

SKRIPSI
SISTEM DETEKSI *PLAGIARISM* PADA JUDUL
TUGAS AKHIR MENGGUNAKAN METODE *RABIN-*
***KARP* BERBASIS WEB**



Disusun Oleh:

Nama : Ahmad Subadri
NIM : 17121035

PROGRAM STUDI SISTEM INFORMASI
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS MERCU BUANA YOGYAKARTA
TAHUN 2022

**SISTEM DETEKSI *PLAGIARISM* PADA JUDUL
TUGAS AKHIR MENGGUNAKAN METODE
RABIN-KARP BERBASIS WEB**

Disusun Oleh:

Nama : Ahmad Subadri
NIM : 17121035

**Skripsi ini diajukan untuk memenuhi persyaratan akademik
sarjana pada Program Studi Sistem Informasi, Fakultas
Teknologi Informasi, Universitas Mercu Buana Yogyakarta.**

Yogyakarta, 13 April 2022
Menyetujui Pembimbing,

Irfan Pratama S.Kom., M.Eng., MCE.
NIDN 0529049102

HALAMAN PENGESAHAN

SISTEM DETEKSI *PLAGIARISM* PADA JUDUL TUGAS AKHIR MENGUNAKAN METODE *RABIN-KARP* BERBASIS WEB

Oleh:
Ahmad Subadri
17121035

Telah dipertanggungjawabkan dan diterima
oleh Tim Penguji pada tanggal
<tgl-bulan-tahun>

Mengetahui Dekan,

Dosen Pembimbing,

(Anief Fauzan Rozi S.Kom.,M.Eng.)
NIDN. 0522088601

(Irfan Pratama S.Kom., M.Eng., MCE.)
NIDN. 0529049102

Dosen Penguji,

1. <dosen penguji 1>
NIDN.

2. <dosen penguji 2>
NIDN.

3. <dosen penguji 3>
NIDN.

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS

Sebagai mahasiswa Universitas Mercu Buana Yogyakarta, saya
yangbertanda tangan di bawah ini:

Nama : Ahmad Subadri
NIM : 17121035
Program Studi : Sistem Informasi
Fakultas : Teknologi Informasi
Jenis Karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk
memberikan kepada Universitas Mercu Buana Yogyakarta Hak Bebas
Royalti Non-eksklusif (*None-exclusive Royalty Free Right*) atas karya
ilmiah saya yang berjudul:

SISTEM DETEKSI *PLAGIARISM* PADA JUDUL TUGAS AKHIR MENGUNAKANMETODE *RABIN-KARP* BERBASIS WEB

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas
Royalti Royalti Non-eksklusif ini Universitas Mercu Buana Yogyakarta
berhak menyimpan, mengalih-media-kan, mengelola dalam bentuk
pangkalan data (*database*), merawat dan mempublikasikan skripsi saya
selama tetap mencantumkan nama saya sebagai penulis dan sebagai pemilik
hak cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : YOGYAKARTA
Pada Tanggal : 12 April 2022

Yang menyatakan,

(Ahmad Subadri)

HALAMAN PERNYATAAN ORISINALITAS KARYA

**Skripsi ini adalah hasil karya saya sendiri,
dan semua sumber baik yang dikutip maupun yang dirujuk, telah saya
nyatakan dengan benar.**

Nama Lengkap : Ahmad Subadri
NIM : 17121035
Judul Skripsi : Deteksi *plagiarism* pada judul tugas akhir
menggunakan metode *Rabin-Karp* berbasis
web
Program Studi : Sistem Informasi
Tanggal : 13 April 2022
Tanda Tangan : <ttd>

HALAMAN PERSEMBAHAN

Skripsi ini saya dedikasikan kepada kedua orang tua tercinta, Bapak dan Ibu di rumah, ketulusanya dari hati atas doa yang tak pernah putus, semangat yang tak ternilai. Serta Untuk Orang-Orang Terdekat saya yang selalu suport dan tidak suport, Dan Untuk Almamater Biru Kebanggaan saya.

Dan tak lupa saya persembahkan juga untuk orang-orang yang selalu tanya.

"Kapan sidang ?"

"Kapan wisuda ?"

Bismillah ini saatnya !

HALAMAN MOTTO

Moto dalam hidup saya

“Jangan takut tidak ada jalan, dan jangan takut membuka jalan”

Jangan pernah jadi manusia Pesimistis

Karena kegagalan adalah sukses yang tertunda

Terus berjuang jika ingin menikmati hidup

KATA PENGANTAR

Segala puji dan syukur kehadiran Allah SWT atas berkah, rahmat dan hidayah-nya yang senantiasa dilimpahkan kepada kita semua, sehingga bisa menyelesaikan Skripsi dengan judul “Sistem Deteksi *Plagiarism* Pada Judul Tugas Akhir Menggunakan Metode *Rabin-Karp* Berbasis Web” Sebagai syarat untuk menyelesaikan Program Sarjana (S1) pada Program studi Sistem Informasi Fakultas Teknologi Informasi.

Dalam penyusunan skripsi ini banyak hambatan serta rintangan yang penulis hadapi namun pada akhirnya dapat melaluinya berkat adanya bimbingan dan bantuan dari berbagai pihak baik secara moral maupun spiritual. Maka dari itu pada kesempatan ini penulis menyampaikan ucapan terimakasih kepada:

1. Dr. Alimatus Sahrah, M.Si, M.M., sebagai Rektor Universitas Mercubuana Yogyakarta, atas bantuan dan kerjasamanya dalam hal urusan di dalam lingkup Universitas Mercu Buana Yogyakarta.
2. Anief Fauzan Rozi, S.Kom., M.Eng.,MCE., sebagai Dekan Fakultas Teknologi Informasi, atas bantuan dan kerjasamanya dalam hal urusan di dalam lingkup Fakultas Teknologi Informasi Universitas Mercu Buana Yogyakarta.
3. Irfan Pratama, S.kom., M.Eng., sebagai Kaprodi Informatika, atas bantuan dan kerjasamanya dalam hal urusan di dalam lingkup Program Studi Sistem Informasi.
4. Putri Taqwa Prasetyaningrum, S.T., M.T.,MCE sebagai Dosen Pembimbing Akademik dan Dosen Pembimbing Skripsi, atas bantuan dan kerjasamanya dalam hal penyusunan Skripsi.
5. Seluruh jajaran Dosen dan Staf Fakultas Teknologi Informasi Program Studi Sistem Informasi Universitas Mercu Buana Yogyakarta, atas bantuan dan kerjasamanya dalam hal perkuliahan.
6. Teman-teman Program Studi Sistem Informasi angkatan 2017, sebagai pendukung, atas bantuan dan kerjasamanya dalam hal proses pengerjaan skripsi

Penulis memohon maaf atas segala kesalahan yang pernah dilakukan.
Semoga skripsi ini dapat memberikan manfaat untuk mendorong penelitian-
penelitian berikutnya.

Yogyakarta, 13 April 2022
Penulis,

Ahmad Subadri

Deteksi *plagiarism* pada judul tugas akhir menggunakan metode *Rabin-karp* berbasis web

Ahmad Subadri, 17121035, 17121035@student.mercubuana-yogya.ac.id

Irfan Pratama S.Kom., M.Eng., MCE., 0529049102, irfanp@mercubuana-yogya.ac.id

ABSTRAK

Dengan adanya perkembangan teknologi informasi di era digital saat ini, dimana semakin mudahnya pertukaran informasi, tidak hanya membawa dampak positif bagi kemajuan teknologi, Penggunaan digital semakin memudahkan mahasiswa untuk melakukan plagiarisme yang mengakibatkan perguruan tinggi atau universitas mengurangi penggunaan kertas dalam proses belajar mengajar untuk mengurangi tingkat plagiarisme mahasiswa. Sehingga di perlukanya sebuah sistem untuk melakukan pengecekan plagiarisme pada judul tugas akhir mahasiswa dengan cepat dan tepat.

Metode yang dapat di gunakan salah satunya adalah algoritma Rabin-Karp. Algoritma Rabin-Karp memiliki keunggulan pencarian string dengan pola yang panjang. Algoritma dalam sistem ini memiliki langkah-langkah *text preprocessing* yang terdiri dari beberapa step diantaranya *case folding*, *filtering*, dan *tokenizing*.

Dari text preprocessing inilah yang akan di proses menggunakan algoritma *Rabin-Karp*, hasil dari metode ini adalah nilai kemiripan dari judul-judul mahasiswa yang di hitung menggunakan rumus *similarity* sebagai tahap akhir cek plagiarisme. Hasil yang di keluarkan adalah dalam bentuk persentase kemiripan dari judul-judul mahasiswa yang di ajukan dengan judul-judul terdahulu.

Hasil pengujian akurasi pada pendeteksian similaritas judul tugas akhir diperoleh hasil akurasi similaritas kemungkinan sebesar 70 %. Algoritma Rabin-Karp ini masih cukup layak untuk digunakan sebagai suatu metode sederhana untuk mendeteksi kemungkinan terjadinya plagiarisme. Tujuan akhir yang diharapkan adalah dengan adanya sistem ini kemungkinan akan dilakukannya praktek plagiarisme dapat dideteksi dan dicegah sedini

Kata kunci : *Plagiarism*, *Rabin-Karp*, *Preprocessing*, *String Matching*.

DAFTAR ISTILAH

Tuliskan dalam bentuk tabel dengan formaturut abjad menaik (A-Z).

No.	Istilah	Keterangan
1	ASCII	American Standard Code for Information Interchange atau Kode Standar Amerika untuk Pertukaran Informasi

DAFTAR ISI

SISTEM DETEKSI <i>PLAGIARISM</i> PADA JUDUL TUGAS AKHIR MENGUNAKAN METODE <i>RABIN-KARP</i> BERBASIS WEB.....	I
HALAMAN PENGESAHAN	II
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIK.....	III
HALAMAN PERNYATAAN ORISINALITAS KARYA	IV
HALAMAN PERSEMBAHAN.....	V
HALAMAN MOTTO	VI
KATA PENGANTAR.....	VII
ABSTRAK	IX
DAFTAR ISTILAH	X
DAFTAR ISI.....	XI
DAFTAR TABEL.....	XIII
DAFTAR GAMBAR	XIV
DAFTAR LAMPIRAN	XV
BAB I PENDAHULUAN.....	1
1.1 LATAR BELAKANG	1
1.2 PERUMUSAN MASALAH.....	2
1.3 TUJUAN PENELITIAN.....	2
1.4 MANFAAT PENELITIAN.....	3
BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI.....	4
2.1 TINJAUAN PUSTAKA.....	4
2.2 LANDASAN TEORI.....	8
2.2.1 <i>Plagiarisme</i>	8
2.2.2 <i>Algoritma Rabin Karp</i>	8
I. Preprocessing	10
II. Filtering	11
III. Synonym Recognition	12
IV. Hashing	12
BAB III METODOLOGI	14
3.1 TAHAPAN PENELITIAN.....	14
3.2 BAHAN PENELITIAN.....	15
3.3 ALAT PENELITIAN	15
3.4 JALAN PENELITIAN	15
3.4.1 <i>Metode Pengembangan Perangkat Lunak</i>	15

3.4.2	<i>Perancangan Proses</i>	17
3.4.3	<i>Skenario Eksperimen Proses Rabin-Karp</i>	18
BAB IV ANALISIS DAN PEMBAHASAN		21
4.1	ANALISA PENDETEKSI KESAMAAN DOKUMEN	21
4.2	ANALISA METODE RABIN KARP.....	21
4.3	PERANCANGAN SISTEM.....	22
4.4	TAHAPAN PROSES SISTEM DETEKSI KESAMAAN DENGAN METODE <i>RABIN-KARP</i> 23	
4.4.1	<i>Tahap input parameter</i>	24
4.4.2	<i>Tahapan preprocessing</i>	25
4.4.2.1	Case Folding	26
4.4.2.2	Filtering	27
4.4.2.3	Tokenizing	28
4.5	DESAIN SISTEM INPUT OUTPUT SISTEM MAHASISWA	35
4.5.1	<i>Design Frontend Interface user</i>	35
4.5.2	<i>Design form login user</i>	36
4.5.3	<i>Design dashboard user mahasiswa</i>	36
4.5.4	<i>Design form upload pengajuan judul dan proposal</i>	38
4.5.5	<i>Design tampilan jadwal dan pengumuman seminar proposal</i>	38
4.6	DESAIN SISTEM INPUT OUTPUT DOSEN KEPALA PROGRAM STUDI	39
4.6.1	<i>Design hasil uji Rabin Karp</i>	39
4.6.2	<i>Design dashboard</i>	40
4.6.3	<i>Design menu bimbingan</i>	40
4.6.4	<i>Design menu skripsi</i>	41
4.6.5	<i>Design menu pendadaran</i>	43
4.6.6	<i>Design menu yudisium</i>	46
4.6.7	<i>Design menu progres bimbingan</i>	47
4.6.8	<i>Design menu progres yudisium</i>	47
BAB V PENUTUP		48
5.1	KESIMPULAN.	48
5.2	SARAN.....	48
DAFTAR PUSTAKA		49
LAMPIRAN		52
	<i>Lampiran A Biodata Peneliti</i>	52
	<i>Lampiran C Artifak/Data Penelitian</i>	54
	<i>Lampiran D Kode Program</i>	55

DAFTAR TABEL

Tabel 1 Judul source uji plagiarism	24
Tabel 2 Judul penguji 1 dan penguji 2 sebagai pembanding	25
Tabel 3 Hasil Case Folding Rabin karp	26
Tabel 4 Hasil Filtering dari data case folding	27
Tabel 5 Hasil parshing k-gram dari data filtering	29
Tabel 6 Hasil hashing data dari data tokenizing	31
Tabel 7 Hasil hash yang sama antara source \cap B dan source \cap C	32

DAFTAR GAMBAR

Image 1 Rumus matematik	10
Image 2 Flowchart preprocessing	11
Image 3 Flowchart Synonym Recognition.....	12
Image 4 Tahapan penelitian	14
Image 5 Metode Pengembangan Perangkat Lunak.....	16
Image 6 Perancangan proses kerja plagiarism	17
Image 7 flowchart preprocessing plagiarism	18
Image 8 Skenario Eksperimen Proses Rabin-Karp	19
Image 9 Flowchart Analisa Metode Rabin Karp.....	22
Image 10 Flowchart perancangan sistem	22
Image 11 Tahapan proses sistem deteksi kesamaan dengan metode Rabin-Karp	23
Image 12 Flowchart Tahap input parameter	24
Image 13 Tahapan preprocessing.....	25
Image 14 Flowchart case folding	26
Image 15 Flowchart Filtering.....	27
Image 16 Flowchart Tokenizing	28
Image 17 flowchart dari proses parsing k-gram.....	30
Image 18 Flowchart proses hashing.....	33
Image 19 Flowchart proses deteksi kesamaan judul	35
Image 20 Design Frontend Interface user	36
Image 21 Design form login user.....	36
Image 22 Design dashboard user mahasiswa.....	37
Image 23 Design form upload pengajuan judul dan proposal.....	38
Image 24 Design tampilan jadwal dan pengumuman seminar proposal	38
Image 25 Design dashboard dosen.....	40
Image 26 Design menu bimbingan dosen	40
Image 27 Design menu skripsi submenu data pengajuan judul proposal	41
Image 28 Design sub menu skripsi Schedule dan pengumuman hasil sempro.....	42
Image 29 Design submenu skripsi Data mahasiswa seminar proposal	42
Image 30 Design submenu skripsi Ploting dosen pembimng	43
Image 31 Design submenu pendadaran Syarat pendadaran	43
Image 32 Design submenu pendadaran Mahasiswa pendadaran	44
Image 33 Detail hasil pendadaran	44
Image 34 Design submenu pendadaran Penentuan jadwal dan penguji.....	45
Image 35 Design submenu pendadaran Pelaksanaan pendadaran	45
Image 36 Design submenu yudisium Syarat yudisium	46
Image 37 Design submenu yudisium Mahasiswa yudisium	46
Image 38 Design menu progres bimbingan	47
Image 39 Design menu progres yudisium.....	47

DAFTAR LAMPIRAN

Lampiran A Biodata Peneliti

.....

xii

Lampiran B *Scan* Dokumen Bimbingan Skripsi

.....

xiii

Lampiran C Artifak/Data Penelitian

.....

xiv

Lampiran D Kode Program

.....

xvi

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi saat ini sangat pesat dan perannya sangat penting untuk mendukung aktivitas manusia agar dapat mengoptimalkan waktu dengan lebih baik. Salah satu contoh teknologi yang membantu aktivitas manusia adalah sistem informasi. Instansi yang bergerak dibidang pendidikan seperti universitas juga membutuhkan dukungan sistem informasi dalam peningkatan mutu pelayanan terhadap proses pendidikan.

Tugas akhir atau biasa disebut dengan skripsi merupakan istilah yang digunakan untuk menggambarkan suatu karya tulis ilmiah berupa tulisan hasil penelitian seorang sarjana yang disusun untuk memenuhi persyaratan memperoleh gelar sesuai dengan jenjangnya (Alamsyah, 2017).

Setiap tahunnya Universitas Mercu Buana Yogyakarta khususnya di semester akhir mahasiswa melaksanakan bimbingan tugas akhir, namun terdapat beberapa kendala dalam hal pengelolaan data administrasi tugas akhir yang saat ini masih terkomputerisasi secara sederhana, dimana data tugas akhir masih dikelola dalam format excel (*.xlsx), DVD, atau word (*.docx), karena pengolahan data yang belum tersistem dan terintegrasi dengan database serta belum adanya fitur deteksi plagiarisme judul tugas akhir yang di ajukan oleh mahasiswa, sehingga membuat dosen sulit menentukan atau mengetahui apakah judul dan laporan tersebut pernah diajukan atau digunakan sebelumnya, karena masih banyak terjadi mahasiswa yang menggunakan judul skripsi yang sudah pernah digunakan angkatan-angkatan yang sebelumnya (Anggadini, 2013).

Dengan adanya permasalahan tersebut, banyak terjadi kasus yang disebut plagiat judul tugas akhir atau penggunaan judul skripsi milik orang lain (*plagiarism*), kemudian beberapa mahasiswa yang tidak melaporkan progres pengerjaan tugas akhir atau tidak melakukan bimbingan secara rutin, maka dari itu diperlukan sebuah sistem informasi yang dapat menyelesaikan masalah tersebut.

Sistem ini diharapkan dapat menyelesaikan permasalahan yang terjadi pada proses pelaksanaan dan manajemen tugas akhir, mengurangi adanya plagiat (*plagiarism*) judul tugas akhir. Untuk itu berdasarkan uraian di atas, maka peneliti dalam hal ini mengambil judul “Sistem Deteksi *plagiarism* pada judul tugas akhir menggunakan metode *rabin-karp* berbasis web“ dengan harapan sistem yang dibuat dapat menyelesaikan masalah dengan tepat dan memberikan kemudahan pelaksanaan tugas akhir.

