<code> return \$p =substr(((\$irisanHash / \$gabunganHash) * 100), 0, 4);</code>
57	<code> }</code>
58	<code>}</code>

Pada *Listing Program 4.1* merupakan kode program *function* `metodeWinnowing($keyword, $keywordDosen)` yang berfungsi untuk melakukan perhitungan *string matching* dalam sistem untuk mengetahui persentase kemiripan antara kata kunci topik mahasiswa dengan kata kunci penelitian dosen. Tahapan perhitungan *string matching* terdapat pada baris 39-51 yang dilakukan dengan memanggil *function* yang ada pada `M_winnowing`. Tahap untuk menghitung besar persentase kemiripan *string* terdapat pada baris 56. Pada baris 52-54 merupakan kondisi apabila irisan yang dihasilkan bernilai kosong, maka seharusnya hasil dari perhitungan adalah tak terhingga. Untuk menghindari hasil tak terhingga maka diperlukan kondisi apabila irisan hash yang dihasilkan kosong, maka hasil perhitungan adalah nol. Hasil persentase akan disimpan pada `M_persetujuan` dan ditampilkan pada *function* `tampilHalamanRekomendasi()` dengan variabel input `$nim` yang akan dijelaskan pada *Listing Program 4.2*.

Tahap rekomendasi dimulai dari mencocokkan kata kunci yang dibandingkan. Kode program untuk menampilkan rekomendasi dosen yang terdapat pada *function* tampilHalamanRekomendasi() dapat dilihat pada *Listing Program 4.2*.

Listing Program 4.2 Kode program *function* tampilHalamanRekomendasi()

No	Kode
59.	function tampilHalamanRekomendasi(\$nim) {
60.	\$Id="";
61.	\$judul = \$this->M_pengajuan->getKeywordMahasiswa(\$nim)->row();
62.	\$dosen = \$this->M_dosen->getPembimbingUtamaRekomendasi();
63.	\$dosen2= \$this->M_dosen->getPembimbingPendampingRekomendasi();
64.	foreach (\$dosen as \$row) {
65.	\$persentase=\$this->metodeWinnowing(\$judul->keyword1, \$row['keyword']);
66.	\$per2=\$this->metodeWinnowing(\$judul->keyword2, \$row['keyword']);
67.	\$per3=\$this->metodeWinnowing(\$judul->keyword3, \$row['keyword']);
68.	\$this->M_dosen->simpanPersentase(\$persentase,\$per2, \$per3, \$row['nip'], \$row['keyword']);
69.	}
70.	foreach (\$dosen2 as \$row) {
71.	\$persentase=\$this->metodeWinnowing(\$judul->keyword1, \$row['keyword']);
72.	\$per2=\$this->metodeWinnowing(\$judul->keyword2, \$row['keyword']);
73.	\$per3=\$this->metodeWinnowing(\$judul->keyword3, \$row['keyword']);
74.	\$this->M_dosen->simpanPersentase(\$persentase,\$per2, \$per3, \$row['nip'], \$row['keyword']);
75.	}
76.	\$Id = \$this->input->post('inibidang');
77.	\$nip = \$this->input->post('nip');
78.	\$view = array(
79.	'biodata' => \$this ->M_pengajuan-> detailPengajuan(\$nim),
80.	'bidang' => \$this ->M_dosen-> getBidang()->result(),
81.	'dosen'=>\$this->M_dosen->sorting(),
82.	'dosen2'=>\$this->M_dosen->sortingPendamping(),
83.	'bimbingan' => \$this ->M_dosen-> getBimbingan(),
84.	'idBidang'=>\$this->M_pengajuan->hasilBidang(\$Id
85.);
86.	\$this->load->view('kombi/header');
87.	\$this->load->view('kombi/v_rekomendasi',\$view);
88.	\$this->load->view('footer');
89.	}

Pada *Listing Program 4.2* merupakan kode program yang ada pada *function* tampilHalamanRekomendasi() yang berfungsi menampilkan persentase perhitungan yang akan disimpan pada *class* M_dosen. Pada *function* tersebut terdapat pengambilan *keyword* milik mahasiswa, pengambilan *keyword* untuk pembimbing utama dan *keyword* pembimbing pendamping. Proses perhitungan

rekomendasi pembimbing utama dilakukan tiga kali dengan membandingkan tiga keyword masukan dari proses pengajuan yang kemudian disimpan pada tiga kolom persentase seperti pada baris 64-69. Perhitungan untuk menghasilkan rekomendasi dosen pembimbing pendamping hampir sama dengan perhitungan pembimbing utama, hanya berbeda pada data dosen yang diambil seperti pada baris 70-75. Hasil dari proses perhitungan akan ditampung oleh variabel \$view pada baris 78-85 yang kemudian akan ditampilkan dalam bentuk view seperti pada baris 86-88.

Tahapan lengkap metode *string matching* algoritma winnowing terdapat pada class M_winning yang sebelumnya telah dipanggil dalam *Listing Program 4.1* dan *Listing Program 4.2*. Kode program untuk menghitung *similarity* mulai dari tahap preprocessing sampai dengan menghasilkan nilai fingerprint dapat dilihat pada *class M_winning* pada *Listing Program 4.3*.

Listing Program 4.3 Kode program M_winning()

No	Kode
14	class M_winning extends CI_Model
15	{
16	function __construct() {
17	parent::__construct();
18	}
19	function removeSpecialChar(\$teks) {
20	\$a = \$teks;
21	\$b = preg_replace("/[^a-z_\-\.]/i", "-", \$a);
22	\$c = explode("-", \$b);
23	\$e = "";
24	\$f = "";
25	for (\$d = 0; \$d < count(\$c); \$d++) {
26	if (trim(\$c[\$d]) != "")
27	\$e .= \$c[\$d] . " ";
28	}
29	\$e = strtolower(substr(\$e, 0, strlen(\$e) - 1));
30	\$f = str_replace(array(" ", ".", "_", ",", " "), "", \$e);
31	return \$f;
32	}
33	function split(\$teks, \$gram) {
34	\$i = 0;
35	\$length = strlen(\$teks);

Dilanjutkan..

Lanjutan..

No	Kode
36.	for (\$i; \$i <= \$length - \$gram; \$i++) {
37.	\$teksSplit[] = substr(\$teks, \$i, \$gram);
38.	}
39.	return \$teksSplit;
40.	
41.	}
42.	
43.	function hashSemuaGram(\$gram) {
44.	\$hashGram = null;
45.	foreach (\$gram as \$a => \$teks) {
46.	\$hashGram[] = \$this->hash(\$teks);
47.	}
48.	return \$hashGram;
49.	}
50.	function hash(\$string) {
51.	\$basis = 11;
52.	\$jpgKarakter = strlen(\$string);
53.	\$hash = 0;
54.	for (\$i = 0; \$i < \$jpgKarakter; \$i++) {
55.	\$ascii = ord(\$string[\$i]);
56.	\$hash += \$ascii * pow(\$basis, \$jpgKarakter - (\$i + 1));
57.	}
58.	return \$hash;
59.	}
60.	function window(\$gramHash, \$window) {
61.	\$kumpulanWindow = null;
62.	\$i = 0;
63.	\$jpgArray = count(\$gramHash);
64.	while (\$i < \$jpgArray - \$window + 1) {
65.	\$j = \$i;
66.	for (\$j; \$j < \$window + \$i; \$j++) {
67.	\$kumpulanWindow[\$i][\$j - \$i] = \$gramHash[\$j];
68.	}
69.	\$i++;
70.	}
71.	return \$kumpulanWindow;
72.	}
73.	function windowTerkecil(\$kumpulanWindow, \$window) {
74.	\$jpgArray = count(\$kumpulanWindow);
75.	for (\$i = 0; \$i < \$jpgArray; \$i++) {
76.	for (\$j = 0; \$j < \$window; \$j++) {
77.	if (!isset(\$kumpulanWindow[\$i]['terkecil']) &&
78.	!isset(\$kumpulanWindow[\$i]['posisiTerkecil'])) {
79.	\$kumpulanWindow[\$i]['terkecil'] = \$kumpulanWindow[\$i][\$j];
80.	\$kumpulanWindow[\$i]['posisiTerkecil'] = \$j + \$i;
81.	}

Dilanjutkan..

Lanjutan..

No	Kode
82.	if (isset(\$kumpulanWindow[\$i][\$j])) {
83.	if (\$kumpulanWindow[\$i][\$j] < \$kumpulanWindow[\$i]['terkecil']) {
84.	\$kumpulanWindow[\$i]['terkecil'] = \$kumpulanWindow[\$i][\$j];
85.	\$kumpulanWindow[\$i]['posisiTerkecil'] = \$j + \$i;
86.	}
87.	}
88.	}
89.	}
90.	return \$kumpulanWindow;
91.	}
92.	function jmlHash(\$window1Terkecil, \$window2Terkecil) {
93.	\$stampung = "";
94.	\$fingerPrint1 = null;
95.	\$jpgArray = count(\$window1Terkecil);
96.	for (\$i = 0; \$i < \$jpgArray; \$i++) {
97.	if (\$stampung != \$window1Terkecil[\$i]['terkecil']) {
98.	\$fingerPrint1[] = \$window1Terkecil[\$i]['terkecil'];
99.	\$stampung = \$window1Terkecil[\$i]['terkecil'];
100.	}
101.	}
102.	\$stampung = "";
103.	\$fingerPrint2 = null;
104.	\$jpgArray = count(\$window2Terkecil);
105.	for (\$i = 0; \$i < \$jpgArray; \$i++) {
106.	if (\$stampung != \$window2Terkecil[\$i]['terkecil']) {
107.	\$fingerPrint2[] = \$window2Terkecil[\$i]['terkecil'];
108.	\$stampung = \$window2Terkecil[\$i]['terkecil'];
109.	}
110.	}
111.	\$stampung1 = null;
112.	\$stampung2 = null;
113.	\$jpgArray1 = count(\$fingerPrint1);
114.	\$jpgArray2 = count(\$fingerPrint2);
115.	for (\$i = 0; \$i < \$jpgArray1; \$i++) {
116.	\$sada = 0;
117.	for (\$j = 0; \$j < \$i; \$j++) {
118.	if (\$fingerPrint1[\$i] == \$fingerPrint1[\$j]) {
119.	\$sada = 1;
120.	}
121.	}
122.	if (\$sada == 0) {
123.	\$stampung1[] = \$fingerPrint1[\$i];
124.	}
125.	}

Dilanjutkan

Lanjutan..

No	Kode
126.	for (\$i = 0; \$i < \$pjgArray2; \$i++) {
127.	\$sada = 0;
128.	for (\$j = 0; \$j < \$i; \$j++) {
129.	if (\$fingerPrint2[\$i] == \$fingerPrint2[\$j]) {
130.	\$sada = 1;
131.	}
132.	}
133.	if (\$sada == 0) {
134.	\$stampung2[] = \$fingerPrint2[\$i];
135.	}
136.	}
137.	\$fingerPrint1 = \$stampung1;
138.	\$fingerPrint2 = \$stampung2;
139.	\$pjgArray1 = count(\$fingerPrint1);
140.	\$pjgArray2 = count(\$fingerPrint2);
141.	\$jml['n'] = 0;
142.	for (\$i = 0; \$i < \$pjgArray1; \$i++) {
143.	for (\$j = 0; \$j < \$pjgArray2; \$j++) {
144.	if (\$fingerPrint1[\$i] == \$fingerPrint2[\$j]) {
145.	\$jml['n']++;
146.	break;
147.	}
148.	}
149.	}
150.	\$jml['u'] = \$pjgArray1 + \$pjgArray2 - \$jml['n'];
151.	return \$jml;
152.	}
153.	}
154.	}

Pada *Listing Program 4.3* merupakan kode program pada class *M_winning* dimana untuk proses *preprocessing* terdapat pada *function* *removeSpecialChar()* yang berfungsi untuk menghilangkan karakter yang tidak relevan seperti spesial karakter dan angka yang terdapat pada baris 19-32. Tahapan selanjutnya yaitu proses pembentukan gram dari hasil *function* *removeSpecialChar()* yang terdapat pada *function* *split()* pada baris kode 33-40. Keluaran dari *function* *split()* akan diolah menjadi nilai hash berbentuk array pada *function* *hashSemuaGram()* 42-48 yang menggunakan *function* *hash()* pada baris 49-58 untuk menghasilkan array nilai hash, kemudian dilakukan proses pemetakan nilai hash kedalam window-window dengan panjang tiga seperti pada *function* *window()* baris 59-71. Hasil

dari *function* `window()` akan diproses untuk menghasilkan nilai fingerprint pada dua kata yang dibandingkan seperti pada *function* `windowTerkecil()` baris 72-90. Perhitungan persentase dilakukan setelah sistem mendapatkan dua nilai fingerprint yang akan dihitung jumlah fingerprint yang sama dari semua fingerprint yang dihasilkan. Proses untuk menghitung jumlah irisan dan gabungan fingerprint terdapat pada `jmlHash()` baris *code* 91- 154

4.5 Pengujian

Tahapan pengujian sistem merupakan suatu tahapan yang dilakukan secara sistematis untuk menguji dan mengevaluasi sistem dengan menggunakan sebuah metode pengujian sistem untuk mengetahui apakah kebutuhan sistem telah terpenuhi dan sistem layak untuk digunakan oleh pengguna. Agar pengujian yang dilakukan lebih valid, maka tahap pengujian sistem rekomendasi dosen pembimbing menggunakan algoritma *winnowing* ini dilakukan dengan dua metode, yaitu *white box* dan *black box*.

4.5.1 Metode White Box

Pengujian sistem dengan metode *white box* dilakukan untuk menguji sistem dari segi desain dan kode program. Hal tersebut bertujuan untuk mengevaluasi apakah sistem mampu menghasilkan fungsi-fungsi, inputan, dan keluaran yang sesuai dengan spesifikasi dari kebutuhan sistem itu sendiri. Pengujian dengan metode *whitebox* dilakukan oleh peneliti dengan cara menghitung *independent path* yaitu dengan menggunakan suatu pengukuran kuantitatif *cyclomatic complexity*, *listing* program, penentuan jalur independen, dan *test case*. Tahapan pengujian menggunakan metode *white box* meliputi:

1. *Listing* Program

Listing program merupakan baris-baris kode yang nantinya akan diuji. Setiap langkah dari kode-kode yang ada diberi nomor baik menjalankan *statement* biasa atau penggunaan kondisi dalam program.

2. Diagram alir

Diagram alir merupakan notasi yang digunakan untuk merepresentasikan aliran kontrol. Aliran kontrol yang digambarkan merupakan hasil penomoran dari listing program. Diagram alir digambarkan dengan node-node (simpul) yang dihubungkan dengan *edge-edge* (garis) yang menggambarkan alur jalannya program.

3. Kompleksitas siklomatik (*cyclomatic complexity*)

Kompleksitas siklomatik merupakan metrik perangkat lunak yang menyediakan ukuran kuantitatif dari kompleksitas logis suatu program. Bila digunakan dalam konteks teknik pengujian jalur dasar, nilai yang dihitung untuk kompleksitas siklomatik mendefinisikan jumlah jalur independen dalam basis set suatu program. Perhitungan kompleksitas siklomatik menggunakan persamaan 3:

$$V(G) = E - N + 2 \dots\dots\dots(3)$$

Keterangan :

V(G) : Kompleksitas siklomatik

E : Jumlah *edge*

N : Jumlah *node*

4. Jalur Independen (*Independent Path*)

Jalur independen adalah setiap jalur yang melalui program yang memperkenalkan setidaknya satu kumpulan pernyataan-pernyataan pemrosesan atau kondisi baru. Bila dinyatakan dalam grafik alir, jalur independen harus bergerak setidaknya sepanjang satu *edge* yang belum dilintasi sebelum jalur tersebut didefinisi.

5. Pengujian Basis Set (*Test Case*)

Pada bagian ini diberikan contoh data yang akan memaksa pelaksanaan jalur di basis set. Data yang dieksekusi dimasukkan ke dalam grafik alir apakah sudah melewati basis set yang tersedia. Sistem telah memenuhi syarat kelayakan perangkat lunak jika salah satu jalur yang dieksekusi setidaknya satu kali.

Pengujian sistem rekomendasi dosen pembimbing menggunakan algoritma *winnowing* menggunakan metode *white box* akan diterapkan pada fitur yang dinilai dapat mewakili sistem rekomendasi sebagai berikut.

A. Listing Program Algoritma Winoowing

```

36  function metodeWinoowing($keyword,$keywordDosen){
37      $gram = 3;
38      $window = 2;
39      $teks1NonKarakter = $this->M_winoowing->removeSpecialChar($keyword);
40      $teks2NonKarakter = $this->M_winoowing->removeSpecialChar($keywordDosen);
41      $teks1Split = $this->M_winoowing->split($teks1NonKarakter, $gram);
42      $teks2Split = $this->M_winoowing->split($teks2NonKarakter, $gram);
43      $teks1Hash = $this->M_winoowing->hashSemuaGram($teks1Split);
44      $teks2Hash = $this->M_winoowing->hashSemuaGram($teks2Split);
45      $teks1Window = $this->M_winoowing->window($teks1Hash, $window);
46      $teks2Window = $this->M_winoowing->window($teks2Hash, $window);
47      $window1Terkecil = $this->M_winoowing->windowTerkecil($teks1Window, $window);
48      $window2Terkecil = $this->M_winoowing->windowTerkecil($teks2Window, $window);
49      $jmlHash = $this->M_winoowing->jmlHash($window1Terkecil, $window2Terkecil);
50      $irisanHash = $jmlHash['n'];
51      $gabunganHash = $jmlHash['u'];
52      if($irisanHash==null){
53          return $p=0;
54      }
55      else{
56          return $p =substr((( $irisanHash / $gabunganHash ) * 100), 0, 4);
57      }
58  }

```

Gambar 4.7 Listing program function metodeWinoowing()

```

12  function removeSpecialChar($teks) {
13      $a = $teks;
14      $b = preg_replace("/[^a-z_\-\.\-]/i", "-", $a);
15      $c = explode("-", $b); //misah string berdasarkan tanda
16      $e = '';
17      $f = '';
18      for ($d = 0; $d < count($c); $d++) {
19          if (trim($c[$d]) != "")
20              $e .= $c[$d] . " ";
21      }
22      $e = strtolower(substr($e, 0, strlen($e) - 1));
23      $f = str_replace(array(" ", ".", "_", ",", ""), "", $e);
24      return $f;
25  }
26

```

Gambar 4.8 Listing program function removeSpesialChar()

```

39  function split($teks, $gram) {
40      $i = 0;
41      $length = strlen($teks);
42      for ($i; $i <= $length - $gram; $i++) {
43          $teksSplit[] = substr($teks, $i, $gram);
44      }
45      return $teksSplit;
46  }
47

```

Gambar 4.9 Listing program function split()

```

50     function hashSemuaGram($gram) {
51         $hashGram = null;
52         foreach ($gram as $a => $teks) {
53             $hashGram[] = $this->hash($teks);
54         }
55         return $hashGram;
56     }

```

Gambar 4.10 Listing program function hashSemuaGram()

```

63     function hash($string) {
64         $basis = 11;
65         $pjpgKarakter = strlen($string);
66         $hash = 0;
67         for ($i = 0; $i < $pjpgKarakter; $i++) {
68             $ascii = ord($string[$i]);
69             $hash += $ascii * pow($basis, $pjpgKarakter - ($i + 1));
70         }
71         return $hash;
72     }

```

Gambar 4.11 Listing program function hash()

```

77     function window($gramHash, $window) {
78         $kumpulanWindow = null;
79         $i = 0;
80         $pjpgArray = count($gramHash);
81         while ($i < $pjpgArray - $window + 1) {
82             $j = $i;
83             for ($j; $j < $window + $i; $j++) {
84                 $kumpulanWindow[$i][$j - $i] = $gramHash[$j];
85             }
86             $i++;
87         }
88         return $kumpulanWindow;
89     }

```

Gambar 4.12 Listing program function window()

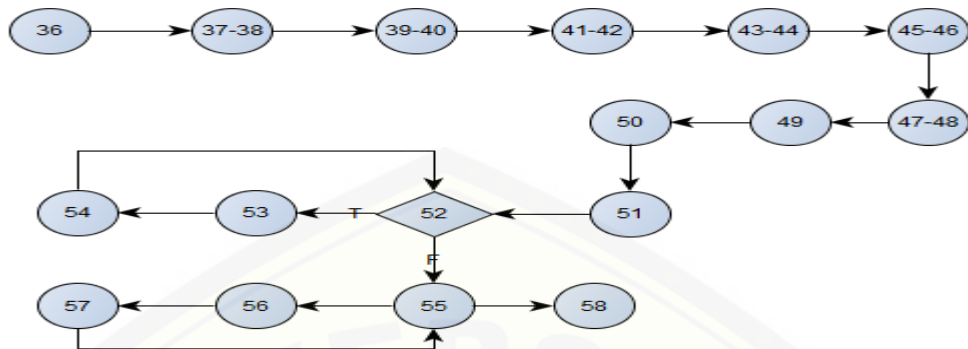
```

96     function windowTerkecil($kumpulanWindow, $window) {
97         $pjpgArray = count($kumpulanWindow);
98         for ($i = 0; $i < $pjpgArray; $i++) {
99             for ($j = 0; $j < $window; $j++) {
100                 if (!isset($kumpulanWindow[$i]['terkecil']) && !isset($kumpulanWindow[$i]['posisiTerkecil'])) {
101                     $kumpulanWindow[$i]['terkecil'] = $kumpulanWindow[$i][$j];
102                     $kumpulanWindow[$i]['posisiTerkecil'] = $j + $i;
103                 }
104                 if (isset($kumpulanWindow[$i][$j])) {
105                     if ($kumpulanWindow[$i][$j] < $kumpulanWindow[$i]['terkecil']) {
106                         $kumpulanWindow[$i]['terkecil'] = $kumpulanWindow[$i][$j];
107                         $kumpulanWindow[$i]['posisiTerkecil'] = $j + $i;
108                     }
109                 }
110             }
111         }
112         return $kumpulanWindow;
113     }

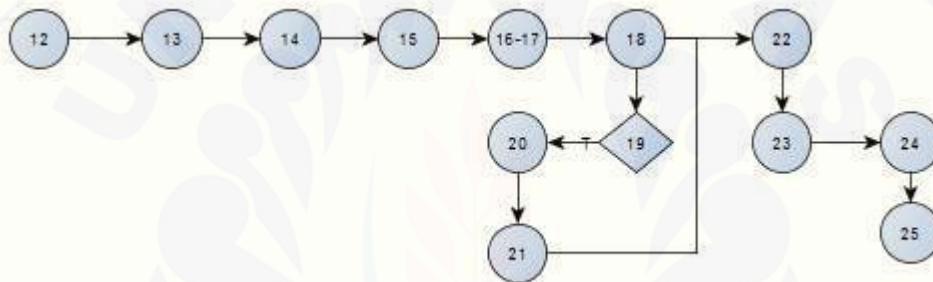
```

Gambar 4.13 Listing program function windowTerkecil()

