



**IMPLEMENTASI ALGORITMA *WINNOWER* BERBASIS *K-GRAM* UNTUK
MENGIDENTIFIKASI *PLAGIARISME* PADA FILE DOKUMEN BERBASIS
TEKS**

SKRIPSI

Oleh

Fiqih Nur Muharrom

NIM 112410101052

PROGRAM STUDI SISTEM INFORMASI

UNIVERSITAS JEMBER

2016



**IMPLEMENTASI ALGORITMA *WINNOWER* BERBASIS *K-GRAM* UNTUK
MENGIDENTIFIKASI *PLAGIARISME* PADA FILE DOKUMEN BERBASIS
TEKS**

SKRIPSI

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat untuk menyelesaikan Program Studi Sistem Informasi (S1) dan mendapatkan gelar Sarjana Komputer

Oleh

Fiqih Nur Muharrom

NIM 112410101052

**PROGRAM STUDI SISTEM INFORMASI
UNIVERSITAS JEMBER**

2016

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Kedua orang tua saya, Ibunda Fatimah dan Ayahanda Nasrawi;
2. Kedua saudara saya, Indah Ainur R dan Nuril Iksan R;
3. Guru-guruku sejak taman kanak-kanak hingga perguruan tinggi;
4. Seluruh teman-teman yang selalu memberikan motivasi;
5. Almamater Program Studi Sistem Informasi Universitas Jember;

MOTTO

“Tidak pernah ada yang namanya orang bodoh, hanya orang pemalas yang membiarkan dirinya terlihat bodoh”



PERNYATAAN

Saya yang bertanda tangan dibawah ini:

nama : Fiqih Nur Muharrom

NIM : 112410101052

menyatakan dengan sesungguhnya bahwa karya tulis ilmiah dengan judul “Implementasi Algoritma *Winnowing* Berbasis *K-Gram* untuk Mengidentifikasi Plagiarisme pada File Dokumen Berbasis Teks” adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada instansi 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 sebenarnya, tanpa ada tekanan dan paksaan dari pihak mana pun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, 23 Februari 2016

Yang menyatakan,

Fiqih Nur Muharrom

NIM.112410101052

SKRIPSI

**IMPLEMENTASI ALGORITMA *WINNOWER* BERBASIS *K-GRAM* UNTUK
MENGIDENTIFIKASI *PLAGIARISME* PADA FILE DOKUMEN BERBASIS
TEKS**

Oleh

Fiqih Nur Muharrom

NIM 112410101052

Pembimbing

Dosen Pembimbing Utama : Anang Andrianto, ST., MT

Dosen Pembimbing Anggota : Yanuar Nurdiansyah, ST., M.Cs

PENGESAHAN PEMBIMBING

Skripsi berjudul “Implementasi Algoritma *Winnowing* Berbasis *K-Gram* untuk Mengidentifikasi *Plagiarisme* pada File Dokumen Berbasis Teks”, telah diuji dan disahkan pada:

hari, tanggal : Selasa, 23 Februari 2016

tempat : Program Studi Sistem Informasi Universitas Jember

Disetujui oleh :

Pembimbing I,

Pembimbing II,

Anang Andrianto, ST., MT
NIP. 196906151997021002

Yanuar Nurdiansyah, ST., M.Cs
NIP. 198201012010121004

PENGESAHAN

Skripsi berjudul “Implementasi Algoritma *Winnowing* Berbasis *K-Gram* untuk Mengidentifikasi *Plagiarisme* pada File Dokumen Berbasis Teks”, telah diuji dan disahkan pada:

hari, tanggal : Selasa, 23 Februari 2016

tempat : Program Studi Sistem Informasi Universitas Jember

Penguji 1,

Penguji 2,

Dr. Saiful Bukhori, ST., M.Kom
NIP. 196811131994121001

M. Arief Hidayat, S.Kom., M.Kom
NIP. 198101232010121003

Mengesahkan
Ketua Program Studi,

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

RINGKASAN

Implementasi Algoritma *Winnowing* Berbasis *K-Gram* untuk Mengidentifikasi *Plagiarisme* pada File Dokumen Berbasis Teks; Fiqih Nur Muharrom, 112410101052; 2016: 119 halaman; Program Studi Sistem Informasi Universitas Jember.

Dampak negatif yang tidak dapat dihindari pada era mudahnya bertukar dan mendapatkan informasi saat ini ialah semakin maraknya penjiplakan atau yang biasa kita kenal dengan *plagiarisme*. Tindakan *plagiarisme* terjadi ketika siswa atau mahasiswa mendapat tugas dan dikejar waktu pengumpulan tugas. Siswa atau mahasiswa menganggap bahwa melakukan palgiat merupakan cara tercepat untuk menyelesaikan tugas-tugasnya. Hal tersebut mempengaruhi hasil evaluasi belajar siswa atau mahasiswa. Permasalahan tersebut juga berdampak terhadap para pengajar yang sulit mendeteksi *plagiarisme* mengingat tidak sedikit *file-file* tugas yang dikumpulkan. Berdasarkan permasalahan diatas, diperlukan adanya sebuah sistem atau aplikasi yang dapat digunakan untuk mengidentifikasi adanya tindakan *plagiarisme* pada dokumen tugas yang sebagian besar berupa dokumen berbasis teks.

Aplikasi pendeteksi *plagiarisme* yang menggunakan algoritma *winnowing* sebagai algoritma pencarian kemiripan dokumen. Sistem menerima *input* berupa dua dokumen teks berbahasa indonesia dengan ekstensi *.doc* dan/atau *.txt* sebagai dokumen asli serta dokumen uji yang kemudian diproses hingga dihasilkan persentase kemiripan dokumen. Persentase kemiripan dokumen asli dengan dokumen uji nantinya dibandingkan dengan persentase minimal *plagiarisme*.

PRAKATA

Puji syukur kehadirat Allah SWT atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul “Implementasi Algoritma *Winnowing* Berbasis *K-Gram* untuk Mengidentifikasi *Plagiarisme* pada File Dokumen Berbasis Teks”. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Strata Satu (S1) pada Program Studi Sistem Informasi Universitas Jember.

Penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

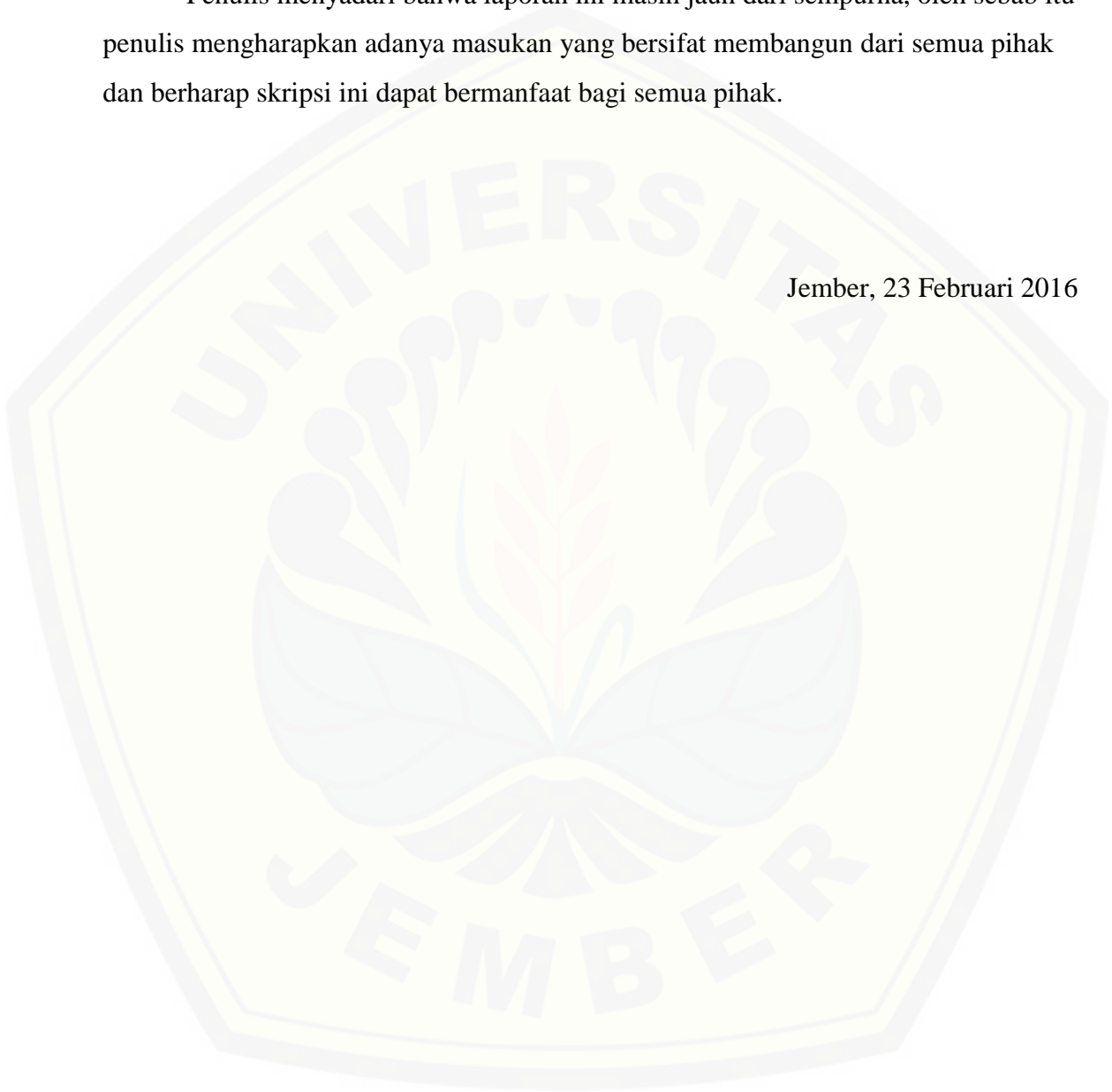
1. Prof. Drs. Slamini, M.Comp.Sc., Ph.D., selaku Ketua Program Studi Sistem Informasi Universitas Jember;
2. Anang Andrianto, ST., MT., selaku Dosen Pembimbing Utama, dan Yanuar Nurdiansyah, ST., M.Cs., selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi ini;
3. Nelly Oktavia Adiwijaya, S.Si., MT., selaku Dosen Pembimbing Akademik yang telah membimbing selama penulis menjadi mahasiswa;
4. Seluruh Bapak dan Ibu dosen beserta staf karyawan Program Studi Sistem Informasi;
5. Ayah, Ibu dan adik-adik tercinta yang telah memberikan do'a dan semangat kepada penulis sehingga skripsi ini dapat diselesaikan;
6. Sahabat terbaik Yopi Wahyu Purwanto, Rozi Rikza Akbar, Nafta Ryandika, Khoirul Anwar, Mohammad Arifin, dan Abdul Afif yang selalu memberikan semangat serta motivasi kepada penulis;
7. Penghuni kontrakan Bengawan Solo, Faisol, Tri, Irman, Huda, Recky, Munir, Agim, dan Dika;
8. Keluarga NEFOTION Program Studi Sistem Informasi;

9. Semua Mahasiswa Program Studi Sistem Informasi Universitas Jember;

10. Semua pihak yang telah membantu dan tidak dapat disebutkan satu-persatu

Penulis menyadari bahwa laporan ini masih jauh dari sempurna, oleh sebab itu penulis mengharapkan adanya masukan yang bersifat membangun dari semua pihak dan berharap skripsi ini dapat bermanfaat bagi semua pihak.

Jember, 23 Februari 2016



DAFTAR ISI

PERSEMBAHAN.....	ii
MOTTO	iii
PERNYATAAN.....	iv
PENGESAHAN PEMBIMBING.....	vi
PENGESAHAN	vii
RINGKASAN	viii
PRAKATA.....	ix
DAFTAR ISI.....	xi
DAFTAR TABEL.....	xv
DAFTAR GAMBAR	xvi
DAFTAR LAMPIRAN.....	xix
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan dan Manfaat	2
1.3.1. Tujuan	3
1.3.2. Manfaat	3
1.4 Batasan Masalah.....	3
1.5 Sistematika Penulisan	4
BAB 2. TINJAUAN PUSTAKA	6
2.1 Penelitian Terdahulu	6

2.2	<i>Plagiarisme</i>	7
2.3	Algoritma <i>Stemming</i> Bahasa Indonesia	8
2.3.1	<i>Confix Stripping Stemmer</i>	8
2.3.2	<i>Enhanced Confix Stripping Stemmer</i>	12
2.4	Algoritma <i>Winnowing</i>	14
2.4.1	Konsep Kerja Algoritma <i>Winnowing</i>	14
2.4.2	<i>K-Gram</i>	15
2.4.3	<i>Rolling Hash</i>	16
2.4.4	<i>Document Fingerprint</i>	17
2.5	Perhitungan Kemiripan (<i>similarity</i>)	18
2.6	Model <i>Waterfall</i>	19
2.6.1	<i>Requirement analysis and specification</i>	20
2.6.2	<i>Design</i>	21
2.6.3	<i>Coding and Unit Testing</i>	21
2.6.4	<i>Integration and System Testing</i>	21
2.6.5	<i>Maintenance</i>	22
BAB 3 METODE PENELITIAN.....		23
3.1	Analisis Kebutuhan Sistem	23
3.1.1	Algoritma <i>Winnowing</i> sebagai algoritma pencarian kemiripan	24
3.1.2	Data yang dibutuhkan sistem	30
3.2	Desain Sistem.....	31
3.3	Implementasi	31
3.4	Testing.....	32

BAB 4 DESAIN DAN PENGEMBANGAN SISTEM	33
4.1 Deskripsi Umum Sistem	33
4.1.1 <i>Statement of Purpose (SOP)</i>	33
4.1.2 Fungsi-fungsi dalam sistem.....	34
4.2 Analisis Kebutuhan	35
4.2.1 Kebutuhan Fungsional	35
4.2.2 Kebutuhan Non Fungsional.....	36
4.3 Desain Sistem.....	36
4.3.1 <i>Business Process</i>	36
4.3.2 <i>Usecase Diagram</i>	37
4.3.3 Scenario Diagram.....	40
4.3.4 <i>Activity Diagram</i>	51
4.3.5 <i>Sequence diagram</i>	61
4.3.6 <i>Class Diagram</i>	71
4.3.7 <i>Entity Relationship Diagram (ERD)</i>	72
4.4 Implementasi Sistem	73
4.5 <i>Testing</i>	73
4.5.1 <i>White Box Testing</i>	74
4.5.1 Black Box Testing.....	80
BAB 5 HASIL DAN PEMBAHASAN.....	81
5.1 Hasil Pembuatan Sistem.....	81
5.1.1 Tampilan Fitur <i>Login</i> sistem.....	82
5.1.2 Tampilan Fitur <i>Register User</i>	82

5.1.3	Tampilan Fitur Tambah Data Jurnal	83
5.1.4	Tampilan Fitur Hapus Data Jurnal	87
5.1.5	Tampilan Fitur <i>Approve</i> data jurnal	88
5.1.6	Tampilan Fitur nonaktifkan <i>user</i>	90
5.1.7	Tampilan Fitur Ubah Profil <i>User</i>	91
5.1.8	Tampilan Fitur Ubah <i>Password User</i>	92
5.1.9	Tampilan Fitur <i>View Log</i> Aktifitas <i>User</i>	94
5.1.10	Tampilan Fitur <i>View Data</i> Jurnal	94
5.1.11	Tampilan Fitur <i>View Data User</i>	95
5.1.12	Tampilan Fitur Mengecek Plagiat Jurnal	96
5.2	Pembahasan.....	100
5.2.1	Perancangan dan Pembangunan Sistem	100
5.2.2	Implementasi Algoritma <i>Winnowing</i> pada Sistem.....	101
BAB 6 PENUTUP		118
6.1	Kesimpulan	118
6.2	Saran.....	119
DAFTAR PUSTAKA		120
LAMPIRAN.....		122

DAFTAR TABEL

Tabel 2.1 Kombinasi awalan-akhiran yang tidak diperbolehkan.....	9
Tabel 2.2 Aturan pemotongan imbuhan “me-”	9
Tabel 2.3 Aturan pemotongan imbuhan “pe-”	10
Tabel 2.4 Aturan pemotongan imbuhan “be-”	10
Tabel 2.5 Aturan pemotongan imbuhan “te-”	11
Tabel 2.6 Modifikasi aturan pada Tabel 2.2	12
Tabel 2.7 Modifikasi aturan pada Tabel 2.3	13
Tabel 4.1 Definisi <i>usecase diagram</i>	39
Tabel 4.2 <i>Scenario login admin</i>	41
Tabel 4.3 <i>Scenario login mahasiswa</i>	42
Tabel 4.4 <i>Scenario register user baru</i>	43
Tabel 4.5 <i>Scenario tambah data jurnal baru</i>	44
Tabel 4.6 <i>Scenario hapus data jurnal</i>	46
Tabel 4.7 <i>Scenario approve data jurnal</i>	48
Tabel 4.8 <i>Test case fitur tambah data jurnal</i>	79
Tabel 5.1 Nilai <i>gram</i> dan <i>window</i>	102
Tabel 5.2 Hasil proses <i>stemming</i>	103
Tabel 5.3 Hasil proses <i>filtering</i>	103
Tabel 5.4 Hasil Pembentukan <i>Gram</i> pada Kalimat tanpa <i>Stemming</i>	106
Tabel 5.5 Hasil Pembentukan <i>Gram</i> pada Kalimat dengan <i>Stemming</i>	107
Tabel 5.6 Nilai hash pada kalimat tanpa <i>stemming</i>	109
Tabel 5.7 Nilai hash pada kalimat dengan <i>stemming</i>	109
Tabel 5.8 Hasil pembentukan <i>window</i> pada kalimat tanpa <i>stemming</i>	111
Tabel 5.9 Hasil pembentukan <i>window</i> pada kalimat dengan <i>stemming</i>	111
Tabel 5.10 Hasil pemilihan <i>fingerprint</i> pada kalimat tanpa <i>stemming</i>	113
Tabel 5.11 Hasil pemilihan <i>fingerprint</i> pada kalimat dengan <i>stemming</i>	113

DAFTAR GAMBAR

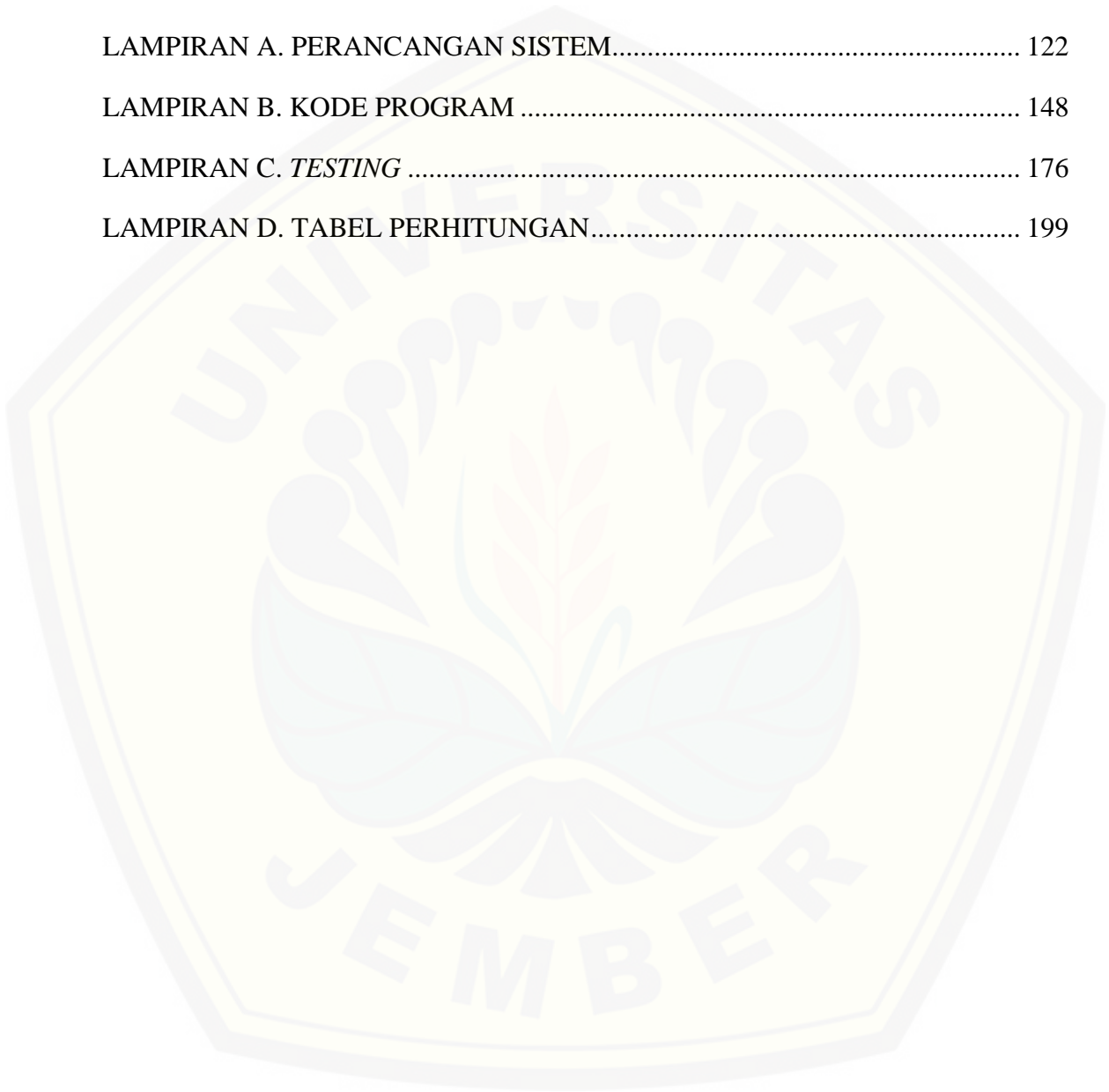
Gambar 2.1 Hasil pembuatan k-gram terhadap teks dengan nilai $k = 5$	16
Gambar 2.2 Nilai <i>hash</i> dari setiap <i>gram</i>	17
Gambar 2.3 Pembentukan <i>window</i> $w = 4$	18
Gambar 2.4 <i>Fingerprint</i> diambil dari nilai <i>hash</i> terkecil	18
Gambar 2.5 <i>Iterative waterfall model</i> (Mall, 2009).....	19
Gambar 4.1 <i>Business process</i> sistem pendeteksi <i>plagiarisme</i>	36
Gambar 4.2 <i>Usecase diagram</i>	38
Gambar 4.3 <i>Activity diagram login</i>	51
Gambar 4.4 <i>Activity diagram register user</i> baru.....	52
Gambar 4.5 <i>Activity diagram upload</i> jurnal.....	53
Gambar 4.6 <i>Activity diagram preview upload</i> jurnal	54
Gambar 4.7 <i>Activity diagram pengecekan</i> dengan jurnal di <i>database</i>	56
Gambar 4.8 <i>Activity diagram</i> simpan data jurnal.....	57
Gambar 4.9 <i>Activity diagram</i> hapus data jurnal.....	58
Gambar 4.10 <i>Activity diagram approve</i> data jurnal	59
Gambar 4.11 <i>Sequence diagram login</i> admin	62
Gambar 4.12 <i>Sequence diagram login</i> mahasiswa	63
Gambar 4.13 <i>Sequence diagram register user</i>	64
Gambar 4.14 <i>Sequence diagram upload</i> jurnal.....	65
Gambar 4.15 <i>Sequence diagram</i> pembentukan <i>fingerprint</i>	66
Gambar 4.16 <i>Sequence diagram</i> cek jurnal	67
Gambar 4.17 <i>Sequence diagram</i> simpan data jurnal.....	68
Gambar 4.18 <i>Sequence diagram</i> hapus data jurnal.....	69
Gambar 4.19 <i>Sequence diagram approve</i> data jurnal	70
Gambar 4.20 <i>Class diagram</i>	72
Gambar 4.21 <i>Entity relationship diagram</i>	73

Gambar 4.22 <i>Method upload_jurnal()</i>	75
Gambar 4.23 <i>Method proses_jurnal()</i>	75
Gambar 4.24 <i>Method check_db()</i>	76
Gambar 4.25 <i>Method simpan_database()</i>	77
Gambar 4.26 Diagram alir <i>method upload_jurnal()</i>	77
Gambar 4.27 Diagram alir <i>method proses_jurnal()</i>	77
Gambar 4.28 Diagram alir <i>method check_db()</i>	78
Gambar 4.29 Diagram alir <i>method simpan_database()</i>	78
Gambar 5.1 Tampilan halaman <i>login</i>	82
Gambar 5.2 Tampilan halaman <i>register user</i>	83
Gambar 5.3 Tampilan <i>form upload jurnal</i>	84
Gambar 5.4 Tampilan <i>pop-up edit penulis</i>	84
Gambar 5.5 Tampilan halaman <i>preview jurnal</i>	85
Gambar 5.6 Tampilan halaman hasil pembentukan <i>fingerprint jurnal</i>	86
Gambar 5.7 Tampilan detail perhitungan pembentukan <i>fingerprint jurnal</i>	86
Gambar 5.8 Tampilan halaman hasil pengecekan dengan <i>database</i>	87
Gambar 5.9 Tampilan <i>pop-up</i> konfirmasi hapus data jurnal.....	88
Gambar 5.10 Tampilan <i>list</i> data jurnal yang berstatus <i>pending</i>	89
Gambar 5.11 Tampilan <i>pop-up</i> detail jurnal yang berstatus <i>pending</i>	89
Gambar 5.12 Tampilan <i>pop-up</i> setuju data jurnal yang berstatus <i>pending</i>	90
Gambar 5.13 Tampilan <i>pop-up</i> hapus data jurnal yang berstatus <i>pending</i>	90
Gambar 5.14 Tampilan <i>pop-up</i> konfirmasi nonaktifkan <i>user</i>	91
Gambar 5.15 Tampilan halaman profil	92
Gambar 5.16 Tampilan <i>pop-up</i> ubah profil.....	92
Gambar 5.17 Tampilan fitur ubah <i>password user</i>	93
Gambar 5.18 Tampilan peringatan penanganan <i>exception</i>	93
Gambar 5.19 Tampilan fitur <i>view log</i> aktifitas <i>user</i>	94
Gambar 5.20 Tampilan fitur <i>view</i> data jurnal	95
Gambar 5.21 Tampilan fitur <i>view</i> data <i>user</i>	95

Gambar 5.22 Tampilan awal fitur mengecek plagiat jurnal.....	96
Gambar 5.23 Tampilan <i>pop-up edit</i> penulis.....	97
Gambar 5.24 Tampilan <i>preview</i> jurnal yang diunggah.....	97
Gambar 5.25 Tampilan hasil pembentukan <i>fingerprint</i>	98
Gambar 5.26 Tampilan detail perhitungan pembentukan <i>fingerprint</i>	98
Gambar 5.27 Tampilan halaman pilih jurnal	99
Gambar 5.28 Tampilan hasil pengecekan jurnal yang diunggah	100
Gambar 5.29 Kalimat Pembanding	102
Gambar 5.30 Kalimat Uji.....	103
Gambar 5.31 <i>Method get_content()</i>	104
Gambar 5.32 Tampilan Hasil Ekstraksi Konten <i>File</i> Dokumen pada Sistem.....	105
Gambar 5.33 <i>Method stemmer()</i>	105
Gambar 5.34 <i>Method filtering()</i>	106
Gambar 5.35 Tampilan Hasil Proses <i>Stemming</i> dan <i>Filtering</i> pada Sistem	106
Gambar 5.36 <i>Method parsing_gram()</i>	108
Gambar 5.37 Tampilan Hasil Pembentukan <i>Gram</i> pada Sistem.....	108
Gambar 5.38 <i>Method convert_hash()</i>	110
Gambar 5.39 Tampilan Hasil Nilai <i>Hash</i> pada Sistem	110
Gambar 5.40 <i>Method parsing_window()</i>	112
Gambar 5.41 Tampilan Hasil Pembentukan <i>Window</i> pada Sistem.....	112
Gambar 5.42 <i>Method fingerprint()</i>	114
Gambar 5.43 Tampilan Hasil Pemilihan <i>Fingerprint</i> pada Sistem.....	115
Gambar 5.44 Baris Kode Program Penghitungan Kemiripan.....	117
Gambar 5.45 Tampilan Hasil Perhitungan Kemiripan.....	117

DAFTAR LAMPIRAN

LAMPIRAN A. PERANCANGAN SISTEM.....	122
LAMPIRAN B. KODE PROGRAM	148
LAMPIRAN C. <i>TESTING</i>	176
LAMPIRAN D. TABEL PERHITUNGAN.....	199



BAB 1. PENDAHULUAN

Bab ini merupakan langkah awal dari penulisan tugas akhir ini. Bab ini berisi penjelasan mengenai latar belakang, rumusan masalah, tujuan dan manfaat, batasan masalah, dan sistematika penulisan.

1.1 Latar Belakang

Perkembangan teknologi yang semakin pesat saat ini tidak hanya memberikan dampak positif bagi kehidupan sehari-hari, tetapi memberikan dampak negatif yang tidak dapat dihindari. Dampak negatif yang tidak dapat dihindari pada era mudahnya bertukar dan mendapatkan informasi saat ini ialah semakin maraknya penjiplakan atau yang biasa kita kenal dengan *plagiarisme*.

Tindakan *plagiarisme* yang disengaja maupun tidak disengaja merupakan pelanggaran hak cipta, juga dapat mematikan kreatifitas seseorang karena sudah terbiasa dengan mengambil sesuatu yang bukan hasil karyanya dengan mudah. Plagiat merupakan perbuatan sengaja atau tidak sengaja dalam memperoleh atau mencoba memperoleh kredit atau nilai untuk suatu karya ilmiah, dengan mengutip sebagian atau seluruh karya dan/atau karya ilmiah pihak lain yang diakui sebagai karya ilmiahnya, tanpa menyatakan sumber secara tepat dan memadai (Republik Indonesia, 2010). Tindakan *plagiarisme* terjadi ketika siswa atau mahasiswa mendapat tugas dan dikejar waktu pengumpulan tugas. Siswa atau mahasiswa menganggap bahwa melakukan plagiat merupakan cara tercepat untuk menyelesaikan tugas-tugasnya (Universitas Pendidikan Indonesia, 2012). Hal tersebut mempengaruhi hasil evaluasi belajar siswa atau mahasiswa. Permasalahan tersebut juga berdampak terhadap para pengajar yang sulit mendeteksi *plagiarisme* mengingat tidak sedikit *file-file* tugas yang dikumpulkan.

Berdasarkan permasalahan diatas, diperlukan adanya sebuah sistem atau aplikasi yang dapat digunakan untuk mengidentifikasi adanya tindakan *plagiarisme*

pada dokumen tugas yang sebagian besar berupa dokumen berbasis teks. Beberapa metode dapat digunakan dalam mendeteksi kemiripan dokumen berbasis teks salah satunya algoritma *winnowing*. Algoritma *winnowing* merupakan algoritma dokumen *fingerprint* yang dapat mendeteksi kemiripan dokumen teks. *Input* dari algoritma *winnowing* adalah *string* dokument dan *output* berupa nilai *hash* yang digunakan sebagai *fingerprint document*. *Fingerprint* dari kedua dokumen diproses dengan fungsi *Jaccard's Coefficient Similarity* sehingga didapatkan persentase kemiripan dokumen.

Pada penelitian ini akan dibangun aplikasi pendeteksi *plagiarisme* yang menggunakan algoritma *winnowing* sebagai algoritma pencarian kemiripan dokumen. Sistem menerima *input* berupa dua dokumen teks berbahasa indonesia dengan ekstensi *.doc* dan/atau *.txt* sebagai dokumen asli serta dokumen uji yang kemudian diproses hingga dihasilkan persentase kemiripan dokumen. Persentase kemiripan dokumen asli dengan dokumen uji nantinya dibandingkan dengan persentase minimal *plagiarisme*.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, rumusan masalah yang akan diselesaikan pada penelitian ini yaitu bagaimana merancang serta membangun sistem dengan pengimplementasian algoritma *winnowing* sebagai algoritma *document fingerprint* untuk mengidentifikasi *plagiarisme* pada *file* dokumen berbasis teks.

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 dalam penelitian adalah menghasilkan aplikasi yang dapat mengidentifikasi *plagiarisme* dengan algoritma *winnowing* sebagai algoritma pencarian kemiripan dokumen.

1.3.2. Manfaat

a. Bagi Akademisi

Secara ilmiah memberikan informasi dan studi literatur bagi dunia pendidikan, khususnya di bidang Ilmu Komputer.

b. Bagi Pengguna

Aplikasi ini bermanfaat sebagai sarana untuk mengidentifikasi *plagiarisme* yang kemungkinan terjadi pada saat penulisan karya ilmiah.

c. Bagi Penulis

Meningkatkan keilmuan serta melatih kemampuan dan menerapkan ilmu pengetahuan tentang Sistem Informasi yang telah diperoleh dan sebagai media dalam menyelesaikan Tugas Akhir untuk jenjang S-1 pada Program Studi Sistem Informasi.

1.4 Batasan Masalah

Dari identifikasi yang ada di atas, diperoleh gambaran tentang permasalahan yang begitu luas. Peneliti menyadari adanya keterbatasan waktu dalam penelitian, maka diberikan batasan masalah pada obyek yang akan diteliti serta sistem yang akan dibuat supaya lebih terfokus pada inti permasalahan. Adapun batasan masalah dalam penelitian ini, antara lain :

1. *Input* dokumen dapat diterima sistem berupa dokumen jurnal berekstensi *.doc*, *.docx* dan *.txt*
2. Dokumen jurnal yang digunakan sebagai dokumen *input* harus sesuai dengan format penulisan skripsi (satu kolom).

3. Dokumen jurnal berupa dokumen berbahasa indonesia.
4. Sistem tidak memperhatikan kesalahan ejaan / penulisan pada dokumen jurnal.
5. Sistem tidak memperhatikan sinonim / persamaan kata.
6. Algoritma yang digunakan dalam sistem adalah algoritma *winnowing* berbasis *k-gram*.
7. Sistem berbasis web.

1.5 Sistematika Penulisan

1. Pendahuluan

Bab ini berisi tentang latar belakang, perumusan masalah, tujuan dan manfaat, batasan masalah, dan sistematika penulisan tugas akhir.

2. Tinjauan Pustaka

Bab ini berisi tentang kajian materi, penelitian terdahulu,, dan informasi apa saja yang digunakan dalam penelitian. Dimulai dari kajian terhadap penelitian terdahulu kemudian mengkaji pustaka mengenai *plagiarisme* sampai dengan Algoritma *winnowing*.

3. Metode Penelitian

Bab ini menguraikan tentang metode yang digunakan serta dikerjakan dalam penelitian. Metode yang dimaksud disini adalah metode perancangan dan pengembangan sistem.

4. Desain dan Pengembangan Sistem

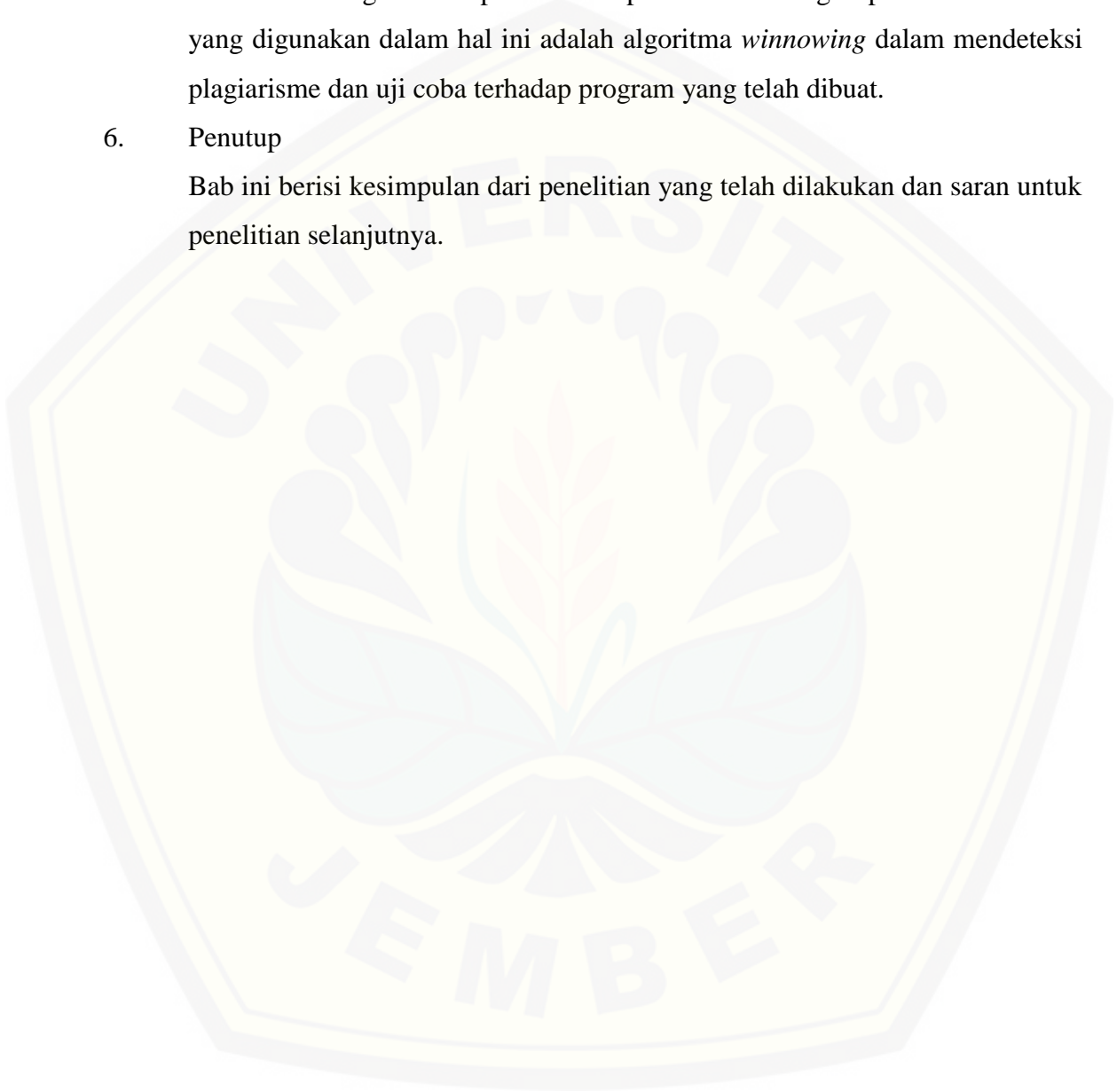
Bab ini berisi gambaran dan kebutuhan sistem secara umum yang ditampilkan dalam bentuk uraian dan gambar. Gambaran dan kebutuhan sistem meliputi kebutuhan fungsional, kebutuhan non fungsional, dan desain perancangan sistem meliputi *usecase diagram*, *scenario*, *activity diagram*, *sequence diagram*, *class diagram*, dan *entity relationship diagram* (ERD).

5. Hasil dan Pembahasan

Bab ini menjelaskan tentang hasil dan pembahasan dari penelitian yang telah dilakukan. Dengan memaparkan hasil penelitian tentang implementasi metode yang digunakan dalam hal ini adalah algoritma *winnowing* dalam mendeteksi plagiarisme dan uji coba terhadap program yang telah dibuat.

6. Penutup

Bab ini berisi kesimpulan dari penelitian yang telah dilakukan dan saran untuk penelitian selanjutnya.



BAB 2. TINJAUAN PUSTAKA

Pada bab ini dipaparkan teori-teori serta sumber pustaka yang dipakai pada saat penelitian dilakukan. Teori-teori diambil dari buku literatur, jurnal maupun website yang sesuai. Teori-teori yang dibahas adalah teori mengenai *plagiarisme*, Algoritma *winnowing*, konsep kerja algoritma *winnowing*, *Jaccard Coefficient Similarity*, dan teori-teori yang mendukung.

2.1 Penelitian Terdahulu

Penelitian terdahulu dengan judul “PERANCANGAN SISTEM DETEKSI PLAGIARISME DOKUMEN TEKS DENGAN MENGGUNAKAN ALGORITMA RABIN-KARP” dilakukan oleh Eko Nugroho mahasiswa Program Studi Ilmu Komputer, Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Brawijaya. Penelitian ini bertujuan untuk membandingkan kemiripan dokumen asli dengan dokumen yang ingin diuji dengan presentase kemiripan kedua dokumen dijadikan bahan pertimbangan apakah dokumen yang diuji merupakan hasil menjiplak karya orang seseorang atau tidak. Peneliti menggunakan algoritma *Rabin-Karp* sebagai metode pencarian pola kesamaan dalam dokumen teks. Dalam penelitian ini juga dilakukan modifikasi algoritma *Rabin-Karp* dengan menyisipkan metode *stemming* pada tahap *preprocessing*, proses *hashing*, dan pada proses *string-matching* yang kemudian dilakukan perbandingan antara algoritma *Rabin-Karp* sebelum dan sesudah dimodifikasi dari sisi waktu proses serta keakuratan dalam mendeteksi kemiripan (*similarity*) dokumen.

Berdasarkan penelitian yang telah dilakukan ada beberapa kekurangan yang dimiliki oleh algoritma *Rabin-Karp* original, yaitu waktu proses yang relatif lama terhadap file berukuran besar. Akurasi nilai *similarity* tergantung pada nilai *Kgram* yang diinputkan, semakin kecil nilai *Kgram* maka akurasi nilai *similarity* semakin tinggi. Peneliti termotivasi untuk melakukan penelitian tentang indentifikasi

plagiarisme pada file dokumen berbasis teks dengan memanfaatkan algoritma *winnowing* sebagai algoritma pencari kesamaan teks dalam dokumen, mengingat algoritma *winnowing* mampu mengurangi waktu proses pada file berukuran besar dengan memanfaatkan teknik *rolling* pada proses *hashing*. Selain itu, nilai akurasi *similarity* algoritma *winnowing* tidak hanya dipengaruhi oleh nilai *input Kgram*, tetapi juga dipengaruhi oleh nilai *input window* yang berfungsi memisahkan hasil *hash* pada setiap *gram*-nya.

2.2 *Plagiarisme*

Plagiarisme adalah tindakan penyalahgunaan, pencurian / perampasan, penerbitan, pernyataan, atau menyatakan sebagai milik sendiri sebuah pikiran ide, tulisan, atau ciptaan yang sebenarnya milik orang lain secara sengaja ataupun tidak sengaja tanpa mencantumkan rujukannya (Ridhatillah, 2003).

Menurut Kamus Besar Bahasa Indonesia (KBBI) *Plagiarisme* adalah penjiplakan atau pengambilan karangan, pendapat, seluruh dan/atau sebagainya karya orang lain dan menjadikannya seolah-olah karangan dan pendapat sendiri (Penjiplakan [Def.], 2005).

Sedangkan menurut Peraturan Pendidikan Nasional Republik Indonesia, Plagiat adalah perbuatan secara atau tidak sengaja dalam memperoleh atau mencoba memperoleh kredit atau nilai untuk suatu karya ilmiah, dengan mengutip sebagian atau seluruh karya dan/atau karya ilmiah pihak lain yang diakui sebagai karya ilmiahnya, tanpa menyatakan sumber secara tepat dan memadai (Republik Indonesia, 2010).

Barnabaum (tanpa tahun) dari Valdosta State University, menggolongkan plagiat menjadi lima jenis, yaitu :

1. “*Copy-paste*”, dalam arti menyalin kalimat atau pendapat orang lain tanpa menggunakan tanda kutip dan tanpa menyertakan sumbernya.
2. “*Word-switch*”, mengambil kalimat orang lain dan mengubah struktur kalimatnya.

3. “*Style*”, dalam arti mengikuti kalimat sumber kata demi kata dan kalimat demi kalimat.
4. “*Metafora*” dalam arti menggunakan metafora orang lain tanpa menyertakan sumbernya.
5. “*Gagasan*”, dalam arti mengambil gagasan atau pikiran orang lain tanpa menyebutkan sumbernya.

Dapat disimpulkan bahwa *plagiarisme* adalah tindakan dengan sengaja atau tidak sengaja meniru atau menjiplak kalimat, gagasan, ide, pikiran, dan/atau karya orang lain yang kemudian diakui dalam karyanya sendiri yang telah diterbitkan tanpa menyebutkan sumber atau rujukan yang tepat.

2.3 Algoritma *Stemming* Bahasa Indonesia

Stemming merupakan bagian dari proses *information retrieval*(IR), yang mengubah beberapa kata ke bentuk kata dasarnya sebelum dilakukan pengindeksan. *Stemming* bahasa indonesia tidak sama dengan *stemming* bahasa inggris. Proses *stemming* bahasa indonesia lebih rumit karena dalam bahasa indonesia terdapat imbuhan awalan (*prefixes*), sisipan(*infixes*), akhiran(*suffixes*), dan kombinasi antara awalan dan akhiran(*confixes*).

2.3.1 *Confix Stripping Stemmer*

Confix stripping (CS) *stemmer* adalah metode *stemming* untuk bahasa indonesia yang diusulkan oleh Nazief dan Adriani (Arifin, *et al*, 2009). Metode ini mengelompokkan kata imbuhan kedalam beberapa kategori sebagai berikut.

- a. *Inflection Suffixes*, merupakan akhiran yang tidak merubah kata dasar. Dibedakan menjadi dua, yaitu :
 - 1) *Particle* (P), akhiran “-lah”, “-kah”, “-tah”, dan “-pun”.
 - 2) *Possesive Pronoun* (PP), akhiran “-ku”, “-mu”, dan “-nya”.
- b. *Derivation Suffixes* (DS), merupakan akhiran yang secara langsung disisipkan pada kata dasar. Jenis akhiran ini adalah “-i”, “-kan”, dan “-an”.

- c. *Derivation Prefixes* (DP), merupakan imbuhan yang dapat secara langsung disisipkan pada kata dasar, atau lebih dari satu imbuhan bisa disisipkan pada kata dasar. Imbuhan ini dibedakan menjadi *complex prefix* (“me-”, “be-”, “pe-”, dan “te-”) dan *plain prefix* (“di-”, “ke-”, dan “se-”).

Dalam metode ini juga terdapat aturan awalan-akhiran yang tidak diperbolehkan. Kombinasi awalan-akhiran yang tidak diperbolehkan dapat dilihat pada tabel 2.1 berikut.

Tabel 2.1 Kombinasi awalan-akhiran yang tidak diperbolehkan

Awalan	Akhiran yang tidak diperbolehkan
be-	-i
di-	-an
ke-	-i, -kan
me-	-an
se-	-i, -kan
te-	-an

Sumber : (Arifin, *et al*, 2009).

Tabel 2.2 Aturan pemotongan imbuhan “me-”

Rule	Construction	Prefix Removal
1	me{l r w y}V...	me-{l r w y}V...
2	mem{b f v}...	mem-{b f v}...
3	mempe...	mem-pe...
4	mem{rV V}...	me-m{rV V}... me-p{rV V}...
5	men{c d j z}...	men-{c d j z}...
6	menV...	me-nV... me-tV
7	meng{g h q k}...	meng-{g h q k}...
8	mengV...	meng-V... meng-kV...
9	menyV...	meny-sV...
10	mempV...	mem-pV... where V != ‘e’

Sumber : (Arifin, *et al*, 2009).

Tabel 2.3 Aturan pemotongan imbuhan “pe-”

Rule	Construction	Prefix Removal
1	pe{w y}V...	pe-{w y}V...
2	perV...	per-V... pe-rV...
3	perCAP	per-CAP... where C != 'r' and P != "er"
4	perCAerV...	per-CAerV... where C != 'r'
5	pem{b f V}...	pem-{b f V}...
6	pem{rV V}...	pe-m{rV V}... pe-p{rV V}...
7	pen{c d j z}...	pen-{c d j z}...
8	penV...	pe-nV... pe-tV...
9	peng{g h q}...	peng-{g h q}...
10	pengV...	peng-V... peng-kV...
11	penyV...	peny-sV...
12	pelV...	pe-lV... except "pelajar", return "ajar"
13	peCerV...	per-erV... where C != {r w y l m n}
14	peCP...	pe-CP... where C != {r w y l m n} and P != 'er'
15	perC ₁ erC ₂ ...	per-C ₁ erC ₂ ... where C ₁ != 'r'
16	peC ₁ erC ₂ ...	pe-C ₁ erC ₂ ... where C ₁ != {r w y l m n}

Sumber : (Arifin, *et al*, 2009).

Tabel 2.4 Aturan pemotongan imbuhan “be-”

Rule	Construction	Prefix Removal
1	berV...	ber-V... be-rV...
2	berCAP...	ber-CAP... where C != 'r' and P != "er"
3	berCAerV...	ber-CaerV... where C != 'r'
4	belajar	bel-ajar
5	beC ₁ erC ₂ ...	be-C ₁ erC ₂ ... where C ₁ != { 'r' 'l' }

Sumber : (Arifin, *et al*, 2009).

