

**SKRIPSI**

**INTEGRASI SISTEM PENILAIAN KELAYAKAN PROPOSAL TUGAS  
AKHIR DENGAN SISTEM DETEKSI PLAGIASI**

**Disusun dan diajukan oleh**

**MALYANA ARIANI**

**D421 16 301**



**DEPARTEMEN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS HASANUDDIN**

**MAKASSAR**

**2021**

**LEMBAR PENGESAHAN SKRIPSI**

**INTEGRASI SISTEM PENILAIAN KELAYAKAN PROPOSAL TUGAS  
AKHIR DENGAN SISTEM DETEKSI PLAGIASI**

Disusun dan diajukan oleh

**MALYANA ARIANI**

**D421 16 301**

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka penyelesaian studi Program Sarjana Program Studi Teknik Informatika Fakultas

Teknik Universitas Hasanuddin

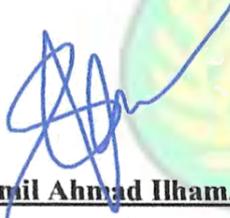
Pada tanggal 05 Februari 2021

dan dinyatakan telah memenuhi syarat kelulusan

Menyetujui

Pembimbing Utama,

Pembimbing Pendamping,

  
Dr. Amil Ahmad Ilham, S.T., M.IT.  
NIP. 19731010 199802 1 001

  
Igra Aswad, S.T., M.T.  
NIP. 19901128 201904 3 001

Ketua Program Studi,



  
Dr. Amil Ahmad Ilham, S.T., M.IT.  
NIP. 19731010 199802 1 001

## PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini :

Nama : MALYANA ARIANI  
NIM : D421 16 301  
Program Studi : Teknik Informatika  
Jenjang : S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul :

### INTEGRASI SISTEM PENILAIAN KELAYAKAN PROPOSAL TUGAS AKHIR DENGAN SISTEM DETEKSI PLAGIASI

Adalah karya tulisan saya sendiri, bukan merupakan pengambilan alihan tulisan orang lain dan bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan isi skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Makassar, 15 Februari 2021

Yang Menyatakan,



*Malyana Ariani*  
MALYANA ARIANI

## **KATA PENGANTAR**

Puji dan syukur penulis panjatkan kepada Allah SWT karena berkat rahmat dan karunia-Nya sehingga penulis dapat mampu menyelesaikan laporan skripsi tugas akhir ini yang berjudul “INTEGRASI SISTEM PENILAIAN KELAYAKAN PROPOSAL TUGAS AKHIR DENGAN SISTEM DETEKSI PLAGIASI” sebagai salah satu syarat dalam menyelesaikan jenjang Strata-1 pada Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin.

Penulis menyadari bahwa dalam penyusunan laporan skripsi tugas akhir ini tidak lepas dari bantuan, bimbingan serta dukungan dari berbagai pihak. Oleh karena itu, penulis menyampaikan ucapan terima kasih sedalam-dalamnya kepada:

1. Allah SWT atas semua berkat rahmat, karunia serta pertolongan-Nya;
2. Kedua Orang tua, saudara dan keluarga penulis yang senantiasa selalu memberikan dukungan, doa, dan semangat serta memberikan dukungan materiel kepada penulis;
3. Bapak Dr. Amil Ahmad Ilham, S.T., M.IT., selaku pembimbing I dan Bapak Iqra Aswad, S.T., M.T., selaku pembimbing II yang selalu menyediakan waktu, tenaga, dan pikiran dalam memberikan bimbingan dan masukan yang bermanfaat kepada penulis dalam penyusunan tugas akhir;
4. Bapak Dr. Amil Ahmad Ilham, S.T., M.IT., selaku Ketua Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin;

5. Bapak Prof. Dr. Ir. Andani, M.T., dan Bapak Dr. Eng. Zulkifli Tahir, S.T., M.Sc., yang senantiasa memberikan nasehat, masukan, serta perhatian kepada penulis dalam penyusunan tugas akhir;
6. Teman-teman Igniter16 FT UH atas dukungan dan semangat yang diberikan selama ini;
7. Segenap Staf Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin yang telah membantu penulis;
8. Serta orang-orang berpengaruh lainnya yang tak sempat penulis sebutkan satu persatu, yang telah banyak meluangkan waktu, tenaga dan pikiran serta tanpa sadar telah menjadi inspirasi penulis.

Akhir kata, penulis berharap semoga Allah SWT. berkenan membalas segala kebaikan dari semua pihak yang telah banyak membantu. Penulis berharap semoga Laporan Skripsi Tugas Akhir ini dapat memberikan manfaat bagi pengembangan ilmu. Amin.

Makassar, Februari 2021

Penulis

## ABSTRAK

Pada era perkembangan teknologi saat ini, sebuah informasi sangat dibutuhkan dalam kehidupan manusia dan sudah banyak digunakan dalam berbagai bidang salah satunya dalam bidang pendidikan di suatu perguruan tinggi. Studi kasus pada Program Studi Teknik Informatika Universitas Hasanuddin menunjukkan bahwa telah menerapkan beberapa sistem informasi dalam membantu proses pekerjaan di kampus, namun belum secara keseluruhan. Masih terdapat beberapa pekerjaan yang belum menerapkan sistem informasi di dalamnya, salah satunya yaitu belum tersedia sistem informasi yang dapat membantu dalam proses pengumpulan proposal tugas akhir mahasiswa yang dimana masih menggunakan media email. Untuk itu pada penelitian ini dilakukan pengembangan perangkat lunak berupa *website* Sistem Penilaian Kelayakan Proposal Tugas Akhir Mahasiswa dengan Sistem Deteksi Plagiasi. Dengan adanya *website* ini, diharapkan dapat membantu institusi dalam mengelola proposal tugas akhir mahasiswa, dapat memudahkan admin dalam pembagian tugas bagi dosen untuk *me-review* proposal tugas akhir dan adapun pada *website* yang akan dibuat nantinya dilengkapi dengan fitur tambahan yaitu pengecekan *similarity* guna untuk mengetahui tingkat kemiripan pada proposal mahasiswa.

**Kata Kunci:** pengembangan perangkat lunak, *webiste*, sistem penilaian, proposal tugas akhir.

## DAFTAR ISI

LEMBAR PENGESAHAN SKRIPSI .....	i
PERNYATAAN KEASLIAN .....	ii
KATA PENGANTAR .....	iii
ABSTRAK .....	v
DAFTAR ISI .....	vi
DAFTAR GAMBAR .....	xi
DAFTAR TABEL .....	xiii
BAB I PENDAHULUAN .....	1
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	2
1.3. Tujuan Penelitian .....	3
1.4. Manfaat Penelitian .....	3
1.5. Batasan Masalah .....	4
1.6. Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA .....	5
2.1. Sistem Informasi .....	5
2.1.1. Kemampuan Sistem Informasi .....	5
2.2. Sistem Informasi Akademik .....	6
2.3. <i>Website</i> .....	7
2.3.1. Kategori <i>Website</i> .....	7
2.4. Bahasa Pemrograman.....	8
2.5. HTML.....	9

2.5.1. Struktur Dasar HTML .....	9
2.6. PHP .....	10
2.6.1. Keunggulan PHP .....	11
2.7. CodeIgniter .....	11
2.7.1. MVC ( <i>Model View Controller</i> ).....	12
2.7.2. Keunggulan CodeIgniter.....	13
2.8. <i>Database</i> .....	13
2.9. MySQL.....	14
2.9.1. Keunggulan MySQL .....	14
2.10. XAMPP .....	15
2.11. PhpMyAdmin.....	15
2.12. Sublime Text.....	15
2.13. Plagiarisme .....	16
2.13.1. Ambang Batas ( <i>Threshold</i> ) Plagiarisme.....	16
2.14. <i>Text Preprocessing</i> .....	17
2.15. Algoritma Rabin Karp.....	17
2.15.1. <i>K-Gram</i> .....	18
2.15.2. <i>Hashing</i> .....	19
2.16. <i>Similarity</i> .....	21
2.16.1. Pengukuran Nilai <i>Similarity</i> .....	22
2.17. Metode Pengembangan Software.....	22
2.18. Penelitian Terkait .....	24
<b>BAB III METEDEOLOGI PENELITIAN.....</b>	<b>26</b>