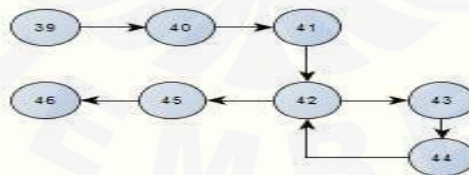
B. Diagram Alir



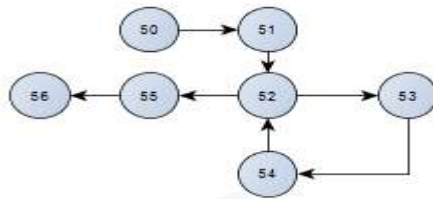
Gambar 4.14 Diagram alir *function* metodeWinnowing()



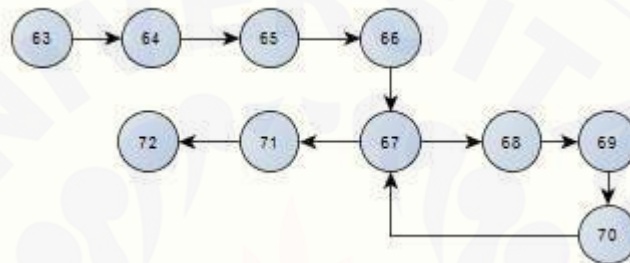
Gambar 4.15 Diagram alir *function* removeSpecialChar()



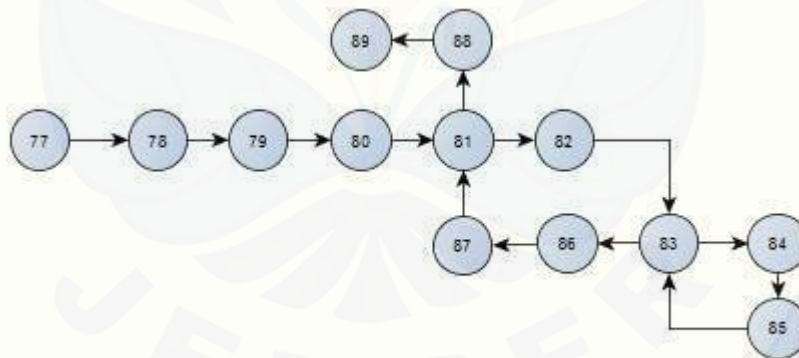
Gambar 4.16 Diagram Alir *function* split()



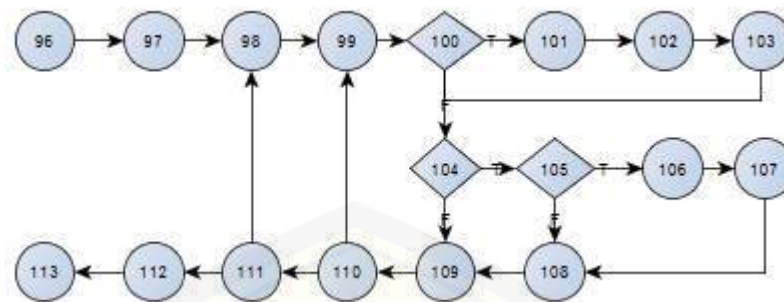
Gambar 4.17 Diagram Alir *function* hashSemuaGram()



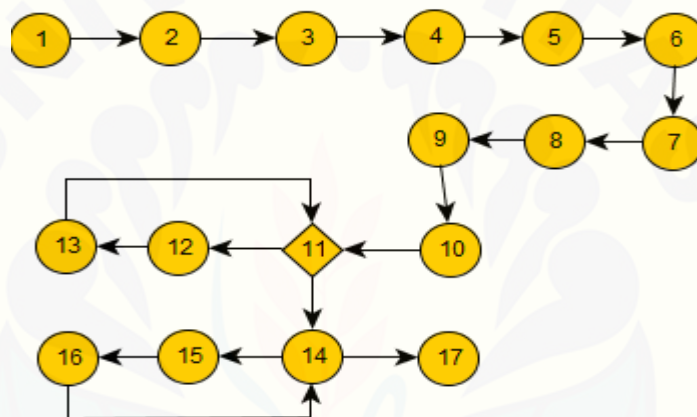
Gambar 4.18 Diagram alir *function* hash()



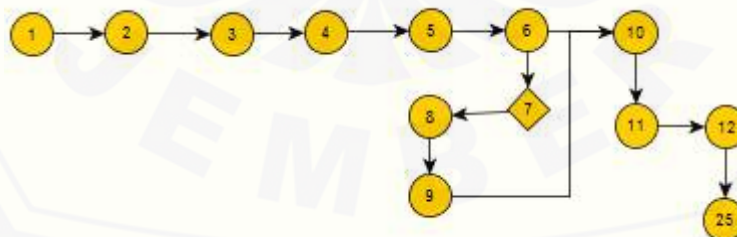
Gambar 4.19 Diagram alir *function* window()



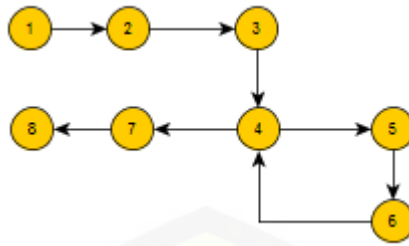
Gambar 4.20 Diagram alir *function* windowTerkecil()



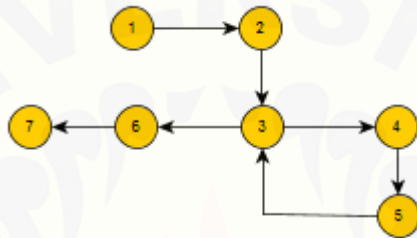
Gambar 4.21 Grafik alir *function* metodeWinnowing()



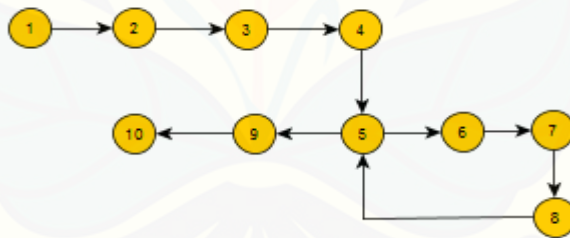
Gambar 4.22 Grafik alir *function* removeSpecialChar()



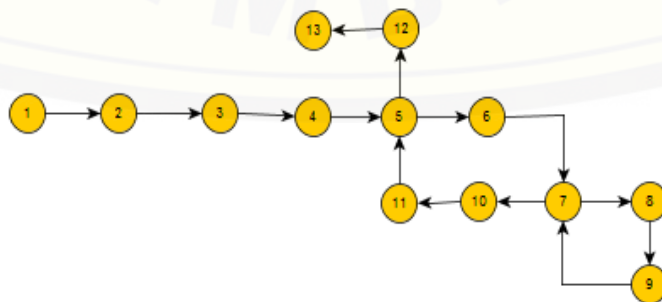
Gambar 4.23 Grafik alir *function* split()



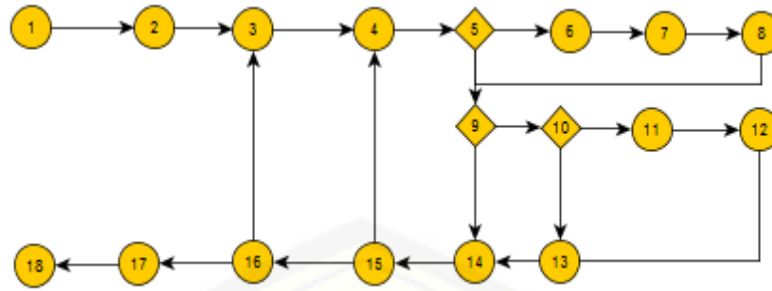
Gambar 4.24 Grafik alir *function* hashSemuaGram()



Gambar 4.25 Grafik alir *function* hash()



Gambar 4.26 Grafik alir *function* window()

Gambar 4.27 Grafik alir *function* windowTerkecil()C. Kompleksitas Siklomatik (*Cyclomatic Complexity*)

1. Function metode Winoinging ()

$$V(G) = E - N + 2$$

$$V(G) = 17 - 17 + 2 = 2$$

2. Function tampilHalamanRekomendasi()

$$V(G) = E - N + 2$$

$$V(G) = 21 - 20 + 2 = 3$$

3. Function removeSpecialChar()

$$V(G) = E - N + 2$$

$$V(G) = 13 - 13 + 2 = 2$$

4. Function split()

$$V(G) = E - N + 2$$

$$V(G) = 8 - 8 + 2 = 2$$

5. Function hashSemuaGram()

$$V(G) = E - N + 2$$

$$V(G) = 7 - 7 + 2 = 2$$

6. Function window()

$$V(G) = E - N + 2$$

$$V(G) = 14 - 13 + 2 = 3$$

7. Function windowTerkecil()

$$V(G) = E - N + 2$$

$$V(G) = 22 - 18 + 2 = 6$$

D. Jalur Independen (*Independent Path*)

1. Function algoritmaWinnowing()

Jalur 1 = 1-2-3-4-5-6-8-9-10-11-14-17

Jalur 2 = 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17

2. Function removeSpecialChar()

Jalur 1 = 1-2-3-4-5-6-10-11-12-13

Jalur 2 = 1-2-3-4-5-6-7-8-9-10-11-12-13

3. Function split()

Jalur 1 = 1-2-3-4-7-8

Jalur 2 = 1-2-3-4-5-6-7-8

4. Function hashSemuaGram()

Jalur 1 = 1-2-3-6-7

Jalur 2 = 1-2-3-4-5-6-7

5. Function hash()

Jalur 1 = 1-2-3-4-5-9-10

Jalur 2 = 1-2-3-4-5-6-7-8-9-10

6. Function window()

Jalur 1 = 1-2-3-4-5-12-13

Jalur 2 = 1-2-3-4-5-6-7-10-11-12-13

Jalur 3 = 1-2-3-4-5-6-7-8-9-10-11-12-13

7. Function windowTerkecil()

Jalur 1 = 1-2-3-4-5-9-14-15-16-17-18

Jalur 2 = 1-2-3-4-5-6-7-8-9-14-15-16-17-18

Jalur 3 = 1-2-3-4-5-6-7-8-9-10-13-14-15-16-17-18

Jalur 4 = 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18

E. Pengujian Basis Set (*Test Case*)Tabel 4.4 *Test case function* metode `Winnowing()`

<i>Test case function</i> metode <code>Winnowing()</code>	
Jalur 1	
<i>Test Case</i>	Jika <code>if(\$irisanHash==null)</code> bernilai benar
Target yang diharapkan	Gagal menghitung hasil persentase dan menampilkan persentase kecocokan 0%
Hasil pengujian	Benar
Path/Jalur	1-2-3-4-5-6-8-9-10-11-14-17
Jalur 2	
<i>Test Case</i>	Jika <code>if(\$irisanHash==null)</code> bernilai salah
Target yang diharapkan	Berhasil menghitung hasil persentase dan menampilkan persentase kecocokan sesuai hasil
Hasil pengujian	Benar
Path/Jalur	1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17

Tabel 4.5 *Test case function* `removeSpecialChar()`

<i>Test case function</i> <code>removeSpecialChar()</code>	
Jalur 1	
<i>Test Case</i>	Jika <code>for (\$d = 0; \$d < count(\$c); \$d++)</code> pada <code>node</code> 6 tidak dilakukan
Target yang diharapkan	Gagal menghilangkan tanda spasi pada teks yang diproses
Hasil pengujian	Benar
Path/Jalur	1-2-3-4-5-6-10-11-12-13

Jalur 2	
<i>Test Case</i>	Jika <i>for (\$d = 0; \$d < count(\$c); \$d++)</i> dilakukan pada <i>node 6</i> , maka jika <i>if (trim(\$c[\$d]) != "")</i>
Target yang diharapkan	Berhasil menghilangkan tanda spasi pada teks
Hasil pengujian	Benar
Path/Jalur	1-2-3-4-5-6-7-8-9-10-11-12-13

Tabel 4.6 *Test case function split()*

Test case function split()	
Jalur 1	
<i>Test Case</i>	Jika <i>for (\$i; \$i <= \$length - \$gram; \$i++)</i> pada <i>node 4</i> tidak dilakukan
Target yang diharapkan	Gagal melakukan proses <i>split</i> pada teks
Hasil pengujian	Benar
Path/Jalur	1-2-3-4-7-8
Jalur 2	
<i>Test Case</i>	Jika <i>for (\$i; \$i <= \$length - \$gram; \$i++)</i> pada <i>node 3</i> dilakukan
Target yang diharapkan	Berhasil melakukan proses <i>split</i> pada teks
Hasil pengujian	Benar
Path/Jalur	1-2-3-4-5-6-7-8

Tabel 4.7 *Test case function* hashSemuaGram()

Test case function hashSemuaGram ()	
Jalur 1	
<i>Test Case</i>	Jika <i>foreach (\$gram as \$a => \$teks)</i> tidak dilakukan pada <i>node 3</i>
Target yang diharapkan	Gagal menghitung nilai hash untuk semua inputan <i>gram</i>
Hasil pengujian	Benar
Path/Jalur	1-2-3-6-7
Jalur 2	
<i>Test Case</i>	Jika <i>foreach (\$gram as \$a => \$teks)</i> dilakukan pada <i>node 3</i>
Target yang diharapkan	Berhasil menghitung nilai hash pada semua gram dan ditampilkan dalam variabel <i>\$hashGram</i>
Hasil pengujian	Benar
Path/Jalur	1-2-3-4-5-6-7

Tabel 4.8 *Test case function* hash()

Test case function hash()	
Jalur 1	
<i>Test Case</i>	Jika <i>for (\$i = 0; \$i < \$jpgKarakter; \$i++)</i> tidak dilakukan pada <i>node 5</i>
Target yang diharapkan	Gagal menghitung nilai hash yang diinputkan
Hasil pengujian	Benar
Path/Jalur	1-2-3-4-5-9-10

Jalur 2	
<i>Test Case</i>	Jika <i>for</i> ($\$i = 0; \$i < \$pjtKarakter; \$i++$) dilakukan pada <i>node 5</i>
Target yang diharapkan	Berhasil menghitung nilai hash dan ditampung dalam variabel $\$hash$ pada <i>gram</i>
Hasil pengujian	Benar
Path/Jalur	1-2-3-4-5-6-7-8-9-10

Tabel 4.9 *Test case function window()*

Test case function window()	
Jalur 1	
<i>Test Case</i>	Jika <i>while</i> ($\$i < \$pjtArray - \$window + 1$) tidak dilakukan pada <i>node 5</i>
Target yang diharapkan	Gagal memetakan nilai hash kedalam window pada $\$kumpulanWindow[\$i]$
Hasil pengujian	Benar
Path/Jalur	1-2-3-4-5-12-13
Jalur 2	
<i>Test Case</i>	Jika <i>for</i> ($\$j; \$j < \$window + \$i; \$j++$) tidak dilakukan pada <i>node 7</i>
Target yang diharapkan	Gagal memetakan nilai hash kedalam window pada $\$kumpulanWindow[\$i][\$j - \$i]$
Hasil pengujian	Benar
Path/Jalur	1-2-3-4-5-6-7-10-11-12-13

Jalur 3	
<i>Test Case</i>	Jika <i>for</i> (\$j; \$j < \$window + \$i; \$j++) dilakukan pada <i>node 7</i>
Target yang diharapkan	Berhasil memetakan nilai hash kedalam window dan menyimpan dalam variabel \$kumpulanWindow
Hasil pengujian	Benar
Path/Jalur	1-2-3-4-5-6-7-8-9-10-11-12-13

Tabel 4.10 *Test case function windowTerkecil()*

Test case function windowTerkecil()	
Jalur 1	
<i>Test Case</i>	Jika <i>if</i> (!isset(\$kumpulanWindow[\$i]['terkecil']) && !isset(\$kumpulanWindow[\$i]['posisiTerkecil'])) tidak dilakukan pada <i>node 5</i>
Target yang diharapkan	Gagal menentukan nilai fingerprint atau nilai <i>window</i> terkecil dan posisi <i>fingerprint</i>
Hasil pengujian	Benar
Path/Jalur	1-2-3-4-5-9-14-15-16-17-18
Jalur 2	
<i>Test Case</i>	Jika <i>if</i> (isset(\$kumpulanWindow[\$i][\$j])) pada <i>node 9</i> tidak dilakukan
Target yang diharapkan	Gagal memetakan nilai <i>window</i> terkecil dan gagal menentukan letak posisi <i>window</i>
Hasil pengujian	Benar
Path/Jalur	1-2-3-4-5-6-7-8-9-14-15-16-17-18

Jalur 3	
<i>Test Case</i>	Jika <i>if</i> (<i>\$kumpulanWindow</i> [<i>\$i</i>] [<i>\$j</i>]< <i>\$kumpulanWindow</i> [<i>\$i</i>] [' <i>terkecil</i> ']) pada node 10 tidak dilakukan
Target yang diharapkan	Gagal memetakan nilai hash kedalam <i>window</i> terkecil dan gagal mengetahui letak pada posisi <i>window</i>
Hasil pengujian	Benar
Path/Jalur	1-2-3-4-5-6-7-8-9-10-13-14-15-16-17-18
Jalur 4	
<i>Test Case</i>	Jika <i>if</i> (<i>\$kumpulanWindow</i> [<i>\$i</i>] [<i>\$j</i>]< <i>\$kumpulanWindow</i> [<i>\$i</i>] [' <i>terkecil</i> ']) pada node 10 dilakukan
Target yang diharapkan	Berhasil memetakan nilai hash kedalam <i>window</i> terkecil dan berhasil mengetahui letak pada posisi <i>window</i>
Hasil pengujian	Benar
Path/Jalur	1-2-3-4-5-6-7-8-9-10-13-14-15-16-17-18

4.5.2 Metode Black Box

Pengujian *black box* dilakukan untuk mengetahui apakah masukan dan keluaran dari sistem sesuai dengan kebutuhan fungsional atau tidak. Pengujian dilakukan pada form untuk setiap *usecase*. Hasil pengujian *black box usecase* dapat dilihat pada Lampiran E (Pengujian *Black Box*).

BAB 6. PENUTUP

Bab ini berisi mengenai kesimpulan dan saran dari peneliti tentang penelitian yang telah dilakukan. Kesimpulan dan saran tersebut diharapkan dapat digunakan sebagai acuan pada penelitian selanjutnya.

6.1 Kesimpulan

Kesimpulan dari penelitian yang telah dilakukan adalah:

1. Sistem rekomendasi penentuan dosen pembimbing mampu memberikan rekomendasi sesuai dengan kompetensi dosen menggunakan algoritma *winnowing* sebagai pendeteksi kesamaan kata kunci sehingga dihasilkan rekomendasi dosen pembimbing.
2. Hasil rekomendasi dosen menggunakan algoritma *winnowing* diperoleh dari perhitungan persentase kesamaan kata kunci topik dan penelitian. Persentase yang dihasilkan akan menjadi hasil rekomendasi dosen, dimana rekomendasi diurutkan berdasarkan nilai persentase terbesar.
3. Besar nilai gram dan window berpengaruh pada persentase yang dihasilkan. Semakin kecil nilai gram dan window, maka keakuratan perhitungan semakin tinggi, dan sebaliknya. Nilai gram yang diberikan pada sistem rekomendasi yaitu 2 dan nilai window 3. Perhitungan kesamaan menggunakan algoritma *winnowing* dan *jaccard coefficient* dengan kata kunci yang sama akan menghasilkan persentase sebesar 100%. Hasil persentase keseluruhan kata kunci didapat dengan menghitung nilai rata-rata pada semua persentase pencocokan kata kunci. Semakin banyak jumlah kata kunci penelitian dosen, maka rekomendasi yang dihasilkan bersifat umum, sedangkan semakin sedikit kata kunci penelitian maka rekomendasi akan bersifat khusus.
4. Perhitungan penentuan dosen pembimbing skripsi menggunakan algoritma *winnowing* dari dua sampel didapatkan persentase untuk dosen Saiful Bukhori sebesar 16,67 % dan dosen Slamain sebesar 1,18% yang menunjukkan

bahwa dari sampel tersebut tingkat persentase dipengaruhi oleh jumlah penelitian dan jumlah kesamaan kata kunci.

6.2 Saran

Adapun saran yang ditujukan untuk memberikan masukan yang lebih baik yaitu :

1. Sistem rekomendasi dosen pembimbing yang akan dikembangkan selanjutnya perlu ditambah bobot alasan pemilihan dosen oleh mahasiswa, sehingga rekomendasi yang di dapat akan lebih akurat.
2. Sistem rekomendasi dosen pembimbing yang akan dikembangkan selanjutnya akan lebih baik jika dapat mendeteksi sinonim kata sehingga persentase yang dihasilkan lebih tinggi dan memiliki tingkat akurasi rekomendasi yang besar.

DAFTAR PUSTAKA

Khoiri, S. W. (2015). *Pedoman Penulisan Skripsi FKM UNEJ*. Jember: FKM UNEJ.

Kurniawati, W. (2008). Perbandingan pendekatan Deteksi Plagiarism Dokumen Dalam Bahasa Inggris. *Seminar Ilmiah Nasional Komputer dan Sistem Intelijen* , 255-260.

Pertiwi. (2015). Implementasi Algoritma Pencocokan String pada Aplikasi Pengarsipan Berbasis Web. *EXTERNAL JOURNAL* , 1-8.

Presman, R. (2001). *Software Engineering a practitioner's approach (5 ed.)*. New York: McGraw-Hill.

Priantara, P. Y. (2011). *Implementasi Deteksi Penjiplakan Dengan Algoritma Winnowing Pada Dokumen Terkelompok*. Surabaya: Institut Teknologi Sepuluh November.

Salam, W. d. (2015). Sistem Rekomendasi Penentuan Dosen Pembimbing Tugas Akhir Dengan Menggunakan Algoritma Rabin-Karp. *Jurnal Techno.COM* , 225-233.

Salmuasih. (2013). *Perancangan Sistem Deteksi Plagiat Pada Dokumen Teks Dengan Konsep Similarity Menggunakan Algoritma Rabin Karp*. Yogyakarta: STMIK Amikom Yogyakarta.