Tabel 2.5 Aturan pemotongan imbuhan “te-”

Rule	Construction	Prefix Removal
1	terV...	ter-V... te-rV...
2	terCerV...	ter-CerV... where C != ‘r’
3	terCP...	ter-CP... where C != ‘r’ and P != ‘er’
4	teC1erC2...	te-C1erC2... where C1 != ‘r’

Sumber : (Arifin, *et al*, 2009).

Pada tabel 2.2 sampai 2.5, simbol C merupakan huruf konsonan, simbol V merupakan huruf vokal, simbol A merupakan huruf vokal atau konsonan, dan simbol P merupakan partikel atau fragmen dari suatu kata, misalnya “er”. Proses pemotongan imbuhan pada algoritma *confix stripping stemmer* adalah sebagai berikut :

- a. *Input* kata terlebih dahulu dicek pada kamus kata dasar. Jika *input* kata ada pada kamus kata dasar, maka *input* kata merupakan kata dasar dan proses dihentikan.
- b. Perhatikan aturan pada tabel 2.1, jika *input* kata sesuai dengan pasangan pada tabel 2.1 maka lakukan pemotongan akhiran terlebih dahulu. Jika tidak sesuai maka dilakukan pemotongan awalan terlebih dahulu.
- c. Lakukan pemotongan *particle* (P) (“-lah”, “-kah”, “-tah”, “-pun”) jika ada pada *input* kata, dan lanjutkan dengan pemotongan *possesive pronoun* (PP) (“-ku”, “-mu”, “-nya”).
- d. Lakukan pemotongan *derivation suffixes* (DS) (“-i”, “-kan”, atau “-an”).
- e. Lakukan pemotongan *derivation prefixes* (DP) (“di-”, “ke-”, “se-”, “me-”, “be-”, “pe”, “te-”) dengan iterasi maksimal tiga kali :
 - 1) Proses dihentikan jika :
 - a) Imbuhan yang ditemukan sama seperti kombinasi awalan-akhiran pada tabel 2.1 yang sudah dilakukan pemotongan pada tahap sebelumnya.
 - b) Imbuhan yang ditemukan sama seperti imbuhan sebelumnya yang sudah dilakukan pemotongan, atau
 - c) Ketiga imbuhan sudah dilakukan pemotongan.

- 2) Cek tipe imbuhan dan lakukan pemotongan. Imbuhan dibedakan menjadi dua tipe, yaitu :
 - a) *Plain*: imbuhan “di-”, “ke-”, dan “se-” yang bisa langsung dilakukan pemotongan.
 - b) *Complex*: imbuhan “me-”, “be-”, “pe”, dan “te-”. Gunakan aturan pemotongan pada seperti pada tabel 2.2, tabel 2.3, tabel 2.4 atau tabel 2.5.
- f. Cek *input* kata yang sudah dilakukan proses pemotongan sampai tahap 5. Jika belum ditemukan pada kamus kata dasar, maka imbuhan dikembalikan dan *input* kata dianggap sebagai kata dasar.

2.3.2 Enhanced Confix Stripping Stemmer

Enhanced confix stripping (ECS) *stemmer* adalah metode *stemming* yang merupakan penyempurnaan dari *confix stripping* (CS) *stemmer*. Pada algoritma ini dilakukan modifikasi aturan pada tabel 2.2 dan table 2.3. Modifikasi aturan yang terdapat pada algoritma *enchanced confix stripping stemmer* dapat dilihat pada tabel 2.6 dan tabel 2.7 berikut.

Tabel 2.6 Modifikasi aturan pada Tabel 2.2

Rule	Construction	Prefix Removal
5	men{c d j s z}...	men-{c d j s z}...
6	mengV...	meng-V... meng-kV... (mengV-... if V='e')
10	mempA...	mem-pA... where A!= 'e'

Sumber : (Arifin, *et al*, 2009).

Tabel 2.7 Modifikasi aturan pada Tabel 2.3

Rule	Construction	Prefix Removal
9	pengC...	peng-C...
10	pengV...	peng-V... peng-kV... (pengV-... if V='e')

Sumber : (Arifin, *et al*, 2009).

Penyempurnaan pada algoritma ini ada dua, yaitu :

- a) Modifikasi beberapa aturan pada tabel 2.2 dan tabel 2.3, jadi *stemming* pada kata dengan format “mem+p...”, “men+s...”, “menge+...”, “penge+...”, dan “peng+k...” dapat diselesaikan.
- b) Penambahan tahap *stemming* untuk menyelesaikan proses pemotongan akhiran. Penambahan tahap *stemming* dinamakan dengan proses *loopPengembalianAkhiran* yang dijalankan ketika proses pada CS *stemmer* selesai dijalankan tapi tetap tidak ditemukan kata dasar pada kamus kata dasar. Pada setiap tahap dalam proses *loopPengembalianAkhiran* dilakukan pengecekan *output stemming* pada kamus data. Proses *loopPengembalianAkhiran* adalah sebagai berikut:
 - 1) Kembalikan seluruh awalan yang telah dilakukan pemotongan, sehingga dihasilkan model kata seperti : $[DP+[DP+[DP]]] + \text{Kata Dasar}$. Lakukan pengecekan pada kamus kata dasar, jika ditemukan maka proses berhenti.
 - 2) Kembalikan akhiran yang telah dilakukan pemotongan dengan urutan DS (“-i”, “-kan”, “-an”) terlebih dahulu, kemudian PP (“-ku”, “-mu”, “-nya”), dan yang terakhir adalah P (“-lah”, “-kah”, “-tah”, “-pun”). Pengembalian akhiran khusus DS dengan akhiran “-kan”, karakter huruf “k” dikembalikan terlebih dahulu. Jika masih belum ditemukan kata dasar pada kamus kata dasar, maka dilanjutkan dengan pengembalian “an”.
 - 3) Lakukan pengecekan ke kamus kata dasar. Apabila kata dasar ditemukan, maka proses dihentikan. Jika gagal, maka lakukan proses pemotongan

awalan berdasarkan aturan pada tabel 2.2(yang sudah dimodifikasi), tabel 2.3 (yang sudah dimodifikasi), tabel 2.4, dan tabel 2.5.

- 4) Lakukan pengecekan pada kamus kata dasar, apabila tetap gagal ditemukan kata dasar, maka awalan-awalan yang telah dilakukan pemotongan dikembalikan lagi.

2.4 Algoritma *Winnowing*

Algoritma *Winnowing* adalah metode pencarian kesamaan (*similarity*) kata dalam dokumen dengan cara membandingkan *fingerprint* dari dokumen (Cornic, 2008), *input* dari algoritma ini adalah dokumen teks yang kemudian diproses sehingga menghasilkan *output* berupa nilai-nilai *hash*. Kumpulan nilai *hash* tersebut selanjutnya disebut *fingerprint*. *Fingerprint* inilah yang digunakan untuk membandingkan kemiripan (*similarity*) antar dokumen teks.

2.4.1 Konsep Kerja Algoritma *Winnowing*

Langkah kerja algoritma *winnowing* untuk mendeteksi kemiripan dokumen diperlukan kebutuhan yang mendasar. Kebutuhan mendasar yang harus dipenuhi oleh algoritma pendeteksi menurut Shleimer, Wilkerson, & Aiken (2003), yaitu :

- a. *Whitespace insensitivity* yaitu pencarian kalimat tidak terpengaruh oleh spasi, tanda baca, jenis huruf (kapital atau normal), dan sebagainya.
- b. *Noise surpression* yaitu menghindari penemuan dengan panjang kata yang terlalu kecil dan bukan kata umum yang digunakan seperti 'the'.
- c. *Position independence* yaitu penemuan kemiripan harus tidak bergantung pada posisi kata-kata sehingga kata dengan urutan berbeda masih bisa dikenali jika terjadi kesamaan.

Secara umum prinsip kerja dari algoritma pendeteksian kemiripan dokumen adalah dengan tahapan sebagai berikut :

- a. Teks yang akan dicari ataupun yang asli diasumsikan sebagai *string s* yang panjangnya *t*

- b. Melakukan pembersihan dari tanda baca, spasi dan sebagainya dengan mengacu pada kebutuhan dasar algoritma pendeteksi
- c. Membagi dokumen menjadi *k-gram*, dimana *k* merupakan parameter yang dipilih pengguna. *K-gram* merupakan substring yang berdampingan dengan panjang karakter *k*.
- d. Mencari nilai *hash* dari setiap *k-gram*.
- e. Memilih beberapa hasil *hash* menjadi dokumen fingerprint.

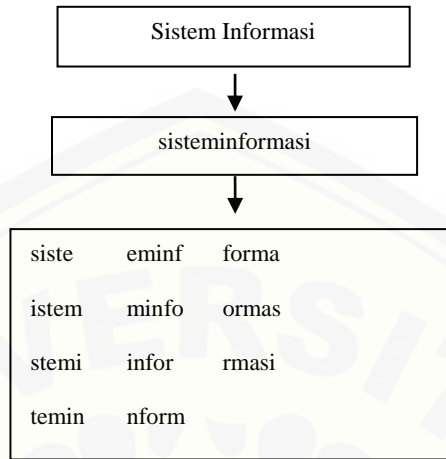
Pembeda Algoritma *winnowing* dengan algoritma pendeteksi kemiripan dokumen yang lain terletak pada proses pemilihan *document fingerprint*. Hasil perhitungan nilai *hash* dibagi kedalam *window w* yang kemudian dipilih nilai paling kecil dari setiap *window* untuk dijadikan *document fingerprint*. Jika terdapat nilai minimum lebih dari satu, maka nilai minimum dipilih salah satu saja.

Propertis dalam pencarian kemiripan dokumen menggunakan algoritma *winnowing* perlu memperhatikan hal berikut :

- a. Jika terdapat string yang sama panjang dengan panjang *t*, dimana *t* merupakan nilai ambang (*guarantee threshold*) yang ditentukan, maka kemiripan terdeteksi
- b. Algoritma tidak dapat melakukan pencocokan jika lebih pendek dari gangguan nilai ambang (*noise threshold*) *k*.
- c. Konstanta *t* dan $k \leq t$ dipilih oleh *user* dengan menghindari pencocokan *string* yang sama dibawah nilai *noise threshold*.

2.4.2 *K-Gram*

Dokumen yang telah dilakukan pembersihan dengan mengacu pada kebutuhan mendasar dilanjutkan dengan membentuk dokumen kedalam rangkaian *k-gram*, dimana nilai *k* ditentukan oleh *user*. *K-gram* adalah rangkaian *substring* yang bersebelahan dengan panjang *k* (Scheleimer, Wilkerson, & Aiken, 2003). Proses pembentukan kata kedalam rangkaian *gram* dapat dilihat pada gambar 2.1.



Gambar 2.1 Hasil pembuatan k-gram terhadap teks dengan nilai k = 5

2.4.3 Rolling Hash

Algoritma *Winnowing* menggunakan fungsi *Rolling Hash* untuk menghasilkan nilai *hash* dari rangkaian *gram* yang sudah dibuat. Fungsi *hash* didefinisikan pada persamaan 1.

$$H_{c1...ck} = c_1 * b^{(k-1)} + c_2 * b^{(k-2)} + \dots + c_{(k-1)} * b + c_k \dots\dots\dots(1)$$

Keterangan :

c : nilai ASCII karakter

b : basis bilangan prima

k : nilai *k-gram* / banyak karakter pada gram

Rolling Hash memiliki keuntungan untuk mencari nilai hash $H_{(c2...ck+1)}$ dengan memanfaatkan nilai yang diperoleh pada hash $H_{c1...ck}$. Fungsi hash $H_{(c2...ck+1)}$ dapat dilakukan dengan persamaan 2.

$$H_{(c2...ck+1)} = (H_{(c1...ck)} - c_1 * b^{(k-1)}) * b + c_{(k+1)} \dots\dots\dots(2)$$

Pada perhitungan *hash* dari *gram* ke-*n*, nilai *hash gram n-1* dikurang dengan nilai karakter pertama dari *gram n-1* kemudian ditambahkan dengan nilai karakter terakhir dari *gram* ke-*n*. Proses ini menghemat waktu komputasi saat menghitung nilai *hash*

dari sebuah *gram*. Hasil perhitungan nilai *hash* pada *gram* di Gambar 2.1 dapat dilihat pada Gambar 2.2

13634 13066 13788 13529 12501 13071 12840 13114 12709 13456
13500

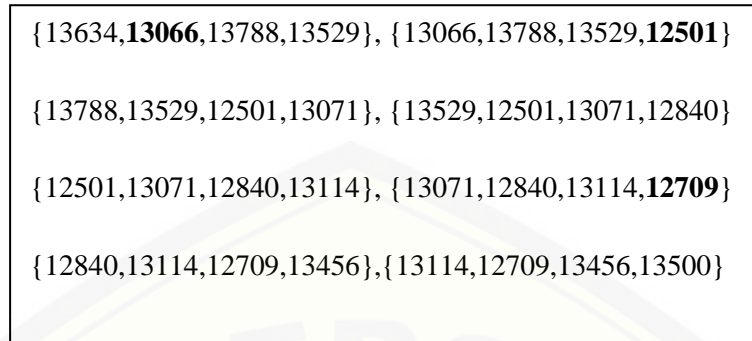
Gambar 2.2 Nilai *hash* dari setiap *gram*

2.4.4 Document Fingerprint

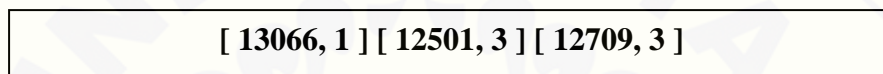
Fingerprint adalah teknik yang bertujuan untuk mencegah penyalinan secara tidak sah dari suatu konten digital. *Fingerprint* tidak mudah terdeteksi karena dirancang dengan cara yang membuat suatu konten digital sulit dipalsukan (van Tilborg & Shushil, 2011).

Document Fingerprint merupakan metode yang dapat digunakan untuk mendeteksi kemiripan dokumen. Prinsip kerja dari metode ini adalah mengkonversi setiap *string* menjadi bilangan dengan menggunakan teknik *hashing*. Nilai *hashing* inilah yang nantinya digunakan sebagai *fingerprint* atau identitas sebuah dokumen.

Algoritma *winnowing* menggunakan *fingerprint* sebagai kata kunci yang dijadikan acuan untuk mencari kemiripan dengan dokumen yang diuji. Nilai *hash* dokumen dibagi dengan menggunakan *window w* sebelum menentukan *fingerprint* dari dokumen asli maupun dari dokumen uji. Hasil dari pembuatan *window w* dipilih nilai minimum sebagai *fingerprint*. Contoh pembuatan *window w* serta pemilihan *fingerprint* dapat dilihat pada Gambar 2.3 dan 2.4



Gambar 2.3 Pembentukan *window* $w = 4$



Gambar 2.4 *Fingerprint* diambil dari nilai *hash* terkecil

2.5 Perhitungan Kemiripan (*similarity*)

Jaccard Coefficient Similarity merupakan standart pengukuran kesamaan yang membandingkan dua set, P dan Q (Naumann & Melanie, 2010). Bentuk umum rumus *Jaccard's Coefficient Similarity* dapat dilihat pada persamaan 3.

$$Jacc(P, Q) = \frac{P \cap Q}{P \cup Q} \dots\dots\dots(3)$$

Jaccard(P, Q) adalah nilai *similarity*, $|P \cap Q|$ adalah jumlah dari *fingerprints* yang sama dari dokumen asli dan dokumen uji, serta $|P \cup Q|$ adalah jumlah *fingerprint* P yang tidak dimiliki Q ditambahkan jumlah *fingerprint* Q yang tidak dimiliki P ditambahkan jumlah *fingerprint* yang sama antara P dan Q.

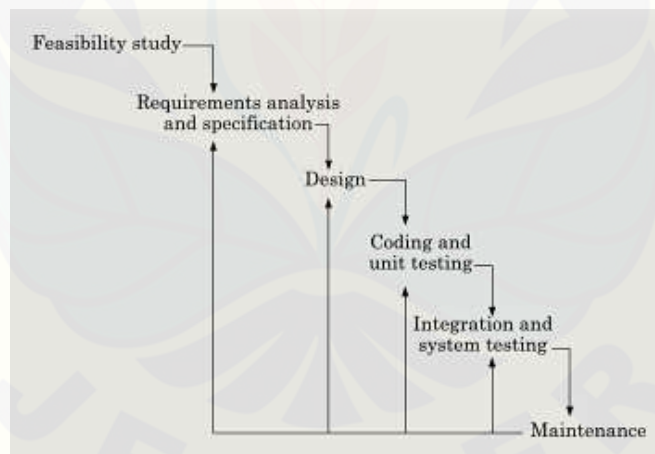
Penentuan jenis *plagiarisme* berdasarkan persentase kemiripan (*similarity*) menurut Mutiara & Agustina (2008) ada 5 jenis, yaitu :

1. Persentase kemiripan dokumen asli dengan dokumen uji menunjukkan nilai 0% berarti kedua dokumen berbeda baik dari segi isi maupun dari segi kalimat
2. Persentase kemiripan dokumen asli dengan dokumen uji menunjukkan nilai <15% berarti kedua dokumen hanya memiliki sedikit kesamaan

3. Persentase kemiripan dokumen asli dengan dokumen uji menunjukkan nilai antara 15%-50% berarti kedua dokumen terindikasi plagiat tingkat sedang
4. Persentase kemiripan dokumen asli dengan dokumen uji menunjukkan nilai >50% berarti dokumen terindikasi plagiat tingkat tinggi
5. Persentase kemiripan dokumen asli dengan dokumen uji menunjukkan nilai 100% berarti kedua dokumen sama persis baik dari segi isi maupun kalimat.

2.6 Model Waterfall

Iterative waterfall model (Mall, 2009) merupakan perubahan dari model *classic waterfall*. Model *iterative waterfall* memiliki keunggulan pada tersedianya alur *feedback* dari setiap tahap ke tahap sebelumnya apabila terjadi kesalahan dan memungkinkan koreksi kesalahan tersebut. Alur *iterative waterfall model* dapat dilihat pada gambar 2.5



Gambar 2.5 *Iterative waterfall model* (Mall, 2009)

Berdasarkan gambar 2.5 diperoleh keterangan tentang *iterative waterfall model* sebagai berikut :

2.6.1 *Requirement analysis and specification*

Tahap *requirement analysis and specification* merupakan tahap mempelajari kebutuhan yang diperlukan oleh sistem sebelum sistem dibangun atau dikembangkan. Tujuan utama tahap ini adalah mendokumentasikan kebutuhan yang diperlukan untuk merancang atau mengembangkan sistem (Fatta, 2005). Ada lima aktivitas utama dalam fase ini, yaitu:

a. Pengumpulan Informasi

Proses pengumpulan informasi tentang proses-proses pada sistem yang lama serta mengidentifikasi kelemahan-kelemahan dari sistem lama yang telah berjalan. Hasil pengumpulan informasi digunakan sebagai acuan untuk membuat sistem baru ataupun mengembangkan sistem yang telah ada.

b. Mendefinisikan Sistem *Requirement*

Hasil pengumpulan informasi dianalisis serta didefinisikan kebutuhan-kebutuhan dalam pembuatan sistem. Tahap ini akan merubah total keseluruhan proses bisnis dalam sistem lama atau hanya perubahan penambahan beberapa proses bisnis baru.

c. Memprioritaskan Kebutuhan

Kebutuhan yang sudah didefinisikan dalam tahap sebelumnya umumnya sangat lengkap. Ketersediaan waktu dan sumber daya untuk menyelesaikan sistem bisa saja tidak mencukupi, sehingga untuk memaksimalkan pembuatan sistem analis akan memprioritaskan kebutuhan-kebutuhan yang dianggap kritis untuk didahulukan.

d. Menyusun dan Mengevaluasi Alternatif

Tahap ini memungkinkan analis membuat rencana alternatif setelah menyusun dan memprioritaskan kebutuhan. Analis harus menyiapkan alternatif jika seandainya susunan kebutuhan nantinya akan ditolak oleh klien dikarenakan tidak sesuai dengan keinginan klien.

e. Mengulas Kebutuhan dengan Pihak Manajemen

Langkah terakhir adalah mengulas kebutuhan-kebutuhan yang sudah dengan pihak klien karena pihak klien lah yang paling tahu kebutuhan akan sistem mereka.

2.6.2 Design

Desain adalah tahapan mengubah kebutuhan yang masih berupa konsep menjadi sistem yang riil. Tahapan desain sistem dapat dibagi menjadi 2 tahap, yaitu desain logis (*logical design*) dan desain fisik (*physical design*). Tahapan desain logis menghasilkan beberapa dokumen, diantaranya dokumen *business process*, *use case diagram*, *system scenario*, *activity diagram*, *sequence diagram*, serta *class diagram*. Tahapan desain fisik biasanya menghasilkan desain *interface* program, *prototype* program, dan lain sebagainya.

2.6.3 Coding and Unit Testing

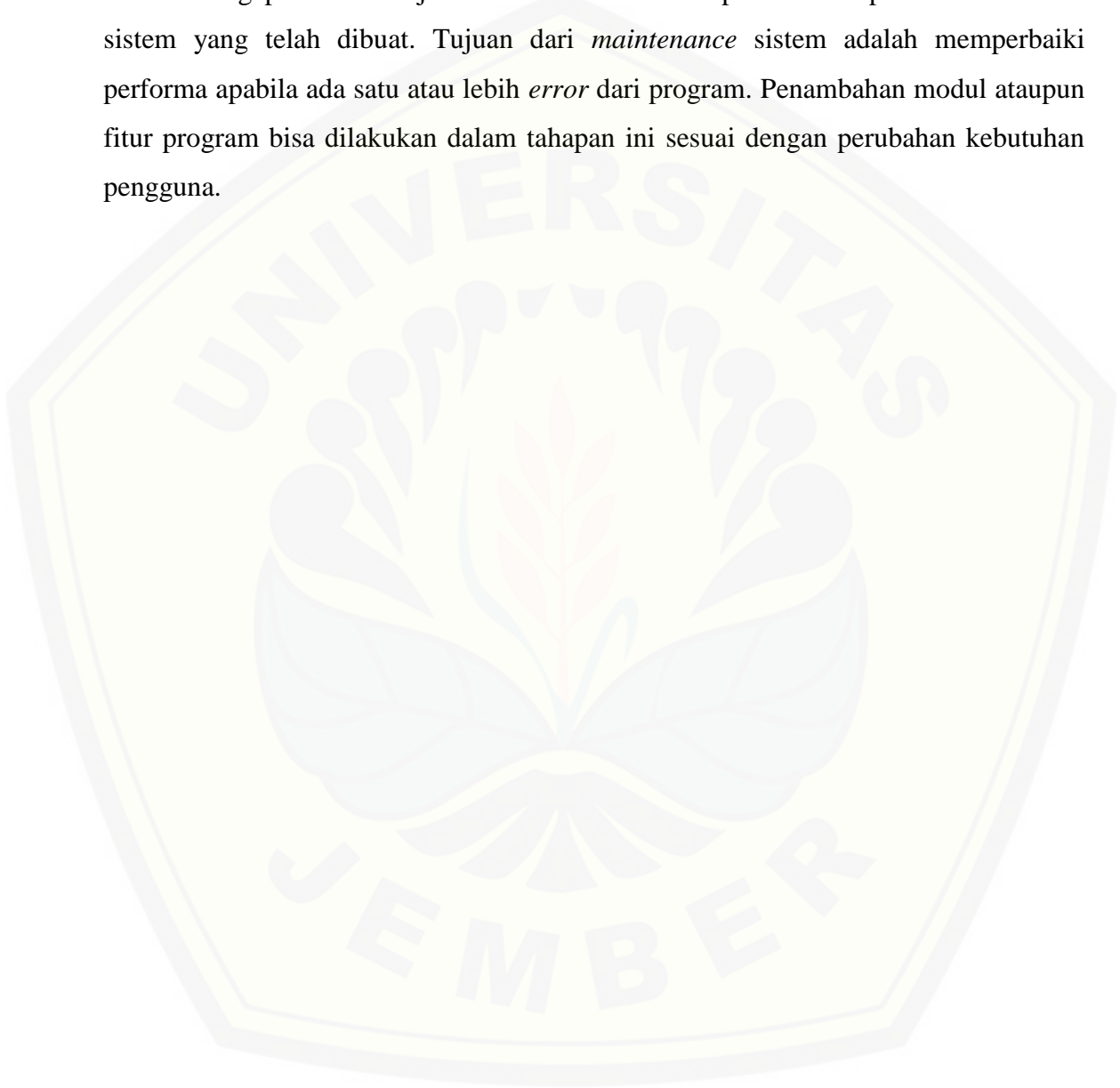
Coding merupakan tahapan pengimplementasian desain yang telah dibuat ke dalam kode program. Tahap *coding* biasanya menggunakan bahasa pemrograman yang dimengerti oleh komputer. Beberapa bahasa pemrograman yang biasanya digunakan antara lain *Java*, *Hypertext Preprocessor* (PHP), *C++*, dan lain sebagainya.

2.6.4 Integration and System Testing

Testing merupakan tahapan menguji hasil kode program yang telah dihasilkan dari tahap implementasi. Tujuan pengujian ada dua, dari sisi pengembang harus menjamin kode program yang dibuat harus bebas dari kesalahan sintaks maupun logika (*white box*) dan dari sisi klien harus bisa menyelesaikan masalah yang ada pada klien dan sistem yang dibuat harus mudah dijalankan dan dipahami oleh pengguna akhir (*black box*).

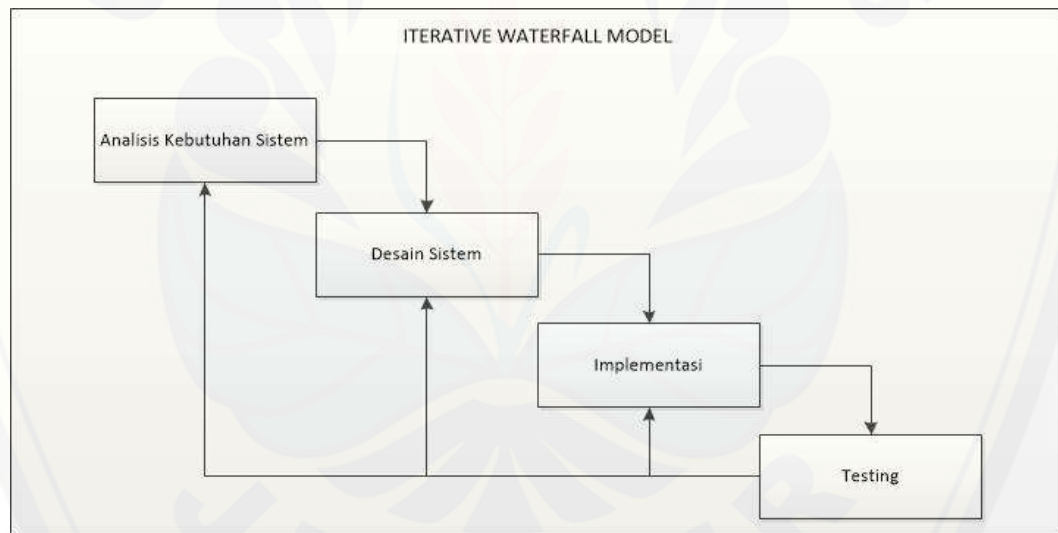
2.6.5 *Maintenance*

Tahapan pemeliharaan adalah tahapan dimana sistem secara berkala dimonitoring performa kerja sistem. Hasil dari tahapan ini berupa versi baru dari sistem yang telah dibuat. Tujuan dari *maintenance* sistem adalah memperbaiki performa apabila ada satu atau lebih *error* dari program. Penambahan modul ataupun fitur program bisa dilakukan dalam tahapan ini sesuai dengan perubahan kebutuhan pengguna.



BAB 3 METODE PENELITIAN

Pada bab ini menjelaskan tentang metode yang digunakan dalam perancangan dan pengembangan sistem. Tahap-tahap perancangan dan pengembangan sistem pendeteksi *plagiarisme* menggunakan *software development life cycle* (SDLC) dengan mengadopsi model *iterative waterfall model* sesuai dengan yang dijelaskan pada tinjauan pustaka. Implementasi dengan model ini mempunyai kelebihan prosesnya teratur dan terjadwal pengerjaan program lebih menentu serta adanya alur *feedback* yang memungkinkan mengoreksi kesalahan pada tahap sebelumnya dan memperbaikinya. Tahapan-tahapan yang dimaksud dapat dilihat pada gambar 3.1



Gambar 3.1 Diagram *iterative waterfall model*

Keterangan dari gambar 3.1 tentang diagram *iterative waterfall model* yang akan digunakan pada penelitian ini adalah sebagai berikut :

3.1 Analisis Kebutuhan Sistem

Tahapan awal dalam perancangan dan pengembangan sistem adalah menentukan kebutuhan-kebutuhan sistem pendeteksi *plagiarisme* yang akan

dibangun. Pada tahapan ini, penulis melakukan studi literatur yang mempelajari tentang sistem informasi *retrieval* dan metode *document fingerprint* dengan algoritma *winning* untuk mengidentifikasi kemiripan dokumen teks melalui berbagai media, antara lain melalui internet, jurnal-jurnal, dan buku yang berhubungan dengan *plagiarisme* dan *text processing* untuk mendapatkan gambaran tentang algoritma *winning* yang akan digunakan pada sistem sebagai algoritma pencarian kemiripan dokumen serta data apa saja yang dibutuhkan untuk membangun sistem tersebut.

3.1.1 Algoritma *Winning* sebagai algoritma pencarian kemiripan dokumen

Perancangan dan pengembangan sistem pendeteksi *plagiarisme* pada penelitian ini menggunakan algoritma *winning* sebagai algoritma pencarian kemiripan dokumen. Pemilihan algoritma ini berdasarkan tingkat akurasi serta kecepatan waktu proses dengan mengacu pada penelitian terdahulu yang sudah dijelaskan pada tinjauan pustaka. Alur dari proses algoritma *winning* yang akan diimplementasikan pada sistem dapat dilihat pada gambar 3.2 berikut



Gambar 3.2 Alur pembentukan *fingerprint* dengan algoritma *winning*

Tahap pertama dalam proses pembentukan *fingerprint* setelah jurnal diinputkan adalah tahap *Text Preprocessing*. Pada proses ini dilakukan tiga proses yaitu proses *parsing* jurnal, proses *stemming* jurnal, dan proses *filtering* jurnal.

Proses *parsing* jurnal bertujuan untuk mengekstrak dokumen jurnal dan didapatkan konten dari jurnal yang ingin dibuat *fingerprint*. *Parsing* jurnal hanya memfokuskan pada jurnal yang berekstensi .txt, .doc, dan .docx serta konten dari jurnal yang di ekstrak hanya konten teks tanpa menyertakan gambar dan tabel.

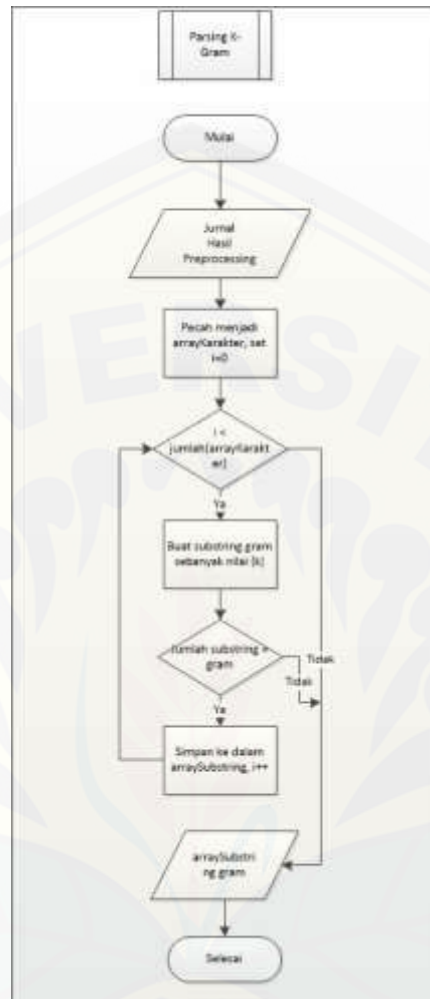
Proses *stemming* jurnal memproses tiap kata dalam jurnal yang telah melalui *parsing* jurnal dan diubah ke dalam kata dasar. *Input* dari proses ini adalah hasil ekstrak jurnal dan *output* berupa konten jurnal yang telah diproses dalam proses *stemming*. Selanjutnya dilakukan proses *filtering* jurnal yang bertujuan untuk menghilangkan tanda baca dan angka dalam konten jurnal yang telah melewati proses *stemming*. Alur proses *text preprocessing* dapat dilihat pada gambar 3.2 berikut





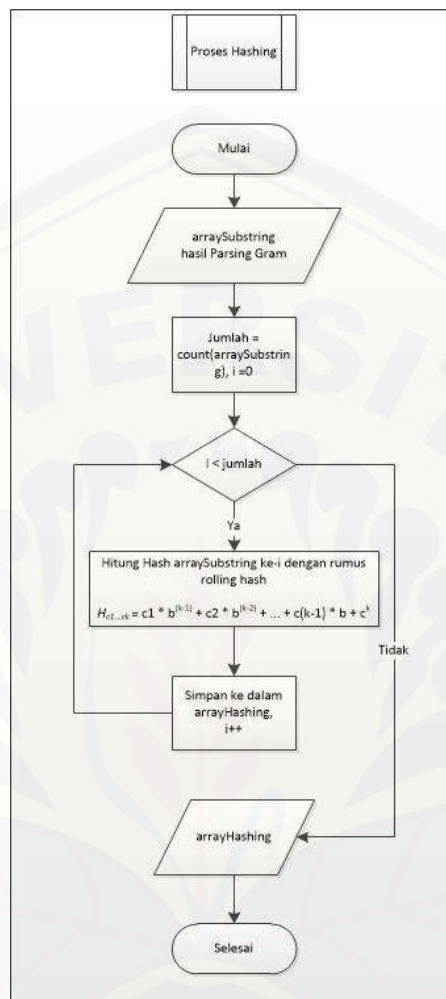
Gambar 3.3 Alur proses *text preprocessing*

Output dari proses *text preprocessing* selanjutnya digunakan sebagai data input pada proses *parsing k-gram*. Nilai k untuk proses *parsing k-gram* ditetapkan oleh penulis guna mempermudah pengguna sistem dalam menjalankan sistem. Tahap pertama dalam proses *parsing k-gram* adalah memecah *string* hasil *text preprocessing* menjadi *array* tiap karakter dan digunakan perulangan (*looping*) untuk membentuk *substring-substring gram* sebanyak nilai k . Jika, terdapat *substring gram* kurang dari nilai k , maka *substring gram* tersebut tidak dimasukkan dalam *array substring gram*. *Output* dari proses ini adalah *arraySubstring* yang nantinya digunakan pada proses *convert* nilai *hashing*. Alur proses *parsing k-gram* dapat dilihat pada gambar 3.4 berikut



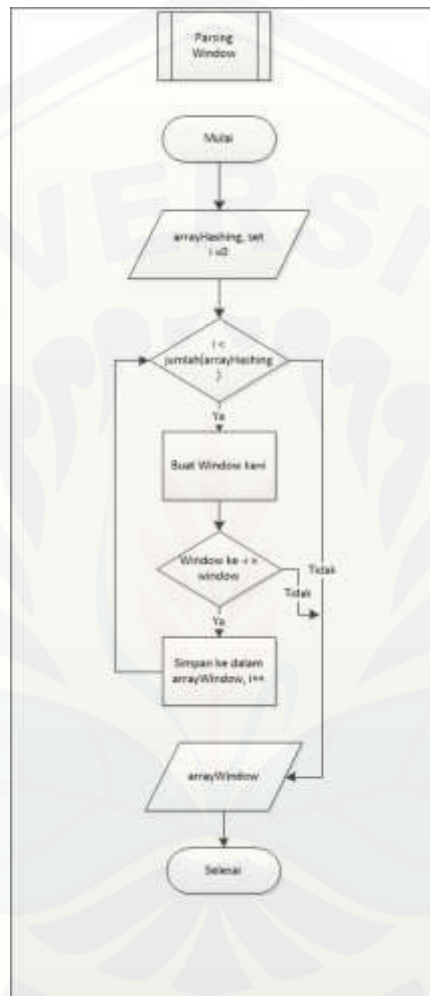
Gambar 3.4 Alur proses *parsing k-gram*

Proses selanjutnya setelah didapatkan *array substring* dari proses *parsing k-gram* adalah proses pengkonversian tiap *substring gram* kedalam nilai *hash*. Pada proses ini digunakan metode *rolling hash* untuk mempercepat waktu komputasi. *Output* dari proses ini adalah *array* nilai *hashing* yang nantinya akan digunakan sebagai data *input* pada proses pembentukan *window w*. Alur pengkonversian tiap *substring gram* kedalam nilai *hash* dapat dilihat pada gambar 3.5 berikut

Gambar 3.5 Alur proses konversi nilai *hash*

Proses selanjutnya setelah proses pengkonversian tiap *substring gram* kedalam nilai *hash* adalah proses pembentukan *window w*. Pembentukan *window* bertujuan untuk membagi *array* nilai *hash* kedalam *window* sebanyak *w* dimana nilai *w* sudah ditetapkan oleh penulis dan ditetapkan sebagai *variabel* di dalam sistem untuk mempermudah pengguna dalam menjalankan sistem. Sama halnya dengan pembentukan *k-gram*, pembentukan *window w* menggunakan perulangan (*looping*) dan apabila ada jumlah sebuah *window* tidak sama dengan nilai *w* maka *window*

tersebut tidak disimpan. *Output* proses pembentukan *window* berupa *array window*. Alur pembentukan *window w* dapat dilihat pada gambar 3.6 berikut.



Gambar 3.6 Alur proses pembentukan *window w*

Proses terakhir pada algoritma *winnowing* adalah pembentukan *fingerprint* dokumen. *Fingerprint* dokumen yang terpilih merupakan nilai terkecil pada tiap *window* dari *array window* dan apabila ada nilai *fingerprint* terpilih yang sama, maka dipilih salah satu untuk mewakili *fingerprint* terpilih. Nilai-nilai *fingerprint* terpilih disimpan dalam *array fingerprint* yang kemudian digunakan sebagai identitas sebuah dokumen jurnal guna dibanding terhadap jurnal yang lain untuk mendapatkan nilai

kemiripan dokumen tersebut. Nilai kemiripan dokumen tersebut digunakan sebagai acuan untuk menentukan tingkat *plagiarisme*. Alur pembentukan *fingerprint* dokumen dapat dilihat pada gambar 3.7 berikut



Gambar 3.7 Alur proses pembentukan *fingerprint*

3.1.2 Data yang dibutuhkan sistem

Data yang digunakan untuk membangun sistem pendeteksi *plagiarisme* ini adalah data jurnal. Data jurnal dalam penelitian ini diperoleh dari tempat penelitian yaitu Program Studi Sistem Informasi Universitas Jember. Jurnal yang digunakan

sebagai bahan pengujian dipilih 10 jurnal acak yang diunduh dari repositori sistem informasi manajemen skripsi Program Studi Sistem Informasi Universitas Jember. Data jurnal pada sistem akan terus diperbarui baik oleh admin maupun mahasiswa sebagai user non admin. Akan tetapi, jurnal yang diunggah oleh mahasiswa diharuskan mendapat *approve* dari admin sebelum disimpan dalam *database* dan digunakan sebagai referensi oleh mahasiswa lain pada pengecekan selanjutnya guna menghindari kerusakan data jurnal pada *database*. *Database* untuk data jurnal dibagi atas 2 tabel, yaitu :

a. Tabel Bidang Studi

Tabel Bidang Studi merupakan tabel yang menyimpan data kategori bidang studi yang berfungsi untuk membagi jurnal sesuai kategori bidangnya.

b. Tabel Jurnal

Tabel Jurnal merupakan tabel yang menyimpan data jurnal baik detail tentang jurnal serta data *fingerprint* hasil dari proses pengolahan dengan algoritma *winning*.

3.2 Desain Sistem

Tahapan desain sistem yang akan dibangun menggunakan *Unified Modeling Language* (UML) yang mendukung konsep pemodelan *programming* berbasis *Object Oriented Programming* (OOP) seperti yang akan diterapkan pada tahap penulisan kode program. Pada tahap ini akan diperoleh dokumentasi pemodelan, antara lain: *Business Process*, *Use Case Diagram*, *Use Case Scenario*, *Sequence Diagram*, *Activity Diagram*, *Class Diagram*, dan *Entity Relationship Diagram* (ERD).

3.3 Implementasi

Tahapan implementasi adalah proses konversi desain sistem ke dalam kode-kode program. Sistem yang akan dibangun ditulis dengan bahasa pemrograman *Page Hypertext Pre-Processor* (PHP) dengan memanfaatkan *framework CodeIgniter* yang sudah menerapkan konsep *Object Oriented Programming* (OOP). Sistem ini juga

menggunakan *local server* dan *database* untuk menyimpan data yang sewaktu-waktu dibutuhkan untuk diakses kembali. *Local Server* menggunakan aplikasi XAMPP yang sudah support *apache* untuk membangun aplikasi berbasis web dan *database* yang digunakan adalah MySql (PhpMyadmin).

3.4 Testing

Tahap selanjutnya setelah proses implementasi selesai dikerjakan adalah tahap *testing* atau pengujian sistem. Pada penelitian ini dilakukan 2 metode pengujian sistem yaitu *white box testing* dan *black box testing*. *White Box Testing* merupakan pengujian pada modul pengkodean program untuk menjamin kode program bebas dari kesalahan sintaks maupun logika. *Black Box Testing* merupakan pengujian yang menekankan pada pengujian fungsionalitas sistem agar keluaran sesuai dengan apa yang diharapkan pengguna.

BAB 4 DESAIN DAN PENGEMBANGAN SISTEM

Pada bab ini menguraikan tentang proses desain dan pengembangan sistem pendeteksi *plagiarisme* dengan menggunakan algoritma *winnowing* sebagai algoritma pencari kemiripan dokumen. Tahap pengembangan sistem dilaksanakan berdasarkan *software development life cycle* (SDLC) dengan mengadopsi model *iterative waterfall model*, dimulai dari analisis kebutuhan fungsional dan kebutuhan non-fungsional sistem, pembuatan desain sistem, implementasi desain sistem kedalam kode program dan pengujian sistem.

4.1 Deskripsi Umum Sistem

Deskripsi umum dari sistem pendeteksi *plagiarisme* dengan algoritma *winnowing* sebagai algoritma pencarian kesamaan dokumen yang dibangun dalam penelitian ini akan dijelaskan pada bagian *statement of purpose* (SOP) dan fungsi sistem.

4.1.1 *Statement of Purpose* (SOP)

Sistem pendeteksi *plagiarisme* pada *file* dokumen berbasis teks merupakan sistem yang digunakan untuk memberikan informasi tentang sejauh mana persentase plagiat pada jurnal yang kita buat terhadap daftar rujukan yang kita gunakan. Tujuan dibangun sistem ini untuk mencegah tindak plagiat baik yang disengaja maupun tidak sengaja kita lakukan sebelum jurnal kita dipublikasikan. Tingkat plagiat berdasarkan persentase kemiripan dokumen jurnal kita terhadap jurnal rujukan dibagi menjadi 5 yaitu :

- a. 0 %, berarti jurnal kita berbeda baik dari segi kata maupun kalimat dengan jurnal rujukan.
- b. <15 %, berarti jurnal kita hanya memiliki sedikit kesamaan dengan jurnal rujukan.

- c. 15% - 50%, berarti jurnal kita terindikasi plagiat tingkat sedang terhadap jurnal rujukan.
- d. >50%, berarti jurnal kita terindikasi plagiat tingkat tinggi terhadap jurnal rujukan.
- e. 100%, berarti jurnal kita sama persis dengan jurnal rujukan.

4.1.2 Fungsi-fungsi dalam sistem

Sistem yang dibangun dalam penelitian ini memberikan batasan hak akses. Fungsi yang mengatur hak akses pengguna dilakukan pada fitur *login* yang mengharuskan pengguna untuk memasukkan *username* dan *password*. Pengguna dalam sistem ini dibagi menjadi dua yang meliputi :

a. Admin

Admin memiliki hak akses penuh pada sistem. Dengan kata lain, admin merupakan pengguna utama yang bertugas manajemen data yang ada di dalam sistem seperti data jurnal dan data user. Admin juga berhak melihat *log* aktifitas user pada sistem guna mencegah hal yang tidak diinginkan seperti pengrusakan data pada sistem. Tugas utama admin terletak pada fitur manajemen data jurnal. Admin berhak untuk melakukan *insert*, *update*, *delete* data jurnal guna melengkapi data jurnal yang nantinya akan digunakan oleh *user* non-admin. Admin juga berhak menentukan apakah jurnal yang nantinya diunggah oleh *user* non-admin bisa disetujui untuk disimpan dalam *database* guna dijadikan data jurnal rujukan pada proses pengecekan oleh *user* non-admin.

b. *User* non-admin (mahasiswa)

User non-admin dalam penerapannya adalah mahasiswa yang ingin menggunakan sistem untuk mengetahui tingkat plagiat terhadap jurnal yang dibuat. Fitur utama *user* non-admin pada sistem ini adalah pengecekan tingkat plagiat jurnal yang dibuat. Mahasiswa yang ingin menggunakan sistem harus melakukan *login* ke sistem dengan memasukkan *username* dan *password* sebelum bisa menggunakan fitur pada sistem. Mahasiswa yang belum memiliki

username dan *password* bisa melakukan *registrasi* pada sistem guna mendapatkan *username* dan *password*. Pada fitur pengecekan jurnal, mahasiswa sebagai *user* non-admin bisa mengunggah jurnal ke dalam sistem setelah melengkapi data detail jurnal. Selanjutnya, mahasiswa bisa memilih daftar rujukan yang sudah ada pada *database* sistem dan juga bisa mengunggah jurnal rujukan secara manual jika pada *database* belum ada data jurnal rujukan. *Output* yang diterima oleh mahasiswa sebagai *user* non-admin adalah persentase kemiripan terhadap tiap jurnal rujukan yang dipilih serta masuk kedalam tipe plagiat apa saja.

4.2 Analisis Kebutuhan

Kebutuhan sistem adalah kemampuan yang harus dimiliki oleh sistem untuk memenuhi apa yang diinginkan oleh pengguna. Analisis kebutuhan juga dijadikan bahan untuk mulai membangun aplikasi pengidentifikasi *plagiarisme*. Analisis kebutuhan yang dilakukan meliputi analisis kebutuhan fungsional untuk menetapkan karakteristik sistem dan kebutuhan non fungsional sebagai tambahan untuk melengkapi aplikasi.

4.2.1 Kebutuhan Fungsional

- a. Sistem menggunakan fitur *login* untuk mengautentifikasi hak akses pengguna sistem.
- b. Sistem dapat menyimpan data *user* pada fitur *register user*.
- c. Sistem dapat melakukan *update* data *user*.
- d. Sistem dapat menampilkan data *user* yang ada pada *database*.
- e. Sistem dapat mengelola (*insert, update, delete*) data jurnal.
- f. Sistem dapat menampilkan data jurnal yang ada pada *database*.
- g. Sistem dapat menampilkan *log* aktifitas *user*.
- h. Sistem dapat menghitung kemiripan dokumen jurnal yang diunggah ke dalam sistem.

- i. Sistem dapat menampilkan hasil perhitungan kemiripan dokumen jurnal yang diunggah.

4.2.2 Kebutuhan Non Fungsional

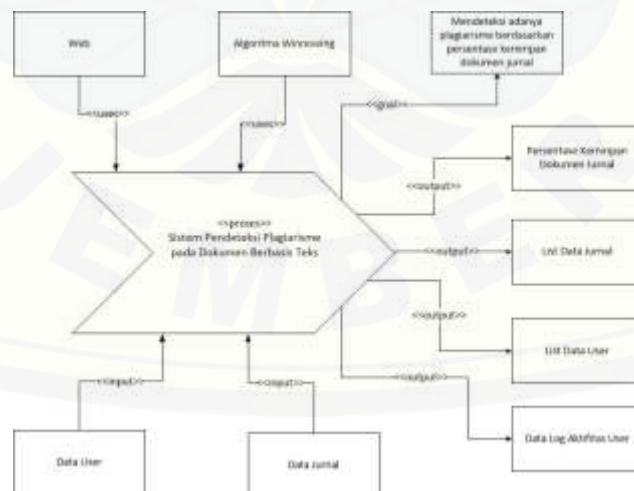
- a. Sistem berbasis *web*.
- b. Sistem menggunakan *framework CodeIgniter*.

4.3 Desain Sistem

Desain sistem adalah tahapan untuk memodelkan sistem yang akan bangun. Dokumen dari desain yang akan dibuat meliputi *Business Process*, *Usecase Diagram*, *Usecase Scenario*, *Sequence Diagram*, *Activity Diagram*, *Class Diagram*, dan *Entity Relationship Diagram (ERD)*.