1.2 Perumusan Masalah.

Berdasarkan uraian latar belakang, maka perumusan masalah dalam penelitian ini adalah :

1. Bagaimana membuat sebuah sistem yang mampu mendeteksi *plagiarism* terhadap sebuah judul skripsi ?
2. Bagaimana menentukan sebuah skripsi termasuk dalam hasil *plagiarism* atau bukan ?
3. Bagaimana sistem yang diterapkan mengenali kata-kata yang dimiliki kesamaan makna ?
4. Bagaimana menerapkan algoritma *Rabin-karp* yang akan digunakan dalam pengembangan aplikasi deteksi plagiat ?

1.3 Tujuan Penelitian.

Penelitian ini bertujuan untuk membangun sebuah sistem informasi yang mampu mendeteksi plagiat (*plagiarism*) pada judul tugas akhir menggunakan metode *Rabin-karp* berbasis web yang dapat mengurangi terjadinya plagiat atau kesamaan judul skripsi, membantu dosen dalam memberikan keputusan dan mempermudah pelaksanaan tugas akhir, mulai dari proses pengajuan judul dan proposal skripsi hingga pelaksanaan ujian atau sidang skripsi, serta mempermudah proses bimbingan yang bisa dilakukan dimanapun dan kapanpun dengan adanya fitur live chat yang ada di dalam sistem tersebut.

1.4 Manfaat Penelitian.

Manfaat dari penelitian dan pembangunan sistem deteksi *plagiarism* pada judul tugas akhir menggunakan metode *Rabin-Karp* berbasis web ini, yakni memudahkan mahasiswa dalam menyelesaikan tugas akhirnya, memudahkan dosen untuk mengontrol mahasiswa bimbinganya dan memantau progres tugas akhir, serta memudahkan penyimpanan arsip skripsi secara digital dan aman.

Adapun manfaat penelitian ini secara khusus diantaranya adalah sebagai berikut :

1. Memudahkan mahasiswa mengajukan judul dan proposal tugas akhir.
2. Memudahkan dosen pembimbing dan kepala jurusan untuk mengontrol perkembangan progress pengerjaan tugas akhir mahasiswa bimbingannya.
3. Memudahkan Fakultas dalam melakukan arsip atau dokumentasi skripsi.
4. Mengurangi kemungkinan terjadinya pengulangan judul tugas akhir yang sama.
5. Memudahkan mahasiswa dalam mempersiapkan syarat-syarat dan berkas pendukung yang diperlukan dalam administrasi tugas akhir.
6. Memudahkan semua aktor yang berperan dalam sistem untuk melihat informasi jadwal seminar proposal, seminar hasil dan sidang tugas akhir.

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka.

Tinjauan pustaka memperlihatkan perbedaan penelitian yang dilakukan dengan penelitian-penelitian terdahulu, jurnal, artikel atau buku yang saya gunakan sebagai berikut :

Didalam penelitian yang pernah dilakukan oleh (Asvarizal Filcha, 2019) dengan judul **“Implementasi Algoritma Rabin-Karp untuk Pendeteksi Plagiarisme pada Dokumen Tugas Mahasiswa”** mengatakan bahwa faktor yang mempermudah mahasiswa melakukan plagiarisme adalah fasilitas pada komputer yang dapat melakukan menyalin dan mengubah teks antar dokumen. Dalam kenyataannya proses plagiarisme tidak hanya pada tugas essay saja tetapi pada kode program. Sebelum menentukan suatu tugas mahasiswa termasuk melakukan plagiarisme atau tidak termasuk melakukan plagiarisme, maka sangatlah penting untuk mengetahui apa definisi plagiat atau plagiarisme.

Saat ini yang sering terjadi di perguruan tinggi untuk melakukan pengecekan adanya tindakan plagiarisme pada dokumen tugas mahasiswa dilakukan secara manual dan membandingkan dengan dokumen tugas yang lain. Dengan begitu, pengecekan plagiarisme akan memakan waktu yang sangat lama. Metode yang dapat digunakan adalah metode pencocokan string yaitu salah satunya menggunakan Algoritma Rabin-Karp.

Plagiarisme berdasarkan persentase dibagi menjadi 3, 1) Plagiarisme ringan , jika tingkat kesamaannya dibawah 30%. 2) Plagiarisme sedang, jika tingkat kesamaannya antara 30% hingga 70%. 3) Plagiarisme berat, jika tingkat kesamaannya diatas 70%. Berdasarkan hasil dan pembahasan maka peneliti dapat mengambil kesimpulan diantaranya Algoritma Rabin-Karp berhasil diimplementasikan pada sistem pendeteksi plagiarisme dokumen tugas mahasiswa. Sistem ini berhasil menampilkan persentase kemiripan dokumen tugas antar mahasiswa. Hasil perhitungan akurasi dengan confusion matrix pada sistem pendeteksi plagiarisme dokumen tugas mahasiswa adalah 90% yang diperoleh dari

20 perbandingan dokumen tugas mahasiswa. (Asvarizal Filcha, 2019)

Penelitian yang dilakukan oleh (Imam Nawawi, 2019) yang berjudul **“Deteksi plagiarisme pada dokumen skripsi berdasarkan tingkat kesamaan dengan menggunakan metode Longest Common Subsequence”** menyatakan bahwa plagiarisme adalah tindakan menyalin, mengambil karangan atau pendapat orang lain tanpa adanya izin tertulis dan menjadikannya seolah-olah pendapatnya sendiri. Hal ini masih menjadi fenomena yang sering terjadi pada instansi akademik atau non akademik. Namun pada jurusan Teknologi Informasi Politeknik Negeri Malang belum ada aplikasi yang dapat digunakan untuk mendeteksi plagiarisme. Berdasarkan permasalahan diatas, maka dibuatlah aplikasi deteksi plagiarisme pada dokumen Tugas Akhir / Skripsi yang bernama Document plagiarism Detection (Doristec) dengan menggunakan metode Longest Common Subsequence (LCS) dengan membuat modifikasi untuk mencapai hasil yang sesuai dengan perancangan. Aplikasi ini dibuat dengan tujuan untuk mengetahui seberapa besar tingkat plagiarisme pada dokumen Tugas Akhir / Skripsi yang nantinya akan diketahui oleh mahasiswa dan panitia Laporan Akhir dan Skripsi. Berdasarkan penelitian yang telah dilakukan, Metode Longest Common Subsequence dapat digunakan untuk deteksi plagiarisme dengan perbandingan dua atau lebih dokumen pembanding, dapat menguji lebih dari satu kalimat dan lebih dari satu kandidat pembanding, serta hasil pengujian akurasi yang dilakukan mendapatkan hasil akurasi yang tinggi. Hal ini dapat menjadi alternatif bagi mahasiswa jurusan Teknologi Informasi Politeknik Negeri Malang dalam melakukan pengujian terhadap penelitiannya dan bagi panitia Laporan Akhir dan Skripsi dapat melakukan monitoring Laporan Akhir dan Skripsi.

Penelitian yang dilakukan oleh (Ibrahim, 2011) yang berjudul **“Pengembangan sistem informasi monitoring tugas akhir berbasis sms”** menjelaskan berdasarkan pengamatan dan survei di Fasilkom Unsri dalam hal pengajuan proposal tugas akhir mahasiswa, mahasiswa harus melalui beberapa prosedur. Proposal yang telah disetujui oleh dosen pembimbing akan diserahkan ke bagian administrasi jurusan dan selanjutnya diserahkan ke Ketua Jurusan untuk diperiksa. Setelah proposal selesai diperiksa dikembalikan ke administrasi beserta

status proposal apakah disetujui/direvisi/ditolak. Jika mahasiswa telah selesai menyelesaikan laporan tugas akhirnya, mahasiswa mendaftar ke administrasi jurusan untuk mengikuti ujian tugas akhir. Kegiatan administrasi jurusan tersebut memiliki beberapa kendala dan terlihat kurang efisien, dan tidak efektif. Hal ini terlihat seperti sistem informasi masih belum dapat melacak posisi proposal saat dalam proses pengajuan hingga selesai, sehingga menyebabkan mahasiswa harus sering ke kampus untuk menanyakan status proposalnya, dan untuk melihat jadwal ujian akhir. Selain itu administrasi harus berkoordinasi terlebih dahulu dengan Ketua Jurusan jika ada mahasiswa yang menanyakan status proposalnya. Akibatnya kinerja administrasi terlihat kurang efektif.

Penelitian yang dilakukan oleh (Satia Suhada, 2017) yang berjudul **“Implementasi algoritma Rabin-Karp dan stemming najief andriani untuk deteksi plagiarisme”** Plagiarisme merupakan pengambilan karya orang lain kemudian menjadikannya nampak seperti karya sendiri. Praktek plagiat sering terjadi di dunia akademis seperti pada dokumen proyek akhir. Untuk meminimalkan tindakan plagiarisme, diperlukan suatu sistem untuk menilai atau mengukur seberapa banyak kemiripan dalam sebuah document seperti tugas akhir dan makalah ilmiah beberapa algoritma telah digunakan dalam proses deteksi plagiarisme namun pada penelitian ini akan menerapkan . Algoritma Rabin Karp dalam mendeteksi plagiarisme karena algoritma ini terbukti efektif untuk membandingkan pattern-pattern yang ada pada sebuah esai dengan menggunakan fungsi hashing yang dapat menemukan bentuk-bentuk / pola dalam teks, untuk lebih meningkatkan keakuratan dalam proses penemuan pola pada sebuah teks kemudian digunakan algoritma Steeming Najief Andriani yang dapat menemukan kata-kata yang setara yang memiliki persamaan kata dasar yang sama, selanjutnya kedua algoritma yang digunakan akan diimplementasikan ke pada sebuah aplikasi web untuk pengujian document tugas akhir.

Menurut penelitian (Sulhan, 2015) yang berjudul **“Sistem Monitoring Tugas Akhir Berbasis User Generated Content Pada Program Studi Sistem Informasi Universitas Kanjuruhan Malang”** Pengerjaan Tugas Akhir diperlukan bimbingan dosen atau konseling agar tugas akhir yang dikerjakan sesuai

dengan ketentuan yang ada dan sesuai dengan kaidah-kaidahnya, Permasalahannya adalah tidak adanya waktu ataupun kesulitan mahasiswa dalam menemui dosen pembimbing untuk konsultasi tentang tugas akhir yang sedang dikerjakannya, yang dikarenakan kesibukan dari dosen itu sendiri ataupun ketidaksesuaian jadwal antara dosen dan mahasiswa. Konsep *User Generated Content* (UGC) dikenal dengan sifat interaksi dua arah yang lebih menghadirkan kolaborasi, partisipasi, hal ini yang dimungkinkan untuk dikembangkan suatu sistem yang dapat memberikan solusi yakni perlunya dikembangkan sistem manajemen monitoring Tugas Akhir berbasis website. Dalam penelitian ini telah berhasil dibuat sebuah Perangkat lunak Sistem manajemen monitoring tugas akhir online dengan menerapkan konsep UGC, dan Setelah dilakukan Uji Coba pada perangkat lunak tersebut bahwa semua fasilitas dapat berjalan dengan baik sesuai dengan yang diharapkan dan khususnya penerapan konsep UGC, sehingga pengguna dapat saling komunikasi dengan memberikan banyak masukan atau berbagi pengetahuan untuk kemudahan dan kualitas hasil akhir dari tugas akhir, dan diharapkan penyelesaian tugas akhir selesai tepat pada waktunya.

Penelitian yang telah dilakukan oleh (Alamsyah, 2017) yang berjudul **“Perbandingan algoritma winnowing dengan algoritma rabin karp untuk mendeteksi plagiarisme pada kemiripan teks judul skripsi”** mengatakan plagiarisme biasanya sering terjadi di dunia Pendidikan, baik itu dalam pengerjaan skripsi atau karya tulis ilmiah, dimana mahasiswa dituntut untuk menyelesaikan skripsinya sebagai syarat kelulusan dan mendapatkan gelar sarjana. Deadline untuk segera menyelesaikan karya tulis ilmiah inilah memicu mereka untuk mencari jalan pintas untuk melakukan copy-paste karya ilmiah atau skripsi orang lain sehingga terjadinya plagiarisme. Untuk membantu mengurangi tingkat plagiarisme telah banyak dibuat sistem pendeteksi plagiarisme untuk memeriksa tingkat kemiripan teks dokumen. Pada umumnya, sistem pendeteksi plagiarisme dikembangkan untuk data teks seperti essay, artikel, jurnal, penelitian dan dokumen teks yang lebih terstruktur seperti source code bahasa pemrograman. Untuk mendeteksi plagiarisme atau kemiripan teks dokumen dengan metode fingerprinting dapat dilakukan dengan perbandingan algoritma yang berkaitan pada

bidang text mining misal perbandingan pendekatan Algoritma Rabin Karp, Algoritma Manber dan Algoritma Winnowing.

2.2 Landasan Teori.

2.2.1 Plagiarisme

Plagiarisme berasal dari kata Latin, “plagiarius”, yang berarti “pencuri”. Plagiarisme didefinisikan sebagai tindakan atau praktik mengambil dan mengumpulkan atau menyampaikan pemikiran, tulisan atau hasil karya orang lain selayaknya hasil karya diri sendiri tanpa persetujuan dari pemilik hasil karya tersebut. Dengan kata lain, mempraktikkan plagiarisme berarti mencuri hasil karya atau kepemilikan intelektual orang lain. Plagiarisme adalah tindakan penyalahgunaan, pencurian/perampasan, penerbitan, pernyataan, atau menyatakan sebagai milik sendiri sebuah pikiran, ide, tulisan, atau ciptaan yang sebenarnya milik orang lain. Sedangkan menurut Kamus Besar Bahasa Indonesia (KBBI), Plagiat merupakan pengambilan karangan (pendapat, dan sebagainya) orang lain dan menjadikannya seolah-olah karangan (pendapat dan sebagainya) sendiri, misalnya menerbitkan karya tulis orang lain atas nama dirinya sendiri (Ngroho, 2011).

Plagiat dapat dianggap sebagai tindak pidana karena mencuri hak cipta orang lain. Pelaku plagiat disebut sebagai plagiator. Sistem pendeteksi plagiarisme dapat dikembangkan untuk :

1. Data teks seperti essay, artikel, jurnal, penelitian dan sebagainya.
 2. Dokumen teks yang lebih terstruktur seperti bahasa pemrograman.
- Berikut ini adalah hal-hal yang tergolong kedalam tindakan plagiarisme, antara lain :
3. Mengakui tulisan orang lain sebagai tulisan sendiri.
 4. Mengakui gagasan orang lain sebagai pemikiran sendiri.
 5. Mengakui temuan orang lain sebagai kepunyaan sendiri.
 6. Mengakui karya kelompok sebagai kepunyaan atau hasil sendiri.

2.2.2 Algoritma Rabin Karp

String matching atau pencocokan *string* adalah subjek yang penting

dalam kaitanya dengan *text-processing*. Penggunaan *string matching* mencakup pencarian pola dalam *DNA sequence*, *search engine* internet, menemukan halaman web yang relevan pada *query*, dapat pula dimanfaatkan untuk mendeteksi adanya *plagiarism* karyatulis. Termasuk dalam algoritma *string matching* diantaranya algoritma Naive, algoritma Rabin Karp, algoritma Finite Automation, dan algoritma Knuth Morris Pratt (T. H. Cormen, 2001).

Algoritma Rabin Karp ditemukan oleh Michael O. Rabin dan Richard M. Karp. Algoritma ini menggunakan metode *hash* dalam mencari suatu kata. Teori ini jarang digunakan untuk mencari katatunggal, namun cukup penting dan sangat efektif bila digunakan untuk pencarian jamak (Atmopawiro, 2006).

Prinsip kerja dari *algoritma Rabin Karp* ini memberikan atau mempresentasikan setiap karakter dalam bentuk desimal digit (*digitradix-d*) $\Sigma = \{0, 1, 2, 3, \dots, d\}$, dimana $d = |\Sigma|$. Sehingga didapat masukan *string k* berturut-turut sebagai perwakilan panjang *k* desimal. Karakter *string* 31415 sesuai dengan jumlah desimal 31,415. Kemudian pola *p* di-*hash* menjadi nilai desimal dan *string* direpresentasikan dengan penjumlahan *digit-digit* angka menggunakan aturan Horner's, misal (Elchison, 2012) :

$$\{ A, B, C, \dots, Z \} \rightarrow \{ 0, 1, 2, \dots, 26 \}$$

$$\bullet \text{ BAN} \rightarrow 1 + 0 + 13 = 14$$

$$\bullet \text{ CARD} \rightarrow 2 + 0 + 17 + 3 = 22$$

Untuk pola yang panjang dan teks yang besar, algoritma ini menggunakan operasi *mod*, setelah dikenai operasi *mod q*, nilainya akan menjadi lebih kecil dari *q*. Tetapi tidak semua nilai *hash* yang cocok berarti polanya cocok. Hal ini sering terjadi pada beberapa kasus, ini disebut *spurious hits*. Kemungkinan terjadinya diantaranya karena:

- Operasi *mod* terinterferensi oleh keunikan nilai *hash* (nilai *mod q* biasanya dipilih bilangan prima sehingga $10q$ hanya cocok dengan 1 kata komputer) $27 \bmod 13 = 1$

- Informasi hilang setelah penjumlahan

BAN $\rightarrow 1 + 0 + 13 = 14$ CAM $\rightarrow 2 + 0 + 12 = 14$

Rumus matematis:

$$t_{s+1} = (d(t_s - T[s+1]h) + T[s+m+1]) \bmod q$$

dimana :

t_s = nilai desimal dengan panjang m dari *substring* $T[s+1 \dots s+m]$, untuk $s = 0, 1, \dots, n-m$
 t_{s+1} = nilai desimal selanjutnya yang dihitung dari t_s

d = radix desimal (bilangan basis 10)

m = panjang pola

$h = d^{m-1}$

q = nilai modulo

n = panjang teks

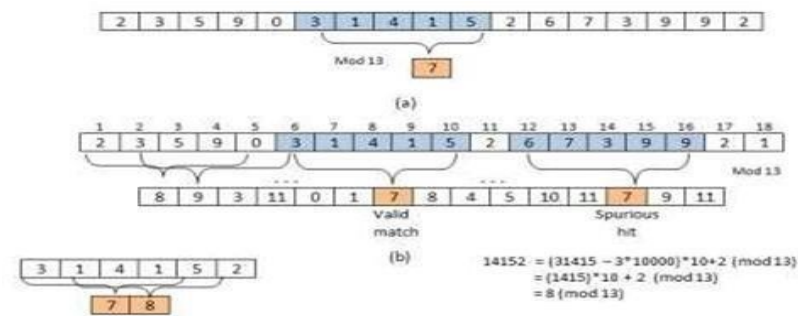


Image 1 Rumus matematik

I. Preprocessing



Tahap ini melakukan analisis semantik (kebenaran arti) dan sintaktik (kebenaran susunan) teks. Tujuan dari pemrosesan awal adalah untuk mempersiapkan teks menjadi data yang akan mengalami pengolahan lebih lanjut. Operasi yang dapat dilakukan pada tahap ini meliputi part – of – speech (PoS) tagging, menghasilkan parse tree untuk tiap – tiap kalimat, dan pembersihan teks (Alamsyah, 2017).