LAMPIRAN

Lampiran A. Use Case Skenario

A.1 Use Case Skenario Login

Tabel A.1 Use case skenario login

ID	USC-01	
Name	Login	
Participating Actor	Semua Pengguna	
Entry Condition	Pengguna akan mengakses sistem	
Exit Condition	Pengguna telah mengakses sistem	
SKENARIO UTAMA "Login"		
	Aksi Aktor	Reaksi Sistem
	<ol style="list-style-type: none"> 1. Mengakses halaman <i>login</i> 2. Memasukkan : <ol style="list-style-type: none"> a. Username b. Password 3. Menekan tombol <i>login</i> 	<ol style="list-style-type: none"> 4. Menampilkan halaman dashboard
SKENARIO ALTERNATIF "Username atau password salah"		
	Aksi Aktor	Reaksi Sistem
	<ol style="list-style-type: none"> 2. Memasukkan : <ol style="list-style-type: none"> c. Username d. Password 3. Menekan tombol <i>login</i> 	<ol style="list-style-type: none"> 4. Menampilkan pesan "Login Gagal! Username atau Password Salah"

SKENARIO ALTERNATIF**“kolom belum diisi”**

Aksi Aktor	Reaksi Sistem
5. Menekan tombol <i>login</i>	6. Menampilkan pesan pada kolom yang belum diisi

A.2 *Use Case* Skenario Melihat Daftar Pengajuan ProposalTabel A.2 *Use Case* Skenario Melihat Daftar Pengajuan Proposal

ID	USC-03
Name	Melihat Daftar Pengajuan Proposal
Participating Actor	Akademik dan Komisi Bimbingan
Entry Condition	Pengguna akan melihat daftar pengajuan proposal
Exit Condition	Pengguna telah melihat daftar pengajuan proposal

SKENARIO UTAMA**“Melihat Daftar Pengajuan Proposal”**

Aksi Aktor	Reaksi Sistem
1. Klik menu pengajuan	2. Menampilkan halaman daftar pengajuan proposal yang berisi tabel pengajuan

A.3 *Use Case* Skenario Mengubah Pengajuan Proposal

Tabel A.3 Use Case Skenario Mengubah Pengajuan Proposal

ID	USC-04
Name	Mengubah Pengajuan Proposal
Participating Actor	Akademik
Entry Condition	Akademik akan mengubah pengajuan proposal
Exit Condition	Akademik telah mengubah pengajuan proposal

SKENARIO UTAMA**“Mengubah Pengajuan Proposal”**

Aksi Aktor	Reaksi Sistem
1. Klik menu pengajuan	2. Menampilkan halaman daftar pengajuan proposal
3. Klik tombol <i>edit</i> pada kolom <i>Action</i>	4. Menampilkan halaman ubah pengajuan proposal
5. Mengubah form dengan <i>field</i> : <ol style="list-style-type: none"> NIM Nama Judul Keyword 1 Keyword 2 Keyword 3 Pembimbing Utama Pembimbing Pendamping 	
6. Klik tombol simpan	7. Menampilkan halaman daftar pengajuan proposal dengan pesan “berhasil edit data”

SKENARIO ALTERNATIF**“Isi Pembimbing Utama dan Pendamping (Sama)”****Aksi Aktor****Reaksi Sistem**

6. Klik tombol simpan

7. Menampilkan pesan “pilih pembimbing yang berbeda”

SKENARIO ALTERNATIF**“Isian Kosong”****Aksi Aktor****Reaksi Sistem**

6. Klik tombol simpan

7. Menampilkan pesan pada kolom yang kosong

A.4 *Use Case* Skenario Melihat Data Dosen*Tabel A.4 Use case skenario melihat data dosen*

ID	USC-05
Name	Melihat Data Dosen
Participating Actor	Akademik dan Komisi Bimbingan
Entry Condition	Pengguna akan melihat Data Dosen
Exit Condition	Pengguna telah melihat Data Dosen

SKENARIO UTAMA**“Melihat Data Dosen”****Aksi Aktor****Reaksi Sistem**

1. Klik menu dosen

2. Klik submenu daftar dosen

3. Menampilkan halaman daftar dosen yang berisi:
 - a. No
 - b. NIP
 - c. Nama
 - d. Tempat & Tgl Lahir
 - e. Jabatan
 - f. Pangkat/Gol
-

A.5 Use Case Skenario Melihat Persetujuan Pengajuan Proposal

Tabel A.5 Use case skenario melihat persetujuan pengajuan proposal

ID	USC-06
Name	Melihat persetujuan pengajuan proposal
Participating Actor	Akademik dan Komisi Bimbingan
Entry Condition	Pengguna akan melihat persetujuan proposal
Exit Condition	Pengguna akan melihat persetujuan proposal

SKENARIO UTAMA

“Melihat Persetujuan Pengajuan Proposal”

Aksi Aktor	Reaksi Sistem
1. Klik menu proposal	
2. Klik submenu persetujuan	
	3. Menampilkan halaman persetujuan yang berisi: <ol style="list-style-type: none"> a. NIM b. Nama c. Judul d. No. Surat e. Tgl Surat f. Action

A.6 Use Case Skenario Menambah Bidang Keahlian Dosen

Tabel A.6 Use Case Skenario Menambah Bidang Keahlian Dosen

ID	USC-07
Name	Menambah Bidang Keahlian Dosen
Participating Actor	Akademik
Entry Condition	Akademik akan menambah bidang keahlian dosen
Exit Condition	Akademik telah menambah bidang keahlian dosen

SKENARIO UTAMA**“Menambah Bidang Keahlian Dosen”**

Aksi Aktor	Reaksi Sistem
1. Klik menu Dosen	
2. Klik submenu Bidang Dosen	
	3. Menampilkan halaman data bidang yang berisi: <ol style="list-style-type: none"> No Nama Bidang Action
4. Klik tombol Tambah pada kolom Action	
	5. Menampilkan halaman form tambah data bidang
6. Memilih nama dosen pada tabel	
7. Klik tombol simpan	
	8. Kembali ke halaman data bidang dan menampilkan pesan “berhasil di tambah”

A.7 Use Case Skenario Melengkapi Surat Tugas

Tabel A.7 Use case skenario melengkapi surat tugas

ID	USC-08
Name	Melengkapi Surat Tugas
Participating Actor	Akademik
Entry Condition	Akademik akan melengkapi surat tugas
Exit Condition	Akademik telah melengkapi surat tugas
SKENARIO UTAMA	
“Melihat Persetujuan Pengajuan Proposal”	
Aksi Aktor	Reaksi Sistem
<ol style="list-style-type: none"> 1. Klik menu proposal 2. Klik submenu persetujuan 4. Memilih tombol edit pada kolom <i>action</i> 6. Mengisi form yang meliputi: <ol style="list-style-type: none"> a. Tanggal Surat b. No Surat 7. Klik tombol Save 	<ol style="list-style-type: none"> 3. Menampilkan halaman persetujuan yang berisi: <ol style="list-style-type: none"> g. NIM h. Nama i. Judul j. No. Surat k. Tgl Surat l. <i>Action</i> 5. Menampilkan form lengkapi surat tugas 8. Menampilkan pesan, “berhasil diedit”

SKENARIO ALTERNATIF	
“Form kosong”	
Aksi Aktor	Reaksi Sistem
7. Klik tombol Save	8. Menampilkan pesan, “berhasil diedit”

A.8 Use Case Skenario Menambah Data Dosen

Tabel A.8 Use case skenario menambah data dosen

ID	USC-09
Name	Menambah Data Dosen
Participating Actor	Akademik
Entry Condition	Akademik akan menambah data dosen
Exit Condition	Akademik Akan menambah data dosen

SKENARIO UTAMA	
“Menambah Data Dosen”	
Aksi Aktor	Reaksi Sistem
1. Klik menu dosen	
2. Klik submenu daftar dosen	
	3. Menampilkan halaman daftar dosen yang berisi: <ul style="list-style-type: none"> g. No h. NIP i. Nama j. Tempat & Tgl Lahir k. Jabatan l. Pangkat/Gol m. <i>Action</i>
4. Klik tombol Tambah	
	5. Menampilkan halaman tambah dosen

Aksi Aktor	Reaksi Sistem
6. Mengisi form tambah dosen : a. NIP b. Nama c. Tempat Tanggal Lahir d. Pangkat/Gol e. Jabatan Fungsional 7. Klik save	8. Menampilkan halaman daftar dosen dan pesan “Berhasil tambah data”

SKENARIO ALTERNATIF

“Form kosong”

Aksi Aktor	Reaksi Sistem
10. Klik tombol Save	11. Menampilkan pesan pada form kosong

A.9 Use Case Skenario Melihat Bidang Keahlian Dosen

Tabel A.9 Use case skenario melihat bidang keahlian dosen

ID	USC-010
Name	Melihat Bidang Keahlian Dosen
Participating Actor	Akademik dan komisi bimbingan
Entry Condition	Pengguna akan melihat bidang dosen
Exit Condition	Pengguna telah melihat bidang dosen

SKENARIO UTAMA

“Melihat Bidang Keahlian Dosen”

Aksi Aktor	Reaksi Sistem
1. Klik menu dosen 2. Klik submenu daftar dosen	9. Menampilkan halaman daftar dosen yang berisi:

Aksi Aktor	Reaksi Sistem
	a. No b. NIP c. Nama d. Tempat & Tgl Lahir e. Jabatan f. Pangkat/Gol
10. Klik tombol bidang pada action	11. Menampilkan halaman daftar bidang dosen yang berisi a. No b. ID Bidang c. Nama Bidang

A.10 Use Case Skenario Mengubah Data Dosen

Tabel A.10 Use case skenario mengubah data dosen

ID	USC-11
Name	Mengubah Data Dosen
Participating Actor	Akademik
Entry Condition	Akademik akan mengubah data dosen
Exit Condition	Akademik telah mengubah data dosen

SKENARIO UTAMA

“Mengubah Data Dosen”

Aksi Aktor	Reaksi Sistem
1. Klik menu dosen	
2. Klik submenu daftar dosen	
	12. Menampilkan halaman daftar dosen yang berisi: g. No h. NIP i. Nama j. Tempat & Tgl Lahir k. Jabatan l. Pangkat/Gol m. Action

- 13. Klik Edit pada kolom action
- 14. Menampilkan halaman edit dosen
- 15. Mengubah data dosen yang berisi:
 - a. NIP
 - b. Nama
 - c. Tempat tanggal lahir
 - d. Pangkat/Gol
 - e. Jabatan Fungsional
- 16. Memilih tombol save
- 17. Menampilkan halaman daftar dosen dan pesan “Berhasil ubah data”

SKENARIO ALTERNATIF	
“Form kosong”	
Aksi Aktor	Reaksi Sistem
16. Klik tombol Save	17. Menampilkan pesan pada form kosong

A.11 *Use Case* Skenario Mencetak Surat Tugas

Tabel A.11 *Use case* skenario mencetak surat tugas

ID	USC-12
Name	Mencetak Surat Tugas
Participating Actor	Akademik
Entry Condition	Akademik akan mencetak surat tugas
Exit Condition	Akademik telah mencetak surat tugas

SKENARIO UTAMA	
“Mencetak Surat Tugas ”	
Aksi Aktor	Reaksi Sistem

- 1. Klik menu proposal
- 2. Klik submenu persetujuan

-
- 3. Menampilkan halaman persetujuan yang berisi:
 - m. NIM
 - n. Nama
 - o. Judul
 - p. No. Surat
 - q. Tgl Surat
 - r. *Action*
 - 4. Klik tombol Cetak
 - 5. Mencetak Surat Tugas dalam format .pdf
-

A.12 *Use Case* Skenario Manajemen Rekomendasi Dosen Pembimbing

Tabel A.12 *Use case* skenario manajemen rekomendasi dosen pembimbing

ID	USC-13
Name	Manajemen Rekomendasi Dosen Pembimbing
Participating Actor	Komisi Bimbingan
Entry Condition	Komisi Bimbingan akan menentukan dosen pembimbing
Exit Condition	Komisi Bimbingan akan menentukan dosen pembimbing
SKENARIO UTAMA	
“Manajemen Rekomendasi Dosen Pembimbing”	
Aksi Aktor	Reaksi Sistem

- 1. Klik menu pengajuan
 - 2. Menampilkan halaman daftar pengajuan yang berisi :
 - a. NIM
 - b. Nama
 - c. Judul
 - d. Pembimbing Utama
 - e. Pembimbing Pendamping
 - f. *Action*
 - 3. Klik tombol rekomendasi pada *action*
-

-
- 4. Menampilkan halaman rekomendasi dosen pembimbing
 - 5. Memilih pembimbing Utama
 - 6. Memilih pembimbing Pendamping
 - 7. Klik tombol Simpan
 - 8. Menampilkan halaman pengajuan.

SKENARIO ALTERNATIF

“Pembimbing Utama dan Pendamping sama ”

Aksi Aktor	Reaksi Sistem
7. Klik tombol Simpan	8. Menampilkan pesan “Pilih pembimbing berbeda”

SKENARIO ALTERNATIF

“Persentase bernilai nol”

Aksi Aktor	Reaksi Sistem
5. Memilih bidang yang sama dengan topik penelitian mahasiswa 6. Klik tombol cari 8. Pilih pembimbing utama 9. Pilih pembimbing pendamping 10. Pilih tombol simpan	7. Menampilkan rekomendasi dosen pembimbing 11. Menampilkan halaman pengajuan

A.13 *Use Case* Skenario Melihat Keyword Penelitian DosenTabel A.13 *Use case* skenario melihat keyword penelitian dosen

ID	USC-14
Name	Melihat Keyword Penelitian Dosen
Participating Actor	Akademik
Entry Condition	Akademik akan melihat keyword penelitian dosen
Exit Condition	Akademik telah melihat keyword penelitian dosen
SKENARIO UTAMA	
“Melihat Keyword Penelitian Dosen”	
Aksi Aktor	Reaksi Sistem
<ol style="list-style-type: none"> 1. Klik menu dosen 2. Klik Submenu Keyword 4. Memilih tombol detail pada kolom <i>Action</i> 	<ol style="list-style-type: none"> 3. Menampilkan data dosen yang berisi: <ol style="list-style-type: none"> a. No b. Nim c. Nama d. <i>Action</i> 5. Menampilkan Keyword Dosen

A.14 *Use Case* Skenario Mengubah Status Lulus MahasiswaTabel A.14 *Use case* skenario mengubah status lulus mahasiswa

ID	USC-15
Name	Mengubah Status Lulus Mahasiswa
Participating Actor	Akademik
Entry Condition	Akademik akan mengubah status lulus

mahasiswa

Exit Condition

Akademik telah mengubah status lulus mahasiswa

SKENARIO UTAMA

“Mengubah Status Lulus Mahasiswa ”

Aksi Aktor	Reaksi Sistem
1. Klik menu proposal	
2. Klik submenu persetujuan	
	3. Menampilkan halaman persetujuan yang berisi: <ul style="list-style-type: none"> a. NIM b. Nama c. Judul d. No. Surat e. Tgl Surat f. Action
4. Memilih tombol ubah status	
	5. Menampilkan pesan “Apakah anda yakin akan mengubah status ?”
6. Memilih tombol Ya	
	7. Menampilkan halaman persetujuan

A.15 *Use Case* Skenario Menambah Bidang Ilmu

Tabel A.15 *Use Case* Skenario Menambah Bidang Ilmu

ID	USC-16
Name	Menambah bidang ilmu
Participating Actor	Akademik
Entry Condition	Akademik akan menambah bidang ilmu
Exit Condition	Akademik telah menambah bidang

SKENARIO UTAMA	
“Menambah Bidang Ilmu ”	
Aksi Aktor	Reaksi Sistem
<ol style="list-style-type: none"> 1. Klik menu dosen 2. Klik submenu bidang 4. Klik tombol tambah 6. Mengisi form tambah bidang ilmu yang berisi: <ol style="list-style-type: none"> a. Nama Bidang 7. Klik tombol Save 	<ol style="list-style-type: none"> 3. Menampilkan data bidang yang berisi: <ol style="list-style-type: none"> a. No b. ID Bidang c. Nama d. Action 5. Menampilkan form tambah data bidang 8. Menampilkan halaman data bidang ilmu

A.16 *Use Case* Skenario Melihat Detail Pengajuan Proposal

Tabel A.16 *Use Case* Skenario Melihat Detail Pengajuan Proposal

ID	USC-17
Name	Melihat Detail Pengajuan Proposal
Participating Actor	Akademik
Entry Condition	Akademik akan melihat detail pengajuan proposal
Exit Condition	Aktor telah melihat detail pengajuan

SKENARIO UTAMA	
“Melihat Detail Pengajuan Proposal”	
Aksi Aktor	Reaksi Sistem
<ol style="list-style-type: none"> 1. Klik menu proposal 2. Klik submenu pengajuan 4. Klik tombol detail pada kolom action 	<ol style="list-style-type: none"> 3. Menampilkan halaman daftar pengajuan yang berisi: <ol style="list-style-type: none"> a. No b. Nim c. Nama d. Judul e. Pembimbing Utama f. Pembimbing Pendamping g. <i>Action</i> 5. Menampilkan halaman detail pengajuan pilihan yang berisi <ol style="list-style-type: none"> a. No b. Nim c. Nama d. Judul e. Pembimbing Utama f. Pembimbing Pendamping g. Keyword 1 h. Keyword 2 i. Keyword 3

A.17 *Use Case* Skenario Menambah *Keyword* Penelitian Dosen

Tabel A.17 *Use Case* Skenario Menambah *Keyword* Penelitian Dosen

ID	USC-18
Name	Menambah <i>Keyword</i> Penelitian Dosen
Participating Actor	Akademik
Entry Condition	Akademik akan menambah <i>keyword</i> penelitian dosen
Exit Condition	Akademik telah menambah <i>keyword</i>

SKENARIO UTAMA	
“Menambah Keyword Penelitian Dosen”	
Aksi Aktor	Reaksi Sistem
1. Klik menu dosen	
2. Klik submenu keyword	
	3. Menampilkan halaman keyword yang berisi : a. NIP b. Nama c. <i>Action</i>
4. Memilih tombol tambah penelitian pada kolom <i>action</i>	
	5. Menampilkan form tambah penelitian
6. Mengisi form tambah penelitian yang berisi: a. Abstrak b. Keyword 1 c. Keyword 2 d. Keyword 3	
7. Klik tombol simpan	
	8. Menampilkan halaman keyword dengan pesan “berhasil tambah data penelitian”

A.18 *Use Case* Skenario Melihat Rekap Bimbingan Dosen

Tabel A.188 *Use Case* Skenario Rekap Bimbingan Dosen

ID	USC-19
Name	Rekap Bimbingan Dosen
Participating Actor	Akademik dan Komisi Bimbingan
Entry Condition	Akademik akan melihat rekap bimbingan dosen
Exit Condition	Akademik telah melihat rekap bimbingan dosen

SKENARIO UTAMA	
“Rekap Bimbingan Dosen ”	
Aksi Aktor	Reaksi Sistem
1. Klik menu dosen	
2. Klik submenu bimbingan	
	3. Menampilkan halaman bimbingan dosen yang berisi : <ul style="list-style-type: none"> a. NIP b. Nama c. Jumlah Bimbingan d. Action
4. Memilih tombol view detail	
	5. Menampilkan detail bimbingan dosen

A.19 Use Case Skenario Melihat Mahasiswa Bimbingan Dosen

Tabel A.199 Use Case Skenario Melihat Mahasiswa Bimbingan Dosen

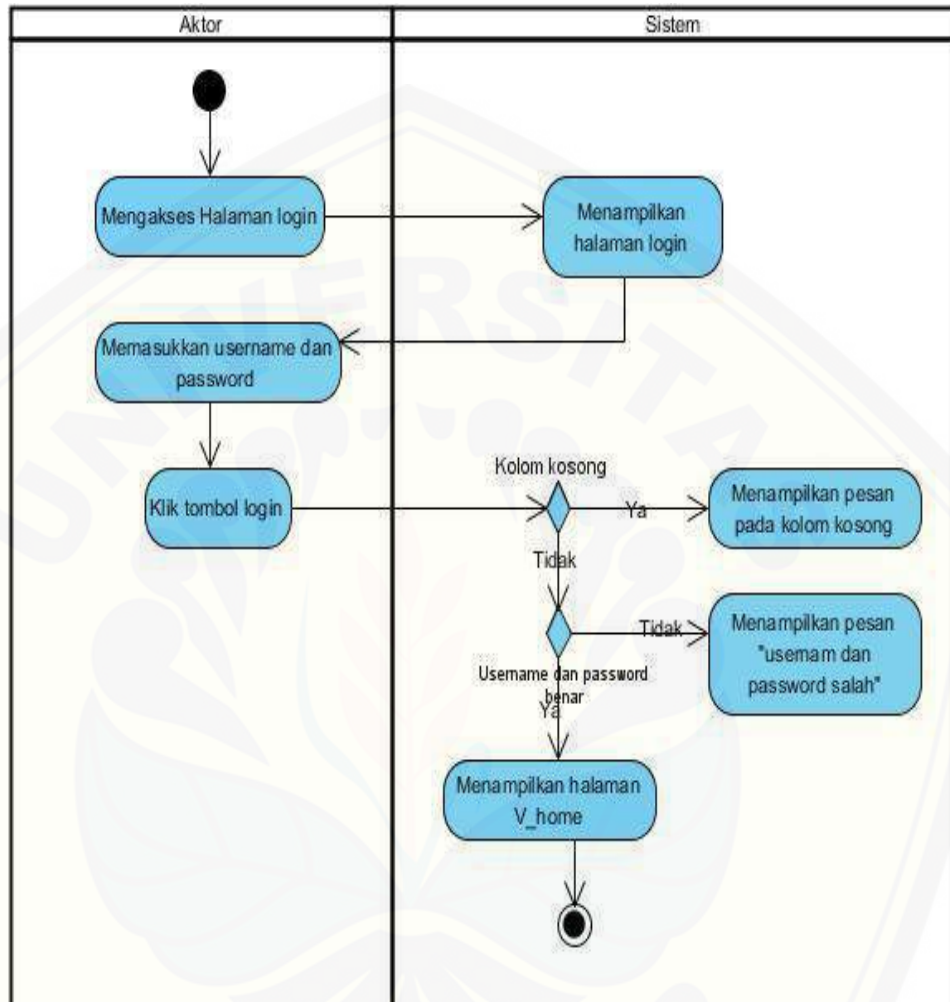
ID	USC-20
Name	Melihat Mahasiswa Bimbingan Dosen
Participating Actor	Dosen
Entry Condition	Dosen akan melihat mahasiswa bimbingan tugas akhir
Exit Condition	Dosen telah melihat mahasiswa bimbingan tugas akhir

SKENARIO UTAMA	
“Melihat Mahasiswa Bimbingan Dosen”	
Aksi Aktor	Reaksi Sistem
1. Klik menu dosen	
2. Klik submenu bimbingan	
	3. Menampilkan halaman mahasiswa bimbingan dosen