4.3.1 Business Process

Business Process merupakan gambaran *input* dan *output* dari sebuah sistem serta tujuan dari sistem itu dibangun. *Business process* sistem pendeteksi *plagiarisme* dapat ditunjukkan pada gambar 4.1



Gambar 4.1 *Business process* sistem pendeteksi *plagiarisme*

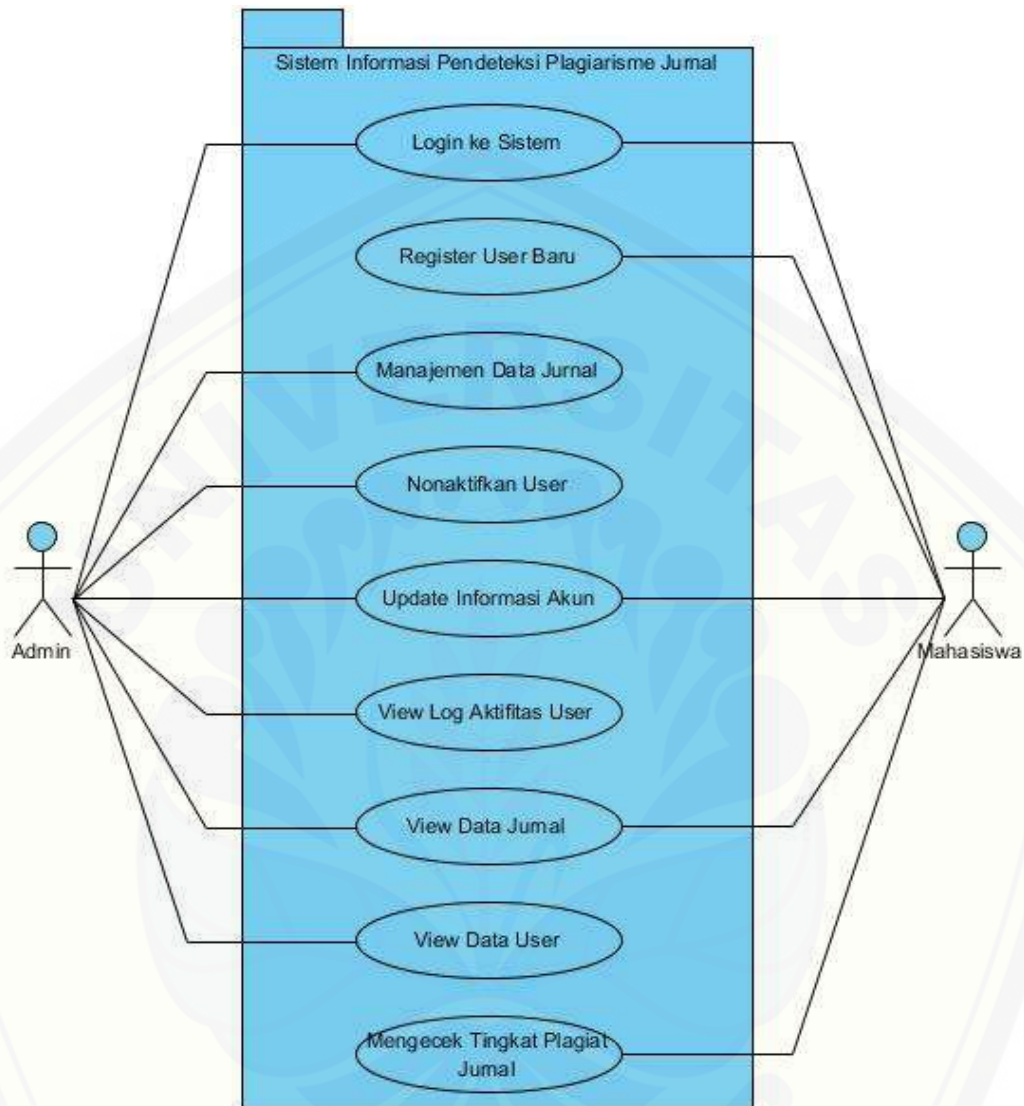
Berdasarkan Gambar 4.1, sistem menerima dua data *input* yaitu data *user* dan data jurnal. Data *user* merupakan data tentang pengguna sistem baik *user* admin maupun *user* non-admin. Data jurnal merupakan data tentang detail jurnal yang nantinya digunakan sebagai daftar rujukan ketika ada *user* non-admin ingin mengetahui tingkat plagiat jurnal yang dibuat. *Output* dari sistem merupakan data keluaran yang dihasilkan sistem. Data keluaran tersebut meliputi list data jurnal, list data user, data log aktifitas *user*, dan persentase kemiripan jurnal yang diuji pada sistem. Media pengembangan sistem menggunakan *web application* dengan bahasa pemrograman PHP dan algoritma yang diimplementasikan pada sistem adalah algoritma *winning*. Tujuan dari pembuatan sistem ini adalah mendeteksi adanya tindak plagiat pada jurnal yang dibuat berdasarkan persentase kemiripan terhadap jurnal rujukan.

4.3.2 Usecase Diagram

Usecase Diagram merupakan diagram yang digunakan untuk menjelaskan fitur-fitur yang terdapat pada sistem. Fitur-fitur yang terdapat pada sistem digambarkan dalam bentuk elips dan *user* dari sistem juga disertakan pada diagram.

Sistem yang akan dibangun memiliki dua tipe *user* yaitu, admin dan mahasiswa. *User* admin memiliki hak akses penuh terhadap sistem. Admin dapat mengelola data jurnal, data user, dan melihat *log* aktifitas *user*. *User* mahasiswa dapat melakukan pengecekan jurnal apakah terindikasi tindakan plagiat terhadap data jurnal rujukan yang ada pada sistem atau data jurnal rujukan hasil yang diunggah.

Usecase diagram pada sistem pendeteksi *plagiarisme* dengan menggunakan algoritma *winning* sebagai algoritma pencarian kemiripan dokumen dapat dilihat pada gambar 4.2 berikut



Gambar 4.2 Usecase diagram

Penjelasan tentang *usecase diagram* sistem pendeteksi plagiarisme dengan menggunakan algoritma *winnowing* sebagai algoritma pencarian kemiripan dokumen dapat dilihat pada tabel 4.1 berikut

Tabel 4.1 Definisi *usecase diagram*

No	Usecase	Aktifitas	Aktor	Penjelasan
1	<i>Login ke Sistem</i>	<i>Login</i>	Admin dan Mahasiswa	Merupakan <i>usecase</i> yang menggambarkan proses <i>login User</i> .
2	<i>Register User Baru</i>	Tambah data <i>user non-admin baru ke dalam database</i>	Mahasiswa	Merupakan <i>usecase</i> yang menggambarkan proses pendaftara <i>user</i> (mahasiswa) baru.
3	Manajemen Data Jurnal	Tambah data jurnal baru	Admin	Merupakan <i>usecase</i> yang menggambarkan proses tambah data jurnal baru ke dalam <i>database</i> .
		Hapus data jurnal	Admin	Merupakan <i>usecase</i> yang menggambarkan proses hapus data jurnal dari <i>database</i> .
		<i>Approve data jurnal</i>	Admin	Merupakan <i>usecase</i> yang menggambarkan proses penyetujuan data jurnal yang berstatus <i>pending</i> .
4	Nonaktifkan <i>User</i>	Nonaktifkan <i>user</i>	Admin	Merupakan <i>usecase</i> yang menggambarkan proses me-nonaktifkan <i>user</i> .
5	<i>Update Informasi Akun</i>	Ubah profil <i>user</i>	Admin dan Mahasiswa	Merupakan <i>usecase</i> yang menggambarkan proses <i>update data profil diri</i> .
		Ubah <i>password login</i>	Admin dan Mahasiswa	Merupakan <i>usecase</i> yang menggambarkan proses <i>update password login user</i> .
6	<i>View Log Aktifitas User</i>	<i>View semua data aktifitas user</i>	Admin	Merupakan <i>usecase</i> yang menggambarkan proses <i>view log aktifitas</i> yang dilakukan <i>user</i> pada sistem.

Dilanjutkan

No	Usecase	Aktifitas	Aktor	Penjelasan
7	View Data Jurnal	View data jurnal yang ada pada database	Admin dan Mahasiswa	Merupakan <i>usecase</i> yang menggambarkan proses untuk melihat <i>list</i> data jurnal.
8	View Data User	View data user pada database	Admin	Merupakan <i>usecase</i> yang menggambarkan proses untuk melihat <i>list</i> data user.
9	Mengecek Tingkat Plagiat Jurnal	Mengecek plagiat jurnal	Mahasiswa	Merupakan <i>usecase</i> yang menggambarkan proses pengecekan <i>plagiarisme</i> jurnal yang dibuat terhadap jurnal rujukan yang ada pada database maupun jurnal rujukan yang diunggah sendiri.

4.3.3 Scenario Diagram

Scenario Diagram merupakan penjelasan alur dari sebuah sistem serta alur alternatif yang ada pada sebuah fitur sistem. Pada *scenario* sistem disertakan juga aktor yang mengakses fitur tersebut. Pembuatan *scenario* sistem mengacu pada fitur-fitur yang terdapat pada *usecase diagram* yang telah dibuat sebelumnya. Berikut ini adalah *scenario* dari sistem yang akan dibuat.

a. *Scenario* dari *Usecase Login* ke Sistem

Scenario login ke sistem menjelaskan alur dari proses *login user* ke dalam sistem. *Scenario* utama adalah alur dari pertama kali *user* membuka *website* sampai *user* berhasil masuk ke dalam sistem. Sedangkan *scenario* alternatif merupakan bagian yang menangani alur pengecualian apabila terdapat kesalahan pada saat *user* ingin masuk ke dalam sistem. *Scenario login* ke sistem oleh *user* admin dapat dilihat pada tabel 4.2 sedangkan *scenario login* ke sistem oleh *user* mahasiswa dapat dilihat pada tabel 4.3.

Tabel 4.2 *Scenario login admin*

Nama Usecase	<i>Login ke Sistem</i>
Aktor	Admin
Pra-Kondisi	Admin ingin mengakses sistem
Pasca-Kondisi	Admin berhasil <i>login</i> ke sistem
Aktor	Sistem
Skenario Utama “Login ke Sistem”	
1. Membuka <i>website</i>	
	2. Menampilkan halaman <i>login</i> untuk masuk ke sistem
3. Memasukkan <i>username</i> dan <i>password</i>	
4. Klik tombol “ <i>sign in</i> ”	
	5. Mengecek data <i>user</i> ke <i>database</i>
	6. Menampilkan halaman utama admin
Skenario Alternatif “Username dan Password Salah”	
4a. Klik tombol “ <i>sign in</i> ”	
	5a. Mengecek data <i>user</i> ke <i>database</i>
	6a. Menampilkan pesan <i>error</i> “ <i>username</i> dan <i>password</i> salah”
Skenario Alternatif “Form login tidak terisi”	
4b. Klik tombol “ <i>sign in</i> ”	
	5b. Menampilkan pesan <i>error</i> pada <i>field</i> yang kosong

Tabel 4.3 *Scenario login mahasiswa*

Nama Usecase	<i>Login ke Sistem</i>
Aktor	Mahasiswa
Pra-Kondisi	Mahasiswa ingin mengakses sistem
Pasca-Kondisi	Mahasiswa berhasil <i>login</i> ke sistem
Aktor	Sistem
Skenario Utama “Login ke Sistem”	
1. Membuka <i>website</i>	
	2. Menampilkan halaman <i>login</i> untuk masuk ke sistem
3. Memasukkan <i>username</i> dan <i>password</i>	
4. Klik tombol “ <i>sign in</i> ”	
	5. Mengecek data <i>user</i> ke <i>database</i>
	6. Menampilkan halaman utama mahasiswa
Skenario Alternatif “Username dan Password Salah”	
4a. Klik tombol “ <i>sign in</i> ”	
	5a. Mengecek data <i>user</i> ke <i>database</i>
	6a. Menampilkan pesan <i>error</i> “ <i>username</i> dan <i>password</i> salah”
Skenario Alternatif “Form login tidak terisi”	
4b. Klik tombol “ <i>sign in</i> ”	
	5b. Menampilkan pesan <i>error</i> pada <i>field</i> yang kosong

b. *Scenario* dari *Usecase Register User Baru*

Scenario ini menjelaskan alur proses pendaftaran *user* baru untuk mendapatkan *username* dan *password* yang nanti bisa digunakan untuk *login* ke sistem. *Scenario* utama merupakan alur utama dari proses *register user* baru. *Scenario* alternatif merupakan alur pengecualian yang ditangani oleh sistem apabila ada kesalahan pada saat mendaftarkan *user* baru. Kondisi akhir dari

scenario ini adalah mahasiswa sebagai *user* non-admin berhasil melakukan *register user* baru dan mendapatkan *username* serta *password* untuk *login* ke sistem. *Scenario register user* baru dapat dilihat pada tabel 4.4 berikut.

Tabel 4.4 *Scenario register user* baru

Nama Usecase	<i>Register User</i> Baru
Aktor	Mahasiswa
Pra-Kondisi	Mahasiswa ingin <i>register user</i> baru
Pasca-Kondisi	Mahasiswa berhasil <i>register user</i>
Aktor	Sistem
Skenario Utama “Register User Baru”	
1. Mengakses halaman <i>sign up</i>	
	2. Menampilkan halaman <i>sign up user</i> baru
3. Mengisi <i>form sign up</i>	
4. Klik tombol “ <i>sign up</i> ”	
	5. Menyimpan data <i>user baru</i> ke <i>database</i>
	6. Menampilkan halaman <i>login</i> dengan pesan sukses <i>register</i>
Skenario Alternatif “Input Tidak Valid”	
4a. Klik tombol “ <i>sign up</i> ”	
	5a. Menampilkan pesan <i>error</i> pada <i>field</i>
Skenario Alternatif “Form sign up tidak terisi”	
4b. Klik tombol “ <i>sign up</i> ”	
	5b. Menampilkan pesan <i>error</i> pada <i>field</i> yang kosong

c. *Scenario* dari *Usecase* Manajemen Data Jurnal – Tambah Data Jurnal

Tambah data jurnal merupakan sub proses dari *usecase* manajemen data jurnal. *Scenario* utama merupakan alur proses menambahkan data jurnal baru ke dalam sistem. Pada proses ini terdapat implementasi *algoritma winnowing* pada

saat pembentukan *fingerprint* dokumen jurnal yang nantinya digunakan untuk mengecek kemiripan dokumen jurnal yang diunggah dengan dokumen jurnal yang sudah ada pada *database* guna menghindari terjadinya duplikasi data jurnal. *Scenario* tambah data jurnal terdapat beberapa alur alternatif yang digunakan untuk menangani alur pengecualian yaitu penanganan *exception* apabila ada *input* tidak *valid*, penanganan “klik tombol cancel” yang menangani pembatalan proses simpan data jurnal, dan “klik tombol tampilkan detail perhitungan” yang menangani proses menampilkan detail perhitungan pembentukan *fingerprint* dokumen. *Scenario* tambah data jurnal baru dapat dilihat pada tabel 4.5 berikut.

Tabel 4.5 *Scenario* tambah data jurnal baru

Nama Usecase	Manajemen Data Jurnal
Aktor	Admin
Pra-Kondisi	Admin <i>login</i> ke dalam sistem
Pasca-Kondisi	Admin berhasil menambahkan data jurnal baru
Aktor	Sistem
Skenario Utama	
“Tambah Data Jurnal Baru”	
1. Klik Menu “ <i>Upload Jurnal</i> ”	
	2. Menampilkan <i>form upload</i> jurnal baru
3. Mengisi <i>form upload</i> jurnal baru	
4. Klik tombol “ <i>upload</i> ”	
	5. Menyimpan <i>file</i> jurnal ke dalam direktori <i>website</i>
	6. Menampilkan <i>preview</i> jurnal yang diunggah
7. Klik tombol “ <i>proses</i> ”	
	8. Melakukan proses pembentukan <i>fingerprint</i> dengan algoritma <i>winnowing</i>
	9. Menampilkan hasil perhitungan
Dilanjutkan	

Aktor	Sistem
10. Klik tombol “proses”	
	11. Melakukan proses cek pada semua data jurnal di <i>database</i> dengan bidang studi sama
	12. Menampilkan persentase kemiripan dokumen hasil pengecekan dari <i>database</i> dalam bentuk tabel
13. Klik tombol “simpan ke <i>database</i> ”	
	14. Menyimpan data jurnal ke dalam <i>database</i>
	15. Menampilkan data <i>list</i> jurnal yang ada di <i>database</i>
Skenario Alternatif “Form Kosong atau Tidak Valid”	
4a. Klik tombol “upload”	
	5a. Menampilkan pesan <i>error</i> pada <i>field</i>
Skenario Alternatif “Klik Tombol Cancel”	
7a. Klik tombol “cancel”	
	8a. Menghapus <i>file</i> jurnal dari direktori <i>website</i>
	9a. Menampilkan halaman <i>form upload</i> jurnal
Skenario Alternatif “Klik Tombol Tampilkan Detail Perhitungan”	
10a. Klik tombol “tampilkan detail <i>perhitungan</i> ”	
	11a. Menampilkan detail perhitungan pembentukan <i>fingerprnt</i> jurnal
Skenario Alternatif “Klik Tombol Cancel”	
10b. Klik tombol “cancel”	
	11b. Menghapus <i>file</i> jurnal dari direktori <i>website</i>
	12b. Menampilkan halaman <i>form upload</i> jurnal
Dilanjutkan	

Aktor	Sistem
Skenario Alternatif “Klik Tombol Cancel”	
13a. Klik tombol “cancel”	
	14a. Menghapus <i>file</i> jurnal dari direktori <i>website</i>
	15a. Menampilkan halaman <i>form upload</i> jurnal

d. *Scenario* dari *Usecase* Manajemen Data Jurnal – Hapus Data Jurnal

Hapus data jurnal merupakan sub proses dari *usecase* manajemen data jurnal. *Scenario* hapus data jurnal menjelaskan alur proses menghapus data jurnal. *Scenario* utama merupakan alur utama dari proses hapus data jurnal. *Scenario* alternatif merupakan alur pengecualian yang menangani *exception* apabila *user* batal menghapus data jurnal. Kondisi akhir dari *scenario* ini adalah admin berhasil menghapus data jurnal. *Scenario* hapus data jurnal baru dapat dilihat pada tabel 4.6 berikut.

Tabel 4.6 *Scenario* hapus data jurnal

Nama Usecase	Manajemen Data Jurnal
Aktor	Admin
Pra-Kondisi	Admin <i>login</i> ke dalam sistem
Pasca-Kondisi	Admin berhasil menghapus data jurnal
Aktor	Sistem
Skenario Utama “Hapus Data Jurnal”	
1. Klik menu “ <i>list jurnal</i> ”	
	2. Menampilkan halaman <i>list</i> jurnal beserta data jurnal
3. Klik <i>icon</i> tombol “ <i>hapus</i> ” pada data yang akan dihapus	
	4. Menampilkan <i>pop-up</i> konfirmasi hapus data jurnal
Dilanjutkan	

Aktor	Sistem
5. Klik tombol “ <i>confirm</i> ” pada <i>pop-up</i>	
	6. Menghapus data jurnal yang dipilih dari <i>database</i>
	7. Menampilkan halaman <i>list</i> jurnal beserta data jurnal
Skenario Alternatif “Batal Hapus Data Jurnal”	
5a. Klik tombol “ <i>close</i> ” pada <i>pop-up</i>	
	6a. Menampilkan halaman <i>list</i> jurnal beserta data jurnal

e. *Scenario* dari *Usecase* Manajemen Data Jurnal – *Approve* Data Jurnal

Approve data jurnal merupakan sub proses dari *usecase* manajemen data jurnal. *Scenario approve* data jurnal menjelaskan alur proses menyetujui data jurnal yang bersatus *pending* oleh *user* admin. *Scenario* utama merupakan alur utama dari proses menyetujui data jurnal dengan status *pending* dan menyimpan data jurnal ke *database* yang kemudian bisa digunakan oleh mahasiswa untuk pengecekan plagiat selanjutnya. *Scenario* alternatif merupakan alur yang menangani alur pengecualian diantaranya alur tidak menyetujui data jurnal, lihat detail jurnal sebelum disetujui, dan batal menyetujui data jurnal. Kondisi akhir dari *scenario* ini adalah admin berhasil menyetujui data jurnal dan menyimpan data jurnal ke *database*. *Scenario approve* data jurnal dapat dilihat pada tabel 4.7 berikut.

Tabel 4.7 *Scenario approve data jurnal*

Nama Usecase	Manajemen Data Jurnal
Aktor	Admin
Pra-Kondisi	Admin <i>login</i> ke dalam sistem
Pasca-Kondisi	Admin berhasil menyetujui data jurnal dengan status <i>pending</i>
Aktor	Sistem
Skenario Utama “Approve Data Jurnal”	
1. Klik menu “ <i>approve jurnal</i> ”	2. Menampilkan halaman <i>approve jurnal</i> beserta data jurnal dengan status <i>pending</i>
3. Klik <i>icon</i> tombol “ <i>setujui</i> ” pada data yang akan disetujui	4. Menampilkan <i>pop-up</i> konfirmasi setuju data jurnal
5. Klik tombol “ <i>confirm</i> ” pada <i>pop-up</i>	6. Menyimpan data jurnal yang dipilih dari <i>database</i>
	7. Menampilkan halaman <i>approve jurnal</i> beserta data jurnal dengan status <i>pending</i>
Skenario Alternatif “Batal Approve Data Jurnal”	
5a. Klik tombol “ <i>close</i> ” pada <i>pop-up</i>	6a. Menampilkan halaman <i>approve jurnal</i> beserta data jurnal dengan status <i>pending</i>
Skenario Alternatif “Tidak Menyetujui Data Jurnal”	
3b. Klik <i>icon</i> tombol “ <i>tidak menyetujui</i> ” pada data jurnal	4b. Menampilkan <i>pop-up</i> konfirmasi tidak menyetujui data jurnal
5b. Klik tombol “ <i>confirm</i> ” pada <i>pop-up</i>	
Dilanjutkan	

Aktor	Sistem
	6b. Menghapus data jurnal dari <i>database</i>
	7b. Menampilkan halaman <i>approve</i> jurnal beserta data jurnal dengan status <i>pending</i>
Skenario Alternatif “Lihat Detail Data Jurnal”	
3c. Klik <i>icon</i> tombol “ <i>detail</i> ” pada data jurnal	
	4c. Menampilkan <i>pop-up</i> detail informasi jurnal
5c. Klik tombol “ <i>close</i> ” pada <i>pop-up</i>	
	6c. Menampilkan halaman <i>approve</i> jurnal beserta data jurnal dengan status <i>pending</i>

f. *Scenario* dari *Usecase* Nonaktifkan *User*

Scenario nonaktifkan *user* menjelaskan alur proses menonaktifkan *user* oleh admin. *Scenario* utama merupakan alur utama dari proses menonaktifkan *user*. Sedangkan *scenario* alternatif merupakan alur pengecualian pada proses menonaktifkan *user*. *User* dengan status nonaktif tidak dapat melakukan *login* ke sistem sampai admin mengaktifkan *user* tersebut. *Scenario* nonaktifkan *user* dapat dilihat pada Lampiran A1.

g. *Scenario* dari *Usecase* Update Informasi Akun – Ubah Profil *User*

Mengubah profi *user* merupakan sub proses dari *usecase* update informasi akun. *User* admin maupun mahasiswa dapat mengubah informasi profil diri. *Scenario* utama merupakan alur utama dari proses ubah profil *user*. Sedangkan *scenario* alternatif merupakan alur pengecualian pada proses ubah profil *user*. *Scenario* ubah profil dapat dilihat pada Lampiran A1.

h. *Scenario* dari *Usecase* Update Informasi Akun – Ubah *Password Login*

Mengubah *password login* merupakan sub proses dari *usecase* update informasi akun. *User* admin maupun mahasiswa dapat mengubah *password login*. *Scenario* utama merupakan alur utama dari proses ubah *password login*.

Sedangkan *scenario* alternatif merupakan alur pengecualian pada proses profil *user*. *Scenario* ubah profil dapat dilihat pada Lampiran A1.

i. *Scenario* dari *Usecase View Log Aktifitas User*

Scenario view log aktifitas *user* menjelaskan alur proses melihat aktifitas yang *user* mahasiswa maupun admin lakukan selama mengakses sistem. *View log* aktifitas hanya bisa dilakukan oleh *user* admin. *Scenario* utama merupakan alur utama dari proses *view log* aktifitas *user*. *Scenario view log* aktifitas *user* dapat dilihat pada Lampiran A1.

j. *Scenario* dari *Usecase View Data Jurnal*

Scenario view data jurnal menjelaskan alur proses melihat data jurnal yang ada pada *database*. *View* data jurnal bisa dilakukan oleh *user* admin maupun mahasiswa. *Scenario* utama merupakan alur utama dari proses *view* data jurnal. *Scenario view* data jurnal dapat dilihat pada Lampiran A1.

k. *Scenario* dari *Usecase View Data User*

Scenario view data *user* menjelaskan alur proses melihat data *user* yang ada pada *database*. *View* data *user* hanya bisa dilakukan oleh *user* admin. *Scenario* utama merupakan alur utama dari proses *view* data *user*. *Scenario view* data jurnal dapat dilihat pada Lampiran A1.

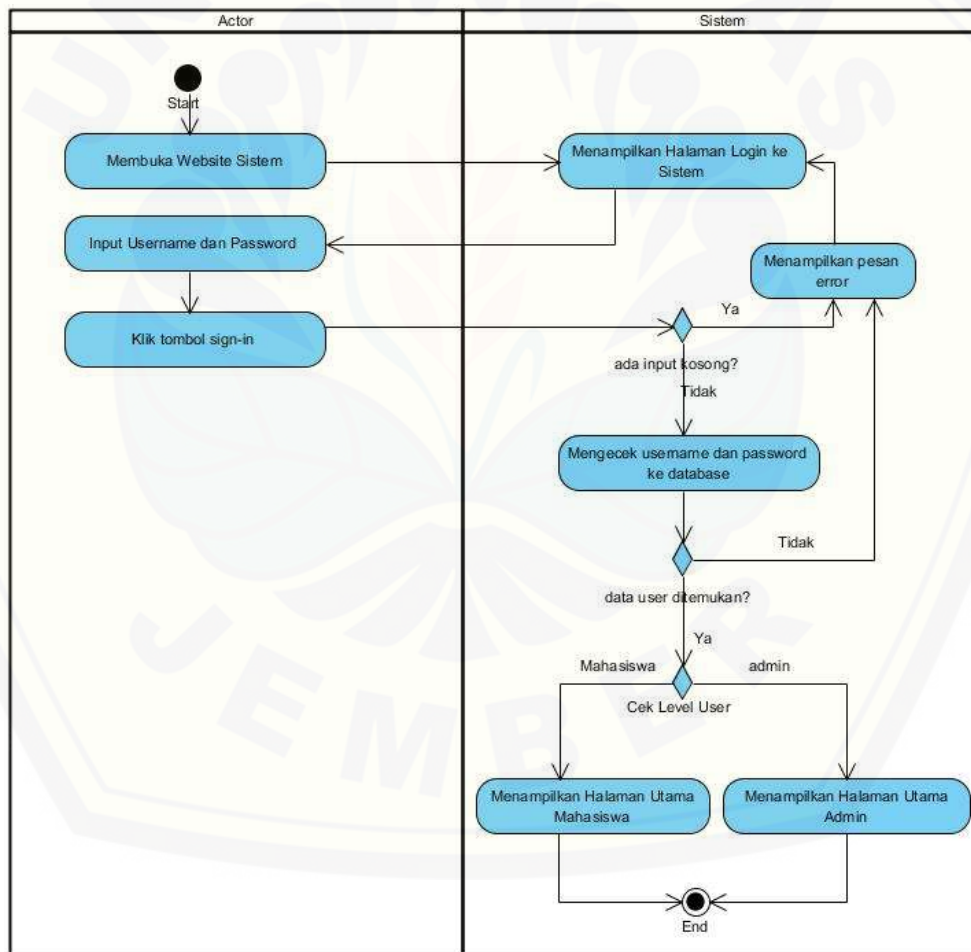
l. *Scenario* dari *Usecase Mengecek Plagiat Jurnal*

Scenario mengecek plagiat jurnal menjelaskan alur proses pengecekan tingkat plagiat jurnal oleh *user* mahasiswa. *User* mahasiswa melakukan proses *upload* jurnal yang ingin diuji yang selanjutnya sistem akan memproses jurnal tersebut dengan menggunakan algoritma *winnowing*. Mahasiswa dapat memilih referensi data jurnal dari *database*, jika data jurnal referensi pada *database* tidak ditemukan, mahasiswa bisa mengupload data jurnal secara manual. Pada proses ini terdapat pengimplementasian algoritma *winnowing* pada proses pembentukan *fingerprint*. *Scenario* mengecek plagiat jurnal dapat dilihat pada Lampiran A1.

4.3.4 Activity Diagram

Activity diagram merupakan gambaran tentang alur aktifitas dalam sistem yang akan dibangun, bagaimana masing-masing alur berasal, *decision* yang mungkin terjadi, serta bagaimana alur berakhir. Pembuatan *activity diagram* sistem pendeteksi *plagiarisme* dengan algoritma *winnowing* sebagai algoritma pencarian kemiripan dokumen mengacu pada *usecase* dan *scenario* yang telah dibuat sebelumnya. Berikut adalah *activity diagram* dari sistem yang akan dibangun

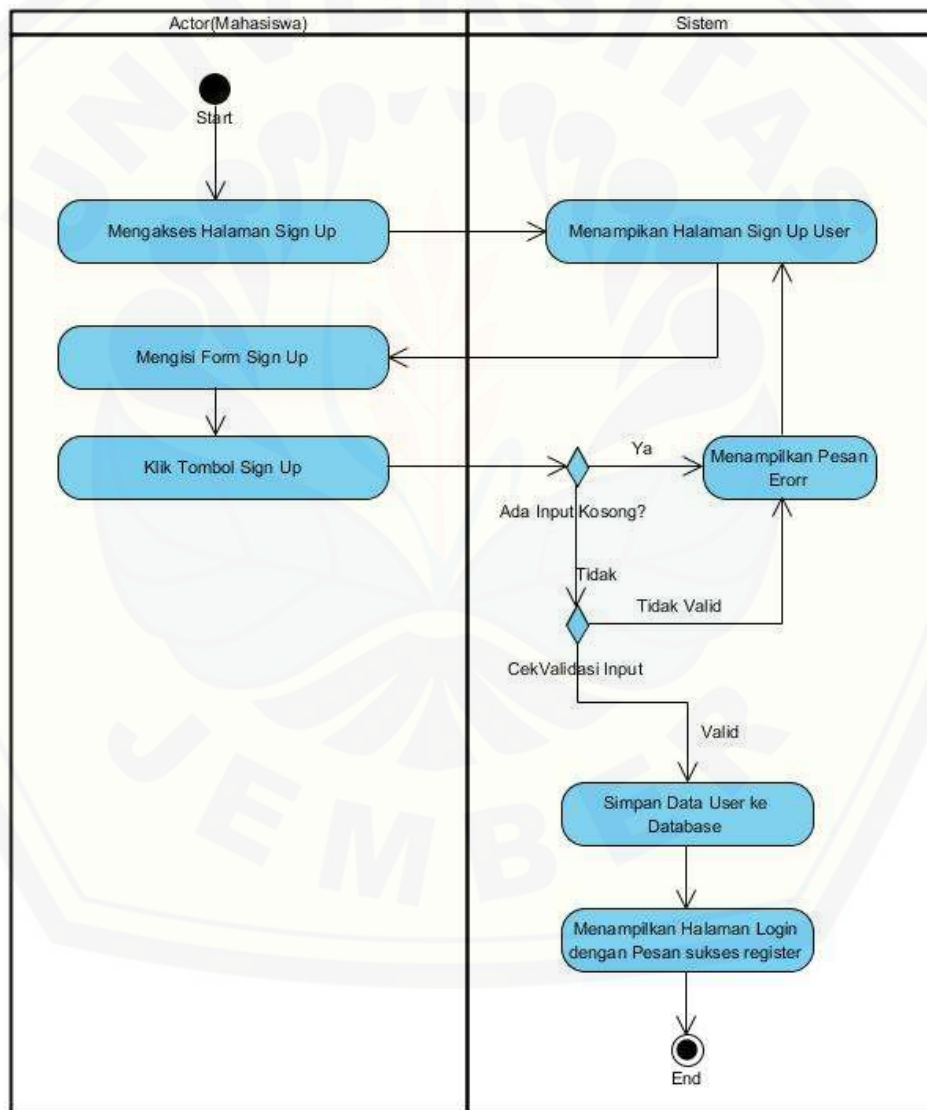
a. Activity Diagram Login ke Sistem



Gambar 4.3 Activity diagram login

Gambar 4.3 merupakan *activity diagram* yang menggambarkan alur yang terjadi ketika *user* ingin mengakses sistem dan melakukan aktivitas *login*. *Login user* menggunakan *username* dan *password* yang sudah terdaftar di dalam *database*, setelah berhasil *login* sistem akan menampilkan halaman utama sesuai level *user* yang digunakan pada saat *login*.

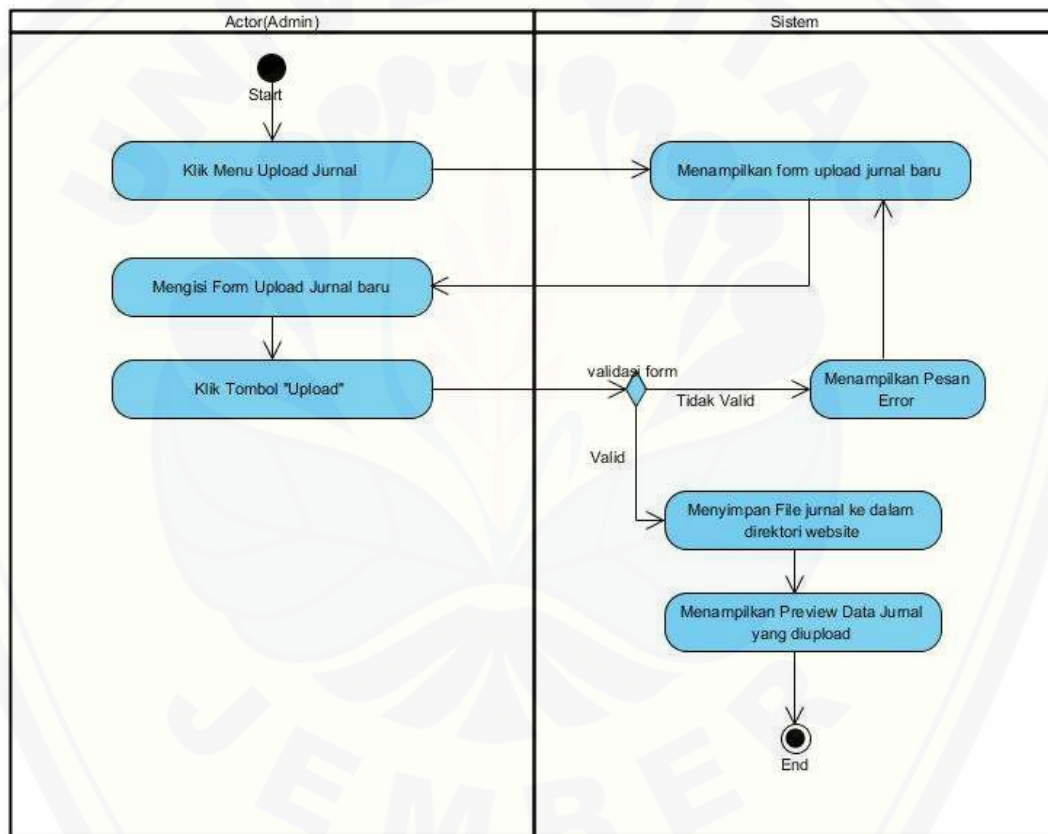
b. *Activity Diagram Register User Baru*



Gambar 4.4 *Activity diagram register user baru*

Gambar 4.4 merupakan *activity diagram* yang menggambarkan alur aktivitas proses *register user* baru yang dilakukan oleh *user* mahasiswa. Sistem melakukan pengecekan sebelum menyimpan data *user* baru ke dalam *database* untuk memastikan data yang dimasukkan oleh *user* merupakan data yang *valid*. Setelah sukses melakukan *registrasi user*, sistem otomatis mengalihkan ke halaman *login*.

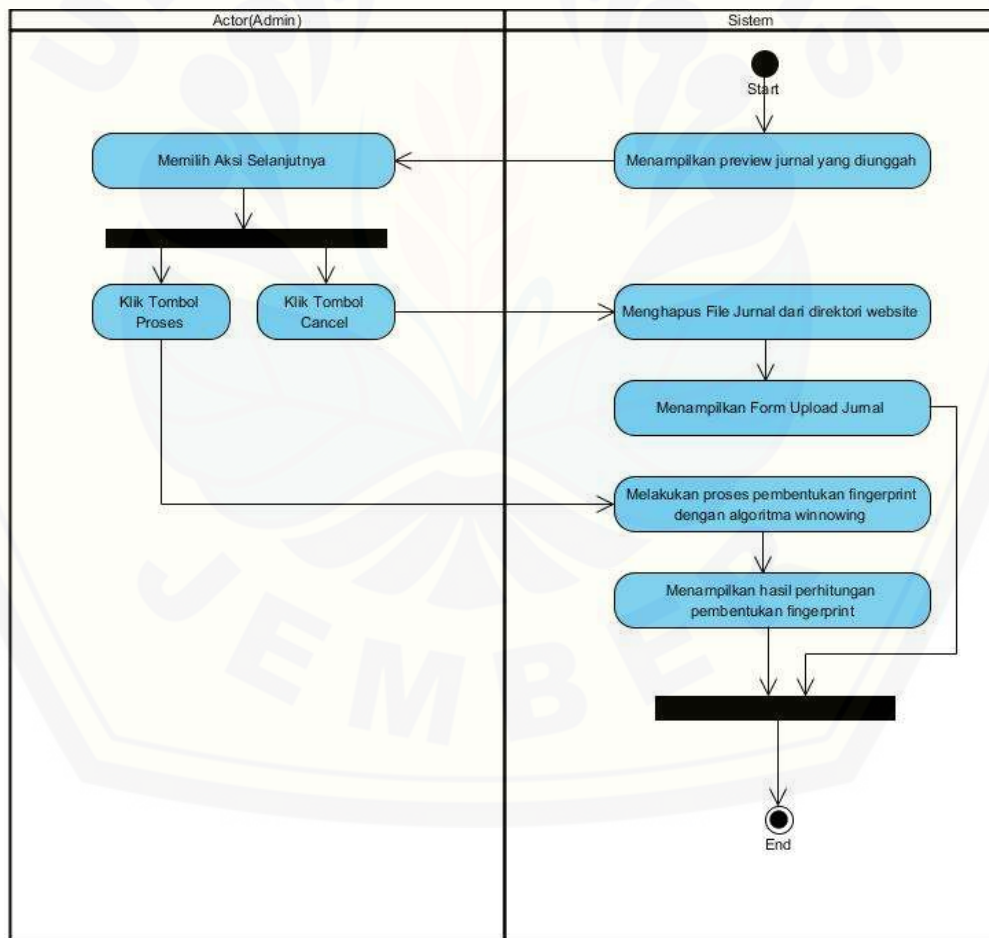
c. *Activity Diagram* Manajemen Data Jurnal – Tambah Data Jurnal



Gambar 4.5 *Activity diagram* upload jurnal

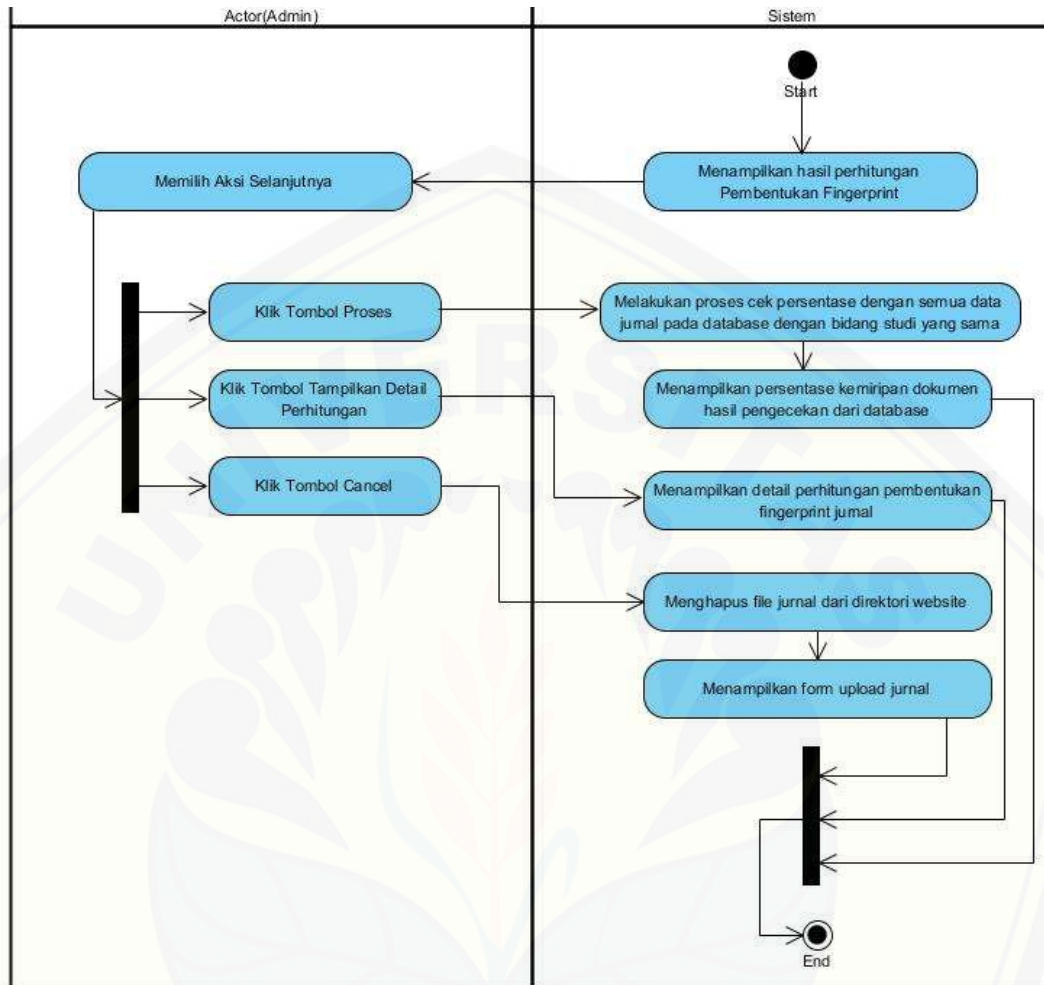
Gambar 4.5 merupakan *activity diagram* yang menggambarkan alur proses *upload* jurnal oleh *user* admin. Halaman awal yang ditampilkan oleh sistem ketika *user* admin memilih menu *upload* jurnal adalah *form upload* jurnal.

Sistem juga melakukan pengecekan terhadap data yang dimasukkan oleh *user*, jika data yang dimasukkan tidak *valid* sesuai dengan ketentuan sistem maka sistem akan menampilkan pesan *error* dan proses tidak dapat dilanjutkan. Jika data yang dimasukkan oleh *user* admin sesuai dengan ketentuan sistem maka akan dilanjutkan dengan menyimpan *file* jurnal yang diunggah ke dalam direktori *website* dan dilanjutkan dengan menampilkan *preview* data jurnal. Proses ini tidak melakukan proses simpan data ke *database*. Proses selanjutnya adalah menampilkan *preview* data jurnal yang diunggah dapat dilihat pada gambar 4.6 berikut.



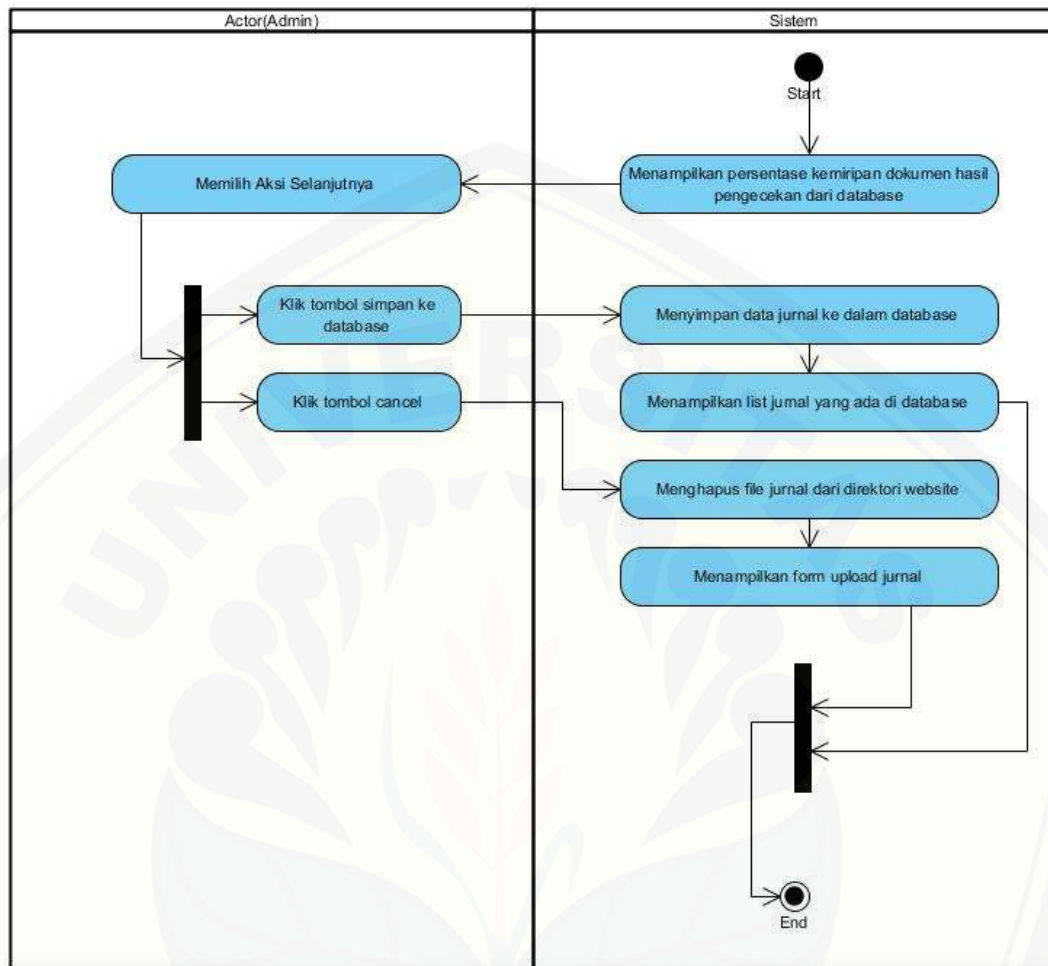
Gambar 4.6 Activity diagram preview upload jurnal

Gambar 4.6 merupakan *activity diagram* yang menggambarkan alur proses setelah *file* jurnal berhasil diunggah. Sebelum disimpan di dalam *database*, *file* jurnal yang berhasil diunggah akan ditampilkan detail informasinya. *User* admin dapat memilih untuk membatalkan proses dengan menekan tombol *cancel* dan sistem akan menghapus *file* jurnal yang ada dalam direktori *website* yang kemudian menampilkan halaman awal *upload* jurnal. Sedangkan untuk melanjutkan proses tambah data jurnal, admin perlu menekan tombol proses yang kemudian akan dilanjutkan dengan proses pembentukan *fingerprint* dokumen jurnal dengan menggunakan algoritma *winnowing* dan sistem akan menampilkan hasil pembentukan *fingerprint* jurnal. Proses selanjutnya setelah diperoleh *fingerprint* dokumen jurnal yang diunggah adalah proses pengecekan dengan dokumen jurnal yang ada pada *database*. *Activity diagram* proses pengecekan jurnal dapat dilihat pada gambar 4.7 berikut.



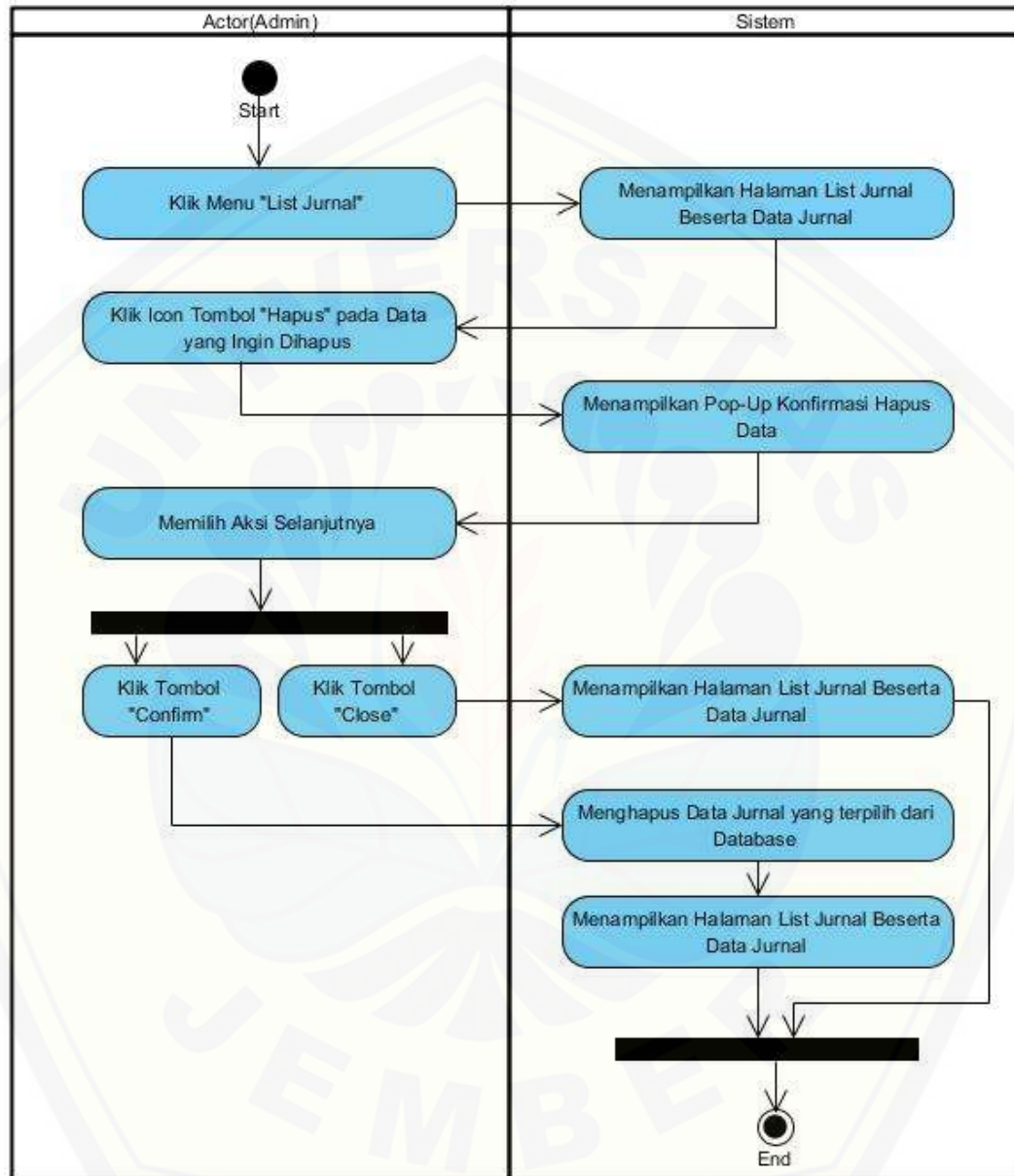
Gambar 4.7 Activity diagram pengecekan dengan jurnal di database

Proses pengecekan antara jurnal yang diunggah dengan jurnal yang ada di database bertujuan untuk menghindari adanya jurnal dengan isi yang sama meskipun judul jurnal berbeda. Proses selanjutnya setelah dilakukan pengecekan dengan jurnal yang ada di database adalah proses menyimpan data jurnal ke dalam database. Proses simpan data jurnal baru dapat dilihat pada gambar 4.8 berikut.



Gambar 4.8 Activity diagram simpan data jurnal

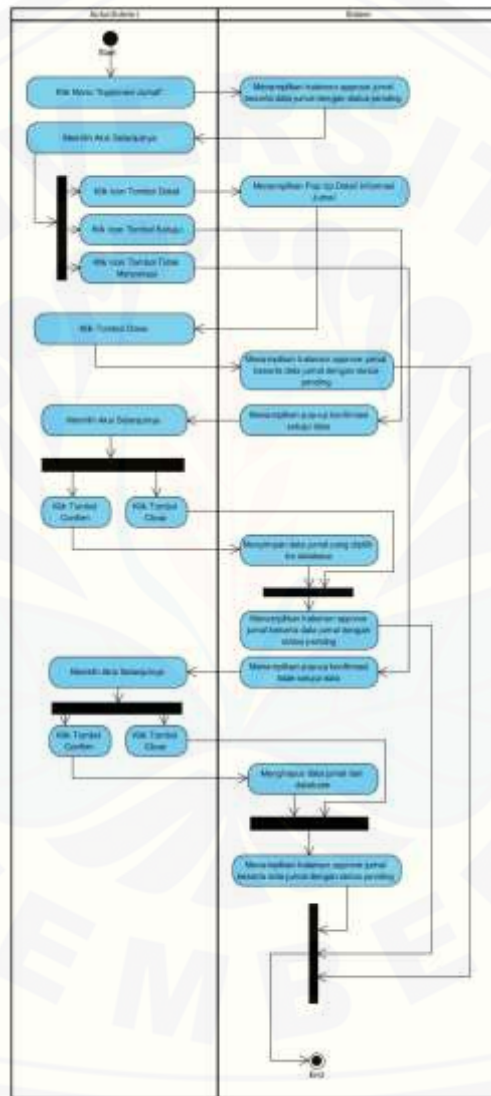
Gambar 4.8 merupakan *activity diagram* simpan data jurnal. Proses ini merupakan proses terakhir dari aktivitas tambah data jurnal. Setelah *user* admin menekan tombol simpan ke database, maka sistem akan menyimpan data jurnal baru ke dalam *database* dan siap digunakan sebagai referensi pada pengecekan oleh mahasiswa.

d. *Activity Diagram* Manajemen Data Jurnal – Hapus Data JurnalGambar 4.9 *Activity diagram* hapus data jurnal

Gambar 4.9 merupakan *activity diagram* yang menggambarkan alur proses hapus data jurnal. Data jurnal yang ditampilkan dalam bentuk tabel

masing-masing memiliki tombol aksi berbentuk *icon*. Ketika admin menekan icon tombol hapus, maka sistem akan menampilkan *pop-up* konfirmasi sebelum data dihapus dari *database*.

e. *Activity Diagram Approve Data Jurnal*



Gambar 4.10 *Activity diagram approve data jurnal*

Gambar 4.10 merupakan *activity diagram* yang menggambarkan alur proses menyetujui data jurnal dengan status *pending* oleh admin. Data jurnal

dengan status *pending* ditampilkan dalam bentuk tabel dan terdapat 3 tombol aksi yaitu detail yang berfungsi untuk melihat detail informasi jurnal, setuju yang berfungsi untuk menyetujui data jurnal dengan status *pending* dan menyimpannya di *database*, tidak setuju yang berfungsi untuk menghapus data jurnal.

f. *Activity Diagram Nonaktifkan User*

Activity diagram yang menggambarkan alur proses menonaktifkan *user* oleh *user* admin. Admin mengakses halaman *list user* dan sistem akan menampilkan data *user* dalam bentuk tabel dengan *icon* tombol aksi pada masing-masing baris data. Ketika admin menekan *icon* tombol nonaktifkan, maka sistem akan menampilkan *pop-up* konfirmasi nonaktifkan *user*. *Activity diagram* nonaktifkan *user* dapat dilihat pada Lampiran A2.

g. *Activity Diagram Update Informasi Akun – Ubah Profil User*

Activity diagram yang menggambarkan alur proses ubah profil *user* yang bisa dilakukan oleh admin maupun mahasiswa. *User* memilih menu “profil” dan sistem akan menampilkan detail informasi akun beserta tombol aksi untuk mengubah data profil *user*. *Activity diagram* ubah profil *user* dapat dilihat pada Lampiran A2.

h. *Activity Diagram Update Informasi Akun – Ubah Password Login*

Activity diagram yang menggambarkan alur proses ubah *password login* yang bisa dilakukan oleh admin maupun mahasiswa. *User* memilih menu “update password” dan sistem akan menampilkan *form update password* untuk mengubah *password login user*. *Activity diagram* ubah profil *user* dapat dilihat pada Lampiran A2.

i. *Activity Diagram View Log Aktifitas User*

Activity diagram yang menggambarkan alur proses *view log* aktifitas *user* yang hanya bisa dilakukan oleh admin. Admin memilih menu “Log” dan sistem akan menampilkan *log* aktifitas dalam bentuk tabel. *Activity diagram view log* aktifitas *user* dapat dilihat pada Lampiran A2.

j. *Activity Diagram View Data Jurnal*

Activity diagram yang menggambarkan alur proses *view* data jurnal yang bisa dilakukan oleh admin maupun mahasiswa. *User* memilih menu “List Jurnal” dan sistem akan menampilkan *list* data jurnal dalam bentuk tabel. *Activity diagram view* data jurnal dapat dilihat pada Lampiran A2.

k. *Activity Diagram View Data User*

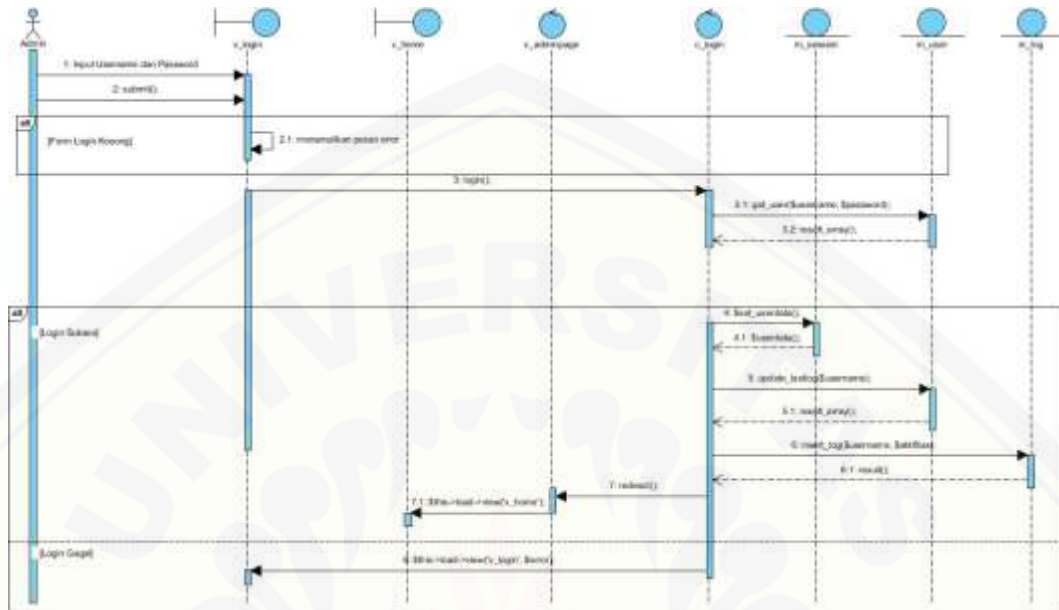
Activity diagram yang menggambarkan alur proses *view* data *user* yang hanya bisa dilakukan oleh admin. Admin memilih menu “List Users” dan sistem akan menampilkan *list* data *user* dalam bentuk tabel. *Activity diagram view* data *user* dapat dilihat pada Lampiran A2.

l. *Activity Diagram Mengecek Plagiat Jurnal*

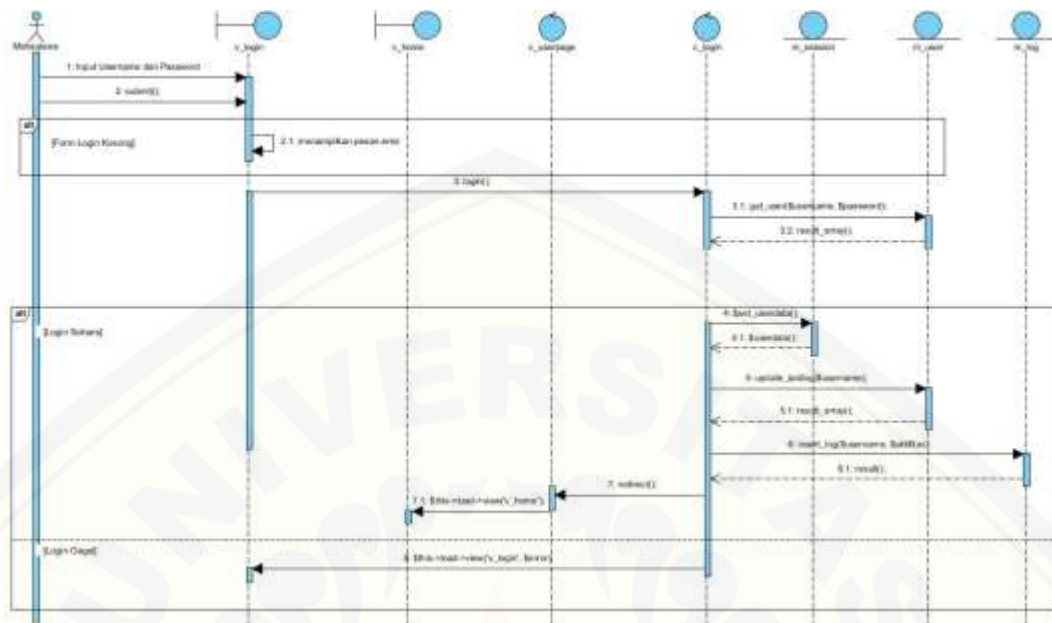
Activity diagram yang menggambarkan alur proses pengecekan plagiat jurnal yang bisa dilakukan oleh mahasiswa. Proses pengecekan plagiat jurnal mirip dengan proses tambah data jurnal yang dilakukan oleh admin, akan tetapi terdapat alur pemilihan jurnal referensi sebelum sistem melakukan pengecekan terhadap jurnal yang ingin diuji. *Activity diagram* mengecek plagiat jurnal dapat dilihat pada Lampiran A2.

4.3.5 *Sequence diagram*

Sequence diagram merupakan suatu diagram yang menampilkan interaksi-interaksi antar objek didalam sistem yang dibangun. Dengan kata lain, *sequence diagram* memodelkan aliran logika dalam sebuah sistem dengan cara yang visual. Berikut ini adalah *sequence diagram* sistem pendeteksi *plagiarisme* dengan algoritma *winnowing* sebagai algoritma pencarian kemiripan dokumen.

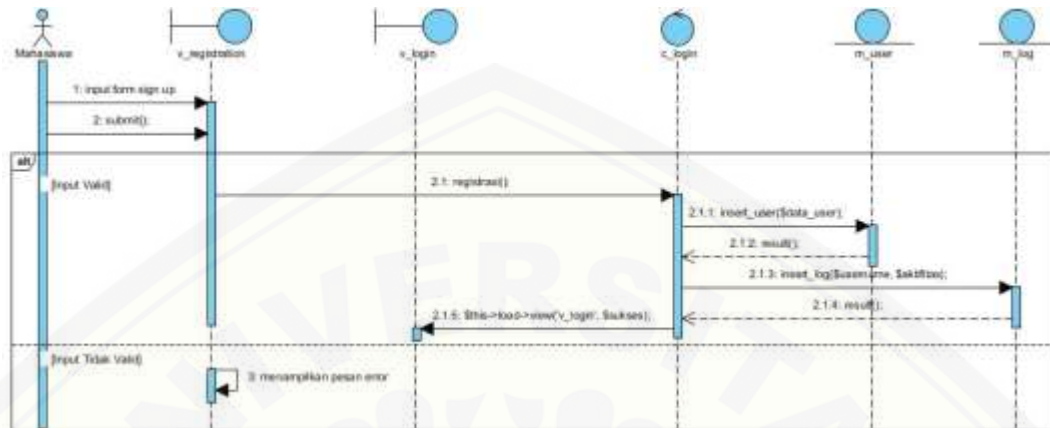
a. *Sequence Diagram Login ke Sistem*Gambar 4.11 *Sequence diagram login admin*

Gambar 4.11 merupakan *sequence diagram* yang menggambarkan alur logika pada fitur *login* dengan aktor admin. *Sequence diagram login admin* terdapat dua *view* dengan *v_login* sebagai *view default* ketika *website* pertama kali diakses. Admin melakukan *input username* dan *password* pada *form login* yang ditampilkan oleh *view v_login*. *Controller* yang menangani proses *login* adalah *c_login* yang berinteraksi dengan *model m_user*. Pada saat proses pengecekan berhasil dilakukan, maka sistem otomatis membuat penyimpanan sementara pada *model m_session*. *Controller c_login* juga berfungsi melakukan pengecekan terhadap level *user* yang didapat dari *model m_user*, *c_login* akan melakukan proses *redirect* sesuai dengan level *user* yang digunakan pada saat *login*. Level *user* yang digunakan pada *sequence diagram login admin* adalah *user admin* yang berarti *c_login* akan berinteraksi dengan *controller c_adminpage* dan *c_adminpage* bertugas menampilkan halaman awal admin.

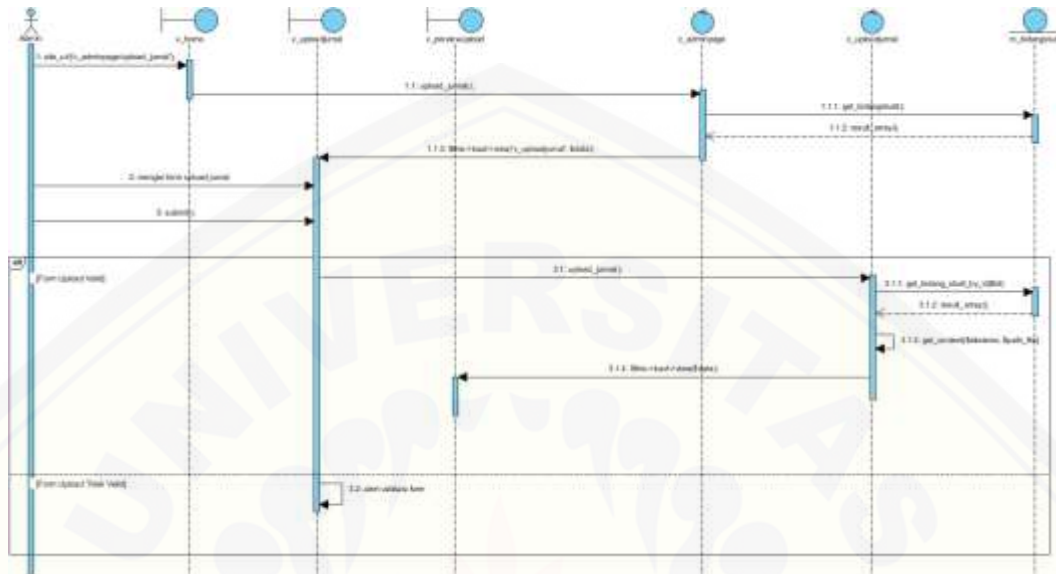


Gambar 4.12 *Sequence diagram* login mahasiswa

Gambar 4.12 merupakan *sequence diagram* yang menggambarkan alur logika pada fitur *login* dengan aktor mahasiswa. *Sequence diagram* login mahasiswa terdapat dua *view* dengan *v_login* sebagai *view default* ketika *website* pertama kali diakses. mahasiswa melakukan *input username* dan *password* pada *form login* yang ditampilkan oleh *view v_login*. *Controller* yang menangani proses *login* adalah *c_login* yang berinteraksi dengan *model m_user*. Pada saat proses pengecekan berhasil dilakukan, maka sistem otomatis membuat penyimpanan sementara pada *model m_session*. *Controller c_login* juga berfungsi melakukan pengecekan terhadap level *user* yang didapat dari *model m_user*, *c_login* akan melakukan proses *redirect* sesuai dengan level *user* yang digunakan pada saat *login*. Level *user* yang digunakan pada *sequence diagram* login mahasiswa adalah *user* mahasiswa yang berarti *c_login* akan berinteraksi dengan *controller c_userpage* dan *c_userpage* bertugas menampilkan halaman awal mahasiswa.

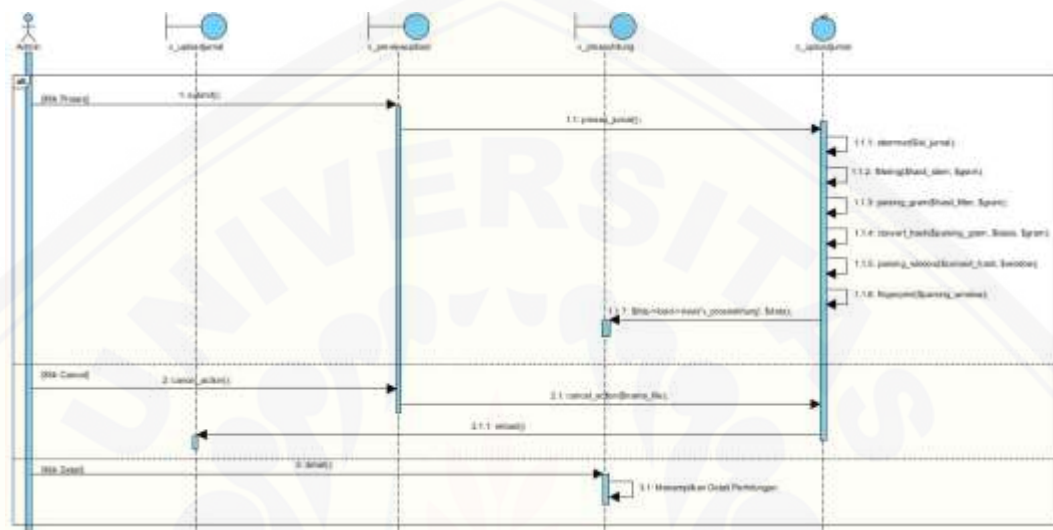
b. *Sequence Diagram Register User Baru*Gambar 4.13 *Sequence diagram register user*

Gambar 4.13 merupakan *sequence diagram* yang menggambarkan alur logika pada fitur *register user* baru oleh mahasiswa. *Sequence diagram register user* menggunakan *view* *v_registration* untuk menampilkan *form sign up* yang harus dilengkapi oleh *user* baru. Selanjutnya, *input* pada *form sign up* akan diproses oleh *controller* *c_login* ketika *user* melakukan proses *submit*. Proses selanjutnya *controller* *c_login* berinteraksi dengan *model* *m_user* untuk menyimpan data *user* baru ke dalam *database* sistem. Penanganan *exception* berupa *input* data tidak *valid* atau *input* kosong akan ditampilkan langsung oleh *view* *v_registration*. Setelah proses *register user* berhasil, maka *controller* *c_login* akan menampilkan *view* *v_login*.

c. *Sequence Diagram* Manajemen Data Jurnal – Tambah Data JurnalGambar 4.14 *Sequence diagram* upload jurnal

Gambar 4.14 merupakan *sequence diagram* yang menggambarkan alur logika pada fitur *upload* data jurnal baru oleh admin. *Controller* *c_adminpage* bertugas menampilkan halaman *view* *v_uploadjurnal* ketika *user* memilih menu *upload* jurnal yang ada pada *view* *v_home*. *View* *v_uploadjurnal* menampilkan *form upload* yang akan digunakan oleh *user* untuk memasukkan data informasi jurnal yang akan diunggah. Selanjutnya ketika *user* melakukan proses *submit form upload*, maka sistem akan mengakases *controller* *c_uploadjurnal* dan melakukan proses *upload* jurnal ke dalam direktori *website* dan menampilkan konten serta detail informasi jurnal pada *view* *v_previewupload*. Penanganan alur pengecualian berupa *input* data tidak *valid* akan ditampilkan langsung pada *view* *v_uploadjurnal*. Selain menggunakan *view* dan *controller*, pada proses ini juga menggunakan *model* yang berinteraksi langsung dengan *database* sistem. *Model* yang digunakan adalah *m_session* yang bertugas mengambil data pada

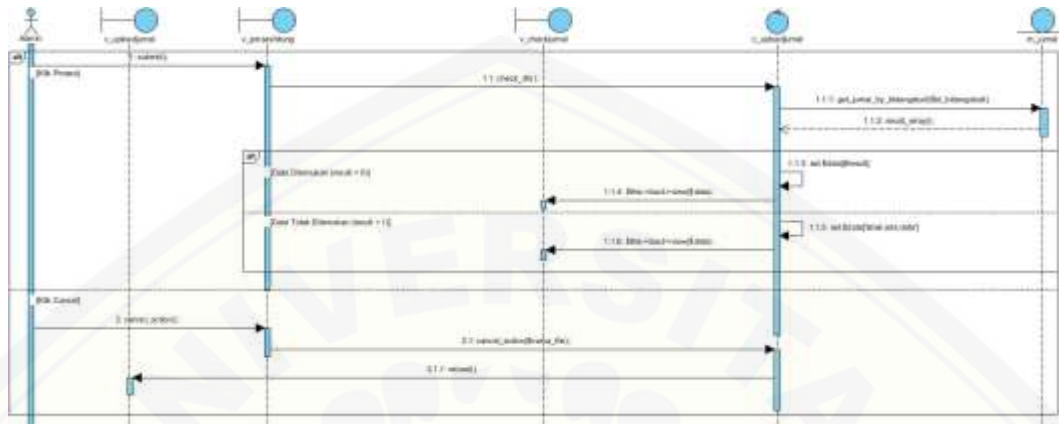
penyimpanan sementara dan `m_bidangstudi` yang bertugas berinteraksi dengan entitas yang ada pada *database*.



Gambar 4.15 *Sequence diagram* pembentukan *fingerprint*

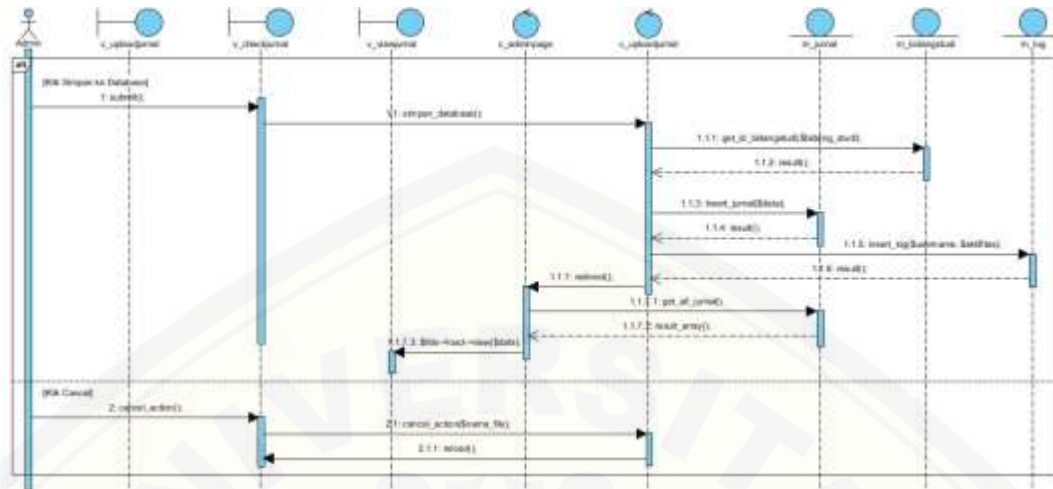
Gambar 4.15 merupakan *sequence diagram* yang menggambarkan alur logika lanjutan dari fitur *upload* jurnal. Setelah jurnal berhasil diunggah dan ditampilkan *preview* oleh sistem, jurnal akan diproses untuk membentuk *fingerprint* jurnal. Proses pembentukan *fingerprint* jurnal dikerjakan pada *controller* `c_uploadjurnal`. Beberapa proses yang dikerjakan oleh *controller* `c_uploadjurnal` mengacu pada konsep kerja algoritma *winnowing* yaitu, proses *stemming* yang bertugas mengubah tiap kata pada jurnal menjadi kata dasar, proses *filtering* yang bertugas untuk menghilangkan tanda baca, proses *parsing gram* yang bertugas menjadikan hasil *filtering* kedalam *substring-substring gram*, proses *convert hash* yang bertugas mengubah hasil proses *parsing gram* kedalam nilai *hash*, proses *parsing window* yang bertugas untuk menjadikan hasil nilai *hash* kedalam *window*, dan proses *fingerprint* yang bertugas memilih

fingerprint dari tiap *window* yang sudah dibangun. Selanjutnya hasil pembentukan *fingerprint* jurnal ditampilkan pada *view* *v_proseshitung*.



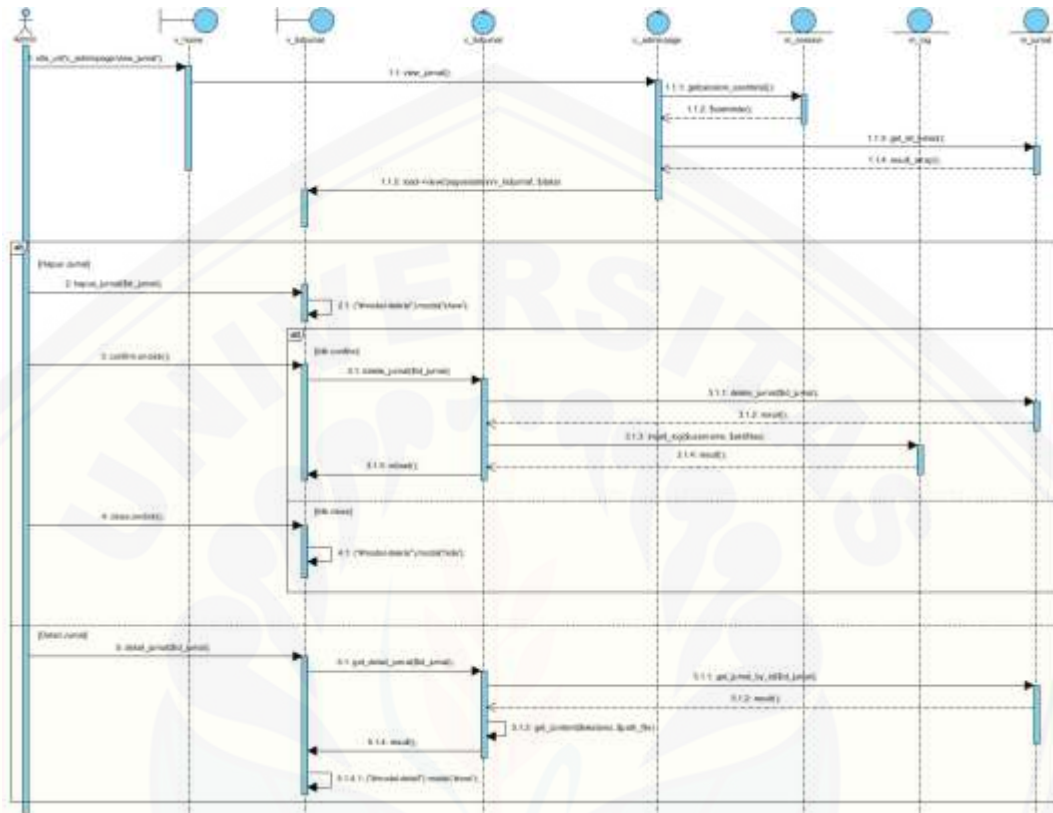
Gambar 4.16 *Sequence diagram* cek jurnal

Gambar 4.16 merupakan *sequence diagram* yang menggambarkan alur logika proses pengecekan jurnal yang telah diunggah dan telah melalui proses pembentukan *fingerprint* jurnal. Proses pengecekan bertujuan untuk menghindari terjadinya duplikasi data jurnal berdasarkan isi jurnal. Jurnal yang telah memiliki *fingerprint* diproses oleh *controller* *c_uploadjurnal*. *Controller* *c_uploadjurnal* berinteraksi dengan *model* *m_jurnal* untuk mengambil data pada *database* sesuai dengan bidang studi jurnal. *Fingerprint* antar jurnal akan dihitung dan *output* berupa persentase kemiripan dokumen yang akan ditampilkan pada *view* *v_checkjurnal*. Hasil dari proses pengecekan jurnal ini menjadi acuan untuk *user* admin apakah jurnal tersebut berhak disimpan kedalam *database* atau tidak. Proses selanjutnya adalah proses simpan data jurnal ke dalam *database* yang dapat dilihat pada gambar 4.17 berikut

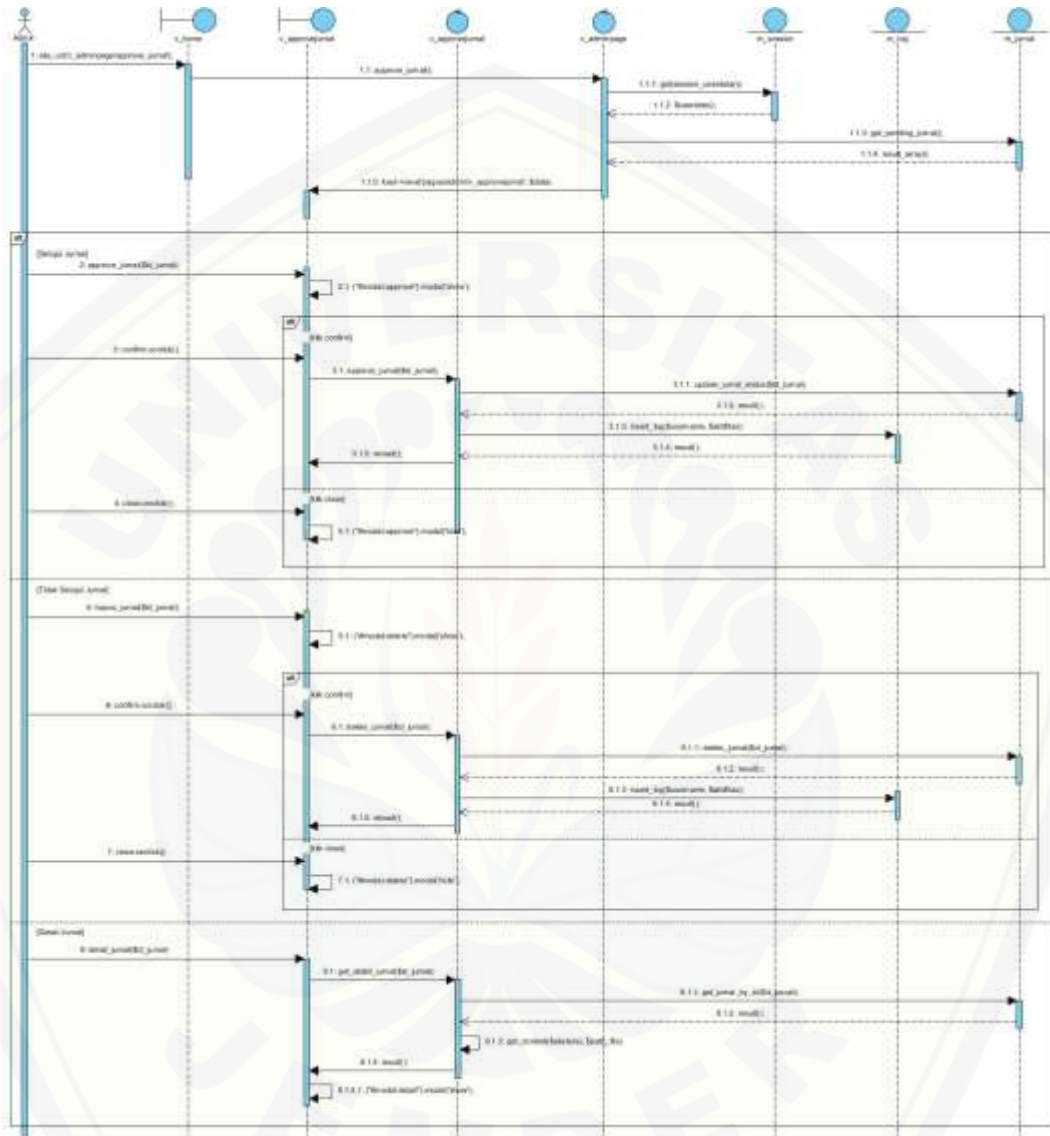


Gambar 4.17 *Sequence diagram* simpan data jurnal

Gambar 4.17 merupakan *sequence diagram* yang menggambarkan alur logika dari proses terakhir *upload* jurnal yaitu proses penyimpanan data jurnal ke dalam *database*. Proses simpan data jurnal ke dalam *database* menggunakan *controller* *c_uploadjurnal* untuk menyimpan data jurnal yang ditampilkan pada *view* *v_checkjurnal* yang kemudian *controller* tersebut berinteraksi dengan *model* *m_jurnal*. Setelah data jurnal berhasil disimpan, maka *controller* *c_uploadjurnal* melakukan proses *redirect* ke *controller* *c_adminpage* untuk menampilkan *list* jurnal yang akan ditampilkan oleh *view* *v_listjurnal* dalam bentuk tabel.

d. *Sequence Diagram* Manajemen Data Jurnal – Hapus Data JurnalGambar 4.18 *Sequence diagram* hapus data jurnal

Pada gambar 4.18 merupakan *sequence diagram* yang menggambarkan alur logika dari proses hapus data jurnal. Data jurnal yang ditampilkan oleh *view* *v_listjurnal* memiliki tombol aksi pada tiap baris data. Ketika *user* admin menekan tombol aksi hapus maka *view* *v_listjurnal* akan menampilkan *modal* konfirmasi. Proses penghapusan perlu konfirmasi, jadi setelah admin mengkonfirmasi proses penghapusan maka *controller* *c_listjurnal* akan melakukan proses penghapusan data pada *database* dengan cara mengirim *query* ke *model* *m_jurnal* yang selanjutnya *m_jurnal* akan berinteraksi dengan *database*.

e. *Sequence Diagram* Manajemen Data Jurnal – *Approve* Data JurnalGambar 4.19 *Sequence diagram* approve data jurnal

Gambar 4.19 merupakan *sequence diagram* yang menggambarkan alur logika dari proses *approve* data jurnal. Data jurnal dengan status *pending* ditampilkan oleh *view* *v_approvejurnal* dalam bentuk tabel. Proses *approve* data jurnal dilakukan oleh *controller* *c_approvejurnal* yang kemudian mengirim *query*

ke *model* m_jurnal untuk menyimpan hasil proses dan *model* m_jurnal berinteraksi dengan *database* sistem.

f. *Sequence Diagram* Nonaktifkan *User*

Sequence diagram yang menggambarkan alur logika dari proses menonaktifkan *user* oleh admin dapat dilihat pada Lampiran A3.

g. *Sequence Diagram Update* Informasi Akun – Ubah Profil *User*

Sequence diagram yang menggambarkan alur logika dari proses ubah profil *user* oleh admin dan mahasiswa dapat dilihat pada Lampiran A3.

h. *Sequence Diagram Update* Informasi Akun – Ubah *Password Login*

Sequence diagram yang menggambarkan alur logika dari proses ubah *password login* oleh admin dan mahasiswa dapat dilihat pada Lampiran A3.

i. *Sequence Diagram View Log* Aktifitas *User*

Sequence diagram yang menggambarkan alur logika dari proses *view log* aktifitas *user* oleh admin dapat dilihat pada Lampiran A3.

j. *Sequence Diagram View* Data Jurnal

Sequence diagram yang menggambarkan alur logika dari proses *view* data jurnal oleh admin dan mahasiswa dapat dilihat pada Lampiran A3.

k. *Sequence Diagram View* Data *User*

Sequence diagram yang menggambarkan alur logika dari proses *view* data *user* oleh admin dapat dilihat pada Lampiran A3.

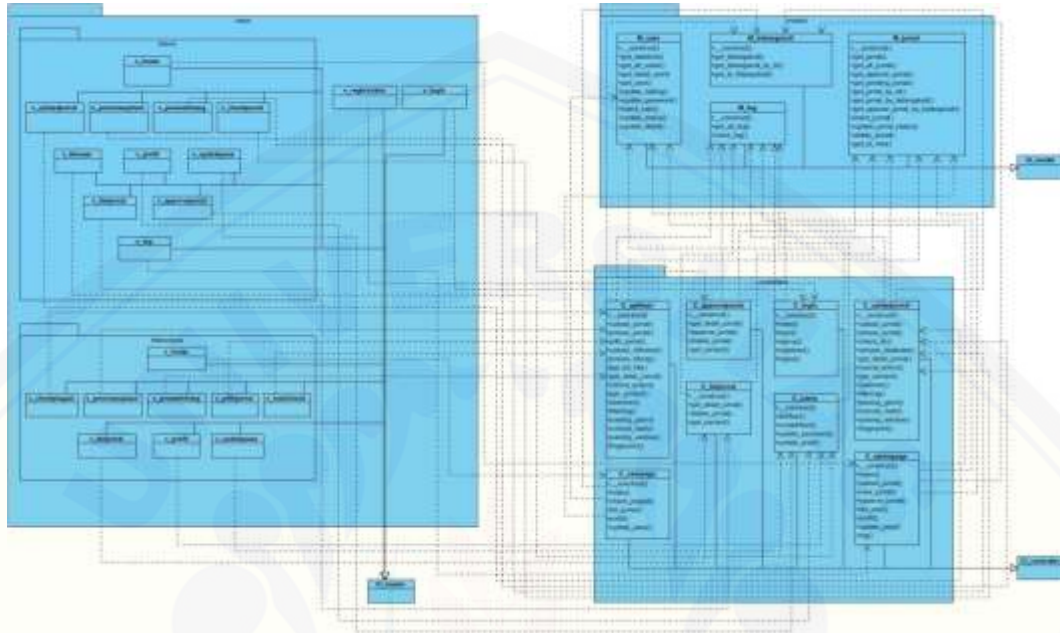
l. *Sequence Diagram* Mengecek Plagiat Jurnal

Sequence diagram yang menggambarkan alur logika dari proses mengecek plagiat jurnal oleh mahasiswa dapat dilihat pada Lampiran A3.

4.3.6 *Class Diagram*

Class diagram merupakan model statis yang menggambarkan struktur dan deskripsi *class*, *package*, dan objek serta hubungan satu sama lain seperti, pewarisan, asosiasi, dan sebagainya. *Class diagram* sistem pendeteksi *plagiarisme* dengan

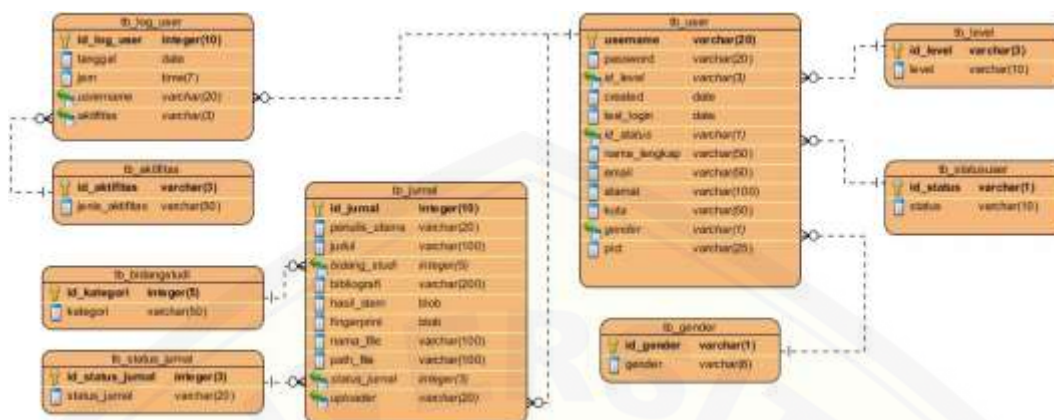
algoritma *winnowing* sebagai algoritma pencarian kemiripan dokumen dapat dilihat pada gambar 4.20 berikut



Gambar 4.20 *Class diagram*

4.3.7 *Entity Relationship Diagram (ERD)*

Entity relationship diagram merupakan model struktur data dan hubungan antar data. Objek dalam ERD biasanya digambarkan dalam sebuah entitas yang memiliki atribut-atribut yang biasanya berelasi dengan entitas lainnya. Proses pembangunan sistem pada penelitian ini menggunakan 9 entitas yang telah dinormalisasi untuk menampung data pada *database* yang direlasikan dengan relasi *one to many*. *Entity relationship diagram* sistem pendeteksi *plagiarisme* dengan algoritma *winnowing* sebagai algoritma pencarian kemiripan dokumen dapat dilihat pada gambar 4.21 berikut



Gambar 4.21 Entity relationship diagram

4.4 Implementasi Sistem

Tahapan implementasi adalah proses konversi desain sistem ke dalam kode-kode program. Sistem yang akan dibangun ditulis dengan bahasa pemrograman *Page Hypertext Pre-Processor* (PHP) dengan memanfaatkan *framework CodeIgniter* yang sudah menerapkan konsep *Object Oriented Programming* (OOP). Sistem ini juga menggunakan *local server* dan *database* untuk menyimpan data yang sewaktu-waktu dibutuhkan untuk diakses kembali. *Local Server* menggunakan aplikasi XAMPP yang sudah support *apache* untuk membangun sistem berbasis web dan *database* yang digunakan adalah MySQL (PhpMyadmin).

Implementasi dari algoritma *winnowing* dalam sistem pendeteksi *plagiarisme* terdapat pada *file controller* `c_uploadjurnal.php` dan `c_aplikasi`. Gambaran baris code keseluruhan sistem dapat dilihat pada lampiran B.

4.5 Testing

Testing atau biasa disebut pengujian sistem merupakan tahapan akhir dalam pembangunan sistem pendeteksi *plagiarisme*. Pada tahapan ini penulis menggunakan dua metode pengujian sistem yaitu *White Box Testing* dan *Black Box Testing*

4.5.1 White Box Testing

White Box Testing merupakan pengujian pada modul pengkodean program untuk menjamin kode program bebas dari kesalahan sintaks maupun logika. Dalam pengujian *white box* terdapat beberapa tahapan pembuatan dokumentasi pengujian yaitu *cyclomatic complexity* (CC), *listing* program, penentuan jalur independen, dan *test case*. Berikut adalah pengujian *white box* pada fitur tambah data jurnal yang menggunakan algoritma *winnowing*, sedangkan gambaran detail mengenai tahapan pengujian fitur lain pada sistem dengan metode *white box testing* dapat dilihat pada lampiran C1.