3.1. Studi Literatur .....	27
3.2. Persiapan Kebutuhan Penelitian .....	27
3.2.1. Persiapan Kebutuhan Data.....	27
3.2.2. Persiapan Kebutuhan Perangkat.....	28
3.3. Perancangan Sistem .....	28
3.3.1. Perancangan <i>Use Case Diagram</i> .....	29
3.3.1.1. Identifikasi Aktor .....	29
3.3.1.2. Identifikasi Diagram <i>Use Case</i> .....	29
3.3.1.3. <i>Use Case Diagram</i> .....	31
3.3.1.4. <i>Use Case</i> Skenario.....	32
3.3.2. Perancangan <i>Activity Diagram</i> .....	40
3.3.2.1. <i>Activity Diagram Login</i> .....	40
3.3.2.2. <i>Activity Diagram</i> Konfigurasi.....	41
3.3.2.3. <i>Activity Diagram</i> Tambah Proposal .....	42
3.3.2.4. <i>Activity Diagram</i> Cek <i>Similarity</i> .....	43
3.3.2.5. <i>Activity Diagram</i> Admin Tolak Proposal .....	45
3.3.2.6. <i>Activity Diagram</i> Bagi Tugas <i>Review</i> .....	46
3.3.2.7. <i>Activity Diagram</i> Penerimaan Proposal.....	48
3.3.2.8. <i>Activity Diagram</i> <i>Review</i> Proposal.....	49
3.3.2.9. <i>Activity Diagram</i> Lewat Batas <i>Review</i> .....	51
3.3.3. Perancangan <i>Database</i> .....	52
3.3.3.1. Tabel <i>User</i> .....	52
3.3.3.2. Tabel Konfigurasi.....	52

3.3.3.3. Tabel Proposal.....	53
3.3.3.4. Tabel <i>Review</i> .....	54
3.3.3.5. Tabel Seminar .....	54
3.4. Pembuatan Sistem.....	55
3.4.1. Implementasi Algoritma Rabin Karp .....	55
3.5. Uji Coba Sistem.....	58
BAB IV HASIL DAN PEMBAHASAN .....	59
4.1. Hasil Pembuatan Sistem.....	59
4.1.1. Halaman <i>Login</i> .....	59
4.1.2. Halaman Admin .....	60
4.1.2.1. Halaman Utama Admin .....	60
4.1.2.2. Halaman Menu Konfigurasi.....	60
4.1.2.3. Halaman Menu Proposal.....	61
4.1.2.4. Halaman Menu Rekap Nilai.....	64
4.1.2.5. Halaman Menu Statistik .....	65
4.1.3. Halaman Dosen .....	65
4.1.3.1. Halaman Menu Penerimaan.....	65
4.1.3.2. Halaman Menu Proposal.....	66
4.1.3.3. Halaman Menu Seminar Proposal.....	67
4.1.4. Halaman Mahasiswa.....	69
4.1.4.1. Halaman Menu Proposal.....	69
4.1.4.2. Halaman Tambah Proposal.....	69
4.1.4.3. Halaman Cek <i>Similarity</i> .....	70

4.1.4.4. Halaman Aksi Detail .....	72
4.2. Penggunaan Algoritma Dalam Pengecekan <i>Similarity</i> .....	74
4.2.1. Konsep Implementasi Algoritma Rabin Karp.....	74
4.2.2. Hasil <i>Similarity</i> .....	78
4.3. Hasil Uji Coba Sistem .....	79
4.3.1. Hasil Uji Coba Halaman <i>Login</i> .....	79
4.3.2. Hasil Uji Coba Halaman Admin .....	80
4.3.3. Hasil Uji Coba Halaman Dosen .....	83
4.3.4. Hasil Uji Coba Halaman Mahasiswa.....	85
BAB V PENUTUP .....	87
5.1. Kesimpulan .....	87
5.2. Saran.....	87
DAFTAR PUSTAKA .....	88
LAMPIRAN .....	91

## DAFTAR GAMBAR

<b>Gambar 2.1.</b> Arsitektur MVC .....	12
<b>Gambar 2.2.</b> Contoh <i>Hash</i> Rabin Karp .....	20
<b>Gambar 2.3.</b> Model <i>Waterfall</i> .....	23
<b>Gambar 3.1.</b> Tahapan Penelitian .....	26
<b>Gambar 3.2.</b> <i>Use Case Diagram</i> .....	31
<b>Gambar 3.3.</b> <i>Activity Diagram Login</i> .....	40
<b>Gambar 3.4.</b> <i>Activity Diagram</i> Konfigurasi.....	41
<b>Gambar 3.5.</b> <i>Activity Diagram</i> Tambah Proposal .....	43
<b>Gambar 3.6.</b> <i>Activity Diagram</i> Cek <i>Similarity</i> .....	44
<b>Gambar 3.7.</b> <i>Activity Diagram</i> Admin Tolak Proposal.....	45
<b>Gambar 3.8.</b> <i>Activity Diagram</i> Bagi Tugas <i>Review</i> .....	46
<b>Gambar 3.9.</b> <i>Activity Diagram</i> Penerimaan Proposal .....	48
<b>Gambar 3.10.</b> <i>Activity Diagram</i> <i>Review</i> Proposal.....	50
<b>Gambar 3.11.</b> <i>Activity Diagram</i> Lewat Batas <i>Review</i> .....	51
<b>Gambar 3.12.</b> Implementasi Algoritma Rabin Karp .....	56
<b>Gambar 4.1.</b> Tampilan Halaman <i>Login</i> .....	59
<b>Gambar 4.2.</b> Tampilan Halaman Utama Admin .....	60
<b>Gambar 4.3.</b> Tampilan Halaman Konfigurasi.....	61
<b>Gambar 4.4.</b> Tampilan Menu Proposal.....	61
<b>Gambar 4.5.</b> Tampilan Aksi Tolak.....	62

<b>Gambar 4.6.</b> Tampilan Aksi Dosen .....	62
<b>Gambar 4.7.</b> Tampilan Aksi Detail .....	63
<b>Gambar 4.8.</b> Tampilan Menu Rekap Nilai.....	64
<b>Gambar 4.9.</b> Tampilan Menu Statistik .....	65
<b>Gambar 4.10.</b> Tampilan Menu Penerimaan .....	66
<b>Gambar 4.11.</b> Tampilan Menu Proposal.....	66
<b>Gambar 4.12.</b> Tampilan Aksi Review .....	67
<b>Gambar 4.13.</b> Tampilan Menu Seminar Proposal.....	68
<b>Gambar 4.14.</b> Tampilan Aksi Nilai Seminar .....	68
<b>Gambar 4.15.</b> Tampilan Menu Proposal.....	69
<b>Gambar 4.16.</b> Tampilan Tambah Proposal .....	70
<b>Gambar 4.17.</b> Tampilan Setelah <i>Input</i> Proposal .....	70
<b>Gambar 4.18.</b> Tampilan Hasil Pengecekan <i>Similarity</i> .....	71
<b>Gambar 4.19.</b> Tampilan Lolos Tahap <i>Similarity</i> .....	71
<b>Gambar 4.20.</b> Tampilan Aksi Detail Proposal Terkirim .....	72
<b>Gambar 4.21.</b> Tampilan Aksi Detail Proposal Ditolak .....	73
<b>Gambar 4.22.</b> Tampilan Aksi Detail .....	73
<b>Gambar 4.21.</b> Hasil <i>Similarity</i> .....	78
<b>Gambar 4.22.</b> Konfigurasi Proposal Threshold .....	79

## DAFTAR TABEL

<b>Tabel 2.1.</b> Contoh <i>K-Gram</i> .....	19
<b>Tabel 2.2.</b> Contoh <i>Rolling Hash</i> .....	20
<b>Tabel 3.1.</b> Identifikasi Aktor.....	29
<b>Tabel 3.2.</b> Identifikasi Diagram <i>Use Case</i> .....	29
<b>Tabel 3.3.</b> <i>Use Case</i> Skenario Konfigurasi .....	32
<b>Tabel 3.4.</b> <i>Use Case</i> Skenario Lihat Proposal .....	32
<b>Tabel 3.5.</b> <i>Use Case</i> Skenario Bagi Tugas <i>Review</i> .....	33
<b>Tabel 3.6.</b> <i>Use Case</i> Skenario Lihat Hasil <i>Review</i> .....	34
<b>Tabel 3.7.</b> <i>Use Case</i> Skenario Lihat Rekap Nilai .....	34
<b>Tabel 3.8.</b> <i>Use Case</i> Skenario Lihat Statistik .....	35
<b>Tabel 3.9.</b> <i>Use Case</i> Skenario Lihat Tugas <i>Review</i> .....	35
<b>Tabel 3.10.</b> <i>Use Case</i> Skenario Mereview Proposal .....	36
<b>Tabel 3.11.</b> <i>Use Case</i> Skenario Penilaian Seminar Proposal .....	37
<b>Tabel 3.12.</b> <i>Use Case</i> Skenario Input Proposal .....	38
<b>Tabel 3.13.</b> <i>Use Case</i> Skenario Mengecek <i>Similarity</i> .....	38
<b>Tabel 3.14.</b> <i>Use Case</i> Skenario <i>Login</i> .....	39
<b>Tabel 3.15.</b> <i>User</i> .....	52
<b>Tabel 3.16.</b> Konfigurasi.....	53
<b>Tabel 3.17.</b> Proposal.....	53
<b>Tabel 3.18.</b> <i>Review</i> .....	54
<b>Tabel 3.19.</b> Seminar .....	54