A.20 Use Case Skenario Logout

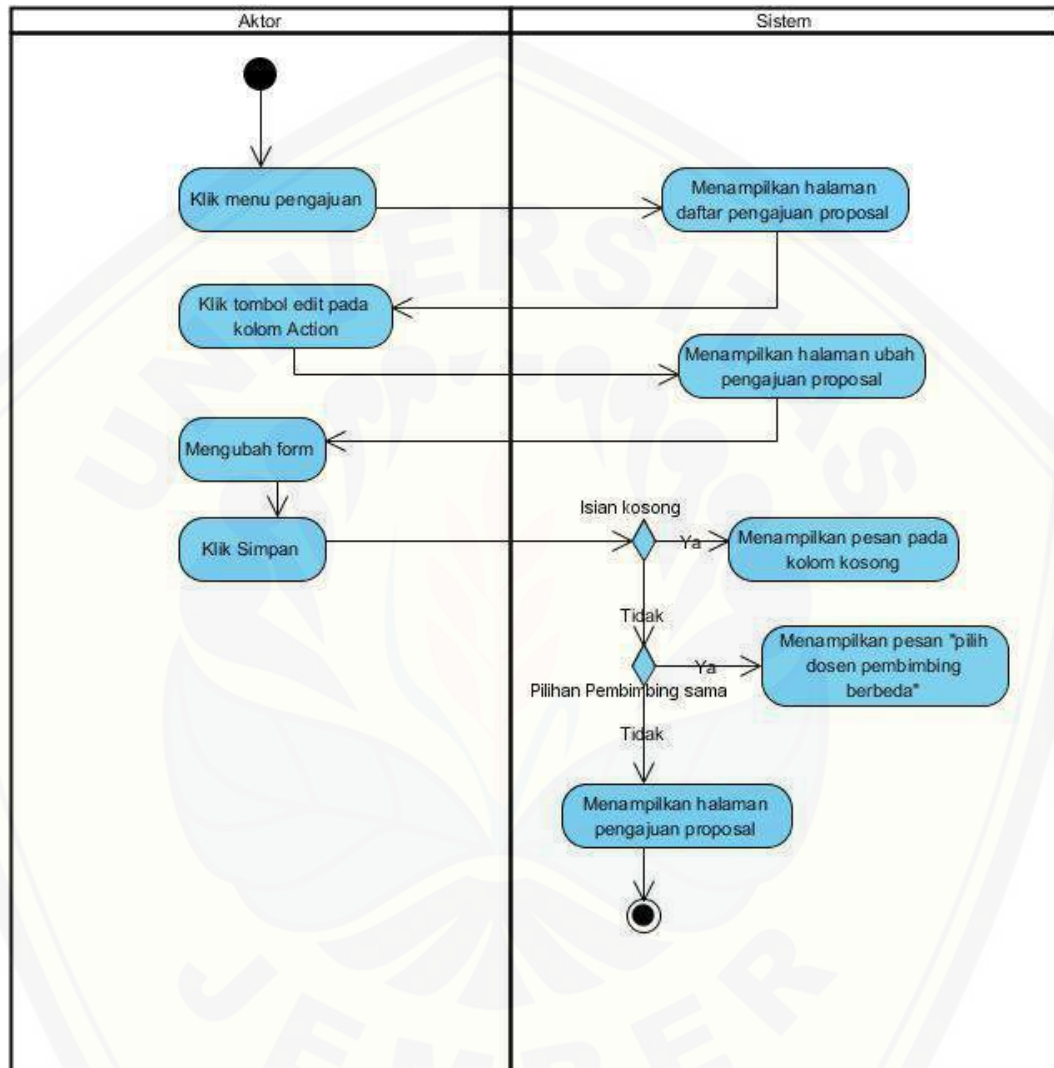
Tabel A.20 Use case skenario Logout

ID	USC-21
Name	<i>Logout</i>
Participating Actor	Semua Aktor
Entry Condition	Aktor akan keluar dari system
Exit Condition	Aktor telah keluar dari system
SKENARIO UTAMA	
“Logout”	
Aksi Aktor	Reaksi Sistem
1. Klik menu <i>logout</i>	2. Menampilkan halaman tampilan awal

Lampiran B. Activity Diagram**B.1 Activity Diagram Login**

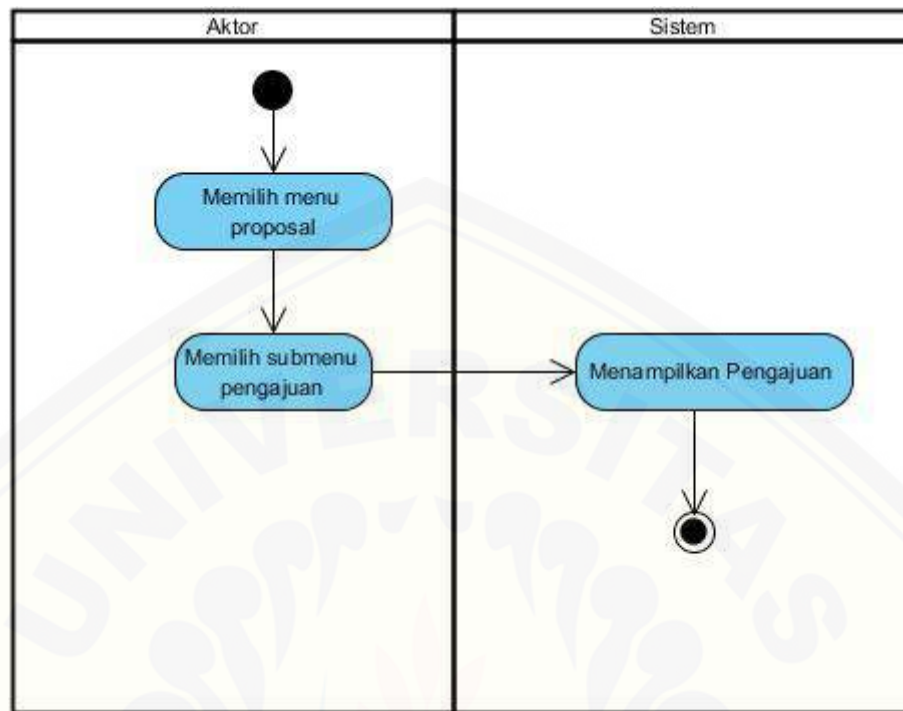
Gambar B.1 Activity diagram login

B.2 Activity Diagram Mengubah Pengajuan Proposal



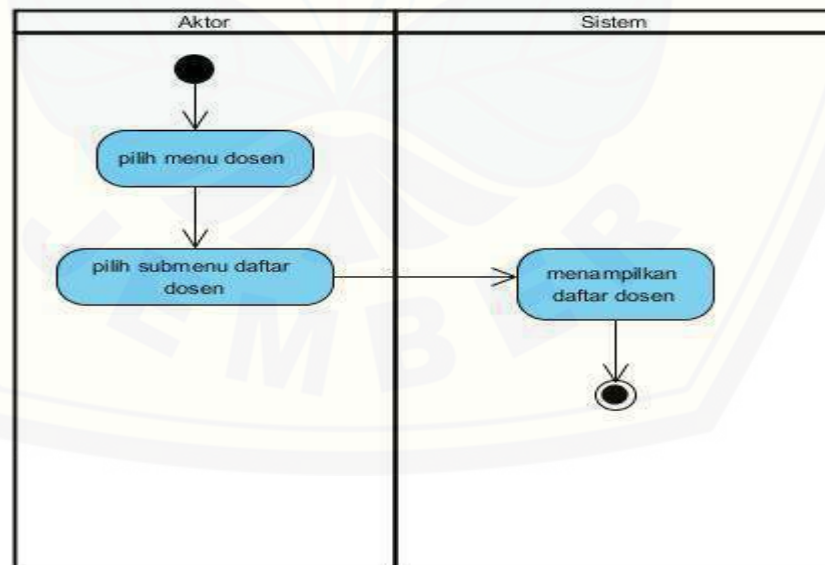
Gambar B.2 Activity Mengubah pengajuan Proposal

B.3 Activity Diagram Melihat Daftar Pengajuan Proposal



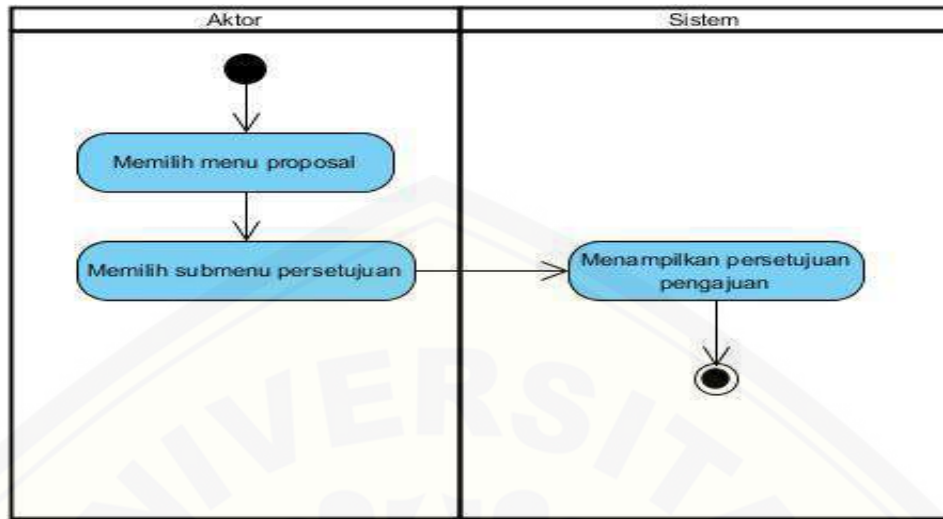
Gambar B.3 Activity diagram melihat daftar pengajuan proposal

B.4 Activity Diagram Melihat Data Dosen



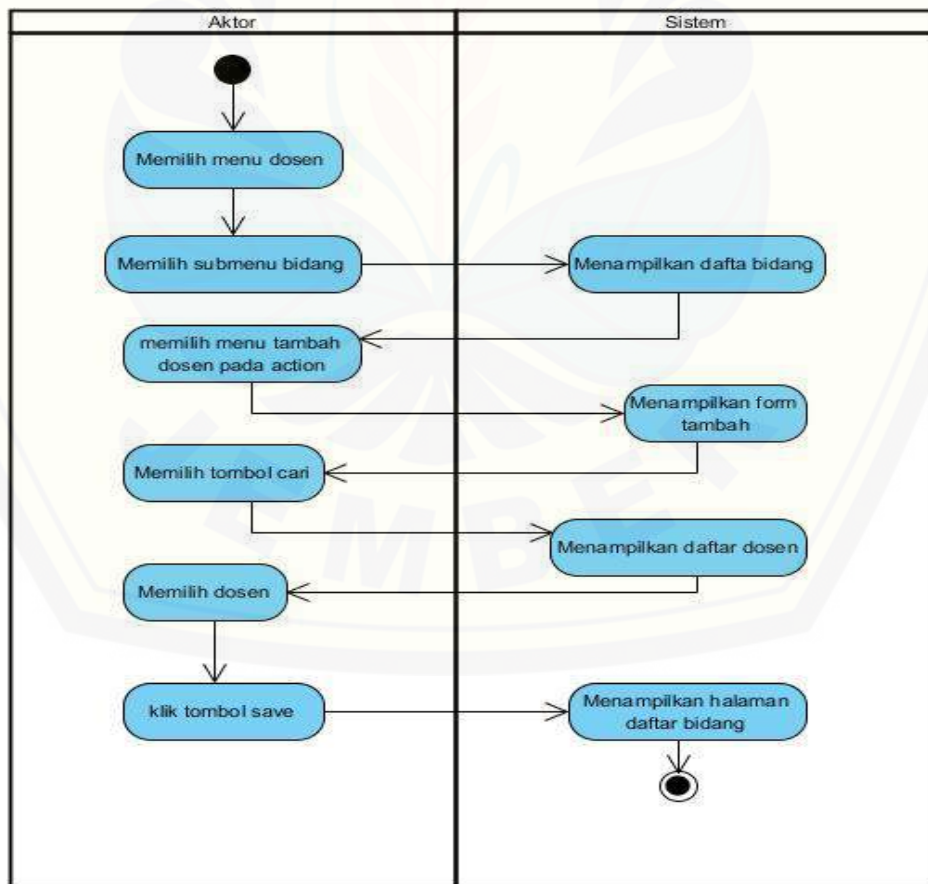
Gambar B.4 Activity diagram melihat data dosen

B.5 Activity Diagram Melihat Persetujuan Pengajuan



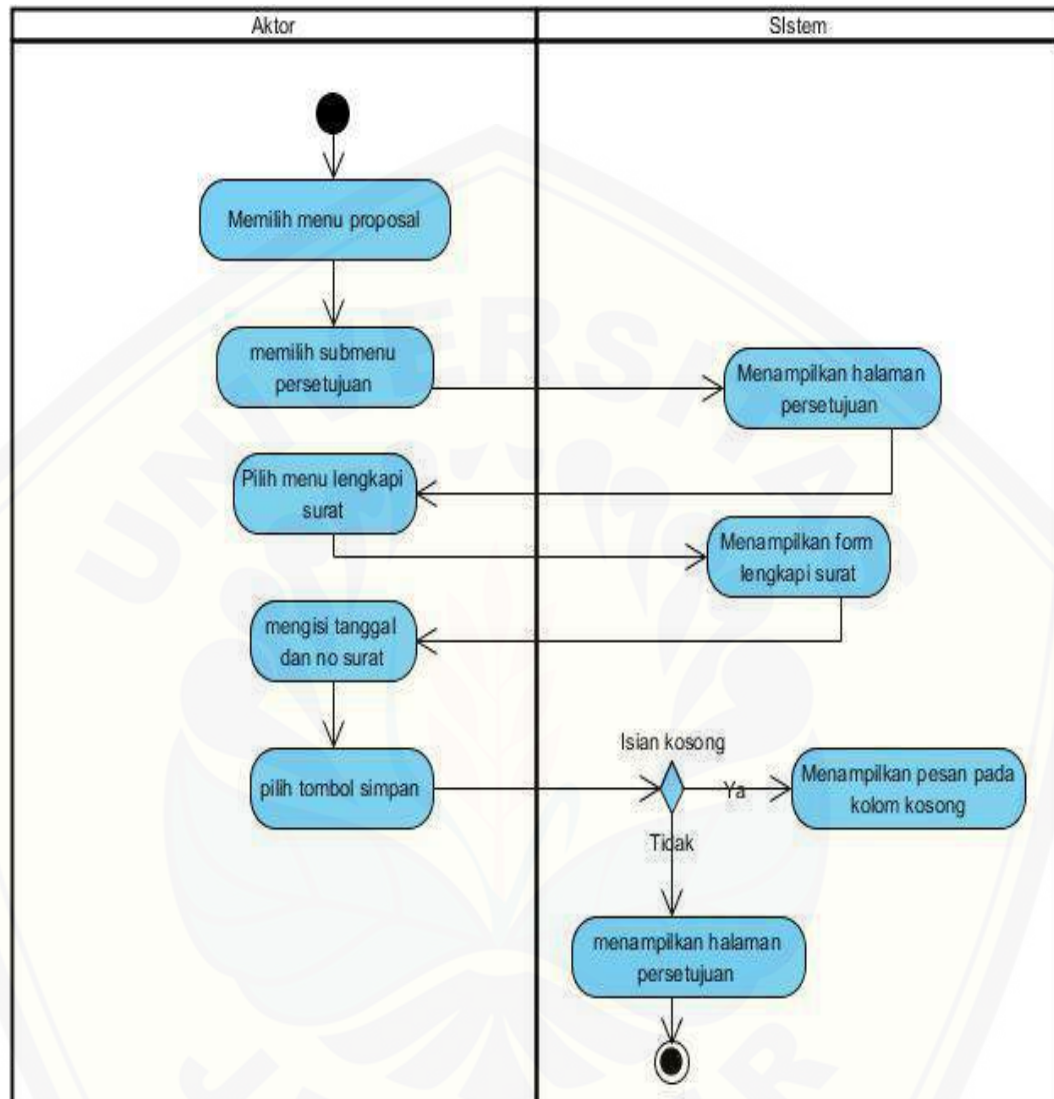
Gambar B.5 Activity diagram persetujuan pengajuan

B.6 Activity Diagram Menambah Bidang Keahlian Dosen



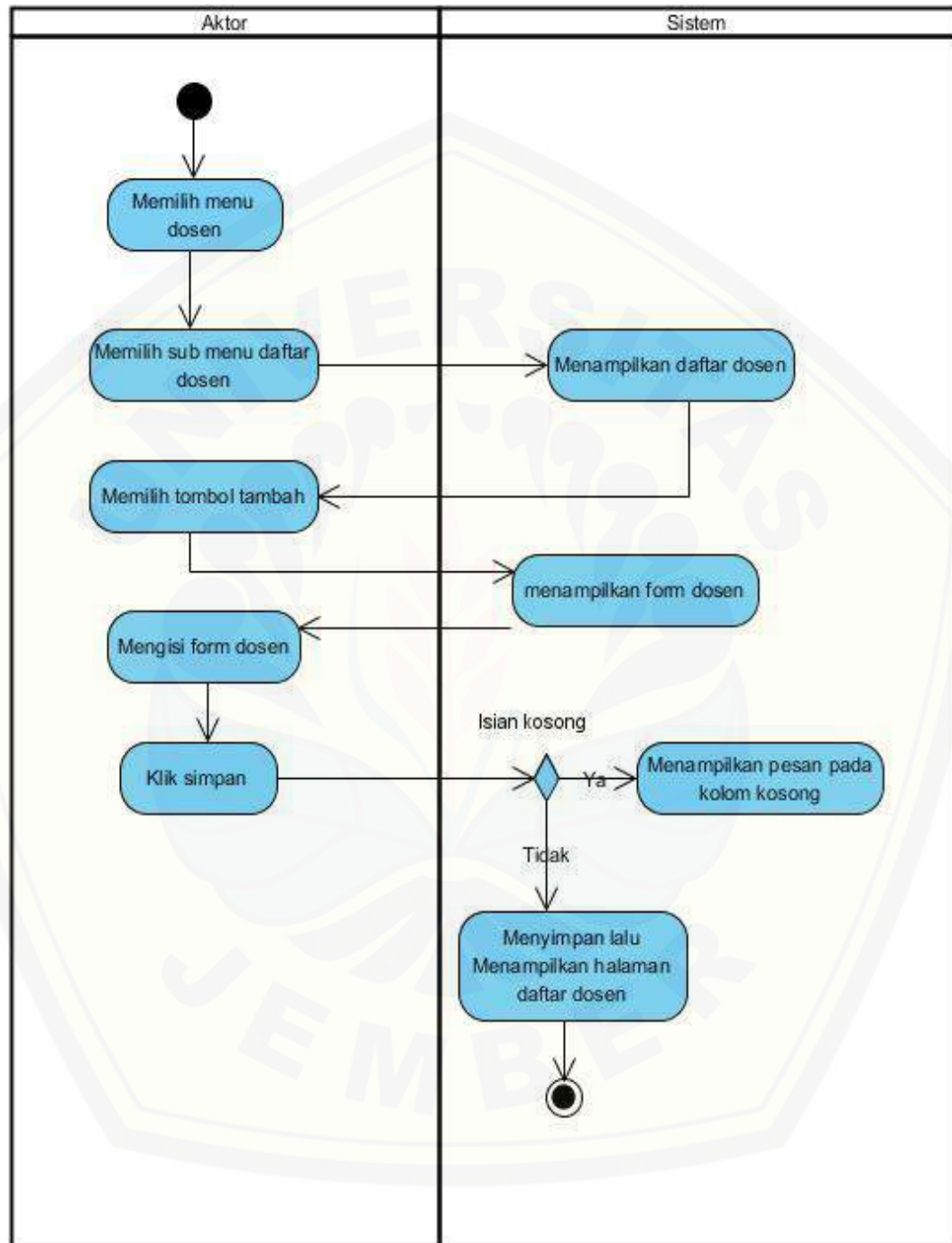
Gambar B.6 Activity diagram persetujuan pengajuan

B.7 Activity Diagram Melengkapi Cetak Surat Tugas



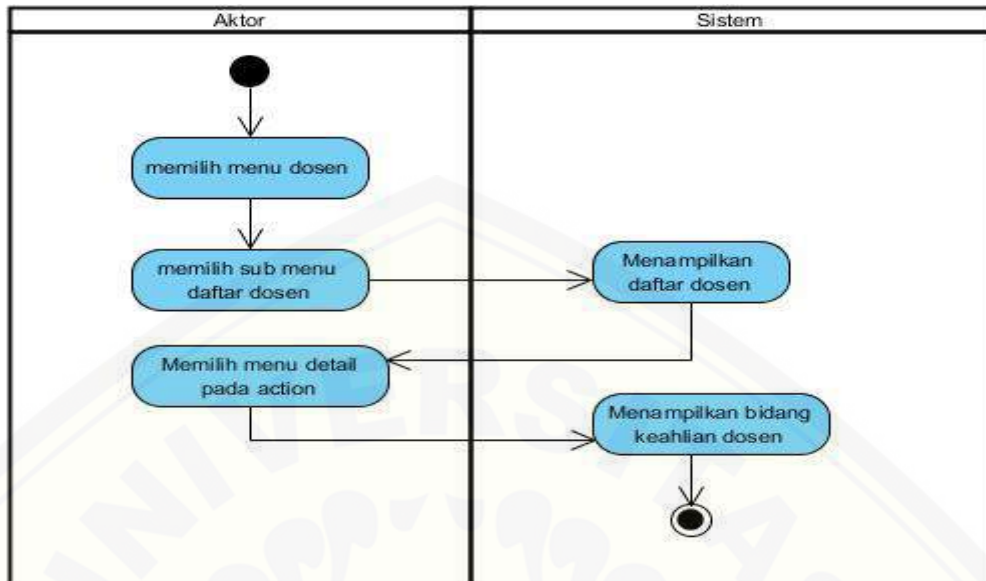
Gambar B.6 Activity melengkapi cetak surat tugas

B.8 Activity Diagram Menambah Data Dosen



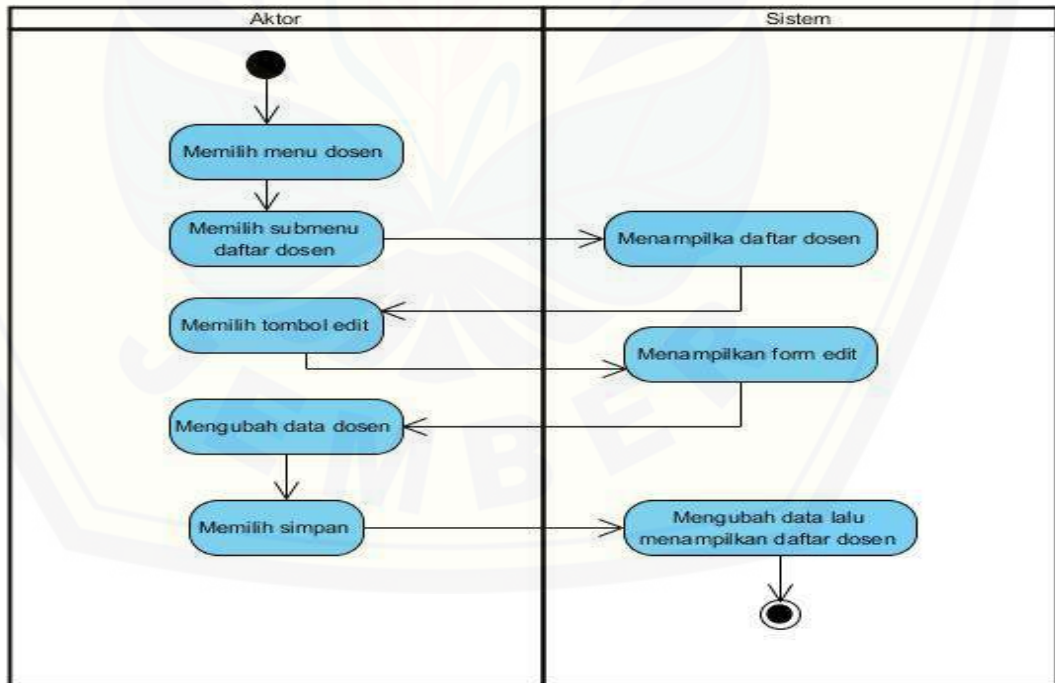
Gambar B.8 Activity diagram menambah data dosen