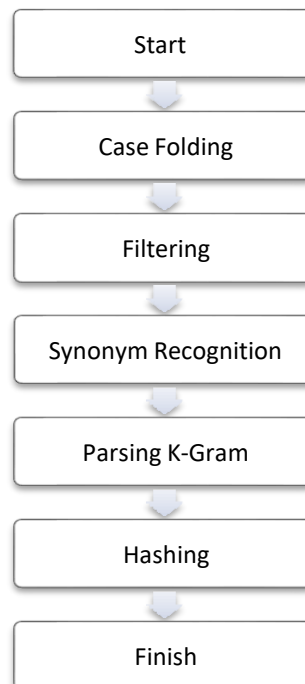


Image 2 Flowchart preprocessing

I. Case Folding

Case folding adalah merupakan tahapan yang mengubah semua huruf dalam dokumen atau isi konten yang terdapat pada suatu web menjadi huruf kecil. Hanya huruf „a“ sampai dengan „z“ yang diterima. Karakter selain huruf dihilangkan dan akan dilakukan proses penghapusan. Pada penelitian ini case folding yang diterima adalah huruf „a“ sampai dengan „z“, selain itu seperti tanda hubung atau tanda baca akan dihapus dan dihilangkan (Putra, 2013).

II. Filtering

Filetring adalah tahap mengambil kata – kata penting dari hasil token. Bisa menggunakan algoritma stoplist (membuang kata yang kurang penting) atau wordlist (menyimpan kata penting). Stoplist / stopword adalah kata-kata yang tidak deskriptif yang dapat dibuang dalam pendekatan bag-of-words. Contoh stopwords adalah “yang”, “dan”, “di”, “dari” dan seterusnya (Langgeni, 2010).

III. *Synonym Recognition*

Synonym Recognition atau pengenalan sinonim adalah pendeteksian plagiarisme melalui pendekatan sinonim. Dalam hal ini dokumen satu dibandingkan dengan dokumen lain dengan mendeteksi kata-kata yang mengandung sinonim sehingga tingkat kesamaan dapat dikatakan lebih akurat.

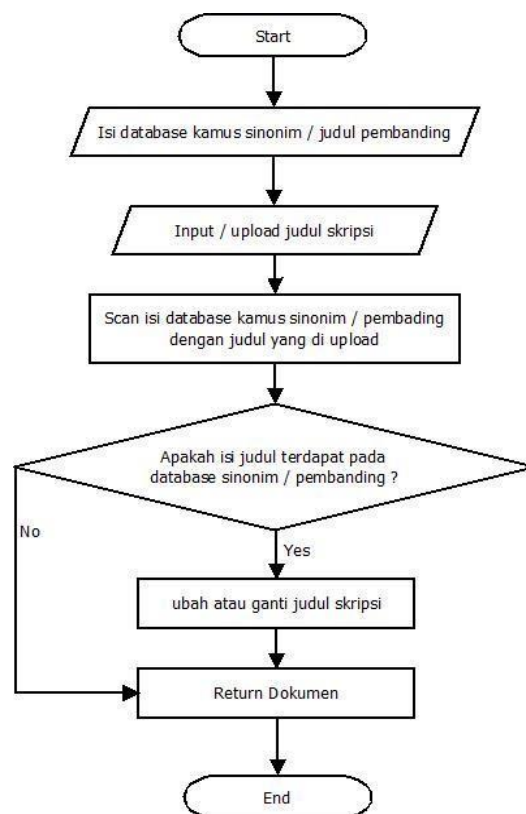


Image 3 Flowchart Synonym Recognition

Pada proses pencocokan sinonim, kaitannya antara isi dokumen dengan kamus sinonim yang terdapat pada database. Jika kata tidak tersedia pada kamus sinonim, maka tidak akan melakukan proses synonym recognition. Seluruh isi dokumen akan discan dan dicocokkan dengan kata yang ada dalam kamus, kemudian akan diubah berdasarkan isi kamus sinonim tersebut (Pratama, 2013).

IV. *Hashing*

Hashing adalah suatu cara untuk mentransformasikan

sebuah string menjadi suatu nilai yang unik dengan panjang tertentu (fixed-length) yang berfungsi sebagai penanda string tersebut. Hash function atau fungsi hash adalah suatu cara menciptakan fingerprint dari berbagai data masukan. Hash function akan mengganti atau akan mentransposekan data tersebut untuk menciptakan fingerprint yang biasa disebut hash value (Firdaus, 2008).

V. Parsing K-grams

K-grams adalah rangkaian terms dengan panjang K. Kebanyakan yang digunakan sebagai terms adalah kata. Kgram merupakan sebuah metode yang diaplikasikan untuk pembangkitan kata atau karakter. Metode K-grams ini digunakan untuk mengambil potonganpotongan karakter huruf sejumlah k dari sebuah kata yang secara kontinuitas dibaca dari teks sumber hingga akhir dari dokumen.

Berikut ini adalah contoh K-grams dengan $k=5$: Teks : “A dorun run run ,a do run run” kemudian dilakukan penghilangan spasi “adorunrunrunadorunrun” sehingga dihasilkan rangkaian 5-grams yang diturunkan dari teks : “adoru dorun orunr runru unrun nrunr runru unrunruna runad unado nador adoru dorun orunr runru unrun” (Schleimer, 2003).

BAB III METODOLOGI

Bab ini memaparkan tentang metodologi penelitian yang terdiri dari desain penelitian, alat dan bahan penelitian, serta metode penelitian yang digunakan.

3.1 Tahapan Penelitian

Metodologi penelitian menjelaskan bagaimana langkah-langkah atau tahapantahapan yang akan dilakukan dalam penelitian untuk dapat menjawab perumusan masalah penelitian sesuai konsep sistem temu balik informasi.

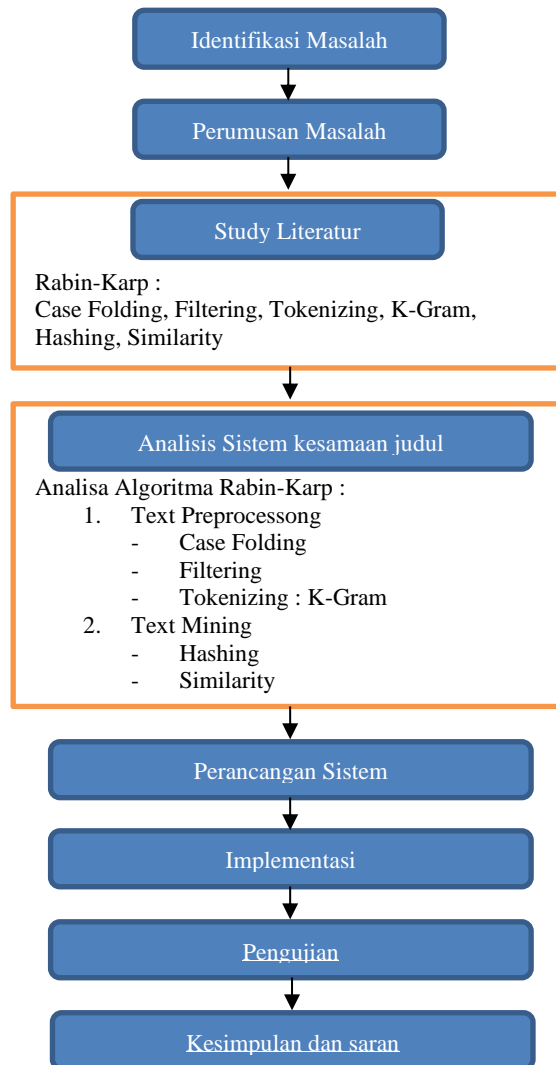


Image 4 Tahapan penelitian

3.2 Bahan Penelitian

Objek penelitian ini adalah Universitas Mercu Buana Yogyakarta. Data kriteria yang akan digunakan sebagai parameter penting untuk perhitungan dalam deteksi plagiarisme adalah judul skripsi dari beberapa tahun sebelumnya.

3.3 Alat Penelitian

Berikut ini adalah pembahasan lebih detail mengenai alat dan bahan penelitian yang digunakan pada penelitian ini.

- a. Perangkat keras yang digunakan berupa laptop dengan spesifikasi
 - *Processor* Intel(R) Celeron(R) CPU N3350 @ 1.10GHz
 - 1.10 GHz
 - Harddisk Drive 500 GB
 - SSD Drive 120 GB
- b. Perangkat Lunak yang digunakan untuk perancangan aplikasi berupa
 - Visual studio code 64-1.46.0
 - Sistem Operasi Windows 10 Pro 64 bit
 - PHP 7
 - XAMPP v3.2.4

3.4 Jalan Penelitian

3.4.1 Metode Pengembangan Perangkat Lunak

Metode pengembangan perangkat lunak ini dilakukan dengan metode waterfall. Model SDLC air terjun (waterfall) sering juga disebut model sekuensial linier (sequential linier). Peneliti menggunakan metode modern waterfall agar jika suatu saat ada kesalahan pada salah satu tahap, bisa dikembalikan ke tahap sebelumnya. Berikut pengertian dari tahap-tahap pada model waterfall.

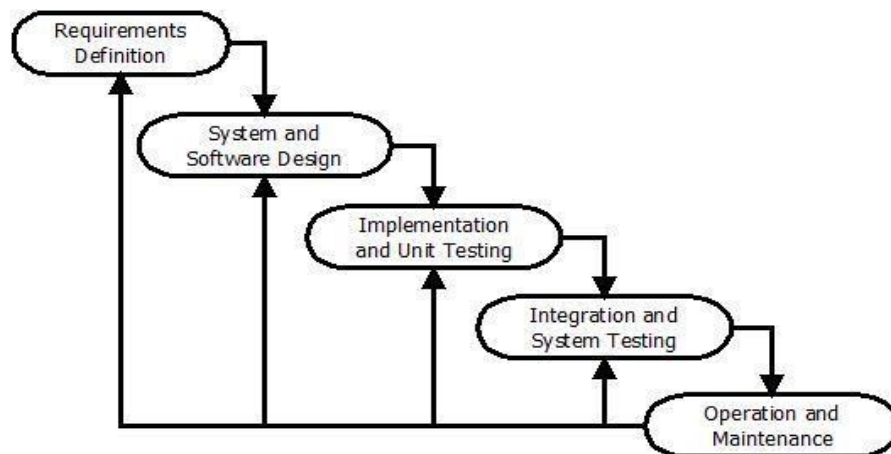


Image 5 Metode Pengembangan Perangkat Lunak

1. *Requirements Analysis and Definition (Analisis)*

Analisis adalah tahap menentukan aplikasi atau software seperti apakah yang akan dibuat. Analisis merupakan tahapan penetapan fitur, ruang lingkup, tujuan dan batasan yang dimiliki oleh perangkat lunak. Semua hal tersebut akan ditetapkan secara rinci dan berfungsi sebagai spesifikasi perangkat lunak. Dalam penelitian ini proses analisis yang dilakukan mencakup analisis terhadap data yang digunakan dan batasannya yaitu perangkat lunak hanya dapat memproses data berbentuk pdf dan keluaran perangkat lunak yaitu persentase kesamaan dokumen input dengan dokumen pada database.

2. *System and Software Design (Perancangan)*

Dalam tahapan ini dilakukan perancangan lebih detail yang dikembangkan dari hasil analisis. Pada penelitian ini, dilakukan perancangan antarmuka atau mock-up dari perangkat lunak. Pada tahapan ini juga dibahas gambaran umum mengenai alur dan proses perangkat lunak yang dibuat.

3. *Implementation and Unit Testing (Coding)*

Coding adalah tahap proses implementasi dari desain, dalam tahapan ini, hasil dari desain perangkat lunak akan direalisasikan sebagai satu set program atau unit program. Setiap unit akan diuji apakah sudah

memenuhi spesifikasinya.

4. *Integration and System Testing (Testing)*

Perangkat lunak yang dibuat akan memasuki tahap pengujian atau testing. Pengujian yang dilakukan meliputi pengujian fungsi-fungsi yang terdapat pada perangkat lunak dan juga pengujian terhadap data yang digunakan, yaitu data dokumen yang telah diplagiat.

5. *Operation and Maintenance (Pemeliharaan)*

Dalam tahapan ini, perangkat lunak mulai digunakan. Selain itu juga memperbaiki error yang tidak ditemukan pada tahap pembuatan. Dalam tahap ini juga dilakukan pengembangan perangkat lunak seperti penambahan fitur dan fungsi baru.

3.4.2 Perancangan Proses

Sistem deteksi plagiat secara umum dirancang untuk dapat mendeteksi kemiripan isi Judul skripsi, yang dimungkinkan kemiripan ini adalah hasil plagiat. Input sistem diperoleh dari judul yang di input oleh mahasiswa atau user. Judul yang di input otomatis akan tersimpan dalam database sistem.

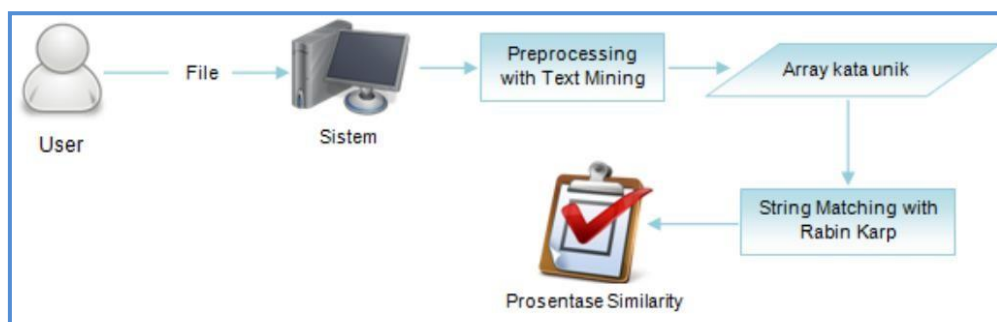


Image 6 Perancangan proses kerja plagiarism

Selanjutnya judul akan melewati tahap *preprocessing* menggunakan *text mining*. Tahap ini terdiri dari *tokenizing* (memecah kalimat menjadi potongan kata sekaligus mengubah isi judul menjadi huruf kecil/*lowercase*), *filtering* (membuang *stopword*/kata yang tidak deskriptif), dan *stemming* (mengembalikan setiap kata ke bentuk dasarnya).

Setelah itu, dokumen hasil *preprocess* akan melewati tahap *processing* menggunakan algoritma Rabin Karp. Tahap ini adalah tahap pencocokan dokumen/judul. Tahap ini terdiri dari *parshing K-gram* (memecah *string* ke dalam potongan *substring* sebanyak *k*), *hashing* (mengubah *substring k* ke dalam nilai *hash*), dan *string matching* (pencocokan *string* hasil *hashing*).

Hasil pencocokan *string* kemudian dihitung nilai *similarity* nya menggunakan *Dice's Similarity coefficient*, yang kemudian nilai tersebut diubah ke dalam bentuk prosentase *similarity*.

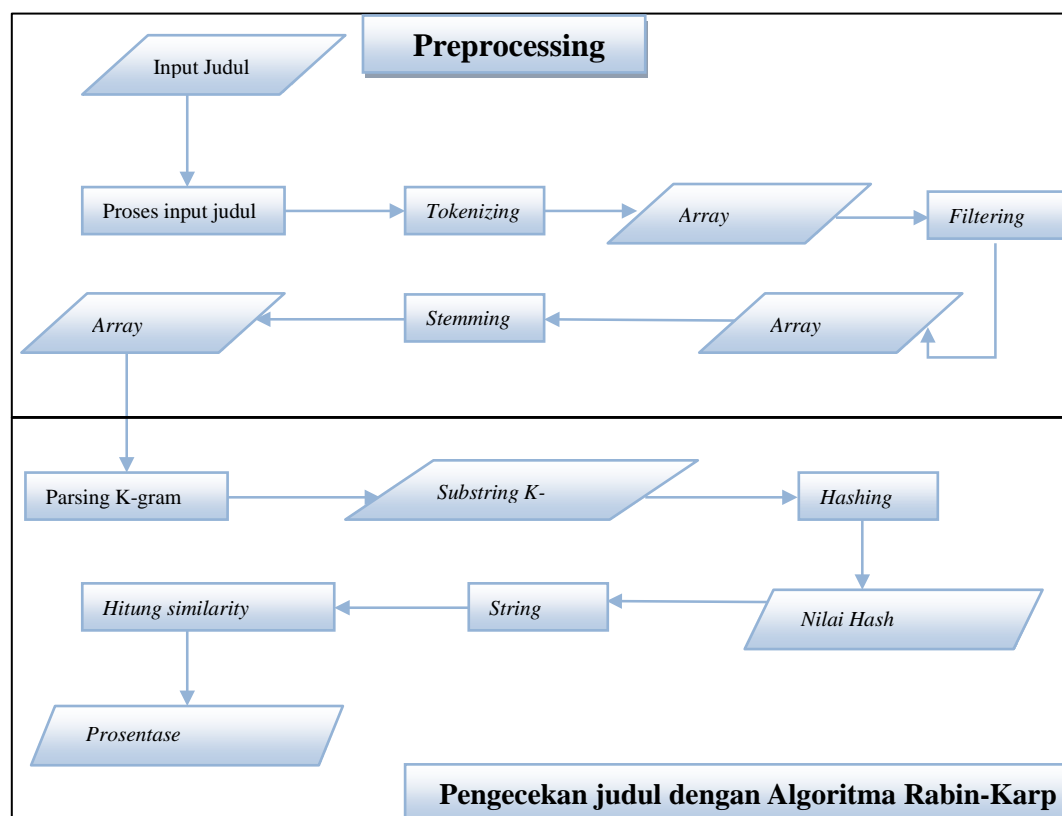


Image 7 flowchart preprocessing plagiarism

3.4.3 Skenario Eksperimen Proses *Rabin-Karp*

Eksperimen dilakukan dengan melakukan pendeteksian plagiarisme dalam beberapa skenario. Skenario eksperimen ditentukan dengan mengatur praproses yang digunakan dan besar K-Gram yang digunakan.