<b>Tabel 4.1.</b> Teks Masukan.....	74
<b>Tabel 4.2.</b> <i>Text Preprocessing</i> .....	74
<b>Tabel 4.3.</b> Hasil <i>Text Preprocessing</i> .....	75
<b>Tabel 4.4.</b> Proses Algoritma Rabin Karp .....	75
<b>Tabel 4.5.</b> Pengujian Black Box Pada Halaman Login.....	79
<b>Tabel 4.6.</b> Pengujian Black Box Pada Menu Konfigurasi Halaman Admin .....	80
<b>Tabel 4.7.</b> Pengujian Black Box Pada Menu Proposal Halaman Admin .....	81
<b>Tabel 4.8.</b> Pengujian Black Box Pada Menu Rekap Nilai Halaman Admin .....	83
<b>Tabel 4.9.</b> Pengujian Black Box Pada Menu Penerimaan Halaman Dosen .....	83
<b>Tabel 4.10.</b> Pengujian Black Box Pada Menu Proposal Halaman Dosen .....	84
<b>Tabel 4.11.</b> Pengujian Black Box Pada Menu Seminar Halaman Dosen.....	85
<b>Tabel 4.12.</b> Pengujian Black Box Pada Halaman Mahasiswa .....	85

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Pada era perkembangan teknologi saat ini, sebuah informasi sangat dibutuhkan dalam kehidupan manusia misalnya untuk menambah ilmu pengetahuan, wawasan, serta memudahkan pekerjaan manusia sehari-hari. Teknologi informasi saat ini juga sudah banyak digunakan dalam berbagai bidang salah satunya dalam bidang pendidikan di suatu perguruan tinggi, yakni penggunaan sistem informasi dalam proses belajar mengajar, dalam proses administrasi maupun dalam proses sistem informasi lainnya yang dapat membantu serta memudahkan pihak kampus, dosen maupun mahasiswa.

Studi kasus pada Program Studi Teknik Informatika Universitas Hasanuddin menunjukkan bahwa telah menerapkan beberapa sistem informasi dalam membantu proses pekerjaan di kampus, namun belum secara keseluruhan. Masih terdapat beberapa pekerjaan yang belum menerapkan sistem informasi di dalamnya, salah satunya yaitu belum tersedia sistem informasi yang dapat membantu dalam proses pengumpulan proposal tugas akhir mahasiswa.

Pada proses pengumpulan proposal tugas akhir di Program Studi Teknik Informatika Unhas saat ini, masih menggunakan media email. Dimana, mahasiswa melakukan pengumpulan proposal tugas akhir ke email admin dan begitupun admin juga akan melakukan proses pembagian tugas review ke dosen-dosen menggunakan media email tersebut. Dengan masih menggunakan media email tersebut mengakibatkan pihak admin menjadi kesulitan, dikarenakan terdapat begitu banyak proposal tugas akhir mahasiswa yang masuk ke email admin tiap periode seminar proposal. Dari sekian banyak proposal yang masuk dari mahasiswa, admin diharuskan untuk membagikan satu persatu proposal tersebut ke email dosen yang di tugaskan untuk mengreview proposal. Kemudian, hasil dari review dosen tersebut dikirimkan kembali ke email admin lalu di forward ke email mahasiswa, sehingga mengharuskan pihak admin mencari email mahasiswa yang sesuai dengan proposal hasil review yang diberikan oleh dosen.

Untuk membantu institusi dalam mengelola proposal tugas akhir mahasiswa, maka pada penelitian ini akan dibangun sebuah sistem informasi berupa sistem penilaian kelayakan proposal tugas akhir mahasiswa. Dalam sistem informasi yang akan dibuat nantinya, untuk lebih menunjang kelayakan proposal akan dilengkapi dengan sistem deteksi plagiasi. Alasan dengan adanya sistem deteksi plagiasi ini karena pada proses pertukaran informasi saat ini terjadi begitu cepat dan sangat mudah diakses, adanya fasilitas internet membuat para pencari informasi menjadi dimanjakan. Dengan mudahnya mendapat informasi beberapa kalangan memanfaatkan untuk kepentingan pribadinya, salah satunya adalah menggunakan hasil karya orang lain tanpa izin seolah olah karya tersebut hasil karyanya, hal ini disebut dengan plagiarisme (Karp, 2015).

Belakangan ini plagiarisme semakin marak terjadi terutama di dunia pendidikan, salah satu contohnya yakni pada proses pembuatan proposal tugas akhir di kalangan mahasiswa. Maka bagi pihak perguruan tinggi dibutuhkan sebuah sistem yang dapat mendeteksi tindak plagiasi pada proposal tugas akhir mahasiswa agar dapat dengan mudah mengetahui kelayakan proposal tersebut. Berdasarkan hal yang telah dijabarkan diatas, maka pada penelitian ini akan dibangun sebuah Sistem Penilaian Kelayakan Proposal Tugas Akhir Dengan Sistem Deteksi Plagiasi.

## **1.2. Rumusan Masalah**

Berdasarkan latar belakang yang telah dipaparkan di atas, adapun masalah yang dapat dirumuskan yaitu sebagai berikut:

1. Bagaimana mengatasi kesulitan admin dalam mengelola proposal tugas akhir mahasiswa?
2. Bagaimana mendeteksi tingkat kemiripan proposal tugas akhir mahasiswa pada sistem yang akan dibuat?

### **1.3. Tujuan Penelitian**

Tujuan yang dicapai dalam penelitian ini adalah sebagai berikut:

1. Membuat sistem informasi berupa *website* yang dapat memudahkan dalam melakukan proses penilaian proposal tugas akhir.
2. Mengimplementasikan algoritma rabin karp untuk mengetahui tingkat kemiripan dari proposal tugas akhir yang terdapat pada sistem.

### **1.4. Manfaat Penelitian**

Setiap penyusunan skripsi atau penelitian diharapkan dapat menjadi salah satu bagian pengembangan bagi dunia teknologi informasi. Adapun manfaat yang diharapkan adalah sebagai berikut:

a. Manfaat bagi Penulis:

1. Sebagai tempat untuk mengimplementasikan ilmu pengetahuan yang diperoleh selama masa pendidikan.
2. Menambah keterampilan tentang bagaimana mendesain suatu aplikasi dan mengimplementasikan suatu algoritma ke dalam suatu sistem.

b. Manfaat terhadap Universitas:

Hasil dari sistem diharapkan dapat membantu pihak kampus dalam mengelola proposal tugas akhir mahasiswa, yakni proses pengumpulan proposal tugas akhir bagi mahasiswa, membantu admin dalam membagi tugas *review* untuk dosen, serta membantu dalam mengetahui tingkat kemiripan proposal mahasiswa.

c. Manfaat bagi Peneliti:

Sebagai referensi bagi pembaca yang berminat mengadakan penelitian lebih lanjut atau pengembangan lebih lanjut dari penelitian ini.

### **1.5. Batasan Masalah**

Agar berfokus pada masalah yang dibahas, maka batasan masalah yang diambil adalah sebagai berikut :

1. Sistem berbasis web ini hanya menyediakan sistem penilaian proposal tugas akhir mahasiswa berbasis *online* yakni *me-review* langsung pada sistem, tidak menyediakan *review* berbasis *offline*.
2. Dalam sistem berbasis web ini nantinya hanya dilakukan pengecekan kemiripan dengan proposal-proposal yang terdapat di *database* server lokal.
3. Lingkup penelitian adalah Teknik Informatika Universitas Hasanuddin.

### **1.6. Sistematika Penulisan**

Untuk memberikan gambaran singkat mengenai isi tulisan ini secara keseluruhan, maka akan diuraikan beberapa tahapan dari penulisan secara sistematis, yaitu:

#### **BAB I PENDAHULUAN**

Bab ini berisi latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika penulisan.