B.9 Activity Diagram Melihat Bidang Keahlian Dosen



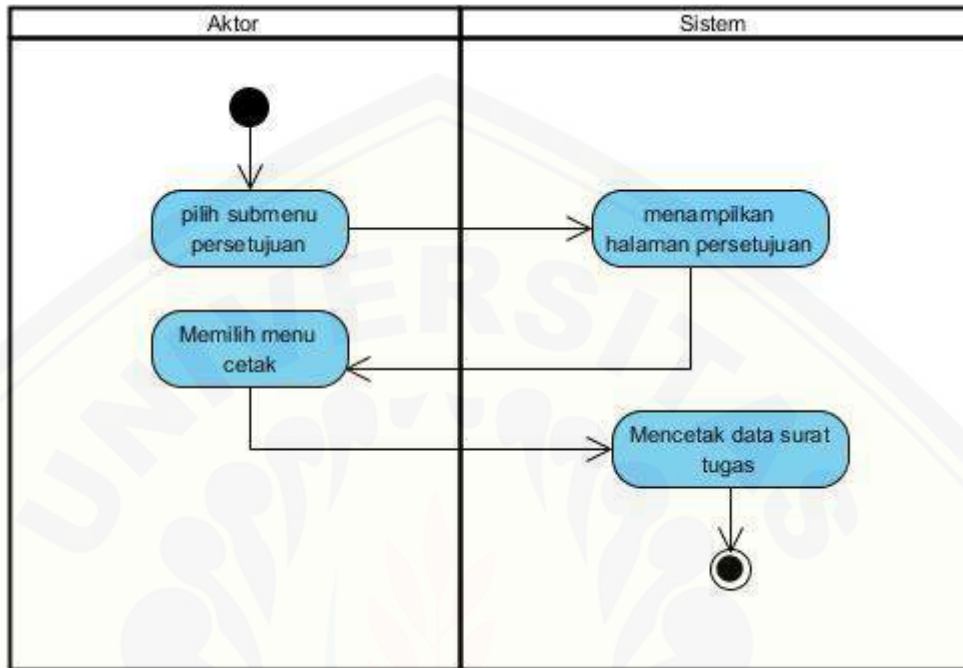
Gambar B.9 Activity diagram melihat bidang keahlian dosen

B.10 Activity Diagram Mengubah Data Dosen



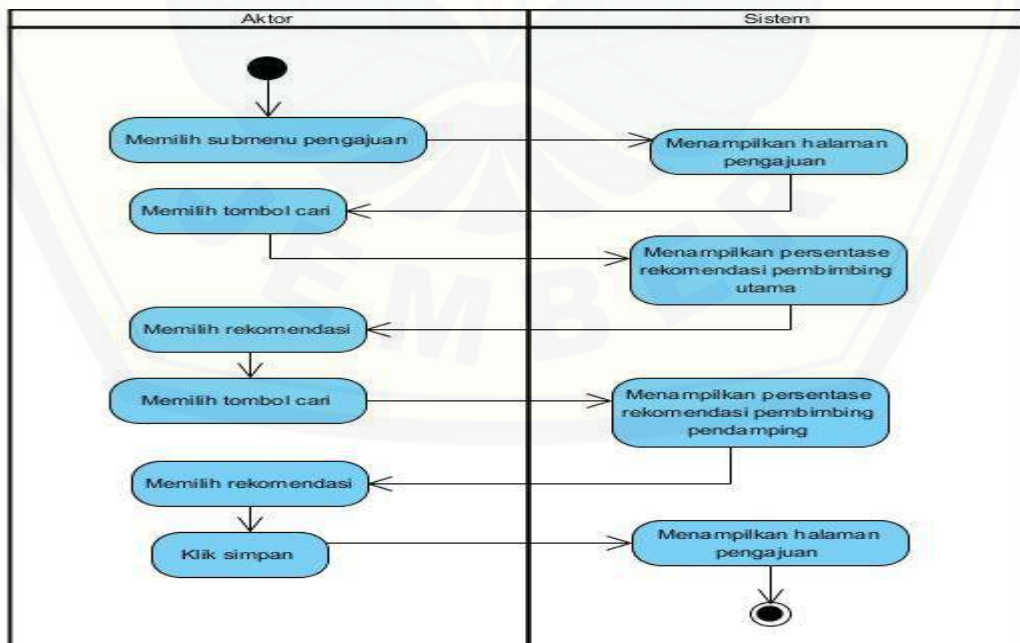
Gambar B.10 Activity diagram mengubah data dosen

B.11 Activity Diagram Mencetak Surat Tugas



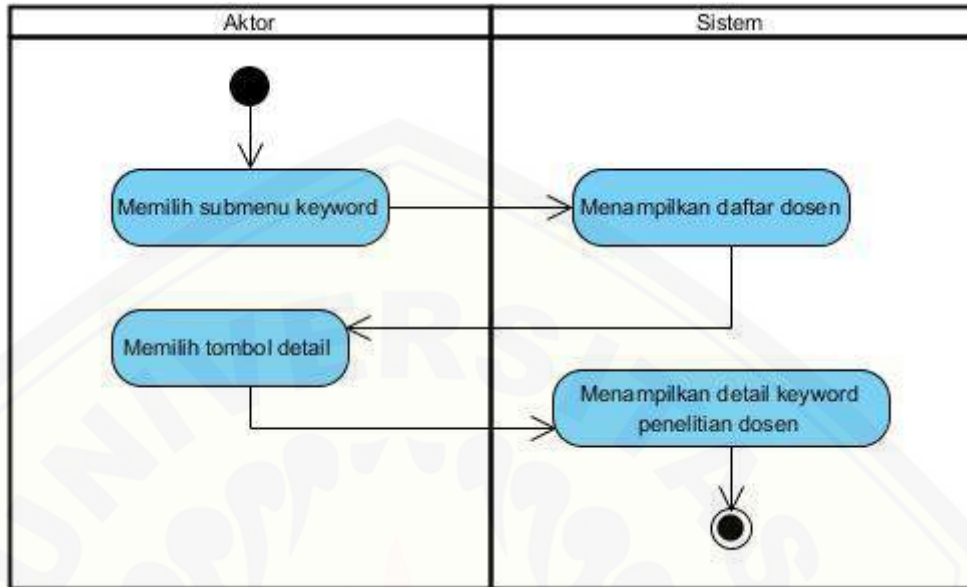
Gambar B.11 Activity diagram mencetak surat tugas

B.12 Activity Diagram Manajemen Rekomendasi Dosen



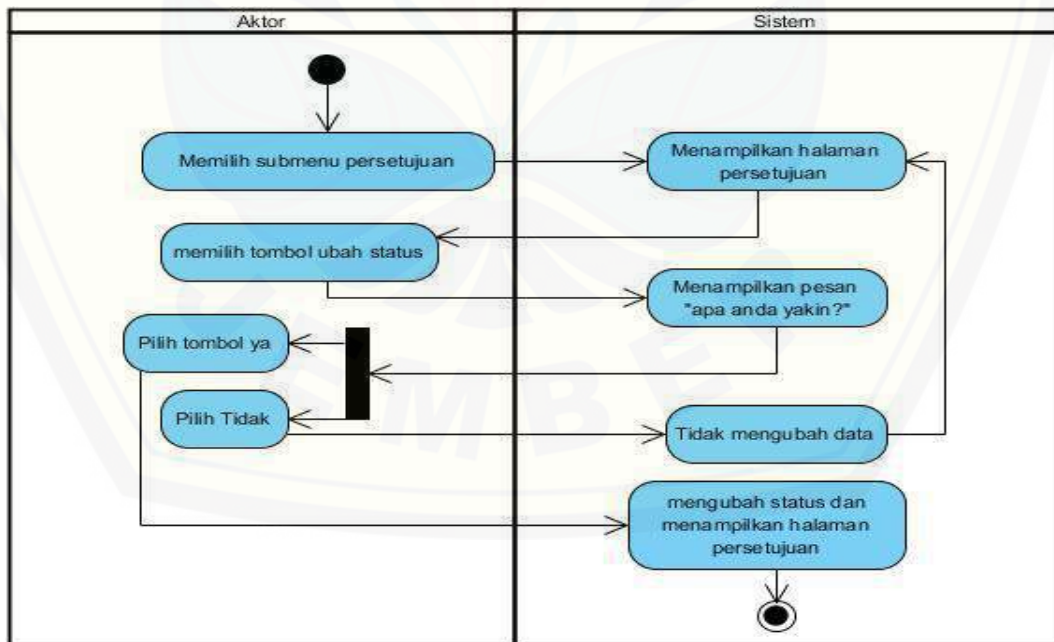
Gambar B.12 Activity Diagram Manajemen Rekomendasi Dosen

B.13 Activity Diagram Melihat Keyword Penelitian Dosen



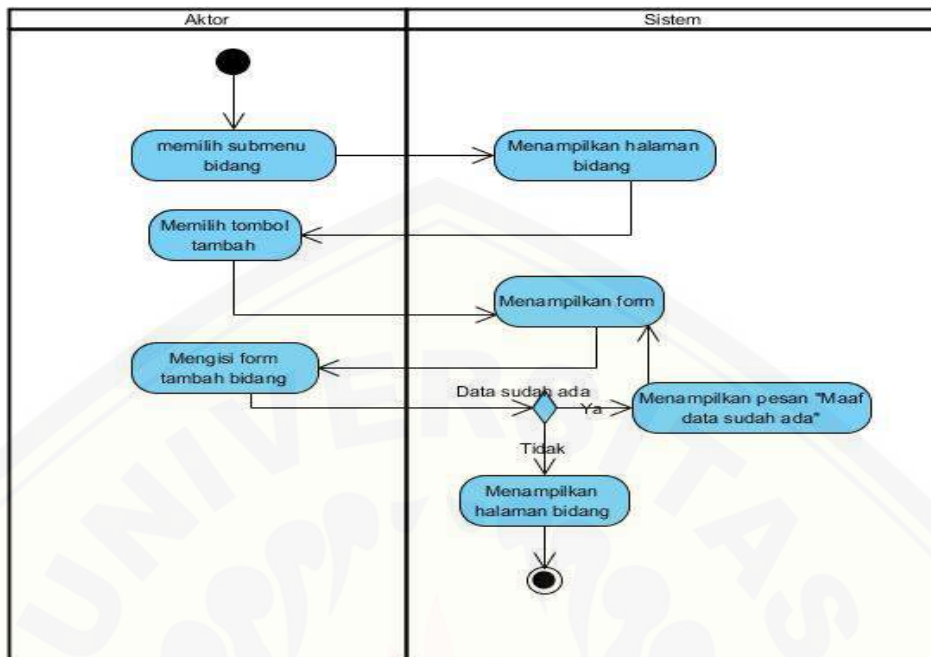
Gambar B.13 Activity Diagram Melihat Keyword Penelitian Dosen

B.14 Activity Diagram Mengubah Status Lulus Mahasiswa



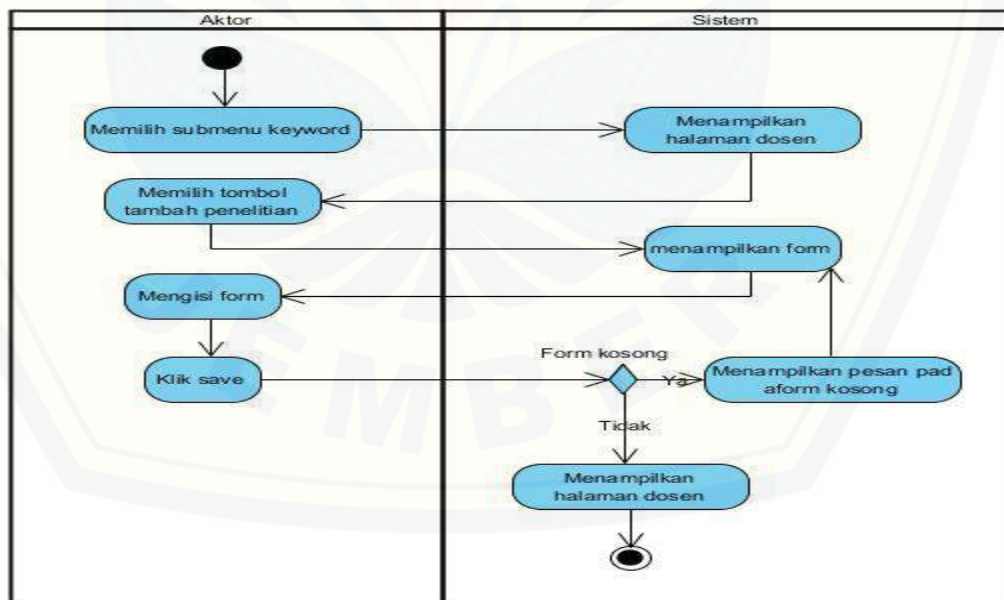
Gambar B.14 Activity diagram mengubah status lulus mahasiswa

B.15 Activity Diagram Menambah Bidang Dosen



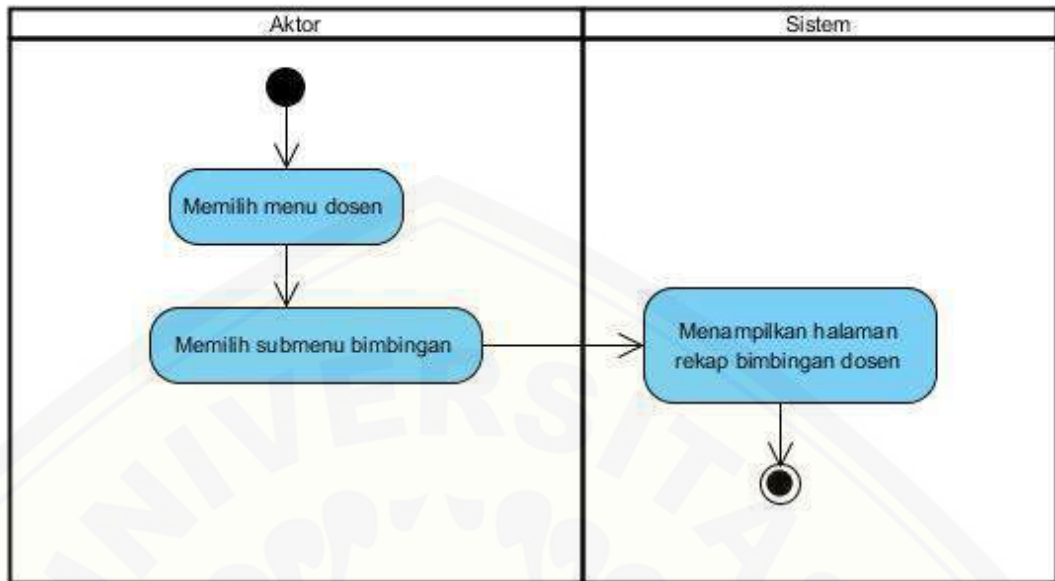
Gambar B.15 Activity Diagram Menambah Bidang Dosen

B.16 Activity Diagram Menambah Keyword Penelitian Dosen



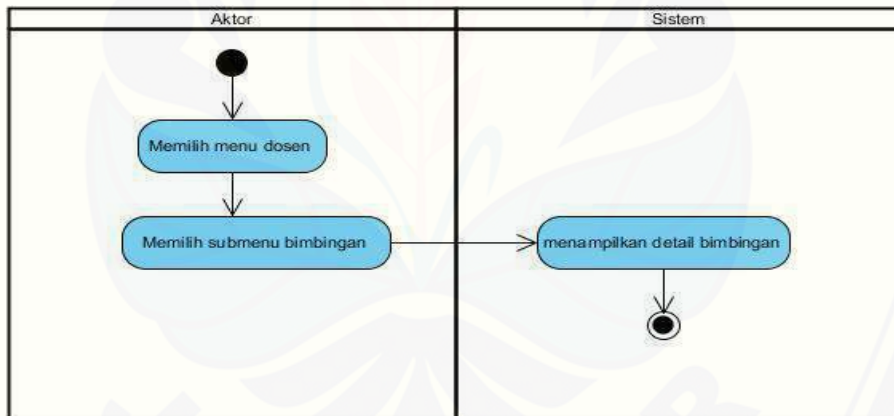
Gambar B.16 Activity diagram menambah keyword dosen

B.17 Activity Diagram Melihat Rekap Mahasiswa Bimbingan Dosen



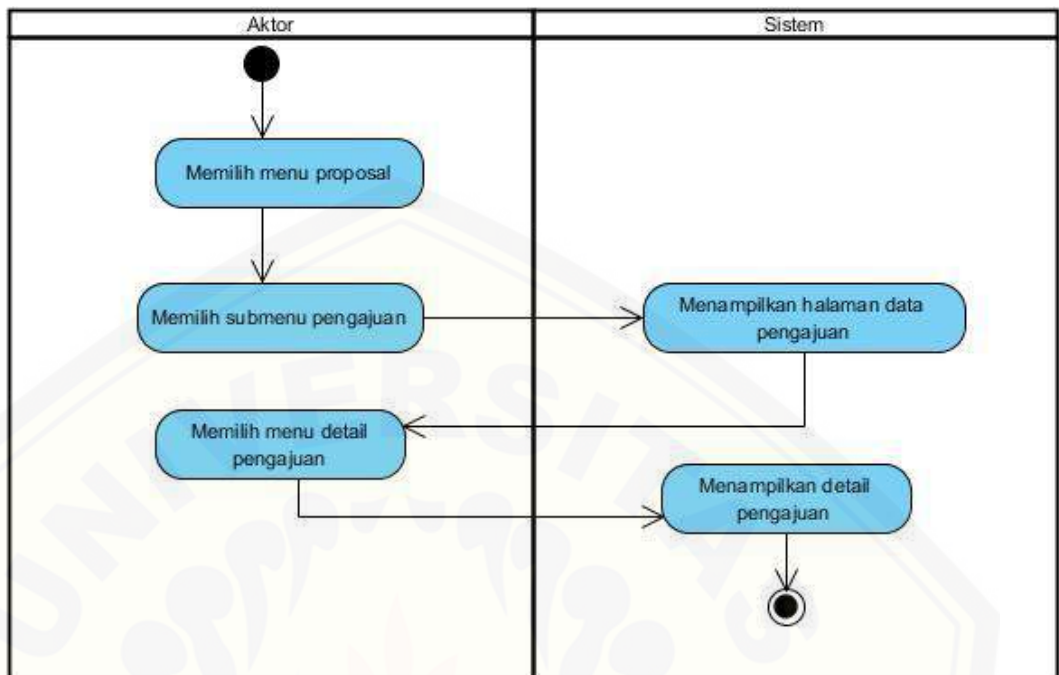
Gambar B.17 Activity melihat rekap mahasiswa bimbingan dosen

B.18 Activity Diagram melihat mahasiswa bimbingan dosen



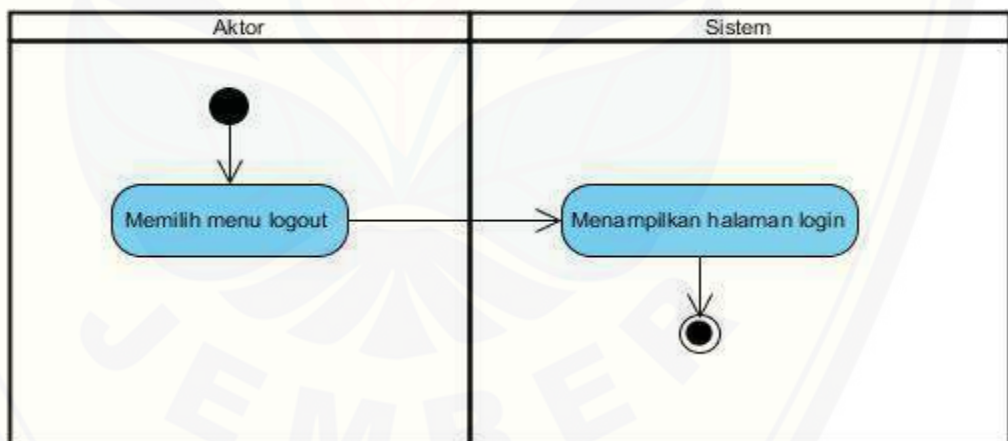
Gambar B.18 Activity melihat rekap mahasiswa bimbingan dosen