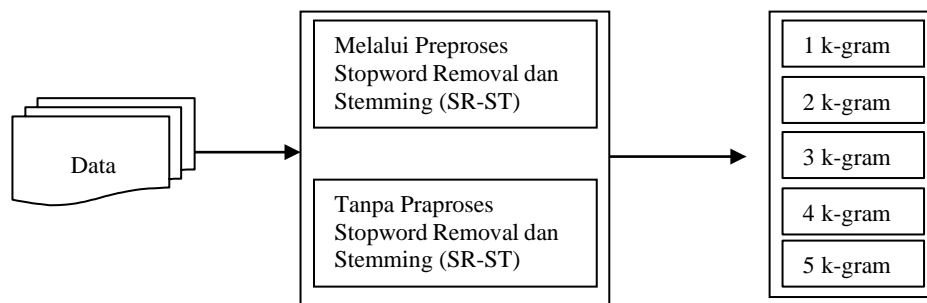


Image 8 Skenario Eksperimen Proses Rabin-Karp

3.4.3.1 Data persiapan judul uji

Dalam persiapan judul untuk pengujian *Plagiarism* terdapat 2 jenis :

- Source judul
Judul yang sudah di ajukan dan siap untuk di uji plagiarisme.
- Judul pembanding
Judul yang sudah disiapkan dalam database yang akan di gunakan untuk membanding source judul atau judul yang di ajukan.

3.4.3.2 Praproses

Sebelum memasuki proses deteksi plagiarisme dilakukan empat tahap praproses, yaitu case folding, tokenize, stopwords removal dan stemming. Untuk proses deteksi plagiarisme sebenarnya cukup dengan praproses case folding dan tokenize, sehingga praproses stopwords removal dan stemming hanya menjadi praproses tambahan agar mencapai hasil yang optimal. Pada eksperimen ini akan dibandingkan skenario eksperimen tanpa menggunakan praproses stopwords removal dan stemming (SR-ST) dan skenario eksperimen dengan menggunakan praproses stopwords removal dan stemming (SR-ST) untuk mengetahui pengaruh penambahan praproses stopwords removal dan stemming pada hasil eksperimen.

3.4.3.3 Analisa Algoritma Rabin Karp

Pada tahap ini dilakukan pengidentifikasian terhadap syarat-syarat algoritma dan langkah-langkah algoritma Rabin karp. Adapun langkah-

langkah algoritma Rabin Karp adalah:

a. Text Preprocessing. Tahapan text preprocessing meliputi:

- Case folding
- Filtering
- Tokenizing : k-gram.

b. Text Mining Tahapan text mining meliputi:

- Hashing
- Similarity

3.4.3.4 K-Gram

K-gram adalah rangkaian terms dengan panjang K. Kebanyakan yang digunakan sebagai terms adalah kata. K- gram merupakan sebuah metode yang diaplikasikan untuk pembangkitan kata atau karakter. Metode K-gram digunakan untuk mengambil potongan-potongan karakterhuruf sejumlah k dari sebuah kata yang secara kontinuitas dibaca dari teks sumber hingga akhir dari dokumen. Contoh penggunaan dari K-grams, trigram dari Menganalisis dan Menganalisa, yaitu “Men, eng, nga, gan, ana, nal, ali, lis, isi, sis”, dan “Men, eng, nga, gan, ana, nal, ali, lis, isa”. Sehingga dapat disimpulkan bahwa tigram dari kedua kata tersebut yang memiliki bentuk yang sama yaitu “Men, eng, nga, gan, ana, nal, ali, lis”.

3.4.3.5 Basis bilangan prima

Basis bilangan prima ini di gunakan sebagai bilangan pengali saat proses tokenizing, yang mana bilangan ini untuk menghitung nilai hashing sebagai.

BAB IV

ANALISIS DAN PEMBAHASAN

4.1 Analisa Pendeteksi Kesamaan Dokumen

Pada bab ini akan dibahas secara garis besar tentang perancangan sistem deteksi kesamaan judul tugas akhir dengan menggunakan algoritma Rabin-Karb. Pada umumnya algoritma pendeteksi kesamaan text atau dokumen atau plagiarisme memiliki tahapan yang sama dalam pemrosesan dokumen teks. Tahapan-tahapan tersebut terdiri dari input, proses, output. Berawal dari tahap input dokumen teks yang akan diuji kemudian text diproses melalui tahapan preprocessing (Case Folding, Tokenizing, Filtering) dan hashing kemudian hitungan similarity text. Setelah diproses maka selanjutnya aplikasi akan menghasilkan informasi persentase kemiripan text atau dokumen.

4.2 Analisa Metode Rabin Karp

Pada tahap ini dilakukan analisa terhadap algoritma rabin karp. Dengan menentukan nilai k-gram dan basis bilangan primanya. Hasil dari hashing asli dan hashing uji kemudian dibandingkan untuk dicari hashing yang sama. Jika hashing yang sama ditemukan, maka dari hasil hashing yang sama tersebut dihitung tingkat persentase kesamaannya (similarity).

Adapun langkah-langkah algoritma Rabin karp yang dilakukan dalam perancangan sistem ini adalah sebagai berikut :

1. Menghilangkan tanda baca.
2. Membagi teks kedalam bentuk k-gram, dimana nilai k merupakan nilai parameter yang dipilih oleh pengguna.
3. Menghitung nilai hash dari setiap k-gram.
4. Memilih nilai hash yang sama.
5. Menghitung nilai tingkat kesamaan judul

Untuk lebih jelasnya dapat dilihat pada flowchart berikut :

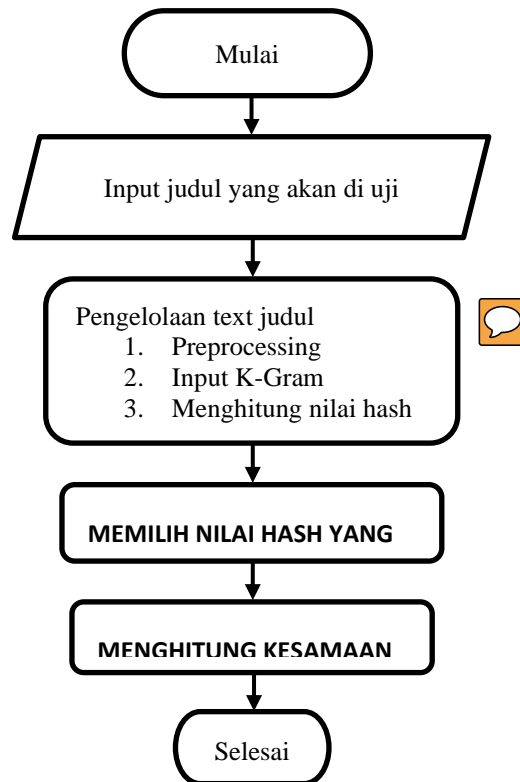


Image 9 Flowchart Analisa Metode Rabin Karp

4.3 Perancangan sistem

Secara garis besar sistem pendeteksi kesamaan pada judul skripsi dengan menggunakan metode Rabin-Karp akan dirancang dan kemudian akan dibuat sistemnya seperti yang tergambar pada skema alur kerja sistem berikut ini:

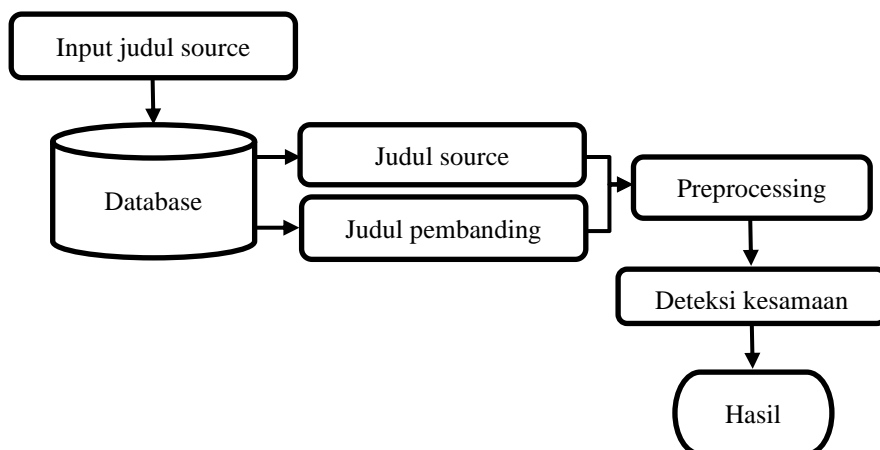


Image 10 Flowchart perancangan sistem

Pada sistem yang akan dibuat ini, judul source merupakan judul yang di ajukan atau di input untuk di uji dan judul pembanding adalah judul yang sudah ada di database yang akan di jadikan sebagai judul pembanding. Kemudian dilakukan preprocessing pada judul tersebut dengan menggunakan metode-metode penunjang sebagai tahapan yang harus dilakukan, seperti yang sudah diuraikan pada bab sebelumnya. Setelah dilakukan tahapan *preprocessing*, kemudian dilakukan pengujian dengan metode Rabin-Karp. Hasil akhir keluaran dari sistem ini adalah persentase kemiripan (*similarity*) dari metode Rabin-Karp. Dari situ dapat diketahui berapa persen tingkat kesamaan antara judul yang di ajukan dengan judul penguji.

4.4 Tahapan proses sistem deteksi kesamaan dengan metode *Rabin-Karp*

Pada perancangan sistem yang akan dibangun ini, penulis menjelaskan tahapantahapan perancangan sistem, baik dari tahapan preprocessing sampai hasil similarity. Pertama kali proses yang dilakukan oleh sistem adalah membaca judul yang sudah di inputkan, kemudian sistem akan masuk ke tahap *preprocessing*. . Pada tahap ini akan dilakukan beberapa proses, yaitu *Case folding* (mengubah semua huruf menjadi huruf kecil), *filtering* (penghilangan kata-kata yang kurang penting), kemudian proses *tokenizing* (pembentukan rangkaian k-gram).Kemudian dilakukan proses hashing dan untuk selanjutnya akan diperoleh hasil *similaritynya*.

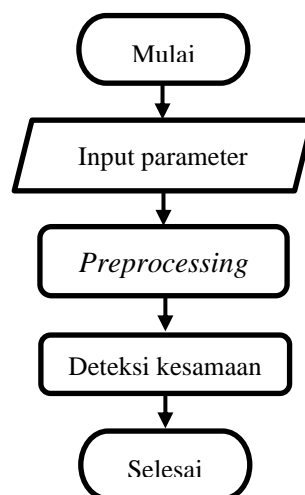


Image 11 Tahapan proses sistem deteksi kesamaan dengan metode *Rabin-Karp*

4.4.1 Tahap input parameter

Berikut adalah rancangan arsitektur sistem deteksi tingkat kesamaan dua dokumen dengan menggunakan metode Rabin Karp, seperti yang terlihat pada gambar di bawah ini.

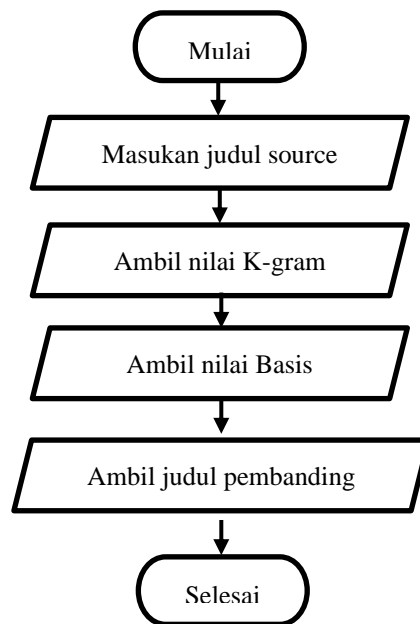


Image 12 Flowchart Tahap input parameter

Pada *flowchart* program tersebut, digambarkan bahwa proses sistem yang akan dibuat dimulai dari memasukkan judul source atau judul asli dan judul uji. Kemudian akan dilakukan pengujian melalui tahapan-tahapan *preprocessing*, yaitu menghilangkan tanda baca, merubah huruf kehuruf kecil, menghilangkan tanda baca, menghilangkan kata kata yang kurang penting, kemudian pembentukan rangkaian *gran* panjang *k*, lalu menghitung nilai *hashing*. Dari pengujian tersebut akan didapatkan hasil akhirnya berupa persentase kemiripan (*similarity*).

Berikut ini penulis memberikan contoh perhitungan kesamaan judul yang akan di proses dengan menggunakan metode Rabin-Karp.

Judul asli atau judul yang di ajukan :

Tabel 1 Judul source uji plagiarism

ID	Judul
J1	Sistem Pakar Diagnosa Penyakit Kulit Pada Manusia Menggunakan Metode Naive Bayes Berbasis Web di RS Depok 2 Yogyakarta

Judul pembanding, disini penulis hanya memberikan contoh 2 judul penguji :

Tabel 2 Judul penguji 1 dan penguji 2 sebagai pembanding

ID	Judul
J1	Sistem Manajemen Inventori Komputer Menggunakan Near Field Commnication Berbasis Android Studi Kasus di STIE Pelita Indonesia Pekanbaru.
J2	Sistem Informasi Registrasi Jenazah Pada Instalasi Forensik Rumah Sakit Abdul Wahab Sjahranie Samarinda Berbasis Web

Nilai K-Gram = 5

Nilai Basis = 3

4.4.2 Tahapan *preprocessing*

Tahap *Preprocessing* harus dilalui untuk menentukan *keyword* pada judul source dan judul. Pada tahap *preprocessing* terdapat beberapa proses yang dilakukan oleh sistem terhadap dokumen yang diinputkan. Proses-proses tersebut antara lain *case folding*, *filtering*, *tokenizing*. Berikut ini adalah *flowchart* dari *preprocessing* seperti yang terlihat pada gambar dibawah ini.

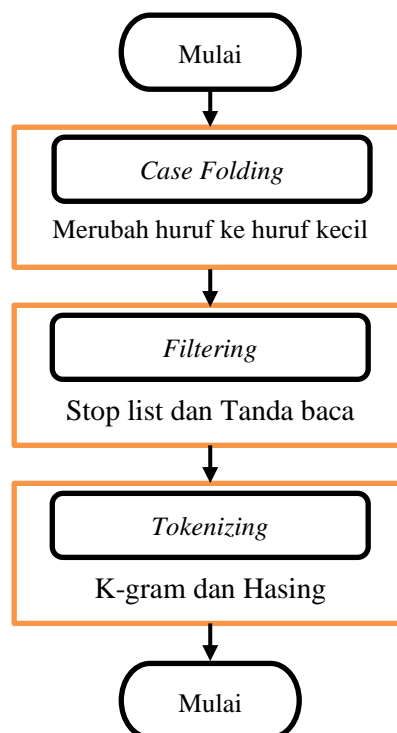


Image 13 Tahapan preprocessing

4.4.2.1 Case Folding

Proses *case folding* (mengubah huruf kapital menjadi huruf kecil) merupakan tahap pertama yang akan dilakukan dari rangkaian tahapan yang terdapat pada *preprocessing*. Berikut adalah contoh hasil proses *case folding* dari judul yang ada di tabel di atas :

Tabel 3 Hasil Case Folding Rabin karp

Judul Source	Pembanding 1	Pembanding 2
sistem pakar diagnosa penyakit kulit pada manusia menggunakan metode naive bayes berbasis web di rs depok 2 yogyakarta	sistem manajemen inventori komputer menggunakan near field commnication berbasis android studi kasus di stie pelita indonesia pekanbaru.	sistem informasi registrasi jenazah pada instalasi forensik rumah sakit abdul wahab sjahrane samarinda berbasis web

Pada contoh proses *case folding* diatas, kalimat yang terdapat huruf kapitalnya adalah pada kata “Sistem” dan pada kata “Manajemen”. Dimana pada kedua kata tersebut setelah dilakukan proses *case folding* akan diubah menjadi huruf kecil semua menjadi “sistem” dan “manajemen”.

Berikut ini adalah proses *case folding* yang digambarkan dalam bentuk *flowchart*, seperti yang terlihat pada gambar dibawah ini.

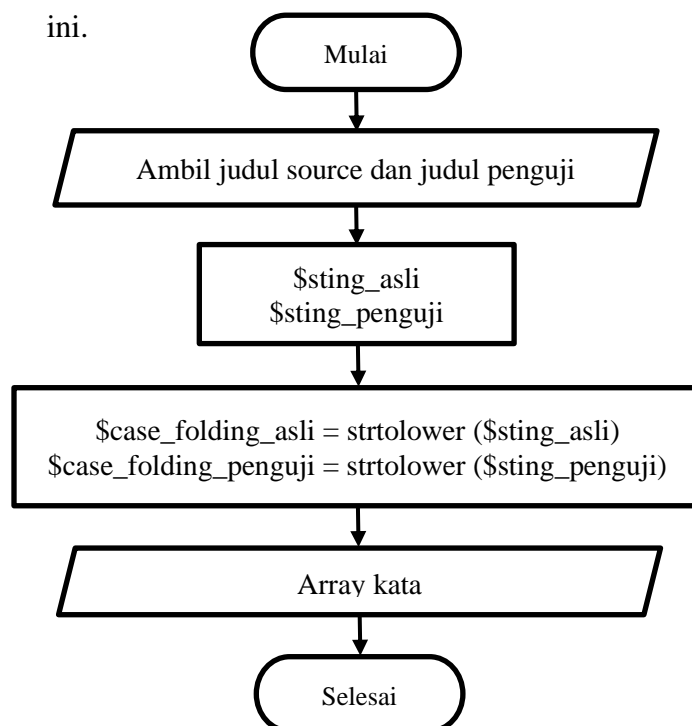


Image 14 Flowchart case folding

4.4.2.2 Filtering

Proses *filtering* dilakukan untuk mengambil kata-kata penting dari hasil *case folding*. Bisa menggunakan algoritma *stop list* (membuang kata yang kurang penting) atau *word list* (menyimpan kata penting). Disini penulis menggunakan algoritma *stop list*. Setelah proses *case folding*, judul selanjutnya masuk ke tahap *filtering*, yaitu proses membuang kata yang tidak penting, spasi, dan membuang tanda baca. Berikut ini contoh hasil proses *filtering*:

Tabel 4 Hasil Filtering dari data case folding

Judul source	Pembanding 1	Pembanding 2
sistempakardiagnosapenyakitkulitpadamanusiamengunakanmetodenaiwebesberbasiswebdirsepok2yogyakarta	sistemmanajemeninventorikomputermenggunakannearfieldcommunicationberbasisandroidstudikasusdistiepelitaindonesi apekanbaru	sisteminformasiregistrasijenazahpadainstalasiforensikrumahsaktabdulwahabsjhraniesamarindaberbasisweb

Pada contoh *filtering* diatas yaitu proses menghilangkan tanda baca dan menghilangkan stop word. Pada contoh judul diatas tanda baca yang di hilangkan adalah spasi, titik (.), koma (,) dan tanda petik (“ ”). Proses *filtering* tersebut seperti yang digambarkan dalam sebuah *flowchart* pada gambar dibawah ini.

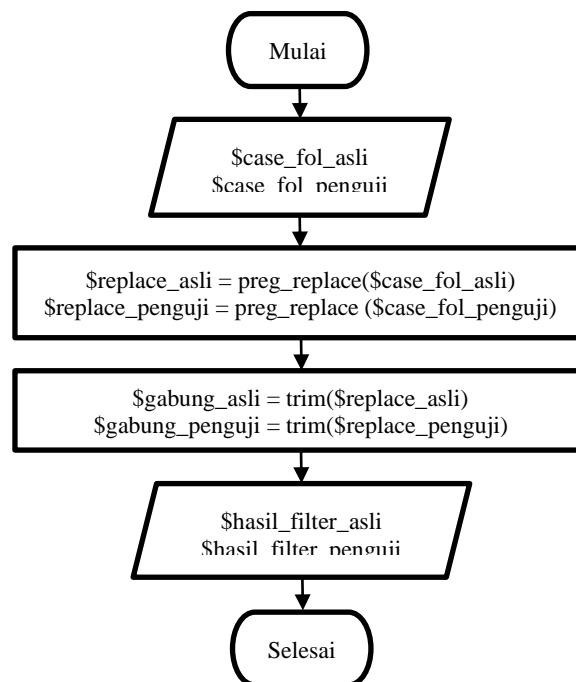


Image 15 Flowchart Filtering

4.4.2.3 Tokenizing

Setelah dilakukan proses *filtering*, selanjutnya adalah proses *tokenizing*, proses ini merupakan proses pembentukan pola kata, dimana pola katanya dalam bentuk gram dengan panjang k . Pada proses *tokenizing* di bagi menja dua sub proses yaitu proses *parsing* k -gram dan proses *hashing*.

Berikut ini adalah proses *tokenizing* yang digambarkan dalam bentuk *flowchart*, seperti yang terlihat pada gambar dibawah ini.

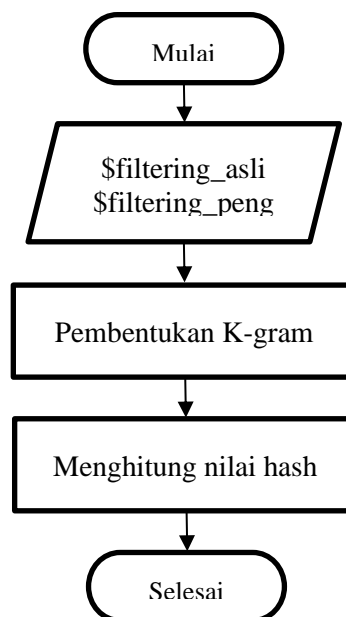


Image 16 Flowchart Tokenizing

Didalam proses *Tokenizing* terdapat proses yang di jalan yaitu *Parsing K-gram* dan *hashing*.

1. *Parsing K-gram*

Langkah selanjutnya dalam proses *tokenizing* adalah *parsing k-gram*, dimana pada proses ini kata dipecah menjadi potongan-potongan dimana setiap potongan mengandung karakter sebanyak k . Berikut ini adalah hasil contoh proses *parsing k-gram = 5* dari judul diatas:

Tabel 5 Hasil parshing k-gram dari data filtering

Judul source	Pembanding 1	Pembanding 2
{siste}{istem}{stemp}{tem pa}{empak}{mpaka}{paka r}{akard}{kardi}{ardia}{rdi ag}{diagn}{iagno}{agnos}{ gnos}{nosap}{osape}{sa pen}{apeny}{penya}{enya k}{nyaki}{yakit}{akitk}{kitk u}{itkul}{tkuli}{kulit}{ulitp}{li tpa}{itpad}{tpada}{padam} {adama}{daman}{amanu}{ manus}{anusi}{nusia}{usi am}{siame}{iamen}{amen g}{mengg}{enggu}{nggun} {gguna}{gunak}{unaka}{n akan}{akanm}{kanme}{an met}{nmeto}{metod}{etod e}{toden}{odena}{denai}{e naiv}{naive}{aiveb}{iveba {vebay}{ebaye}{bayes}{ay esb}{yesbe}{esber}{sberb {berba}{erbas}{rbasi}{bas is}{asisw}{siswe}{isweb}{s webd}{webdi}{ebdir}{bdirs {dirsd}{irsde}{rsdep}{sde po}{depok}{epok2}{pok2y {ok2yo}{k2yog}{2yogy}{y ogya}{ogyak}{gyaka}{yak ar}{akart}{karta}	{siste}{istem}{stemm}{tem ma}{emman}{mmana}{ma nai}{anaje}{najem}{ajeme {jemen}{emeni}{menin}{e ninv}{ninve}{inven}{nvent {vento}{entor}{ntori}{torik {oriko}{rikom}{ikomp}{kom pu}{omput}{mpute}{puter {uterm}{terme}{ermen}{rm eng}{mengg}{enggu}{ngg un}{gguna}{gunak}{unaka {nakan}{akann}{kanne}{a nnea}{nnear}{nearf}{earfi {arfie}{rfiel}{field}{ieldc}{el dco}{ldcom}{dcomm}{com mn}{omni}{mmnic}{mnic a}{nicat}{icati}{catio}{ation {tionb}{ionbe}{onber}{nbe rb}{berba}{erbas}{rbasi}{b asis}{asisa}{sisan}{isand}{ sandr}{andro}{ndroi}{droid {roids}{oidst}{idstu}{dstud {studi}{tudik}{udika}{dika s}{ikasu}{kasus}{asusd}{s usdi}{usdis}{sdist}{disti}{is tie}{stiep}{tiepe}{iepel}{ep eli}{pelit}{elita}{litai}{itain}{ taint}{aindo}{indon}{ndon e}{dones}{onesi}{nesia}{e siap}{siape}{iapek}{apeka {pekan}{ekanb}{kanba}{a nbar}{nbaru}	{siste}{istem}{stemi}{temi n}{eminf}{minfo}{infor}{nfo rm}{forma}{ormas}{rmasi} {masir}{asire}{sireg}{iregi} {regis}{egist}{gistr}{istra} stras}{trasi}{rasij}{asije}{si jen}{ijena}{jenaz}{enaza}{ nazah}{azahp}{zahpa}{ah pad}{hpada}{padai}{adain {dains}{ainst}{insta}{nstal {stala}{talas}{alasi}{lasif}{ asifo}{sifor}{ifore}{foren}{o rens}{rensi}{ensik}{nsikr}{ sikru}{ikrum}{kruma}{rum ah}{umahs}{mahsa}{ahsa k}{hsaki}{sakit}{akita}{kita b}{itabd}{tabdu}{abdul}{bd ulw}{dulwa}{ulwah}{lwaha {wahab}{ahabs}{habsj}{a bsja}{bsjah}{sjahr}{jahra}{ ahran}{hrani}{ranie}{anies {niesa}{iesam}{esama}{s amar}{amari}{marin}{arind {rinda}{indab}{ndabe}{da ber}{aberb}{berba}{erbas {rbasi}{basis}{asisw}{sisw e}{isweb}

Pada contoh *parsing k-gram* diatas adalah proses pembentukan pola kata dalam bentuk gram dengan panjang karakter k=5, sehinga isi kalimat pada judul asli dan judul uji dirubah dalam bentuk kgram. Berikut ini adalah *flowchart* dari proses *parsing k-gram*. Proses *parsing k-gram* digambarkan pada sebuah *flowchart* berikut ini :

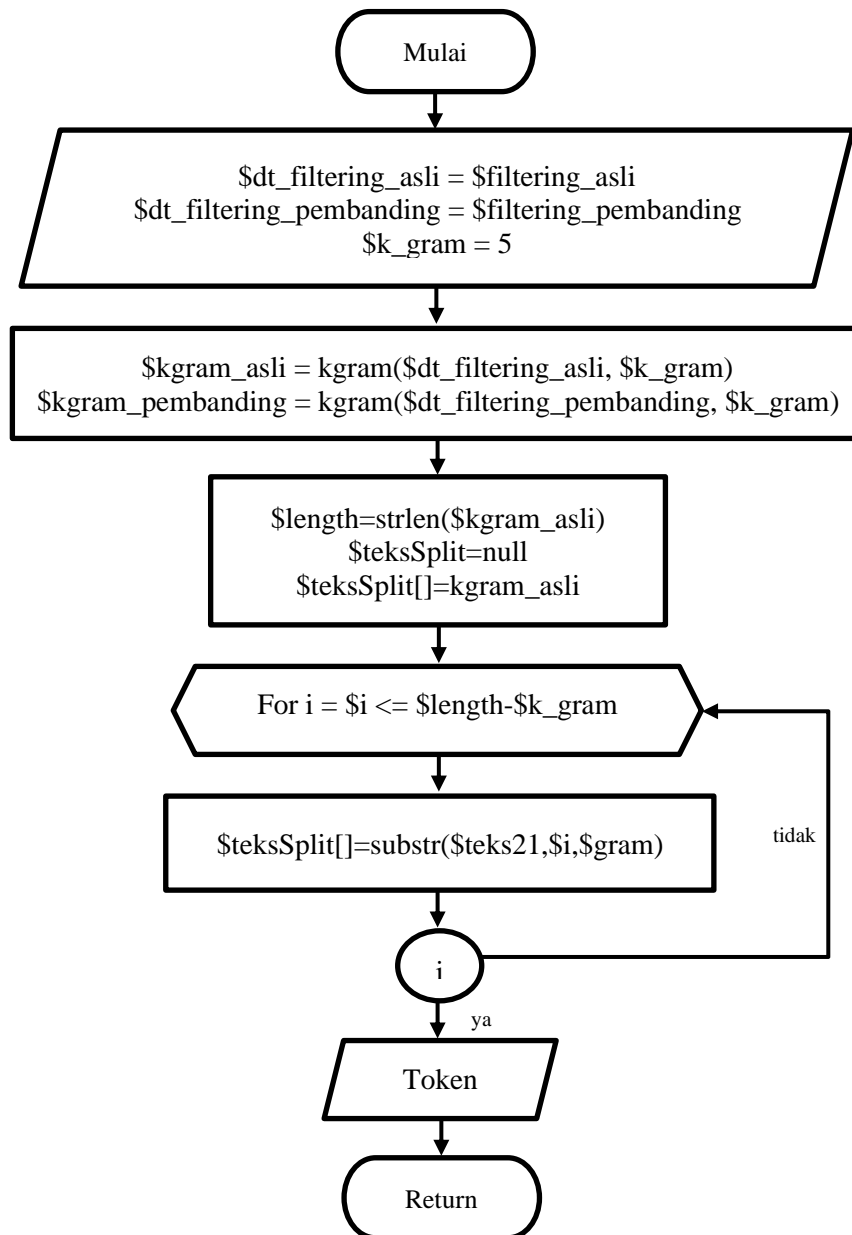


Image 17 flowchart dari proses parsing k-gram

2. Proses Hashing

Kemudian dilakukan proses *hashing*, dimana pada proses ini hash berfungsi untuk mengkonvert setiap string menjadi bilangan. Dengan cara mengalikan nilai ASCII hasil huruf hasil *k-gram* dengan basis bilangan tertentu, dimana basis bilang merupakan bilangan prima. Metode *Rabin-Karp* didasarkan pada fakta jika dua buah *string* sama maka harga hash valuenya pasti sama. Sebagai contoh kita ambil kata dari hasil *k-gram* yang

pertama pada dokumen asli, yaitu kata { siste }. Contoh proses *hashing* untuk menghitung nilai hash dari kata { siste},

dengan nilai $k=5$ dan $b=3$.

Nilai ASCII dari kata { siste}

ASCII (s) = 115

ASCII (i) = 105

ASCII (s) = 115

ASCII (t) = 116

ASCII (e) = 101

$$H(C_1 \dots C_K) = C_1.b^{(K-1)} + C_2.b^{(K-2)} + \dots + C_{(k-1)}.b^{(k)} + C_k$$

$$H = (115.3^4) + (105.3^3) + (115.3^2) + (116.3^1) + (101.3^0)$$

$$H = (115.81) + (105.27) + (115.9) + (116.3) + (101.1)$$

$$H = (9315) + (2835) + (1035) + (348) + (101)$$

$$H = \mathbf{13634}$$

Jadi nilai hash pada judul asli *k-gram* yang pertama adalah 13634, proses perhitungan nilai hash di ulang kembali hingga *k-gram* keseluruhan dihitung. Berikut ini adalah nilai hasil hash dari judul asli dan hash judul uji yang sudah dilakukan *Group by* yang mana fungsinya adalah untuk mengelompokkan data dalam sebuah kolom yang ditunjuk. Fungsi ini akan menghasilkan kelompok data dengan menghilangkan data yang sama dalam satu tabel. Maka apabila dalam satu kolom terdapat beberapa data yang sama maka data yang akan ditampilkan hanya salah satu:

Tabel 6 Hasil hashing data dari data tokenizing

Judul Source	Pembanding 1	Pembanding 2
{13634}{13066}{13795}{13537}{12530}{13144}{13059}{12061}{12717}{12247}{13273}{12227}{12492}{12076}{12754}{13345}{13406}{13355}{12241}{13249}{12638}{13476}{13814}{12146}{12984}{13059}{13767}{13229}{13798}{13060}{13036}{13690}{12991}{11854}{12101}{12120}{12904}{12330}{13516}{13	{13634}{13066}{13792}{13528}{12506}{13072}{12835}{12119}{12895}{12056}{12707}{12468}{12971}{12544}{13190}{12950}{13451}{13734}{12642}{13488}{13841}{13446}{13474}{12832}{13098}{13409}{13355}{13692}{13969}{13577}{12653}{13519}{12958}{12504}{13079}{12604}{12890}{13738}{12893}{12	{13634}{13066}{13788}{13529}{12501}{13071}{12840}{13114}{12709}{13456}{13500}{12912}{12350}{13582}{12906}{13318}{12368}{12675}{13093}{13879}{13797}{13309}{12326}{13517}{12703}{12716}{12487}{13022}{12448}{13870}{12064}{12718}{12987}{11855}{12109}{12143}{12955}{13458}{13741}{13

927}{13451}{12518}{1214 2}{12958}{12504}{13079} {12604}{12890}{13738}{1 2893}{12058}{12704}{122 27}{13221}{13033}{12713 }{13706}{13027}{12213}{ 12457}{12929}{12155}{12 991}{13579}{12164}{1206 4}{12476}{13958}{12585} {13310}{12082}{12547}{1 3203}{12022}{12371}{136 43}{13082}{13831}{13653 }{12156}{12040}{12406}{ 13019}{13654}{13371}{12 275}{12575}{13303}{1280 4}{11542}{8746}{14185}{ 13259}{12901}{13788}{12 077}{12757} Total source = 97	059}{12707}{12217}{1319 4}{12954}{12237}{12269} {13344}{12430}{12603}{1 2405}{12781}{12208}{124 34}{13350}{13176}{13138 }{13043}{12504}{12108}{ 12377}{13658}{12887}{13 260}{12905}{12082}{1254 7}{13203}{12022}{12349} {13586}{12913}{13338}{1 2180}{13074}{12592}{135 91}{13187}{12705}{12700 }{13905}{13877}{13540}{ 12304}{12729}{12787}{12 460}{13914}{13912}{1342 1}{12423}{13070}{13807} {13577}{12651}{12543}{1 3202}{12487}{13023}{129 35}{13390}{12093}{12818 }{13040}{12505}{13320}{ 13084}{12634}{13460}{12 542}{12208}{13163}{1237 1}{12667}{12114}{12888} Total P1 = 115	393}{12096}{12819}{1232 4}{13515}{12701}{12698} {13423}{13401}{12608}{1 3395}{13572}{12880}{132 22}{13769}{13720}{12826 }{12098}{12828}{13328}{ 12136}{12935}{12904}{13 314}{11862}{12134}{1268 5}{13859}{13243}{13583} {11947}{12376}{11953}{1 2392}{13476}{12580}{120 92}{12810}{13259}{12190 }{13096}{12667}{12583}{ 13320}{12120}{12899}{12 310}{13456}{12764}{1287 8}{12018}{11852}{12082} {12547}{13203}{12022}{1 2371}{13643}{13082} Total P2 = 98
--	--	--

Setelah di dapatkan hasil *hashing* dan *group by* sistem akan menjumlah total hasil hashing yang akan di gunakan untuk menghitung *similarity*, hasil d tunjukan di bawah ini :

Jumlah hash pada judul asli adalah= $\sum \text{source} = 97$

Jumlah hash pada judul asli adalah= $\sum \text{P1} = 115$

Jumlah hash pada judul asli adalah= $\sum \text{P2} = 98$

Hash yang sama antara **source** \cap **B** dan **source** \cap **C** :

Tabel 7 Hasil hash yang sama antara **source** \cap **B** dan **source** \cap **C**

source \cap B	source \cap C
{12022}{12082}{12504}{12547}{12604}{12890}{13066}{12958}{12893}{13079}{13203}{13634}{13738}	{12371}{12022}{12082}{12547}{13066}{13082}{13203}{13634}{13643}

Hasil kesamaan diatas di dapat dari perbandingan antara **source**:B dan **source**:C dengan fungsi *query inner join* sehingga di dapat hasil

$\sum \text{source} \cap \text{B} = 13$ dan $\sum \text{source} \cap \text{C} = 9$

Gambaran proses *hashing* seperti yang terlihat pada *flowchart* berikut ini.

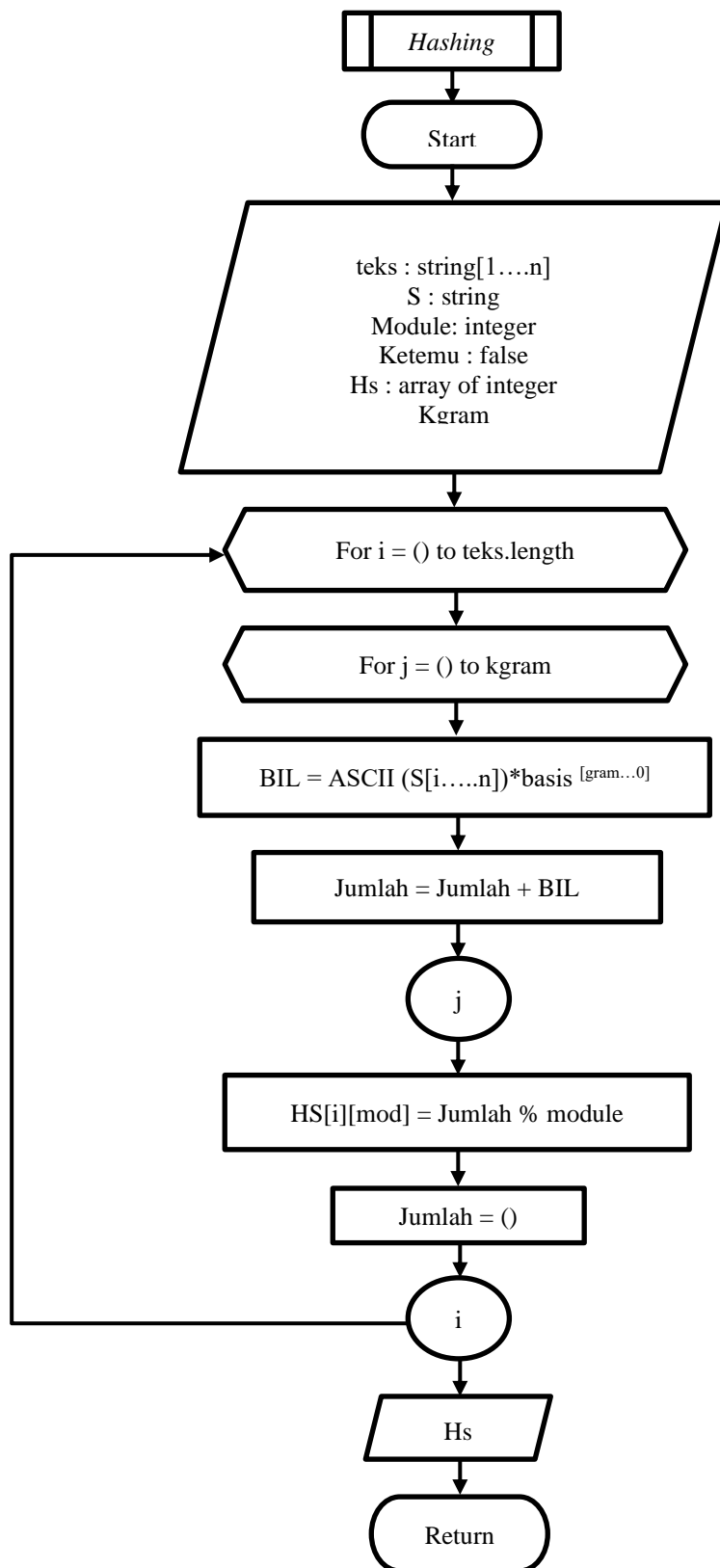


Image 18 Flowchart proses hashing

3. Deteksi Kesamaan

Setelah diketahui nilai *hashing* nya, jumlah hash pada judul asli 97, jumlah hash pada judul uji A 115 dan jumlah hash pada judul uji B 98, hash yang sama pada kedua judul $\sum \text{source} \cap B = 13$ dan $\sum \text{source} \cap C = 9$. Proses selanjutnya adalah menghitung *similarity* yaitu tingkat kesamaan judul asli dan judul penguji A B, yaitu berapa persen tingkat kesamaannya dengan menggunakan persamaan 2. Berikut ini adalah proses menghitung *similarity* 3 judul di atas:

Jumlah hash pada judul asli adalah= $\sum \text{source} = 97$

Jumlah hash pada judul asli adalah= $\sum \text{P1} = 115$

Jumlah hash pada judul asli adalah= $\sum \text{P2} = 98$

Similarity $(2 * X / (P1 + P2)) * 100$

$$XA = S \cap A : 13$$

$$XB = S \cap B : 9$$

$$S = \sum \text{source} : 97$$

$$A = \sum \text{P1} : 115$$

$$B = \sum \text{P2} : 98$$

$$\begin{aligned} \text{Similarity SA} &= (2 * \$XA / (\$S + \$A)) * 100 \\ &= (2 * 13 / (97 + 115)) * 100 \\ &= (26 / 212) * 100 \\ &= 0.122642 * 100 \\ &= \mathbf{12.26415 \%} \end{aligned}$$

$$\begin{aligned} \text{Similarity SB} &= (2 * \$XB / (\$S + \$B)) * 100 \\ &= (2 * 9 / (97 + 98)) * 100 \\ &= (18 / 195) * 100 \\ &= 0.092308 * 100 \\ &= \mathbf{9.230769 \%} \end{aligned}$$

Hasil perhitungan *similarity* dari Judul source dengan Pembanding 1 diatas (SA) adalah **12.26415 %** dan Judul source dengan Pembanding 2 diatas (SB) adalah **9.230769 %**, dan judul yang diuji bias dikatakan tidak plagiat, karena tingkat kesamaannya sangat rendah.

Gambaran proses deteksi kesamaan judul seperti yang terlihat pada flowchart berikut ini.

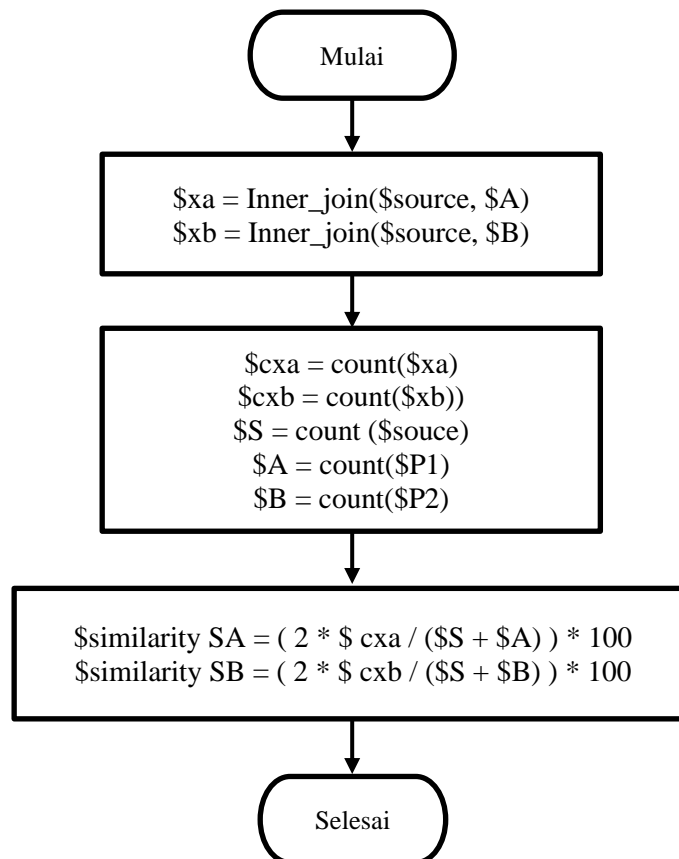


Image 19 Flowchart proses deteksi kesamaan judul

4.5 Desain sistem input output sistem mahasiswa

Pada desain sistem input output ini, digambarkan desain tampilan sistem yang akan dibuat dan diimplementasikan pada sistem untuk tampilan pada mahasiswa.

4.5.1 Design Frontend Interface user

Design tampilan *Frontend user* seperti yang terlihat pada gambar dibawah ini. Halaman utama dirancang dengan beberapa menu yang dapat diakses semua user, terdiri dari halaman *home* atau halaman utama itu sendiri, halaman *about* yaitu halaman yang dirancang untuk informasi tentang sistem tersebut, halaman *timeline* merupakan halaman yang dirancang untuk menampilkan timeline tugas akhir, halaman *announcement* halaman yang dirancang untuk pengumuman, halaman

download yaitu halaman yang dirancang untuk *download* file yang di butuhkan dalam tugas akhir, halaman *contact* halaman yang di rancang untuk informasi *contact* sistem, dan halaman *login* berisi halaman yang di rancang sebagai *form* akses *dashboard user*

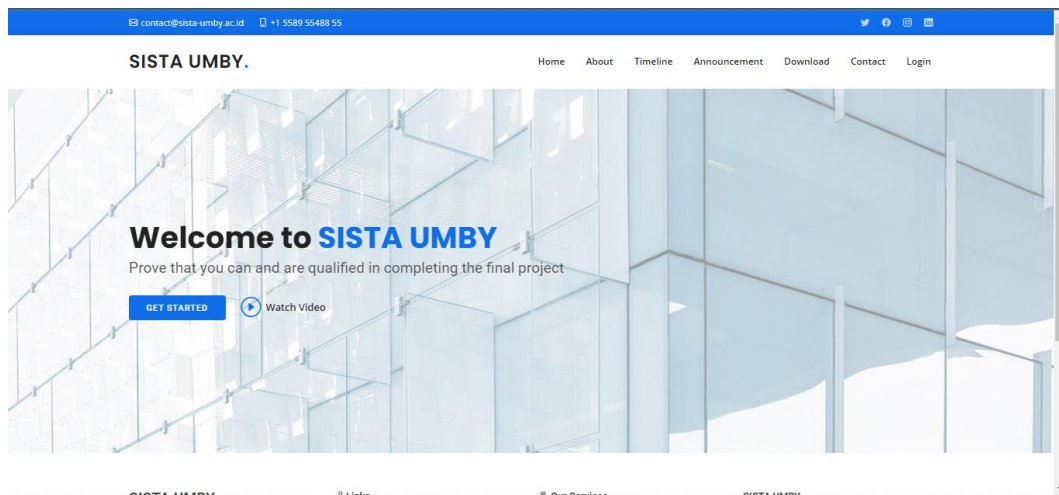


Image 20 Design Frontend Interface user

4.5.2 Design form login user

Design form login user ini di rancang sebagai form akses untuk masuk kedalam dashboard masing masing user.

Image 21 Design form login user

4.5.3 Design dashboard user mahasiswa

Design dashboard ini di rancang sebagai lembar kerja user yang akan di

gunakan sebagai pendukung pelaksanaan tugas akhir hingga selesai. Halaman utama dirancang dengan beberapa menu yang dapat diakses oleh user, terdiri dari halaman *Dashboard* atau halaman utama itu sendiri, halaman bimbingan yaitu halaman yang dirancang sebagai tempat bimbingan mahasiswa, halaman skripsi memiliki sub menu yaitu menu pengajuan judul dan proposal yang digunakan untuk pengajuan judul atau proposal tugas akhir dan sub menu *schedule* dan pengumuman yang menampilkan informasi jadwal seminar proposal dan pengumuman hasil seminar proposal, halaman pendadaran memiliki 3 sub menu 1) Upload syarat pendadaran yang menampilkan form pengumpulan syarat pendadaran yang berfungsi setelah bimbingan sudah selesai dan laporan di terima. 2) jadwal pendadaran merupakan tampilan informasi jadwal pelaksanaan pendadaran. 3) Pengumuman hasil pendadaran menampilkan informasi pengumuman setelah pelaksanaan pendadaran. Selanjutnya menu yudisium memiliki 1 submenu yaitu upload syarat yudisium yang menampilkan form upload syarat yudisium sebagai syarat akhir sebelum wisuda, halaman ini akan di buka setelah ujian pendadaran skripsi di nyatakan lulus.

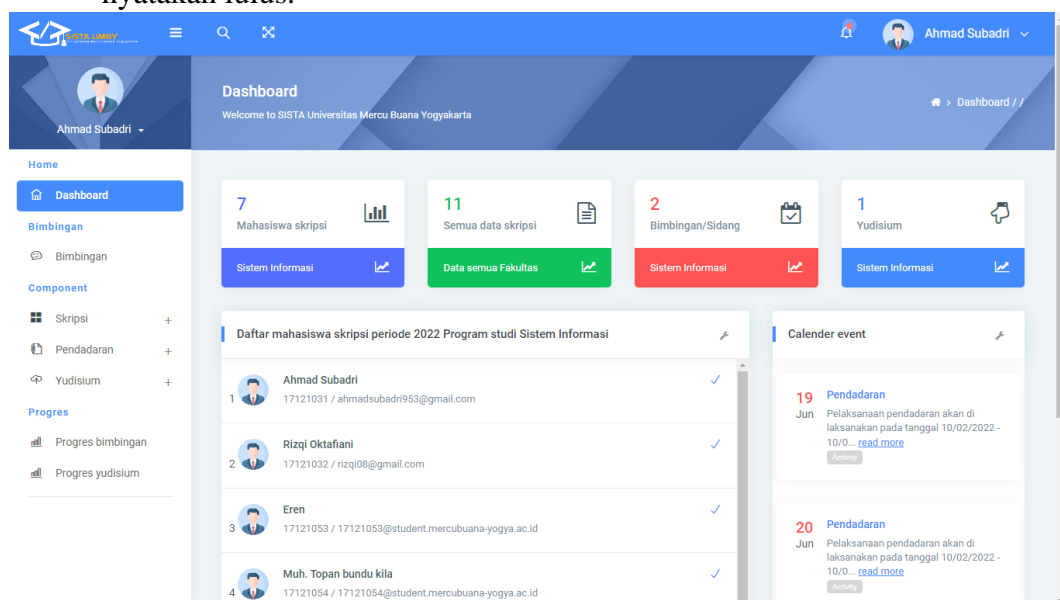


Image 22 Design dashboard user mahasiswa

4.5.4 Design form upload pengajuan judul dan proposal

Design form upload ini yang akan digunakan oleh mahasiswa untuk mengajukan judul dan proposal tugas akhir.

Note: Sebelum melakukan pengajuan judul & proposal pastikan sudah anda siapkan dan anda konsultasikan ke dosen akademik anda.

Form pengajuan judul dan proposal skripsi

Judul area Input

Putri taqwa prasetyaningrum S.Kom

Upload File

Choose file | No file chosen

Upload Pengajuan

Riwayat pengajuan

SISTEM PAKAR DIAGNOSA PENYAKIT KULIT PADA MANUSIA MENGGUNAKAN METODE NAIVE BAYES BERBASIS WEB

Update : 2022-07-17 23:26:57

Diserika

Image 23 Design form upload pengajuan judul dan proposal

4.5.5 Design tampilan jadwal dan pengumuman seminar proposal

Tampilan jadwal dan pengumuman seminar proposal ini di rancang untuk informasi kapan akan di laksanakan seminar proposal untuk mahasiswa itu sendiri dan pengumuman hasil seminar proposal setelahnya.

Detail seminar proposal

Penguji Seminar proposal	Jadwal seminar proposal
Putri taqwa prasetyaningrum S.Kom	Hari/tgl : Rabu, 13 Juli 2022 / 08:00
Albert Yakobus Chandra S.Kom	Tempat : Ruang 52
Anief fauzan rozi S.Kom, M.Eng.	Kegiatan : Seminar Proposal

Pengumuman hasil seminar proposal

Pengumuman hasil Seminar proposal

*Judul dan proposal anda Diserika

Pengumuman hasil akhir seminar proposal

Note: Tahap lanjutan ke Bimbingan skripsi setelah plotting dosen pembimbing skripsi sudah di sahkan

— vital Records

Image 24 Design tampilan jadwal dan pengumuman seminar proposal

4.6 Desain sistem input output dosen kepala program studi

Pada desain sistem input output ini, digambarkan desain tampilan sistem yang akan dibuat dan diimplementasikan pada sistem untuk tampilan sistem pada dosen.

4.6.1 *Design* hasil uji Rabin Karp



4.6.2 Design dashboard

Didalam tampilan dashboard dosen ini memiliki menu utama yang tidak jauh beda dengan menu yang ada di dashboard mahasiswa, yang membedakan sub menu dari masing masing menu utama nya saja.

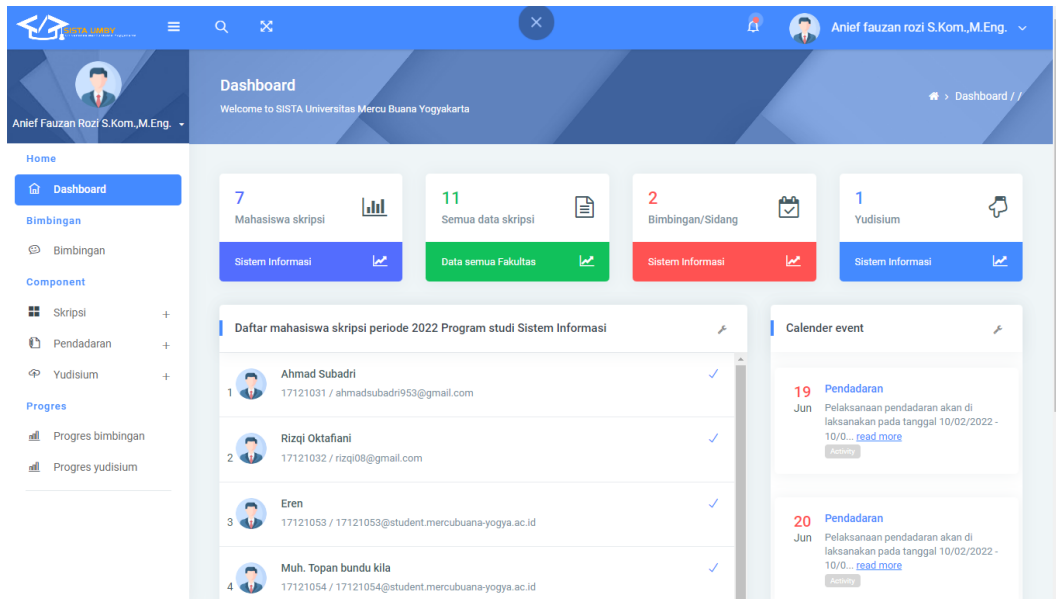


Image 25 Design dashboard dosen

4.6.3 Design menu bimbingan

menu yang di gunakan untuk bimbingan mahasiswa dan dosen pembimbing.

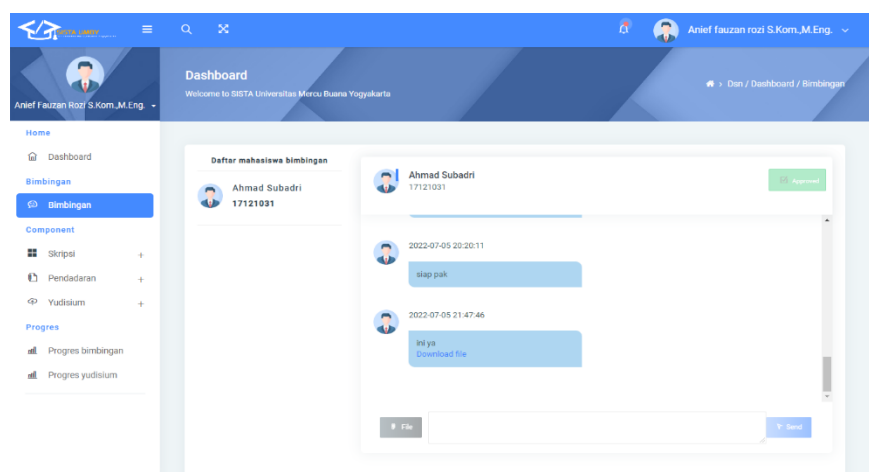


Image 26 Design menu bimbingan dosen

4.6.4 Design menu skripsi

1. Data pengajuan judul & proposal : menampilkan seluruh data pengajuan tugas akhir dari mahasiswa.

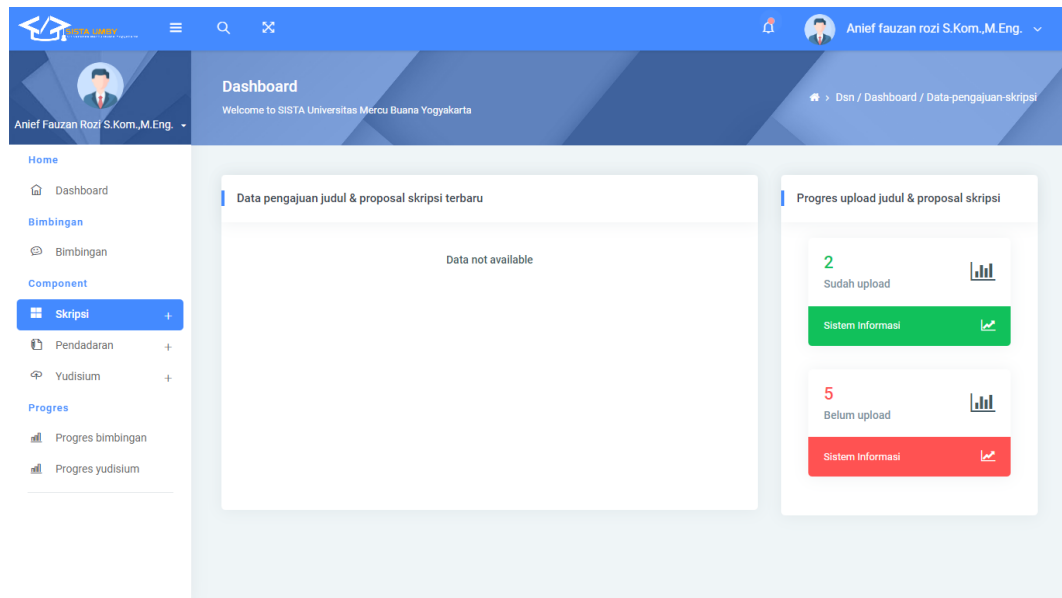


Image 27 Design menu skripsi submenu data pengajuan judul proposal

2. Schedule dan pengumuman hasil sempro : menu ini berisi beberapa fitur yang penting yaitu 1) Form plotting pengujian seminar proposal. 2) Setting jadwal pelaksanaan seminar proposal. 3) Detail hasil yang berisi informasi hasil cek *plagiarism* dan hasil *feedback* dari pengujian seminar proposal. 4) Kirim pengumuman sempro berisi form *feedback* keputusan hasil seminar proposal yang akan di kirim ke mahasiswa sebagai pengumuman hasil seminar proposal.

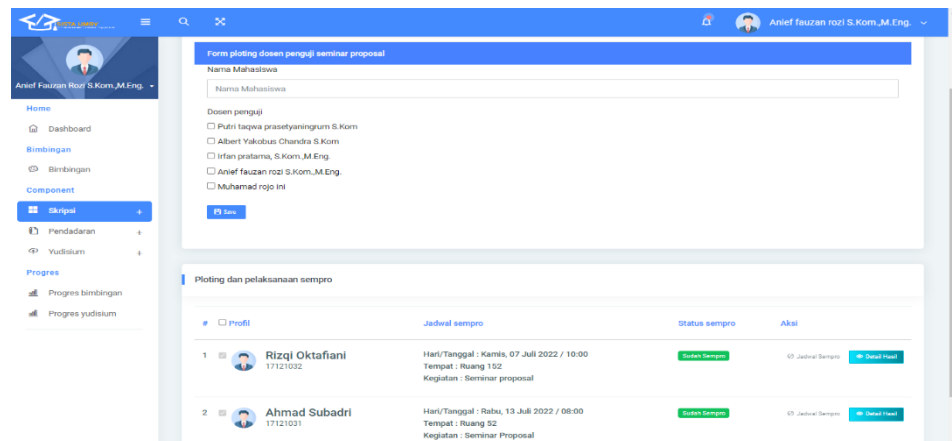


Image 28 Design sub menu skripsi Schedule dan pengumuman hasil sempro

3. Data mahasiswa seminar proposal : berisi data mahasiswa yang akan dosen uji saat seminar proposal jika dosen tersebut terpilih sebagai penguji sempro.

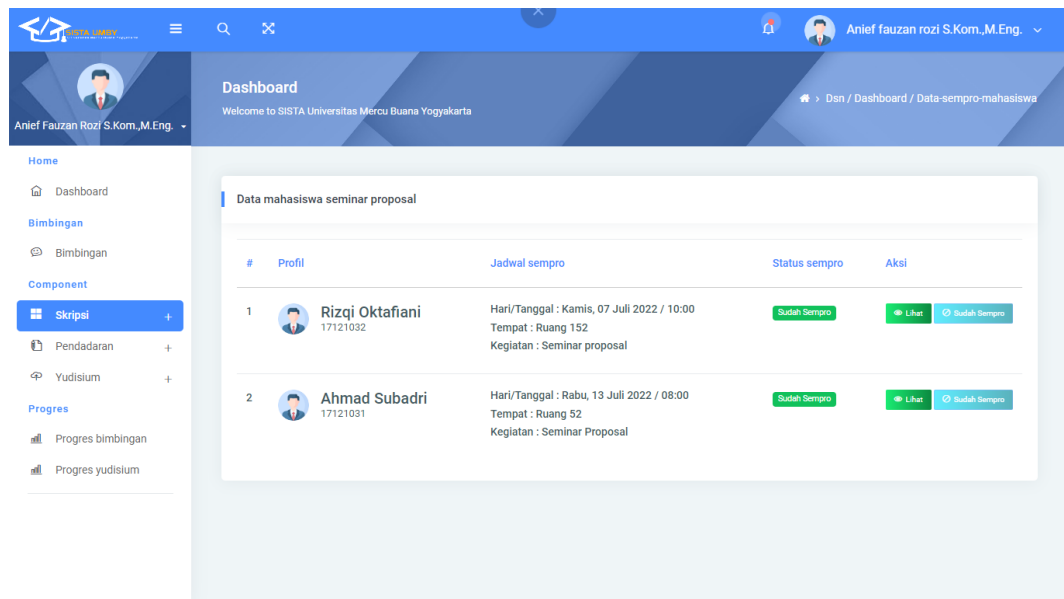


Image 29 Design submenu skripsi Data mahasiswa seminar proposal

4. Ploting dosen pembimng : menu ini di jadikan sebagai menu untuk penentuan dosen pembimbing tugas akhir, fitur ini terbuka setelah seminar proposal sudah di terima dan di nyatakan lulus sempro.

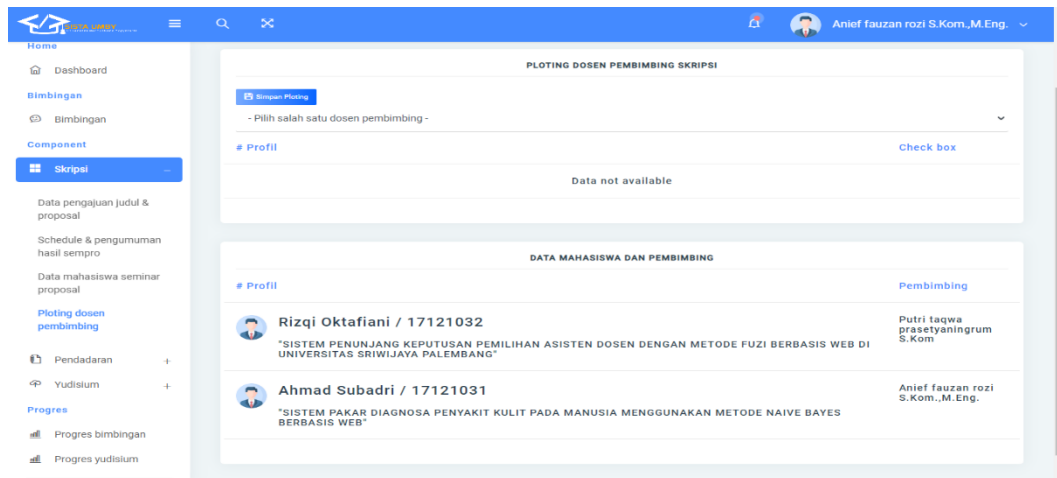


Image 30 Design submenu skripsi Ploting dosen pembimng

4.6.5 Design menu pendadaran

1. Syarat pendadaran : submenu yang di gunakan sebagai setting syarat pendadaran, mulai dari tambah syarat, *edit* syarat, *delete* syarat dan *publish/unpublish* syarat.

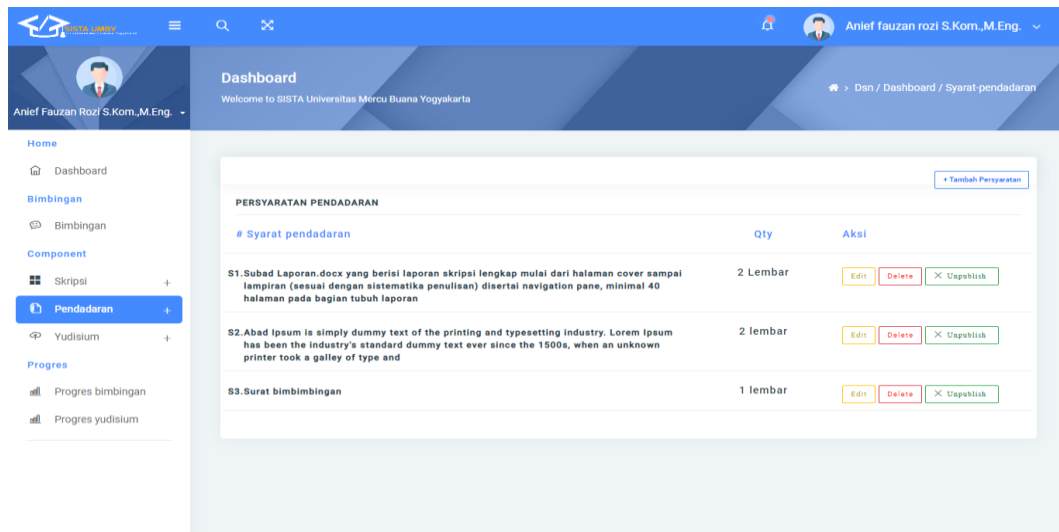


Image 31 Design submenu pendadaran Syarat pendadaran

2. Mahasiswa pendadaran : berisi daftar semua mahasiswa yang sudah masuk ke tahap pendadaran.

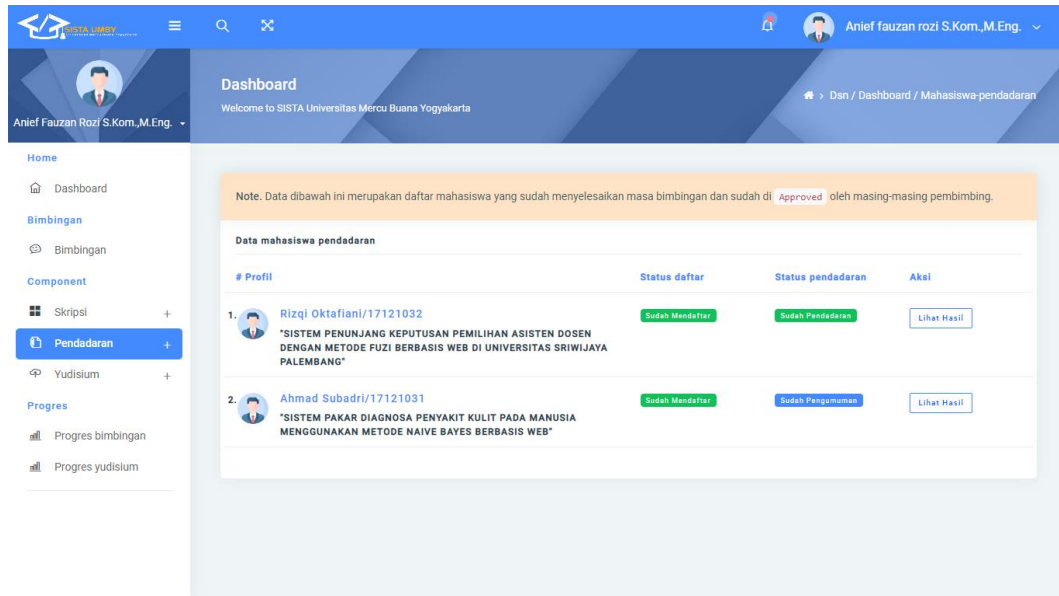


Image 32 Design submenu pendadaran Mahasiswa pendadaran

dan memiliki fitur tambahan yaitu Lihat hasil yang berisi hasil nilai pendadaran setelah pendadaran sudah di laksanakan.