#### **BAB II TINJAUAN PUSTAKA**

Bab ini berisi pembahasan mengenai teori terkait hal-hal yang berhubungan dengan penelitian dan yang menunjang percobaan penelitian yang dilakukan.

#### **BAB III METODOLOGI PENELITIAN**

Bab ini berisi tentang tahapan penelitian mulai dari studi literatur, persiapan kebutuhan perancangan sistem, perancangan dan proses pembuatan sistem, penerapan algoritma rabin karp dan skenario pengujian sistem.

#### **BAB IV HASIL DAN PEMBAHASAN**

Bab ini berisi tentang hasil penelitian dan pembahasan terkait sistem yang telah dibuat.

#### **BAB V PENUTUP**

Bab ini berisi kesimpulan dari hasil penelitian yang telah dilakukan dan mencantumkan saran-saran untuk pengembangan sistem yang lebih lanjut.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1. Sistem Informasi**

Sistem informasi merupakan seperangkat fungsi operasional manajemen kepada yang mampu menghasilkan suatu keputusan yang tepat, cepat dan jelas yang merupakan suatu susunan yang disusun secara sistematis dan teratur dari jaringan-jaringan informasi yang menghubungkan setiap bagian dari suatu sistem, sehingga dimungkinkan diadakan komunikasi antar bagian fungsional. Sistem informasi adalah suatu sistem didalam suatu organisasi yang merupakan kombinasi dari orang-orang, fasilitas, teknologi, media, prosedur-prosedur dan pengendalian yang ditujukan untuk mendapatkan jalur komunikasi yang penting, memproses tipe transaksi rutin tertentu, memberi sinyal manajemen terhadap kejadian-kejadian internal dan eksternal yang penting dan menyediakan suatu dasar informasi untuk pengambilan keputusan yang baik (Filcha & Hayaty, 2019).

##### **2.1.1. Kemampuan Sistem Informasi**

Efraim Turban, McCean, dan James Waterbe, dalam buku *Information Technology for Management Making Connection for Strategies Advantages*, menyebutkan kemampuan sistem informasi sebagaimana berikut (Muzakhi, 2016):

1. Melakukan komputasi numerik bervolume besar dengan kecepatan tinggi.
2. Menyediakan komunikasi dalam organisasi atau antar organisasi yang murah dan cepat.
3. Menyimpan informasi dalam jumlah yang besar dalam ruang kecil, tetapi mudah diakses.
4. Memungkinkan pengaksesan informasi yang sangat banyak di seluruh dunia dengan cepat dan murah.
5. Meningkatkan efektivitas dan efisiensi orang-orang yang bekerja dalam kelompok pada suatu lokasi.
6. Menyajikan informasi dengan jelas yang menggugah pikiran manusia.

7. Mengoptimisasikan proses-proses bisnis yang semi otomatis dan tugas-tugas yang dikerjakan secara manual.
8. Mempercepat pengetikan dan penyuntingan.
9. Melaksanakan hal-hal di atas jauh lebih murah dari pada apabila dikerjakan secara manual.

## **2.2. Sistem Informasi Akademik**

Sistem informasi akademik merupakan sistem yang mengolah data dan melakukan proses kegiatan akademik yang melibatkan antara mahasiswa, dosen, administrasi akademik, penilaian dan data atribut lainnya. Sistem informasi akademik melakukan kegiatan administrasi akademik, melakukan proses-proses transaksi belajar-mengajar antara dosen dan mahasiswa, melakukan proses administrasi akademik yang baik menyangkut kelengkapan dokumen dan biaya yang muncul pada kegiatan registrasi ataupun kegiatan operasional harian administrasi akademik (Muzakhi, 2016).

Pemanfaatan sistem informasi di perguruan tinggi atau biasa disebut sebagai Sistem Informasi Akademik telah banyak digunakan dalam proses belajar mengajar, administrasi maupun dalam proses sistem informasi lainnya yang dapat membantu serta memudahkan pihak kampus, dosen maupun mahasiswa. Sistem informasi akademik dirancang sebagai alat penunjang berupa software aplikasi berbasis web untuk menunjang seluruh kegiatan perguruan tinggi yang mempunyai tujuan sebagai berikut :

- Adanya *database* yang terpusat dan terintegrasi.
- Tersedianya informasi untuk mendukung perencanaan dan pengambilan keputusan.
- Tersedianya layanan informasi bagi komunitas kampus seperti mahasiswa, dosen, staf tata usaha, pimpinan, maupun alumni.
- Sebagai salah satu pilar pembangunan perguruan tinggi berbasis teknologi informasi.

### **2.3. Website**

*Website* merupakan sistem dengan standar yang diterima secara universal untuk menelusuri, memformat dan menyimpan informasi melalui arsitektur klien atau server. Web bisa menerima semua jenis informasi digital, termasuk teks, hipermedia, grafis dan suara. Web didasari oleh hiperteks standar yang disebut HyperText Markup Language (HTML), yang memformat dokumen dan memadukan link hiperteks dinamis ke dokumen-dokumen lainnya yang disimpan di dalam komputer yang sama atau berbeda (Nugroho, 2014).

#### **2.3.1. Kategori Website**

Web dapat dibagi menjadi dua kategori, yaitu (Nugroho, 2014):

##### **1. Web Statis**

Web Statis adalah web yang content atau isinya tidak berubah-ubah. Maksudnya adalah isi dari dokumen yang ada di web tersebut tidak dapat diubah secara mudah. Ini dikarenakan karena script yang digunakan untuk membuat web statis tidak mendukung untuk mengubah isi dokumen. Script yang digunakan untuk membuat web statis ini seperti HTML dan CSS. Maka dari itu untuk perubahan isi dokumen pada web statis harus mengubah isi file HTML atau CSS tersebut. Web statis ini biasanya digunakan oleh website yang menggunakan HTML, *Web Search Engine* atau *Web Company Profile*.

##### **2. Web Dinamis**

Web dinamis adalah web yang content atau isinya dapat berubah-ubah setiap saat. Karena dalam teknologi pembuatan web dinamis sudah dirancang semudah mungkin bagi user yang menggunakan web dinamis tersebut. Untuk perubahan content atau isi dokumen dalam sebuah web dinamis lebih mudah daripada web statis. Ketika akan mengubah content atau isi dari sebuah web dinamis hanya perlu masuk ke bagian administrator web yang telah disediakan oleh script web dinamis.

Untuk membuat web dinamis diperlukan beberapa komponen yaitu:

a. *Client Side Scripting*

Merupakan jenis script yang pengolahannya (penerjemahannya) dilakukan di sisi client. Contoh *Client Side Scripting* : HTML (*Hypertext Markup Language*), Javascript, XML (*Extensible Markup Language*), CSS (*Cascading Style Sheet*). *Client Side Scripting* merupakan script yang digunakan untuk membuat halaman web statis.

b. *Server Side Scripting*

Merupakan *script* yang pengolahannya di sisi server. Server yang dimaksud disini adalah sebuah komponen yang biasa disebut web server. Contoh *Server Side Scripting* : PHP (*Hypertext Preprocessor*), ASP (*Active Server Pages*), JSP (*Java Server Pages*). *Server Side Scripting* digunakan untuk membuat halaman web dinamis. Hal ini disebabkan karena *Server Side Scripting* mempunyai kelebihan, diantaranya : mampu berinteraksi dengan banyak perangkat lunak basis data, mampu mengolah sumber daya yang terdapat di sistem operasi dan perangkat keras komputer, mampu dijalankan di semua sistem operasi (*multiplatform*), aman karena *script*-nya diproses di server yang tidak dapat dilihat dari sisi *client*.

#### **2.4. Bahasa Pemrograman**

Bahasa pemrograman, atau sering diistilahkan juga dengan bahasa komputer atau bahasa pemrograman komputer, adalah instruksi standar untuk memerintah komputer. Bahasa pemrograman ini merupakan suatu himpunan dari aturan sintaks dan semantik yang dipakai untuk mendefinisikan program komputer. Bahasa ini memungkinkan seorang *programmer* dapat menentukan secara persis data mana yang akan diolah oleh komputer, bagaimana data ini akan disimpan atau diteruskan, dan jenis langkah apa secara persis yang akan diambil dalam berbagai situasi.

Bahasa Pemrograman merupakan sebuah instruksi standar untuk memerintah komputer agar menjalankan fungsi tertentu. Bahasa pemrograman ini merupakan suatu himpunan dari aturan sintaks dan semantik yang dipakai untuk mendefinisikan program komputer. Bahasa ini memungkinkan seorang programmer dapat menentukan secara persis data mana yang akan diolah oleh komputer, bagaimana data ini akan disimpan atau diteruskan, dan jenis langkah apa secara persis yang akan diambil dalam berbagai situasi (Saragih, 2018).