B.19 Activity Diagram Melihat Detail Pengajuan



Gambar B.19 Activity diagram melihat detail pengajuan

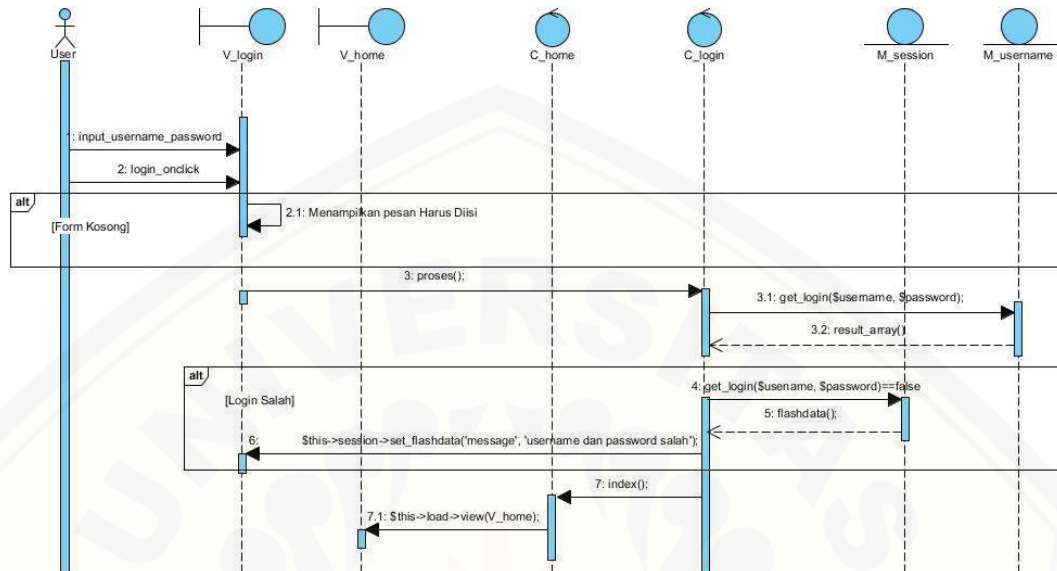
B.20 Activity Logout



Gambar B.20 Activity diagram melihat detail pengajuan

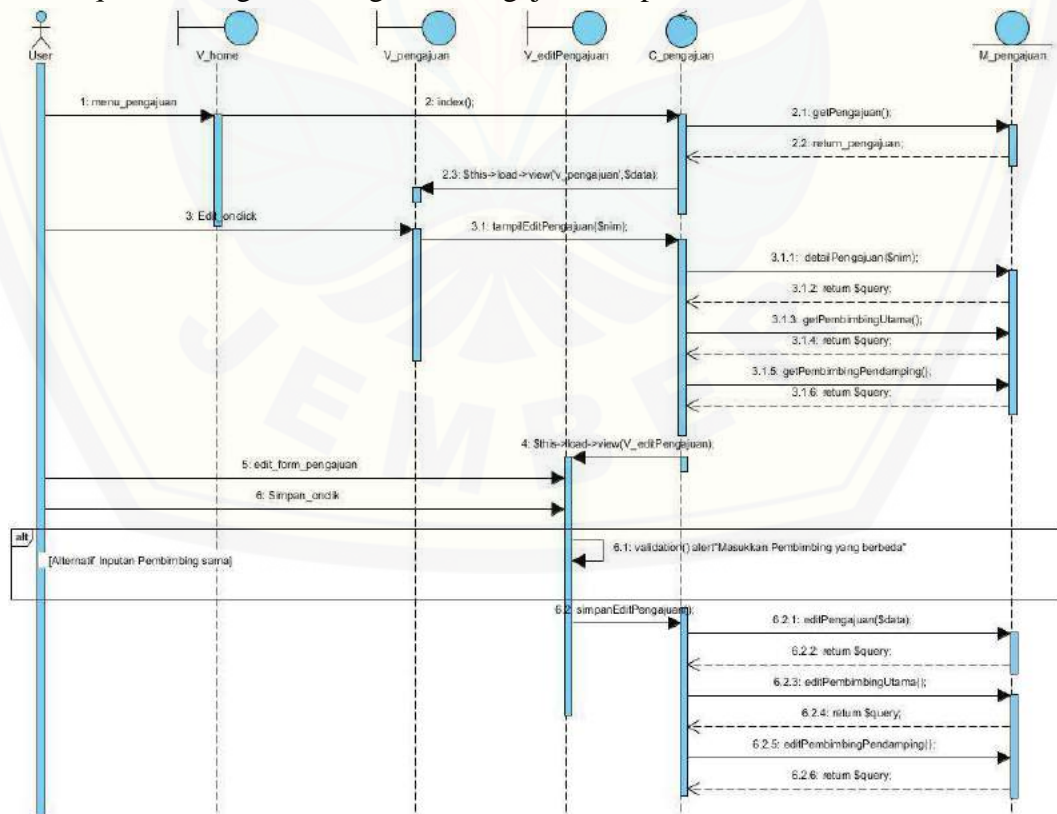
Lampiran C. Sequence Diagram

C.1 Sequence Diagram Login



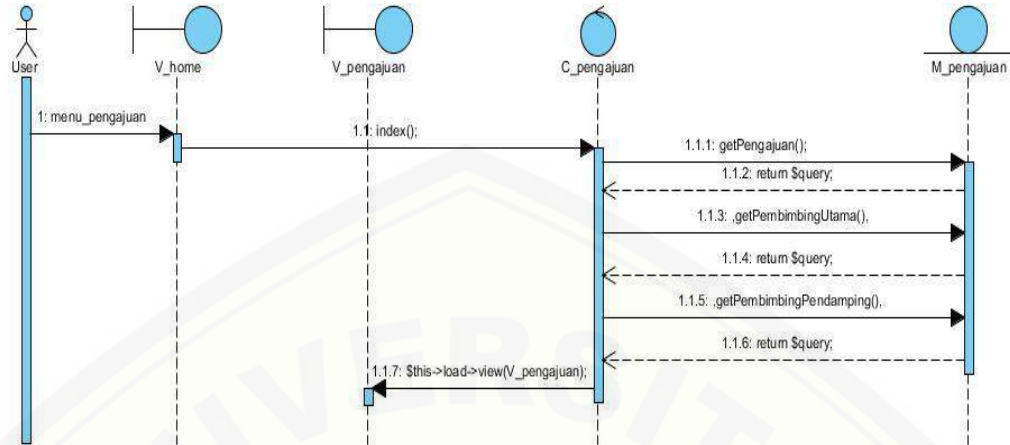
Gambar C.1 Sequence diagram login

C.2 Sequence Diagram Mengubah Pengajuan Proposal



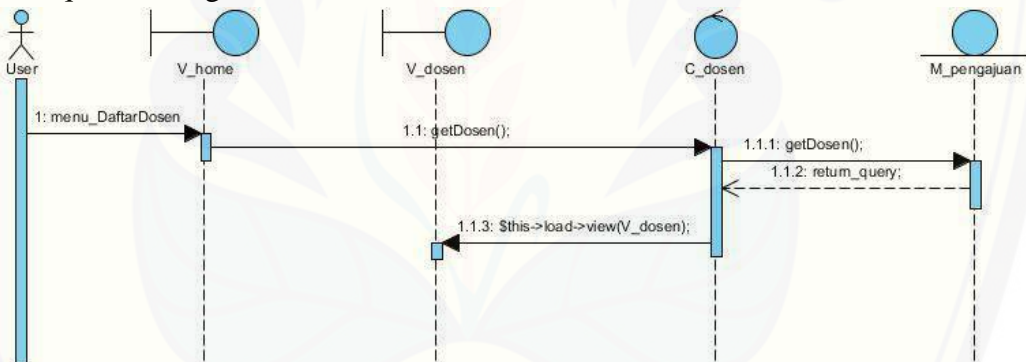
Gambar C.2 Sequence diagram mengubah pengajuan proposal

C.3 Sequence Diagram Melihat Daftar Pengajuan Proposal



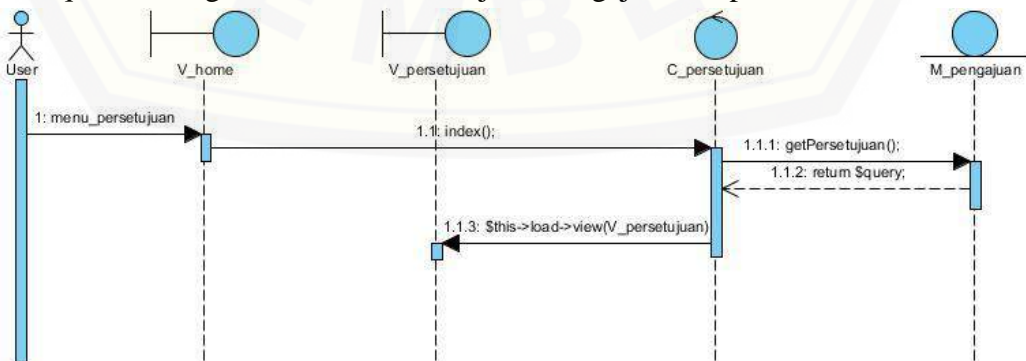
Gambar C.3 Sequence diagram melihat daftar pengajuan proposal

C.4 Sequence Diagram Melihat Data Dosen



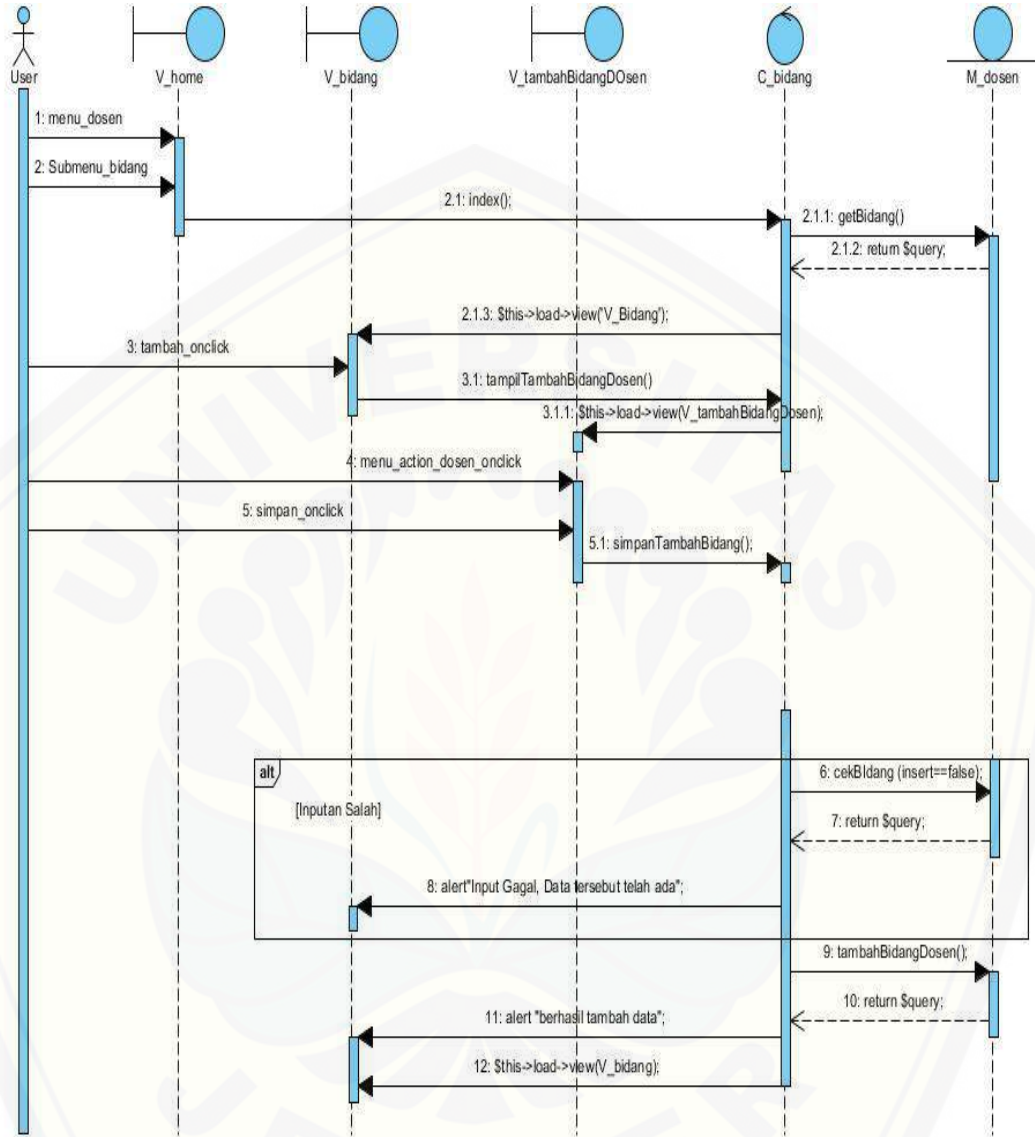
Gambar C.4 Sequence diagram melihat data dosen

C.5 Sequence Diagram Melihat Persetujuan Pengajuan Proposal



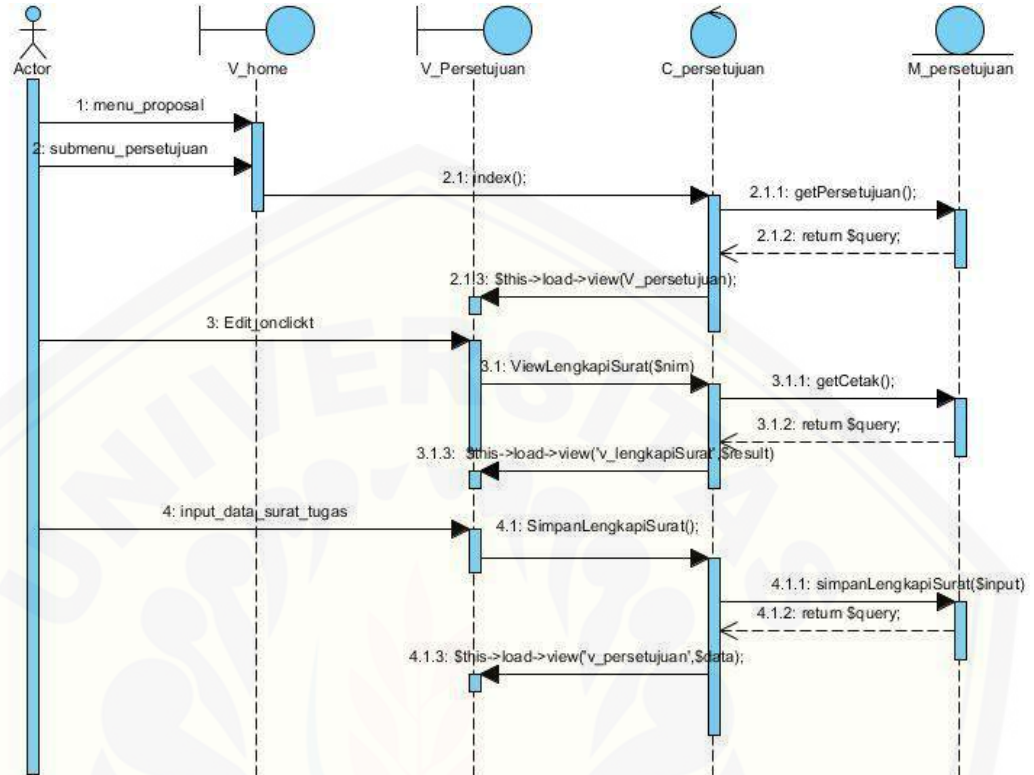
Gambar C.5 Sequence diagram melihat persetujuan pengajuan proposal

C.6 Sequence Diagram Menambah Bidang Keahlian Dosen



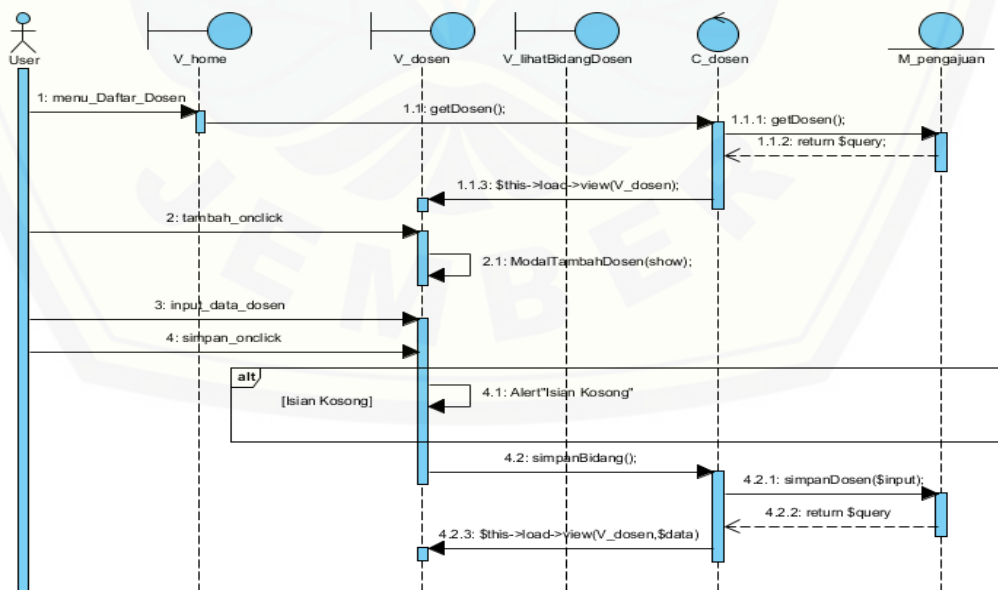
Gambar C.6 sequence diagram menambah bidang keahlian dosen

C.7 Sequence Diagram Melengkapi Cetak Surat Tugas



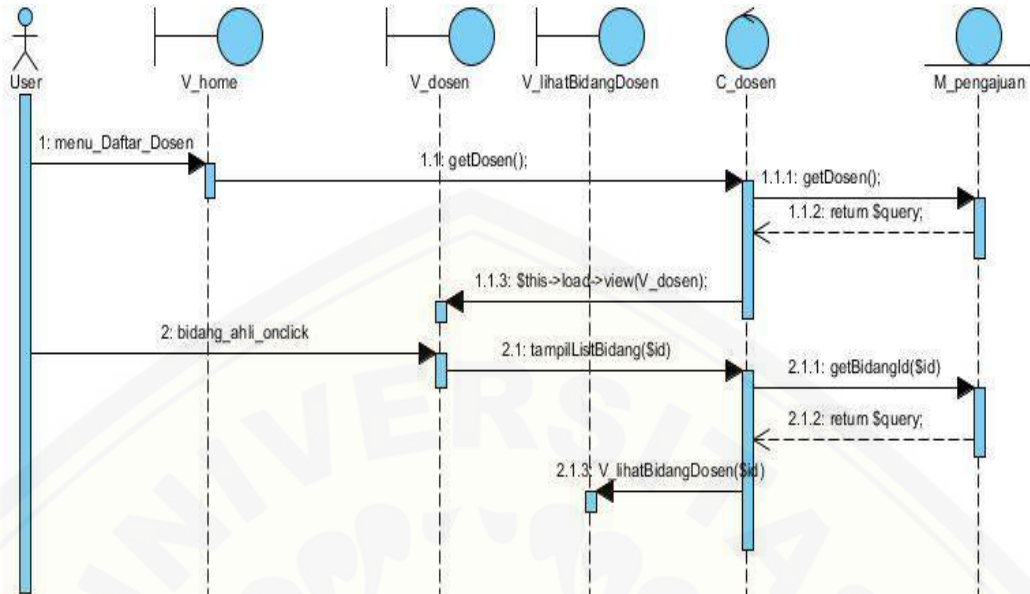
Gambar C.7 Sequence diagram melengkapi cetak surat tugas

C.8 Sequence Diagram Menambah Data Dosen



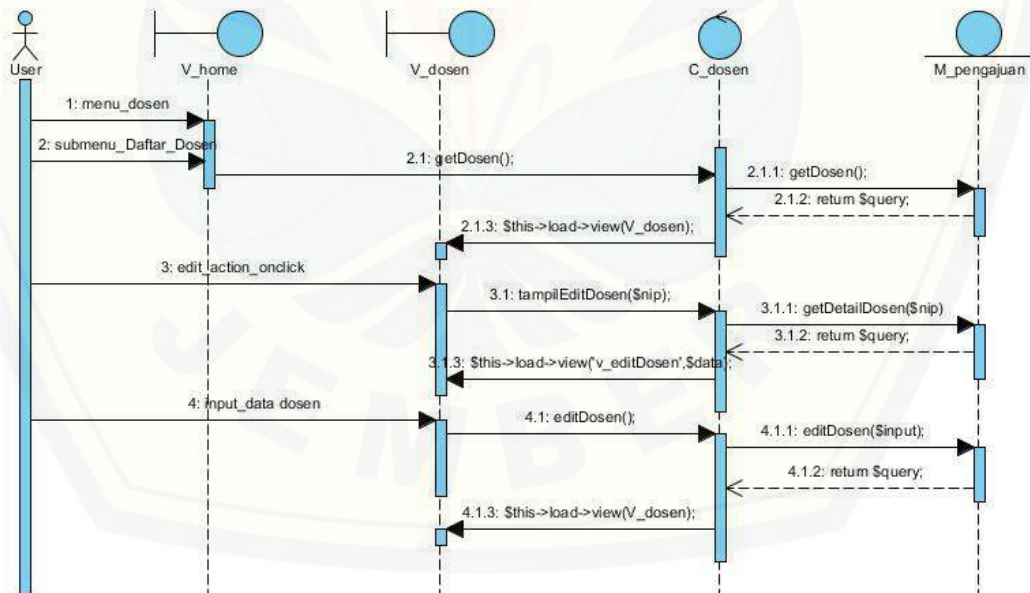
Gambar C.8 Sequence diagram menambah data dosen

C.9 Sequence Diagram Melihat Bidang Keahlian



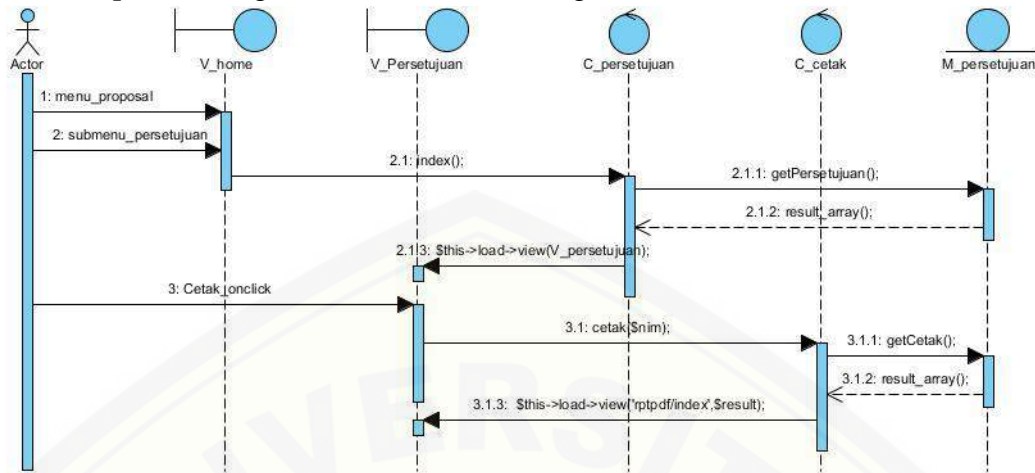
Gambar C.9 Sequence diagram melihat bidang keahlian

C.10 Sequence Diagram Mengubah Data Dosen



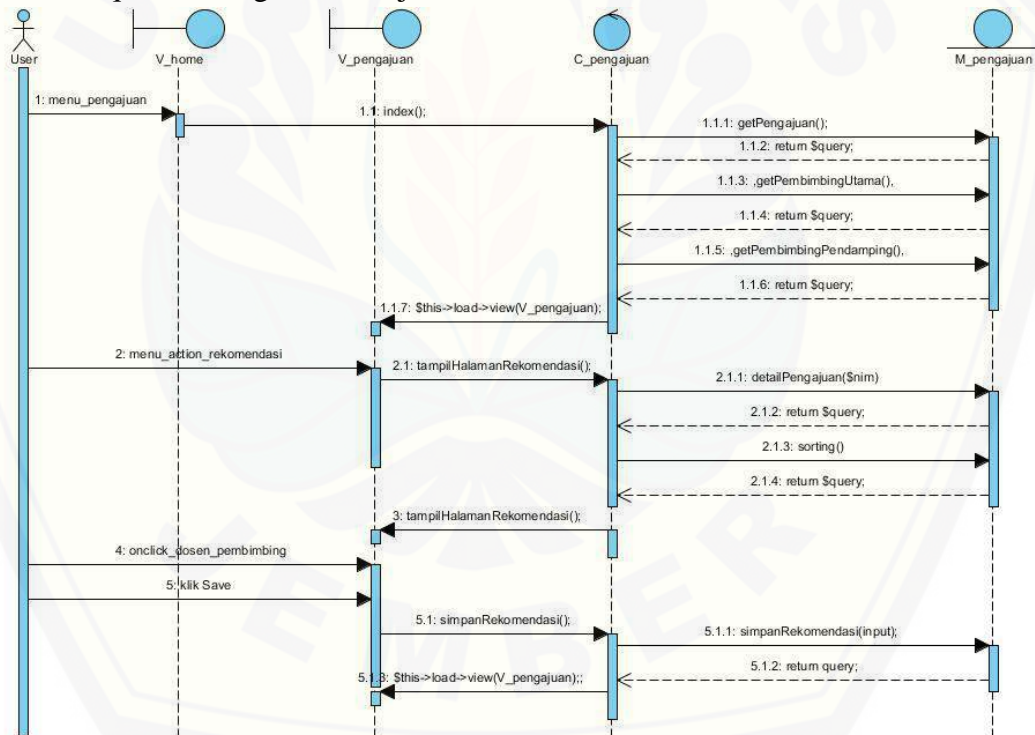
Gambar C.10 Sequence diagram mengubah data dosen