Pengujian *white box* pada fitur tambah data jurnal meliputi pengujian pada *method-method* yang digunakan selama proses tambah data jurnal.

a. Listing Program

```
13 function upload_jurnal()
14 {
15     $penulis = $this->input->post('penulis');
16     $judul = $this->input->post('judul');
17     $tahun = $this->input->post('tahun');
18     $studi = $this->model->get_bidangstudi_by_id($this->input->post('studi'))->row('kategori');
19     $peneliti = $this->input->post('peneliti');
20
21     if($this->input->post('volume') != null){
22         $volume = "vol. " . $this->input->post('volume') . " ";
23     }
24     $volume = "";
25
26     $nama = "";
27
28     $expl = explode(", ", $penulis);
29     if(count($expl) == "1"){
30         $nama .= $expl[0];
31     }
32     else if(count($expl) == "2"){
33         $nama .= $expl[0] . " & " . $expl[1];
34     }
35     else if(count($expl) == "3"){
36         $nama .= $expl[0] . " , " . $expl[1] . " & " . $expl[2];
37     }
38     else{
39         $nama .= $expl[count($expl)-1] . " et al.";
40     }
41
42     $bibliografi = $nama . (" $tahun ", " $judul ", " $studi ", " $volume);
```

```

54 $config['upload_path'] = 'uploads/';
55 $config['allowed_types'] = 'gif|jpg|png|txt|doc|docx';
56 $this->load->library('upload', $config);
57 if ( ! $this->upload->is_allowed('jurnal') ) {
58
59     $data['error'] = "File yang anda upload tidak diperbolehkan (hanya .txt, .doc, .docx,....)";
60     $data['upload_artif'] = "active";
61     $data['level'] = $this->session->userdata('level');
62     $data['username'] = $this->session->userdata('username');
63     $data['nama_langkap'] = $this->session->userdata('nama_langkap');
64     $data['pict'] = $this->session->userdata('pict');
65     $data['bidangstudi'] = $this->session->userdata('bidangstudi');
66
67     $this->load->view('elements/admin_header', $data);
68     $this->load->view('pages/admin/v_uploadjurnal', $data);
69 }
70
71 $ext = $this->upload->data('file_ext');
72 $path = $this->upload->data('full_path');
73
74 //data header
75 $data['upload_artif'] = "active";
76 $data['level'] = $this->session->userdata('level');
77 $data['username'] = $this->session->userdata('username');
78 $data['nama_langkap'] = $this->session->userdata('nama_langkap');
79 $data['pict'] = $this->session->userdata('pict');
80 //data header
81 $data['penulis_utama'] = $expl[count($expl)];
82 $data['judul'] = $judul;
83 $data['bidang_studi'] = $studi;
84 $data['bibliografi'] = $bibliografi;
85 $data['file_name'] = $this->upload->data('file_name');
86 $data['isi_file'] = $this->get_content($ext, $path);
87
88 $this->load->view('elements/admin_header', $data);
89 $this->load->view('pages/admin/v_previewupload', $data);
90
91 //end of upload
92
93 }

```

Gambar 4.22 Method upload_jurnal()

```

94 function proses_jurnal()
95 {
96     $hasil = 0;
97     $gram = 7;
98     $window = 6;
99
100     $penulis_utama = $this->input->post('penulis');
101     $judul = $this->input->post('judul');
102     $bidang_studi = $this->input->post('bidang_studi');
103     $bibliografi = $this->input->post('bibliografi');
104     $nama_file = $this->input->post('nama_file');
105     $isi_jurnal = $this->input->post('isi_jurnal');
106     $hasil_stem = $this->stemmer($isi_jurnal);
107
108     $hasil_filter = $this->filtering($hasil_stem);
109     $parsing_gram = $this->parsing_gram($hasil_filter, $gram);
110     $convert_hash = $this->convert_hash($parsing_gram, $gram, $hasil);
111     $parsing_window = $this->parsing_window($convert_hash, $window);
112     $fingerprint = $this->fingerprint($parsing_window);
113
114     //data header
115     $data['upload_artif'] = "action";
116     $data['level'] = $this->session->userdata('level');
117     $data['username'] = $this->session->userdata('username');
118     $data['nama_langkap'] = $this->session->userdata('nama_langkap');
119     $data['pict'] = $this->session->userdata('pict');
120
121     //data header
122     $data['judul'] = $judul;
123     $data['bidang_studi'] = $bidang_studi;
124     $data['bibliografi'] = $bibliografi;
125     $data['file_name'] = $nama_file;
126     $data['isi_file'] = $hasil_stem;
127     $data['fingerprint'] = $fingerprint;
128     $data['gram'] = $gram;
129     $data['window'] = $window;
130     $data['hasil_filter'] = $hasil_filter;
131     $data['parsing_gram'] = $parsing_gram;
132     $data['convert_hash'] = $convert_hash;
133     $data['parsing_window'] = $parsing_window;
134
135     $this->load->view('elements/admin_header', $data);
136     $this->load->view('pages/admin/v_prosesjurnal', $data);
137
138 }

```

Gambar 4.23 Method proses_jurnal()

```
181 function check_db()
182 {
183     $penulis_utama = $this->input->post('penulis');
184     $judul = $this->input->post('judul');
185     $bidang_studi = $this->input->post('bidang_studi');
186     $bibliografi = $this->input->post('bibliografi');
187     $nama_file = $this->input->post('nama_file');
188     $hasil_stem = $this->input->post('isi_jurnal');
189     $fingerprint = $this->input->post('fingerprint_jurnal');
190     $id_bidang_studi = $this->m_bidangstudi->get_id_bidangstudi($this->input->post('bidang_studi'))->row('id_bidangstudi');
191
192     if($this->m_jurnal->get_jurnal_by_bidangstudi($id_bidang_studi)->num_rows() > 0){
193         //jika ada jurnal dengan bidang studi sama di ran this plagiatnya sebelum diupload
194         $result = array();
195         $result_array = array();
196         $data_jurnal = $this->m_jurnal->get_jurnal_by_bidangstudi($id_bidang_studi);
197         $arr_fingerprint = explode(" ", $fingerprint);
198         array_push($arr_fingerprint);
199
200         foreach ($data_jurnal->result_array() as $value) {
201             $result['id_jurnal'] = $value['id_jurnal'];
202             $result['bibliografi'] = $value['bibliografi'];
203             $result['fingerprint'] = $value['fingerprint'];
204
205             //hitung persentase
206             $arr_fingerprint_jurnal = explode(" ", $value['fingerprint']);
207             array_push($arr_fingerprint_jurnal);
208
209             $intersect1 = array_intersect($arr_fingerprint, $arr_fingerprint_jurnal);
210             $intersect2 = array_intersect($arr_fingerprint_jurnal, $arr_fingerprint);
211             $dif1 = array_diff($arr_fingerprint, $arr_fingerprint_jurnal);
212             $dif2 = array_diff($arr_fingerprint_jurnal, $arr_fingerprint);
213             $same = count(array_intersect($arr_fingerprint_jurnal, $arr_fingerprint));
214             $jumlah = (count($dif1) + count($dif2) + $same);
215             $hasil = ($same / $jumlah) * 100;
216
217             $result['hasil'] = round($hasil, 2);
218
219             array_push($result_array, $result);
220         }
221
222         $data['result'] = $result_array;
223     }
224     else{
225         //jika tidak ada jurnal dengan bidang studi sama
226         $data['tidak_ada_data'] = "Saat ini tidak ada jurnal dengan bidang studi sama";
227     }
228
229     $data['penulis_utama'] = $penulis_utama;
230     $data['judul'] = $judul;
231     $data['bidang_studi'] = $bidang_studi;
232     $data['biblio'] = $bibliografi;
233     $data['file_name'] = $nama_file;
234     $data['isi_file'] = $hasil_stem;
235     $data['fingerprint'] = $fingerprint;
236
237     //data session
238     $data['upload_aktif'] = "active";
239     $data['level'] = $this->session->userdata('level');
240     $data['username'] = $this->session->userdata('username');
241     $data['nama_lengkap'] = $this->session->userdata('nama_lengkap');
242     $data['pic'] = $this->session->userdata('pic');
243     //data header
244
245     $this->load->view('elements/admin_header', $data);
246     $this->load->view('pages/admin/check_jurnal', $data);
247 }
```

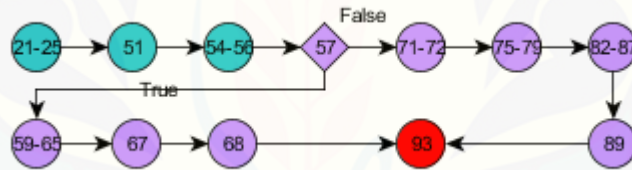
Gambar 4.24 Method check_db()


```

209 function simpan_database()
210 {
211     $penulis_utama = $this->input->post('penulis');
212     $judul = $this->input->post('judul');
213     $id_bidang_studi = $this->bidangstudi->get_id_bidangstudi($this->input->post('bidang_studi'))->row('id_kategori');
214     $bibliografi = $this->input->post('bibliografi');
215     $hasil_stem = $this->input->post('isi_jurnal');
216     $fingerprint = $this->input->post('fingerprint_jurnal');
217     $nama_file = $this->input->post('nama_file');
218     $path_file = "uploads/" . $nama_file;
219     $status_jurnal = 1;
220     $uploader = $this->session->userdata('username');
221
222     $data = array(
223         'id_jurnal' => '',
224         'penulis_utama' => $penulis_utama,
225         'judul' => $judul,
226         'bidang_studi' => $id_bidang_studi,
227         'bibliografi' => $bibliografi,
228         'hasil_stem' => $hasil_stem,
229         'fingerprint' => $fingerprint,
230         'nama_file' => $nama_file,
231         'path_file' => $path_file,
232         'status_jurnal' => $status_jurnal,
233         'uploader' => $uploader
234     );
235
236     $this->m_jurnal->insert_jurnal($data);
237
238     $aktifitas = "0";
239     $this->m_log->insert_log($uploader, $aktifitas);
240
241     redirect(site_url('c_adminpage/view_jurnal'));
242 }
    
```

Gambar 4.25 Method *simpan_database()*

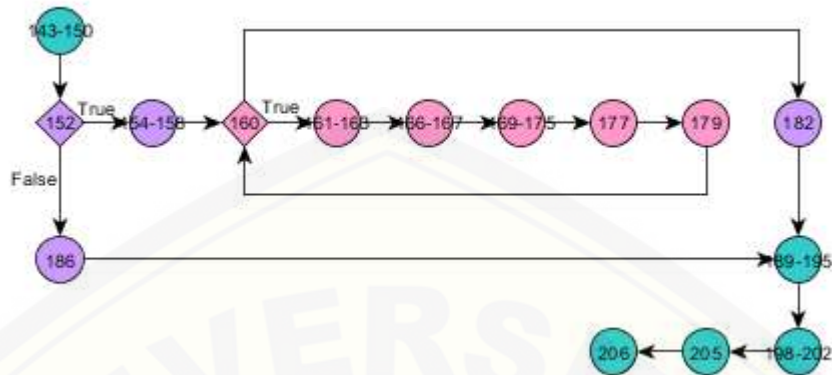
b. Diagram Alir



Gambar 4.26 Diagram alir *method upload_jurnal()*



Gambar 4.27 Diagram alir *method proses_jurnal()*



Gambar 4.28 Diagram alir *method check_db()*



Gambar 4.29 Diagram alir *method simpan_database()*

c. Perhitungan *Cyclomatic Complexity*

Perhitungan *cyclomatic complexity* $V(G)$ adalah sebagai berikut :

- Method upload_jurnal()* : $V(G) = \text{Edge} - \text{Node} + 2 = 12 - 12 + 2 = 2$
- Method proses_jurnal()* : $V(G) = \text{Edge} - \text{Node} + 2 = 6 - 7 + 2 = 1$
- Method check_db()* : $V(G) = \text{Edge} - \text{Node} + 2 = 16 - 15 + 2 = 3$
- Method simpan_database()* : $V(G) = \text{Edge} - \text{Node} + 2 = 5 - 6 + 2 = 1$

d. Basis Set

- Method upload_jurnal()* : 21-22-23-24-25-51-54-55-56-57-59-60-61-62-63-64-65-67-68-93(Path 1)
21-22-23-24-25-51-54-55-56-57-71-72-75-76-77-78-79-82-83-84-85-86-87-89-93(Path 2)
- Method proses_jurnal()* : 97-98-101-102-103-104-105-106-107-109-110-

111-112-113-116-117-118-119-120-123-135-137-138

Method check_db(): 143-144-145-146-147-148-149-150-152-154-155-156-157-158-160-161-162-163-166-167-169-170-171-172-173-174-175-177-179-182-189-190-191-192-193-194-195-198-199-200-201-202-205-206(Path 1)

143-144-145-146-147-148-149-150-152-186-189-190-191-192-193-194-195-198-199-200-201-202-205-206(Path 2)

Method simpan_database(): 211-212-213-214-215-216-217-218-219-220-222-236-238-239-241

e. *Test Case*

Tabel 4.8 *Test case* fitur tambah data jurnal

<i>Method upload_jurnal()</i>	
<i>Test case</i>	Jika <i>file</i> yang diunggah selain ekstensi “txt, doc, docx”
Target yang diharapkan	Proses berhenti dan menampilkan pesan error
Hasil Pengujian	Benar
Path	21-22-23-24-25-51-54-55-56-57-59-60-61-62-63-64-65-67-68-93(Path 1)
<i>Method upload_jurnal()</i>	
<i>Test case</i>	Jika berhasil upload jurnal
Target yang diharapkan	Menampilkan <i>preview upload</i> jurnal
Hasil Pengujian	Benar
Path	21-22-23-24-25-51-54-55-56-57-71-72-75-76-77-78-79-82-83-84-85-86-87-89-93(Path 2)
<i>Method proses_jurnal()</i>	
<i>Test case</i>	Proses pembentukan <i>fingerprint</i> jurnal dengan algoritma <i>winnowing</i>
Target yang diharapkan	Pembentukan <i>fingerprint</i> berhasil dan menampilkan hasilnya
Hasil Pengujian	Benar
Path	97-98-101-102-103-104-105-106-107-109-110-111-112-113-116-117-118-119-120-123-135-137-138

<i>Method check_db()</i>	
<i>Test case</i>	Jika pada <i>database</i> ada file jurnal dengan bidang studi sama dengan yang diunggah
Target yang diharapkan	Melakukan pengecekan dengan semua file jurnal pada <i>database</i> dengan bidang studi sama
Hasil Pengujian	Benar
Path	143-144-145-146-147-148-149-150-152-154-155-156-157-158-160-161-162-163-166-167-169-170-171-172-173-174-175-177-179-182-189-190-191-192-193-194-195-198-199-200-201-202-205-206(Path 1)
<i>Method check_db()</i>	
<i>Test case</i>	Jika pada <i>database</i> tidak ada file jurnal dengan bidang studi sama dengan yang diunggah
Target yang diharapkan	Menampilkan pesan tidak ada jurnal
Hasil Pengujian	Benar
Path	143-144-145-146-147-148-149-150-152-186-189-190-191-192-193-194-195-198-199-200-201-202-205-206(Path 2)
<i>Method simpan_database()</i>	
<i>Test case</i>	Jika simpan data jurnal berhasil
Target yang diharapkan	Menyimpan data jurnal ke <i>database</i>
Hasil Pengujian	Benar
Path	211-212-213-214-215-216-217-218-219-220-222-236-238-239-241

4.5.1 Black Box Testing

Black Box Testing merupakan pengujian yang menekankan pada pengujian fungsionalitas sistem agar keluaran sesuai dengan apa yang diharapkan pengguna. Dokumentasi hasil pengujian dengan metode *black box testing* dapat dilihat pada lampiran C2.

BAB 6 PENUTUP

Pada bab ini merupakan bagian akhir di dalam penulisan skripsi, berisi tentang kesimpulan dan saran. Kesimpulan yang ditulis merupakan kesimpulan dari hasil penelitian yang telah dilakukan dan saran yang diharapkan dapat digunakan sebagai acuan untuk penelitian selanjutnya.

6.1 Kesimpulan

Kesimpulan dari penelitian yang telah dilakukan adalah sebagai berikut :

1. Sistem dengan pengimplementasian algoritma *winnowing* sebagai algoritma *document fingerprint* untuk mengidentifikasi *plagiarisme* pada *file* dokumen berbasis teks berguna untuk memberikan informasi tentang sejauh mana persentase plagiat pada jurnal yang kita buat terhadap daftar rujukan yang kita gunakan. Tujuan dibangun sistem ini untuk mencegah tindak plagiat baik yang disengaja maupun tidak sengaja kita lakukan sebelum jurnal kita dipublikasikan. Sistem yang dibangun memiliki dua hak akses yaitu admin dan mahasiswa. Peran utama admin dalam sistem adalah memajemen data jurnal agar data jurnal yang terdapat pada *database* terhindar dari kerusakan. Sedangkan mahasiswa dapat menggunakan sistem untuk mendapatkan informasi tentang tingkat plagiat terhadap jurnal yang dibuatnya. Perancangan dan pembangunan sistem menggunakan *software development life cycle* (SDLC) dengan mengadopsi model *iterative waterfall model*. Model pengembangan tersebut dipilih karena cocok diterapkan untuk membangun sistem yang tidak terlalu kompleks dan dengan sumber daya manusia yang terbatas. Dokumentasi perancangan dan pengembangan sistem teroganisir dengan baik sesuai dengan alur model *iterative waterfall model*.

2. Algoritma *winnowing* sebagai algoritma *document fingerprint* dan *jaccard coefficient similarity* sebagai standart pengukuran kesamaan (*similarity*) dapat diterapkan dalam mencari persentase kemiripan dokumen teks. Dokumen jurnal sebagai *input* diproses menggunakan algoritma *winnowing* dalam pembuatan *fingerprint* dokumen. *Fingerprint* antar dokumen dihitung menggunakan persamaan *jaccard coefficient similarity* untuk mendapatkan persentase kemiripan. Peneliti juga menambahkan proses *stemming* pada algoritma *winnowing* untuk mendapatkan akurasi perhitungan yang lebih tinggi. Algoritma *winnowing* dengan penambahan proses *stemming* memiliki nilai *error* lebih kecil yaitu sebesar 0,69% sedangkan algoritma *winnowing* tanpa penambahan proses *stemming* memiliki nilai *error* sebesar 1,63%.
3. Akurasi pendeteksian plagiarisme juga dipengaruhi oleh nilai *gram* dan *window*. Nilai *gram* dan *window* yang digunakan harus sesuai dengan banyaknya kata yang akan diuji. Semakin sedikit kata yang diuji, penggunaan *gram* dan *window* yang kecil akan menghasilkan nilai akurasi yang lebih baik tetapi lebih membutuhkan waktu proses yang lebih lama.

6.2 Saran

Beberapa saran dan masukan berikut diharapkan dapat memberikan perbaikan dalam penelitian selanjutnya, yaitu :

1. Penambahan proses penghilangan kata-kata yang tidak deskriptif contohnya “yang”, “dan”, “di”, “dari”, dan seterusnya untuk mendapatkan akurasi yang lebih tinggi.
2. Dokumen jurnal yang dapat diproses oleh sistem lebih beragam dengan ditambahkan dokumen berekstensi *.pdf* yang sudah sesuai dengan format penulisan jurnal (dua kolom).
3. Penentuan nilai *gram* dan *window* menggunakan metode yang lebih relevan agar didapatkan hasil akurasi yang lebih tinggi.

DAFTAR PUSTAKA

- Penjiplakan [Def.]. (2005). *Kamus Besar Bahasa Indonesia (Edisi Ketiga)*. Jakarta: Balai Pustaka.
- Arifin, A. Z., Mahendra, I. A., & Ciptaningtyas, H. T. (2009). Enhanced Confix Stripping Stemmer and Ants Algorithm for Classifying News Document in Indonesian Language. *The 5th International Conference on Information & Communication Technology and System*.
- Barnabaum, C. (t.thn.). *Plagiarism : A Student's Guide to Recognizing It and Avoiding It*.
- Cornic, P. (2008). *Software Plagiarism Detection Using Model-Driven Software Development in Eclipse Platform*. Diambil kembali dari http://studentnet.cs.manchester.ac.uk/resources/library/thesis_abstracts/MSc08/Abstracts/CornicPierre-fulltext.pdf
- Fatta, H. A. (2005). *Analisis dan Perancangan Sistem Informasi*. Yogyakarta: CV Andi Offset.
- Mall, R. (2009). *Fundamental of Software Engineering*. New Delhi: Rajkamal Electric Press.
- Mutiara, B., & Agustina, S. (2008). *Anti Plagiarism Application with Algoritmn Karp-Rabin at Thesis in Gunadarma University*.
- Naumann, F., & Melanie, H. (2010). *An Introduction to Duplicate Detection*. Canada: Morgan & Claypool Publisher.
- Nugroho, E. (2011). *Perancangan Sistem Deteksi Plagiarisme Dokumen Teks Dengan Menggunakan Algoritma Rabin-Karp*.

Republik Indonesia. (2010). Peraturan Menteri Pendidikan Nasional No. 17 Tahun 2010. Jakarta: Menteri Pendidikan Nasional Republik Indonesia.

Ridhatillah, A. (2003). Dealing With Plagiarism in the Information System Research Community: A Look at Factors that Drive Plagiarism and Ways to Adress Them. *MIS Quartely*.

Scheleimer, S., Wilkerson, D. S., & Aiken, A. (2003). Winnowing: Local Algorithm for Document Fingerprinting. *Proceedings of the 2003 ACM SIGMOD International Conference* (hal. 76-78). New York: ACM Press.

Universitas Pendidikan Indonesia. (2012). PETUNJUK TEKNIS PENCEGAHAN PLAGIAT.

van Tilborg, H. C., & Shushil, J. (2011). *Encyclopedia of Cryptography and Security*. New York: Springer Science & Business Media.

LAMPIRAN

LAMPIRAN A. PERANCANGAN SISTEM

A1. Perancangan *Scenario*

1. *Scenario* dari *Usecase* Nonaktifkan *User*

Nama <i>Usecase</i>	Nonaktifkan <i>User</i>
Aktor	Admin
Pra-Kondisi	Admin <i>login</i> ke dalam sistem
Pasca-Kondisi	Admin berhasil menonaktifkan <i>user</i>
Aktor	Sistem
Skenario Utama “Nonaktifkan <i>User</i>”	
1. Klik menu “ <i>list user</i> ”	2. Menampilkan halaman <i>list user</i> beserta data <i>user</i>
3. Klik <i>icon</i> tombol “ <i>nonaktifkan user</i> ” pada data <i>user</i>	4. Menampilkan <i>pop-up</i> konfirmasi nonaktifkan <i>user</i>
5. Klik tombol “ <i>confirm</i> ” pada <i>pop-up</i>	6. <i>Update</i> status <i>user</i> menjadi “nonaktif” dan menyimpan di database
	7. Menampilkan halaman <i>list user</i> beserta data <i>user</i>
Skenario Alternatif “Batal Nonaktifkan <i>User</i>”	
5a. Klik tombol “ <i>close</i> ” pada <i>pop-up</i>	6a. Menampilkan halaman <i>list user</i> beserta data <i>user</i>

2. *Scenario* dari *Update* Informasi Akun – Ubah Profil *User* Admin

Nama Usecase	<i>Update</i> Informasi Akun
Aktor	Admin
Pra-Kondisi	Admin <i>login</i> ke dalam sistem
Pasca-Kondisi	Admin berhasil memperbarui detail informasi diri
Aktor	Sistem
Skenario Utama “Ubah Profil <i>User</i>”	
1. Klik menu “ <i>profil</i> ”	2. Menampilkan halaman profil <i>user</i>
3. Klik tombol “ <i>ubah profil</i> ”	4. Menampilkan <i>pop-up update</i> profil beserta <i>form update</i> profil
5. Mengisi form <i>update</i> profil	
6. Klik tombol “ <i>simpan</i> ” pada <i>pop-up</i>	7. Memperbarui detail <i>user</i> pada <i>database</i>
	8. Menampilkan halaman profil <i>user</i>
Skenario Alternatif “Batal Ubah Profil <i>User</i>”	
5a. Klik tombol “ <i>batal</i> ” pada <i>pop-up</i>	6a. Menampilkan halaman profil <i>user</i>

3. *Scenario* dari *Update* Informasi Akun – Ubah Profil *User* Mahasiswa

Nama Usecase	<i>Update</i> Informasi Akun
Aktor	Mahasiswa
Pra-Kondisi	Mahasiswa <i>login</i> ke dalam sistem
Pasca-Kondisi	Mahasiswa berhasil memperbarui detail informasi diri
Aktor	Sistem
Skenario Utama “Ubah Profil <i>User</i>”	
1. Klik menu “ <i>profil</i> ”	
Dilanjutkan	

Aktor	Sistem
	2. Menampilkan halaman profil <i>user</i>
3. Klik tombol “ <i>ubah profil</i> ”	
	4. Menampilkan <i>pop-up update</i> profil beserta <i>form update</i> profil
5. Mengisi form <i>update</i> profil	
6. Klik tombol “ <i>simpan</i> ” pada <i>pop-up</i>	
	7. Memperbarui detail <i>user</i> pada <i>database</i>
	8. Menampilkan halaman profil <i>user</i>

Skenario Alternatif
“Batal Ubah Profil User”

5a. Klik tombol “ <i>batal</i> ” pada <i>pop-up</i>	
	6a. Menampilkan halaman profil <i>user</i>

4. *Scenario* dari *Update* Informasi Akun – *Ubah Password Login Admin*

Nama Usecase	<i>Update</i> Informasi Akun
Aktor	Admin
Pra-Kondisi	Admin <i>login</i> ke dalam sistem
Pasca-Kondisi	Admin berhasil memperbarui <i>password login</i>

Aktor	Sistem
Skenario Utama “Ubah Password Login”	
1. Klik menu “ <i>update password</i> ”	
	2. Menampilkan halaman <i>update password</i>
3. Mengisi <i>form update password</i>	
4. Klik tombol “ <i>update password</i> ”	
	5. Memperbarui <i>password login user</i>
	6. Menampilkan halaman <i>update password</i>

Skenario Alternatif
“Input kosong atau tidak valid”

4a. Klik tombol “ <i>update password</i> ”	
	6a. Menampilkan pesan <i>error</i> pada <i>field</i>

5. *Scenario* dari *Update Informasi Akun – Ubah Password Login Mahasiswa*

Nama Usecase	<i>Update Informasi Akun</i>
Aktor	Mahasiswa
Pra-Kondisi	Mahasiswa <i>login</i> ke dalam sistem
Pasca-Kondisi	Mahasiswa berhasil memperbarui <i>password login</i>

Aktor	Sistem
-------	--------

Skenario Utama “Ubah Password Login”

1. Klik menu “ <i>update password</i> ”	
	2. Menampilkan halaman <i>update password</i>
3. Mengisi <i>form update password</i>	
4. Klik tombol “ <i>update password</i> ”	
	5. Memperbarui <i>password login user</i>
	6. Menampilkan halaman <i>update password</i>

Skenario Alternatif “Input kosong atau tidak valid”
--

4a. Klik tombol “ <i>update password</i> ”	
	6a. Menampilkan pesan <i>error</i> pada <i>field</i>

6. *Scenario* dari *Usecase View Log Aktifitas User*

Nama Usecase	<i>View log aktifitas user</i>
Aktor	Admin
Pra-Kondisi	Admin <i>login</i> ke dalam sistem
Pasca-Kondisi	Admin berhasil melihat <i>log aktifitas user</i>

Aktor	Sistem
-------	--------

Skenario Utama “View Log Aktifitas User”

1. Klik menu “ <i>log</i> ”	
	2. Menampilkan halaman <i>log</i> aktifitas <i>user</i> beserta data <i>log</i> aktifitas <i>user</i>

Skenario Alternatif

7. *Scenario* dari *View Data Jurnal – User Admin*

Nama Usecase	<i>View Data Jurnal</i>
Aktor	Admin
Pra-Kondisi	Admin <i>login</i> ke dalam sistem
Pasca-Kondisi	Admin berhasil melihat data jurnal beserta detail data jurnal
Aktor	Sistem
Skenario Utama “View Data Jurnal”	
1. Klik menu “ <i>list jurnal</i> ”	2. Menampilkan halaman <i>list</i> jurnal beserta semua data jurnal
3. Klik <i>icon</i> tombol “ <i>detail</i> ” pada data jurnal	4. Menampilkan <i>pop-up</i> detail informasi jurnal
5. Klik tombol “ <i>close</i> ” pada <i>pop-up</i>	6. Menampilkan halaman <i>list</i> jurnal beserta semua data jurnal
Skenario Alternatif	

8. *Scenario* dari *View Data Jurnal – User Admin*

Nama Usecase	<i>View Data User</i>
Aktor	Admin
Pra-Kondisi	Admin <i>login</i> ke dalam sistem
Pasca-Kondisi	Admin berhasil melihat data <i>user</i> beserta detail data <i>user</i>
Aktor	Sistem
Skenario Utama “View Data Jurnal”	
1. Klik menu “ <i>list user</i> ”	2. Menampilkan halaman <i>list user</i> beserta data <i>user</i>

Dilanjutkan

Aktor	Sistem
3. Klik <i>icon</i> tombol “ <i>detail user</i> ” pada data <i>user</i>	
	4. Menampilkan <i>pop-up</i> detail informasi <i>user</i>
5. Klik tombol “ <i>close</i> ” pada <i>pop-up</i>	
	6. Menampilkan halaman <i>list user</i> beserta data <i>user</i>

Skenario Alternatif

9. *Scenario* dari *View Data User*

Nama Usecase	<i>View Data User</i>
Aktor	Admin
Pra-Kondisi	Admin <i>login</i> ke dalam sistem
Pasca-Kondisi	Admin berhasil melihat data <i>user</i> beserta detail data <i>user</i>

Aktor	Sistem
Skenario Utama “<i>View Data Jurnal</i>”	
1. Klik menu “ <i>list user</i> ”	
	2. Menampilkan halaman <i>list user</i> beserta data <i>user</i>
3. Klik <i>icon</i> tombol “ <i>detail user</i> ” pada data <i>user</i>	
	4. Menampilkan <i>pop-up</i> detail informasi <i>user</i>
5. Klik tombol “ <i>close</i> ” pada <i>pop-up</i>	
	6. Menampilkan halaman <i>list user</i> beserta data <i>user</i>

Skenario Alternatif

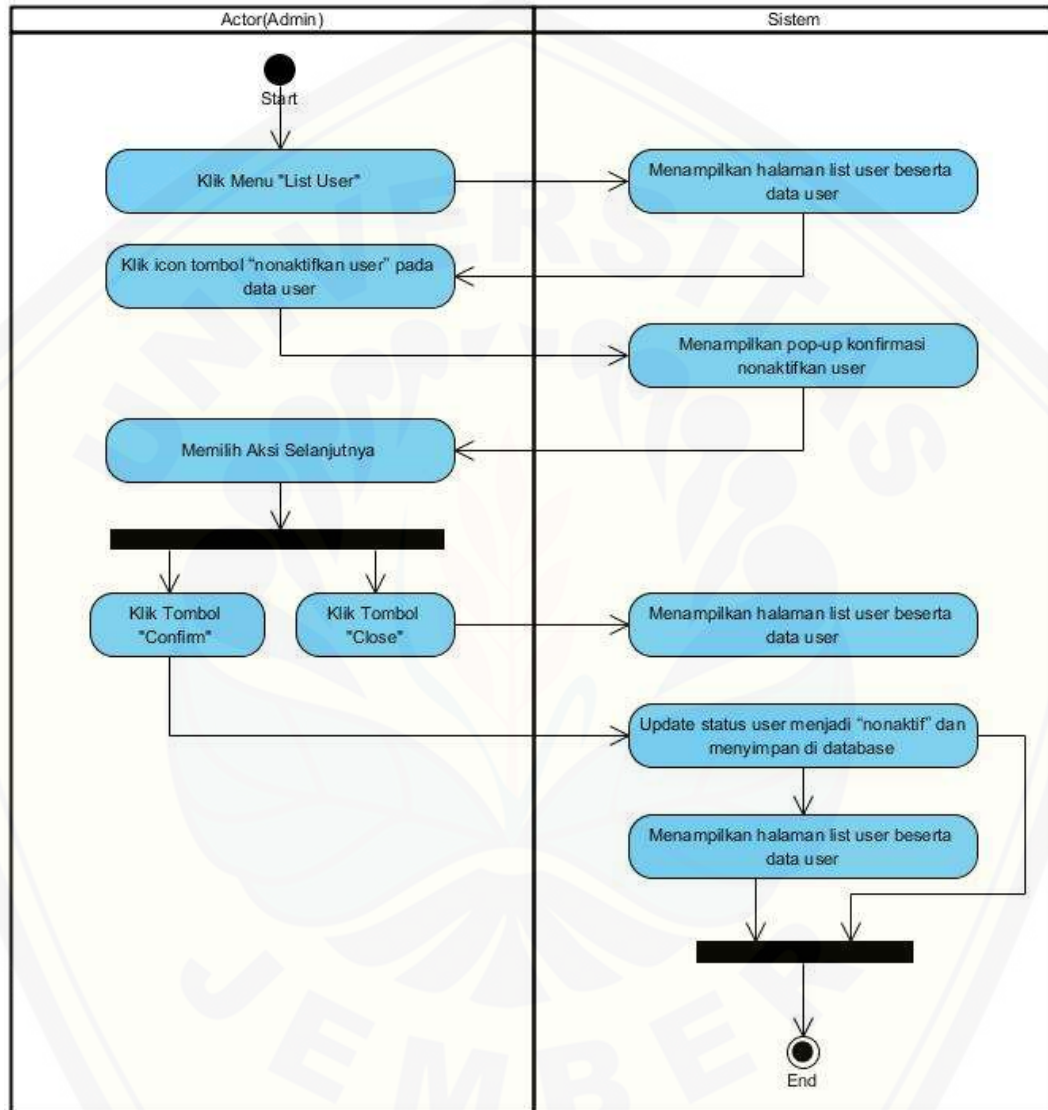
10. Scenario dari Usecase Mengecek Plagiat Jurnal

Nama Usecase	Mengecek Plagiat Jurnal
Aktor	Mahasiswa
Pra-Kondisi	Mahasiswa <i>login</i> ke dalam sistem
Pasca-Kondisi	Mahasiswa berhasil melakukan pengecekan tingkat plagiat jurnal
Aktor	Sistem
Skenario Utama “Tambah Data Jurnal Baru”	
1. Klik Menu “ <i>cek plagiat</i> ”	2. Menampilkan <i>form upload</i> jurnal yang ingin dicek
3. Mengisi <i>form upload</i> jurnal	
4. Klik tombol “ <i>upload</i> ”	5. Menyimpan <i>file</i> jurnal ke dalam direktori <i>website</i>
	6. Menampilkan <i>preview</i> jurnal yang diunggah
7. Klik tombol “ <i>proses</i> ”	8. Melakukan proses pembentukan <i>fingerprint</i> dengan algoritma <i>winnowing</i>
	9. Menampilkan hasil perhitungan
10. Klik tombol “ <i>proses</i> ”	11. Menampilkan halaman “ <i>pilih jurnal</i> ”
12. Memilih bidang studi pada <i>combobox</i> bidang studi	
13. Memilih jurnal yang ada di <i>database</i> pada <i>combobox</i> list jurnal	
14. Klik “ <i>proses</i> ”	15. Melakukan proses cek jurnal yang diunggah dengan jurnal terpilih
	16. Menampilkan persentase plagiat jurnal
Skenario Alternatif “Form Kosong atau Tidak Valid”	
Dilanjutkan	

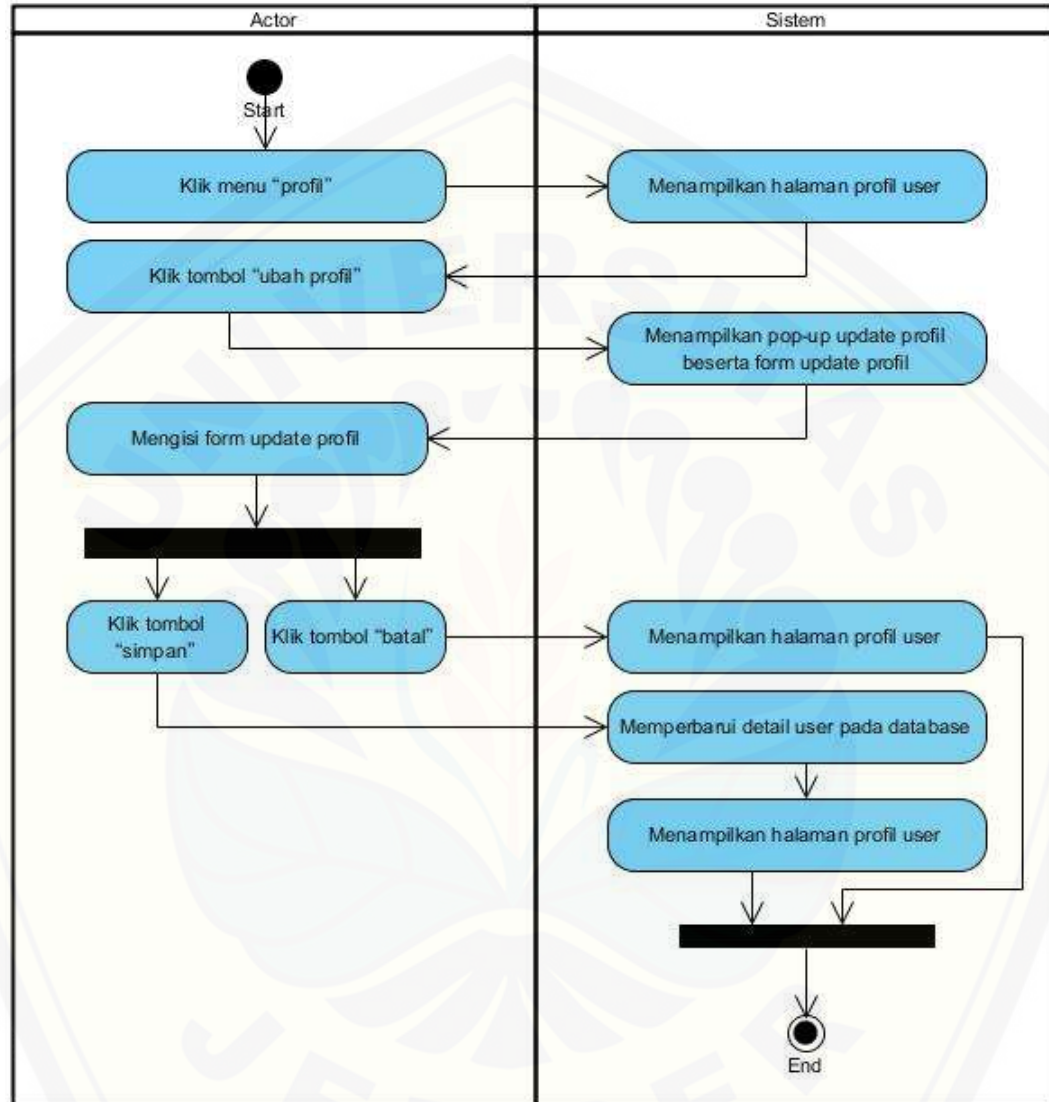
Aktor	Sistem
4a. Klik tombol “ <i>upload</i> ”	
	5a. Menampilkan pesan <i>error</i> pada <i>field</i>
Skenario Alternatif “Klik Tombol Cancel”	
7a. Klik tombol “ <i>cancel</i> ”	
	8a. Menghapus <i>file</i> jurnal dari direktori <i>website</i>
	9a. Menampilkan halaman <i>form upload</i> jurnal
Skenario Alternatif “Klik Tombol Tampilkan Detail Perhitungan”	
10a. Klik tombol “ <i>tampilkan detail perhitungan</i> ”	
	11a. Menampilkan detail perhitungan pembentukan <i>fingerprint</i> jurnal
Skenario Alternatif “Klik Tombol Cancel”	
10b. Klik tombol “ <i>cancel</i> ”	
	11b. Menghapus <i>file</i> jurnal dari direktori <i>website</i>
	12b. Menampilkan halaman <i>form upload</i> jurnal
Skenario Alternatif “Menambahkan Jurnal Referensi Manual”	
12a. Klik tombol “ <i>tambahkan</i> ”	
	14a. Menampilkan <i>pop-up upload</i> jurnal referensi
15a. Klik <i>upload</i>	
	16a. Menyimpan <i>file</i> jurnal referensi ke dalam direktori <i>website</i>
17a. Klik tombol “ <i>confirm</i> ”	
	17a. Menambahkan jurnal referensi yang diunggah ke dalam <i>list</i> jurnal referensi

A2. Perancangan Activity Diagram

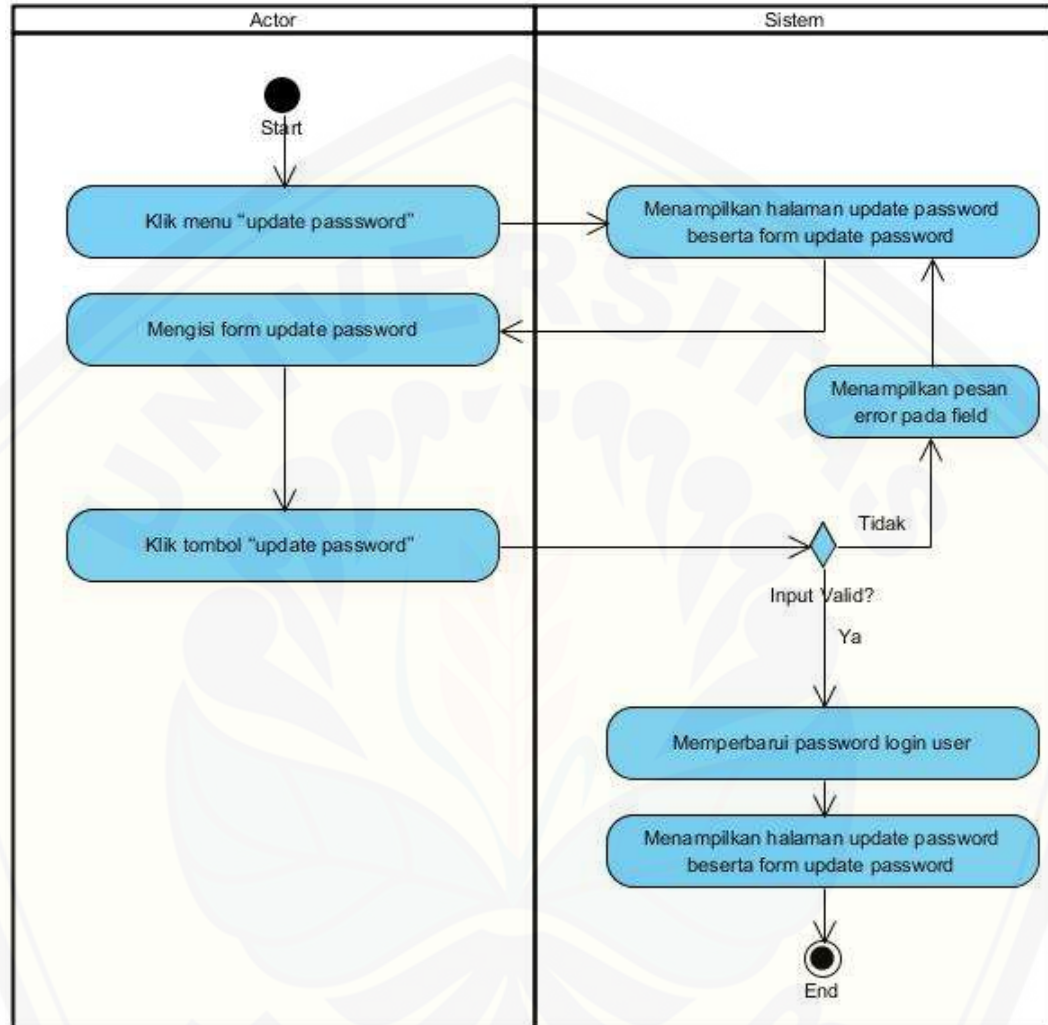
1. Activity Diagram Nonaktifkan User



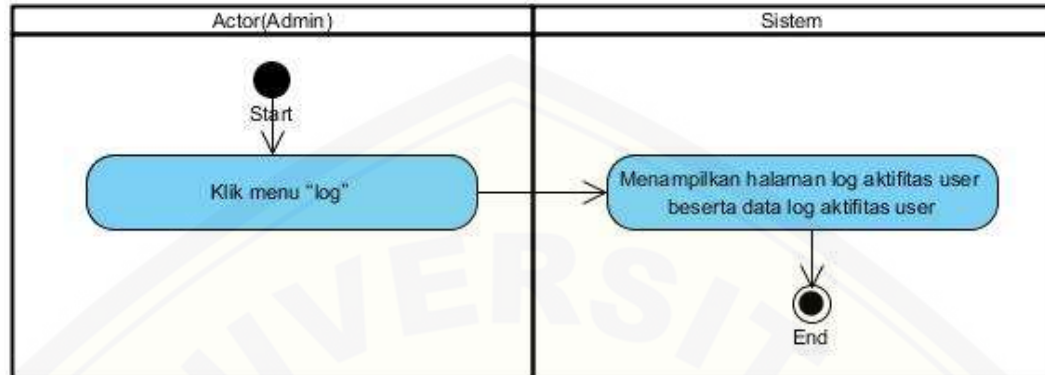
2. Activity Diagram Ubah Profil User



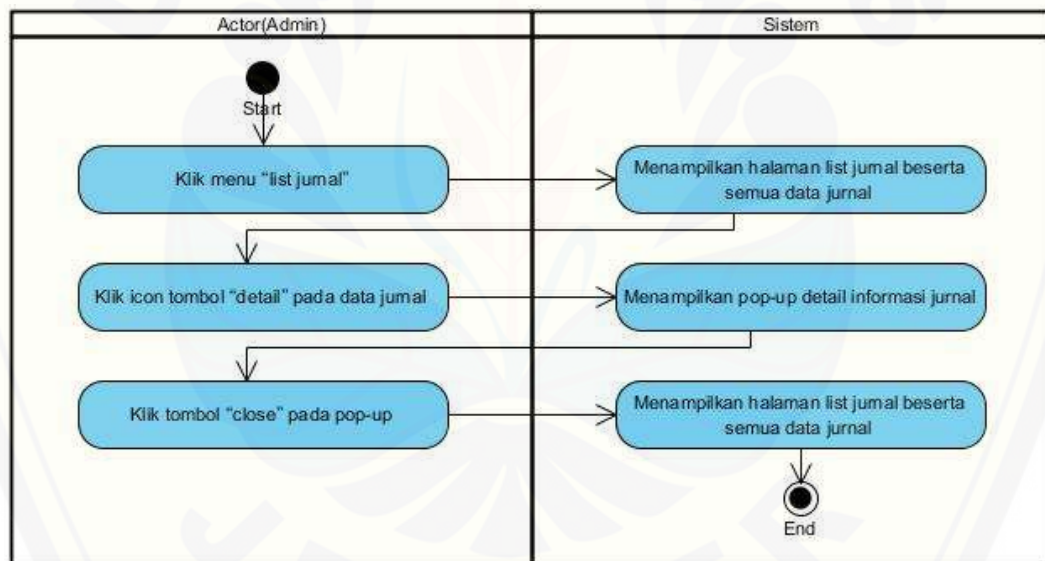
3. Activity Diagram Ubah Password Login User



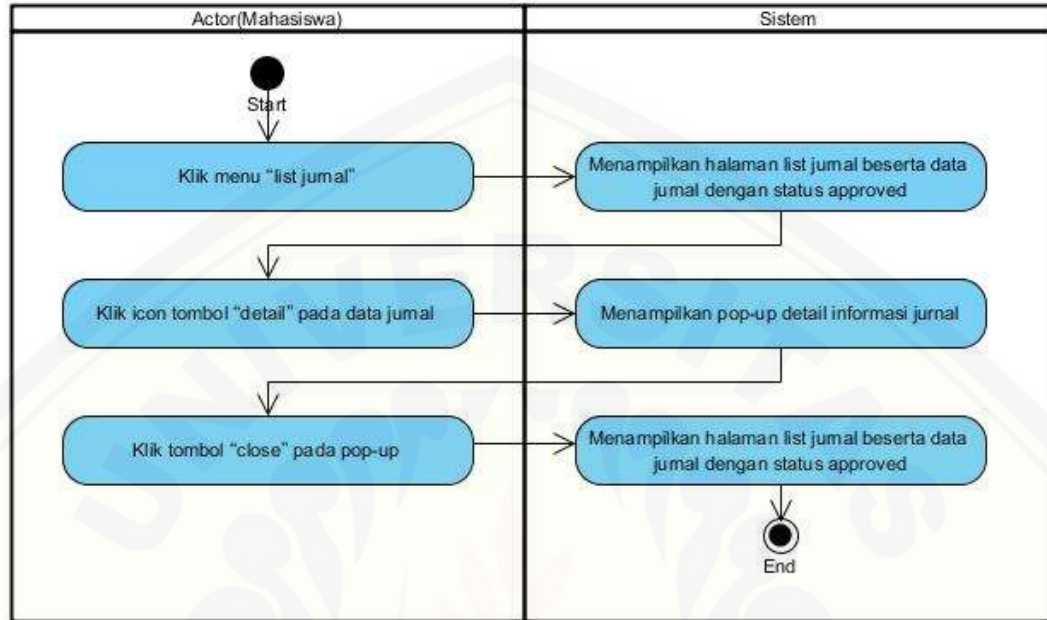
4. Activity Diagram View Log Aktifitas User



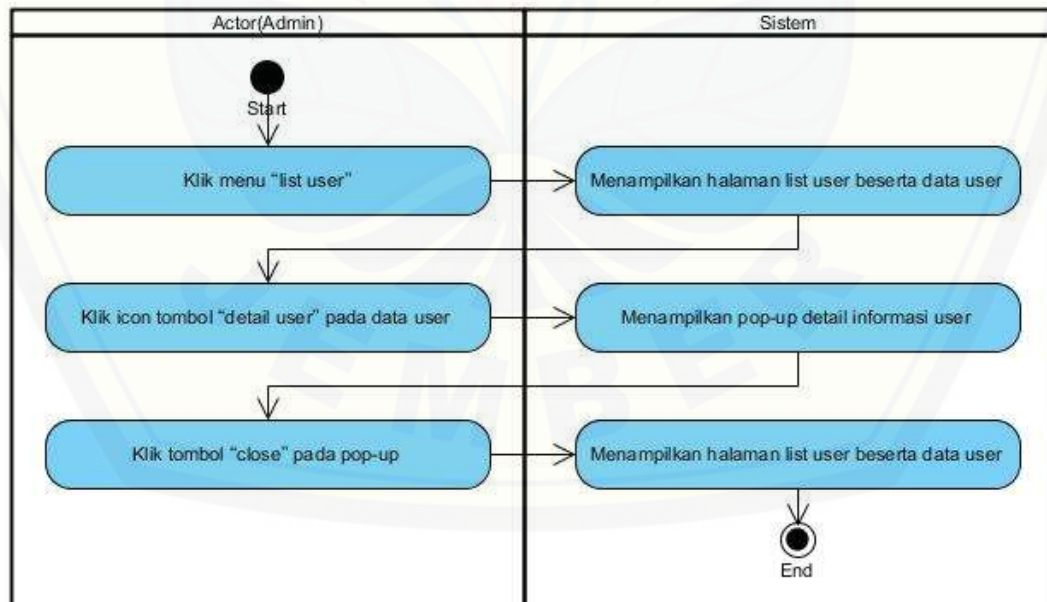
5. Activity Diagram View Data Jurnal – Admin



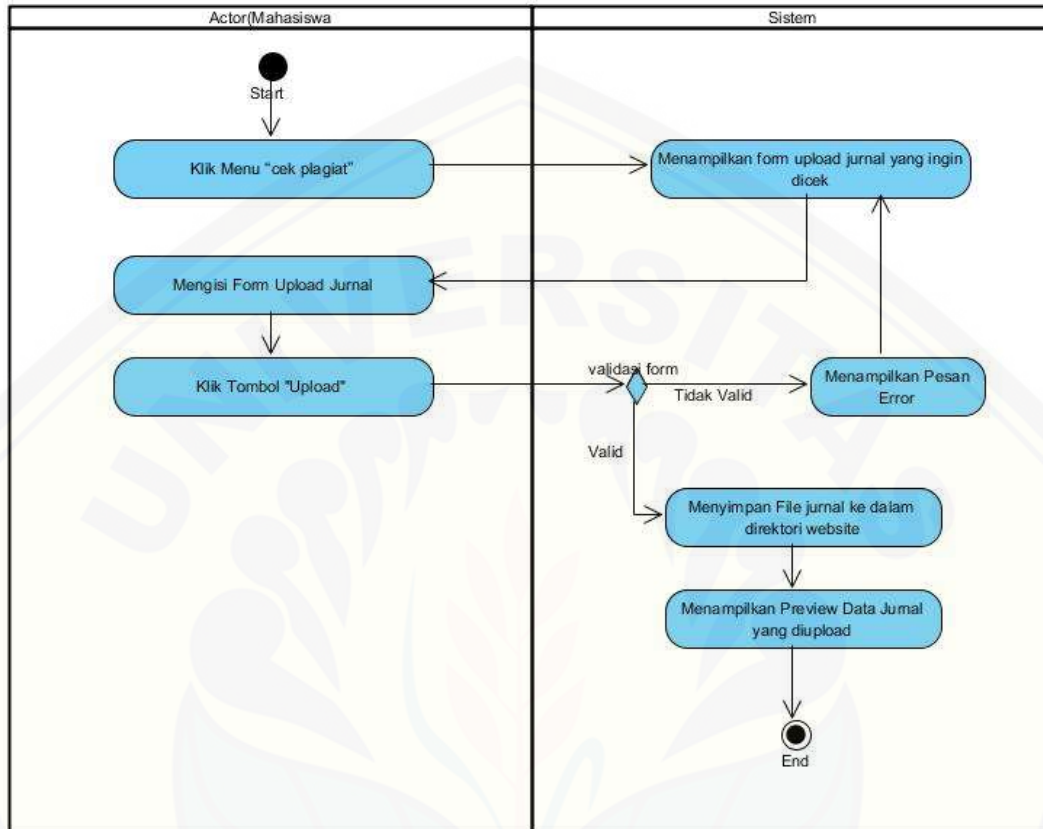
6. Activity Diagram View Data Jurnal – Mahasiswa

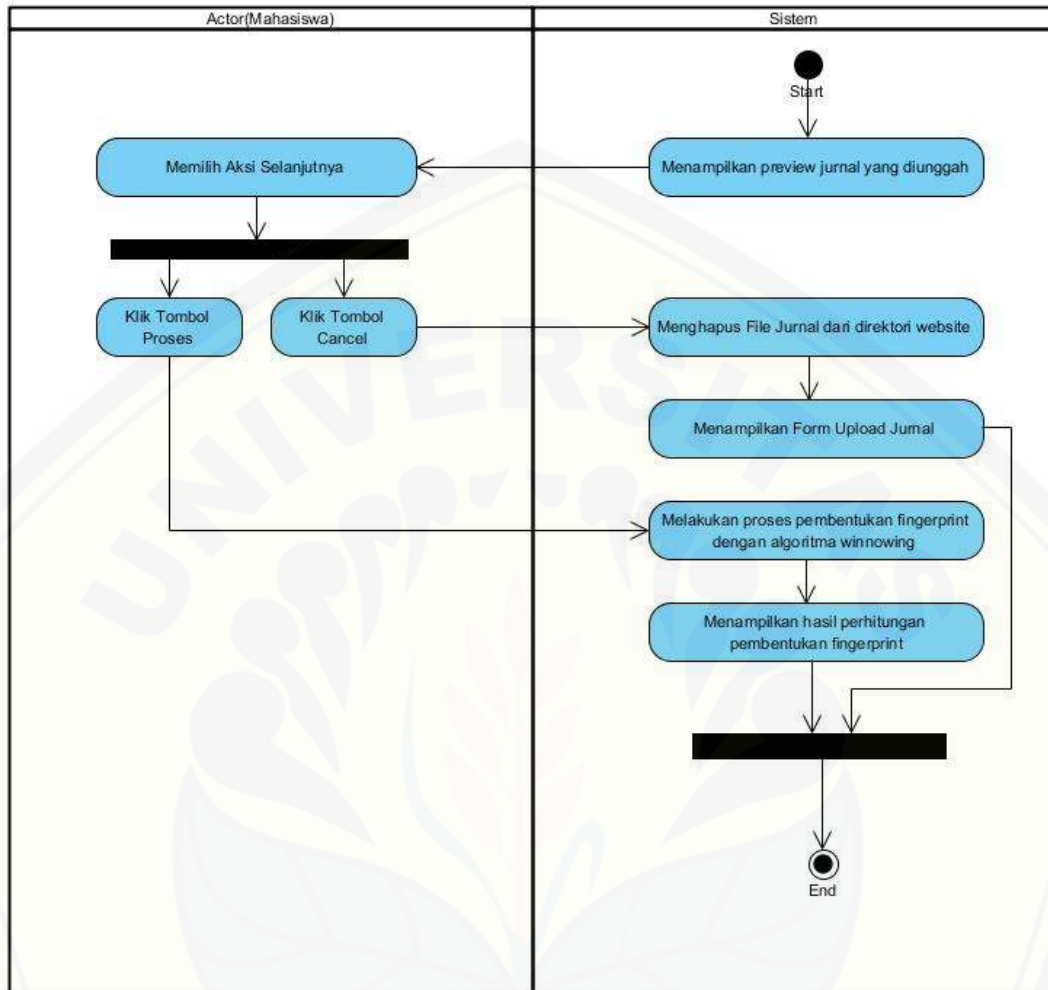


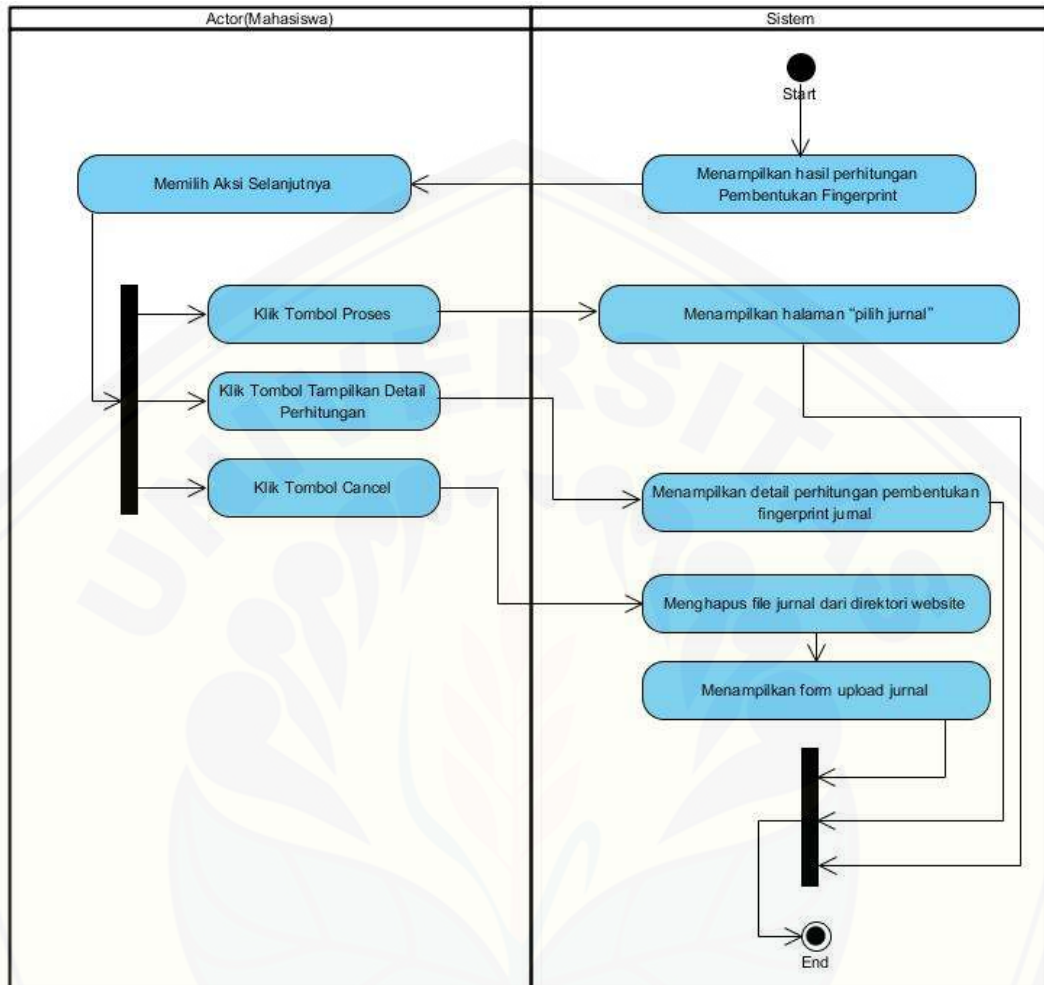
7. Activity Diagram View Data User

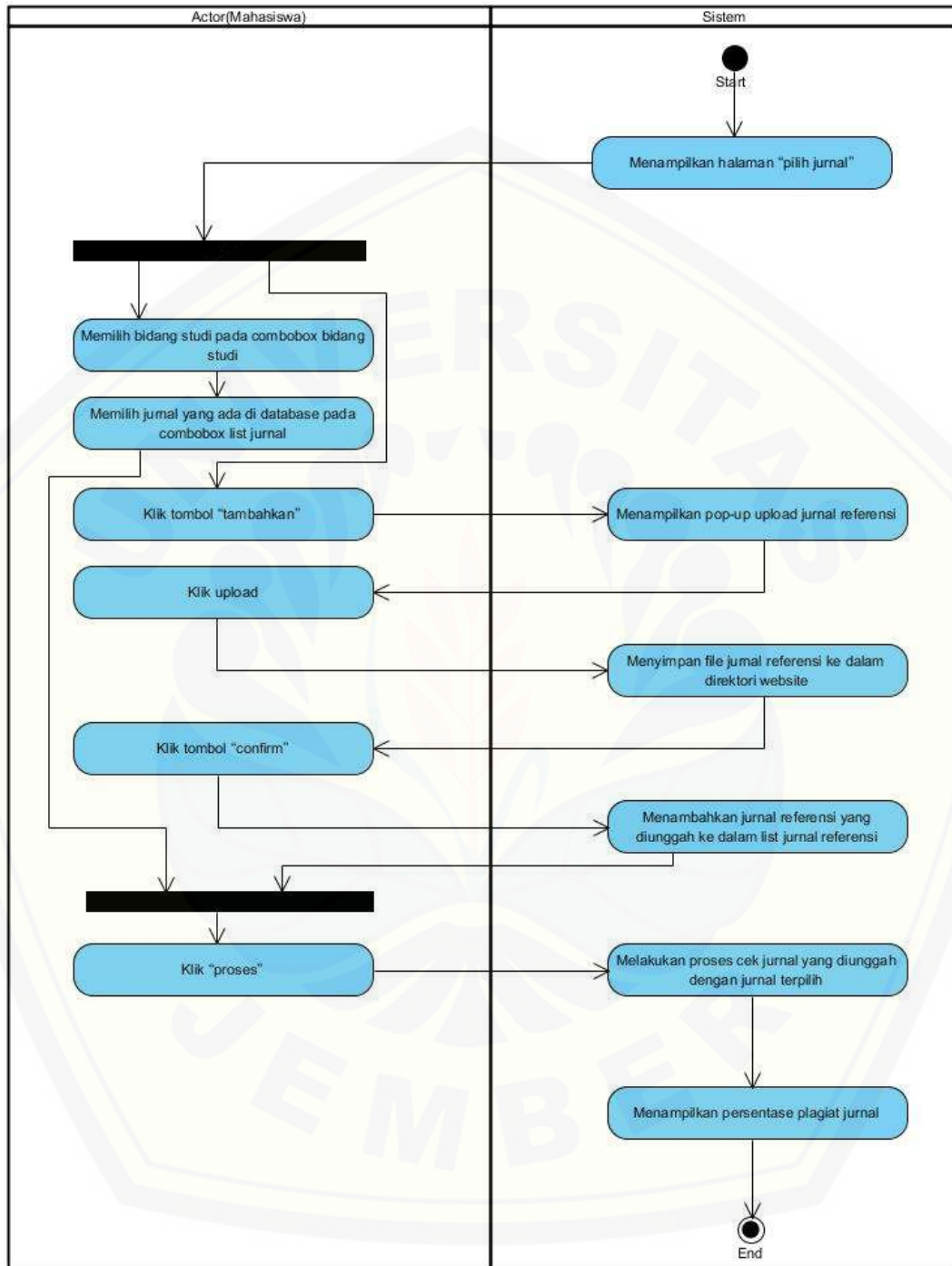


8. Activity Diagram Mengecek Plagiat Jurnal



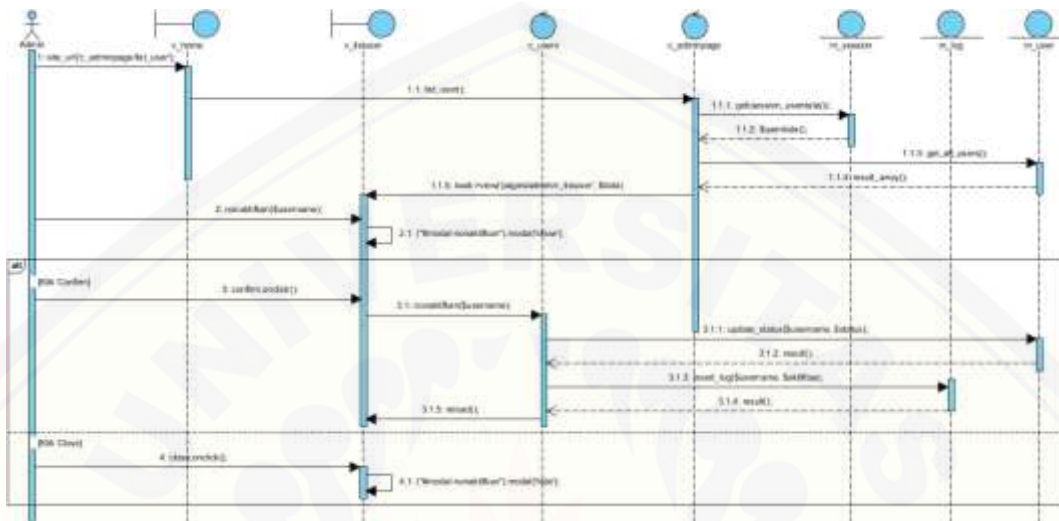




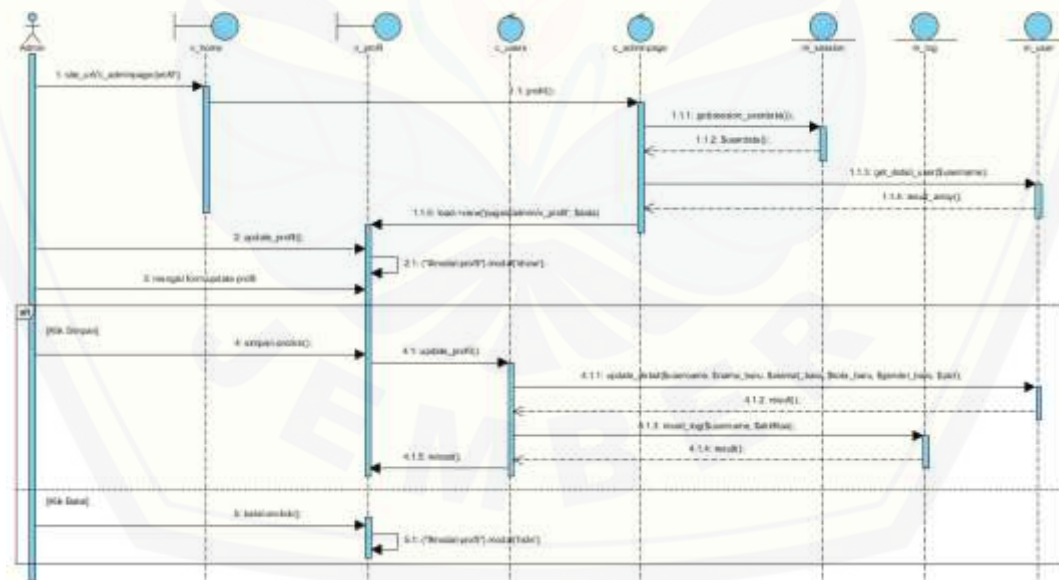


A3. Perancangan Sequence Diagram

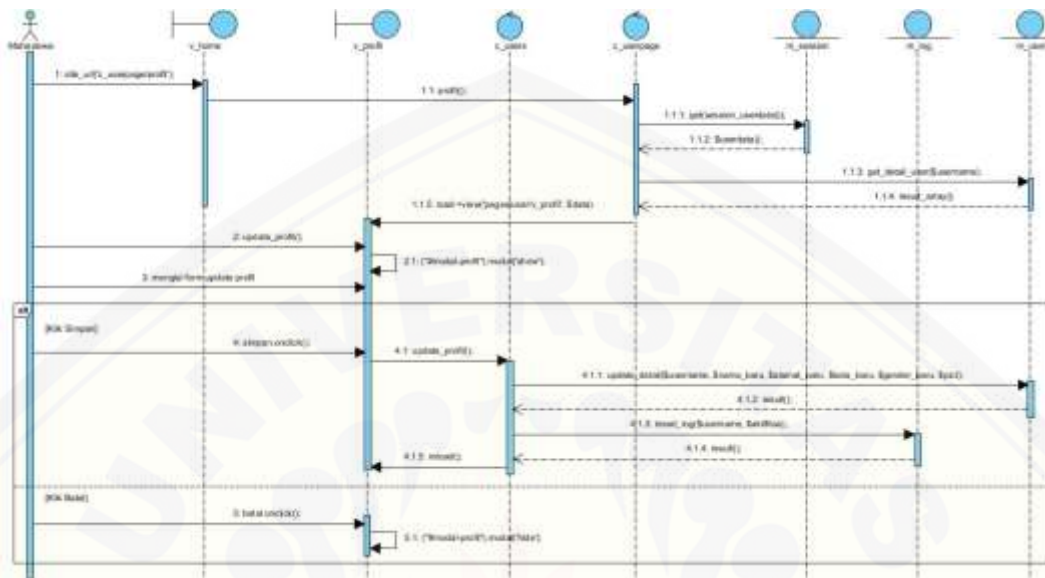
1. Sequence Diagram Nonaktifkan User



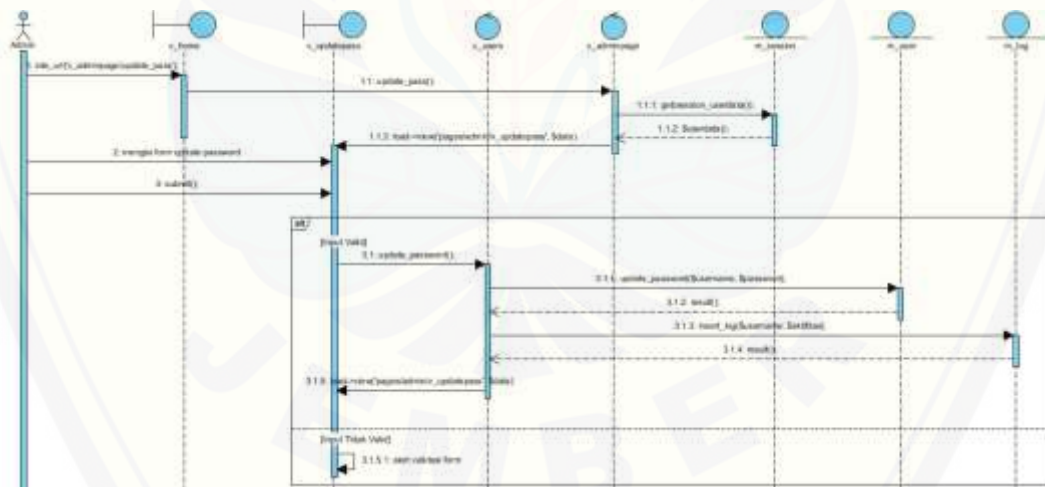
2. Sequence Diagram Ubah Profil User Admin



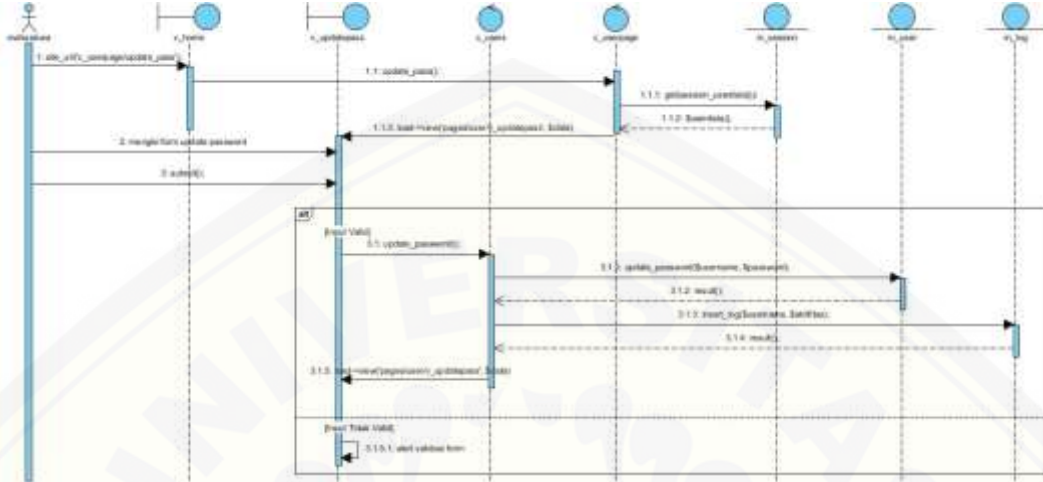
3. *Sequence Diagram* Ubah Profil User Mahasiswa



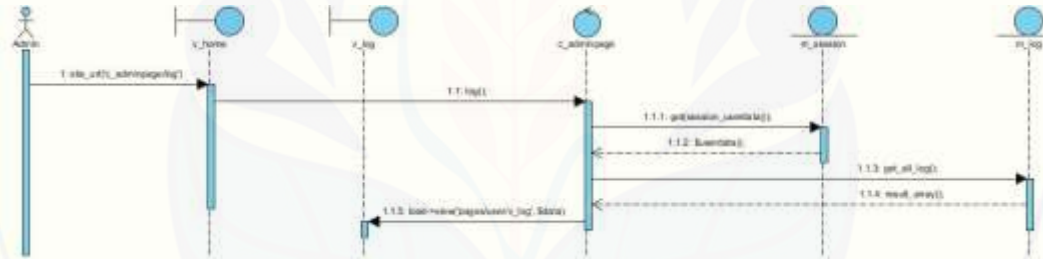
4. *Sequence Diagram* Ubah Password Login User Admin



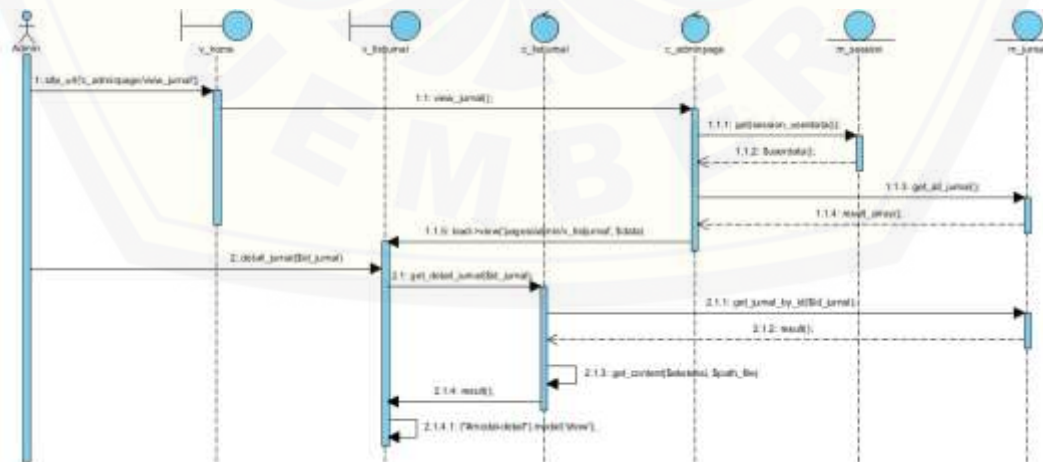
5. Sequence Diagram Ubah Password Login User Mahasiswa



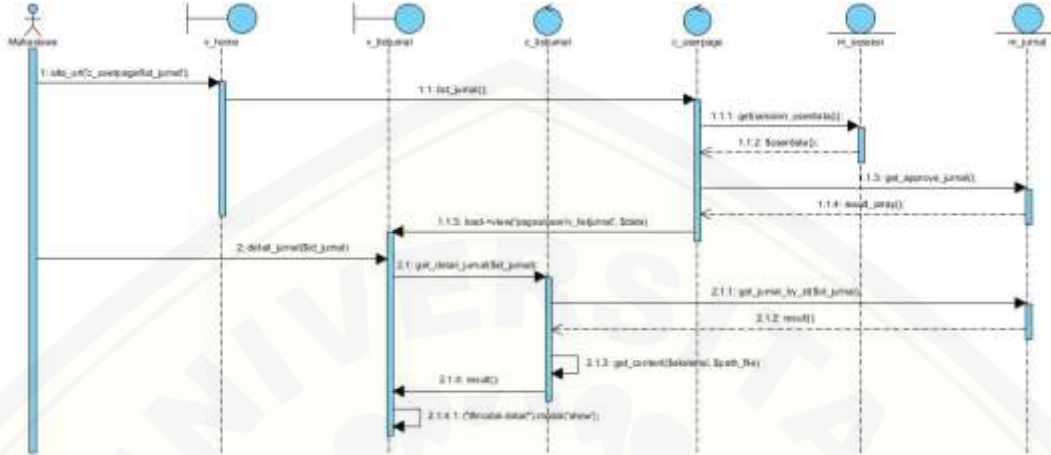
6. Sequence Diagram View Log Aktifitas User



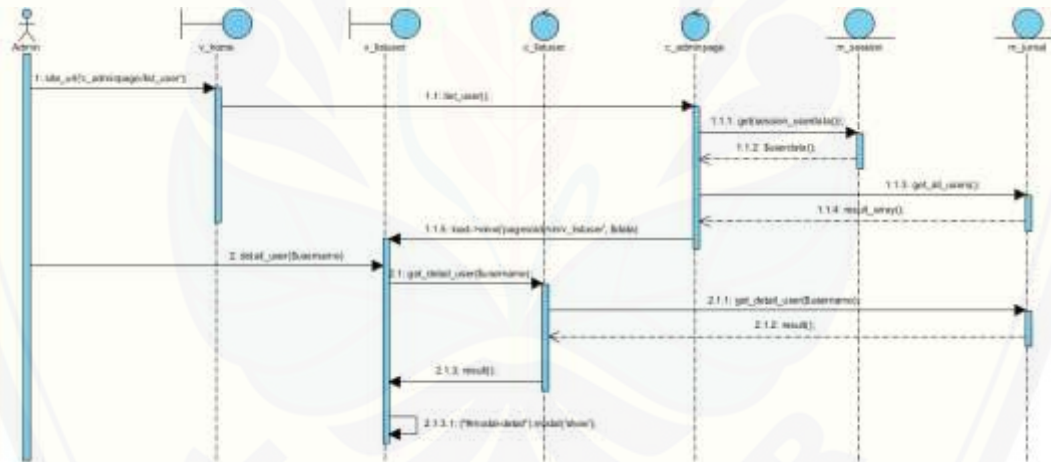
7. Sequence Diagram View Data Jurnal – Admin



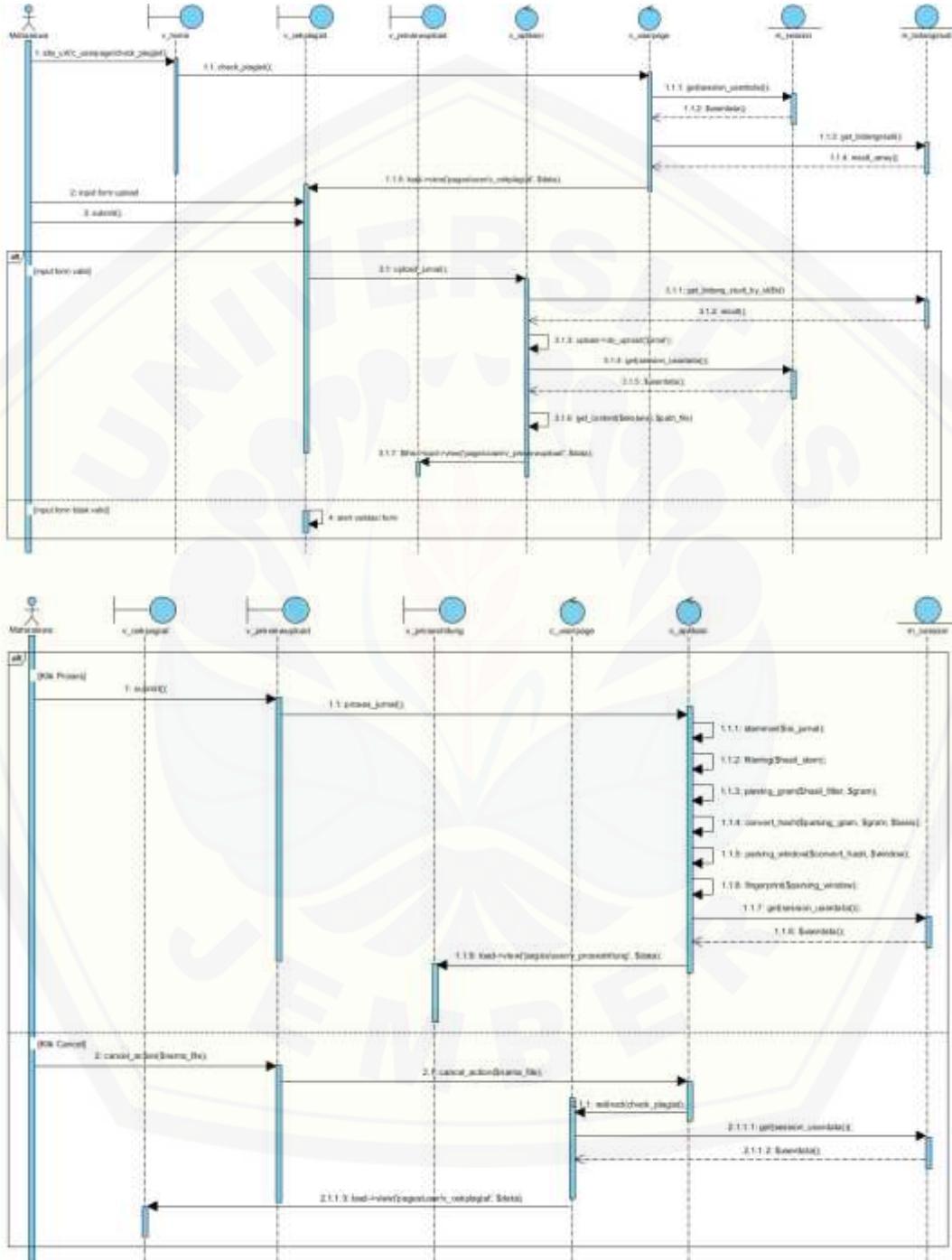
8. Sequence Diagram View Data Jurnal – Mahasiswa

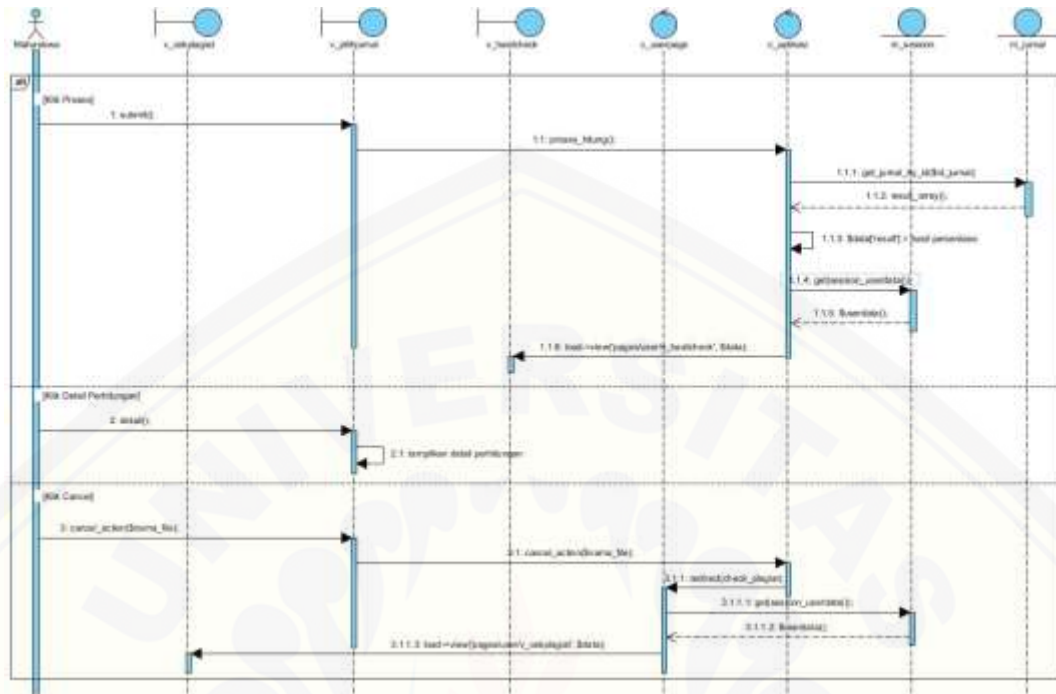


9. Sequence Diagram View Data User



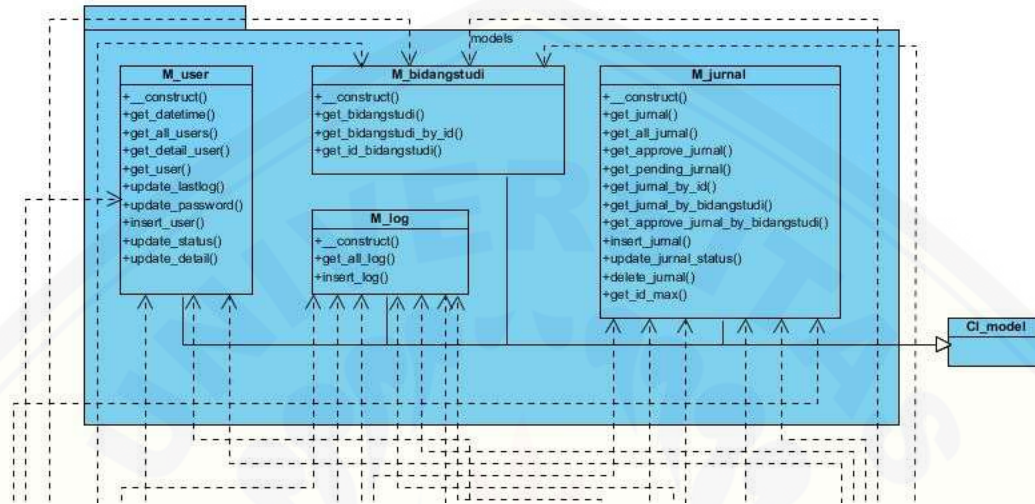
10. Sequence Diagram Mengecek Plagiat Jurnal



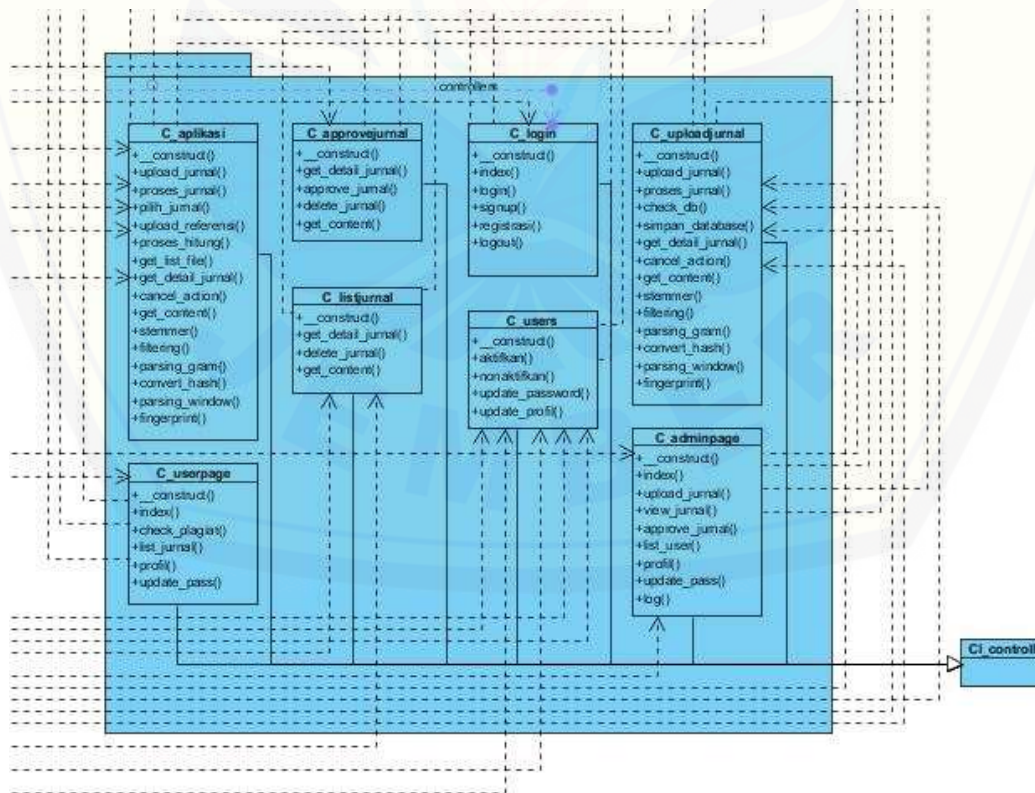


A4. Perancangan Class Diagram

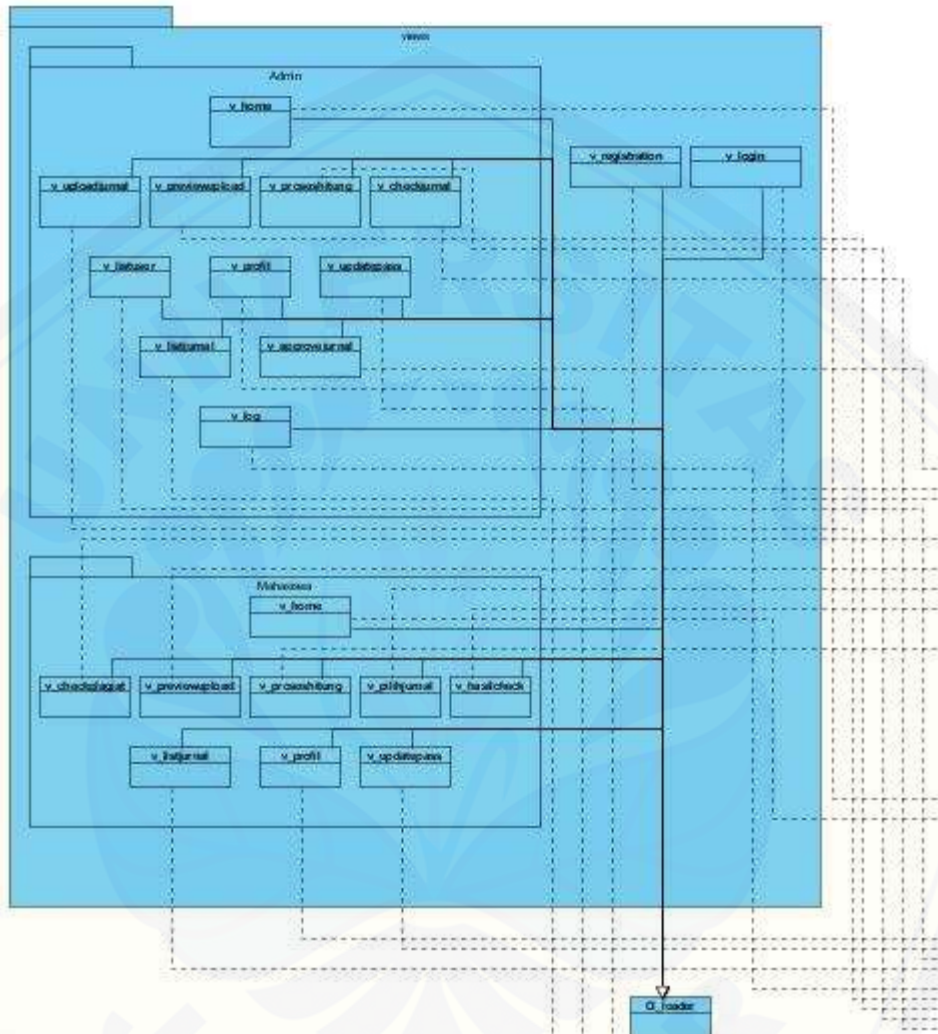
1. Model



2. Controller



3. View



LAMPIRAN B. KODE PROGRAM

1. Kode Program C_uploadjurnal.php

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

/**
 *
 */
class C_uploadjurnal extends CI_controller
{

    function __construct()
    {
        parent::__construct();
        require_once __DIR__ . '/sastrawi-
master/vendor/autoload.php';
        $this->load->model('m_bidangstudi');
        $this->load->model('m_jurnal');
        $this->load->model('m_log');
    }

    function upload_jurnal()
    {
        $penulis = $this->input->post('penulis');
        $judul = $this->input->post('judul');
        $tahun = $this->input->post('tahun');
        $studi = $this->m_bidangstudi-
>get_bidangstudi_by_id($this->input->post('studi'))-
>row('kategori');
        $penerbit = $this->input->post('penerbit');

        if($this->input->post('volume') != null){
            $volume = "Vol. ".$this->input-
>post('volume').".";
        }
        else{
            $volume = "";
        }

        $nama = "";

        $expl = explode(";", $penulis);
```

```
if(count($expl) == "2"){
    $nama .= $expl[0];
}
else if(count($expl) == "3"){
    $nama .= $expl[1].", & ".$expl[0];
}
else if(count($expl) == "4"){
    $nama .= $expl[2].", ".$expl[1].", & ".$expl[0];
}
else{
    $nama .= $expl[count($expl)-2].", et al.";
}

$bibliografi = $nama." (".$tahun."). ".$judul.".
".$studi.". ".$volume;

//upload jurnal
$config['upload_path'] = 'uploads/';
$config['allowed_types'] = 'doc|docx|txt';
$this->load->library('upload', $config);
if ( ! $this->upload->do_upload('jurnal')){
    //$error = $this->upload->display_errors();

    $data['error'] = "File yang anda upload
tidak diperbolehkan (hanya .txt, .doc, .docx)...";
    $data['upload_aktif'] = "active";
    $data['level'] = $this->session-
>userdata('level');
    $data['username'] = $this->session-
>userdata('username');
    $data['nama_lengkap'] = $this->session-
>userdata('nama_lengkap');
    $data['pict'] = $this->session-
>userdata('pict');
    $data['bidangstudi'] = $this->m_bidangstudi-
>get_bidangstudi();

    $this->load->view('elements/admin_header',
$data);
    $this->load->view('pages/admin/v_uploadjurnal',
$data);
}
else{
    $ext = $this->upload->data('file_ext');
    $path = $this->upload->data('full_path');
```

```
        //data header
        $data['upload_aktif'] = "active";
        $data['level']      = $this->session-
>userdata('level');
        $data['username']  = $this->session-
>userdata('username');
        $data['nama_lengkap'] = $this->session-
>userdata('nama_lengkap');
        $data['pict']      = $this->session-
>userdata('pict');
        //data header

        $data['penulis_utama'] = $expl[count($expl)-2];
        $data['judul']         = $judul;
        $data['bidang_studi']  = $studi;
        $data['biblio']        = $bibliografi;
        $data['file_name']     = $this->upload-
>data('file_name');
        $data['isi_file']     = $this-
>get_content($ext, $path);

        $this->load->view('elements/admin_header',
        $data);
        $this->load->view('pages/admin/v_previewupload',
        $data);
    }
    //end of upload
}

function proses_jurnal()
{
    $basis      = 3;
    $gram       = 7;
    $window     = 6;

    $penulis_utama = $this->input->post('penulis');
    $judul          = $this->input->post('judul');
    $bidang_studi  = $this->input-
>post('bidang_studi');
    $bibliografi   = $this->input->post('bibliografi');
    $nama_file     = $this->input->post('nama_file');
    $isi_jurnal    = $this->input->post('isi_jurnal');
    $hasil_stem    = $this->stemmer($isi_jurnal);
}
```

```
        $hasil_filter          =                               $this->
>filtering($hasil_stem);
        $parsing_gram          =                               $this->
>parsing_gram($hasil_filter, $gram);
        $convert_hash          =                               $this->
>convert_hash($parsing_gram, $gram, $basis);
        $parsing_window        =                               $this->
>parsing_window($convert_hash, $window);
        $fingerprint           =                               $this->
>fingerprint($parsing_window);

        //data header
        $data['upload_aktif']   = "active";
        $data['level']          =                               $this->session->
>userdata('level');
        $data['username']       =                               $this->session->
>userdata('username');
        $data['nama_lengkap']   =                               $this->session->
>userdata('nama_lengkap');
        $data['pict']           =                               $this->session->
>userdata('pict');
        //data header

        $data['penulis_utama']  = $penulis_utama;
        $data['judul']          = $judul;
        $data['bidang_studi']    = $bidang_studi;
        $data['biblio']         = $bibliografi;
        $data['file_name']       = $nama_file;
        $data['isi_file']        = $hasil_stem;
        $data['fingerprint']    = $fingerprint;
        $data['gram']           = $gram;
        $data['window']         = $window;
        $data['hasil_filter']    = $hasil_filter;
        $data['parsing_gram']    = $parsing_gram;
        $data['convert_hash']    = $convert_hash;
        $data['parsing_window'] = $parsing_window;

        $this->load->view('elements/admin_header', $data);
        $this->load->view('pages/admin/v_proseshitung',
        $data);
    }

    function check_db()
    {
        $penulis_utama          = $this->input->post('penulis');
```

```

        $judul = $this->input->post('judul');
        $bidang_studi = $this->input->post('bidang_studi');
    >post('bidang_studi');
        $bibliografi = $this->input->post('bibliografi');
        $nama_file = $this->input->post('nama_file');
        $hasil_stem = $this->input->post('isi_jurnal');
        $fingerprint = $this->input->post('fingerprint_jurnal');
    >post('fingerprint_jurnal');
        $id_bidang_studi = $this->m_bidangstudi->get_id_bidangstudi($this->input->post('bidang_studi'))->row('id_kategori');

        if($this->m_jurnal->get_jurnal_by_bidangstudi($id_bidang_studi)->num_rows() > 0){
            //jika ada jurnal dengan bidang studi sama di cek dulu plagiatnya sebelum disimpan
            $result = array();
            $result_array = array();
            $data_jurnal = $this->m_jurnal->get_jurnal_by_bidangstudi($id_bidang_studi);
            $arr_fingerprint = explode(" ", $fingerprint);
            array_pop($arr_fingerprint);

            foreach ($data_jurnal->result_array() as $value)
            {
                $result['id_jurnal'] = $value['id_jurnal'];
                $result['bibliografi'] = $value['bibliografi'];
                $result['fingerprint'] = $value['fingerprint'];

                //hitung persentasi
                $arr_fingerprint_jurnal = explode(" ", $value['fingerprint']);
                array_pop($arr_fingerprint_jurnal);

                $intersect1 = array_intersect($arr_fingerprint_jurnal, $arr_fingerprint);
                $intersect2 = array_intersect($arr_fingerprint, $arr_fingerprint_jurnal);
                $dif1 = array_diff($arr_fingerprint_jurnal, $arr_fingerprint);
                $dif2 = array_diff($arr_fingerprint, $arr_fingerprint_jurnal);
            }
        }
    }
}

```

```
        $sama =
count(array_intersect($arr_fingerprint_jurnal, $arr_fingerprint));
        $jumlah = (count($dif1)-1) +
count($dif2) + $sama;
        $hasil = ($sama / $jumlah) *
100;

        $result['hasil'] = round($hasil, 2);

        array_push($result_array, $result);
    }

    $data['result'] = $result_array;
}
else{
    //jika tidak ada jurnal dengan bidang studi sama
    $data['tidak_ada_data'] = "Saat ini tidak ada
jurnal dengan bidang studi sama";
}

    $data['penulis_utama'] = $penulis_utama;
    $data['judul'] = $judul;
    $data['bidang_studi'] = $bidang_studi;
    $data['biblio'] = $bibliografi;
    $data['file_name'] = $nama_file;
    $data['isi_file'] = $hasil_stem;
    $data['fingerprint'] = $fingerprint;

    //data header
    $data['upload_aktif'] = "active";
    $data['level'] = $this->session-
>userdata('level');
    $data['username'] = $this->session-
>userdata('username');
    $data['nama_lengkap'] = $this->session-
>userdata('nama_lengkap');
    $data['pict'] = $this->session-
>userdata('pict');
    //data header

    $this->load->view('elements/admin_header', $data);
    $this->load->view('pages/admin/v_checkjurnal', $data);
}

function simpan_database()
```



```
{
    $penulis_utama    = $this->input->post('penulis');
    $judul            = $this->input->post('judul');
    $id_bidang_studi = $this->m_bidangstudi-
>get_id_bidangstudi($this->input->post('bidang_studi'))-
>row('id_kategori');
    $bibliografi     = $this->input->post('bibliografi');
    $hasil_stem      = $this->input->post('isi_jurnal');
    $fingerprint     = $this->input-
>post('fingerprint_jurnal');
    $nama_file       = $this->input->post('nama_file');
    $path_file       = "uploads/".$nama_file;
    $status_jurnal   = 1;
    $uploader        = $this->session-
>userdata('username');

    $data = array(
        'id_jurnal' => '',
        'penulis_utama' => $penulis_utama,
        'judul'      => $judul,
        'bidang_studi' => $id_bidang_studi,
        'bibliografi' => $bibliografi,
        'hasil_stem' => $hasil_stem,
        'fingerprint' => $fingerprint,
        'nama_file' => $nama_file,
        'path_file' => $path_file,
        'status_jurnal' => $status_jurnal,
        'uploader' => $uploader
    );

    $this->m_jurnal->insert_jurnal($data);

    $aktifitas = "6";
    $this->m_log->insert_log($uploader, $aktifitas);

    redirect(site_url('c_adminpage/view_jurnal'));
}

function get_detail_jurnal($id_jurnal)
{
    $query = $this->m_jurnal-
>get_jurnal_by_id($id_jurnal);
    $result = array();

    foreach ($query->result_array() as $item) {
```

```
        $result['bibliografi'] = $item['bibliografi'];
        $result['fingerprint'] = $item['fingerprint'];
        $path_file = $item['path_file'];
    }

    $exp = explode(".", $path_file);
    $ext = ".".end($exp);

    $result['isi_file'] = $this->get_content($ext,
$path_file);

    header('Content-Type: application/x-json; charset=utf-
8');
    echo(json_encode($result));
}

function cancel_action($nama_file)
{
    $path = "uploads/".$nama_file;

    unlink($path);
}

function get_content($ext, $file_path)
{
    $isi_file = '';
    if($ext == ".txt"){
        $isi_file = file_get_contents($file_path);
    }
    else if($ext == ".docx"){
        $striped_content = '';
        $content = '';

        if(!$file_path || !file_exists($file_path)) return
false;

        $zip = zip_open($file_path);

        if (!$zip || is_numeric($zip)) return false;

        while ($zip_entry = zip_read($zip)) {

            if (zip_entry_open($zip, $zip_entry) == FALSE)
continue;
```

```

        if (zip_entry_name($zip_entry) !=
"word/document.xml") continue;

        $content .= zip_entry_read($zip_entry,
zip_entry_filesize($zip_entry));

        zip_entry_close($zip_entry);
    } // end while

    zip_close($zip);

    $content = str_replace('</w:r></w:p></w:tc><w:tc>', "
", $content);
    $content = str_replace('</w:r></w:p>', "\r\n",
$content);
    $stripped_content = strip_tags($content);

    $isi_file = $stripped_content;
}
else{
    if(file_exists($file_path)) {
        if(($fh = fopen($file_path, 'r')) != false) {
            $headers = fread($fh, 0xA00);
            $n1 = ( ord($headers[0x21C]) - 1 ); // 1 =
(ord(n)*1) ; Document has from 0 to 255 characters
            $n2 = ( ( ord($headers[0x21D]) - 8 ) * 256
); // 1 = ((ord(n)-8)*256) ; Document has from 256 to 63743
characters
            $n3 = ( ( ord($headers[0x21E]) * 256 ) * 256
); // 1 = ((ord(n)*256)*256) ; Document has from 63744 to 16775423
characters
            $n4 = ( ( ( ord($headers[0x21F]) * 256 ) * 256
) * 256 ); // 1 = (((ord(n)*256)*256)*256) ; Document has from
16775424 to 4294965504 characters
            $textLength = ($n1 + $n2 + $n3 + $n4); // Total
length of text in the document
            $extracted_plaintext = fread($fh,
$textLength);
            $extracted_plaintext =
mb_convert_encoding($extracted_plaintext, 'UTF-8');
            $isi_file = $extracted_plaintext;
        } else {
            $isi_file = 'Cannot Load File Content';
        }
    }
}
}

```

```
        else {
            $isi_file = 'File Not Exist';
        }
    }

    return $isi_file;
}

function stemmer($isi_jurnal)
{
    // create stemmer
    // cukup dijalankan sekali saja, biasanya didaftarkan
di service container
    $stemmerFactory = new
\Sastrawi\Stemmer\StemmerFactory();
    $stemmer = $stemmerFactory->createStemmer();

    // stem
    $hasil_stem = $stemmer->stem($isi_jurnal);

    return $hasil_stem;
}

function filtering($string)
{
    $filter = preg_replace("/[^a-zA-Z]/", "",
$string);
    $filter_final = strtolower($filter);

    return $filter_final;
}

function parsing_gram($filter_final, $gram)
{
    $arr1 = str_split($filter_final);

    for($i=0;$i<count($arr1); $i++){
        array_slice(array_slice($arr1, $i), 0, $gram);

        if(count(array_slice(array_slice($arr1, $i), 0,
$gram) == $gram){
            $arr2[$i] = array_slice(array_slice($arr1,
$i), 0, $gram);
        }
    }
}
```

```
        return $arr2;
    }

    function convert_hash($parsing_gram, $gram, $basis)
    {
        for($x=0;$x<count($parsing_gram); $x++){
            $sum = 0;
            for($y=0;$y<$gram; $y++){
                $pangkat = $gram-($y+1);
                $sum_temp =
ord($parsing_gram[$x][$y])*(pow($basis, $pangkat));
                $sum = $sum+$sum_temp;
            }
            $arr3[]=$sum;
        }

        return $arr3;
    }

    function parsing_window($convert_hash, $window)
    {
        for($i=0;$i<count($convert_hash); $i++){
            array_slice(array_slice($convert_hash, $i), 0,
$window);

            if(count(array_slice(array_slice($convert_hash,
$i), 0, $window)) == $window){
                $arr4[$i] =
array_slice(array_slice($convert_hash, $i), 0, $window);
            }
        }

        return $arr4;
    }

    function fingerprint($parsing_window)
    {
        $t=0;
        while($t<count($parsing_window){
            $arr5[] = min($parsing_window[$t]);
            $t++;
        }
    }
}
```

```
        $fingerprint = array_unique($arr5);

        return $fingerprint;
    }
}
```

2. Kode Program C_aplikasi.php

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

/**
 *
 */
class C_aplikasi extends CI_controller
{

    function __construct()
    {
        parent::__construct();
        require_once __DIR__ . '/sastrawi-
master/vendor/autoload.php';
        $this->load->model('m_bidangstudi');
        $this->load->model('m_jurnal');
        $this->load->model('m_log');
    }

    function upload_jurnal()
    {
        $penulis    = $this->input->post('penulis');
        $judul      = $this->input->post('judul');
        $tahun      = $this->input->post('tahun');
        $studi      = $this->m_bidangstudi-
>get_bidangstudi_by_id($this->input->post('studi'))-
>row('kategori');
        $penerbit   = $this->input->post('penerbit');

        if($this->input->post('volume') != null){
            $volume    = "Vol. " . $this->input-
>post('volume') . ".";
        }
        else{
            $volume = "";
        }
    }
}
```

```
}

$nama          = "";

$expl = explode(";", $penulis);

if(count($expl) == "2"){
    $nama .= $expl[0];
}
else if(count($expl) == "3"){
    $nama .= $expl[1].", & ".$expl[0];
}
else if(count($expl) == "4"){
    $nama .= $expl[2].", ".$expl[1].", & ".$expl[0];
}
else{
    $nama .= $expl[count($expl)-2].", et al.";
}

$bibliografi = $nama." (".$tahun."). ".$judul.".
".$studi.". ".$volume;

//upload jurnal
$config['upload_path'] = 'uploads/';
$config['allowed_types'] = 'doc|docx|txt';
$this->load->library('upload', $config);
if ( ! $this->upload->do_upload('jurnal')){
    //$error = $this->upload->display_errors();

    $data['error'] = "File yang anda upload
tidak diperbolehkan (hanya .txt, .doc, .docx)...";
    $data['check_aktif'] = "active";
    $data['level'] = $this->session-
>userdata('level');
    $data['username'] = $this->session-
>userdata('username');
    $data['nama_lengkap'] = $this->session-
>userdata('nama_lengkap');
    $data['pict'] = $this->session-
>userdata('pict');
    $data['bidangstudi'] = $this->m_bidangstudi-
>get_bidangstudi();

    $this->load->view('elements/user_header',
$data);
```

```
        $this->load->view('pages/user/v_cekplagiat',
$data);
    }
    else{
        $ext = $this->upload->data('file_ext');
        $path = $this->upload->data('full_path');

        //data header
        $data['check_aktif'] = "active";
        $data['level'] = $this->session-
>userdata('level');
        $data['username'] = $this->session-
>userdata('username');
        $data['nama_lengkap'] = $this->session-
>userdata('nama_lengkap');
        $data['pict'] = $this->session-
>userdata('pict');
        //data header

        $data['penulis_utama'] = $expl[count($expl)-2];
        $data['judul'] = $judul;
        $data['bidang_studi'] = $studi;
        $data['biblio'] = $bibliografi;
        $data['file_name'] = $this->upload-
>data('file_name');
        $data['isi_file'] = $this-
>get_content($ext, $path);

        $this->load->view('elements/user_header',
$data);
        $this->load->view('pages/user/v_previewupload',
$data);
    }
    //end of upload
}

function proses_jurnal()
{
    $basis = 3;
    $gram = 7;
    $window = 6;

    $penulis_utama = $this->input->post('penulis');
    $judul = $this->input->post('judul');
    $bidang_studi = $this->input-
```



```
>post('bidang_studi');
    $bibliografi      = $this->input->post('bibliografi');
    $nama_file        = $this->input->post('nama_file');
    $isi_jurnal       = $this->input->post('isi_jurnal');
    $hasil_stem       = $this->stemmer($isi_jurnal);

    $hasil_filter     = $this-
>filtering($hasil_stem);
    $parsing_gram     = $this-
>parsing_gram($hasil_filter, $gram);
    $convert_hash     = $this-
>convert_hash($parsing_gram, $gram, $basis);
    $parsing_window   = $this-
>parsing_window($convert_hash, $window);
    $fingerprint      = $this-
>fingerprint($parsing_window);

    //data header
    $data['check_aktif'] = "active";
    $data['level']      = $this->session-
>userdata('level');
    $data['username']   = $this->session-
>userdata('username');
    $data['nama_lengkap'] = $this->session-
>userdata('nama_lengkap');
    $data['pict']       = $this->session-
>userdata('pict');
    //data header

    $data['penulis_utama'] = $penulis_utama;
    $data['judul']         = $judul;
    $data['bidang_studi']  = $bidang_studi;
    $data['biblio']        = $bibliografi;
    $data['file_name']     = $nama_file;
    $data['isi_file']      = $hasil_stem;
    $data['fingerprint']   = $fingerprint;
    $data['gram']          = $gram;
    $data['window']        = $window;
    $data['hasil_filter']  = $hasil_filter;
    $data['parsing_gram']  = $parsing_gram;
    $data['convert_hash']  = $convert_hash;
    $data['parsing_window'] = $parsing_window;

    $this->load->view('elements/user_header', $data);
    $this->load->view('pages/user/v_proseshitung', $data);
```

```
}

function pilih_jurnal()
{
    $penulis_utama = $this->input->post('penulis');
    $judul          = $this->input->post('judul');
    $bidang_studi   = $this->input->post('bidang_studi');
    $bibliografi    = $this->input->post('bibliografi');
    $nama_file      = $this->input->post('nama_file');
    $hasil_stem     = $this->input->post('isi_jurnal');
    $fingerprint    = $this->input->post('fingerprint_jurnal');

    $data['penulis_utama'] = $penulis_utama;
    $data['judul']        = $judul;
    $data['bidang_studi'] = $bidang_studi;
    $data['biblio']       = $bibliografi;
    $data['file_name']    = $nama_file;
    $data['isi_file']     = $hasil_stem;
    $data['fingerprint'] = $fingerprint;

    $data['bidangstudi'] = $this->m_bidangstudi->get_bidangstudi();

    //data header
    $data['check_aktif'] = "active";
    $data['level']       = $this->session->userdata('level');
    $data['username']    = $this->session->userdata('username');
    $data['nama_lengkap'] = $this->session->userdata('nama_lengkap');
    $data['pict']        = $this->session->userdata('pict');

    //data header
    $this->load->view('elements/user_header', $data);
    $this->load->view('pages/user/v_pilihjurnal', $data);
}

function upload_referensi()
{
    $basis = 3;
    $gram  = 7;
}
```

```
$window = 6;

$penulis    = $this->input->post('penulis');
$jjudul     = $this->input->post('judul');
$tahun      = $this->input->post('tahun');
$studi      = $this->m_bidangstudi-
>get_bidangstudi_by_id($this->input->post('studi'))-
>row('kategori');
$penerbit   = $this->input->post('penerbit');

if($this->input->post('volume') != null){
    $volume  = "Vol. ".$this->input-
>post('volume').".";
}
else{
    $volume = "";
}

$nama       = "";

$expl = explode(";", $penulis);

if(count($expl) == "2"){
    $nama .= $expl[0];
}
else if(count($expl) == "3"){
    $nama .= $expl[1].", & ".$expl[0];
}
else if(count($expl) == "4"){
    $nama .= $expl[2].", ".$expl[1].", & ".$expl[0];
}
else{
    $nama .= $expl[count($expl)-2].", et al.";
}

$bibliografi = $nama." (".$tahun."). ".$judul.".
".$studi.". ".$volume;

//upload jurnal
$config['upload_path'] = 'uploads/';
$config['allowed_types'] = 'doc|docx|txt';
$this->load->library('upload', $config);
if ( ! $this->upload->do_upload('jurnal')){
echo $this->upload->display_errors();
}
}
```

```
else{
    $ext = $this->upload->data('file_ext');
    $path = $this->upload->data('full_path');

    $penulis_utama = $expl[count($expl)-2];
    $file_name = $this->upload-
>data('file_name');
    $isi_file = $this->get_content($ext,
$path);
}
//end of upload

$hasil_stem = $this->stemmer($isi_file);

$hasil_filter = $this-
>filtering($hasil_stem);
$parsing_gram = $this-
>parsing_gram($hasil_filter, $gram);
$convert_hash = $this-
>convert_hash($parsing_gram, $gram, $basis);
$parsing_window = $this-
>parsing_window($convert_hash, $window);
$fingerprint = $this-
>fingerprint($parsing_window);
$hasil_fingerprint = "";

foreach($fingerprint as $x => $x_value) {
    $hasil_fingerprint .= "[" . $x_value . " ] ";
}

$path_file = "uploads/". $file_name;
$status_jurnal = 2;
$uploader = $this->session-
>userdata('username');

$data = array(
    'id_jurnal' => '',
    'penulis_utama' => $penulis_utama,
    'judul' => $judul,
    'bidang_studi' => $this->input->post('studi'),
    'bibliografi' => $bibliografi,
    'hasil_stem' => $hasil_stem,
    'fingerprint' => $hasil_fingerprint." ",
    'nama_file' => $file_name,
    'path_file' => $path_file,
```

```
        'status_jurnal' => $status_jurnal,
        'uploader' => $uploader
    );

    $this->m_jurnal->insert_jurnal($data);

    $aktifitas = "6";
    $this->m_log->insert_log($uploader, $aktifitas);

    $id_max = $this->m_jurnal->get_id_max()-
>row('id_max');
    $bibli = $this->m_jurnal->get_jurnal_by_id($id_max)-
>row('bibliografi');

    $result['id'] = $id_max;
    $result['bibliografi'] = $bibli;

    header('Content-Type: application/x-json; charset=utf-
8');
    echo(json_encode($result));
}

function proses_hitung()
{
    $penulis_utama = $this->input->post('penulis');
    $judul = $this->input->post('judul');
    $bidang_studi = $this->input-
>post('bidang_studi');
    $bibliografi = $this->input->post('bibliografi');
    $nama_file = $this->input->post('nama_file');
    $hasil_stem = $this->input->post('isi_jurnal');
    $fingerprint = $this->input-
>post('fingerprint_jurnal');

    $result = array();
    $result_array = array();
    $arr_fingerprint = explode(" ", $fingerprint);
    array_pop($arr_fingerprint);

    if($this->input->post('file_asli') < 1){
        if($this->input->post('jurnal_referensi') < 1){
            echo "Tidak Ada File Yang Akan Di
Bandingkan";
        }
    }
    else{
```

```

        foreach ($this->input-
>post('jurnal_referensi') as $id_referensi){
            $query = $this->m_jurnal-
>get_jurnal_by_id($id_referensi);

            $arr_fingerprint_jurnal = explode("
", $query->row('fingerprint'));
            array_pop($arr_fingerprint_jurnal);

            $intersect1 =
array_intersect($arr_fingerprint_jurnal, $arr_fingerprint);
            $intersect2 =
array_intersect($arr_fingerprint, $arr_fingerprint_jurnal);
            $dif1 =
array_diff($arr_fingerprint_jurnal, $arr_fingerprint);
            $dif2 =
array_diff($arr_fingerprint, $arr_fingerprint_jurnal);
            $sama =
count(array_intersect($arr_fingerprint_jurnal, $arr_fingerprint));
            $jumlah = (count($dif1)-1
+ count($dif2) + $sama;
            $hasil = ($sama /
$jumlah) * 100;

            $result['id_jurnal'] = $query-
>row('id_jurnal');
            $result['bibliografi'] = $query-
>row('bibliografi');
            $result['fingerprint'] = $query-
>row('fingerprint');

            $result['sama'] = $sama;
            $result['dif1'] = count($dif1)-1;
            $result['dif2'] = count($dif2);
            $result['hasil'] = round($hasil, 2);

            array_push($result_array, $result);
        }

        $data['result'] = $result_array;

        //data header
        $data['check_aktif'] = "active";
        $data['level'] = $this-
>session->userdata('level');
        $data['username'] = $this-

```

```

>session->userdata('username');
                $data['nama_lengkap'] = $this->session-
>userdata('nama_lengkap');
                $data['pict']          = $this-
>session->userdata('pict');
                //data header

                $data['penulis_utama'] = $penulis_utama;
                $data['judul']         = $judul;
                $data['bidang_studi']  = $bidang_studi;
                $data['biblio']        = $bibliografi;
                $data['file_name']     =

$nama_file;
                $data['isi_file']     =

$hasil_stem;
                $data['fingerprint']  = $fingerprint;

                $this->load->view('elements/user_header',
$data);
                $this->load-
>view('pages/user/v_hasilcheck', $data);
            }
        }
        else{
            foreach ($this->input->post('file_asli') as
$id_jurnal) {
                $query = $this->m_jurnal-
>get_jurnal_by_id($id_jurnal);

                $arr_fingerprint_jurnal = explode(" ",
$query->row('fingerprint'));
                array_pop($arr_fingerprint_jurnal);

                $intersect1 =
array_intersect($arr_fingerprint_jurnal, $arr_fingerprint);
                $intersect2 =
array_intersect($arr_fingerprint, $arr_fingerprint_jurnal);
                $dif1 =
array_diff($arr_fingerprint_jurnal, $arr_fingerprint);
                $dif2 = array_diff($arr_fingerprint,
$arr_fingerprint_jurnal);
                $sama =
count(array_intersect($arr_fingerprint_jurnal, $arr_fingerprint));
                $jumlah = (count($dif1)-1) +
count($dif2) + $sama;

```

```

100;
    $hasil = ($sama / $jumlah) *
>row('id_jurnal');
    $result['id_jurnal'] = $query-
>row('bibliografi');
    $result['bibliografi'] = $query-
>row('fingerprint');
    $result['fingerprint'] = $query-
    $result['hasil'] = round($hasil, 2);
    array_push($result_array, $result);
}

    if($this->input->post('jurnal_referensi') > 0){
        foreach ($this->input-
>post('jurnal_referensi') as $id_referensi){
            $query = $this->m_jurnal-
>get_jurnal_by_id($id_referensi);

            $arr_fingerprint_jurnal = explode("
", $query->row('fingerprint'));
            array_pop($arr_fingerprint_jurnal);

            $intersect1 =
array_intersect($arr_fingerprint_jurnal, $arr_fingerprint);
            $intersect2 =
array_intersect($arr_fingerprint, $arr_fingerprint_jurnal);
            $dif1 =
array_diff($arr_fingerprint_jurnal, $arr_fingerprint);
            $dif2 =
array_diff($arr_fingerprint, $arr_fingerprint_jurnal);
            $sama =
count(array_intersect($arr_fingerprint_jurnal, $arr_fingerprint));
            $jumlah = (count($dif1)-1)
+ count($dif2) + $sama;
            $hasil = ($sama /
$jumlah) * 100;

            $result['id_jurnal'] = $query-
>row('id_jurnal');
            $result['bibliografi'] = $query-
>row('bibliografi');
            $result['fingerprint'] = $query-
>row('fingerprint');

```



```
        $result['hasil'] = round($hasil, 2);

        array_push($result_array, $result);
    }
}

$data['result'] = $result_array;

//data header
$data['check_aktif'] = "active";
$data['level'] = $this->session->userdata('level');
$data['username'] = $this->session->userdata('username');
$data['nama_lengkap'] = $this->session->userdata('nama_lengkap');
$data['pict'] = $this->session->userdata('pict');

//data header
$data['penulis_utama'] = $penulis_utama;
$data['judul'] = $judul;
$data['bidang_studi'] = $bidang_studi;
$data['biblio'] = $bibliografi;
$data['file_name'] = $nama_file;
$data['isi_file'] = $hasil_stem;
$data['fingerprint'] = $fingerprint;

$this->load->view('elements/user_header',
$data);
$this->load->view('pages/user/v_hasilcheck',
$data);
}

}

function get_list_file($id_bidang_studi)
{
    $query = $this->m_jurnal->get_approve_jurnal_by_bidangstudi($id_bidang_studi);
    $listfile = array();

    if($query->num_rows() != 0){
```

```
        foreach ($query->result() as $file) {
            $listfile[$file->id_jurnal] = $file->bibliografi;
        }
    } else {
        $listfile['0'] = "Tidak Ada File Jurnal";
    }

    header('Content-Type: application/x-json; charset=utf-
8');
    echo(json_encode($listfile));
}

function get_detail_jurnal($id_jurnal)
{
    $query = $this->m_jurnal-
>get_jurnal_by_id($id_jurnal);
    $result = array();

    foreach ($query->result_array() as $item) {
        $result['bibliografi'] = $item['bibliografi'];
        $result['fingerprint'] = $item['fingerprint'];
        $path_file = $item['path_file'];
    }

    $exp = explode(".", $path_file);
    $ext = ".".end($exp);

    $result['isi_file'] = $this->get_content($ext,
$path_file);

    header('Content-Type: application/x-json; charset=utf-
8');
    echo(json_encode($result));
}

function cancel_action($nama_file)
{
    $path = "uploads/". $nama_file;

    unlink($path);
}

function get_content($ext, $file_path)
{
    $isi_file = '';
```

```
if($ext == ".txt"){
    $isi_file = file_get_contents($file_path);
}
else if($ext == ".docx"){
    $striped_content = '';
    $content = '';

    if(!$file_path || !file_exists($file_path)) return
false;

    $zip = zip_open($file_path);

    if (!$zip || is_numeric($zip)) return false;

    while ($zip_entry = zip_read($zip)) {
        if (zip_entry_open($zip, $zip_entry) == FALSE)
continue;

        if (zip_entry_name($zip_entry) !=
"word/document.xml") continue;

        $content .= zip_entry_read($zip_entry,
zip_entry_filesize($zip_entry));

        zip_entry_close($zip_entry);
    } // end while

    zip_close($zip);

    //echo $content;
    //echo "<hr>";
    //file_put_contents('1.xml', $content);

    $content = str_replace('</w:r></w:p></w:tc><w:tc>', "
", $content);
    $content = str_replace('</w:r></w:p>', "\r\n",
$content);
    $striped_content = strip_tags($content);

    $isi_file = $striped_content;
}
else{
    if(file_exists($file_path)) {
        if(($fh = fopen($file_path, 'r')) != false) {
```

```

        $headers = fread($fh, 0xA00);
        $n1 = ( ord($headers[0x21C]) - 1 );// 1 =
(ord(n)*1) ; Document has from 0 to 255 characters
        $n2 = ( ( ord($headers[0x21D]) - 8 ) * 256
);// 1 = ((ord(n)-8)*256) ; Document has from 256 to 63743
characters
        $n3 = ( ( ord($headers[0x21E]) * 256 ) * 256
);// 1 = ((ord(n)*256)*256) ; Document has from 63744 to 16775423
characters
        $n4 = ( ( ( ord($headers[0x21F]) * 256 ) * 256
) * 256 );// 1 = (((ord(n)*256)*256)*256) ; Document has from
16775424 to 4294965504 characters
        $textLength = ($n1 + $n2 + $n3 + $n4);// Total
length of text in the document
        $extracted_plaintext = fread($fh,
$textLength);
        $extracted_plaintext =
mb_convert_encoding($extracted_plaintext,'UTF-8');
        // if you want to see your paragraphs in a
new line, do this
        // return nl2br($extracted_plaintext);
        $isi_file = $extracted_plaintext;
    } else {
        $isi_file = 'Cannot Load File Content';
    }
}
else {
    $isi_file = 'File Not Exist';
}
}

    return $isi_file;
}

function stemmer($isi_jurnal)
{
    // create stemmer
    // cukup dijalankan sekali saja, biasanya didaftarkan
di service container
    $stemmerFactory = new
\Sastrawi\Stemmer\StemmerFactory();
    $stemmer = $stemmerFactory->createStemmer();

    // stem
    $hasil_stem = $stemmer->stem($isi_jurnal);

```

```
        return $hasil_stem;
    }

    function filtering($string)
    {
        $filter          = preg_replace("/[^a-zA-Z]/", "",
$string);
        $filter_final    = strtolower($filter);

        return $filter_final;
    }

    function parsing_gram($filter_final, $gram)
    {
        $arr1 = str_split($filter_final);

        for($i=0;$i<count($arr1); $i++){
            array_slice(array_slice($arr1, $i), 0, $gram);

            if(count(array_slice(array_slice($arr1, $i), 0,
$gram)) == $gram){
                $arr2[$i] = array_slice(array_slice($arr1,
$i), 0, $gram);
            }
        }

        return $arr2;
    }

    function convert_hash($parsing_gram, $gram, $basis)
    {
        for($x=0;$x<count($parsing_gram); $x++){
            $sum = 0;
            for($y=0;$y<$gram; $y++){
                $pangkat = $gram-($y+1);
                $sum_temp =
ord($parsing_gram[$x][$y])*(pow($basis, $pangkat));
                $sum = $sum+$sum_temp;
            }
            $arr3[]=$sum;
        }

        return $arr3;
    }
}
```

```
function parsing_window($convert_hash, $window)
{
    for($i=0;$i<count($convert_hash); $i++){
        array_slice(array_slice($convert_hash, $i), 0,
$window);

        if(count(array_slice(array_slice($convert_hash,
$i), 0, $window)) == $window){
            $arr4[$i] =
array_slice(array_slice($convert_hash, $i), 0, $window);
        }
    }

    return $arr4;
}

function fingerprint($parsing_window)
{
    $t=0;
    while($t<count($parsing_window)){
        $arr5[] = min($parsing_window[$t]);
        $t++;
    }

    $fingerprint = array_unique($arr5);

    return $fingerprint;
}
}
```

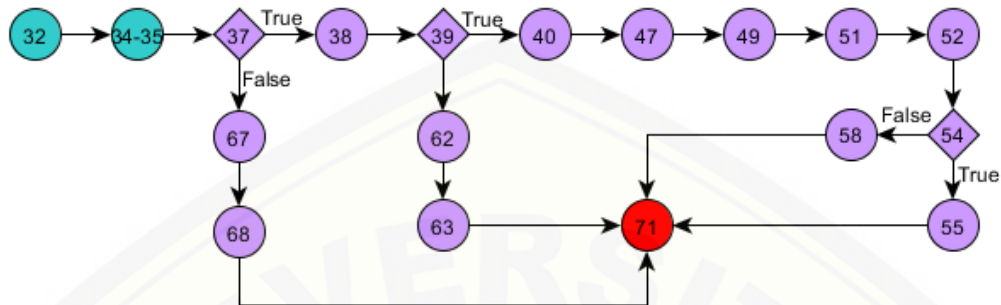
LAMPIRAN C. TESTING**C1. White Box Testing**

1. Login ke Sistem

a) Listing Program

```
32 function login()
33 {
34     $username = $this->input->post('username');
35     $password = $this->input->post('password');
36
37     if($this->m_user->get_user($username, $password)->num_rows() > 0){
38         $user = $this->m_user->get_user($username, $password)->row();
39         if($user->id_status == '1'){
40             $array_items = array(
41                 'level' => $user->level,
42                 'username' => $user->username,
43                 'nama_lengkap' => $user->nama_lengkap,
44                 'pict' => $user->pict
45             );
46
47             $this->session->set_userdata($array_items);
48
49             $this->m_user->update_lastlog($username);
50
51             $aktifitas = "2";
52             $this->m_log->insert_log($username, $aktifitas);
53
54             if($this->session->userdata('level') == 'admin'){
55                 redirect(site_url('c_adminpage'));
56             }
57             else{
58                 redirect(site_url('c_userpage'));
59             }
60         }
61         else{
62             $data['error'] = "User Sudah di Nonaktifkan oleh Admin.";
63             $this->load->view('v_login', $data);
64         }
65     }
66     else{
67         $data['error'] = "Username dan Password Anda Salah.";
68         $this->load->view('v_login', $data);
69     }
70 }
71 }
```

b) Grafik Alir



c) Perhitungan *Cyclomatic Complexity*

Perhitungan *cyclomatic complexity* $V(G)$ untuk grafik alir fitur *login* adalah $V(G) = \text{Edge} - \text{Node} + 2 = 20 - 18 + 2 = 4$

d) Basis Set

Basis set menghasilkan 4 jalur independen

Path 1 : 32-34-35-37-38-39-40-41-42-43-44-45-47-49-51-52-54-55-71

Path 2 : 32-34-35-37-38-39-40-41-42-43-44-45-47-49-51-52-54-58-71

Path 3 : 32-34-35-37-38-39-62-63-71

Path 4 : 32-34-35-37-67-68-71

e) *Test Case* Fitur *Login*

<i>Test Case</i> Fitur <i>Login</i>	
<i>Test case</i>	Jika <i>login</i> dengan level <i>user</i> admin
Target yang diharapkan	<i>Login</i> berhasil dan diarahkan ke halaman admin
Hasil Pengujian	Benar
Path	32-34-35-37-38-39-40-41-42-43-44-45-47-49-51-52-54-55-71 (Path 1)

<i>Test Case</i> Fitur <i>Login</i>	
<i>Test case</i>	Jika <i>login</i> dengan level <i>user</i> mahasiswa
Target yang diharapkan	<i>Login</i> berhasil dan diarahkan ke halaman mahasiswa

Hasil Pengujian	Benar
Path	32-34-35-37-38-39-40-41-42-43-44-45-47-49-51-52-54-55-71 (Path 2)

Test Case Fitur Login

<i>Test case</i>	Jika <i>login</i> dengan status <i>user nonaktif</i>
Target yang diharapkan	<i>Login gagal</i> dan menampilkan pesan <i>error</i>
Hasil Pengujian	Benar
Path	32-34-35-37-38-39-62-63-71 (Path 3)

Test Case Fitur Login

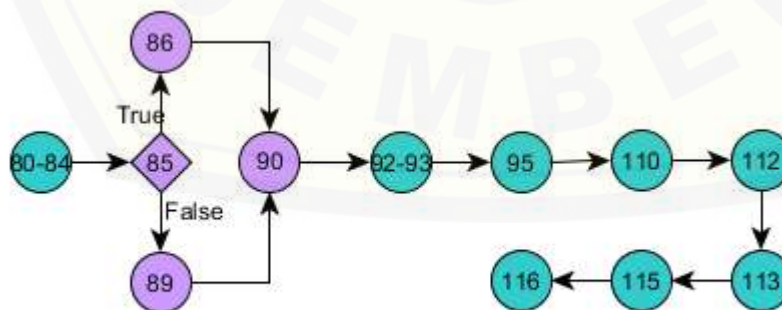
<i>Test case</i>	Jika <i>login gagal</i>
Target yang diharapkan	Menampilkan pesan <i>error</i>
Hasil Pengujian	Benar
Path	32-34-35-37-67-68-71 (Path 4)

2. Register User Baru

a) Listing Program

```
78 function registrasi()  
79 {  
80     $nama_lengkap = $this->input->post('nama_lengkap');  
81     $email         = $this->input->post('email');  
82     $alamat       = $this->input->post('alamat');  
83     $kota         = $this->input->post('kota');  
84     $gender       = $this->input->post('jenis_kelamin');  
85     if($gender == "m"){  
86         $pict = "male.png";  
87     }  
88     else{  
89         $pict = "female.png";  
90     }  
91  
92     $username     = $this->input->post('username');  
93     $password     = $this->input->post('password');  
94  
95     $data_user = array(  
96         'username' => $username,  
97         'password' => $password,  
98         'id_level' => 'usr',  
99         'created'  => $this->m_user->get_datetime(),  
100        'last_login' => '',  
101        'id_status' => '1',  
102        'nama_lengkap' => $nama_lengkap,  
103        'email'       => $email,  
104        'alamat'     => $alamat,  
105        'kota'       => $kota,  
106        'gender'     => $gender,  
107        'pict'       => $pict  
108    );  
109  
110    $this->m_user->insert_user($data_user);  
111  
112    $aktifitas = "1";  
113    $this->m_log->insert_log($username, $aktifitas);  
114  
115    $data['sukses'] = "Akun Berhasil dibuat";  
116    $this->load->view('v_login', $data);  
117 }
```

b) Grafik Alir



c) Perhitungan *Cyclomatic Complexity*

Perhitungan *cyclomatic complexity* $V(G)$ untuk grafik alir fitur *register user* baru adalah $V(G) = \text{Edge} - \text{Node} + 2 = 12 - 12 + 2 = 2$

d) Basis Set

Basis set menghasilkan 2 jalur independen

Path 1 : 80-81-82-83-84-85-86-90-92-93-95-110-112-113-115-116

Path 2 : 80-81-82-83-84-85-89-90-92-93-95-110-112-113-115-116

e) *Test Case* Fitur *Register User* Baru

<i>Test Case</i> Fitur <i>Register User</i> Baru	
<i>Test case</i>	Menambahkan data <i>user</i> baru ke <i>database</i>
Target yang diharapkan	Data <i>user</i> baru berhasil disimpan ke <i>database</i>
Hasil Pengujian	Benar
Path	80-81-82-83-84-85-86-90-92-93-95-110-112-113-115-116 (Path 1)

3. Hapus Data Jurnal

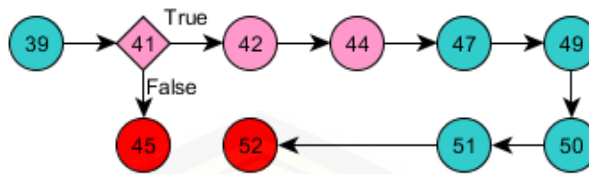
a) *Listing* Program

```

37     function delete_jurnal($id_jurnal)
38     {
39         $jurnal = $this->m_jurnal->get_jurnal_by_id($id_jurnal);
40
41         foreach($jurnal->result_array() as $row){
42             $path = $row['path_file'];
43
44             unlink($path);
45         }
46
47         $this->m_jurnal->delete_jurnal($id_jurnal);
48
49         $username = $this->session->userdata('username');
50         $aktifitas = "7";
51         $this->m_log->insert_log($username, $aktifitas);
52     }

```

b) Grafik Alir



c) Perhitungan *Cyclomatic Complexity*

Perhitungan *cyclomatic complexity* $V(G)$ untuk grafik alir fitur hapus data jurnal adalah $V(G) = \text{Edge} - \text{Node} + 2 = 9 - 10 + 2 = 1$

d) Basis Set

Basis set menghasilkan jalur 1 independen

Path 1 : 39-41-42-44-47-49-50-51-52

e) *Test Case*

<i>Test Case</i> Fitur Hapus Data Jurnal	
<i>Test case</i>	Menghapus data jurnal yang dipilih
Target yang diharapkan	Data jurnal yang dipilih berhasil dihapus dari <i>database</i>
Hasil Pengujian	Benar
Path	39-41-42-44-47-49-50-51-52

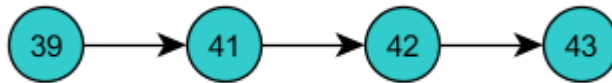
4. Approve Data Jurnal

a) *Listing Program*

```

37 function approve_jurnal($id_jurnal)
38 {
39     $this->m_jurnal->update_jurnal_status($id_jurnal);
40
41     $username = $this->session->userdata('username');
42     $aktifitas = "8";
43     $this->m_log->insert_log($username, $aktifitas);
44 }
  
```

b) Grafik Alir



c) Perhitungan *CyclomaticComplexity*

Perhitungan *cyclomatic complexity* $V(G)$ untuk grafik alir fitur *approve* data jurnal adalah $V(G) = \text{Edge} - \text{Node} + 2 = 3 - 4 + 2 = 1$

d) Basis Set

Basis set menghasilkan jalur 1 independen

Path 1 : 39-41-42-43

e) *Test Case*

<i>Test Case</i> Fitur <i>Approve</i> Data Jurnal	
<i>Test case</i>	Menyetujui data jurnal yang dipilih
Target yang diharapkan	Data jurnal yang dipilih berhasil disetujui dan disimpan ke <i>database</i>
Hasil Pengujian	Benar
Path	39-41-42-43

6. Nonaktifkan *User*

a) *Listing* Program

```

23     function nonaktifkan($username)
24     {
25         $status = 0;
26         $this->m_user->update_status($username, $status);
27
28         $user = $this->session->userdata('username');
29         $aktifitas = "8";
30         $this->m_log->insert_log($user, $aktifitas);
31     }

```

b) Grafik Alir



c) Perhitungan *Cyclomatic Complexity*

Perhitungan *cyclomatic complexity* $V(G)$ untuk grafik alir fitur nonaktifkan *user* adalah $V(G) = \text{Edge} - \text{Node} + 2 = 4 - 5 + 2 = 1$

d) Basis Set

Basis set menghasilkan 1 jalur independen

Path 1 : 25-26-27-28-29-30

e) *Test Case*

<i>Test Case</i> Fitur Nonaktifkan <i>User</i>	
<i>Test case</i>	Menonaktifkan <i>user</i> yang dipilih
Target yang diharapkan	<i>User</i> berhasil dinonaktifkan dan disimpan ke <i>database</i>
Hasil Pengujian	Benar
Path	25-26-27-28-29-30

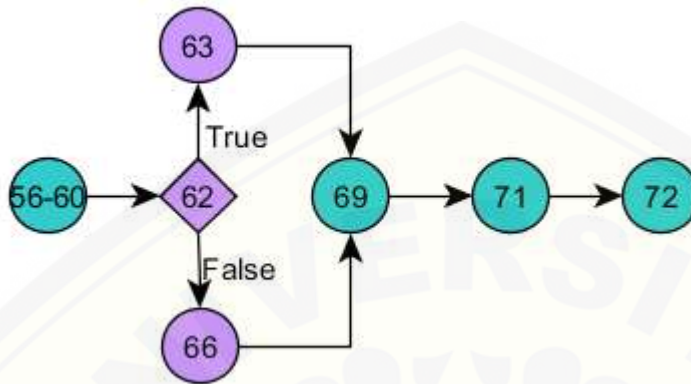
7. Ubah Profil *User*

a) *Listing* Program

```

54  function update_profil()
55  {
56      $username = $_POST['username'];
57      $nama_baru = $_POST['nama_baru'];
58      $alamat_baru = $_POST['alamat_baru'];
59      $kota_baru = $_POST['kota_baru'];
60      $gender_baru = $_POST['gender_baru'];
61
62      if($gender_baru == "m"){
63          $pict = "male.png";
64      }
65      else{
66          $pict = "female.png";
67      }
68
69      $this->m_user->update_detail($username, $nama_baru, $alamat_baru, $kota_baru, $gender_baru, $pict);
70
71      $aktifitas = "4";
72      $this->m_log->insert_log($username, $aktifitas);
73  }
  
```

b) Grafik Alir

c) Perhitungan *Cyclomatic Complexity*

Perhitungan *cyclomatic complexity* $V(G)$ untuk grafik alir fitur ubah profil *user* adalah $V(G) = \text{Edge} - \text{Node} + 2 = 7 - 7 + 2 = 2$

d) Basis Set

Basis set menghasilkan 2 jalur independen

Path 1 : 56-57-58-59-60-62-63-69-71-72

Path 2 : 56-57-58-59-60-62-66-69-71-72

e) *Test Case*

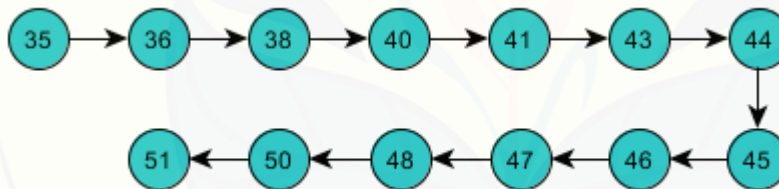
<i>Test Case</i> Fitur Ubah Profil <i>User</i>	
<i>Test case</i>	Mengubah profil <i>user</i>
Target yang diharapkan	<i>Update</i> profil <i>user</i> berhasil disimpan ke <i>database</i>
Hasil Pengujian	Benar
Path	56-57-58-59-60-62-63-69-71-72 (Path 1)

8. Ubah *Password Login User*a) *Listing Program*

```

33  function update_password()
34  {
35      $username = $this->session->userdata('username');
36      $password = $this->input->post('password');
37
38      $this->m_user->update_password($password, $username);
39
40      $aktifitas = "5";
41      $this->m_log->insert_log($username, $aktifitas);
42
43      $data['sukses'] = "Password Anda Berhasil Diperbarui";
44      $data['pass_aktif'] = "active";
45      $data['level'] = $this->session->userdata('level');
46      $data['username'] = $this->session->userdata('username');
47      $data['nama_lengkap'] = $this->session->userdata('nama_lengkap');
48      $data['pict'] = $this->session->userdata('pict');
49
50      $this->load->view('elements/admin_header', $data);
51      $this->load->view('pages/admin/v_updatepass', $data);
52  }

```

b) *Grafik Alir*c) *Perhitungan Cyclomatic Complexity*

Perhitungan *cyclomatic complexity* $V(G)$ untuk grafik alir fitur *ubah password login user* adalah $V(G) = \text{Edge} - \text{Node} + 2 = 12 - 13 + 2 = 1$

d) *Basis Set*

Basis set menghasilkan 1 jalur independen

Path 1 : 35-36-38-40-41-43-44-45-46-47-48-50-51

e) *Test Case*

Test Case Fitur Ubah Password Login User	
Test case	Mengubah password login user
Target yang diharapkan	Update password login user berhasil disimpan ke database
Hasil Pengujian	Benar
Path	35-36-38-40-41-43-44-45-46-47-48-50-51

9. View Log Aktifitas User

a) Listing Program

```

124 function log()
125 {
126     $data['log_aktif'] = "active";
127     $data['level']     = $this->session->userdata('level');
128     $data['username']  = $this->session->userdata('username');
129     $data['nama_lengkap'] = $this->session->userdata('nama_lengkap');
130     $data['pict']      = $this->session->userdata('pict');
131
132     $data['log']       = $this->m_log->get_all_log()->result_array();
133
134     $this->load->view('elements/admin_header', $data);
135     $this->load->view('pages/admin/v_log', $data);
136 }

```

b) Grafik Alir



c) Perhitungan Cyclomatic Complexity

Perhitungan *cyclomatic complexity* $V(G)$ untuk grafik alir fitur *view log* aktifitas *user* adalah $V(G) = \text{Edge} - \text{Node} + 2 = 7 - 8 + 2 = 1$

d) Basis Set

Basis set menghasilkan 1 jalur independen

Path 1 : 126-127-128-129-130-132-134-135

e) Test Case

Test Case Fitur View Log Aktifitas User	
Test case	View log aktifitas user
Target yang diharapkan	Sistem menampilkan log aktifitas user yang ada pada database
Hasil Pengujian	Benar
Path	126-127-128-129-130-132-134-135

10. View Data Jurnal

a) Listing Program

```

44     function view_jurnal()
45     {
46         $data['view_aktif']     = "active";
47         $data['level']         = $this->session->userdata('level');
48         $data['username']      = $this->session->userdata('username');
49         $data['nama_lengkap']  = $this->session->userdata('nama_lengkap');
50         $data['pict']          = $this->session->userdata('pict');
51
52         $data['jurnal'] = $this->m_jurnal->get_all_jurnal()->result_array();
53
54         $this->load->view('elements/admin_header', $data);
55         $this->load->view('pages/admin/v_listjurnal');
56     }

```

b) Grafik Alir



c) Perhitungan Cyclomatic Complexity

Perhitungan *cyclomatic complexity* V (G) untuk grafik alir fitur view data jurnal adalah $V(G) = \text{Edge} - \text{Node} + 2 = 7 - 8 + 2 = 1$

d) Basis Set

Basis set menghasilkan 1 jalur independen

Path 1 : 46-47-48-49-50-52-54-55

e) Test Case

Test Case Fitur View Data Jurnal	
Test case	View data jurnal
Target yang diharapkan	Sistem menampilkan data jurnal yang ada pada <i>database</i>
Hasil Pengujian	Benar
Path	46-47-48-49-50-52-54-55

11. View Data User

a) Listing Program

```

72 function list_user()
73 {
74     $data['users_aktif'] = "active";
75     $data['level']      = $this->session->userdata('level');
76     $data['username']   = $this->session->userdata('username');
77     $data['nama_lengkap'] = $this->session->userdata('nama_lengkap');
78     $data['pict']       = $this->session->userdata('pict');
79
80     $data['users'] = $this->m_user->get_all_users()->result_array();
81
82     $this->load->view('elements/admin_header', $data);
83     $this->load->view('pages/admin/v_listuser', $data);
84 }

```

b) Grafik Alir



c) Perhitungan *Cyclomatic Complexity*

Perhitungan cyclomatic complexity $V(G)$ untuk grafik alir fitur view data *user* adalah $V(G) = \text{Edge} - \text{Node} + 2 = 7 - 8 + 2 = 1$

d) Basis Set

Basis set menghasilkan 1 jalur independen

Path 1 : 74-75-76-77-78-80-82-83

e) *Test Case*

Test Case Fitur View Data User	
Test case	View data user
Target yang diharapkan	Sistem menampilkan data <i>user</i> yang ada pada <i>database</i>
Hasil Pengujian	Benar
Path	74-75-76-77-78-80-82-83

12. Mengecek Plagiat Jurnal

a) Listing Program

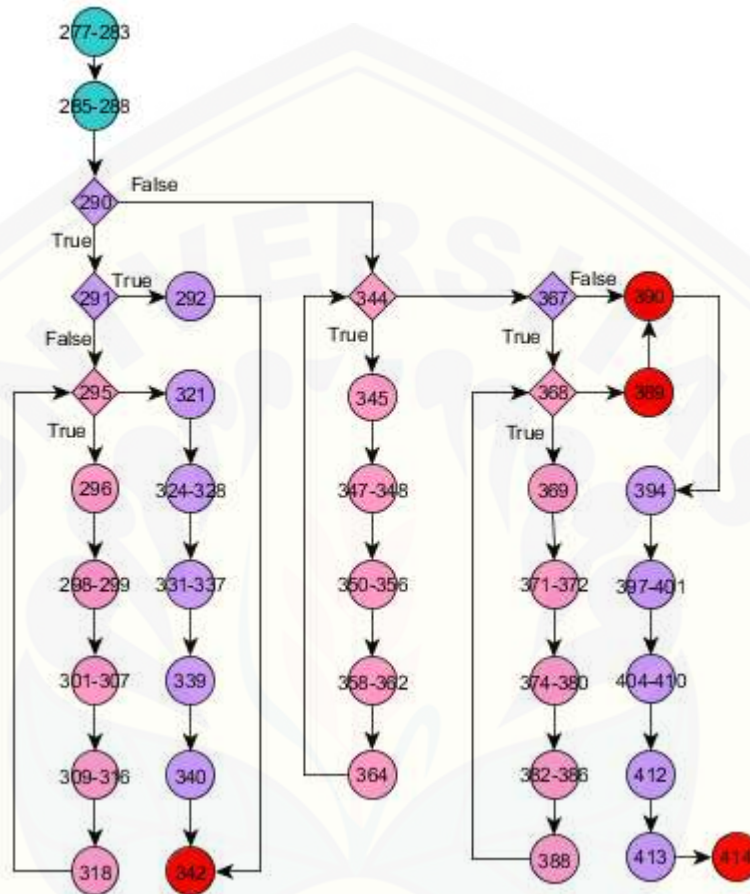
```

275 function proses_plagiat()
276 {
277     $penulis_utama = $this->input->post('penulis');
278     $judul = $this->input->post('judul');
279     $bidang_studi = $this->input->post('bidang_studi');
280     $bibliografi = $this->input->post('bibliografi');
281     $nama_file = $this->input->post('nama_file');
282     $hasil_stem = $this->input->post('isi_jurnal');
283     $fingerprint = $this->input->post('fingerprint_jurnal');
284
285     $result = array();
286     $result_array = array();
287     $arr_fingerprint = explode(" ", $fingerprint);
288     array_map($arr_fingerprint);
289
290     if($this->input->post('file_sali') > 0){
291         if($this->input->post('jurnal_referensi') > 1){
292             echo "Tidak Ada File yang Akan Di Bandingkan";
293         }
294         else{
295             foreach ($this->input->post('jurnal_referensi') as $id_referensi){
296                 $query = $this->m_jurnal->get_jurnal_by_id($id_referensi);
297
298                 $arr_fingerprint_jurnal = explode(" ", $query->row('fingerprint'));
299                 array_map($arr_fingerprint_jurnal);
300
301                 $intersect1 = array_intersect($arr_fingerprint_jurnal, $arr_fingerprint);
302                 $intersect2 = array_intersect($arr_fingerprint, $arr_fingerprint_jurnal);
303                 $diff1 = array_diff($arr_fingerprint_jurnal, $arr_fingerprint);
304                 $diff2 = array_diff($arr_fingerprint, $arr_fingerprint_jurnal);
305                 $sama = count(array_intersect($arr_fingerprint_jurnal, $arr_fingerprint));
306                 $jumlah = (count($diff1) + count($diff2) + $sama);
307                 $hasil = ($sama / $jumlah) * 100;
308
309                 $result['id_jurnal'] = $query->row('id_jurnal');
310                 $result['bibliografi'] = $query->row('bibliografi');
311                 $result['fingerprint'] = $query->row('fingerprint');
312                 $result['sama'] = $sama;
313                 $result['diff1'] = count($diff1);
314                 $result['diff2'] = count($diff2);
315                 $result['hasil'] = round($hasil, 2);
316                 $result['tipe'] = $this->tipe_plagiat($hasil);
317
318                 array_push($result_array, $result);
319             }
320
321             $data['result'] = $result_array;
322
323             //data header
324             $data['check_srtif'] = "active";
325             $data['level'] = $this->session->userdata('level');
326             $data['username'] = $this->session->userdata('username');
327             $data['nama_lengkap'] = $this->session->userdata('nama_lengkap');
328             $data['pict'] = $this->session->userdata('pict');
329
330             //data footer
331             $data['penulis_utama'] = $penulis_utama;
332             $data['judul'] = $judul;
333             $data['bidang_studi'] = $bidang_studi;
334             $data['bibliografi'] = $bibliografi;
335             $data['file_name'] = $nama_file;
336             $data['isi_jurnal'] = $hasil_stem;
337             $data['fingerprint'] = $fingerprint;
338
339             $this->load->view('elements/user_header', $data);
340             $this->load->view('pages/user/ri_hasilcheck', $data);
341         }
342     }
343     else{
344         foreach ($this->input->post('file_sali') as $id_jurnal) {
345             $query = $this->m_jurnal->get_jurnal_by_id($id_jurnal);

```

```
347 $arr_fingerprint_jurnal = explode(" ", $query->row('fingerprint'));
348 array_pop($arr_fingerprint_jurnal);
349
350 $intersect1 = array_intersect($arr_fingerprint_jurnal, $arr_fingerprint);
351 $intersect2 = array_intersect($arr_fingerprint, $arr_fingerprint_jurnal);
352 $diff1 = array_diff($arr_fingerprint_jurnal, $arr_fingerprint);
353 $diff2 = array_diff($arr_fingerprint, $arr_fingerprint_jurnal);
354 $sama = count(array_intersect($arr_fingerprint_jurnal, $arr_fingerprint));
355 $jumlah = (count($diff1) + count($diff2) + $sama);
356 $hasil = ($sama / $jumlah) * 100;
357
358 $result['id_jurnal'] = $query->row('id_jurnal');
359 $result['bibliografi'] = $query->row('bibliografi');
360 $result['fingerprint'] = $query->row('fingerprint');
361 $result['hasil'] = round($hasil, 2);
362 $result['tipe'] = $this->tipe_slaglat($hasil);
363
364 array_push($result_array, $result);
365
366 }
367
368 if($this->input->post('jurnal_referensi') > 0){
369     $search = ($this->input->post('jurnal_referensi') && $id_referensi){
370         $query = $this->m_jurnal->get_jurnal_by_id($id_referensi);
371         $arr_fingerprint_jurnal = explode(" ", $query->row('fingerprint'));
372         array_pop($arr_fingerprint_jurnal);
373
374         $intersect1 = array_intersect($arr_fingerprint_jurnal, $arr_fingerprint);
375         $intersect2 = array_intersect($arr_fingerprint, $arr_fingerprint_jurnal);
376         $diff1 = array_diff($arr_fingerprint_jurnal, $arr_fingerprint);
377         $diff2 = array_diff($arr_fingerprint, $arr_fingerprint_jurnal);
378         $sama = count(array_intersect($arr_fingerprint_jurnal, $arr_fingerprint));
379         $jumlah = (count($diff1) + count($diff2) + $sama);
380         $hasil = ($sama / $jumlah) * 100;
381
382         $result['id_jurnal'] = $query->row('id_jurnal');
383         $result['bibliografi'] = $query->row('bibliografi');
384         $result['fingerprint'] = $query->row('fingerprint');
385         $result['hasil'] = round($hasil, 2);
386         $result['tipe'] = $this->tipe_slaglat($hasil);
387
388         array_push($result_array, $result);
389     }
390 }
391
392
393 $data['result'] = $result_array;
394
395 //data header
396 $data['check_aktif'] = "active";
397 $data['level'] = $this->session->userdata('level');
398 $data['username'] = $this->session->userdata('username');
399 $data['nama_lengkap'] = $this->session->userdata('nama_lengkap');
400 $data['pic'] = $this->session->userdata('pic');
401
402 //data header
403 $data['penulis_utama'] = $penulis_utama;
404 $data['judul'] = $judul;
405 $data['bidang_studi'] = $bidang_studi;
406 $data['biblio'] = $bibliografi;
407 $data['file_name'] = $sama_file;
408 $data['int_file'] = $hasil_stem;
409 $data['fingerprint'] = $fingerprint;
410
411
412 $this->load->view('layouts/user/header', $data);
413 $this->load->view('pages/user/v_hasilcheck', $data);
414
415 }
416 }
```

b) Grafik Alir



c) Perhitungan *Cyclomatic Complexity*

Perhitungan cyclomatic complexity $V(G)$ untuk grafik alir fitur mengecek plagiat jurnal adalah $V(G) = \text{Edge} - \text{Node} + 2 = 42 - 38 + 2 = 6$

d) Basis Set

Basis set menghasilkan 4 jalur independen

Path 1 : 277-278-279-280-281-282-283-285-286-287-288-290-291-292-342

Path 2 : 277-278-279-280-281-282-283-285-286-287-288-290-291-295-296-298-299-301-302-303-304-305-306-307-309-310-311-312-313-314-315-316-318-321-324-325-326-327-328-331-332-333-334-335-336-

337-339-340-342

Path 3 : 277-278-279-280-281-282-283-285-286-287-288-290-344-345-347-348-350-351-352-353-354-355-356-358-359-360-361-362-364-367-368-369-371-372-374-375-376-377-378-379-380-382-383-384-385-386-388-389-390-394-397-398-399-400-401-404-405-406-407-408-409-410-412-413-414

Path 4 : 277-278-279-280-281-282-283-285-286-287-288-290-344-345-347-348-350-351-352-353-354-355-356-358-359-360-361-362-364-367-390-394-397-398-399-400-401-404-405-406-407-408-409-410-412-413-414

e) *Test Case*

<i>Test Case</i> Fitur Mengecek Plagiat Jurnal	
<i>Test case</i>	Jika jurnal referensi dari <i>database</i> dan jurnal referensi <i>upload</i> tidak terisi
Target yang diharapkan	Menampilkan pesan tidak ada jurnal yang dibandingkan
Hasil Pengujian	Benar
Path	277-278-279-280-281-282-283-285-286-287-288-290-291-292-342
<i>Test Case</i> Fitur Mengecek Plagiat Jurnal	
<i>Test case</i>	Jika hanya jurnal referensi <i>upload</i> yang terisi
Target yang diharapkan	Melakukan penghitungan perbandingan dengan jurnal referensi yang diunggah
Hasil Pengujian	Benar
Path	277-278-279-280-281-282-283-285-286-287-288-290-291-295-296-298-299-301-302-303-304-305-306-307-309-310-311-312-313-314-315-316-318-321-324-325-326-327-328-331-332-333-334-335-336-337-339-340-342
<i>Test Case</i> Fitur Mengecek Plagiat Jurnal	
<i>Test case</i>	Jika jurnal referensi dari <i>database</i> dan jurnal referensi <i>upload</i> terisi
Target yang diharapkan	Melakukan penghitungan perbandingan dengan jurnal referensi terpilih
Hasil Pengujian	Benar
Path	277-278-279-280-281-282-283-285-286-287-288-290-344-345-347-348-350-351-352-353-354-355-356-358-359-360-361-362-364-367-368-369-371-372-374-375-376-377-378-379-380-382-383-384-385-386-388-389-390-394-397-398-

	399-400-401-404-405-406-407-408-409-410-412-413-414
	<i>Test Case</i> Fitur Mengecek Plagiat Jurnal
<i>Test case</i>	Jika hanya jurnal dari <i>database</i> yang dipilih
Target yang diharapkan	Melakukan penghitungan perbandingan dengan jurnal dari <i>database</i>
Hasil Pengujian	Benar
Path	277-278-279-280-281-282-283-285-286-287-288-290-344-345-347-348-350-351-352-353-354-355-356-358-359-360-361-362-364-367-390-394-397-398-399-400-401-404-405-406-407-408-409-410-412-413-414

C2. Black Box Testing

1. Fitur Login ke Sistem

Kasus	Hasil	Ket
<i>Field username</i> dan <i>password</i> terisi, kemudian <i>user</i> klik tombol “ <i>sign in</i> ”	Melakukan proses <i>login</i> dan menampilkan halaman utama sesuai level <i>user</i>	[√] Berhasil [] Gagal
<i>Field username</i> dan <i>password</i> kosong, kemudian <i>user</i> klik tombol “ <i>sign in</i> ”	Menampilkan pesan <i>error</i> pada <i>field</i> yang kosong	[√] Berhasil [] Gagal
<i>Username</i> dan <i>password</i> tidak cocok, kemudian <i>user</i> klik tombol “ <i>sign in</i> ”	Menampilkan pesan <i>error</i> “ <i>username dan password salah</i> ”	[√] Berhasil [] Gagal

2. Fitur Register User Baru

Kasus	Hasil	Ket
<i>Field</i> pada <i>form sign up</i> terisi dengan benar, kemudian <i>user</i> klik tombol “ <i>sign up</i> ”	Menyimpan data <i>user</i> baru dan menampilkan halaman <i>login</i>	[√] Berhasil [] Gagal
<i>Field</i> pada <i>form sign up</i> tidak terisi, kemudian <i>user</i> klik tombol “ <i>sign up</i> ”	Menampilkan pesan <i>error</i> pada <i>field</i> yang kosong	[√] Berhasil [] Gagal
<i>Input</i> tidak <i>valid</i> pada <i>form sign up</i> , kemudian <i>user</i> klik tombol “ <i>sign up</i> ”	Menampilkan pesan <i>error</i> pada <i>field</i>	[√] Berhasil [] Gagal

3. Fitur Tambah Data Jurnal

Kasus	Hasil	Ket
<i>Field</i> pada <i>form upload</i> terisi dengan benar, kemudian <i>user</i> klik tombol “ <i>upload</i> ”	Menyimpan <i>file</i> jurnal ke dalam direktori <i>website</i> dan menampilkan <i>preview</i> jurnal yang diunggah	[√] Berhasil [] Gagal
<i>Field</i> pada <i>form upload</i> tidak terisi dengan benar atau kosong, kemudian <i>user</i> klik tombol “ <i>upload</i> ”	Menampilkan pesan <i>error</i> pada <i>field</i>	[√] Berhasil [] Gagal
Klik tombol “ <i>proses</i> ” pada halaman <i>preview</i> jurnal	Melakukan proses pembentukan <i>fingerprint</i> dan menampilkan hasil perhitungan	[√] Berhasil [] Gagal
Klik tombol “ <i>cancel</i> ” pada halaman <i>preview</i> jurnal	Menghapus <i>file</i> jurnal dan menampilkan halaman <i>form upload</i> jurnal	[√] Berhasil [] Gagal
Klik tombol “ <i>proses</i> ” pada halaman hasil perhitungan <i>fingerprint</i>	Melakukan proses cek pada semua data jurnal di <i>database</i> dan menampilkan persentase kemiripan dokumen hasil pengecekan dari <i>database</i> dalam bentuk tabel	[√] Berhasil [] Gagal
Klik tombol “ <i>tampilkan detail perhitungan</i> ” pada halaman hasil perhitungan <i>fingerprint</i>	Menampilkan detail perhitungan pembentukan <i>fingerprint</i> jurnal	[√] Berhasil [] Gagal
Klik tombol “ <i>simpan ke database</i> ” pada halaman hasil cek	Menyimpan data jurnal ke dalam <i>database</i> dan menampilkan data <i>list</i> jurnal yang ada di <i>database</i>	[√] Berhasil [] Gagal
Klik tombol “ <i>cancel</i> ” pada halaman hasil cek	Menghapus <i>file</i> jurnal dan menampilkan halaman <i>form upload</i> jurnal	[√] Berhasil [] Gagal

4. Fitur Hapus Data Jurnal

Kasus	Hasil	Ket
Klik <i>icon</i> tombol “ <i>hapus</i> ” pada data yang akan dihapus	Menampilkan <i>pop-up</i> konfirmasi hapus data jurnal	[√] Berhasil [] Gagal
Klik tombol “ <i>confirm</i> ” pada <i>pop-up</i>	Menghapus data jurnal dan menampilkan halaman <i>list</i> jurnal	[√] Berhasil [] Gagal
Klik tombol “ <i>close</i> ” pada <i>pop-up</i>	Menampilkan halaman <i>list</i> jurnal beserta data jurnal	[√] Berhasil [] Gagal

5. Fitur Approve Data Jurnal

Kasus	Hasil	Ket
Klik <i>icon</i> tombol “ <i>setujui</i> ” pada data yang akan disetujui	Menampilkan <i>pop-up</i> konfirmasi setuju data jurnal	[√] Berhasil [] Gagal
Klik tombol “ <i>confirm</i> ” pada <i>pop-up</i> konfirmasi setuju data jurnal	Menyimpan data jurnal dan menampilkan halaman <i>approve</i> jurnal beserta data jurnal dengan status <i>pending</i>	[√] Berhasil [] Gagal
Klik tombol “ <i>close</i> ” pada <i>pop-up</i>	Menampilkan halaman <i>approve</i> jurnal beserta data jurnal dengan status <i>pending</i>	[√] Berhasil [] Gagal
Klik <i>icon</i> tombol “ <i>tidak menyetujui</i> ” pada data jurnal	Menampilkan <i>pop-up</i> konfirmasi tidak menyetujui data jurnal	[√] Berhasil [] Gagal
Klik tombol “ <i>confirm</i> ” pada <i>pop-up</i> konfirmasi tidak menyetujui data jurnal	Menghapus data jurnal dan menampilkan halaman <i>approve</i> jurnal beserta data jurnal dengan status <i>pending</i>	[√] Berhasil [] Gagal
Klik <i>icon</i> tombol “ <i>detail</i> ” pada data jurnal	Menampilkan <i>pop-up</i> detail informasi jurnal	[√] Berhasil [] Gagal
Klik tombol “ <i>close</i> ” pada <i>pop-up</i> detail informasi jurnal	Menampilkan halaman <i>approve</i> jurnal beserta data jurnal dengan status <i>pending</i>	[√] Berhasil [] Gagal

6. Fitur Nonaktifkan User

Kasus	Hasil	Ket
Klik <i>icon</i> tombol “ <i>nonaktifkan user</i> ” pada data <i>user</i>	Menampilkan <i>pop-up</i> konfirmasi nonaktifkan <i>user</i>	[√] Berhasil [] Gagal
Klik tombol “ <i>confirm</i> ” pada <i>pop-up</i>	Mengupdate status <i>user</i> dan menampilkan halaman <i>list user</i>	[√] Berhasil [] Gagal
Klik tombol “ <i>close</i> ” pada <i>pop-up</i>	Menampilkan halaman <i>list user</i> beserta data <i>user</i>	[√] Berhasil [] Gagal

7. Fitur Ubah Profil User

Kasus	Hasil	Ket
Klik tombol “ <i>ubah profil</i> ”	Menampilkan <i>pop-up update</i> profil beserta <i>form update</i> profil	[√] Berhasil [] Gagal
Klik tombol “ <i>simpan</i> ” pada <i>pop-up</i>	Mengupdate detail <i>user</i> dan menampilkan halaman <i>profil user</i>	[√] Berhasil [] Gagal
Klik tombol “ <i>batal</i> ” pada <i>pop-up</i>	Menampilkan halaman <i>profil user</i>	[√] Berhasil [] Gagal

8. Fitur Ubah Password Login User

Kasus	Hasil	Ket
Klik tombol “ <i>update password</i> ” dan <i>field</i> terisi dengan benar	Memperbarui <i>password login user</i> dan menampilkan halaman <i>update password</i>	[√] Berhasil [] Gagal
Klik tombol “ <i>update password</i> ” dan <i>field</i> tidak terisi	Menampilkan pesan <i>error</i> pada <i>field</i>	[√] Berhasil [] Gagal

9. Fitur View Log Aktifitas User

Kasus	Hasil	Ket
Klik menu “ <i>log</i> ”	Menampilkan halaman <i>log</i> aktifitas <i>user</i> beserta data <i>log</i> aktifitas <i>user</i>	[√] Berhasil [] Gagal

10. Fitur *View Data Jurnal*

Kasus	Hasil	Ket
Klik menu “ <i>list jurnal</i> ”	Menampilkan halaman <i>list jurnal</i> beserta semua data jurnal aktifitas <i>user</i>	[√] Berhasil [] Gagal
Klik <i>icon</i> tombol “ <i>detail</i> ” pada data jurnal	Menampilkan <i>pop-up</i> detail informasi jurnal	[√] Berhasil [] Gagal
Klik tombol “ <i>close</i> ” pada <i>pop-up</i>	Menampilkan halaman <i>list jurnal</i> beserta semua data jurnal	[√] Berhasil [] Gagal

11. Fitur *View Data User*

Kasus	Hasil	Ket
Klik menu “ <i>list user</i> ”	Menampilkan halaman <i>list user</i> beserta data <i>user</i>	[√] Berhasil [] Gagal
Klik <i>icon</i> tombol “ <i>detail user</i> ” pada data <i>user</i>	Menampilkan <i>pop-up</i> detail informasi <i>user</i>	[√] Berhasil [] Gagal
Klik tombol “ <i>close</i> ” pada <i>pop-up</i>	Menampilkan halaman <i>list user</i> beserta data <i>user</i>	[√] Berhasil [] Gagal

12. Fitur Mengecek Plagiat Jurnal

Kasus	Hasil	Ket
<i>Field</i> pada <i>form upload</i> terisi dengan benar, kemudian klik tombol “ <i>upload</i> ”	Menyimpan <i>file</i> jurnal ke dalam direktori <i>website</i> dan menampilkan <i>preview</i> jurnal yang diunggah	[√] Berhasil [] Gagal
<i>Field</i> pada <i>form upload</i> tidak terisi dengan benar atau kosong, kemudian <i>user</i> klik tombol “ <i>upload</i> ”	Menampilkan pesan <i>error</i> pada <i>field</i>	[√] Berhasil [] Gagal
Klik tombol “ <i>proses</i> ” pada halaman <i>preview</i> jurnal	Melakukan proses pembentukan <i>fingerprint</i> dan menampilkan hasil perhitungan	[√] Berhasil [] Gagal
Klik tombol “ <i>cancel</i> ” pada halaman <i>preview</i> jurnal	Menghapus <i>file</i> jurnal dan menampilkan halaman <i>form upload</i> jurnal	[√] Berhasil [] Gagal

Klik tombol “ <i>proses</i> ” pada halaman hasil perhitungan <i>fingerprint</i>	Menampilkan halaman “ <i>pilih jurnal</i> ”	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
Klik tombol “ <i>tampilkan detail perhitungan</i> ” pada halaman hasil perhitungan <i>fingerprint</i>	Menampilkan detail perhitungan pembentukan <i>fingerprint</i> jurnal	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
Memilih bidang studi pada <i>combobox</i> bidang studi	Menampilkan jurnal yang ada di <i>database</i> sesuai bidang studi terpilih pada <i>combobox</i> list jurnal	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
Klik tombol “ <i>tambahkan</i> ” pada halaman “ <i>pilih jurnal</i> ”	Menampilkan <i>pop-up upload</i> jurnal referensi	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
<i>Field</i> pada <i>form upload</i> terisi dengan benar pada <i>pop-up upload</i> jurnal referensi, kemudian <i>user</i> klik tombol “ <i>upload</i> ”	Menyimpan <i>file</i> jurnal referensi ke dalam direktori <i>website</i> dan menampilkan <i>pop-up</i> berhasil <i>upload</i>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
Klik tombol “ <i>confirm</i> ” <i>pop-up</i> berhasil <i>upload</i>	Menambahkan jurnal referensi yang diunggah ke dalam <i>list</i> jurnal referensi	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
Klik “ <i>proses</i> ” pada halaman “ <i>pilih jurnal</i> ”	Melakukan proses cek jurnal yang diunggah dengan jurnal terpilih dan menampilkan persentase plagiat jurnal	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

LAMPIRAN D. TABEL PERHITUNGAN

1. Detail Tabel Hasil Pembentukan *Gram* pada Kalimat tanpa *Stemming*

Kalimat Pembanding	Kalimat Uji
[sistema][istemad][stemada][temadal][emada] dala][madalah][adalahs][dalahsu][alahsua] [lahsuat][ahsuatu][hsuatuk][suatuke][uatukes] [atukesa][tukesat][ukesatu][kesatua][esatuan] [satuany][atuanya][tuanyan][uanyang] [anyangt][nyangte][yangter][angterd][ngterdi] [gterdir][gterdir][terdiri][erdirid][rdirida][diridar] [iridari][ridarik][idariko][darikom][arikomp] [rikompo][ikompon][kompone][omponen] [mponena][ponenat][onenata][nenatau] [enataue][natauel][atauele][tauelem][aueleme] [uelemen][elemend][lemenda][emendan] [mendand][endandi][ndandih][dandihu] [andihub][ndihubu][dihubun][ihubung] [hubungk][ubungka][bungkan][ungkanb] [ngkanbe][gkanber][kanbers][anbersa][nbersam] [bersama][ersamau][rsamaun][samaunt] [amauntu][mauntuk][auntukm][untukme] [ntukmem][tukmemu][ukmemud][kmemuda] [memudah][emudahk][mudahka][udahkan] [dahkana][ahkanal][hkanali][kanalir][analira] [naliran][alirani][liranin][iraninf][raninfo] [aninfor][ninform][informa][nformas] [formasi][ormasid][rmaside][masidem][asidemi] [sidemim][idemime][demimen][emimenc] [mimenca][imencap][mencapa][encapai] [ncapait][capaitu][apaituj][paituju][aitujuan] [itujuan]	[sistema][istemad][stemada][temadal][emada] ala][madalah][adalahs][dalahsu][alahsua][lahsuat] [ahsuatu][hsuatuk][suatuke][uatukes][atukesa] [tukesat][ukesatu][kesatua][esatuan][satuany] [atuanya][tuanyan][uanyang][anyangt][nyangte] [nyangter][angterd][ngterdi][gterdir][terdiri] [erdirid][rdirida][diridar][iridari][ridarik] [idariko][darikom][arikomp][rikompo][ikompon] [kompone][omponen][mponena][ponenat] [onenata][nenatau][enataue][natauel] [atauele][tauelem][aueleme][uelemen] [elemend][lemenda][emendan][mendand] [endandi][ndandih][dandihu][andihub] [ndihubu][dihubun][ihubung][hubungk] [ubungka][bungkan][ungkanb][ngkanbe] [gkanber][kanbers][anbersa][nbersam] [bersama][ersamau][rsamaun][samaunt] [amauntu][mauntuk][auntukm][untukme] [ntukmem][tukmemu][ukmemud][kmemuda] [memudah][emudahk][mudahka][udahkan] [dahkana][ahkanal][hkanali][kanalir] [analira][naliran]

2. Detail Tabel Hasil Pembentukan *Gram* pada Kalimat dengan *Stemming*

Kalimat Pembanding	Kalimat Uji
[sistema][istemad][stemada][temadal][emadala][madalah][adalahs][dalahsu][alahsua][lahsuat][ahsuatu][hsuatus][suatusa][uatusat][atusatu][tusatuy][usatuya][satuyan][atuyang][tuyangd][uyangdi][yangdir][angdiri][ngdirid][gdirida][diridar][iridari][ridarik][idariko][darikom][arikomp][rikompo][ikompon][kompone][omponen][mponena][ponenat][onenata][nenataue][natauel][atauele][tauelem][aueleme][uelemen][elemend][lemenda][emendan][mendanh][endanhu][ndanhub][danhubu][anhubun][nhubung][hubungs][ubungsa][bungsam][ungsama][ngsamau][gsamaun][samaunt][amauntu][mauntuk][auntukm][untukmu][ntukmud][tukmuda][ukmudah][kmudaha][mudahal][udahali][dahalir][ahaliri][halirin][alirinf][lirinfo][irinfor][rinform][informa][nformas][formasi][ormasid][rmaside][masidem][asidemi][sidemic][idemica][demicap][emicapa][micapai][icapait][capaitu][apaituj][paituju]	[sistema][istemad][stemada][temadal][emadala][madalah][adalahs][dalahsu][alahsua][lahsuat][ahsuatu][hsuatus][suatusa][uatusat][atusatu][tusatuy][usatuya][satuyan][atuyang][tuyangd][uyangdi][yangdir][angdiri][ngdirid][gdirida][diridar][iridari][ridarik][idariko][darikom][arikomp][rikompo][ikompon][kompone][omponen][mponena][ponenat][onenata][nenataue][natauel][atauele][tauelem][aueleme][uelemen][elemend][lemenda][emendan][mendanh][endanhu][ndanhub][danhubu][anhubun][nhubung][hubungs][ubungsa][bungsam][ungsama][ngsamau][gsamaun][samaunt][amauntu][mauntuk][auntukm][untukmu][ntukmud][tukmuda][ukmudah][kmudah][udahali][dahalir]

3. Detail Tabel Nilai *Hash* pada Kalimat tanpa *Stemming*

Kalimat Pembanding	Kalimat Uji
123130, 117985, 124417, 121854, 111967, 115118, 107086, 109236, 109105, 115292, 109797, 117359, 124730, 122800, 112618, 125831, 123918, 115972, 114017, 121285, 112447, 125312, 122347, 111278, 121796, 124932, 110269, 118773, 115863, 122433, 113707, 120331, 111789, 116772, 120788, 113157, 109945, 111247, 121713, 115931, 118259, 120878, 119974, 121655, 120118, 117714, 112673, 117240, 111251, 121723, 111578, 122705, 112336, 116218, 112568, 116917, 112473, 116636, 109455, 109763, 117267, 111341, 115426, 116750, 122899, 112928, 124556, 117890, 113214, 114496, 109576, 116698, 109621, 114654, 123185, 120353, 109671, 116981, 112669, 125969, 122137, 125958, 124282, 117064, 117287, 113585, 119965, 121622, 109084, 108660, 113946, 114504, 109600, 116771, 109848, 117515, 116451, 119829, 110283, 118819, 115984, 118432, 114831, 121519, 121901, 116494, 111204, 121582, 113342, 110501, 112902, 117916, 115477, 116893, 112401, 116432, 108843, 110122, 118344, 110185, 118526	123130, 117985, 124417, 121854, 111967, 115118, 107086, 109236, 109105, 115292, 109797, 117359, 124730, 122800, 112618, 125831, 123918, 115972, 114017, 121285, 112447, 125312, 122347, 111278, 121796, 124932, 110269, 118773, 115863, 122433, 113707, 120331, 111789, 116772, 120788, 113157, 109945, 111247, 121713, 115931, 118259, 120878, 119974, 121655, 120118, 117714, 112673, 117240, 111251, 121723, 111578, 122705, 112336, 116218, 112568, 116917, 112473, 116636, 109455, 109763, 117267, 111341, 115426, 116750, 122899, 112928, 124556, 117890, 113214, 114496, 109576, 116698, 109621, 114654, 123185, 120353, 109671, 116981, 112669, 125969, 122137, 125958, 124282, 117064, 117287, 113585, 119965, 121622, 109084, 108660, 113946, 114504, 109600, 116771

4. Detail Tabel Nilai *Hash* pada Kalimat dengan *Stemming*

Kalimat Pembanding	Kalimat Uji
123130, 117985, 124417, 121854, 111967, 115118, 107086, 109236, 109105, 115292, 109797, 117367, 124750, 122861, 112821, 126445, 125740, 121451, 112951, 126814, 126855, 124800, 109878, 117595, 112312, 111789, 116772, 120788, 113157, 109945, 111247, 121713, 115931, 118259, 120878, 119974, 121655, 120118, 117714, 112673, 117240, 111251, 121723, 111578, 122705, 112336, 116218, 112568, 116921, 112497, 116702, 109653, 110369, 119071, 116758, 122923, 112999, 124768, 118542, 115166, 120353, 109671, 116981, 112669, 125985, 122176, 126055, 124577, 117949, 119946, 121560, 108915, 108150, 112421, 109917, 117723, 117087, 121735, 115984, 118432, 114831, 121519, 121901, 116494, 111204, 121572, 113308, 110401, 112600, 117018, 112787, 108843, 110122, 118344	123130, 117985, 124417, 121854, 111967, 115118, 107086, 109236, 109105, 115292, 109797, 117367, 124750, 122861, 112821, 126445, 125740, 121451, 112951, 126814, 126855, 124800, 109878, 117595, 112312, 111789, 116772, 120788, 113157, 109945, 111247, 121713, 115931, 118259, 120878, 119974, 121655, 120118, 117714, 112673, 117240, 111251, 121723, 111578, 122705, 112336, 116218, 112568, 116921, 112497, 116702, 109653, 110369, 119071, 116758, 122923, 112999, 124768, 118542, 115166, 120353, 109671, 116981, 112669, 125985, 122176, 126055, 124577, 117949, 119946, 121560, 108915

5. Detail Tabel Hasil Pembentukan *window* pada Kalimat tanpa *Stemming*

Kalimat Pembanding	Kalimat Uji
[123130,117985,124417,121854,111967,115118]	[123130,117985,124417,121854,111967,115118]
[117985,124417,121854,111967,115118,107086]	[117985,124417,121854,111967,115118,107086]
[124417,121854,111967,115118,107086,109236]	[124417,121854,111967,115118,107086,109236]
[121854,111967,115118,107086,109236,109105]	[121854,111967,115118,107086,109236,109105]
[111967,115118,107086,109236,109105,115292]	[111967,115118,107086,109236,109105,115292]
[115118,107086,109236,109105,115292,109797]	[115118,107086,109236,109105,115292,109797]
[107086,109236,109105,115292,109797,117359]	[107086,109236,109105,115292,109797,117359]
[109236,109105,115292,109797,117359,124730]	[109236,109105,115292,109797,117359,124730]
[109105,115292,109797,117359,124730,122800]	[109105,115292,109797,117359,124730,122800]
[115292,109797,117359,124730,122800,112618]	[115292,109797,117359,124730,122800,112618]
[109797,117359,124730,122800,112618,125831]	[109797,117359,124730,122800,112618,125831]
[117359,124730,122800,112618,125831,123918]	[117359,124730,122800,112618,125831,123918]
[124730,122800,112618,125831,123918,115972]	[124730,122800,112618,125831,123918,115972]
[122800,112618,125831,123918,115972,114017]	[122800,112618,125831,123918,115972,114017]
[112618,125831,123918,115972,114017,121285]	[112618,125831,123918,115972,114017,121285]
[125831,123918,115972,114017,121285,112447]	[125831,123918,115972,114017,121285,112447]
[123918,115972,114017,121285,112447,125312]	[123918,115972,114017,121285,112447,125312]
[115972,114017,121285,112447,125312,122347]	[115972,114017,121285,112447,125312,122347]
[114017,121285,112447,125312,122347,111278]	[114017,121285,112447,125312,122347,111278]
[121285,112447,125312,122347,111278,121796]	[121285,112447,125312,122347,111278,121796]
[112447,125312,122347,111278,121796,124932]	[112447,125312,122347,111278,121796,124932]

[125312,122347,111278,121796,124932,110269]	[125312,122347,111278,121796,124932,110269]
[122347,111278,121796,124932,110269,118773]	[122347,111278,121796,124932,110269,118773]
[111278,121796,124932,110269,118773,115863]	[111278,121796,124932,110269,118773,115863]
[121796,124932,110269,118773,115863,122433]	[121796,124932,110269,118773,115863,122433]
[124932,110269,118773,115863,122433,113707]	[124932,110269,118773,115863,122433,113707]
[110269,118773,115863,122433,113707,120331]	[110269,118773,115863,122433,113707,120331]
[118773,115863,122433,113707,120331,111789]	[118773,115863,122433,113707,120331,111789]
[115863,122433,113707,120331,111789,116772]	[115863,122433,113707,120331,111789,116772]
[122433,113707,120331,111789,116772,120788]	[122433,113707,120331,111789,116772,120788]
[113707,120331,111789,116772,120788,113157]	[113707,120331,111789,116772,120788,113157]
[120331,111789,116772,120788,113157,109945]	[120331,111789,116772,120788,113157,109945]
[111789,116772,120788,113157,109945,111247]	[111789,116772,120788,113157,109945,111247]
[116772,120788,113157,109945,111247,121713]	[116772,120788,113157,109945,111247,121713]
[120788,113157,109945,111247,121713,115931]	[120788,113157,109945,111247,121713,115931]
[113157,109945,111247,121713,115931,118259]	[113157,109945,111247,121713,115931,118259]
[109945,111247,121713,115931,118259,120878]	[109945,111247,121713,115931,118259,120878]
[111247,121713,115931,118259,120878,119974]	[111247,121713,115931,118259,120878,119974]
[121713,115931,118259,120878,119974,121655]	[121713,115931,118259,120878,119974,121655]
[115931,118259,120878,119974,121655,120118]	[115931,118259,120878,119974,121655,120118]
[118259,120878,119974,121655,120118,117714]	[118259,120878,119974,121655,120118,117714]
[120878,119974,121655,120118,117714,112673]	[120878,119974,121655,120118,117714,112673]
[119974,121655,120118,117714,112673,117240]	[119974,121655,120118,117714,112673,117240]
[121655,120118,117714,112673,117240,111251]	[121655,120118,117714,112673,117240,111251]

[120118,117714,112673,117240,111251,121723]	[120118,117714,112673,117240,111251,121723]
[117714,112673,117240,111251,121723,111578]	[117714,112673,117240,111251,121723,111578]
[112673,117240,111251,121723,111578,122705]	[112673,117240,111251,121723,111578,122705]
[117240,111251,121723,111578,122705,112336]	[117240,111251,121723,111578,122705,112336]
[111251,121723,111578,122705,112336,116218]	[111251,121723,111578,122705,112336,116218]
[121723,111578,122705,112336,116218,112568]	[121723,111578,122705,112336,116218,112568]
[111578,122705,112336,116218,112568,116917]	[111578,122705,112336,116218,112568,116917]
[122705,112336,116218,112568,116917,112473]	[122705,112336,116218,112568,116917,112473]
[112336,116218,112568,116917,112473,116636]	[112336,116218,112568,116917,112473,116636]
[116218,112568,116917,112473,116636,109455]	[116218,112568,116917,112473,116636,109455]
[112568,116917,112473,116636,109455,109763]	[112568,116917,112473,116636,109455,109763]
[116917,112473,116636,109455,109763,117267]	[116917,112473,116636,109455,109763,117267]
[112473,116636,109455,109763,117267,111341]	[112473,116636,109455,109763,117267,111341]
[116636,109455,109763,117267,111341,115426]	[116636,109455,109763,117267,111341,115426]
[109455,109763,117267,111341,115426,116750]	[109455,109763,117267,111341,115426,116750]
[109763,117267,111341,115426,116750,122899]	[109763,117267,111341,115426,116750,122899]
[117267,111341,115426,116750,122899,112928]	[117267,111341,115426,116750,122899,112928]
[111341,115426,116750,122899,112928,124556]	[111341,115426,116750,122899,112928,124556]
[115426,116750,122899,112928,124556,117890]	[115426,116750,122899,112928,124556,117890]
[116750,122899,112928,124556,117890,113214]	[116750,122899,112928,124556,117890,113214]
[122899,112928,124556,117890,113214,114496]	[122899,112928,124556,117890,113214,114496]
[112928,124556,117890,113214,114496,109576]	[112928,124556,117890,113214,114496,109576]
[124556,117890,113214,114496,109576,116698]	[124556,117890,113214,114496,109576,116698]

[117890,113214,114496,109576,116698,109621]	[117890,113214,114496,109576,116698,109621]
[113214,114496,109576,116698,109621,114654]	[113214,114496,109576,116698,109621,114654]
[114496,109576,116698,109621,114654,123185]	[114496,109576,116698,109621,114654,123185]
[109576,116698,109621,114654,123185,120353]	[109576,116698,109621,114654,123185,120353]
[116698,109621,114654,123185,120353,109671]	[116698,109621,114654,123185,120353,109671]
[109621,114654,123185,120353,109671,116981]	[109621,114654,123185,120353,109671,116981]
[114654,123185,120353,109671,116981,112669]	[114654,123185,120353,109671,116981,112669]
[123185,120353,109671,116981,112669,125969]	[123185,120353,109671,116981,112669,125969]
[120353,109671,116981,112669,125969,122137]	[120353,109671,116981,112669,125969,122137]
[109671,116981,112669,125969,122137,125958]	[109671,116981,112669,125969,122137,125958]
[116981,112669,125969,122137,125958,124282]	[116981,112669,125969,122137,125958,124282]
[112669,125969,122137,125958,124282,117064]	[112669,125969,122137,125958,124282,117064]
[125969,122137,125958,124282,117064,117287]	[125969,122137,125958,124282,117064,117287]
[122137,125958,124282,117064,117287,113585]	[122137,125958,124282,117064,117287,113585]
[125958,124282,117064,117287,113585,119965]	[125958,124282,117064,117287,113585,119965]
[124282,117064,117287,113585,119965,121622]	[124282,117064,117287,113585,119965,121622]
[117064,117287,113585,119965,121622,109084]	[117064,117287,113585,119965,121622,109084]
[117287,113585,119965,121622,109084,108660]	[117287,113585,119965,121622,109084,108660]
[113585,119965,121622,109084,108660,113946]	[113585,119965,121622,109084,108660,113946]
[119965,121622,109084,108660,113946,114504]	[119965,121622,109084,108660,113946,114504]
[121622,109084,108660,113946,114504,109600]	[121622,109084,108660,113946,114504,109600]
[109084,108660,113946,114504,109600,116771]	[109084,108660,113946,114504,109600,116771]
[108660,113946,114504,109600,116771,109848]	[108660,113946,114504,109600,116771,109848]

[113946,114504,109600,116771,109848,117515]
[114504,109600,116771,109848,117515,116451]
[109600,116771,109848,117515,116451,119829]
[116771,109848,117515,116451,119829,110283]
[109848,117515,116451,119829,110283,118819]
[117515,116451,119829,110283,118819,115984]
[116451,119829,110283,118819,115984,118432]
[119829,110283,118819,115984,118432,114831]
[110283,118819,115984,118432,114831,121519]
[118819,115984,118432,114831,121519,121901]
[115984,118432,114831,121519,121901,116494]
[118432,114831,121519,121901,116494,111204]
[114831,121519,121901,116494,111204,121582]
[121519,121901,116494,111204,121582,113342]
[121901,116494,111204,121582,113342,110501]
[116494,111204,121582,113342,110501,112902]
[111204,121582,113342,110501,112902,117916]
[121582,113342,110501,112902,117916,115477]
[113342,110501,112902,117916,115477,116893]
[110501,112902,117916,115477,116893,112401]
[112902,117916,115477,116893,112401,116432]
[117916,115477,116893,112401,116432,108843]
[115477,116893,112401,116432,108843,110122]

[116893,112401,116432,108843,110122,118344]
[112401,116432,108843,110122,118344,110185]
[116432,108843,110122,118344,110185,118526]

6. Detail Tabel Hasil Pembentukan *window* pada Kalimat dengan *Stemming*

Kalimat Pembanding	Kalimat Uji
[123130,117985,124417,121854,111967,115118]	[123130,117985,124417,121854,111967,115118]
[117985,124417,121854,111967,115118,107086]	[117985,124417,121854,111967,115118,107086]
[124417,121854,111967,115118,107086,109236]	[124417,121854,111967,115118,107086,109236]
[121854,111967,115118,107086,109236,109105]	[121854,111967,115118,107086,109236,109105]
[111967,115118,107086,109236,109105,115292]	[111967,115118,107086,109236,109105,115292]
[115118,107086,109236,109105,115292,109797]	[115118,107086,109236,109105,115292,109797]
[107086,109236,109105,115292,109797,117367]	[107086,109236,109105,115292,109797,117367]
[109236,109105,115292,109797,117367,124750]	[109236,109105,115292,109797,117367,124750]
[109105,115292,109797,117367,124750,122861]	[109105,115292,109797,117367,124750,122861]
[115292,109797,117367,124750,122861,112821]	[115292,109797,117367,124750,122861,112821]
[109797,117367,124750,122861,112821,126445]	[109797,117367,124750,122861,112821,126445]
[117367,124750,122861,112821,126445,125740]	[117367,124750,122861,112821,126445,125740]
[124750,122861,112821,126445,125740,121451]	[124750,122861,112821,126445,125740,121451]
[122861,112821,126445,125740,121451,112951]	[122861,112821,126445,125740,121451,112951]
[112821,126445,125740,121451,112951,126814]	[112821,126445,125740,121451,112951,126814]
[126445,125740,121451,112951,126814,126855]	[126445,125740,121451,112951,126814,126855]
[125740,121451,112951,126814,126855,124800]	[125740,121451,112951,126814,126855,124800]
[121451,112951,126814,126855,124800,10]	[121451,112951,126814,126855,124800,10]

9878]	9878]
[112951,126814,126855,124800,109878,117595]	[112951,126814,126855,124800,109878,117595]
[126814,126855,124800,109878,117595,112312]	[126814,126855,124800,109878,117595,112312]
[126855,124800,109878,117595,112312,111789]	[126855,124800,109878,117595,112312,111789]
[124800,109878,117595,112312,111789,116772]	[124800,109878,117595,112312,111789,116772]
[109878,117595,112312,111789,116772,120788]	[109878,117595,112312,111789,116772,120788]
[117595,112312,111789,116772,120788,113157]	[117595,112312,111789,116772,120788,113157]
[112312,111789,116772,120788,113157,109945]	[112312,111789,116772,120788,113157,109945]
[111789,116772,120788,113157,109945,111247]	[111789,116772,120788,113157,109945,111247]
[116772,120788,113157,109945,111247,121713]	[116772,120788,113157,109945,111247,121713]
[120788,113157,109945,111247,121713,115931]	[120788,113157,109945,111247,121713,115931]
[113157,109945,111247,121713,115931,118259]	[113157,109945,111247,121713,115931,118259]
[109945,111247,121713,115931,118259,120878]	[109945,111247,121713,115931,118259,120878]
[111247,121713,115931,118259,120878,119974]	[111247,121713,115931,118259,120878,119974]
[121713,115931,118259,120878,119974,121655]	[121713,115931,118259,120878,119974,121655]
[115931,118259,120878,119974,121655,120118]	[115931,118259,120878,119974,121655,120118]
[118259,120878,119974,121655,120118,117714]	[118259,120878,119974,121655,120118,117714]
[120878,119974,121655,120118,117714,112673]	[120878,119974,121655,120118,117714,112673]
[119974,121655,120118,117714,112673,117240]	[119974,121655,120118,117714,112673,117240]
[121655,120118,117714,112673,117240,111251]	[121655,120118,117714,112673,117240,111251]
[120118,117714,112673,117240,111251,121723]	[120118,117714,112673,117240,111251,121723]
[117714,112673,117240,111251,121723,111578]	[117714,112673,117240,111251,121723,111578]
[112673,117240,111251,121723,111578,122705]	[112673,117240,111251,121723,111578,122705]
[117240,111251,121723,111578,122705,11	[117240,111251,121723,111578,122705,11

2336]	2336]
[111251,121723,111578,122705,112336,116218]	[111251,121723,111578,122705,112336,116218]
[121723,111578,122705,112336,116218,112568]	[121723,111578,122705,112336,116218,112568]
[111578,122705,112336,116218,112568,116921]	[111578,122705,112336,116218,112568,116921]
[122705,112336,116218,112568,116921,112497]	[122705,112336,116218,112568,116921,112497]
[112336,116218,112568,116921,112497,116702]	[112336,116218,112568,116921,112497,116702]
[116218,112568,116921,112497,116702,109653]	[116218,112568,116921,112497,116702,109653]
[112568,116921,112497,116702,109653,110369]	[112568,116921,112497,116702,109653,110369]
[116921,112497,116702,109653,110369,119071]	[116921,112497,116702,109653,110369,119071]
[112497,116702,109653,110369,119071,116758]	[112497,116702,109653,110369,119071,116758]
[116702,109653,110369,119071,116758,122923]	[116702,109653,110369,119071,116758,122923]
[109653,110369,119071,116758,122923,112999]	[109653,110369,119071,116758,122923,112999]
[110369,119071,116758,122923,112999,124768]	[110369,119071,116758,122923,112999,124768]
[119071,116758,122923,112999,124768,118542]	[119071,116758,122923,112999,124768,118542]
[116758,122923,112999,124768,118542,115166]	[116758,122923,112999,124768,118542,115166]
[122923,112999,124768,118542,115166,120353]	[122923,112999,124768,118542,115166,120353]
[112999,124768,118542,115166,120353,109671]	[112999,124768,118542,115166,120353,109671]
[124768,118542,115166,120353,109671,116981]	[124768,118542,115166,120353,109671,116981]
[118542,115166,120353,109671,116981,112669]	[118542,115166,120353,109671,116981,112669]
[115166,120353,109671,116981,112669,125985]	[115166,120353,109671,116981,112669,125985]
[120353,109671,116981,112669,125985,122176]	[120353,109671,116981,112669,125985,122176]
[109671,116981,112669,125985,122176,126055]	[109671,116981,112669,125985,122176,126055]
[116981,112669,125985,122176,126055,124577]	[116981,112669,125985,122176,126055,124577]
[112669,125985,122176,126055,124577,11	[112669,125985,122176,126055,124577,11

7949,] [125985,122176,126055,124577,117949,119946] [122176,126055,124577,117949,119946,121560] [126055,124577,117949,119946,121560,108915] [124577,117949,119946,121560,108915,108150] [117949,119946,121560,108915,108150,112421,] [119946,121560,108915,108150,112421,109917] [121560,108915,108150,112421,109917,117723] [108915,108150,112421,109917,117723,117087] [108150,112421,109917,117723,117087,121735] [112421,109917,117723,117087,121735,115984] [109917,117723,117087,121735,115984,118432] [117723,117087,121735,115984,118432,114831] [117087,121735,115984,118432,114831,121519] [121735,115984,118432,114831,121519,121901] [115984,118432,114831,121519,121901,116494] [118432,114831,121519,121901,116494,111204] [114831,121519,121901,116494,111204,121572] [121519,121901,116494,111204,121572,113308] [121901,116494,111204,121572,113308,110401] [116494,111204,121572,113308,110401,112600] [111204,121572,113308,110401,112600,117018] [121572,113308,110401,112600,117018,112787] [113308,110401,112600,117018,112787,10	7949,] [125985,122176,126055,124577,117949,119946] [122176,126055,124577,117949,119946,121560] [126055,124577,117949,119946,121560,108915] [124577,117949,119946,121560,108915,108150] [117949,119946,121560,108915,108150,112421,] [119946,121560,108915,108150,112421,109917] [121560,108915,108150,112421,109917,117723] [108915,108150,112421,109917,117723,117087] [108150,112421,109917,117723,117087,121735] [112421,109917,117723,117087,121735,115984] [109917,117723,117087,121735,115984,118432] [117723,117087,121735,115984,118432,114831] [117087,121735,115984,118432,114831,121519] [121735,115984,118432,114831,121519,121901] [115984,118432,114831,121519,121901,116494] [118432,114831,121519,121901,116494,111204] [114831,121519,121901,116494,111204,121572] [121519,121901,116494,111204,121572,113308] [121901,116494,111204,121572,113308,110401] [116494,111204,121572,113308,110401,112600] [111204,121572,113308,110401,112600,117018] [121572,113308,110401,112600,117018,112787] [113308,110401,112600,117018,112787,10
---	---

8843] [110401,112600,117018,112787,108843,11 0122] [112600,117018,112787,108843,110122,11 8344]