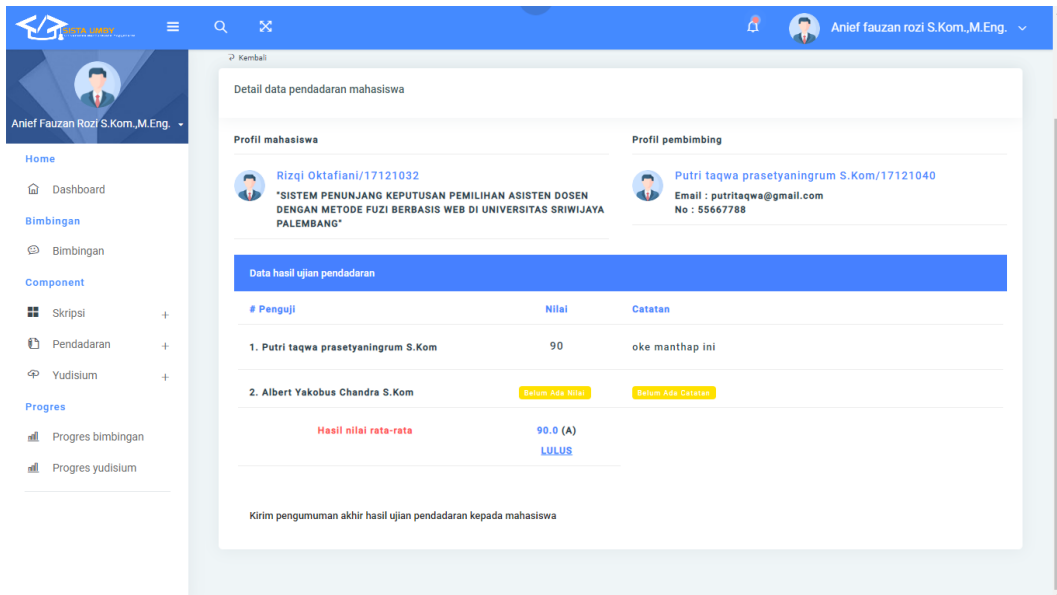


Image 33 Detail hasil pendadaran

3. Penentuan jadwal dan penguji : setting jadwal pendadaran dan penguji sidang pendadaran.

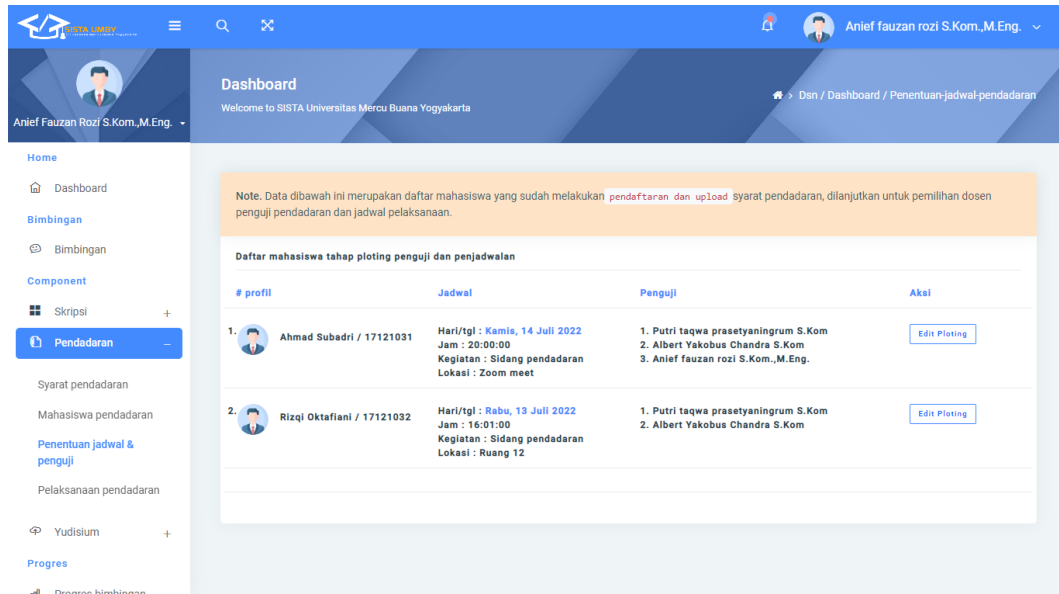


Image 34 Design submenu pendadaran Penentuan jadwal dan penguji

4. Pelaksanaan pendadaran : daftar mahasiswa yang akan di uji pendadaran dan pelaksanaan pendadaran.

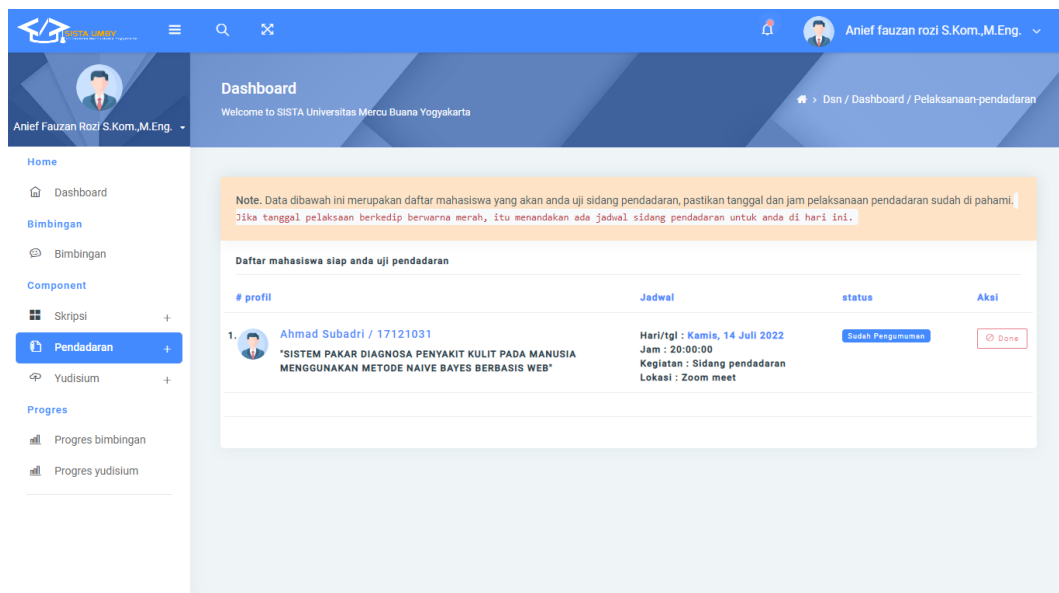


Image 35 Design submenu pendadaran Pelaksanaan pendadaran

4.6.6 Design menu yudisium

1. Syarat yudisium : *setting* syarat yudisium

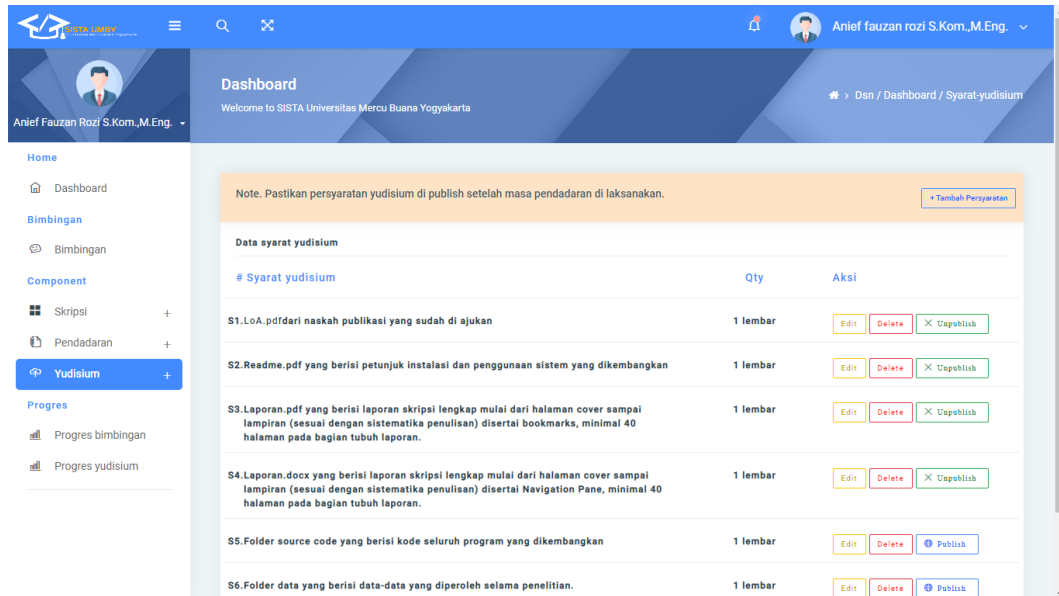


Image 36 Design submenu yudisium Syarat yudisium

2. Mahasiswa yudisium : menampilkan data mahasiswa yang sudah masuk tahap yudisium.

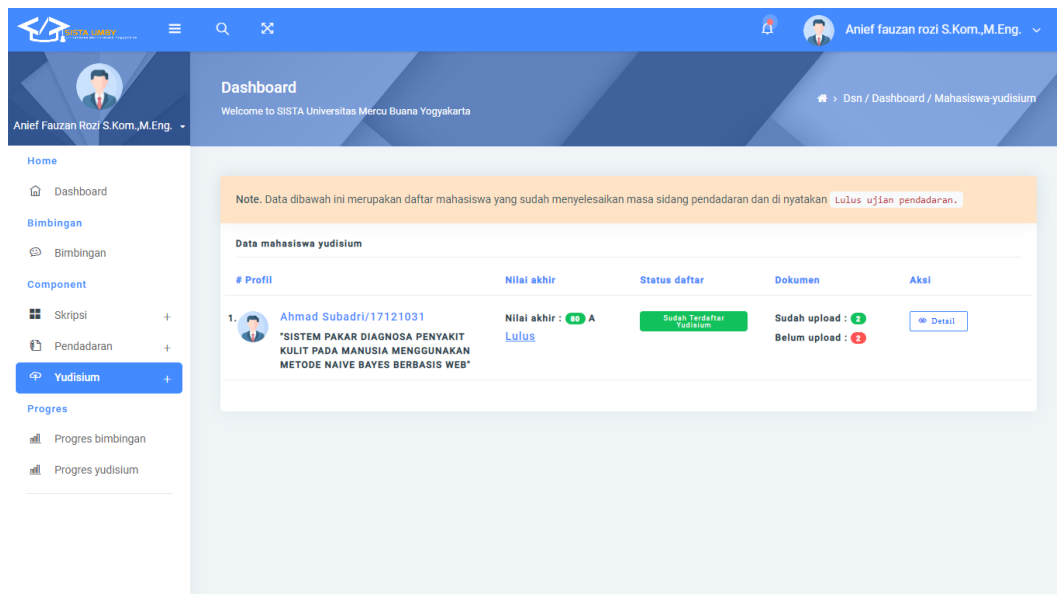


Image 37 Design submenu yudisium Mahasiswa yudisium

4.6.7 Design menu progres bimbingan

Menu ini berisi daftar progres bimbingan seluruh mahasiswa. Mulai dari total kirim bimbingan dan total balasan dari pembimbing.

The screenshot shows a dashboard for a user named Anief Fauzan Rozi S.Kom.,M.Eng. The main content area is titled 'Daftar progres bimbingan' and contains a table with the following data:

# Profil	Total bimbingan	Total feedback	Aksi
1. Ahmad Subadri / 17121031	Total bimbingan : 6 X	Total feedback : 4 X <i>Total feedback kurang dari ketentuan</i>	Detail
2. Rizqi Oktafiani / 17121032	Total bimbingan : 0 X <i>Total bimbingan kurang dari ketentuan</i>	Total feedback : 1 X <i>Total feedback kurang dari ketentuan</i>	Detail

Image 38 Design menu progres bimbingan

4.6.8 Design menu progres yudisium

Menu ini berisi progres upload syarat seluruh mahasiswa.

The screenshot shows a dashboard for a user named Anief Fauzan Rozi S.Kom.,M.Eng. The main content area is titled 'Data progres upload dokumen yudisium mahasiswa' and contains a table with the following data:

# Profil	S1	S2	S3	S4	S5	S6
1. Ahmad Subadri / 17121031	✓	✓	✗	✗	✗	✗

Image 39 Design menu progres yudisium

BAB V PENUTUP

5.1 Kesimpulan.

Berdasarkan hasil dari pembahasan dan uji coba sistem maka peneliti dapat menyimpulkan bahwa Algoritma Rabin-Karp berhasil diimplementasikan pada sistem deteksi *plagiarism* pada judul tugas akhir berbasis web. Sistem ini berhasil mendeteksi dan menampilkan persentase kemiripan judul antar mahasiswa. Hasil perhitungan akurasi dengan metode Rabin-Karp mencapai 70% akurat. Dan algoritma yang di gunakan pada sistem ini tidak memiliki perbedaan persentase saat urutan perbandingan diubah.

5.2 Saran.

Dalam penyusunan penelitian ini, masih terdapat banyak kekurangan yang sangat memungkinkan untuk diperbaiki dan disempurnakan pada pengembangan selanjutnya. Beberapa saran yang dapat diberikan untuk pengembangan adalah:

1. Sistem ini diharapkan dapat membantu untuk mencegah secara dini kemungkinan terjadinya kegiatan plagiarisme dalam penggunaan judul tugas akhir mahasiswa.
2. Sistem ini perlu disempurnakan dengan penambahan fasilitas untuk mendeteksi plagiarisme langsung pada dokumen tugas akhir, sehingga akurasi ada atau tidaknya suatu kegiatan plagiarisme dapat diukur secara luas dan pasti.

DAFTAR PUSTAKA

- Alamsyah, N. (2017). PERBANDINGAN ALGORITMA WINNOWING DENGAN ALGORITMA RABIN KARP UNTUK MENDETEKSI PLAGIARISME PADA KEMIRIPAN TEKS JUDUL SKRIPSI. *Technologia*, 124-134.
- Al-Fedaghi, S. (2011). Developing Web Applications. *International Journal of Software*.
- Andi Riansyah, E. N. (2017). Penerapan Sistem Informasi Monitoring Tugas Akhir Untuk Memantau Perkembangan Tugas Akhir Mahasiswa Universitas Islam Sultan Agung. *Jurnal Transistor Elektro dan Informatika (TRANSISTOR EI)*, 100-109.
- Anggadini, S. D. (2013). Analisis Sistem Informasi Manajemen Berbasis Komputer dalam Proses Pengambilan Keputusan. *Maj. Ilm Unikom*, 176–187.
- Asti Herliana, P. M. (2016). SISTEM INFORMASI MONITORING PENGEMBANGAN SOFTWARE PADA TAHAP DEVELOPMENT BERBASIS WEB. *Jurnal Informatika*, 41-50.
- Asvarizal Filcha, M. H. (2019). Implementasi Algoritma Rabin-Karp untuk Pendeteksi Plagiarisme pada Dokumen Tugas Mahasiswa. *JUITA p-ISSN: 2086-9398 (print); e-ISSN: 2579-9801 (online); Volume VII, Nomor 1, Mei 2019*, 25-32.
- Atmopawiro, A. (2006). *Pengkajian dan Analisis Tiga Algoritma Efisien Rabin-Karp, Knuth-morris-Pratt, dan Boyer-Moore dalam pencarian Pola dalam Suatu Teks*. Bandung: STEI ITB.
- Basuki Awan, A. p. (2010). Membangun Web berbasis PHP dengan framwork Codeigniter. *Lokomedia*.
- Elchison, J. M. (2012). Cedarville University - CS3410-Algorithm. [Online]. Available: http://www.cedarville.edu/personal/personalpages/shomper/cs3410_web/resources/rabin_karp_matching.ppt.
- Erlangga, F. (2013). Perancangan Sistem Manajemen Kehadiran Kuliah Mahasiswa di Kelas DenganKartu Identitas Berbasis Teknologi RFID. *Perancangan Sistem Manajemen Kehadiran Kuliah* , ol. XV. No.1.
- Firdaus, H. B. (2008). Deteksi Plagiat Dokumen Menggunakan Algoritma Rabin-Karp.
- Ibrahim, A. (2011). Pengembangan Sistem Informasi Monitoring Tugas Akhir Berbasis Short Message Service (SMS) Gateway di Fasilkom Unsri. *08-JUSI-Vol-1-No-2- _Pengembangan-Sistem-Informasi-Monitoring-TA-Berbasis-SMS-di-Fasilkom-Unsri*.
- Imam Nawawi, P. P. (2019). DETEKSI PLAGIARISME PADA DOKUMEN

SKRIPSI BERDASARKAN TINGKAT KESAMAAN DENGAN
MENGUNAKAN METODE LONGEST COMMON SUBSEQUENCE .
JANAPATI, 217-226.

- J.L, W. (2007). *Systems Analysis and Design Methods*. New York: McGraw-Hill.
- Langgeni, D. e. (2010). *Clustering Artikel Berita Berbahasa Indonesia Menggunakan Unsupervised Feature Selection*. Yogyakarta: Seminar Nasional Informatika. (22 Mei 2010).
- Neva Satyahadewi, N. M. (2019). SISTEM INFORMASI MONITORING TUGAS AKHIR (SIMTA) BERBASIS WEB FAKULTAS MIPA UNIVERSITAS TANJUNGPURA. *CESS (Journal of Computer Engineering System and Science)*, 83-87.
- Ngroho, E. (2011). *Perancangan Sistem Deteksi Plagiarism Dokumen Teks Dengan Menggunakan Algoritma Rabin-Karp Tugas Akhir*. Malang: Universitas Brawijaya Malang.
- O'Brien, J. a. (2010). *Introduction to Information Systems Fifteenth Edition*. McGraw-Hill International Edition.
- Pratama, M. e. (2013). *Aplikasi Pendeteksi Dupliasi Dokumen Teks Bahasa Indonesia Menggunakan Algoritma Winnowing Dengan Metode K-Gram Dan Synonym Recognition*. Malang: Jurusan Teknik Informatika Universitas Muhammadiyah Malang.
- Putra, I. (2013). *Case Folding Pada Konten Web Dengan PHP*. Bandung: Bandung: Jurusan Teknologi Informasi Fakultas Teknik Universitas Udayana. Merpati Vol. 1, No. 3.
- Satia Suhada, S. B. (2017). IMPLEMENTASI ALGORITMA RABIN KARP DAN STEMMING NAJIEF ANDRIANI UNTUK DETEKSI PLAGIARISME DOKUMEN. *SWABUMI*, 84-89.
- Schleimer, S. D. (2003). Winnowing: Local Algorithms for Document Fingerprinting. in *SIGMOD, Sandiego, CA*.
- Setiawan, R. &. (2016). Pengembangan Aplikasi Pengendalian Tugas Akhir Berbasis Web Sisi Mahasiswa, Dosen, dan Staff USI. *Algoritma*, 34-40.
- sidh, R. (2013). PERANAN BRAINWARE DALAM SISTEM INFORMASI MANAJEMEN. *Jurnal Computech & Bisnis*, 19-29.
- Solusi, D. S. (2009). Aplikasi Terintegrasi Sistem Informasi Akademik Perguruan Tinggi, Acuan Teknik Untuk Pengembang. *Delta Sistem Solusi*.
- Sulhan, M. (2015). Sistem Monitoring Tugas Akhir Berbasis User Generated Content Pada Program Studi Sistem Informasi Universitas Kanjuruhan Malang. *SMATIKA Jurnal STIKI Informatika Jurnal*, 58-68.
- Susanto, A. (2004). *Sistem Informasi Manajemen Konsep dan Pengembanganya*. Bandung: Penerbit Lingga Jaya.
- T. H. Cormen, C. E. (2001). *Introduction to algoritma*. USA: MIT Press.

- Taftayani, M. B. (2019). Perancangan Aplikasi Pendidikan Berbasis Website Menggunakan Framework Laravel Dan Framework Css Materialize Pada Startup Sinau Yo. *eProceedings of Engineering*.
- Wulansari, E. (2013). Perancangan Sistem Informasi Manajemen Tugas Akhir Berbasis WebJurusan Teknik Sipil Fakultas Teknik Universitas Tanjungpura. *Perancangan_Sistem_Informasi_Manajemen_T*.

LAMPIRAN

Lampiran A Biodata Peneliti

Nama Lengkap : Ahmad subadri
NIM : 17121035
Alamat Asal : RT 12 RW 03 Desa Margo Mulyo Kec. Tungkal Jaya Kab.
Musi banyuasin, Sumatra Selatan
Alamat di Yogyakarta: Jl. Afandi soropadan no 28B, Condongcatur, Kec. Depok,
Kota Sleman Yogyakarta
No. Hp : +62 812-1843-6055
e-Mail : 17121035@student.mercubuana-yogya.ac.id
ahmadsubadri953@gmail.com
Website : bio-as.com
Fakultas : Fakultas Teknologi Informasi
Program Studi : Sistem Informasi
Universitas : Universitas Mercu Buana Yogyakarta

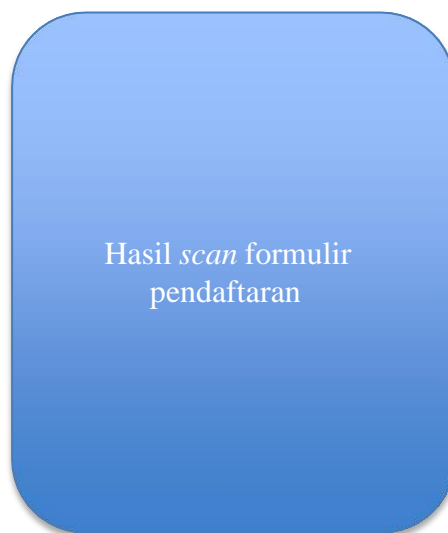
Lampiran B *Scan* Dokumen Bimbingan Skripsi

Dokumen bimbingan skripsi yang dilampirkan berbentuk hasil scan yang telah terisi lengkap berikut tanda tangan dosen pembimbing.

Lampiran C Artifak/Data Penelitian

Isi yang dilampirkan sebaiknya hasil *scan* dari bentuk fisik yang nyata. Misalnya pada instrumen pendaftaran terdapat 2 formulir yaitu formulir pendaftaran dan surat pernyataan kesanggupan pembayaran administrasi.

Format penamaan lampiran hasil scan seperti pada penamaan gambar, yaitu terletak di bawah, spasi 1, ukuran tulisan 9pt, Times New Roman. Contoh:



Gambar C. 1 Scan Formulir Pendaftaran

Lampiran D Kode Program

Berikut adalah kode program untuk fitur.....

```
1 <?php
2 class proposal extends controller
3 {
4     function proposal()
5     {
6         parent::Controller();
7         $this->security_model->login_mhs_check();
8     }
9
10    function create()
11    {
12        if($_SERVER['REQUEST_METHOD'] == "POST")
13        {
14            $this->form_validation->set_rules('proposal_tema', 'Tema / Topik Tesis', 'trim|xss_clean');
15
16            if ($this->form_validation->run() == FALSE)
17            {
18                $data['err'] = error(validation_errors());
19                $tmp['content'] = $this->load->view('mahasiswa/proposal/create', $data, TRUE);
20            }
21            else
22            {
23                $file_size = $_FILES["image"]["size"];
24                $new_image = $_FILES["image"]["name"];
25
26                $this->proposal_lib->add($mhs_id, $new_image, $file_size, $this->session->userdata('username_mahasiswa'), $ambil_tema_dart);
27            }
28
29            $data['judul'] = "Form Pengajuan Proposal Tesis";
30            // load template
31            $tmp['content'] = $this->load->view("mahasiswa/proposal/create", $data, TRUE);
32            $this->load->view("template", $tmp);
33        }
34    }
35 }
36
37 7>
```

Gambar D. 1 Kode Program Form Pendaftaran

```
1 <?php
2 class proposal extends controller
3 {
4     function proposal()
5     {
6         parent::Controller();
7         $this->security_model->login_mhs_check();
8     }
9
10    function create()
11    {
12        if($_SERVER['REQUEST_METHOD'] == "POST")
13        {
14            $this->form_validation->set_rules('proposal_tema', 'Tema / Topik Tesis', 'trim|xss_clean');
15
16            if ($this->form_validation->run() == FALSE)
17            {
18                $data['err'] = error(validation_errors());
19                $tmp['content'] = $this->load->view('mahasiswa/proposal/create', $data, TRUE);
20            }
21            else
22            {
23                $file_size = $_FILES["image"]["size"];
24                $new_image = $_FILES["image"]["name"];
25
26                $this->proposal_lib->add($mhs_id, $new_image, $file_size, $this->session->userdata('username_mahasiswa'), $ambil_tema_dart);
27            }
28
29            $data['judul'] = "Form Pengajuan Proposal Tesis";
30            // load template
31            $tmp['content'] = $this->load->view("mahasiswa/proposal/create", $data, TRUE);
32            $this->load->view("template", $tmp);
33        }
34    }
35 }
36
37 7>
```

Gambar D. 2 Kode Program Proses Penyimpanan Data ke Database.