C.11 Sequence Diagram Mencetak Surat Tugas



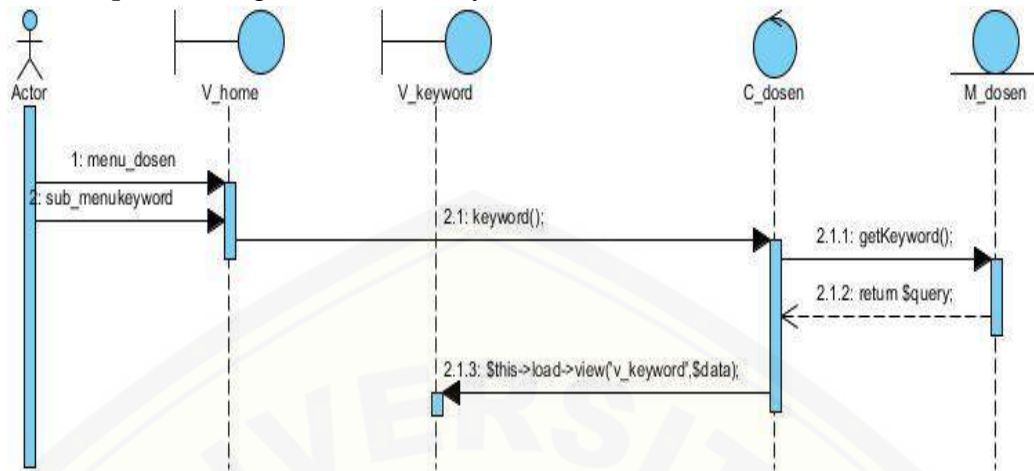
Gambar C.11 Sequence diagram Mencetak Surat Tugas

C.12 Sequence Diagram Manajemen Rekomendasi Dosen



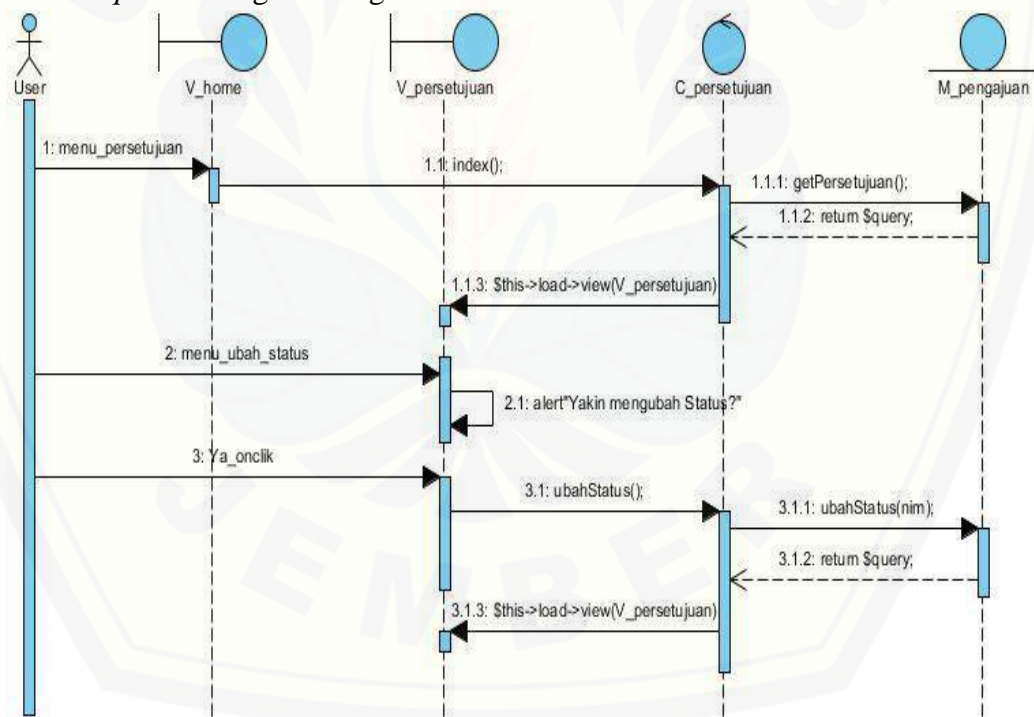
Gambar C.12 Sequence diagram manajemen rekomendasi dosen

C.13 Sequence Diagram Melihat Keyword Penelitian Dosen



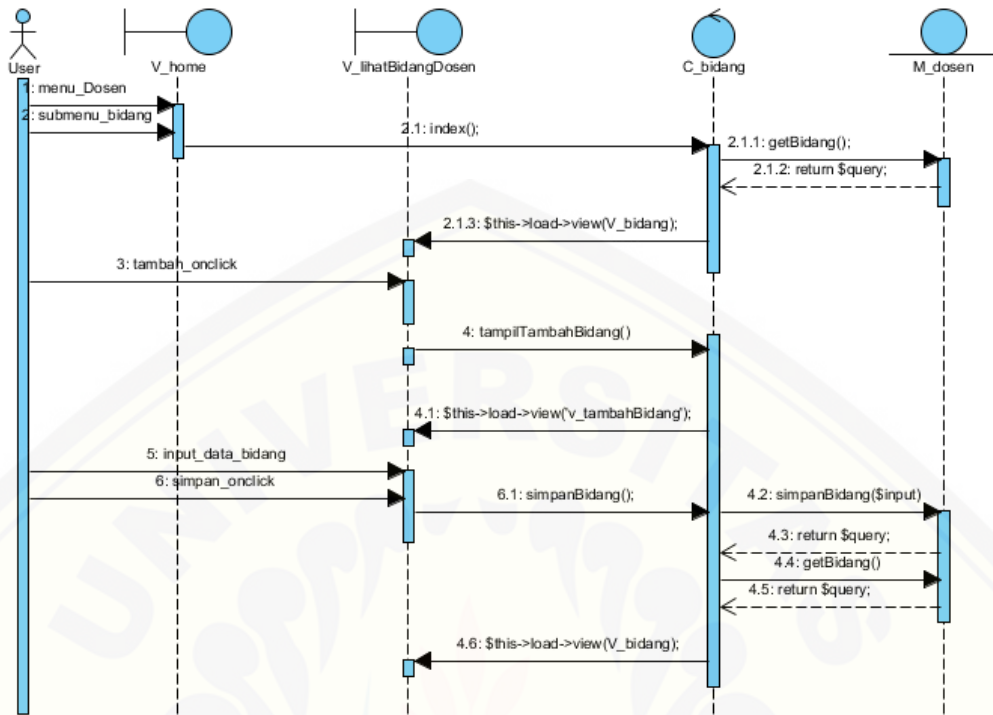
Gambar C.13 Sequence diagram melihat keyword penelitian dosen

C.14 Sequence Diagram Mengubah Status Lulus



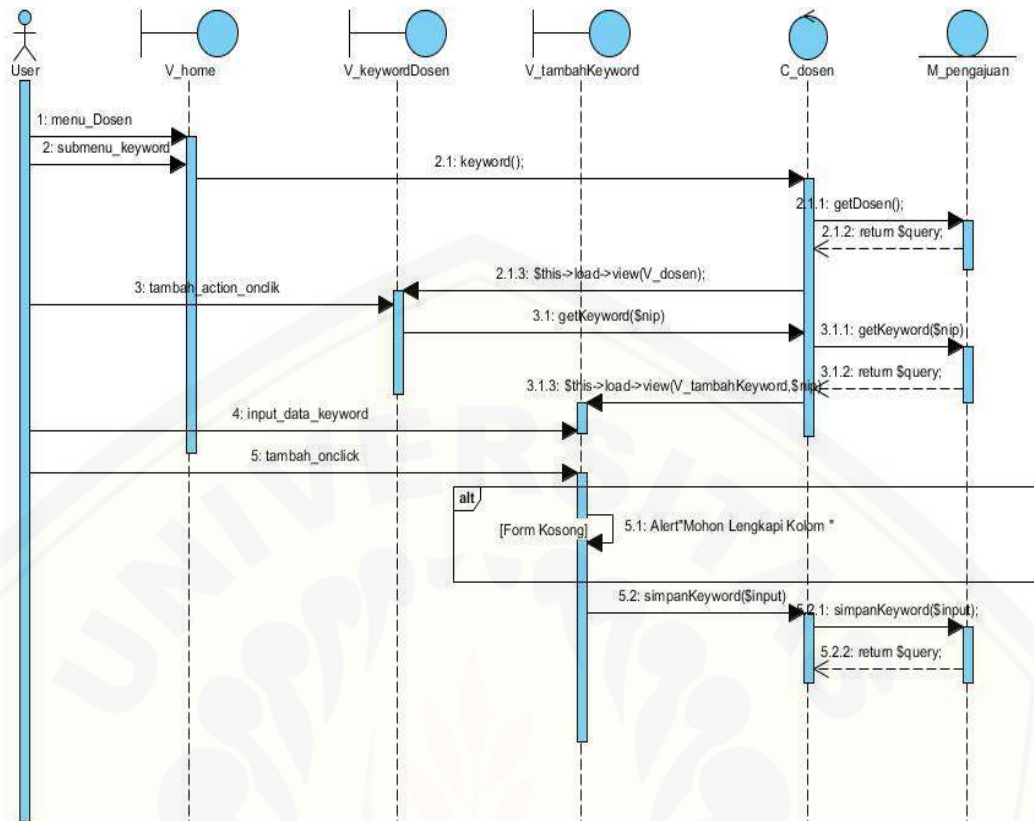
Gambar C.14 Sequence diagram mengubah status lulus

C.15 Sequence Diagram Menambah Bidang



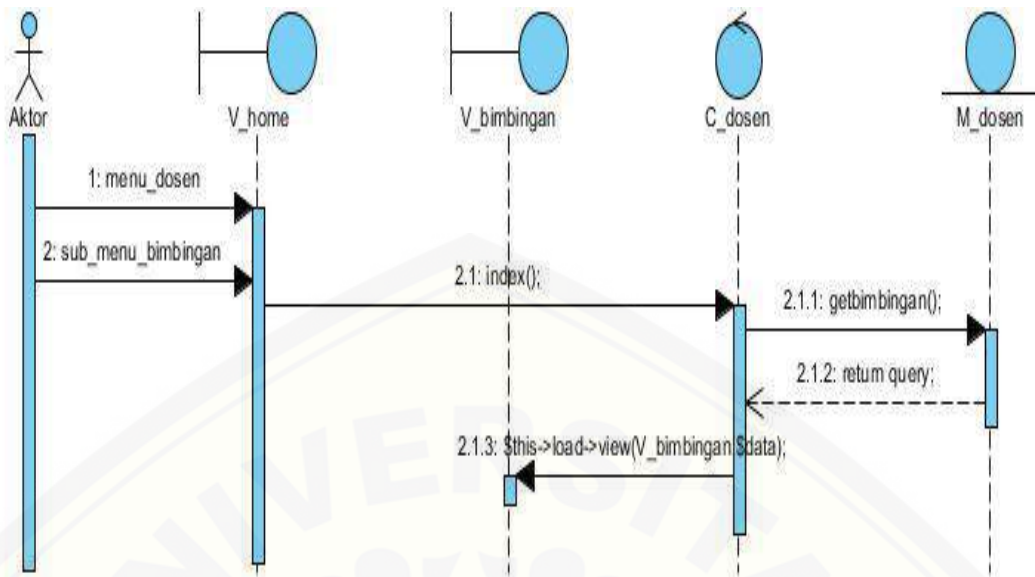
Gambar C.15 sequence diagram menambah bidang

C.16 Sequence Diagram Menambah Keyword Penelitian



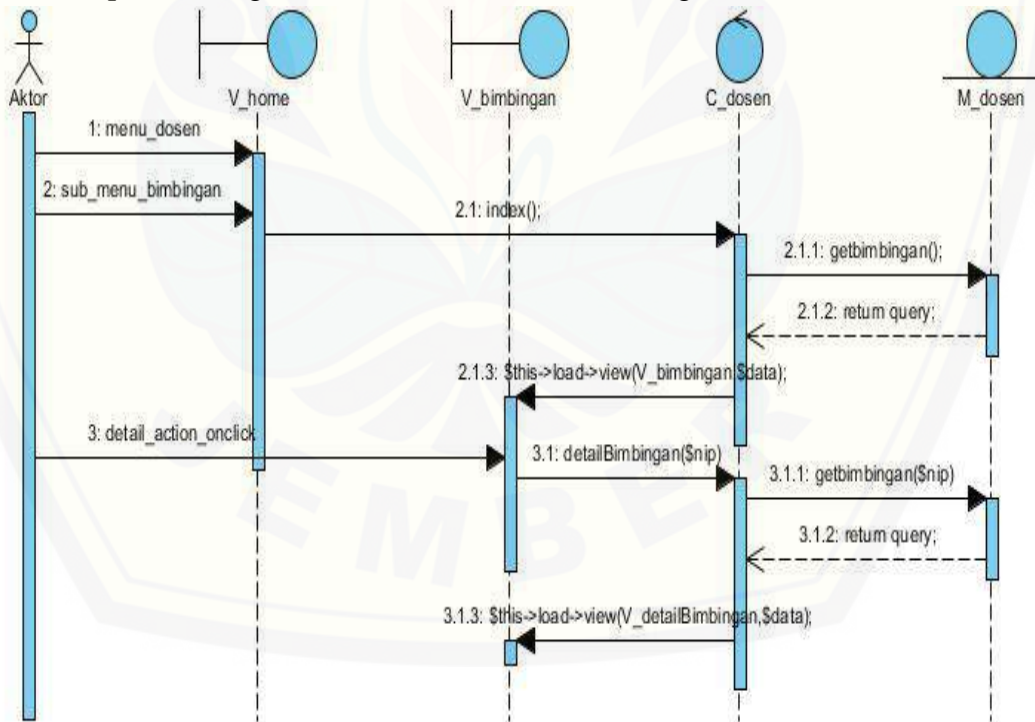
Gambar C.16 *sequence* diagram menambah keyword penelitian

C.17 *Sequence* Diagram Melihat Rekap Mahasiswa Bimbingan Dosen



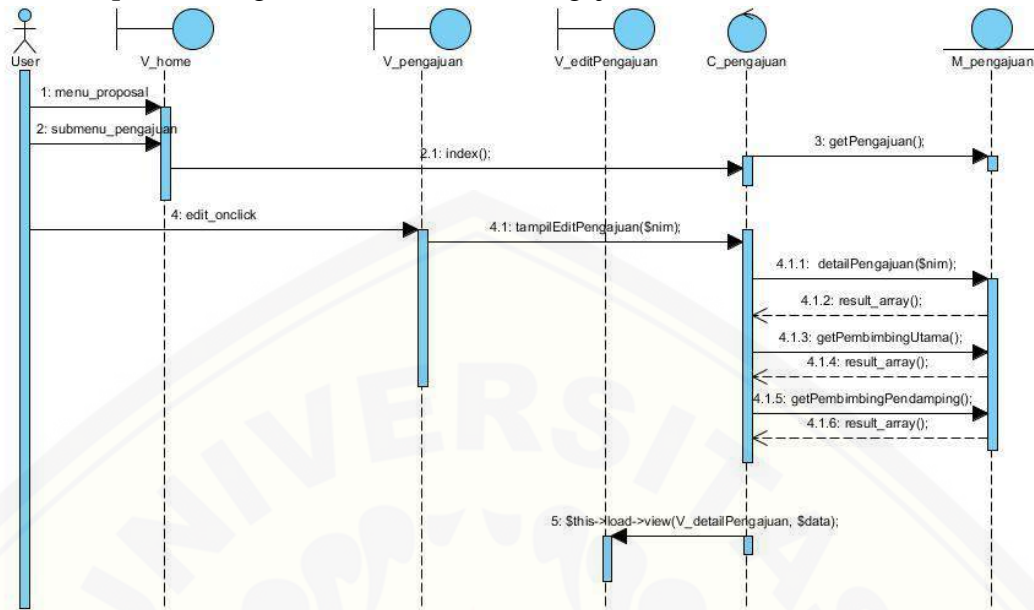
Gambar C.17 *Sequence* diagram melihat rekap mahasiswa bimbingan dosen

C.18 *Sequence* Diagram Melihat Mahasiswa Bimbingan Dosen



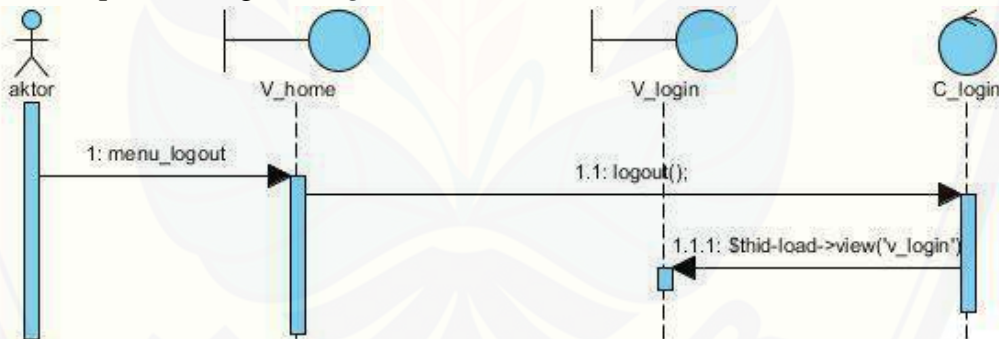
Gambar C.18 *Sequence* diagram melihat mahasiswa bimbingan dosen

C.19 Sequence Diagram Melihat Detail Pengajuan



Gambar C.19 sequence diagram melihat detail pengajuan

C.20 Sequence Diagram Logout



Gambar C.20 sequence diagram logout

Lampiran B. Kode Program

D.1 Kode Program *class* controller/C_pengajuan

```

<?php
if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class C_pengajuan extends CI_Controller {

    public function __construct() {
        parent::__construct();
        $this->load->model('M_pengajuan');
        $this->load->model('M_dosen');
        $this->load->model('M_session');
        $this->load->model('M_winning');
    }
    function index() {
        $data = array(
            'pengajuan' => $this->M_pengajuan-> getPengajuan(),
            'pembimbingUtama' => $this->M_pengajuan-
>getPembimbingUtama()->result(),
            'pembimbingPendamping' => $this->M_pengajuan-
>getPembimbingPendamping()->result()
        );
        $this->load->view('akademik/header');
        $this->load->view('akademik/v_pengajuan',$data);
        $this->load->view('footer');
    }
    function kombi() {
        $view = array(
            'pengajuan' => $this->M_pengajuan-> getPengajuan(),
            'pembimbingUtama' => $this->M_pengajuan-
>getPembimbingUtama()->result()
        );
        $this->load->view('kombi/header');
        $this->load->view('kombi/v_pengajuan',$view);
        $this->load->view('footer');
    }
}

function metodeWinning($keyword,$keywordDosen){
    $gram = 3;
    $window = 2;
    $teks1NonKarakter = $this->M_winning-
>removeSpecialChar($keyword);
    $teks2NonKarakter = $this->M_winning-
>removeSpecialChar($keywordDosen);
    $teks1Split = $this->M_winning->split($teks1NonKarakter,

```

```

$gram);
        $teks2Split = $this->M_winning->split($teks2NonKarakter,
$gram);
        $teks1Hash = $this->M_winning-
>hashSemuaGram($teks1Split);
        $teks2Hash = $this->M_winning-
>hashSemuaGram($teks2Split);
        $teks1Window = $this->M_winning->window($teks1Hash,
$window);
        $teks2Window = $this->M_winning->window($teks2Hash,
$window);
        $window1Terkecil = $this->M_winning-
>windowTerkecil($teks1Window, $window);
        $window2Terkecil = $this->M_winning-
>windowTerkecil($teks2Window, $window);
        $jmlHash = $this->M_winning-
>jmlHash($window1Terkecil, $window2Terkecil);
        $risanHash = $jmlHash['n'];
        $gabunganHash = $jmlHash['u'];
        if($risanHash==null){
            return $p=0;
        }
        else{
            return $p =substr((( $risanHash / $gabunganHash) * 100),
0, 4);
        }
    }

function tampilHalamanRekomendasi($nim) {
    $judul = $this->M_pengajuan->getKeywordMahasiswa($nim)->row();
    $dosen = $this->M_dosen->getPembimbingUtamaRekomendasi();
    foreach ($dosen as $row) {
        $persentase=$this->metodeWinning($judul->keyword1,
$row['keyword']);
        $per2=$this->metodeWinning($judul->keyword2, $row['keyword']);
        $per3=$this->metodeWinning($judul->keyword3, $row['keyword']);
        $this->M_dosen->simpanPersentase($persentase,$per2, $per3,
$row['nip'], $row['keyword']);
    }

    $id = $this->input->post('inibidang');
    $nip = $this->input->post('nip');
    $view = array(
        'biodata' => $this ->M_pengajuan-> detailPengajuan($nim),
        'bidang' => $this ->M_dosen-> getBidang()->result(),
        'dosen'=>$this->M_dosen->sorting(),
        'bimbingan' => $this ->M_dosen-> getBimbingan()
        // 'idBidang'=>$this->M_pengajuan->hasilBidang($id)
    );
    $this->load->view('kombi/header');

```

```

$this->load->view('kombi/v_rekomendasi',$view);
$this->load->view('footer');
}
function pembimbing2(){
    $nip=$this->input->post('nip');
    $nim=$this->input->post('nim');
    $judul = $this->M_pengajuan->getKeywordMahasiswa($nim)->row();
    $dosen2= $this->M_dosen->getPembimbingPendampingRekomendasi();
    foreach ($dosen2 as $row) {
        $persentase=$this->metodeWinnowing($judul->keyword1,
        $row['keyword']);
        $per2=$this->metodeWinnowing($judul->keyword2, $row['keyword']);
        $per3=$this->metodeWinnowing($judul->keyword3, $row['keyword']);
        $this->M_dosen->simpanPersentase($persentase,$per2, $per3, $row['nip'],
        $row['keyword']);
    }
    $dataDosen2=$this->M_dosen->sortingPendamping($nip);
    if($dataDosen2->num_rows()>0){
        $dataDosen2=$dataDosen2->result_array();
        $no=1;
        foreach ($dataDosen2 as $row) {
            echo
            "<tr><td>". $no."</td><td>". $row['nip']."</td><td>". $row['namaDosen']."</td><td>
            ". $row['Jumlahbimbingan']."</td><td>". $row['jumlah']."</td><td style='text-align:
            center;'><input type='radio' name='klas'
            value='". $row['nip']."'." . $row['namaDosen']."'
            onclick='setid2(this.value)'></td></tr>";
            $no++;
        }
    }
}
function rekomBidang(){
    $id=$this->input->post('idBidang');
    $bidang= $this->M_pengajuan->hasilBidang($id);
    if($bidang->num_rows()>0){
        $bidang=$bidang->result_array();
        $no=1;
        foreach ($bidang as $row) {
            echo
            "<tr><td>". $no."</td><td>". $row['nip']."</td><td>". $row['namaDosen']."</td><td>
            ". $row['namaBidang']."</td></tr>";
            $no++;
        }
    }
}
}

public function simpanRekomendasi(){
    $input['nim'] = $this->input->post('nim');
}