## 2.5. HTML

HTML (*Hypertext Markup Language*) adalah sebuah bahasa dasar untuk web scripting yang bersifat *client side* yang digunakan untuk menampilkan informasi dalam bentuk teks, grafik serta multimedia dan digunakan juga untuk menghubungkan antar tampilan web *page*. Pemrograman web pada dasarnya digunakan untuk mendesain halaman situs web yang dinamis dan interaktif. Suatu halaman web dibangun dengan menggunakan bahasa HTML dan perlu di digaris bawahi bahwa HTML bukan bahasa pemrograman, HTML hanyalah melakukan *markup* (penandaan) pada suatu teks sehingga akan menghasilkan format tertentu apabila dibaca oleh browser. Itu sebabnya HTML hanya bersifat statis. Pemrograman web didesain untuk dapat mengubah-ubah *output* HTML tersebut sehingga tidak lagi statis melainkan dinamis (Saragih, 2018).

### 2.5.1. Struktur Dasar HTML

Semua file HTML harus memiliki beberapa *tag* HTML standar. Dimana *tag* HTML ini sendiri terdiri dari kurung buka siku “<”, nama *tag*, dan kurung tutup siku “>”. *Tag* HTML biasanya memiliki penutup, contoh: <HEAD> ditutup dengan </HEAD>, penutup ini menandakan akhir dari pernyataan (*statement*) *tag* tersebut. *Tag* penutup persis seperti *tag* pembuka, hanya ditambahkan garis miring “/” didepannya. Semua dokumen HTML terdiri dari *head* dan *body*. *Head* terdiri dari *title*, kemudian *body* terdiri dari tulisan dan gambar atau animasi yang ingin ditampilkan di layar, seperti jenis paragraph, list, warna, dan sebagainya. *Tag* yang penting adalah <html>, <head> dan <body> (Zulkifli, 2011).

Berikut sedikit penjelasan mengenai beberapa *tag* HTML:

1. HTML

*Tag* ini memberitahu *browser* bahwa *file* yang dibaca adalah *file* HTML. *Tag* ini diletakkan pada baris paling awal dari dokumen HTML.

2. HEAD

*Tag* ini adalah bagian paling atas dari dokumen HTML. Title adalah salah satu contoh *tag* yang dapat disertakan pada *tag* ini.

3. TITLE

Berisi judul halaman web yang dibuat, dan mengidentifikasi isi dari halaman tersebut secara global. Title biasanya ditampilkan pada *title bar* di *browser*. Title juga digunakan sebagai penentu halaman web kita pada beberapa *search engine*.

4. BODY

Merupakan bagian kedua dari sebuah dokumen HTML dan biasanya memiliki bagian yang paling banyak dari sebuah dokumen HTML apabila dibandingkan dengan head. Terdiri dari isi utama halaman web tersebut. *Tag-tag* HTML yang akan dijelaskan selanjutnya diletakkan dalam body.

## 2.6. PHP

PHP (*Hypertext Preprocessor*) adalah bahasa *scripting language* yang dirancang khusus untuk pembuatan web. PHP adalah *tool* yang digunakan untuk membuat halaman web dinamis, kaya akan fitur yang membuat perancangan web dan pemrograman lebih mudah. PHP merupakan bahasa pemrograman *server-side* karena di proses di komputer server. Awalnya, PHP adalah singkatan dari *Personal Home Page*. PHP digunakan untuk membuat halaman web pribadi. Dalam beberapa tahun perkembangannya, PHP berubah menjadi bahasa pemrograman yang sangat luar biasa dan banyak digunakan oleh *website-website* besar seperti wikipedia, facebook, joomla, wordpress, dan lain- lain. Saat ini PHP merupakan singkatan dari PHP: *Hypertext Preprocessor*. PHP berbasis *freeware* dan *open source*, artinya kita bisa mengubah nya sesuai kemauan kita (Saragih, 2018).

Bahasa pemrograman PHP sangat sering dipakai orang karena kemudahan dalam mempelajari dan menerapkannya. Dilansir dari W3Tech, PHP merupakan bahasa pemrograman *server-side* yang paling banyak dipakai. PHP dapat memanfaatkan *database* untuk menghasilkan halaman web yang dinamis. Sistem manajemen *database* yang sering digunakan bersama PHP adalah MySQL. Namun PHP juga mendukung sistem manajemen *Database Oracle, Microsoft Acces, Interbase, d-Base, PostgreSQL* dan sebagainya.

### **2.6.1. Keunggulan PHP**

Bahasa pemrograman PHP (*Hypertext Preprocessor*) memiliki beberapa keunggulan ialah sebagai berikut (Zulkifli, 2011):

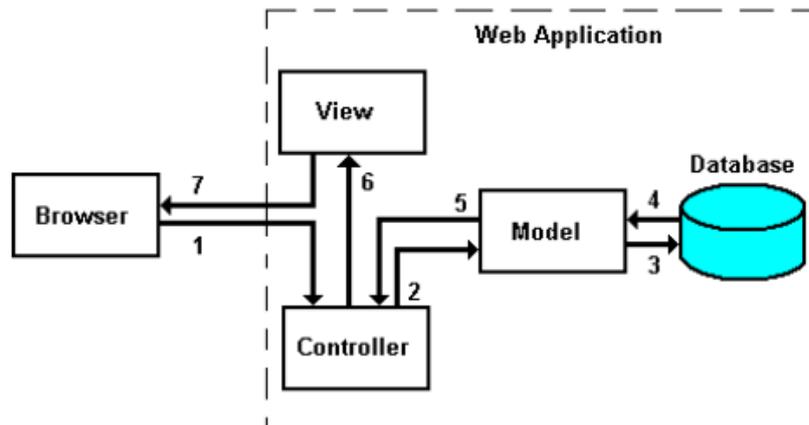
1. Dapat menjadikan web yang kita buat menjadi dinamis.
2. PHP dapat berjalan dalam web server yang berbeda dan dalam Sistem Operasi yang berbeda.
3. PHP diterbitkan secara gratis.
4. PHP dapat berjalan pada web server apapun, seperti: Apache, IIS, Microsoft personal Web Server, dan lain sebagainya.

### **2.7. CodeIgniter**

CodeIgniter merupakan sebuah *framework* yang dibuat dengan menggunakan bahasa PHP, yang dapat digunakan untuk pengembangan web secara cepat. Menurut Novianto (2016) menjelaskan bahwa codeigniter adalah sebuah *web application framework* yang bersifat *open source* digunakan untuk membangun aplikasi web. Tujuan utama pengembangan codeigniter adalah untuk membantu *developer* dalam pengerjaan aplikasi yang lebih cepat daripada menulis semua *code* dari awal dan codeigniter merupakan salah satu *framework* php tercepat yang ada saat ini. Codeigniter memiliki banyak fitur yang membantu para pengembang PHP untuk dapat membuat aplikasi web secara mudah dan cepat. Codeigniter mengizinkan para pengembang untuk menggunakan framework secara parsial maupun secara keseluruhan. Codeigniter menggunakan pola design *Model-View-Controller* (MVC) (S. Suci Romadhon, 2019).

### 2.7.1. MVC (*Model View Controller*)

Codeigniter menggunakan pola design *Model-View-Controller* (MVC) sehingga satu file tidak terlalu berisi banyak kode. Arsitektur MVC dapat dilihat pada Gambar 2.1. (Herdianto, 2014).



**Gambar 2.1.** Arsitektur MVC

Berikut penjelasan fungsi dari masing-masing bagian pada arsitektur mvc:

a. *Model*

Bertanggung jawab untuk melakukan pengelolaan data dalam basis data, di dalamnya biasa dituliskan perintah untuk mengambil, mengubah, menghapus, dan menambahkan data;

b. *View*

*View* merupakan tempat untuk meletakkan apa yang akan ditampilkan di halaman perambah (*browser*), sebuah berkas *view* umumnya berisi kode bahasa pemrograman sisi klien (*client-side scripting*);

c. *Controller*

*Controller* merupakan pengatur utama hubungan antara *model*, *view*, dan juga sumber daya lain yang tersedia, sumber daya ini diperoleh dari kelompok atau tipe kelas yang dapat disebut dengan elemen *framework* CodeIgniter.

### 2.7.2. Keunggulan CodeIgniter

Adapun keunggulan dari Codeigniter sebagai berikut:

1. Codeigniter adalah *framework* yang bersifat *free* dan *open source*.
2. Codeigniter memiliki ukuran kecil dibandingkan dengan *framework* lain.
3. Aplikasi yang dibuat menggunakan codeigniter bisa berjalan cepat.
4. Codeigniter menggunakan pola design *Model-View-Controller* (MVC) sehingga satu file tidak terlalu berisi banyak kode.
5. Codeigniter dapat diperluas sesuai dengan kebutuhan.
6. Codeigniter terdokumentasi dengan baik.
7. Informasi tentang pustaka kelas dan fungsi yang disediakan oleh codeigniter dapat diproses melalui dokumentasi yang disertai di dalam paket distribusinya.

### 2.8. Database

*Database* adalah sebuah tempat penyimpanan yang besar dimana terdapat kumpulan data yang tidak hanya berisi data operasional tetapi juga deskripsi data. Seperti yang disampaikan oleh (Connolly & Begg, 2010) bahwa database adalah kumpulan data yang saling terhubung secara logis dan deskripsi dari data tersebut, dirancang untuk menemukan informasi yang dibutuhkan oleh sebuah organisasi. Dalam merancang *database*, salah satu hal yang perlu diperhatikan adalah efisiensi. Banyaknya data yang redundansi dapat mengurangi efisiensi pada database sehingga perlu dilakukan normalisasi. *Database* ini digunakan tidak hanya oleh satu orang maupun satu departemen, *database* dapat digunakan oleh seluruh departemen dalam perusahaan. *Database* ini akan menjadi sumber data yang digunakan secara bersama dalam perusahaan. Hal ini kembali ditegaskan oleh (Connolly & Begg, 2010) *database* tidak lagi dimiliki oleh satu departemen tetapi sumber perusahaan yang saling berbagi. Dengan hanya *database* saja tidak cukup, diperlukan *Database Management System* (DBMS) untuk dapat menggunakan *database*.

## 2.9. MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: *database management system*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL adalah *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL. Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian *database*, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Keandalan suatu sistem *database* (DBMS) dapat diketahui dari cara kerja *optimizer*-nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh *user* maupun program-program aplikasinya. Sebagai *database server*, MySQL dapat dikatakan lebih unggul dibandingkan *database server* lainnya dalam *query* data. Hal ini terbukti untuk *query* yang dilakukan oleh *single user*, kecepatan *query* MySQL bisa sepuluh kali lebih cepat dari PostgreSQL dan lima kali lebih cepat dibandingkan Interbase (Standisyah & Restu, 2017).

### 2.9.1. Keunggulan MySQL

Sebagai *database server* yang memiliki konsep *database* modern, MySQL memiliki banyak keunggulan. Berikut ini beberapa keunggulan yang dimiliki oleh MySQL (Dewi Maharani, 2017):

1. *Portability* MySQL dapat berjalan stabil pada berbagai system operasi seperti Windows, Linux, FreeBSD, Mac OS X Server, dan lain-lain.
2. *Open Source* MySQL didistribusikan secara *Open Source*, sehingga dapat digunakan secara bebas.
3. *Multi-user* MySQL dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami masalah. Hal ini memungkinkan sebuah *database server* MySQL dapat diakses *client* secara bersamaan.

## **2.10. Xampp**

XAMPP adalah perangkat lunak bebas, yang mendukung banyak system operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (localhost), yang terdiri atas program Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMP merupakan singkatan dari X (empat system operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU (*General Public License*) dan bebas, merupakan web server yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis (Priyanti & Iriani, 2013).

## **2.11. PhpMyAdmin**

PhpMyAdmin adalah sebuah aplikasi/perangkat lunak bebas (*opensource*) yang ditulis dalam bahasa pemrograman PHP yang digunakan untuk menangani administrasi *database* MySQL melalui jaringan lokal maupun internet. phpMyAdmin mendukung berbagai operasi MySQL, diantaranya (mengelola basis data, tabel-tabel, bidang (*fields*), relasi (*relations*), indeks, pengguna (*users*), perijinan (*permissions*), dan lain- lain. Perbedaan phpMyAdmin dengan MySQL terletak pada fungsi. PhpMyAdmin merupakan alat untuk memudahkan dalam mengoperasikan *database* MySQL, sedangkan MySQL adalah *database* tempat penyimpanan data. Phpmyadmin sendiri digunakan sebagai alat untuk mengolah atau mengatur data pada MySQL (Standisyah & Restu, 2017).

## **2.12. Sublime Text**

Sublime text adalah text editor berbasis Python, sebuah text editor yang elegan, kaya fitur, cross platform, mudah dan simple yang cukup terkenal dikalangan *developer* (pengembang) dan desainer. Sublime Text digunakan sebagai editor dari bahasa pemrograman PHP dalam melakukan pengelolaan konten di dalam aplikasi server (Pradiatiningtyas, 2017).

### 2.13. Plagiarisme

Plagiarisme atau biasa disebut sebagai plagiat menurut Kamus Besar Bahasa Indonesia adalah penjiplakan atau pengambilan karangan, pendapat, dan sebagainya dari orang lain dan menjadikannya seolah karangan dan pendapat sendiri (KBBI, 1997:775). Sedangkan menurut Peraturan Menteri Pendidikan Nasional Republik Indonesia No. 17 Tahun 2010, plagiat adalah perbuatan secara sengaja maupun 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 (Muhammad & Ramadhan, 2019).

Plagiarisme merupakan praktik penyalahgunaan hak kekayaan intelektual milik orang lain dan karya itu diakui tidak sah sebagai akibat dari pekerjaan pribadi. Studi empiris yang dilakukan oleh Hutton and French in Hartanto mengemukakan bahwa bahwa faktor-faktor yang menyebabkan plagiarisme adalah kemalasan mereka sendiri, karena mereka merasa stres, memiliki keyakinan bahwa perilakunya tidak akan diketahui, dan perilakunya bukanlah hal yang salah untuk dilakukan atau berbahaya (Filcha & Hayaty, 2019).

#### 2.13.1. Ambang Batas (*Threshold*) Plagiarisme

Menurut Sastroasmoro, *threshold* plagiarisme merupakan nilai ambang batas untuk menentukan apakah dokumen tersebut termasuk kedalam tindak plagiarisme atau tidak, terdapat 3 klasifikasi dalam menentukan ambang batas plagiarisme, yaitu:

1. Plagiarisme ringan : < 30%
2. Plagiarisme sedang : < 30% - 70%
3. Plagiarisme berat : > 70%

Namun angka-angka tersebut dapat disesuaikan dan ditentukan sendiri oleh universitas atau fakultas masing-masing. Metode untuk mendeteksi plagiarisme dapat diklasifikasikan menjadi tiga metode yaitu metode perbandingan teks lengkap, metode dokumen *fingerprinting* dan metode kesamaan kata kunci (Alamsyah, 2017).

#### **2.14. Text Preprocessing**

*Text Preprocessing* adalah proses yang sering digunakan untuk melakukan *text mining*. Tujuan dari *text preprocessing* adalah untuk mengembalikan teks menjadi bahasa yang alami (Filcha & Hayaty, 2019). Secara umum dalam tahap-tahap *text preprocessing*, yaitu:

1. *Case Folding*

*Case Folding* adalah proses untuk memanipulasi teks, semua masukan teks akan diubah menjadi huruf kecil.

2. *Tokenizing*

*Tokenizing* adalah proses pemisahan kata berdasarkan susunan kata. Hasil dari pemisahan kata disebut token.

3. *Punctuation Removal*

*Punctuation Removal* adalah proses menghapus karakter-karakter unik seperti karakter tanda seru, tanda tanya, tanda koma dan sebagainya.

4. *Stopword Removal*

*Stopword Removal* adalah kata-kata yang tidak deskriptif yang dapat dibuang. Contoh *stopword* adalah “yang”, “dan”, “di”, “dari” dan sebagainya. *Stopword Removal* adalah proses penghapusan kata yang tidak relevan dalam teks.

5. *Stemming*

*Stemming* adalah proses untuk merubah kata menjadi kata dasar.

#### **2.15. Algoritma Rabin Karp**

*String matching* atau pencocokan *string* adalah subjek yang penting dalam kaitannya 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 plagiarisme karya tulis. Salah satu yang termasuk dalam algoritma *string matching* ialah Algoritma Rabin Karp (Sunyoto, 2013).

Algoritma Rabin Karp merupakan sebuah algoritma pencarian *string* yang diciptakan oleh Richard M. Karp dan Michael O. Rabin yang menggunakan fungsi *hash* untuk menemukan kemunculan sebuah pola pada teks. Algoritma Rabin Karp ini merupakan algoritma pencarian kata yang mencari sebuah pola berupa *substring* dalam sebuah teks menggunakan *hashing* dan algoritma ini sangat efektif untuk pencocokan kata dengan pola banyak. Salah satu aplikasi praktis dari algoritma Rabin Karp adalah dalam pendeteksian plagiarisme (Hidayat, 2016).

Algoritma Rabin Karp memiliki kompleksitas sebesar  $O(m+n)$  untuk mencari teks dengan panjang  $n$  dan pola dengan panjang  $m$ . Dalam kasus terburuk, pencari pola pada teks tersebut akan memiliki kompleksitas sebesar  $O(mn)$ . Algoritma ini menggunakan fungsi hash untuk membandingkan pola dan teks yang disediakan. Oleh karena itu, kecepatan eksekusi program yang menggunakan algoritma ini juga ditentukan berdasarkan fungsi *hash* yang digunakan (Muhammad & Ramadhan, 2019).

Langkah-langkah dalam algoritma Rabin Karp (Hidayat, 2016):

1. Menghilangkan tanda baca dan mengubah ke teks sumber dan kata yang ingin dicari menjadi kata-kata tanpa huruf kapital.
2. Membagi teks ke dalam gram-gram yang ditentukan nilai *k-gram* nya.
3. Mencari nilai *hash* dengan fungsi *rolling hash* dari tiap gram yang terbentuk.
4. Mencari nilai *hash* yang sama antara 2 teks.
5. Menentukan persamaan 2 buah teks dengan persamaan *Dice's Similarity Coefficient*.

### **2.15.1. K-Gram**

*K-Gram* adalah rangkaian *terms* dengan panjang  $k$ . Kebanyakan yang digunakan sebagai *terms* adalah kata. Metode *k-gram* ini digunakan untuk mengambil potongan-potongan karakter huruf sejumlah  $k$  dari sebuah kata yang secara kontinuitas dibaca dari teks sumber hingga akhir dari dokumen (Putra et al., 2015).

Berikut contoh *K-Gram* dengan nilai  $k = 4$  (Filcha & Hayaty, 2019) dapat dilihat pada Tabel 2.1.

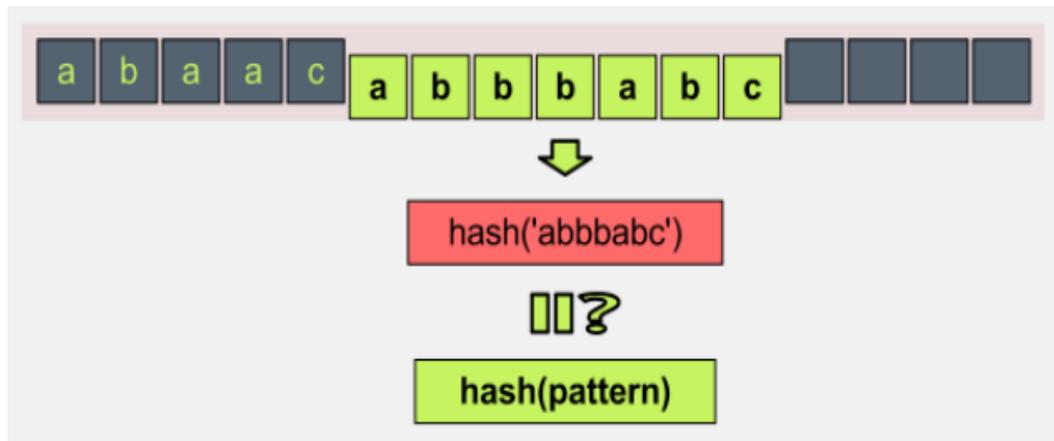
**Tabel 2.1.** Contoh *K-Gram*

Kalimat :	komputer adalah perangkat elektronik
Preprocessing :	Komputerperangkatelektronik
K-Gram{4} :	{komp} {ompu} {mput} {pute} {uter} {terp} {erpe} {rper} {pera} {eran} {rang} {angk} {ngka} {gkat} {kate} {atel} {tele} {elek} {lekt} {ektr} {ktro} {tron} {roni} {onik}

### 2.15.2. Hashing

*Hashing* adalah suatu cara untuk mentransformasi sebuah *string* menjadi suatu nilai yang unik dengan panjang tertentu (*fixed-length*) yang berfungsi sebagai penanda *string* tersebut. Fungsi untuk menghasilkan nilai ini disebut fungsi *hash*, sedangkan nilai yang dihasilkan disebut nilai *hash*. Misalnya penggunaan *hashing* dalam pencarian data pada *database*. Apabila tidak di-*hash*, pencarian akan dilakukan karakter per karakter pada nama-nama yang panjangnya bervariasi dan ada 26 kemungkinan pada setiap karakter. Namun pencarian akan menjadi lebih efisien setelah di-*hash* karena hanya akan membandingkan empat digit angka dengan cuma 10 kemungkinan setiap angka (Putra et al., 2015).

Algoritma Rabin Karp menentukan nilai *hash* berdasarkan kata yang sama (Gambar 2.1). Akan tetapi ada dua masalah yang timbul dalam menentukan nilai *hash*. Pertama, ada begitu banyak *string* yang berbeda dalam kalimat tertentu, permasalahan ini dapat dipecahkan dengan meng-*assign* beberapa *string* dengan *hash value* yang sama. Kedua, belum tentu *string* yang mempunyai *hash value* yang sama cocok untuk mengatasinya maka untuk setiap *string* yang di-*assign* dilakukan pencocokan *string* secara *Brute-Force* (Putri et al., 2017).



**Gambar 2.2.** Contoh *Hash* Rabin Karp

Proses pengubahan menjadi nilai *hash* menggunakan fungsi *rolling hash*, persamaan *rolling hash* dapat dilihat pada Persamaan 2.1 (Filcha&Hayaty, 2019).

$$H(c_1 \dots c_k) = (c_1 * b^{(k-1)} + c_2 * b^{(k-2)} + \dots + c_{(k-1)} * b^k + c_k) \text{mod } q \quad (2.1)$$

Keterangan:

H : *substring*

c : nilai ascii per-karakter

b : konstan bilangan prima

k : banyak karakter

q: modulo bilangan prima

Berikut contoh perhitungan *rolling hash* terhadap *substring* makan dengan nilai *K-Gram* 4 dapat dilihat pada Tabel 2.2.

**Tabel 2.2.** Contoh *Rolling Hash*

Atribut	Nilai Array
<i>Rolling hash</i> pertama	<p>[0] =&gt; maka</p> <p>m=109, a = 97, k=107, a=97, basis=11, mod = 10007</p> $H=c_m * b^{(k-1)} + c_a * b^{(k-2)} + c_k * b^{(k-3)} + c_a * b^{(k-4)}$ $H=109 * 11^3 + 97 * 11^2 + 107 * 11^1 + 97 * 11^0$ $H=145079+11737+1177+97$ $H=158090 \text{ Mod } 10007$ $H= 7985$

Atribut	Nilai Array
Rolling hash kedua	<p>[1] =&gt; akan</p> <p><math>a = 97, k=107, a=97, n=110, \text{basis}=11, \text{mod} = 10007</math></p> $H=c_a * b^{(k-1)} + c_k * b^{(k-2)} + c_a * b^{(k-3)} + c_n * b^{(k-4)}$ $H=97 * 11^3 + 107 * 11^2 + 97 * 11^1 + 110 * 11^0$ $H=129107+12947+1067+110$ $H=143231 \text{ Mod } 10007$ $H= 3133$

### 2.16. Similarity

Konsep similarity sering terjadi terutama pada bidang ilmu pengetahuan. Zaka dalam Salmuasih, dan Sunyoto mengemukakan tiga macam teknik yang dibangun untuk menentukan nilai *similarity* (kemiripan) dokumen, sebagai berikut (Purba & Situmorang, 2017):

a. *Distance-based similarity measure*

*Distance-based similarity measure* mengukur tingkat kesamaan dua buah objek dari segi jarak geometris dari variabel-variabel yang tercakup di dalam kedua objek tersebut. Metode *Distance-based similarity* ini meliputi *Minkowski Distance, Manhattan/City block distance, Euclidean distance, Jaccard Distance, Dice's Coefficient, Cosine similarity, Levenshtein Distance, Hamming Distance, dan Soundex distance.*

b. *Feature-based similarity measure*

*Feature-based similarity measure* melakukan penghitungan tingkat kemiripan dengan merepresentasikan objek ke dalam bentuk *feature-feature* yang ingin diperbandingkan. *Feature-based similarity measure* banyak digunakan dalam melakukan pengklasifikasian atau pattern matching untuk gambar dan teks.

c. *Probabilistic-based similarity measure*

*Probabilistic-based similarity measure* menghitung tingkat kemiripan dua objek dengan merepresentasikan dua set objek yang dibandingkan dalam bentuk *probability*. *Kullback Leibler Distance* dan *Posterior Probability* termasuk dalam metode ini.

### 2.16.1. Pengukuran Nilai *Similarity*

Mengukur *similarity* (kemiripan) dan jarak antara dua entitas informasi adalah syarat inti pada semua kasus penemuan informasi, seperti pada *Information Retrieval* dan *Data Mining* yang kemudian dikembangkan dalam bentuk aplikasi, salah satunya adalah sistem deteksi plagiat (Sunyoto, 2013).

Untuk mengukur nilai *similarity* menggunakan *Dice's Similarity Coefficient* pendekatan *k-gram*, rumus matematisnya ditunjukkan pada persamaan 2.2 berikut:

$$S = \frac{2 * C}{(A+B)} \quad (2.2)$$

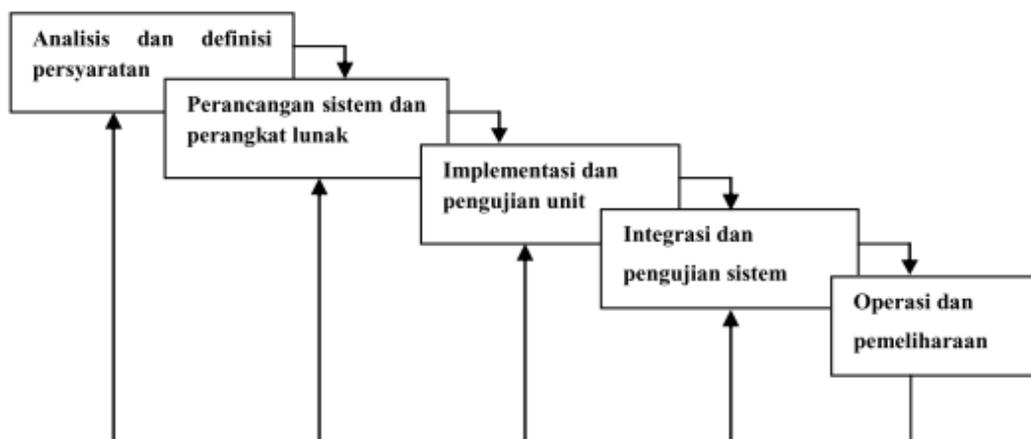
Dimana S adalah nilai *similarity*, A dan B adalah jumlah dari kumpulan *k-grams* dalam teks 1 dan teks 2. C adalah jumlah dari *k-grams* yang sama dari teks yang dibandingkan. Penggunaan ukuran *similarity* yang tepat tidak hanya meningkatkan kualitas pilihan informasi tetapi juga membantu mengurangi waktu dan biaya proses (Adi, 2018).

### 2.17. Metode Pengembangan Software

Langkah penyelesaian masalah tugas akhir ini sesuai dengan tahapan pengembangan perangkat lunak menggunakan model proses atau paradigma *waterfall*. Sebagai paradigma kehidupan klasik, model *waterfall* memiliki tempat penting dalam rekayasa perangkat lunak. Alasan dari penggunaan metode model *waterfall* dalam pembuatan sistem informasi penilaian kelayakan proposal ialah pada tahapan model *waterfall* mengambil kegiatan dasar yang digunakan dalam hampir semua pengembangan perangkat lunak, sehingga dapat lebih mudah untuk dipahami terlebih bila hanya digunakan dalam mengembangkan perangkat lunak yang tidak begitu besar dan kompleks.

Model *Waterfall* merupakan salah satu model proses perangkat lunak yang mengambil kegiatan proses dasar seperti spesifikasi, pengembangan, validasi, dan evolusi, dan merepresentasikannya sebagai fase-fase proses yang berbeda seperti analisis dan definisi persyaratan, perancangan perangkat lunak, implementasi, pengujian unit, integrasi sistem, pengujian sistem, operasi dan pemeliharaan (Samantha, D. P, 2011).

Tahap-tahap utama dari model *waterfall* dapat dilihat pada Gambar 2.3 (Samantha, D. P, 2011):



**Gambar 2.3.** Model *Waterfall*

#### 1. Analisis dan Definisi Persyaratan

Proses mengumpulkan informasi kebutuhan sistem/perangkat lunak melalui konsultasi dengan user system. Proses ini mendefinisikan secara rinci mengenai fungsi-fungsi, batasan dan tujuan dari perangkat lunak sebagai spesifikasi sistem yang akan dibuat.

#### 2. Perancangan Sistem dan Perangkat Lunak

Proses perancangan sistem ini difokuskan pada empat atribut, yaitu struktur data, arsitektur perangkat lunak, representasi antarmuka, dan detail (algoritma) prosedural. Yang dimaksud struktur data adalah representasi dari hubungan logis antara elemen-elemen data individual.

### 3. Implementasi dan Pengujian Unit

Pada tahap ini, perancangan perangkat lunak direalisasikan sebagai serangkaian program atau unit program. Kemudian pengujian unit melibatkan verifikasi bahwa setiap unit program telah memenuhi spesifikasinya.

### 4. Integrasi dan Pengujian Sistem

Unit program/program individual diintegrasikan menjadi sebuah kesatuan sistem dan kemudian dilakukan pengujian. Dengan kata lain, pengujian ini ditujukan untuk menguji keterhubungan dari tiap-tiap fungsi perangkat lunak untuk menjamin bahwa persyaratan sistem telah terpenuhi. Setelah pengujian sistem selesai dilakukan, perangkat lunak dikirim ke pelanggan/user.

### 5. Operasi dan Pemeliharaan

Tahap ini biasanya memerlukan waktu yang paling lama. Sistem diterapkan (di- install) dan dipakai. Pemeliharaan mencakup koreksi dari beberapa kesalahan yang tidak diketemukan pada tahapan sebelumnya, perbaikan atas implementasi unit sistem dan pengembangan pelayanan sistem, sementara persyaratan-persyaratan baru ditambahkan.

## 2.18. Penelitian Terkait

Berdasarkan hasil dan pembahasan pada penelitian berjudul “Implementasi Algoritma Rabin-Karp untuk Pendeteksi Plagiarisme pada Dokumen Tugas Mahasiswa”, maka peneliti dapat mengambil kesimpulan diantaranya Algoritma Rabin-Karp berhasil diimplementasikan pada sistem ini dengan 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. Algoritma yang digunakan pada sistem pendeteksi plagiarisme dokumen tugas mahasiswa tidak memiliki perbedaan persentase saat urutan perbandingan diubah (Filcha & Hayaty, 2019).

Kemudian pada penelitian berjudul “*K-Gram as a Determinant Of Plagiarism Level in Rabin Karp Algorithm*”, dimana penentuan nilai *K-Gram* sangat mempengaruhi persentase tingkat kemiripan plagiarisme. Setiap bahasa

memiliki nilai *K-Gram* yang berbeda. Untuk nilai *K-Gram* yang lebih tinggi cenderung memiliki tingkat kemiripan yang rendah, sedangkan nilai *K-Gram* yang lebih rendah meningkatkan persentase kemiripan. Tetapi ini tidak mutlak sebagai penentu keaslian dokumen. Metode ini dapat membantu analisis untuk melihat lebih banyak tentang teknik yang digunakan dalam membandingkan dokumen. Penentuan *K-Gram* membutuhkan pertimbangan yang benar sehingga nilainya akan lebih akurat. Metode ini hanya digunakan untuk membantu para ahli dalam menganalisis (Siahaan et al., 2017).

Pada penelitian lain berjudul “Pendeteksian Plagiarisme Menggunakan Algoritma Rabin-Karp Dengan Metode *Rolling Hash*”, menggunakan algoritma Rabin-Karp dengan metode *rolling hash* dari hasil pengujian 30 dokumen teks yang sudah dijelaskan pada bab sebelumnya menghasilkan tingkat akurasi yang terbesar yaitu 47.58%. Hasil persentase tersebut termasuk dalam kategori tingkat plagiat 15-50%, berarti menandakan dokumen tersebut termasuk plagiat tingkat sedang. Sedangkan tingkat akurasi yang terkecil yaitu 19.28%, berarti menandakan dokumen tersebut termasuk plagiat tingkat sedang (Priambodo, 2018).