```

```

$input['nip']          = $this->input->post('nip');
    $input['nip2']      = $this->input->post('nip2');
    $input['status']    = 'Aktif';
$insert = $this->M_pengajuan->simpanRekomendasi($input);
$tambahU= $this->M_pengajuan->editPembimbingUtama($input);
$tambahP= $this->M_pengajuan->editPembimbingPendamping($input);
redirect('C_pengajuan/kombi');
}

public function insertPengajuan() {
    $input['nim']       = $this->input->post('nim');
    $input['nama']      = $this->input->post('nama');
    $input['nip']       = $this->input->post('pembimbingUtama');
        $input['nip2']   = $this->input-
>post('pembimbingPendamping');
    $input['judul']    = $this->input->post('judul');
    $input['keyword1'] = $this->input->post('keyword1');
    $input['keyword2'] = $this->input->post('keyword2');
    $input['keyword3'] = $this->input->post('keyword3');
    $input['utama']    = 'Utama';
    $input['pendamping'] = 'Pendamping';

    $cek=$this->M_pengajuan->cekPengajuan($input);
    if($cek ==false){
        // echo "<script>alert('Pilih Pembimbing Utama dan Pendamping yang
berbeda..');</script>";
        $this->session->set_flashdata('message', 'Input Gagal! Cek Nim');
        redirect('C_pengajuan');
    }
    $insert = $this->M_pengajuan->insertPengajuan($input);
    $tambahU= $this->M_pengajuan->insertBimbingan($input);
    redirect('C_pengajuan');
}

public function ubahPengajuan()
{
    $input['nim']          = $this->input->post('nim');
    $input['nama']        = $this->input->post('nama');
    $input['nip']         = $this->input-
>post('pembimbingUtama');
        $input['nip2']     = $this->input-
>post('pembimbingPendamping');
    $input['judul']      = $this->input->post('judul');
    $input['keyword1']   = $this->input->post('keyword1');
    $input['keyword2']   = $this->input->post('keyword2');
    $input['keyword3']   = $this->input->post('keyword3');
    $insert = $this->M_pengajuan->editPengajuan($input);
    $tambahU= $this->M_pengajuan->editPembimbingUtama($input);
    $tambahP= $this->M_pengajuan->editPembimbingPendamping($input);
    redirect('C_pengajuan');
}

```

```

        public function TampilDetailPengajuan($nim){
            $data = array(
                'biografia' => $this->M_pengajuan-> detailPengajuan($nim),
                'pembimbingUtama' => $this->M_pengajuan->
                >getPembimbingUtama()->result(),
                'pembimbingPendamping' => $this->M_pengajuan->
                >getPembimbingPendamping()->result()
            );
            $this->load->view('akademik/header');
            $this->load->view('akademik/V_editPengajuan',$data);
            $this->load->view('footer');
        }
        public function TampilDetailPengajuan($nim){
            $data = array(
                'biografia' => $this->M_pengajuan-> detailPengajuan($nim),
                'pembimbingUtama' => $this->M_pengajuan->
                >getPembimbingUtama()->result(),
                'pembimbingPendamping' => $this->M_pengajuan->
                >getPembimbingPendamping()->result()
            );
            $this->load->view('akademik/header');
            $this->load->view('akademik/V_detailPengajuan',$data);
            $this->load->view('footer');
        }
    }
}
?>

```

D.2 Kode Program *class* controller/C_login

```

<?php
if (!defined('BASEPATH'))
    exit('No direct script access allowed');
class C_login extends CI_Controller {

    public function __construct() {
        parent::__construct();
        $this->load->model('M_session');
    }

    function index() {
        $session = $this->M_session->get_session_level();
        if (!$session['session_level']) {
            $this->load->view('V_login');
        } else {
            if ($session['session_level'] == 'akademik') {
                echo "tidak ada admin";
                redirect('C_home');
            } else if ($session['session_level'] == 'kombi') {

```

```

        redirect('C_home/kombi');
    } else if ($session['session_level'] == 'dosen') {
        redirect('C_home/dosen');
    } else {
        redirect('C_home');
    }
}
}
function proses() {
    $username = $this->input->post('username');
    $password = $this->input->post('password');
    $this->load->model('M_login');
    $logic = $this->M_login->get_login($username, $password);
    if ($logic == false) {
        $this->session->set_flashdata('message', 'username dan password salah');
        redirect('C_login');
    } else {
        $scoba = $this->M_login->get_data($username, $password);
        foreach ($scoba->result_array() as $d) {
            $data = array(
                'username' => $d['username'],
                'level' => $d['level'],
            );
        }
        $this->M_session->store_session($data);
        redirect('C_login');
    }
}
function logout() {
    $this->M_session->destroy_session();
    $this->index();
}
}
?>

```

D.3 Kode Program *class* controller/C_home

```

<?php
class C_home extends CI_Controller{

    function __construct(){
        parent::__construct();
        $this->load->model('M_login');
        $this->load->model('M_persetujuan');
        $this->load->model('M_dosen');
        $this->load->library(array('form_validation','session'));
    }
    function index(){
        $data= array(
            'semua' =>$this->M_persetujuan->jmlSemua()
        );
        $this->load->view('akademik/header');
        $this->load->view('akademik/V_home',$data);
    }
}

```

```

function kombi(){
    $data= array(
        'semua' =>$this->M_persetujuan->jmlSemua()
    );
    $this->load->view('kombi/header');
    $this->load->view('kombi/V_home',$data);
}

function dosen(){
    $data=array('dosen'=>$this->M_dosen->jmlBimbingan());
    $this->load->view('dosen/header');
    $this->load->view('dosen/V_home',$data);
}

function lulus(){
    $data= array(
        'lulus' =>$this->M_persetujuan->lulus()
    );
    $this->load->view('akademik/header');
    $this->load->view('akademik/V_lulus',$data);
    $this->load->view('footer');
}

function lulusKombi(){
    $data= array(
        'lulus' =>$this->M_persetujuan->lulus()
    );
    $this->load->view('kombi/header');
    $this->load->view('kombi/V_lulus',$data);
    $this->load->view('footer');
}

}

```

D.4 Kode Program *class* controller/C_persetujuan

```

<?php
if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class C_persetujuan extends CI_Controller {

    public function __construct() {
        parent::__construct();
        $this->load->model('M_persetujuan');
        $this->load->model('M_session');
    }

    function index() {
        $data['persetujuan'] = $this->M_persetujuan->getPersetujuan();
        $this->load->view('akademik/header');
        $this->load->view('akademik/v_persetujuan',$data);
        $this->load->view('footer');
    }
}

```

```

}
function kombi() {
    $data['persetujuan'] = $this->M_persetujuan->getPersetujuan();
    $this->load->view('kombi/header');
    $this->load->view('kombi/v_persetujuan',$data);
    $this->load->view('footer');
}

public function ViewLengkapiSurat($nim){
    $result['data'] = $this->M_persetujuan->getCetak($nim);
    $this->load->view('akademik/header');
    $this->load->view('akademik/v_lengkapiSurat',$result);
    $this->load->view('footer');
}

public function SimpanLengkapiSurat(){
    $input['tanggal'] = $this->input->post('tanggal');
    $input['noSurat'] = $this->input->post('noSurat');
    $input['nim'] = $this->input->post('nim');
    $logic = $this->M_persetujuan->simpanLengkapiSurat($input);

    if ($logic==true) {
        $this->session->set_flashdata('message','Edit berhasil');
        redirect('C_persetujuan');
    } else {
        $this->session->set_flashdata('message','input gagal');
        redirect('C_persetujuan/ViewLengkapiSurat');
    } redirect('C_persetujuan');
}

public function ubahStatus($nim){
    $this->M_persetujuan->ubahStatus($nim);
    redirect('C_persetujuan');
}

}
?>

```

D.5 Kode Program *class* controller/C_dosen

```

<?php
if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class C_dosen extends CI_Controller {

    public function __construct() {
        parent::__construct();
        $this->load->model('M_dosen');
        $this->load->model('M_session');
    }
    function index() {
        $data['dosen'] = $this->M_dosen->getBimbingan();
    }
}

```



```
$this->load->view('akademik/header');
$this->load->view('akademik/v_bimbingan',$data);
$this->load->view('footer');
}

function kombi() {
    $data['dosen'] = $this->M_dosen->getBimbingan();
    $this->load->view('kombi/header');
    $this->load->view('kombi/v_bimbingan',$data);
    $this->load->view('footer');
}

function keyword(){
    $data['dosen'] = $this->M_dosen->getKeyword();
    $this->load->view('akademik/header');
    $this->load->view('akademik/v_keywordDosen',$data);
    $this->load->view('footer');
}

function tampilUpdateKeyword($nip){
    $data['dosen'] = $this->M_dosen->keyword($nip);
    $this->load->view('akademik/header');
    $this->load->view('akademik/v_tambahKeyword',$data);
    $this->load->view('footer');
}

function updateKeyword(){
    $input['nip'] = $this->input->post('nip');
    $input['keyword1'] = $this->input->post('keyword1');
    $input['keyword2'] = $this->input->post('keyword2');
    $input['keyword3'] = $this->input->post('keyword3');
    $insert = $this->M_dosen->updateKeyword($input);
    redirect('C_dosen/keyword');
}

function getDosen() {

    $data['dosen'] = $this->M_dosen->getDosen();
    $this->load->view('akademik/header');
    $this->load->view('akademik/v_dosen',$data);
    $this->load->view('footer');
}

function getDosenKombi() {

    $data['dosen'] = $this->M_dosen->getDosen();
    $this->load->view('kombi/header');
    $this->load->view('kombi/v_dosen',$data);
    $this->load->view('footer');
}

public function insertDosen() {
    $input['nip'] = $this->input->post('nip');
    $input['namaDosen'] = $this->input->post('namaDosen');
    $input['ttl'] = $this->input->post('ttl');
    $input['Pangkat_Gol'] = $this->input->post('Pangkat/Gol');
    $input['jabatanFungsional'] = $this->input->post('jabatanFungsional');
    $input['keyword'] = $this->input->post('keyword');
    $cek=$this->M_dosen->cekNip($input);
    if($cek ==false){
        $this->session->set_flashdata('message', 'Input Gagal! Cek Nip');
```

```
        redirect('C_dosen/getDosen');
    }
    elseif($input['keyword']==""){
        echo "<script>alert('Pilih Pembimbing Utama dan Pendamping yang
berbeda..');</script>";
    }
    $insert = $this->M_dosen->insertDosen($input);
    redirect('C_dosen/getDosen');
}

function detailBimbingan($nip){
    $data= array('bimbingan'=> $this->M_dosen->detailBimbingan($nip),
        'dosen'=> $this->M_dosen->getDetailDosen($nip)
    );
    $this->load->view('akademik/header');
    $this->load->view('akademik/v_detailBimbingan',$data);
    $this->load->view('footer');
}

function detailBimbinganKombi($nip){
    $data['bimbingan'] = $this->M_dosen-> detailBimbingan($nip);
    $this->load->view('kombi/header');
    $this->load->view('kombi/v_detailBimbingan',$data);
    $this->load->view('footer');
}

function tampilEditDosen($nip){
    $data= array('dosen'=> $this->M_dosen->getDetailDosen($nip));
    $this->load->view('akademik/header');
    $this->load->view('akademik/v_editDosen',$data);
    $this->load->view('footer');
}

function editDosen()
{
    $input['nip']      = $this->input->post('nip');
    $input['namaDosen'] = $this->input->post('namaDosen');
    $input['ttl']      = $this->input->post('ttl');
    $input['Pangkat_Gol'] = $this->input->post('Pangkat/Gol');
    $input['jabatanFungsional'] = $this->input->post('jabatanFungsional');
    $insert = $this->M_dosen->editDosen($input);
    redirect('C_dosen/getDosen');
}

function bimbinganLuar(){
    $data['dosen'] = $this->M_dosen-> getBimbingan();
    $this->load->view('V_bimbingan',$data);
    $this->load->view('footer');
}

function detailBimbinganLuar($nip){
    $data['bimbingan'] = $this->M_dosen-> detailBimbingan($nip);
    $this->load->view('v_detailBimbingan',$data);
}

function profil(){
```

```

    $data=array('dosen'=>$this->M_dosen->ambilProfil());
    $this->load->view('dosen/header');
    $this->load->view('dosen/V_profil',$data);
    $this->load->view('footer');
}
function bimbingan(){
    $data=array('dosen'=>$this->M_dosen->ambilBimbingan(),
        'biodata'=>$this->M_dosen->ambilProfil());
    $this->load->view('dosen/header');
    $this->load->view('dosen/V_bimbingan',$data);
    $this->load->view('footer');
}
}
?>

```

D.6 Kode Program *class* controller/C_bidang

```

<?php
if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class C_bidang extends CI_Controller {

    public function __construct() {
        parent::__construct();
        $this->load->model('M_dosen');
        $this->load->model('M_session');
    }
    function index(){
        $data['bidang']= $this->M_dosen->getBidang()->result_array();

        $this->load->view('akademik/header');
        $this->load->view('akademik/V_bidang',$data);
        $this->load->view('footer');
    }
    function kombi(){
        $data['bidang']= $this->M_dosen->getBidang()->result_array();
        $this->load->view('kombi/header');
        $this->load->view('kombi/V_bidang',$data);
        $this->load->view('footer');
    }

    function tambahBidang(){
        $input['namaBidang']= $this->input->post('namaBidang');
        $hasil=$this->M_dosen->insertBidang($input);
        redirect('C_bidang');
    }
    function tambahBidangKombi(){
        $input['namaBidang']= $this->input->post('namaBidang');
        $hasil=$this->M_dosen->insertBidang($input);
        redirect('C_bidang/kombi');
    }
}

```

```
function tampilEdit($id){
    $data['bidang']= $this->M_dosen->getIdBidang($id);
    $this->load->view('akademik/header');
    $this->load->view('akademik/V_editBidang',$data);
    $this->load->view('footer');
}
function tampilEditKombi($id){
    $data['bidang']= $this->M_dosen->getIdBidang($id);
    $this->load->view('kombi/header');
    $this->load->view('kombi/V_editBidang',$data);
    $this->load->view('footer');
}
function editBidang($id){
    $input['namaBidang']= $this->input->post('namaBidang');
    $input['idBidang']= $this->input->post('idBidang');
    $hasil=$this->M_dosen->updateBidang($input);
    redirect('C_bidang');
}
function editBidangKombi($id){
    $input['namaBidang']= $this->input->post('namaBidang');
    $input['idBidang']= $this->input->post('idBidang');
    $hasil=$this->M_dosen->updateBidang($input);
    redirect('C_bidang/kombi');
}
function tampilTambahBidangDosen($id){
    $data = array(
        'bidang' =>$this->M_dosen->getIdBidang($id),
        'dosen' =>$this->M_dosen->getDosen()
    );
    $this->load->view('akademik/header');
    $this->load->view('akademik/V_tambahBidangDosen',$data);
    $this->load->view('footer');
}
function tampilTambahBidangDosenKombi($id){
    $data = array(
        'bidang' =>$this->M_dosen->getIdBidang($id),
        'dosen' =>$this->M_dosen->getDosen()
    );
    $this->load->view('kombi/header');
    $this->load->view('kombi/V_tambahBidangDosen',$data);
    $this->load->view('footer');
}
function simpanTambahBidang(){
    $input['idBidang'] =$this->input->post('idBidang');
    $input['nip'] =$this->input->post('nip');
    $cek=$this->M_dosen->cekBidang($input);
    if($cek==false){
        $this->session->set_flashdata('message', 'Input Gagal! Data tersebut sudah
ada');
        redirect('C_bidang');
    }
}
```

```
        else{
            $this->session->set_flashdata('message', 'Input Berhasil');
            $simpan= $this->M_dosen->tambahBidangDosen($input);
            redirect('C_bidang');

        }
        redirect('C_bidang');
    }
    function simpanTambahBidangKombi(){
        $input['idBidang'] =$this->input->post('idBidang');
        $input['nip'] =$this->input->post('nip');
        $cek=$this->M_dosen->cekBidang($input);
        if($cek==false){
            $this->session->set_flashdata('message', 'Input Gagal! Data tersebut sudah
ada');
            redirect('C_bidang/kombi');
        }
        else{
            $this->session->set_flashdata('message', 'Input Berhasil');
            $simpan= $this->M_dosen->tambahBidangDosen($input);
            redirect('C_bidang/kombi');
        }
        redirect('C_bidang/kombi');
    }
    function tampilListBidang($id){
        $data['bidang']=$this->M_dosen->getBidangId($id);
        $data['dosen']=$this->M_dosen->getDetailDosen($id);
        $this->load->view('akademik/header');
        $this->load->view('akademik/V_lihatBidangDosen',$data);
        $this->load->view('footer');
    }
    function tampilListBidangKombi($id){
        $data['bidang']=$this->M_dosen->getBidangId($id);
        $this->load->view('kombi/header');
        $this->load->view('kombi/V_lihatBidangDosen',$data);
        $this->load->view('footer');
    }
}
}
```

D.7 Kode Program *class* controller/Cetak

```
<?php
if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Cetak extends CI_Controller{
    public function __construct()
    {
        parent::__construct();
        $this->load->helper('form');
        $this->load->library('fpdf');
    }
    public function cetak($nim) {
```

```
$this->load->model('m_persetujuan');  
$result['data'] = $this->m_persetujuan->getCetak($nim);  
$this->load->view('rtpdf/index',$result);  
}  
}
```



Lampiran C. Pengujian Black Box

No.	Fitur	Aksi	Hasil	Kesimpulan
1.	<i>Login</i>	Mengisi <i>username</i> dan <i>password</i> lalu klik tombol <i>login</i>	<i>Login</i> berhasil dan menampilkan halaman awal masing-masing aktor	[√] Berhasil [] Gagal
		Username atau password salah	Menampilkan pesan “Login gagal! <i>Username</i> atau <i>password</i> salah	[√] Berhasil [] Gagal
2.	Menambah pengajuan proposal	Mengisi data pengajuan lalu mengklik tombol simpan	Menyimpan pengajuan dan menampilkan pesan “data berhasil ditambah”	[√] Berhasil [] Gagal
		Data pengajuan telah ada	Menampilkan pesan “data pengajuan telah ada”	[√] Berhasil [] Gagal
		Memilih Dosen pembimbing yang sama	Menampilkan pesan “Pilih dosen yang berbeda”	[√] Berhasil [] Gagal
		Kolom belum diisi	Menampilkan pesan dengan menunjuk pada kolom yang belum diisi	[√] Berhasil [] Gagal
3.	Mengubah pengajuan proposal	Mengisi data pengajuan lalu mengklik tombol simpan	Menyimpan pengajuan dan menampilkan halaman pengajuan	[√] Berhasil [] Gagal
		Memilih Dosen pembimbing yang sama	Menampilkan pesan “Pilih dosen yang berbeda”	[√] Berhasil [] Gagal

4.	Melihat data pengajuan proposal	Klik submenu pengajuan	Menampilkan data pengajuan	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
5.	Melihat data dosen	Klik submenu data dosen	Menampilkan data dosen	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
6.	Melihat data persetujuan	Klik submenu data persetujuan	Menampilkan data persetujuan	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
7.	Menambah bidang keahlian dosen	Klik tombol cari, pilih dosen lalu mengklik tombol simpan	Menyimpan bidang dosen dan menampilkan pesan “berhasil ditambah”	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Kolom belum diisi	Menampilkan pesan pada kolom yang kosong	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
8.	Melengkapi cetak surat tugas	Mengisi kolom tanggal dan nomer surat dan pilih tombol simpan	Menyimpan data pengajuan dan menampilkan pesan berhasil lengkapi surat tugas	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Kolom belum diisi	Menampilkan pesan pada kolom yang kosong	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
9.	Menambah data dosen	Mengisi form tambah data dosen lalu klik simpan	Menyimpan data dosen dan menampilkan pesan berhasil di simpan	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Kolom belum diisi	Menampilkan pesan pada kolom yang kosong	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

10.	Melihat bidang keahlian	Memilih sub menu bidang	Menampilkan data bidang keahlian	<input checked="" type="checkbox"/> Berhasil
				<input type="checkbox"/> Gagal
11.	Mengubah data dosen	Mengisi data dosen lalu klik simpan	Menyimpan perubahan dan menampilkan data dosen	<input checked="" type="checkbox"/> Berhasil
				<input type="checkbox"/> Gagal
12.	Mencetak surat tugas	Memilih tombol cetak	Menampilkan surat tugas format .pdf	<input checked="" type="checkbox"/> Berhasil
				<input type="checkbox"/> Gagal
13.	Manajemen rekomendasi dosen pembimbing	Memilih pengajuan proposal	Menampilkan halaman pengajuan	<input checked="" type="checkbox"/> Berhasil
				<input type="checkbox"/> Gagal
		Memilih tombol cari pembimbing utama lalu pilih dosen	Menampilkan tabel rekomendasi pembimbing utama	<input checked="" type="checkbox"/> Berhasil
				<input type="checkbox"/> Gagal
		Memilih tombol cari pembimbing pendamping lalu pilih dosen	Menampilkan tabel rekomendasi pembimbing pendamping	<input checked="" type="checkbox"/> Berhasil
				<input type="checkbox"/> Gagal
		Memilih tombol simpan	Menyimpan rekomendasi	<input checked="" type="checkbox"/> Berhasil
				<input type="checkbox"/> Gagal
14.	Melihat keyword penelitian dosen	Memilih submenu keyword	Menampilkan data dosen	<input checked="" type="checkbox"/> Berhasil
				<input type="checkbox"/> Gagal
		Memilih menu lihat penelitian	Menampilkan data keyword penelitian dosen	<input checked="" type="checkbox"/> Berhasil
				<input type="checkbox"/> Gagal

15.	Mengubah status lulus mahasiswa	Mengklik tombol delete	Mengubah status mahasiswa menjadi tidak aktif	<input checked="" type="checkbox"/> Berhasil
		Klik tombol “ya”		<input type="checkbox"/> Gagal
		Klik tombol “tidak”	Tidak mengubah status mahasiswa	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
16.	Menambah bidang	Mengisi nama bidang	Menambah bidang dan menampilkan pesan “berhasil tambah bidang”	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Klik simpan		
		Data telah ada	Menampilkan pesan “data telah ada	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
17.	Menambah <i>keyword</i> dosen	Mengisi <i>keyword</i> penelitian	Menyimpan data dan menampilkan pesan berhasil tambah data	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Klik simpan		
18.	Melihat rekap bimbingan dosen	Klik submenu bimbingan	Menampilkan data jumlah bimbingan	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
19.	Melihat mahasiswa bimbingan dosen	Memilih menu detail	Menampilkan detail mahasiswa bimbingan dosen	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		bimbingan		
20.	Melihat detail pengajuan	Memilih menu proposal	Menampilkan data proposal	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Memilih menu detail	Menampilkan detail pengajuan	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
21.	<i>Logout</i>	Klik tombol <i>logout</i>	Berhasil keluar sistem dan menampilkan halaman awal	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal