

**SKRIPSI**

**SISTEM PENDETEKSI DINI PLAGIARISME PADA  
PROSES PENGAJUAN SKRIPSI MAHASISWA  
TEKNIK INFORMATIKA MENGGUNAKAN  
ALGORITMA *LEVENSHTEIN DISTANCE***



**M. HENDRA FEBIAWAN  
16.0504.0161**

**PROGRAM STUDI TEKNIK INFORMATIKA S1  
FAKULTAS TEKNIK  
UNIVERSITAS MUHAMMADIYAH MAGELANG**

**Januari, 2019**

## **HALAMAN PENEGASAN**

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

Nama : Muhamad Hendra Febiawan

NPM : 16.0504.0161

Magelang, 23 Januari 2019

**MUHAMAD HENDRA FEBIAWAN**  
**16.0504.0161**

## **PERNYATAAN KEASLIAN**

Yang bertanda tangan di bawah ini:

Nama : Muhamad Hendra Febiawan  
NPM : 16.0504.0161  
Program Studi : Teknik Informatika S1  
Fakultas : Teknik  
Alamat : Ds. Kalitidu RT 07 / RW 01 Kec. Kalitidu, Bojonegoro,  
Jawa Timur  
Judul Skripsi : SISTEM PENDETEKSI DINI PLAGIARISME PADA  
PROSES PENGAJUAN SKRIPSI MAHASISWA  
TEKNIK INFORMATIKA MENGGUNAKAN  
ALGORITMA *LEVENSHTTEIN DISTANCE*

Dengan ini menyatakan bahwa Skripsi ini merupakan hasil karya sendiri dan bukan merupakan plagiat dari hasil karya orang lain. Dan bila di kemudian hari terbukti bahwa karya ini merupakan plagiat, maka saya bersedia menerima sanksi administrasi maupun sanksi apapun.

Demikian surat pernyataan ini saya buat dengan penuh kesadaran dan sebenarnya serta penuh tanggungjawab.

Magelang, 23 Januari 2019

**MUHAMAD HENDRA FEBIAWAN**  
**16.0504.0161**

HALAMAN PENGESAHAN

**SKRIPSI**

**SISTEM PENDETEKSI DINI PLAGIARISME  
PADA PROSES PENGAJUAN SKRIPSI  
MAHASISWA TEKNIK INFORMATIKA  
MENGUNAKAN ALGORITMA *LEVENSHTAIN DISTANCE***

Disusun Oleh :

**MUHAMAD HENDRA FEBIAWAN**

**NPM. 16.0504.0161**

Telah dipertahankan di depan Dewan Penguji  
Pada Tanggal 23 Januari 2019

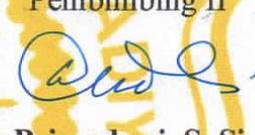
Susunan Dewan Penguji

Pembimbing I



Agus Setiawan, M. Eng.  
NIDN. 0617088801

Pembimbing II



Ardhin Primadewi, S. Si, M. TI.  
NIDN. 0619048501

Penguji I



Andi Widiyanto, S. Kom., M. Kom  
NIDN. 0623087901

Penguji II



Sunarni, M. T  
NIDN. 0620079101

Skripsi ini telah diterima sebagai salah satu persyaratan  
untuk memperoleh gelar Sarjana Komputer  
Tanggal, 23 Januari 2019

Dekan



Yun Arifatul Fatimah, ST., MT., Ph.D  
NIK. 987408139

## **KATA PENGANTAR**

Puji syukur dipanjatkan kehadirat Allah SWT, karena atas segala rahmat dan hidayah-Nya sehingga dapat diselesaikannya laporan skripsi ini dengan lancar. Skripsi ini disusun sebagai salah satu syarat untuk mencapai gelar Sarjana Komputer di Program Studi S1 Teknik Informatika Fakultas Teknik Universitas Muhammadiyah Magelang. Pada kesempatan ini diucapkan terimakasih sebesar-besarnya kepada:

1. Ir. Eko Muh. Widodo, MT selaku Rektor Universitas Muhammadiyah Magelang.
2. Yun Arifatul Fatimah, S.T., M.T., Ph.D. selaku Dekan Fakultas Teknik Universitas Muhammadiyah Magelang.
3. Agus Setiawan, M.Kom. selaku Ketua Program Studi Teknik Informatika S1 Universitas Muhammadiyah Magelang.
4. Agus Setiawan, M.Kom dan Ardhin Primadewi, S. Si., M. TI. selaku Dosen Pembimbing yang telah memberikan nasehat dan bimbingan dalam penyusunan skripsi ini.
5. Seluruh Dosen Fakultas Teknik Universitas Muhammadiyah Magelang yang telah memberikan ilmu dan pengetahuan yang bermanfaat.
6. Kedua orang tua dan keluarga yang telah memberikan dukungan baik secara moril maupun materi hingga terselesaikannya skripsi ini.
7. Eka Putri Aprilia yang telah memberikan doa, dukungan dan semangatnya.
8. Teman-teman Kampus Universitas Muhammadiyah Teknik Informatika yang telah banyak membantu dalam pengujian.

9. Semua pihak yang telah membantu namun tidak dapat disebutkan namanya satu persatu. Semoga Allah membalas kebaikan semua pihak yang telah membantu dan semoga skripsi ini dapat bermanfaat bagi semua pihak.

**Magelang, 23 Januari 2019**

**MUHAMAD HENDRA FEBIAWAN**  
**16.0504.0161**

## DAFTAR ISI

|  |                                     |
|--|-------------------------------------|
| SKRIPSI.....   | i                                   |
| HALAMAN PENGESAHAN.....                                    | ii                                  |
| PERNYATAAN KEASLIAN.....                                   | iii                                 |
| HALAMAN PENGESAHAN.....                                    | <b>Error! Bookmark not defined.</b> |
| KATA PENGANTAR .....                                       | v                                   |
| DAFTAR ISI.....  | vii                                 |
| INTISARI.....  | xii                                 |
| ABSTRACT.....  | xiii                                |
| BAB I.....   | 1                                   |
| A. Latar Belakang .....                                    | 1                                   |
| B. Rumusan Masalah .....                                   | 3                                   |
| C. Tujuan Penelitian .....                                 | 3                                   |
| D. Manfaat Penelitian .....                                | 4                                   |
| BAB II.....  | 5                                   |
| A. Penelitian Relevan.....                                 | 5                                   |
| B. Penjelasan Secara Teoritis Masing-Masing Variable ..... | 7                                   |
| 1. <i>Algoritma Levenshtein Distance</i> .....             | 7                                   |
| 2. Plagiarisme .....                                       | 8                                   |
| 3. Dokumen .....   | 9                                   |
| 4. Teks Mining.....  | 9                                   |
| 5. Basis Data ( <i>Database</i> ).....                     | 10                                  |
| 6. <i>Unified Modeling Language (UML)</i> .....            | 10                                  |
| 7. <i>Perl Hypertext Preprocessor (PHP)</i> .....          | 14                                  |
| 8. <i>Web Application Framework</i> .....                  | 14                                  |
| 9. Laravel.....  | 15                                  |
| 10. <i>Arsitektur Model View Controller (MVC)</i> .....    | 15                                  |
| C. Landasan Teori.....                                     | 15                                  |
| BAB III.....   | 17                                  |
| A. Analisis Sistem.....                                    | 17                                  |
| 1. <i>Algoritma Levenshtein Distance</i> .....             | 17                                  |
| 2. Sistem Yang Berjalan .....                              | 18                                  |
| 3. Sistem Yang Diusulkan .....                             | 20                                  |

|   |                                     |
|---|-------------------------------------|
| B. Perancangan Sistem .....                                 | 21                                  |
| 1. Perancangan UML.....                                     | 21                                  |
| C. Pemodelan Data .....                                     | 34                                  |
| 1. Deskripsi Data Objek .....                               | 34                                  |
| 2. EER ( <i>Enhanced Entity Relationship</i> ) Diagram..... | 36                                  |
| 3. Relasi dan Kardinalitas.....                             | 37                                  |
| D. Perancangan <i>Interface</i> .....                       | 39                                  |
| BAB IV .....  | <b>Error! Bookmark not defined.</b> |
| A. Implementasi Sistem .....                                | <b>Error! Bookmark not defined.</b> |
| 1. Kebutuhan Sistem.....                                    | <b>Error! Bookmark not defined.</b> |
| 2. Implementasi Tampilan .....                              | <b>Error! Bookmark not defined.</b> |
| B. Pengujian Sistem.....                                    | <b>Error! Bookmark not defined.</b> |
| 1. Pengujian Script Algoritma Levenshtein Distance.....     | <b>Error! Bookmark not defined.</b> |
| 2. Pengujian Sistem Pendeteksi Dini <i>Plagiarisme</i> .... | <b>Error! Bookmark not defined.</b> |
| BAB V.....  | <b>Error! Bookmark not defined.</b> |
| A. Hasil Pengujian Sistem .....                             | <b>Error! Bookmark not defined.</b> |
| B. Pembahasan.....  | <b>Error! Bookmark not defined.</b> |
| BAB VI .....  | 44                                  |
| A. Kesimpulan .....   | 44                                  |
| B. Saran.....   | 44                                  |
| DAFTAR PUSTAKA .....  | 46                                  |

## DAFTAR TABEL

|   |                                     |
|---|-------------------------------------|
| Tabel 2. 1 Tabel Matriks Perhitungan Edit Distance .....                | 8                                   |
| Tabel 2. 2 Simbol dan Keterangan Use case Diagram.....                  | 11                                  |
| Tabel 2. 3 Simbol dan Keterangan Class Diagram .....                    | 12                                  |
| Tabel 2. 4 Simbol dan Keterangan Sequence Diagram .....                 | 12                                  |
| Tabel 3. 1 Use Case Description dari Login Admin .....                  | 22                                  |
| Tabel 3. 2 Use Case Description Login Mahasiswa .....                   | 23                                  |
| Tabel 3. 3 Use Case Description Login Dosen .....                       | 23                                  |
| Tabel 3. 4 Use Case Description Mahasiswa Upload Proposal Skripsi ..... | 24                                  |
| Tabel 3. 5 Use Case Description Cek Plagiarisme .....                   | 24                                  |
| Tabel 3. 6 Use Case Description Mahasiswa Upload Naskah Publikasi.....  | 25                                  |
| Tabel 3. 7 Admin .....  | 34                                  |
| Tabel 3. 8 Mahasiswa.....   | 35                                  |
| Tabel 3. 9 Dosen .....  | 35                                  |
| Tabel 3. 10 Bimbingan.....  | 35                                  |
| Tabel 3. 11 Doc_Plagiarisme .....                                       | 35                                  |
| Tabel 3. 12 Skripsi.....  | 36                                  |
| Tabel 4. 1 Kebutuhan Hardware Sistem .....                              | <b>Error! Bookmark not defined.</b> |
| Tabel 4. 2 Kebutuhan Software .....                                     | <b>Error! Bookmark not defined.</b> |
| Tabel 4. 3 Hasil Uji Script Plagiarisme .....                           | <b>Error! Bookmark not defined.</b> |
| Tabel 4. 4 Hasil Uji Kecepatan Perhitungan .....                        | <b>Error! Bookmark not defined.</b> |
| Tabel 4. 5 Hasil Uji Sistem Pendeteksi Dini Plagiarisme .....           | <b>Error! Bookmark not defined.</b> |
| Tabel 5. 1 Daftar Dokumen Sumber.....                                   | <b>Error! Bookmark not defined.</b> |
| Tabel 5. 2 Hasil Uji Waktu Respon.....                                  | <b>Error! Bookmark not defined.</b> |

## DAFTAR GAMBAR

|  |                                     |
|--|-------------------------------------|
| Gambar 3. 1 Flowchart Alur Pendaftaran Sampai Seminar Proposal / Perkembangan .....                  | 18                                  |
| Gambar 3. 2 Flowchart Sistem yang Diajukan .....   | 21                                  |
| Gambar 3. 3 Use Case Sistem Pendeteksi Dini Plagiarisme .....  | 22                                  |
| Gambar 3. 4 Activity Diagram Login Mahasiswa .....   | 26                                  |
| Gambar 3. 5 Activity Diagram Login Dosen.....  | 27                                  |
| Gambar 3. 6 Activity Diagram Cek Plagiarisme .....   | 28                                  |
| Gambar 3. 7 Activity Diagram Upload Naskah Publikasi .....   | 29                                  |
| Gambar 3. 8 Activity Diagram Admin Kelola Data .....   | 30                                  |
| Gambar 3. 9 Sequence Diagram Cek Plagiarisme .....   | 31                                  |
| Gambar 3. 10 Sequence Diagram Upload Dokumen Naskah Publikasi .....                                  | 32                                  |
| Gambar 3. 11 Sequence Diagram Admin Kelola Data Dosen .....  | 32                                  |
| Gambar 3. 12 Sequence Diagram Admin Kelola Data Mahasiswa .....                                      | 33                                  |
| Gambar 3. 13 Sequence Diagram Admin Kelola Data Dokumen.....   | 33                                  |
| Gambar 3. 14 Class Diagram Sistem Pendeteksi Dini Plagiarisme .....                                  | 34                                  |
| Gambar 3. 15 EER Sistem Pendeteksi Dini Plagiarisme .....  | 36                                  |
| Gambar 3. 16 Relasi Dosen dan Admin .....  | 37                                  |
| Gambar 3. 17 Relasi Mahasiswa dan Admin.....   | 37                                  |
| Gambar 3. 18 Mahasiswa dan Bimbingan .....   | 37                                  |
| Gambar 3. 19 Relasi Dosen dan Bimbingan.....   | 38                                  |
| Gambar 3. 20 Relasi Bimbingan dan Doc_plagiarisme .....  | 38                                  |
| Gambar 3. 21 Relasi Bimbingan dan Skripsi.....   | 38                                  |
| Gambar 3. 22 Rancangan Halaman Dashboard Admin .....   | 39                                  |
| Gambar 3. 23 Rancangan Halaman Data Dosen Pada Admin .....   | 39                                  |
| Gambar 3. 24 Rancangan Data Mahasiswa Pada Admin.....  | 40                                  |
| Gambar 3. 25 Halaman dashboard Dosen .....   | 40                                  |
| Gambar 3. 26 Halaman Cek Plagiarisme.....  | 41                                  |
| Gambar 3. 27 Halaman Login .....   | 41                                  |
| Gambar 3. 28 Halaman Dashboard Mahasiswa.....  | 42                                  |
| Gambar 3. 29 Halaman Data Proposal .....   | 42                                  |
| Gambar 3. 30 Halaman Data Skripsi .....  | 43                                  |
| Gambar 4. 1 Implementasi Tampilan Mahasiswa Upload Dokumen . <b>Error! Bookmark not defined.</b>     |                                     |
| Gambar 4. 2 Source Code Tampilan Mahasiswa Upload Dokumen... <b>Error! Bookmark not defined.</b>     |                                     |
| Gambar 4. 3 Implementasi Tampilan Upload File .....  | <b>Error! Bookmark not defined.</b> |
| Gambar 4. 4 Mahasiswa Klik Tombol Tambah.....  | <b>Error! Bookmark not defined.</b> |
| Gambar 4. 5 Implementasi Tampilan Dokumen Akan Dicek Plagiarisme <b>Error! Bookmark not defined.</b> |                                     |
| Gambar 4. 6 Implementasi Tampilan Dokumen Akan Dicek Plagiarisme <b>Error! Bookmark not defined.</b> |                                     |
| Gambar 4. 7 Implementasi Tampilan Pilih Dokumen Pembanding .... <b>Error! Bookmark not defined.</b>  |                                     |

Gambar 4. 8 Implementasi Tampilan Pilih Dokumen Pembanding .... **Error! Bookmark not defined.**

Gambar 4. 9 Implementasi Tampilan Hasil Cek Plagiarisme ..... **Error! Bookmark not defined.**

Gambar 4. 10 Implementasi Tampilan Hasil Cek Plagiarisme ..... **Error! Bookmark not defined.**

Gambar 4. 11 Implementasi Notifikasi Mahasiswa Telah Dicek Plagiarisme ..... **Error! Bookmark not defined.**

Gambar 4. 12 Implementasi Notifikasi Mahasiswa Telah Dicek Plagiarisme ..... **Error! Bookmark not defined.**

Gambar 4. 13 Implementasi Algoritma Levenshtein Distance ..... **Error! Bookmark not defined.**

Gambar 4. 14 Baris Kode String Sumber dan Pembanding.. **Error! Bookmark not defined.**

Gambar 4. 15 Class Plagiarisme .php..... **Error! Bookmark not defined.**

Gambar 4. 16 Function getDifferenceSimilarity()..... **Error! Bookmark not defined.**

Gambar 4. 17 Hasil Perhitungan Algoritma Levenshtein Distance .... **Error! Bookmark not defined.**

Gambar 4. 18 Grafik Uji Kecepatan Perhitungan ..... **Error! Bookmark not defined.**

Gambar 5. 1 Hasil Admin Input Data Mahasiswa ..... **Error! Bookmark not defined.**

Gambar 5. 2 Hasil Admin Input Data Dosen..... **Error! Bookmark not defined.**

Gambar 5. 3 Hasil Klik Tombol Menu Dokumen Mahasiswa..... **Error! Bookmark not defined.**

Gambar 5. 4 Hasil Klik Tombol Tambah Untuk Upload Dokumen ... **Error! Bookmark not defined.**

Gambar 5. 5 Hasil Dashboard Menu Dosen Cek Plagiarisme ..... **Error! Bookmark not defined.**

Gambar 5. 6 Hasil Tampilan Modal Cek Plagiat ..... **Error! Bookmark not defined.**

Gambar 5. 7 Hasil Perhitungan Cek Plagiat ..... **Error! Bookmark not defined.**

Gambar 5. 8 Spesifikasi Minimum Komputer..... **Error! Bookmark not defined.**

Gambar 5. 9 Hasil Uji Waktu Respon Perhitungan ..... **Error! Bookmark not defined.**

## INTISARI

### SISTEM PENDETEKSI DINI *PLAGIARISME* PADA PROSES PENGAJUAN SKRIPSI MAHASISWA TEKNIK INFORMATIKA MENGGUNAKAN ALGORITMA *LEVENSHTEIN DISTANCE*

Oleh : Muhamad Hendra Febiawan  
Pembimbing : 1. Agus Setiawan, M. Eng.  
2. Ardhin Primadewi, S. Si, M. TI.

Dunia akademis di Indonesia sudah berkembang pesat. Hal ini ditandai dengan perkembangan ilmu pengetahuan dan teknologi. Dengan segala kemudahannya, teknologi membawa dampak positif dan negatif pada kehidupan. Salah satu dampak negatif yang ditimbulkan adalah plagiarisme. Fenomena plagiarisme sering terjadi pada kalangan mahasiswa. Oleh karena itu, pendeteksian plagiarisme perlu dilakukan. Penelitian ini bertujuan untuk mendeteksi kemiripan isi teks dokumen menggunakan algoritma *Levenshtein Distance*. Tipe dokumen yang digunakan adalah .pdf . Dokumen yang digunakan adalah proposal skripsi dan Naskah Publikasi. Pada uji kecepatan perhitungan, dokumen sumber yang digunakan memiliki *count words* 4405, dengan 3 dokumen pembanding yang memiliki *count words* 13465. Hal ini menghasilkan lama perhitungan selama 3,57 detik.

**Kata Kunci :** *Plagiarisme, Algoritma Levenshtein Distance, Dokumen.*

## **ABSTRACT**

### ***DYNAMIC PLAGIARISM DETECTION SYSTEM IN THE SUBMISSION PROCESS OF STUDENTS ESSAY INFORMATICS ENGINEERING USING ALGORITHM LEVENSHTTEIN DISTANCE***

*By* : Muhamad Hendra Febiawan  
*Supervisors* : 1. Agus Setiawan, M. Eng.  
2. Ardhin Primadewi, S. Si, M. TI.

*The academic world in Indonesia has been growing rapidly. This is noticed by the development of science and technology. With all the facilities, technology has a positive and negative impact on life. One of negative impacts is plagiarism. The plagiarism often occurs among students. Therefore detection of plagiarism needs to be done. This aims to detect the similarity of the text content of the document using the Levenshtein Distance algorithm. The type of document used is .pdf . The documents used are thesis proposals and publication papers. In the calculation speed test, the source document used has count words of 4405 with 3 comparative documents that has count words of 13465. This results in a calculation duration of 3.57 seconds.*

***Keywords*** : *Plagiarism, Levenshtein Distance Algorithm, Documents.*



# **BAB I**

## **PENDAHULUAN**

### **A. Latar Belakang**

Di Indonesia terdapat berbagai macam perguruan tinggi baik negeri atau swasta. Menurut data dari Pangkalan Data Pendidikan Tinggi (PDDIKTI) Kementerian Riset, Teknologi dan Pendidikan Tinggi (KEMENRISTEKDIKTI, 2018), terdapat total 4.696 perguruan tinggi di Indonesia dengan rincian 1.062 Akademi, 279 Politeknik, 2.535 Sekolah Tinggi, 218 Institut, 584 Universitas dan 20 Akademi Komunitas. Data tersebut adalah total keseluruhan perguruan tinggi baik negeri dan swasta, sedangkan rincian total perguruan tinggi negeri di Indonesia adalah 436 perguruan tinggi dan total perguruan tinggi swasta di Indonesia adalah 4.260 perguruan tinggi.

Di Provinsi Jawa Tengah sendiri menurut (KEMENRISTEKDIKTI, 2018), terdapat 380 perguruan tinggi dengan rincian 38 perguruan tinggi negeri dan 342 perguruan tinggi swasta. Salah satu perguruan tinggi swasta di Provinsi Jawa Tengah adalah Universitas Muhammadiyah Magelang yang memiliki jumlah total dosen tetap berjumlah 183 Dosen dan 6.592 Mahasiswa dengan rasio dosen tetap/jumlah mahasiswa adalah 1 : 36 berdasarkan Data Pelaporan Tahun 2017/2018 menurut PDDIKTI (PDDIKTI, 2018).

Berdasarkan data diatas dunia akademis di Indonesia sudah semakin pesat berkembang yang ditandai dengan semakin banyaknya perguruan tinggi di Indonesia juga dengan perkembangan ilmu pengetahuan dan teknologi. Kemampuan mahasiswa dibidang ilmu pengetahuan dan teknologi terutama dibidang komputer menjadi salah satu faktor yang mendukung dengan berkembangnya dunia akademis di Indonesia. Dengan segala kemudahan yang ditawarkan teknologi modern akan sangat membantu proses kegiatan belajar mengajar di lingkungan kampus, salah satunya dalam penulisan karya tulis ilmiah di Perguruan Tinggi yang merupakan bagian dari tuntutan formal akademik.

Dilihat dari tujuan penulisannya, karya tulis ilmiah dibedakan menjadi dua, pertama untuk memenuhi tugas-tugas perkuliahan seperti makalah, laporan dan

lain-lain. Kedua, karya tulis ilmiah yang digunakan sebagai salah satu syarat mahasiswa untuk menyelesaikan program studi, yaitu skripsi untuk S1 dan tugas akhir untuk D3 (Panji Novantara, 2018).

Seperti di perguruan tinggi pada umumnya, skripsi di Universitas Muhammadiyah Magelang menjadi sangat penting dan merupakan bagian dari karya tulis ilmiah untuk menyelesaikan Program Sarjana S1. Skripsi merupakan kemampuan akademik mahasiswa dalam meneliti suatu kasus yang sesuai bidang keilmuannya, melalui skripsi mahasiswa dituntut untuk berpikir sistematis.

Hasil akhir dari karya tulis skripsi sendiri berbentuk Laporan Skripsi dan Naskah Publikasi. Naskah Publikasi merupakan naskah skripsi mahasiswa yang ditulis kembali dalam bentuk jurnal. Naskah publikasi ini merupakan salah satu bentuk karya tulis ilmiah mahasiswa pada kegiatan penelitian yang terbimbing oleh dosen. Penulisan naskah publikasi dapat diketahui terdapat konten penjiplakan atau *plagiarisme*.

Tahap penyimpanan data – data Skripsi pada Teknik Informatika di Universitas Muhammadiyah masih manual sehingga seorang mahasiswa memiliki peluang untuk melakukan *copy – paste* atau *plagiarisme* proposal atau laporan akhir Skripsi dari awal hingga akhir tanpa diketahui oleh jurusan atau dosen. Dengan demikian *Plagiarisme* adalah tindakan mengambil ide orang lain, mengambil tulisan orang lain, dan mengambil teks secara keseluruhan dan mengakuinya sebagai milik pribadi (Rocky Yefrenes Dillak, 2016).

Fenomena *plagiarisme* yang lebih spesifik sering terjadi pada dunia akademis, khususnya dilakukan mahasiswa. Hal ini dikarenakan kegiatan tulis menulis sering dilakukan mahasiswa untuk menyelesaikan tugas kuliah maupun tugas akhir. Mahasiswa sering berinteraksi dengan komputer, sehingga mempermudah praktik plagiat (Tudesman, 2014).

Ada dua cara untuk mengurangi tingkat *plagiarisme*, yaitu dengan mencegah dan mendeteksi. Mencegah berarti menjaga atau menghalangi agar *plagiarisme* tidak dilakukan. Usaha ini harus dilakukan sedini mungkin terutama pada sistem pendidik dan moral masyarakat (Tudesman, 2014).

Cara mendeteksi dokumen yang memiliki indikasi plagiarisme dapat dilakukan secara manual dengan cara membandingkan dua dokumen secara manual namun hal tersebut tidak efektif yang memakan banyak waktu dan tenaga. Oleh karena itu diperlukan sebuah sistem pendeteksi *plagiarisme* pada dokumen teks yang dilakukan secara kompatibel.

Salah satu metode yang dapat digunakan dalam melakukan deteksi kemiripan dokumen teks adalah dengan melakukan perhitungan dengan metode *Levenshtein Distance*. *Levenshtein Distance* sendiri merupakan salah satu algoritma *text similarity*, yaitu algoritma untuk menghitung kemiripan dua *string input* yang dibandingkan (Hamidillah Ajie, 2017).

Berdasarkan pada permasalahan diatas maka penulis merancang “Sistem Pendeteksi Dini Plagiarisme Pada Pengajuan Skripsi Mahasiswa Jurusan Teknik Informatika Menggunakan Algoritma *Levenshtein Distance*”, dengan *output* dari aplikasi berupa persentase kemiripan dan perbedaan isi teks dokumen agar dapat membantu dosen mendeteksi dini tindak *plagiarisme* sehingga dapat mengurangi tingkat *plagiarisme* pada kalangan mahasiswa Teknik Informatika Universitas Muhammadiyah Magelang.

## **B. Rumusan Masalah**

Bagaimana membuat sistem pendeteksi dini *plagiarisme* pada proses pengajuan skripsi mahasiswa Teknik Informatika di Universitas Muhammadiyah Magelang dengan algoritma *Levenshtein Distance* ?

## **C. Tujuan Penelitian**

Untuk membuat Sistem Pendeteksi Dini *Plagiarisme* Pada Proses Pengajuan Skripsi Mahasiswa Teknik Informatika Menggunakan Algoritma *Levenshtein Distance* di Universitas Muhammadiyah Magelang.

**D. Manfaat Penelitian**

Manfaat yang diharapkan dari hasil penelitian ini adalah :

1. Bagi peneliti hal ini dapat membantu menambah wawasan ilmu pengetahuan baik teori maupun praktik khususnya perancangan sistem pendeteksi dini *plagiarisme* menggunakan Algoritma *Levenshtein Distance*.
2. Bagi dosen diharapkan dapat membantu dalam membimbing mahasiswa untuk mengurangi tingkat *plagiarisme* di jurusan Teknik Informatika Universitas Muhammadiyah Magelang.
3. Bagi mahasiswa diharapkan dapat mengurangi tindakan *plagiarisme* dalam penulisan karya tulis ilmiah.

## BAB II

### TINJAUAN PUSTAKA

#### A. Penelitian Relevan

1. Penelitian yang dilakukan (Dwi Susanto, 2016) berjudul “Deteksi Plagiat Dokumen Tugas Daring Laporan Praktikum Mata Kuliah Desain Web Menggunakan Metode Naive Bayes”. Hasil dari penelitian tersebut adalah aplikasi pendeteksi plagiarisme yang dibuat untuk mendeteksi kesamaan antar dokumen dan menghasilkan nilai persentase plagiarisme. *Naive Bayes* digunakan sebagai metode untuk mendeteksi kesamaan antar dokumen tugas mahasiswa, dengan hal tersebut maka dapat memudahkan Dosen Jaga praktikum mahasiswa untuk memeriksa kesamaan dokumen unggahan mahasiswa.
2. Penelitian yang dilakukan (Rocky Yefrenes Dillak, 2016) berjudul “Sistem Deteksi Dini Plagiarisme Tugas Akhir Mahasiswa Menggunakan Algoritma N-Grams dan Winnowing”. Hasil dari penelitian tersebut adalah sistem plagiarisme yang membandingkan file dokumen yang berbeda dari Tugas Akhir Mahasiswa memberikan hasil berupa persentase *similarity* , namun semakin banyak isi sebuah file yang dideteksi, waktu prosesnya akan semakin lama.
3. Penelitian yang dilakukan (Panji Novantara, 2018) berjudul “Implementasi Algoritma Jaro-Winkler Distance Untuk Sistem Pendeteksi Plagiarisme Pada Dokumen Skripsi”. Hasil dari penelitian tersebut adalah sistem pendeteksi plagiarisme yang dapat digunakan untuk mendeteksi kesamaan dokumen skripsi dengan cara melakukan perbandingan antara dokumen asli dan dokumen uji yang diinputkan untuk mengetahui tingkat kemiripan (*similarity*) dari dokumen skripsi yang diuji.
4. Penelitian yang dilakukan (Na'firul Hasna Ariyani, 2016) berjudul “Aplikasi Pendeteksi Kemiripan Isi Teks Dokumen Menggunakan Metode *Levenshtein Distance*”. Hasil dari penelitian tersebut adalah sistem pendeteksi kemiripan dokumen teks dengan tipe dokumen yang diuji .pdf .docx dan .txt. Dokumen yang digunakan untuk perbandingan teks ini adalah dokumen berbahasa

Indonesia. Pada pengujian data real yaitu data dokumen berplagiat dengan algoritma *Levenshtein Distance* menghasilkan nilai similarity yang tinggi yaitu diatas 77% sampai 100% dengan ketentuan proporsi nilai plagiat dibawah 40%.

5. Penelitian yang dilakukan (Mufti Ari Bianto, 2018) berjudul “Perancangan Sistem Pendeteksi Plagiarisme Terhadap Topik Penelitian Menggunakan Metode *K-Means Clustering* dan Model *Bayesian*”. Hasil dari penelitian tersebut adalah sistem pendeteksi plagiarisme dengan metode K-Means yang dapat mempartisi data ke dalam *cluster* sehingga data yang memiliki karakteristik sama dikelompokkan ke dalam satu *cluster* yang sama dan data yang mempunyai karakteristik berbeda di kelompokkan ke dalam *cluster* lain yang kemudian dicari nilai terdekat dari kemiripan antar dokumen. Hasil perhitungan menggunakan *Microsoft Excel* menampilkan sebuah persentase kemiripan.

Dari lima penelitian tersebut, dapat diambil kesimpulan bahwa terdapat perbedaan pada setiap metode yang diterapkan dalam mencari nilai kesamaan dokumen. Selain itu juga terdapat perbedaan pada platform yang digunakan sebagai sistemnya yang kebanyakan masih berbasis bahasa *java*. Sistem pendeteksi dini plagiarisme pada konten teks digital dokumen naskah publikasi dengan algoritma *Levenshtein Distance*, karena *Levenshtein Distance* sendiri merupakan salah satu algoritma *text similarity*, yaitu algoritma untuk menghitung kemiripan dua *string input* yang dibandingkan. Algoritma ini memiliki kompleksitas yang kecil, yaitu  $O(m*n)$  dimana  $m$  dan  $n$  adalah panjang dari *string input 1* dan *string input 2* sehingga lebih efektif dalam melakukan perhitungan dibanding algoritma *text similarity* lain, terutama untuk kasus *string* yang panjang. Diharapkan akan menjadi solusi bagi pihak dosen dan prodi dalam membimbing mahasiswa mengerjakan penelitian skripsi agar menghindari tindakan plagiarisme yang merugikan diri sendiri dan kampus.

## B. Penjelasan Secara Teoritis Masing-Masing Variable

### 1. *Algoritma Levenshtein Distance*

*Levenshtein Distance* dibuat oleh Vladimir Levenshtein tahun 1965.

Perhitungan edit distance didapatkan dari matriks yang digunakan untuk menghitung jumlah perbedaan string antara dua string. Perhitungan jarak antara dua string ini ditentukan dari jumlah minimum operasi perubahan untuk membuat string A menjadi string B. Ada 3 macam operasi utama yang dapat dilakukan oleh algoritma ini yaitu :

#### a. Operasi Pengubahan Karakter

Operasi pengubahan karakter merupakan operasi menukar sebuah karakter dengan karakter lain contohnya penulis menuliskan *string* 'yang' menjadi 'yng'. Dalam kasus ini karakter 'm' diganti dengan huruf 'n'.

#### b. Operasi Penambahan Karakter

Operasi penambahan karakter berarti menambahkan karakter ke dalam suatu *string*. Contohnya *string* 'kepad' menjadi 'kepada', dilakukan penambahan karakter 'a' diakhir *string*. Penambahan karakter tidak hanya dilakukan diakhir kata, namun bisa ditambahkan diawal maupun disisipkan ditengah *string*.

#### c. Operasi Penghapusan Karakter

Operasi penghapusan karakter dilakukan untuk menghilangkan karakter dari suatu *string*. Contohnya *string* 'baru' diubah menjadi 'baru'. Pada operasi ini dilakukan penghapusan huruf 'r'.

Algoritma ini berjalan mulai dari pojok kiri atas sebuah *array* dua dimensi yang telah diisi sejumlah karakter *string* awal dan *string* target dan diberikan nilai *cost*. Nilai *cost* pada ujung kanan bawah menjadi nilai *edit distance* yang menggambarkan jumlah perbedaan dua *string*.

Tabel 2. 1Tabel Matriks Perhitungan *Edit Distance*

|   |   |   |   |   |   |
|---|---|---|---|---|---|
|   |   | S | A | Y | A |
|   | 0 | 1 | 2 | 3 | 4 |
| S | 1 | 0 | 1 | 2 | 3 |
| Y | 2 | 1 | 2 | 1 | 2 |
| A | 3 | 2 | 1 | 2 | 1 |

Contoh dari perhitungan *levenshtein* menggunakan 2 *string* yang berbeda kemudian dihitung *edit distance* pada tabel 2.1 dapat dilihat hasil perhitungan *edit distance* antara 2 *string* ‘sya’ dan ‘saya’ adalah 1 perbedaan. Pengecekan dimulai dari iterasi awal dari kedua *string* kemudian dilakukan operasi penambahan, penyisipan dan penghapusan. Nilai *edit distance* yaitu pada ujung kanan bawah matriks. Hanya ada satu proses penyisipan yang dilakukan yaitu penyisipan karakter ‘a’ pada *string* ‘sya’ sehingga menjadi ‘saya’.

## 2. Plagiarisme

Plagiarisme merupakan tindakan kriminal yang sering terjadi dalam dunia akademis. Plagiarisme itu sendiri berasal dari kata latin “*Plagiarus*” yang berarti penculik dan “*Plagiare*” yang berarti mencuri. Jadi, secara sederhana plagiat berarti mengambil ide, kata-kata, dan kalimat seseorang dan memosisikannya sebagai hasil karyanya sendiri atau menggunakan ide, kata-kata, dan kalimat tanpa mencantumkan sumber dimana seorang penulis mengutipnya. (Sastroasmoro, 2007)

Jenis *plagiarisme* berdasarkan klasifikasinya diantaranya, Jenis *plagiarisme* berdasarkan aspek yang dicuri yaitu kategori *plagiarisme* ide, *plagiarisme* isi, *plagiarisme* kata, kalimat, paragraf, dan *plagiarisme* total. Klasifikasi berdasarkan sengaja atau tidaknya *plagiarisme* yaitu *plagiarisme* sengaja dan *plagiarisme* tidak sengaja. Berdasarkan pada pola *plagiarisme* yaitu *plagiarisme* kata demi kata (*word for word plagiarizing*) dan *plagiarisme mosaik*. Klasifikasi berdasarkan proporsi atau persentase kata, kalimat, paragraf yang dibajak yaitu :

- a) Plagiarisme ringan, plagiarisme yang jumlah proporsi atau persentase kata, kalimat, paragraf yang dibajak tidak melebihi 30 persen (<30%).
- b) Plagiarisme sedang, plagiarisme yang jumlah proporsi atau persentase kata, kalimat, paragraf yang dibajak antara 30-70%.
- c) Plagiarisme berat, plagiarisme yang jumlah proporsi atau persentase kata, kalimat, paragraf yang dibajak lebih dari 70 persen (>70%).

### 3. Dokumen

Dokumen menurut bahasa Inggris berasal dari kata “*document*” yang mempunyai arti suatu yang tertulis atau yang tercetak dan segala benda yang memiliki berbagai keterangan dipilih untuk di disusun, di kumpulkan, di sediakan ataupun untuk disebar (Sora, 2018). Sedangkan menurut Kamus Umum Bahasa Indonesia menyebutkan dokumen adalah sesuatu yang tertulis atau tercetak yang dapat dipergunakan sebagai bukti atau keterangan. Dokumen merupakan salah satu hal yang sangat penting karena merupakan sumber informasi yang diperlukan oleh suatu instansi, organisasi, atau negara. Tanpa dokumen kita akan kehilangan data – data yang diperlukan untuk kegiatan kantor/organisasi masa yang akan datang.

### 4. Teks Mining

*Text Mining* memiliki definisi menambang data yang berupa teks dimana sumber data biasanya di dapatkan dari dokumen, dan tujuannya adalah mencari katakata yang dapat mewakili isi dari dokumen sehingga dapat dilakukan analisa keterhubungan antar dokumen. Text mining merupakan penerapan konsep dan teknik data mining untuk mencari pola dalam teks, yaitu proses penganalisisan teks guna menyarikan informasi yang bermanfaat untuk tujuan tertentu. Berdasarkan ketidakteraturan struktur data teks, maka proses text mining memerlukan beberapa tahap awal yang pada intinya adalah mempersiapkan agar teks dapat diubah menjadi lebih terstruktur. (Harlian, 2015)

## 5. Basis Data (*Database*)

Sistem basis data merupakan suatu gabungan dan juga perpaduan antara basis data (*database*) dengan suatu sistem manajemen basis data (SMBD) atau yang juga lebih sering dikenal dengan istilah *Database Management System* (DBMS) (Waliyanto, 2000). Basis data dapat diartikan sebagai kumpulan informasi yang saling berhubungan satu dengan lainnya dan tersimpan dalam komputer secara terstruktur. Basis data juga dapat digunakan sebagai media penyimpanan yang dapat diubah, ditambah, dan dihapus. Basis data merupakan komponen utama dari sistem informasi, karena dengan adanya basis data maka informasi dapat didefinisikan sesuai dengan jenisnya, dan basis data juga berguna sebagai perlindungan data, karena data hanya bisa diakses dan diubah oleh pihak yang diberikan hak tertentu.

## 6. *Unified Modeling Language* (UML)

“Beberapa literature menyebutkan bahwa UML menyediakan sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa diagram yang digabung, misalnya diagram komunikasi, diagram urutan dan diagram pewaktuan digabung menjadi diagram interaksi” (Widodo, 2011). Namun demikian model-model itu dapat dikelompokkan berdasarkan sifatnya yaitu statis atau dinamis. Jenis diagram itu antara lain :

### a. Diagram Statis

#### 1) *Use Case Diagram*

*UseCase Diagram* adalah diagram yang mendeskripsikan interaksi antara pengguna dengan aplikasi. Kesimpulannya *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan dapat dilihat pada tabel 2.2 (Nugroho, 2015).

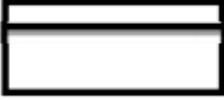
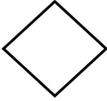
Tabel 2. 2 Simbol dan Keterangan *Use case Diagram*

| Simbol  | Keterangan  |
|---|---|
| Aktor<br>              | Mewakili peran orang, sistem yang lain atau alat ketika berkomunikasi dengan <i>use case</i>                                    |
| <i>UseCase</i><br>     | Abstraksi dan interaksi antara sistem dan aktor   |
| <i>Association</i><br> | Abstraksi dari penghubung antara aktor dan <i>use case</i>  |
| Generalisasi<br>       | Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i>  |
| <<include>><br>      | Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya                        |
| <<extend>><br>       | Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi |

## 2) *Class Diagram*

*Class Diagram* merupakan diagram yang memodelkan sekumpulan kelas, interface, kolaborasi dan relasinya. Diagram kelas digambarkan dengan bentuk kotak dapat dilihat pada table dibawah ini (Nugroho, 2015).

Tabel 2. 3 Simbol dan Keterangan *Class Diagram*

| Simbol   | Keterangan  |
|--|---|
| <p><i>Class</i></p>             | Himpunan dari objek-objek yang berbagai atribut serta operasi yang sama.  |
| <p><i>Nary Association</i></p>  | Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.   |
| <p><i>Generalization</i></p>    | Hubungan dimana objek anak ( <i>descendent</i> ) berbagai perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>oncestor</i> ) |
| <p><i>Realization</i></p>     | Operasi yang benar-benar dilakukan suatu objek.   |

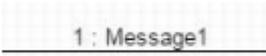
## b. Diagram Dinamis

1) *Sequence Diagram*

*Sequence Diagram* biasanya digunakan untuk tujuan analisa dan desain, memfokuskan pada identifikasi metode di dalam sebuah sistem (Nugroho, 2015). Simbol dan keterangan *Sequence Diagram* dapat dilihat pada tabel dibawah ini (Nugroho, 2015).

Tabel 2. 4 Simbol dan Keterangan *Sequence Diagram*

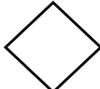
| Simbol   | Keterangan  |
|--|---|
| <p>Objek</p>  | Berpartisipasi secara berurutan dengan mengirimkan atau menerima pesan dan ditempatkan di bagian atas diagram |
| Garis Hidup Objek  |   |

|  |  |
|--|--|
|   | Menandakan kehidupan objek selama urutan dan diakhiri tanda X pada titik dimana kelas tidak lagi berinteraksi                            |
| <p>Objek sedang aktif berinteraksi</p>  | Persegi panjang yang sempit panjang ditempatkan diatas sebuah garis hidup dan menandakan ketika suatu onjek mengirim atau menerima pesan |
| <p><i>Message</i></p>                  | Perilaku sistem yang menandai adanya suatu alur informasi atau transisi kendali antar elemen   |

## 2) Activity Diagram

*Activity Diagram* menggambarkan alur aktivitas dalam aplikasi, menjelaskan proses masing-masing alur berawal dan proses aplikasi berakhir. Diagram aktivitas juga menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi, dapat dilihat pada table dibawah ini (Nugroho, 2015).

Tabel 2. 5 Simbol dan Keterangan Activity Diagram

| Simbol  | Keterangan  |
|---|---|
| <p>Status Awal</p>           | Status awal aktivitas sistem, sebuah diagram aktivitas memiliki status awal   |
| <p>Aktivitas</p>             | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja |
| <p>Percabangan/Decision</p>  | Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.       |

|   |   |
|---|---|
| Status Akhir  | Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir |
|  |   |

### 7. *Perl Hypertext Preprocessor (PHP)*

*Perl Hypertext Preprocessor* (PHP) adalah bahasa serverside scripting yang menyatu dengan HTML untuk membuat halaman web yang dinamis (Rudianto, 2011). Karena PHP merupakan serverside-scripting maka sintaks dan perintah-perintah PHP akan dieksekusi diserver kemudian hasilnya akan dikirimkan ke *browser* dengan format HTML.

PHP adalah sebuah bahasa pemrograman yang digunakan untuk membangun halaman web. Salah satu keunggulan dari PHP adalah kode program yang ditulis dalam PHP tidak akan terlihat oleh user, sehingga keamanan halaman web lebih terjamin. PHP dirancang untuk membuat halaman web menjadi lebih dinamis. Kelebihan lainnya dari PHP adalah bahasa pemrograman atau sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya, Web Server yang mendukung PHP mudah untuk ditemukan seperti, IIS, Lighttpd, hingga Xitami dengan konfigurasi yang relatif mudah, mudah untuk dikembangkan, karena banyaknya milis-milis dan *developer* yang siap membantu dalam pengembangan, dalam sisi pemahaman, PHP adalah bahasa scripting yang paling mudah karena refrensinya sangat banyak dan mudah untuk ditemukan (Rudianto, 2011).

### 8. *Web Application Framework*

*Web Application Framework* merupakan sebuah kerangka perangkat lunak yang dirancang untuk membantu pembangunan web dinamis, aplikasi web, web services, dan web resources. Dengan menggunakan sebuah framework, proses pembangunan web akan menjadi semakin mudah, cepat dan hemat biaya. Ini dikarenakan sebagian besar *framework* telah mengimplementasikan fitur-fitur seperti *Data Persistence*, *Session Management*, *User Authentication*, *Security*, *Caching*, dan *Administrative Interface* (DocForge, 2014)..

## 9. Laravel

Laravel adalah salah satu *web application framework* yang bersifat open source. *Framework* ini berjalan diatas PHP 5 dan berbasis MVC (*Model View Controller*). Laravel pertama kali dirilis pada 22 Februari 2012 dan versi stabil adalah 4.2.11 yang dirilis 04 Oktober 2014 (F, 2014).

## 10. Arsitektur *Model View Controller (MVC)*

MVC adalah sebuah metode untuk membuat sebuah aplikasi dengan memisahkan data (*Model*) dari tampilan (*View*) dan cara bagaimana memprosesnya (*Controller*) (LLC, 2015). Terdapat tiga komponen pembangun suatu MVC yaitu :

### a) Model

Model adalah bagian yang berperan menghubungkan *controller* dengan *database*. Tugas dari model adalah melakukan manipulasi data ke *database* seperti CRUD (*create, read, update, delet*).

### b) View

*View* mengatur bagaimana data akan ditampilkan kepada user. Data yang didapat dari model akan diproses oleh *controller* kemudian oleh *view* ditampilkan ke user.

### c) Controller

*Controller* merupakan bagian yang menjadi penghubung antara *model* dan *view*. *Controller* berfungsi memroses fungsi atau perintah dari user kemudian menentukan bagaimana aplikasi dijalankan.

## C. Landasan Teori

Berdasarkan teori-teori diatas, beberapa jurnal telah membuktikan ada banyak algortima yang dapat digunakan untuk mendeteksi plagiarisme, namun penulis menggunakan algoritma *Levenshtein Distance* untuk mengukur kesamaan antara 2 string, yaitu string sumber (s) dengan string target (t) atau string pembanding. Selain itu, algoritma ini juga telah digunakan untuk *Spell Checking, Speech Recognition, DNA Alanlysis, Plagiarism Detection* dan untuk penerapan pada sistem

juga sudah dapat diterapkan melalui bahasa pemrograman PHP. Maka dari itu algoritma *Levenshtein Distance* dapat diterapkan untuk mengurangi tingkat *plagiarisme* pada sistem pendeteksi dini *plagiarisme* pada konten teks *digital* Naskah Publikasi mahasiswa di jurusan Teknik Informatika Universitas Muhammadiyah Magelang.

## BAB III

### ANALISIS DAN PERANCANGAN SISTEM

#### A. Analisis Sistem

##### 1. *Algoritma Levenshtein Distance*

*Levenshtein Distance* sendiri merupakan salah satu algoritma *text similarity*, yaitu algoritma untuk menghitung kemiripan dua *string input* yang dibandingkan. Dalam kasus ini pengecekan *plagiarisme* memanfaatkan dokumen sumber dan dokumen target melalui tahap sebagai berikut :

##### a. *Case Folding*

Proses *Case Folding* adalah tahap mengubah semua huruf dalam dokumen menjadi huruf kecil hanya huruf a sampai z yang diterima.

##### b. *Tokenizing*

Proses *Tokenizing* yaitu proses memisahkan setiap kata yang menyusun suatu dokumen. Umumnya setiap kata teridentifikasi atau terpisahkan dengan kata lain oleh karakter spasi.

##### c. *Operasi Pengubahan Karakter*

Operasi pengubahan karakter merupakan operasi menukar sebuah karakter dengan karakter lain contohnya penulis menuliskan *string* 'yamg' menjadi 'yang'. Dalam kasus ini karakter 'm' diganti dengan huruf 'n'.

##### d. *Operasi Penambahan Karakter*

Operasi penambahan karakter berarti menambahkan karakter ke dalam suatu *string*. Contohnya *string* 'kepad' menjadi 'kepada', dilakukan penambahan karakter 'a' diakhir *string*. Penambahan karakter tidak hanya dilakukan diakhir kata, namun bisa ditambahkan diawal maupun disisipkan ditengah *string*.

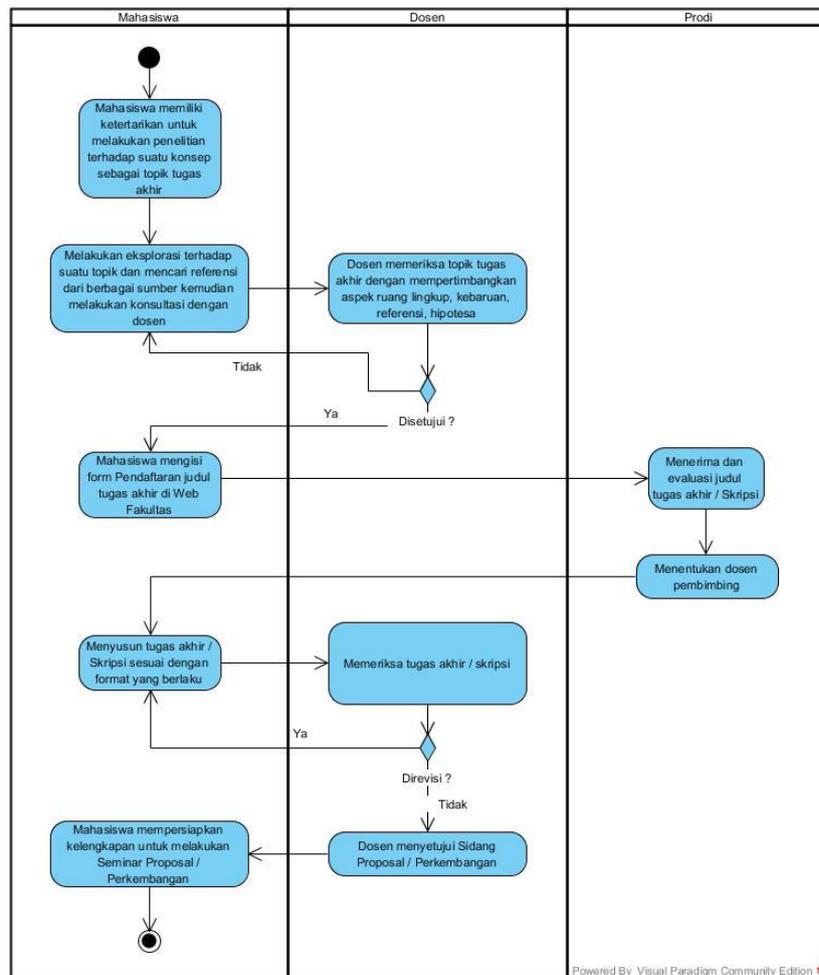
### e. Operasi Penghapusan Karakter

Operasi penghapusan karakter dilakukan untuk menghilangkan karakter dari suatu *string*. Contohnya *string* 'barur' diubah menjadi 'baru'. Pada operasi ini dilakukan penghapusan huruf 'r'.

## 2. Sistem Yang Berjalan

### a. Uraian Sistem Yang Berjalan

Alur sistem pengerjaan tugas akhir / skripsi mulai pendaftaran tugas akhir / skripsi, bimbingan dosen, sampai disetujui mengikuti sidang seminar perkembangan di Teknik Informatika masih bersifat manual tanpa ada sistem cek *plagiarisme* didalamnya, seperti yang dapat dilihat pada Gambar 3.1



Gambar 3. 1 *Flowchart* Alur Pendaftaran Sampai Seminar Proposal / Perkembangan

Gambar 3.1 menjelaskan alur sistem pengerjaan skripsi mulai dari pendaftaran sampai disetujui untuk mengikuti seminar proposal / perkembangan. Pada sistem manual, mahasiswa memiliki ketertarikan untuk melakukan penelitian terhadap suatu konsep sebagai topik skripsi. Kemudian mahasiswa melakukan eksplorasi terhadap topik yang akan diangkat dengan mencari referensi dari berbagai sumber setelah itu mahasiswa melakukan konsultasi dengan dosen. Dosen memeriksa topik yang akan diangkat dengan memperhatikan aspek ruang lingkup, kebaruan, referensi, dan hipotesa. Apabila dosen tertarik dan menyetujui dengan topik yang diangkat, maka mahasiswa dapat mendaftarkan diri untuk mengambil skripsi ke Prodi. Setelah masa pendaftaran, prodi melakukan evaluasi dan membagi dosen pembimbing sesuai topik yang diangkat masing-masing mahasiswa. Setelah dosen pembimbing dibagi, mahasiswa dapat mulai menyusun format laporan sesuai aturan baku dan melakukan bimbingan seperti biasa kepada dosen pembimbing.

#### **b. Identifikasi Permasalahan**

Berdasarkan sistem yang berjalan, diperoleh permasalahan yang ada pada sistem yang sedang berjalan. Permasalahan sebagai berikut :

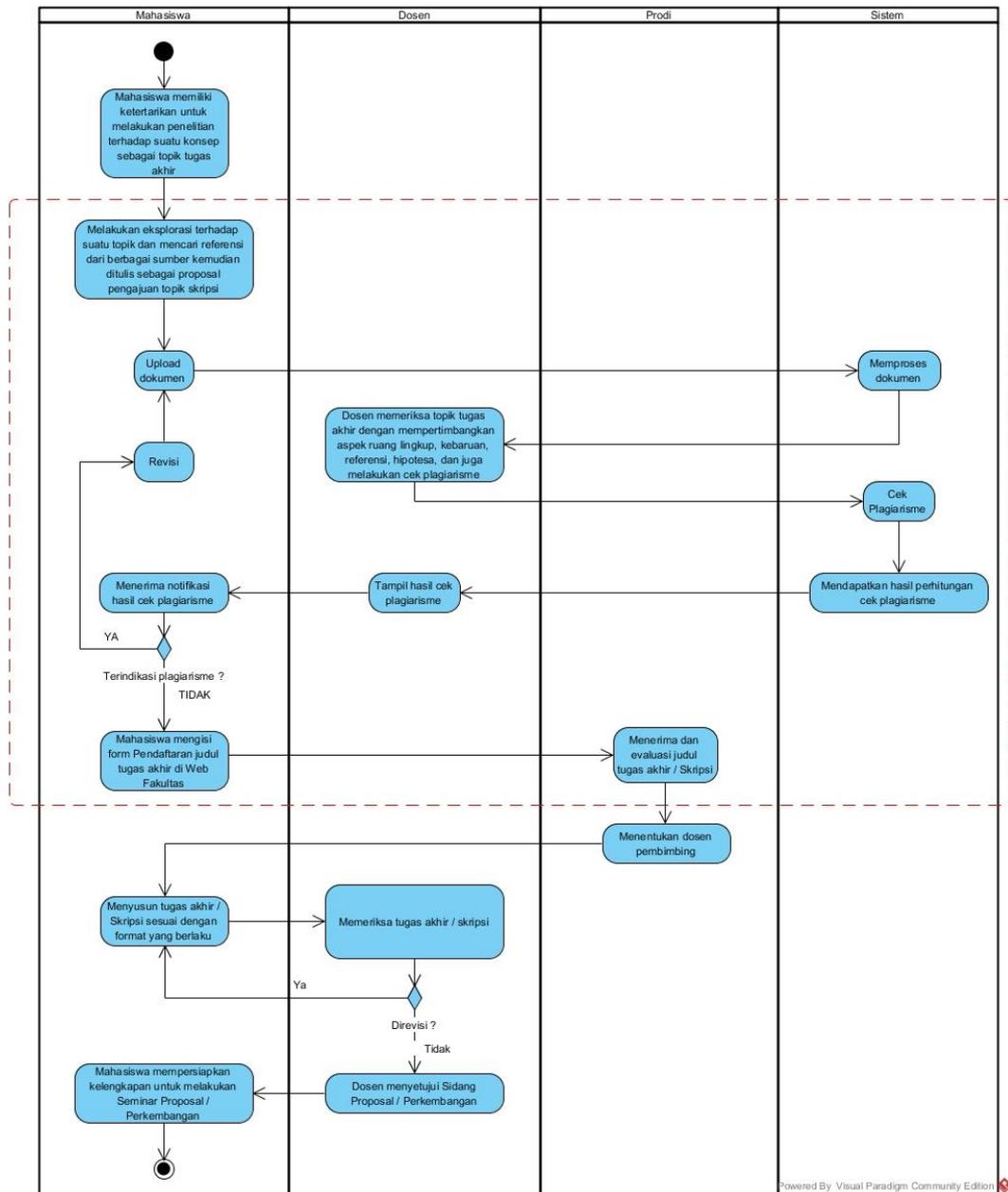
- 1) Setelah mahasiswa melakukan eksplorasi terhadap topik yang diangkat dengan mencari referensi dari berbagai sumber, mahasiswa melakukan konsultasi kepada dosen, kemudian dosen memeriksa topik yang akan diangkat tanpa adanya cek *plagiarisme* pada dokumen mahasiswa.
- 2) Dosen tidak bisa mengetahui karya ilmiah dari mahasiswa yang terindikasi *plagiarisme*.

- 3) Tanpa melalui cek *plagiarisme* pada saat konsultasi, indikasi penjiplakan akan terus berlanjut pada pengerjaan penulisan karya ilmiah seperti tugas akhir / skripsi.

### 3. Sistem Yang Diusulkan

Setelah menganalisa sistem yang berjalan di Jurusan Teknik Informatika, maka sistem yang diusulkan sebagai berikut. Pada awal pengerjaan karya ilmiah tugas akhir / skripsi mulai dari ketertarikan terhadap topik, konsultasi, pendaftaran, pembagian dosen pembimbing, penulisan, sampai bimbingan sama seperti dengan sistem yang sedang berjalan. Pada tahap bimbingan dosen tidak hanya memeriksa format penulisan, referensi, dan isi, namun dosen juga dapat melakukan cek *plagiarisme* dengan cara mahasiswa mengunggah dokumen dalam format .pdf ke dalam sistem cek *plagiarisme*, kemudian dosen membuka sistem dan memeriksa dokumen terbaru yang diunggah mahasiswa, setelah itu dosen memilih dokumen naskah publikasi milik mahasiswa yang telah melakukan sidang pendarasan dan juga telah mengunggah dokumen naskah publikasi kedalam sistem.

Fungsinya adalah untuk membandingkan dua dokumen atau lebih untuk mengecek indikasi adanya tindak *plagiarisme* pada karya ilmiah mahasiswa. Setelah sistem berhasil melakukan perhitungan dan mendapatkan hasil perhitungan cek *plagiarisme*, maka hasil ditampilkan dalam bentuk *pop up web* di halaman cek *plagiarisme* dosen, kemudian dosen mengirim notifikasi ke mahasiswa berupa hasil cek *plagiarisme*. Apabila terdeteksi *plagiarisme*, maka mahasiswa wajib melakukan revisi dan mengupload dokumen baru ke dosen agar dapat melakukan bimbingan lagi sampai tidak terdeteksi *plagiarisme*. Gambar 3.2 menjelaskan alur sistem cek *plagiarisme* yang diajukan ditunjukkan dengan garis putus – putus.



Gambar 3. 2 Flowchart Sistem yang Diajukan

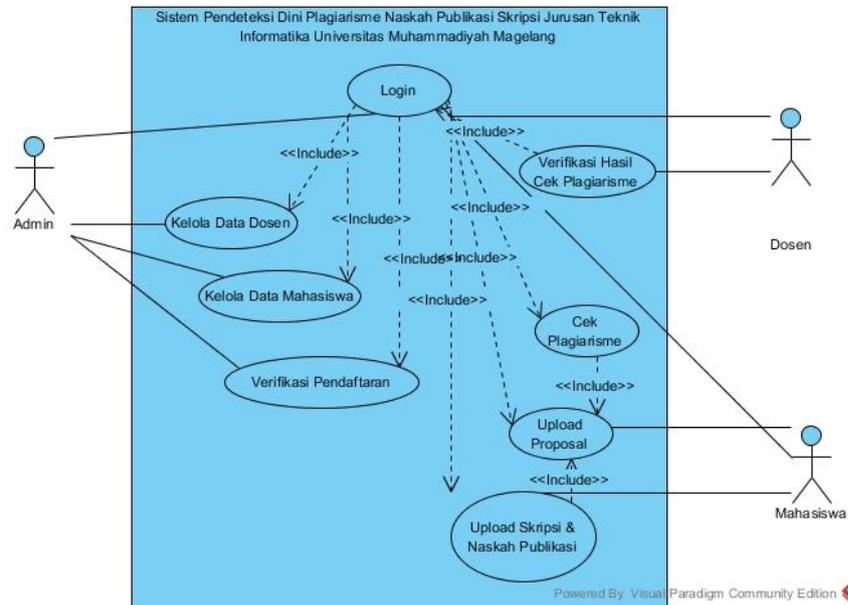
## B. Perancangan Sistem

### 1. Perancangan UML

#### a. Diagram Use Case Sistem Pendeteksi Dini *Plagiarisme*

Sistem pendeteksi dini *plagiarisme* yang akan dibuat memiliki 3 aktor utama, yaitu dosen, mahasiswa, dan admin. Dosen dapat melakukan login sistem dan cek *plagiarisme*. Mahasiswa dapat melakukan *login* sistem, *upload* proposal tugas akhir / skripsi dan *upload* naskah publikasi setelah sidang

pendadaran. Admin dapat melakukan *login* sistem, mengelola data dosen, data mahasiswa, dan data bimbingan. Rancangan *Use Case* dapat dilihat pada Gambar 3.3 dibawah ini.



Gambar 3. 3 *Use Case* Sistem Pendeteksi Dini *Plagiarisme*

Tabel 3. 1 *Use Case Description* dari *Login Admin*

|                             |   |
|-----------------------------|---|
| <i>Use Case Name</i>        | <i>Login Admin</i>  |
| <i>Primary Actor</i>        | Admin   |
| <i>Supporting Actor</i>     | Sistem Pendeteksi Dini <i>Plagiarisme</i>   |
| <i>Summary</i>              | Admin <i>login</i> ke Sistem dengan memasukkan <i>username</i> dan <i>password</i> kemudian klik tombol <i>login</i> . Sistem mengecek kebenaran <i>username</i> dan <i>password</i> .  |
| <i>Pre-condition</i>        | Admin memiliki <i>password</i> dan <i>username</i> .<br>Sistem Pendeteksi Dini <i>Plagiarisme</i> beroperasi.   |
| <i>Normal Flor of Event</i> | <ol style="list-style-type: none"> <li>1. Admin mengeksekusi sistem.</li> <li>2. Aplikasi menampilkan <i>login dialog</i>.</li> <li>3. Admin memasukkan <i>username</i> dan <i>password</i>.</li> <li>4. Klik tombol <i>login</i>.</li> <li>5. Sistem validasi <i>login</i>.</li> <li>6. <i>Dashboard</i> terbuka dan menu tersedia.</li> </ol> |
| <i>Post-condition</i>       | <ol style="list-style-type: none"> <li>1. <i>Dashboard</i> Sistem terbuka.</li> <li>2. Menu tersedia.</li> </ol>  |

Tabel 3. 2 Use Case Description Login Mahasiswa

|                             |   |
|-----------------------------|---|
| <i>Use Case Name</i>        | <i>Login Mahasiswa</i>  |
| <i>Primary Actor</i>        | Mahasiswa   |
| <i>Supporting Actor</i>     | Sistem Pendeteksi Dini <i>Plagiarisme</i>   |
| <i>Summary</i>              | Mahasiswa <i>login</i> ke Sistem dengan memasukkan <i>username</i> dan <i>password</i> kemudian klik tombol <i>login</i> . Sistem mengecek kebenaran <i>username</i> dan <i>password</i> .  |
| <i>Pre-condition</i>        | Mahasiswa memiliki <i>password</i> dan <i>username</i> yang diberikan Admin.<br>Sistem Pendeteksi Dini <i>Plagiarisme</i> beroperasi.   |
| <i>Normal Flor of Event</i> | <ol style="list-style-type: none"> <li>1. Mahasiswa mengeksekusi sistem.</li> <li>2. Aplikasi menampilkan <i>login dialog</i>.</li> <li>3. Mahasiswa memasukkan <i>username</i> dan <i>password</i>.</li> <li>4. Klik tombol <i>login</i>.</li> <li>5. Sistem validasi <i>login</i>.</li> <li>6. <i>Dashboard</i> terbuka dan menu tersedia.</li> </ol> |
| <i>Post-condition</i>       | <ol style="list-style-type: none"> <li>1. <i>Dashboard</i> Sistem terbuka.</li> <li>2. Menu tersedia.</li> </ol>  |

Tabel 3. 3 Use Case Description Login Dosen

|                             |   |
|-----------------------------|---|
| <i>Use Case Name</i>        | <i>Login Dosen</i>  |
| <i>Primary Actor</i>        | Dosen   |
| <i>Supporting Actor</i>     | Sistem Pendeteksi Dini <i>Plagiarisme</i>   |
| <i>Summary</i>              | Dosen <i>login</i> ke Sistem dengan memasukkan <i>username</i> dan <i>password</i> kemudian klik tombol <i>login</i> . Sistem mengecek kebenaran <i>username</i> dan <i>password</i> .  |
| <i>Pre-condition</i>        | Dosen memiliki <i>password</i> dan <i>username</i> .<br>Sistem Pendeteksi Dini <i>Plagiarisme</i> beroperasi.   |
| <i>Normal Flor of Event</i> | <ol style="list-style-type: none"> <li>1. Dosen mengeksekusi sistem.</li> <li>2. Aplikasi menampilkan <i>login dialog</i>.</li> <li>3. Dosen memasukkan <i>username</i> dan <i>password</i>.</li> <li>4. Klik tombol <i>login</i>.</li> <li>5. Sistem validasi <i>login</i>.</li> <li>6. <i>Dashboard</i> terbuka dan menu tersedia.</li> </ol> |
| <i>Post-condition</i>       | <ol style="list-style-type: none"> <li>1. <i>Dashboard</i> Sistem terbuka.</li> <li>2. Menu tersedia.</li> </ol>  |

Tabel 3. 4 Use Case Description Mahasiswa Upload Proposal Skripsi

|                             |  |
|-----------------------------|--|
| <i>Use Case Name</i>        | Mahasiswa <i>Upload</i> Proposal Skripsi   |
| <i>Primary Actor</i>        | Mahasiswa  |
| <i>Supporting Actor</i>     | Sistem Pendeteksi Dini <i>Plagiarisme</i>  |
| <i>Summary</i>              | Mahasiswa memilih menu Dokumen.<br><i>Layout</i> terbuka dan tersedia tombol tambah.   |
| <i>Pre-condition</i>        | Mahasiswa memiliki dokumen proposal skripsi dengan format .pdf (Minimal BAB 1)<br>Sistem Pendeteksi Dini <i>Plagiarisme</i> beroperasi.  |
| <i>Normal Flor of Event</i> | <ol style="list-style-type: none"> <li>1. Mahasiswa mengeksekusi sistem.</li> <li>2. Aplikasi menampilkan <i>layout</i> Dokumen.</li> <li>3. Tampil <i>button</i> Tambah.</li> <li>4. Klik <i>button</i> Tambah.</li> <li>5. Tampil Modal untuk <i>Upload</i> dokumen.</li> <li>6. Tersedia <i>button</i> browse <i>file</i>.</li> <li>7. Klik Simpan.</li> <li>8. Tampil hasil dokumen yang <i>diupload</i>.</li> </ol> |
| <i>Post-condition</i>       | <ol style="list-style-type: none"> <li>1. <i>Layout</i> menu Dokumen terbuka.</li> <li>2. Menu tersedia.</li> </ol>  |

Tabel 3. 5 Use Case Description Cek Plagiarisme

|                             |  |
|-----------------------------|--|
| <i>Use Case Name</i>        | Cek <i>Plagiarisme</i>   |
| <i>Primary Actor</i>        | Dosen  |
| <i>Supporting Actor</i>     | Sistem Pendeteksi Dini <i>Plagiarisme</i>  |
| <i>Summary</i>              | Dosen memilih menu Cek <i>Similarity</i> .<br><i>Layout</i> terbuka dan tampil daftar dokumen proposal skripsi yang <i>diupload</i> mahasiswa.   |
| <i>Pre-condition</i>        | Daftar dokumen yang akan dicek telah <i>diupload</i> .<br>Sistem Pendeteksi Dini <i>Plagiarisme</i> beroperasi.  |
| <i>Normal Flor of Event</i> | <ol style="list-style-type: none"> <li>1. Dosen mengeksekusi sistem.</li> <li>2. Aplikasi menampilkan daftar dokumen yang akan dicek.</li> <li>3. Dosen memilih dokumen yang akan dicek.</li> <li>7. Klik tombol cek.</li> <li>8. Tampil Modal informasi dokumen sumber dan daftar dokumen pembanding.</li> <li>9. Klik tombol cek <i>plagiarisme</i>.</li> <li>10. Sistem melakukan perhitungan.</li> <li>11. Tampil persentase hasil.</li> <li>12. Tersedia tombol Terima dan Tolak</li> <li>13. Klik Tombol Terima, mahasiswa bisa daftar skripsi.</li> <li>14. Klik Tombol Tolak, mahasiswa revisi dokumen.</li> </ol> |
| <i>Post-condition</i>       | <ol style="list-style-type: none"> <li>1. <i>Dashboard</i> Sistem terbuka.</li> <li>2. Menu tersedia.</li> </ol>   |

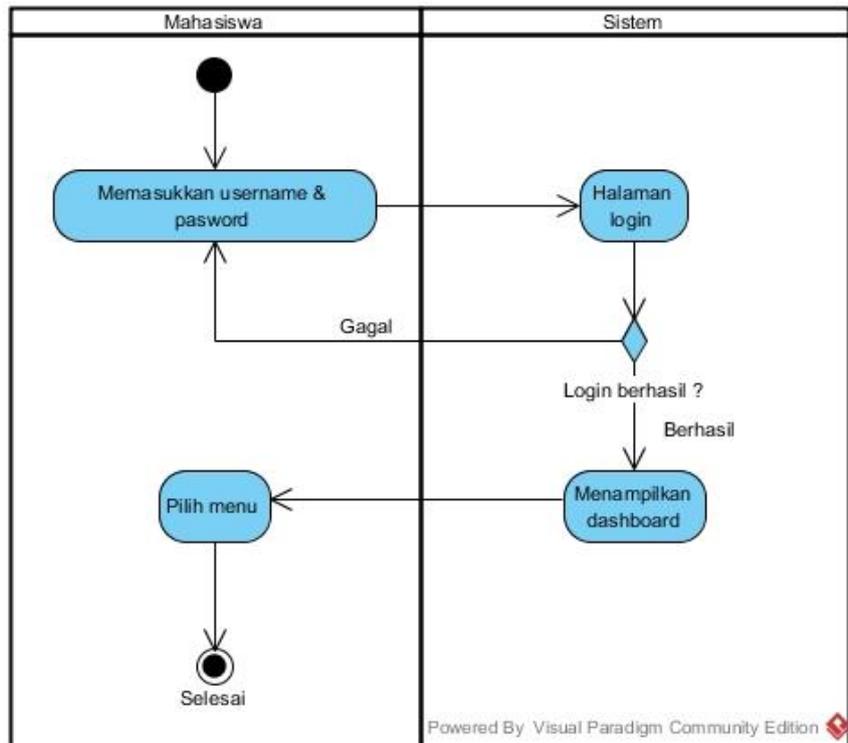
Tabel 3. 6 *Use Case Description* Mahasiswa Upload Naskah Publikasi

|                             |  |
|-----------------------------|--|
| <i>Use Case Name</i>        | Mahasiswa <i>Upload</i> Naskah Publikasi   |
| <i>Primary Actor</i>        | Mahasiswa  |
| <i>Supporting Actor</i>     | Sistem Pendeteksi Dini <i>Plagiarisme</i>  |
| <i>Summary</i>              | Mahasiswa memilih menu Pendaftaran.<br><i>Layout</i> terbuka dan tersedia tombol <i>Upload</i> dokumen.<br>Sudah melakukan Sidang Pendadaran.<br>Sudah diverifikasi Admin.   |
| <i>Pre-condition</i>        | Sudah terverifikasi oleh Admin<br>Mahasiswa memiliki dokumen Naskah Publikasi dengan format .pdf.<br>Sistem Pendeteksi Dini <i>Plagiarisme</i> beroperasi.   |
| <i>Normal Flor of Event</i> | <ol style="list-style-type: none"> <li>1. Mahasiswa mengeksekusi sistem.</li> <li>2. Aplikasi menampilkan <i>layout Upload</i> dokumen Naskah Publikasi.</li> <li>3. Tampil <i>button</i> Tambah.</li> <li>4. Klik <i>button</i> Tambah.</li> <li>5. Tampil Modal untuk <i>Upload</i> dokumen.</li> <li>6. Tersedia <i>button</i> <i>browse file</i>.</li> <li>7. Klik Simpan.</li> <li>8. Tampil hasil dokumen yang <i>diupload</i>.</li> </ol> |
| <i>Post-condition</i>       | <ol style="list-style-type: none"> <li>1. <i>Layout</i> menu Pendaftaran terbuka.</li> <li>2. Menu tersedia.</li> </ol>  |

## b. Diagram Activity Sistem Pendeteksi Dini *Plagiarisme*

### 1) Diagram Activity *Login* Mahasiswa

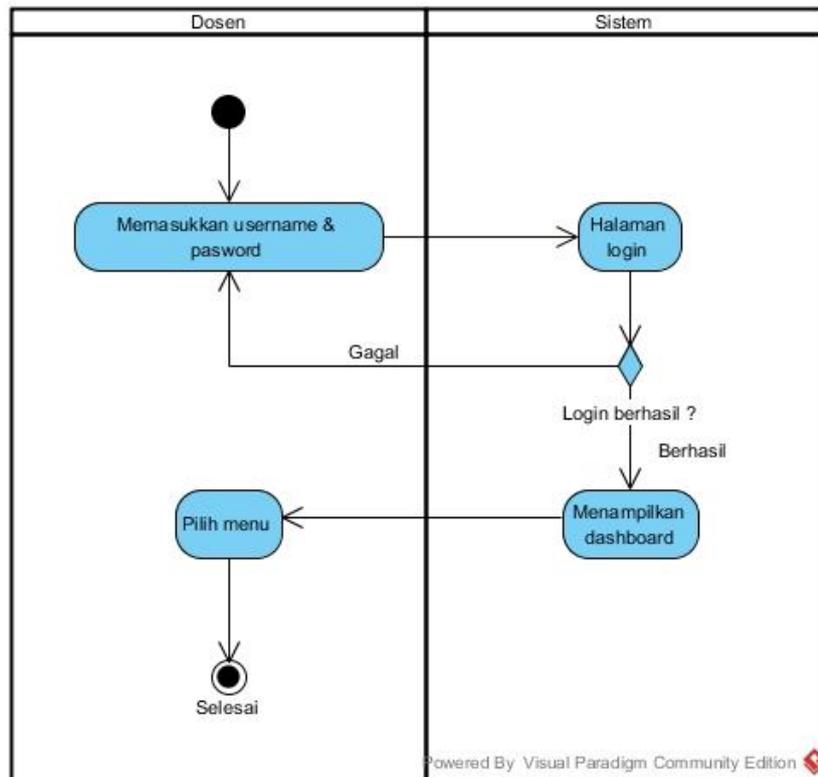
Pada Gambar 3.4 dibawah ini dijelaskan urutan tahap *login* sistem dari mahasiswa. Mahasiswa membuka web sistem kemudian memasukkan *username* dan *password*, setelah itu sistem akan melakukan cek validasi apakah data yang dimasukkan sudah benar, apabila gagal *login* maka mahasiswa harus melakukan *input* data lagi sampai benar. Apabila *login* berhasil akan muncul *dashboard* dan mahasiswa dapat memilih menu yang tersedia.



Gambar 3. 4 Activity Diagram Login Mahasiswa

## 2) Diagram Activity *Login Dosen*

Pada Gambar 3.5 dibawah ini dijelaskan urutan tahap *login* sistem dari dosen. Dosen membuka web sistem kemudian memasukkan *username* dan *password*, setelah itu sistem akan melakukan cek validasi apakah data yang dimasukkan sudah benar, apabila gagal *login* maka dosen harus melakukan *input* data lagi sampai benar. Apabila *login* berhasil akan muncul *dashboard* dan dosen dapat memilih menu yang tersedia.

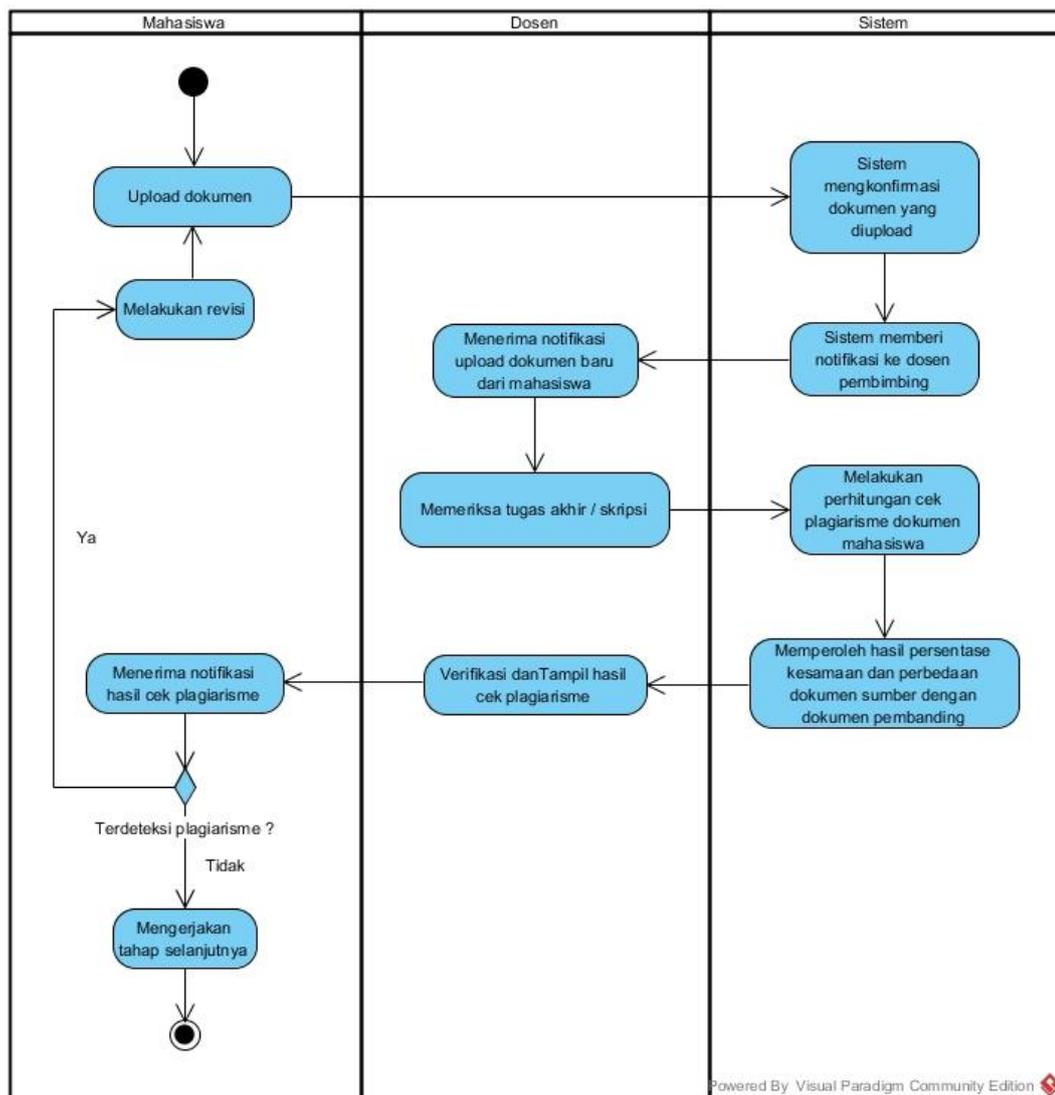


Gambar 3. 5 Activity Diagram Login Dosen

### 3) Diagram Activity Cek *Plagiarisme*

Diagram activity cek *plagiarisme* dilakukan oleh dua aktor utama dan dijalankan oleh sistem. Pertama setelah mahasiswa melakukan *login*, mahasiswa dapat memilih menu *upload document*. Setelah itu dosen akan mendapat notifikasi bahwa terdapat dokumen baru yang diunggah oleh mahasiswa. Kemudian dosen akan memeriksa dokumen baru yang diunggah, tahap selanjutnya dosen akan melakukan cek *plagiarisme* dengan dokumen sumber dari proposal tugas akhir / skripsi mahasiswa dan mencari dokumen pembanding berupa naskah publikasi yang mungkin menurut calon dosen pembimbing memiliki kemiripan dengan dokumen sumber. Tahap selanjutnya sistem akan melakukan proses cek *plagiarisme*, setelah sistem mendapatkan hasil maka akan muncul *pop up* dihalaman web dosen pembimbing. Kemudian

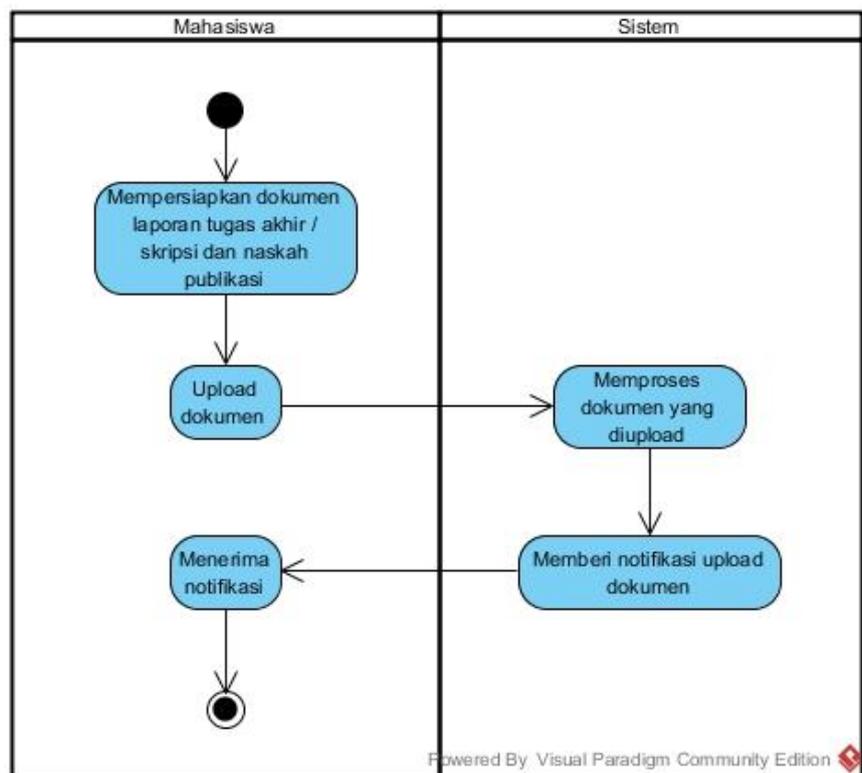
dosen akan mengirim notifikasi ke mahasiswa mengenai hasil cek *plagiarisme*. Apabila dokumen terdeteksi *plagiarisme*, maka mahasiswa wajib melakukan revisi dan mengulangi tahap sebelumnya sampai tidak terdeteksi *plagiarisme*. Jika dokumen tidak terdeteksi *plagiarisme*, maka mahasiswa dapat melanjutkan ke tahap berikutnya seperti ditunjukkan pada proses Gambar 3.6 dibawah ini.



Gambar 3. 6 Activity Diagram Cek Plagiarisme

#### 4) Diagram Activity *Upload* dokumen Naskah Publikasi

Diagram activity *upload* dokumen naskah publikasi dilakukan oleh mahasiswa, namun mahasiswa yang boleh melakukan *upload* dokumen naskah publikasi adalah mahasiswa yang sudah melakukan sidang pendadaran dan sudah menyelesaikan revisi. Seperti tahap sebelumnya, mahasiswa melakukan *login* sistem kemudian memilih menu untuk *upload* dokumen naskah publikasi. Selanjutnya sistem akan memproses dokumen dan disimpan pada database dan mahasiswa mendapat notifikasi seperti pada Gambar 3.7 dibawah.

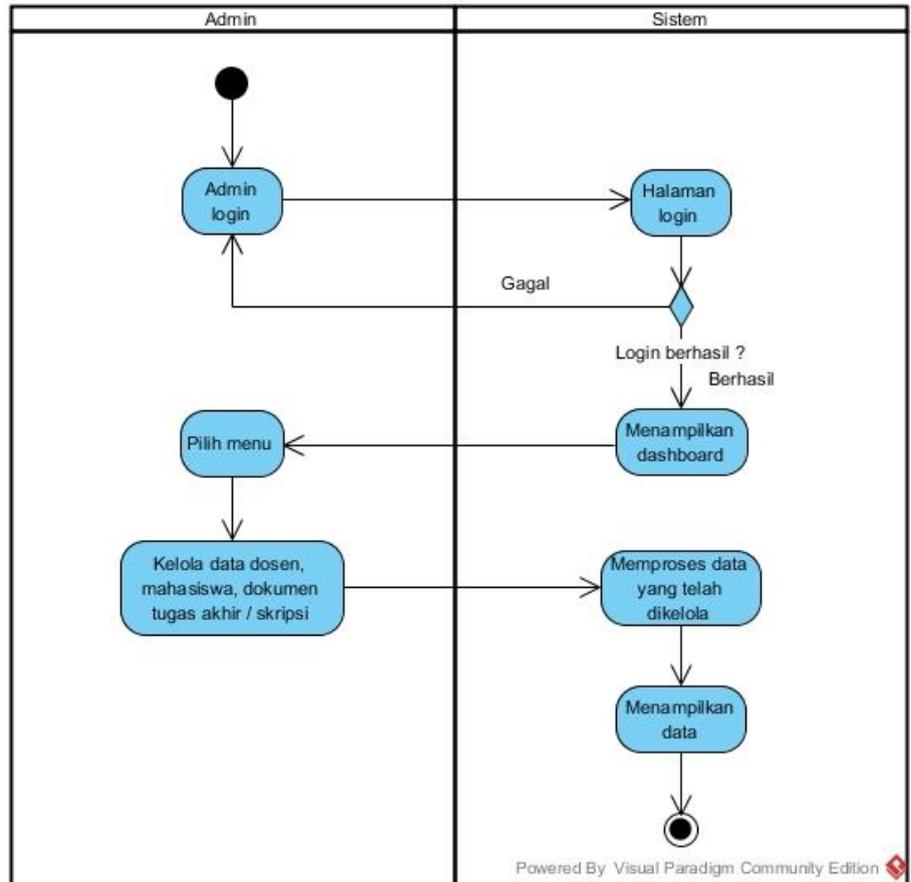


Gambar 3. 7 Activity Diagram *Upload* Naskah Publikasi

#### 5) Diagram Activity Admin Kelola Data

Pada Gambar 3.8 dibawah dijelaskan mengenai diagram activity admin kelola data. Admin pada sistem ini dapat mengelola data mahasiswa, data dosen, data bimbingan. Alur pertama yang dilakukan admin adalah melakukan *login*, apabila *login* berhasil maka akan tampil halaman *dashboard*,

apabila gagal admin akan diarahkan ke halaman *login* lagi. Setelah masuk *dashboard*, admin dapat memilih menu kelola data.



Gambar 3. 8 Activity Diagram Admin Kelola Data

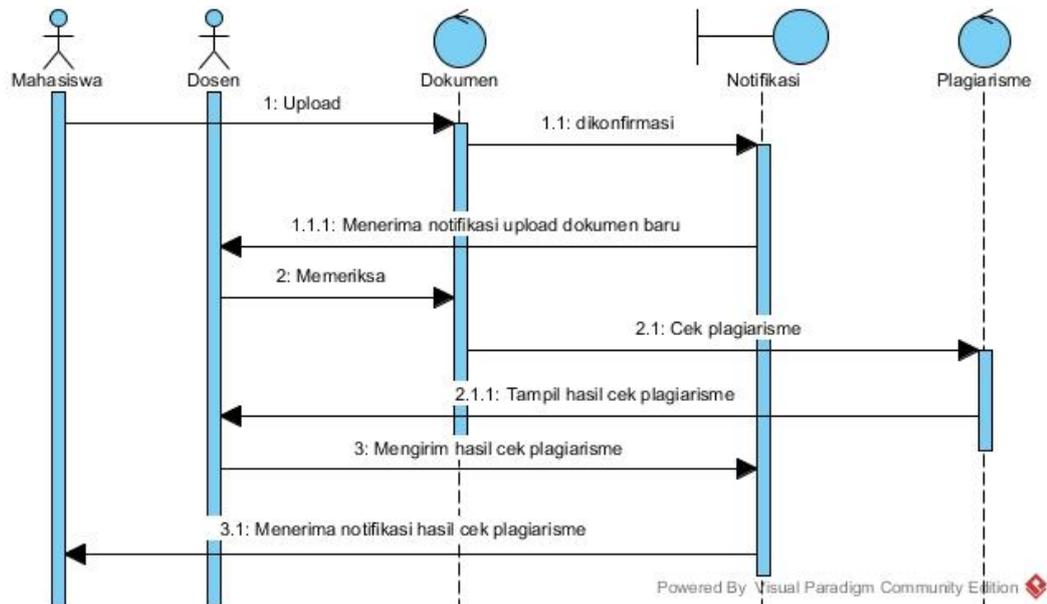
### c. Sequence Diagram Sistem Pendeteksi Dini Plagiarisme

*Sequence Diagram* digunakan untuk memberikan rangkaian pesan antar objek pada aktivitas tertentu yang selanjutnya akan dijalankan oleh sistem. Pada sistem ini terdapat 3 *Sequence Diagram*, yaitu sebagai berikut.

#### 1) Sequence Diagram Cek Plagiarisme

Cek *plagiarisme* dilakukan oleh dua aktor, mahasiswa dan dosen. Mahasiswa melakukan *upload* dokumen ke dosen melalui sistem pendeteksi dini *plagiarisme* dokumen akan diproses sistem dan dosen mendapatkan notifikasi bahwa terdapat dokumen baru yang di *upload* mahasiswa.

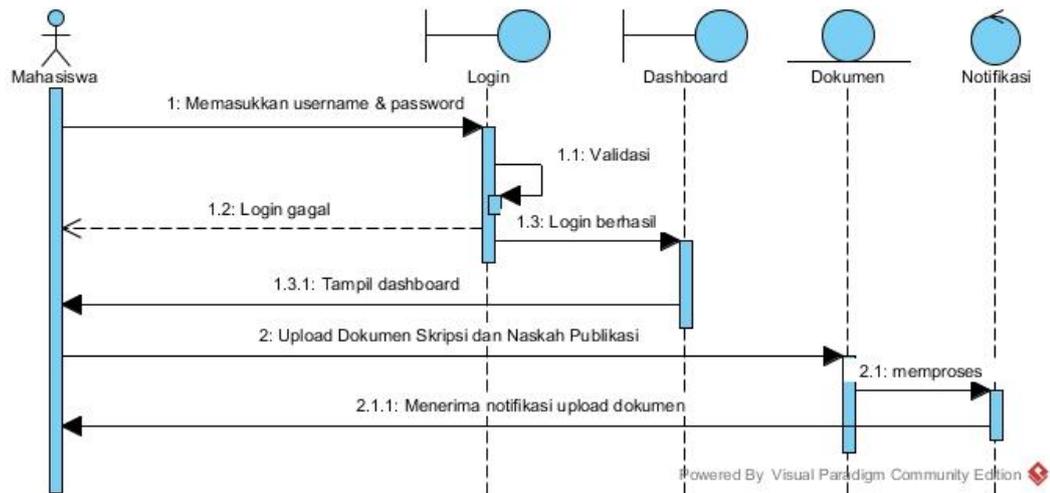
Kemudian dosen memeriksa dokumen dan melakukan cek *plagiarisme*. Sistem melakukan perhitungan cek *plagiarisme*, setelah mendapatkan hasil, selanjutnya hasil akan ditampilkan melalui *pop up* di halaman dosen dan dosen memberi notifikasi hasil ke mahasiswa. *Sequence diagram* cek *plagiarisme* tersaji pada Gambar 3.9 seperti berikut.



Gambar 3. 9 *Sequence Diagram* Cek *Plagiarisme*

## 2) *Sequence Diagram Upload Naskah Publikasi*

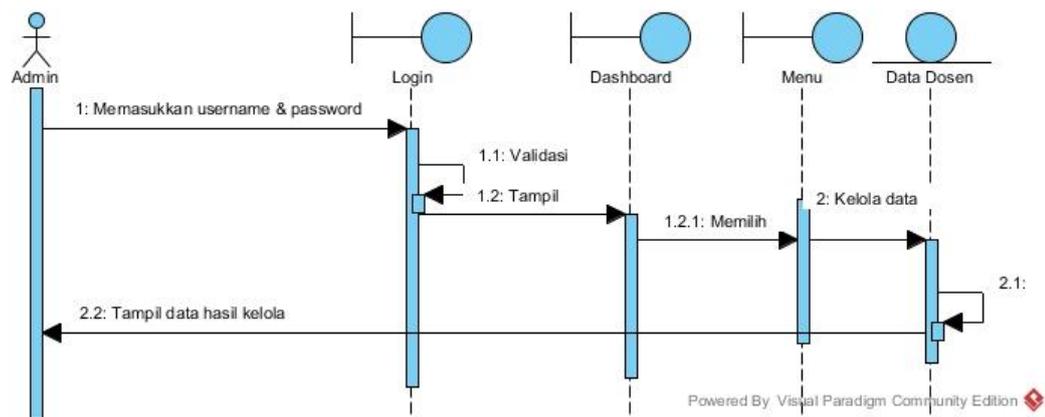
Naskah publikasi diupload oleh mahasiswa yang sudah mengikuti sidang pendaran dan sudah melakukan revisi. Mahasiswa melakukan *login* sistem, apabila *login* gagal maka mahasiswa akan ditampilkan lagi halaman *login* dan jika *login* berhasil maka akan tampil halaman *dashboard* selanjutnya mahasiswa dapat memilih menu *upload* dokumen Naskah Publikasi. Alur *Sequence Diagram* dapat dilihat seperti Gambar 3.10 dibawah ini.



Gambar 3. 10 *Sequence Diagram Upload* Dokumen Naskah Publikasi

### 3) *Sequence Diagram Admin Kelola Data Dosen*

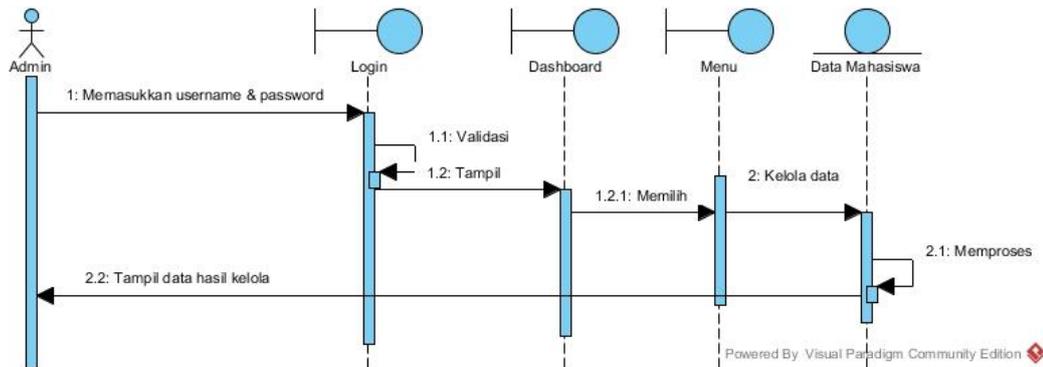
Data dosen dikelola oleh admin didalam sistem. Admin dapat melakukan *login*, kemudian bila berhasil admin akan ditampilkan *dashboard* admin, selanjutnya admin dapat memilih menu untuk mengelola data. Alur diagram disajikan seperti pada Gambar 3.11 berikut



Gambar 3. 11 *Sequence Diagram Admin Kelola Data Dosen*

#### 4) *Sequence Diagram Admin Kelola Data Mahasiswa*

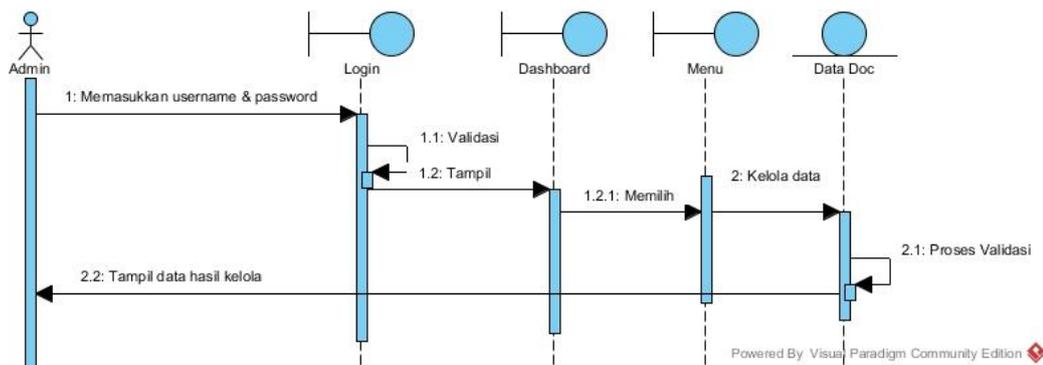
Data mahasiswa dikelola oleh admin didalam sistem. Admin dapat melakukan *login*, kemudian bila berhasil admin akan ditampilkan *dashboard* admin, selanjutnya admin dapat memilih menu untuk mengelola data. Alur diagram disajikan seperti pada Gambar 3.12 berikut



Gambar 3. 12 *Sequence Diagram Admin Kelola Data Mahasiswa*

#### 5) *Sequence Diagram Admin Kelola Data Dokumen*

Data dokumen dikelola oleh admin didalam sistem. Admin dapat melakukan *login*, kemudian bila berhasil admin akan ditampilka *dashboard* admin, selanjutnya admin dapat memilih menu untuk mengelola data. Alur diagram disajikan seperti pada Gambar 3.13 berikut

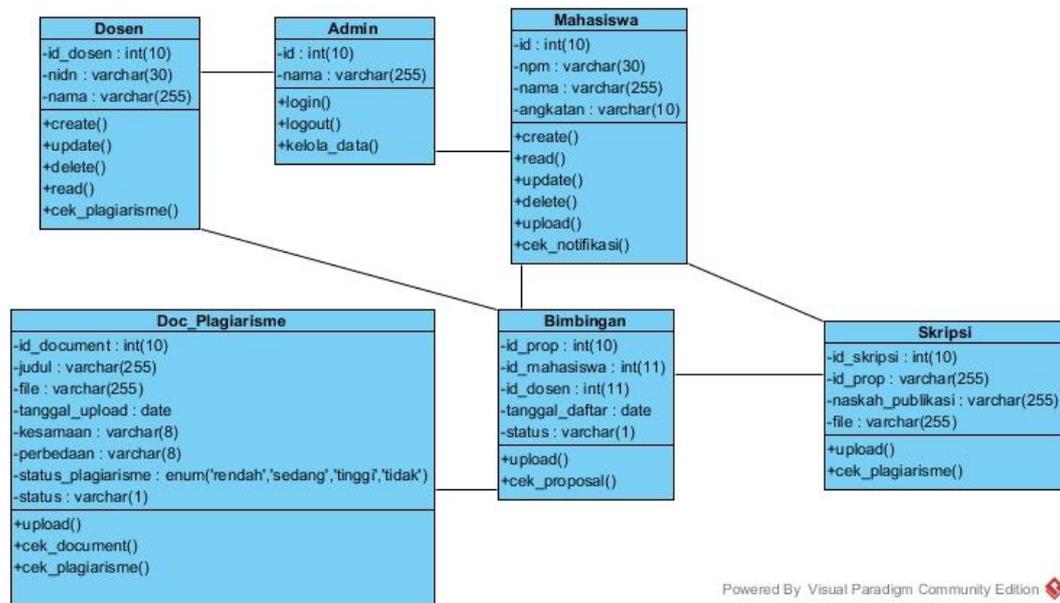


Gambar 3. 13 *Sequence Diagram Admin Kelola Data Dokumen*

#### d. *Class Diagram Sistem Pendeteksi Dini Plagiarisme*

*Class Diagram* menggambarkan struktur dan deskripsi *class*, *package*, dan objek beserta hubungan satu sama lain. Pada sistem ini terdapat 6 *class*, yaitu Admin, Mahasiswa, Dosen, Bimbingan,

Doc\_Plagiarisme, dan Skripsi. *Class Diagram* pada sistem ini dapat dilihat pada Gambar 3.14 .



Gambar 3. 14 *Class Diagram* Sistem Pendeteksi Dini Plagiarisme

### C. Pemodelan Data

Tahap pemodelan data ini meliputi deskripsi data objek, ERD, relasi dan kardinalitas.

#### 1. Deskripsi Data Objek

Deskripsi Data Objek merupakan penjabaran entitas beserta atribut yang ada pada rancangan basis data dari sistem yang dibuat.

Tabel 3. 7 Admin

| Admin |                      |
|-------|----------------------|
| Id    | Kode unik dari admin |
| Nama  | Nama dari admin      |

Tabel 3. 8 Mahasiswa

| Mahasiswa |                            |
|-----------|----------------------------|
| Id_mhs    | Kode unik setiap mahasiswa |
| NPM       | Nomer Pokok Mahasiswa      |
| Nama      | Nama dari mahasiswa        |
| Angkatan  | Tahun angkatan mahasiswa   |

Tabel 3. 9 Dosen

| Dosen    |                      |
|----------|----------------------|
| Id_dosen | Kode unik tiap dosen |
| NIDN     | Nomer Induk Dosen    |
| Nama     | Nama dari dosen      |

Tabel 3. 10 Bimbingan

| Bimbingan      |   |
|----------------|---|
| Id_doct        | Kode unik dari dokumen bimbingan                                  |
| Tanggal_daftar | Tanggal pertama kali bimbingan                                    |
| Status         | Status pada sistem untuk mengetahui valid tidaknya data bimbingan |

Tabel 3. 11 Doc\_Plagiarisme

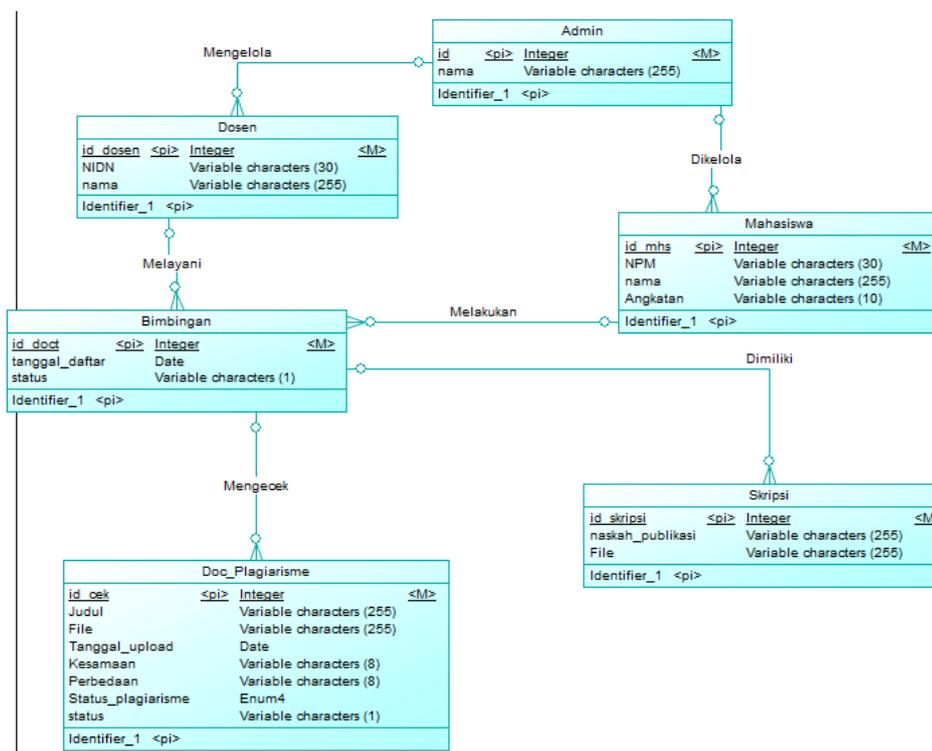
| Doc_Plagiarisme    |  |
|--------------------|--|
| Id_cek             | Kode unik setiap melakukan cek plagiarisme   |
| Judul              | Judul proposal tugas akhir / skripsi   |
| File               | <i>Field</i> untuk menyimpan file  |
| Tanggal_upload     | Tanggal pada saat <i>upload</i>  |
| Kesamaan           | Persentase hasil kesamaan dokumen pada perhitungan cek <i>plagiarisme</i>            |
| Perbedaan          | Persentase hasil perbedaan dokumen pada perhitungan cek <i>plagiarisme</i>           |
| Status_plagiarisme | Hasil yang menyatakan status <i>plagiarisme</i> 'ringan', 'sedang', 'berat', 'tidak' |

Tabel 3. 12 Skripsi

| Skripsi          |                                   |
|------------------|-----------------------------------|
| Id_skripsi       | Kode unik dokumen skripsi         |
| Naskah_Publikasi | Judul dokumen                     |
| File             | <i>Field</i> untuk menyimpan file |

## 2. EER (Enhanced Entity Relationship) Diagram

EER (Enhanced Entity Relationship) Diagram adalah ERD yang diberikan kemampuan untuk memiliki *supertype*, *subtype* dan *instance*. EER pada sistem ini disajikan pada Gambar 3.15.



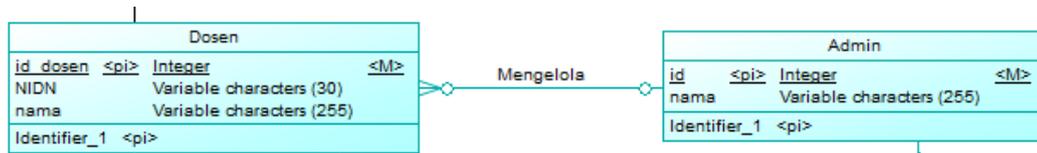
Gambar 3. 15 EER Sistem Pendeteksi Dini Plagiarisme

### 3. Relasi dan Kardinalitas

Relasi dan kardinalitas merupakan tahapan untuk memetakan model deskripsi data objek ke model basis data relasional.

#### a. Relasi Dosen dan Admin

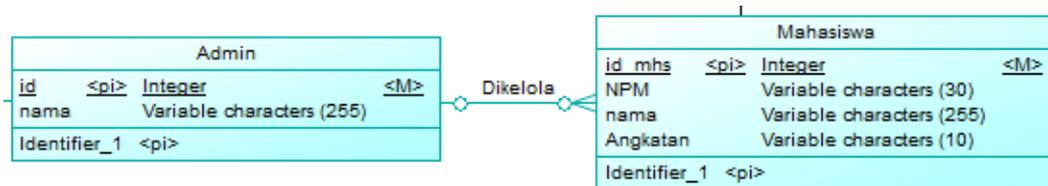
Entitas Dosen dengan admin memiliki hubungan 1:N karena satu admin dapat mengelola data dari beberapa dosen. Relasi disajikan pada gambar 3.16 berikut.



Gambar 3. 16 Relasi Dosen dan Admin

#### b. Relasi Mahasiswa dan Admin

Entitas mahasiswa dengan admin memiliki hubungan 1:N karena satu admin dapat mengelola data dari beberapa mahasiswa. Relasi disajikan pada Gambar 3.17 dibawah ini.



Gambar 3. 17 Relasi Mahasiswa dan Admin

#### c. Relasi Mahasiswa dan Bimbingan

Entitas mahasiswa dengan bimbingan memiliki hubungan 1:N, karena satu mahasiswa dapat melakukan beberapa kali bimbingan. Relasi disajikan pada Gambar 3.18 dibawah ini.



Gambar 3. 18 Mahasiswa dan Bimbingan

d. Relasi Dosen dan Bimbingan

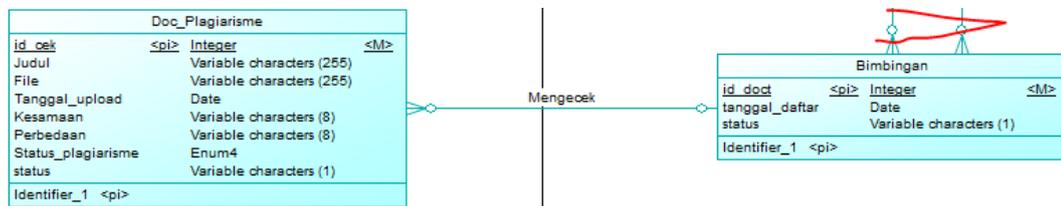
Entitas dosen dengan bimbingan memiliki hubungan 1:N, karena satu dosen dapat melayani beberapa kali bimbingan. Relasi disajikan pada Gambar 3.19 berikut.



Gambar 3. 19 Relasi Dosen dan Bimbingan

e. Relasi Bimbingan dan Doc\_plagiarisme

Entitas Bimbingan dengan Doc\_plagiarisme memiliki hubungan 1:N, karena pada satu bimbingan dapat dilakukan pengecekan *plagiarisme* beberapa kali. Relasi disajikan pada Gambar 3.20 berikut.



Gambar 3. 20 Relasi Bimbingan dan Doc\_plagiarisme

f. Relasi Bimbingan dan Skripsi

Entitas Bimbingan dengan skripsi memiliki hubungan 1:N, karena beberapa dokumen skripsi dimiliki pada saat satu bimbingan dan untuk mengetahui dokumen naskah publikasi yang *diupload* dimiliki oleh mahasiswa siapa dan bimbingan dengan dosen siapa. Relasi disajikan pada Gambar 3.21.

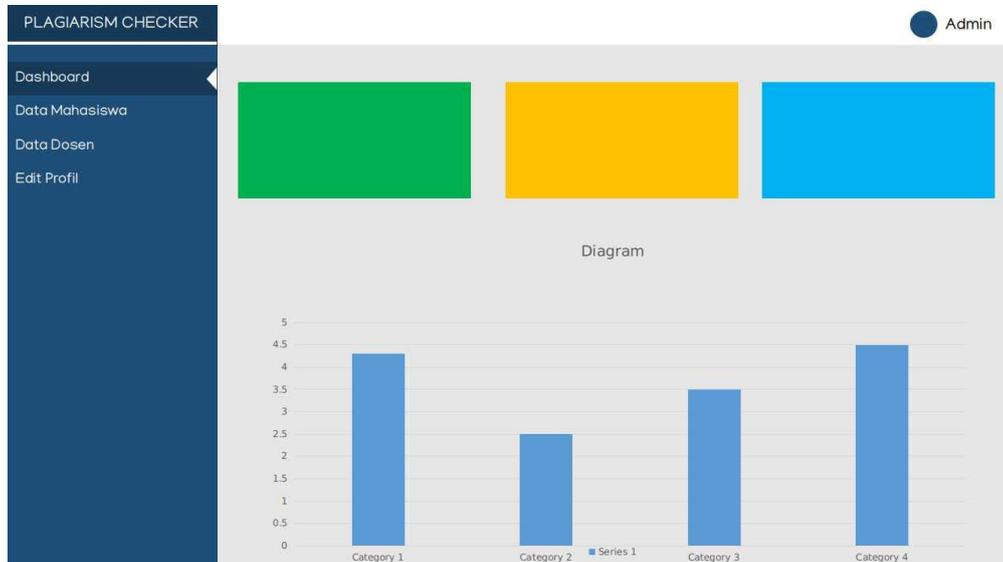


Gambar 3. 21 Relasi Bimbingan dan Skripsi

## D. Perancangan *Interface*

### 1. Rancangan Halaman *Dashboard Admin*

Halaman *dashboard* admin digunakan sebagai halaman utama admin untuk memonitoring data dosen, data mahasiswa, dan data bimbingan dapat dilihat pada Gambar 3.22.



Gambar 3. 22 Rancangan Halaman Dashboard Admin

### 2. Rancangan Halaman Data Dosen Pada Admin

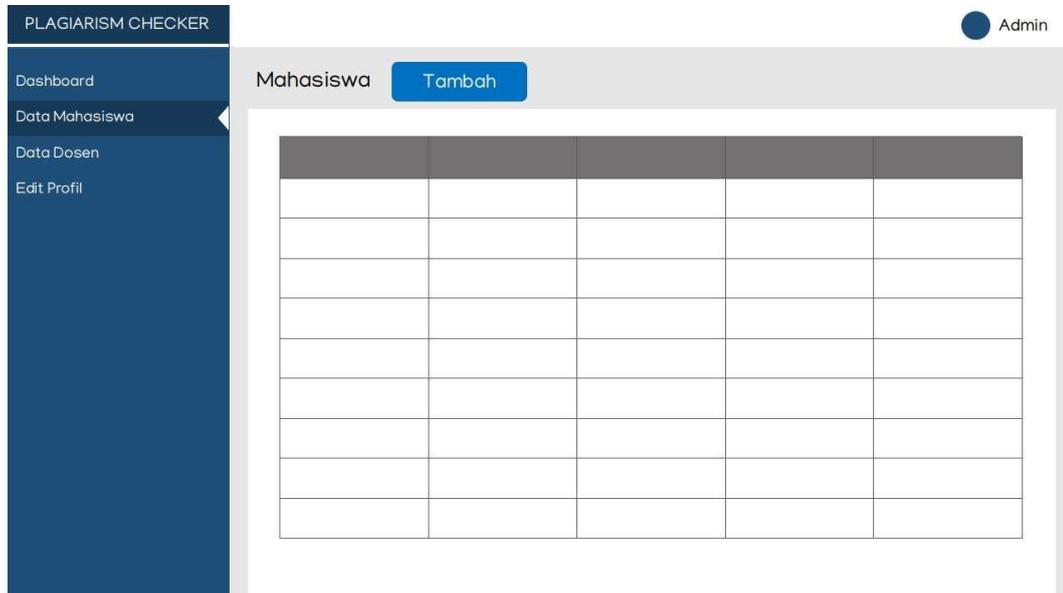
Halaman data dosen menampilkan data dosen yang sedang membimbing mahasiswa pada tahun ajaran tersebut. Dapat dilihat pada Gambar 3.23

| Dosen |  |  |  |  |
|-------|--|--|--|--|
|       |  |  |  |  |
|       |  |  |  |  |
|       |  |  |  |  |
|       |  |  |  |  |
|       |  |  |  |  |
|       |  |  |  |  |
|       |  |  |  |  |
|       |  |  |  |  |
|       |  |  |  |  |

Gambar 3. 23 Rancangan Halaman Data Dosen Pada Admin

### 3. Halaman Data Mahasiswa Pada Admin

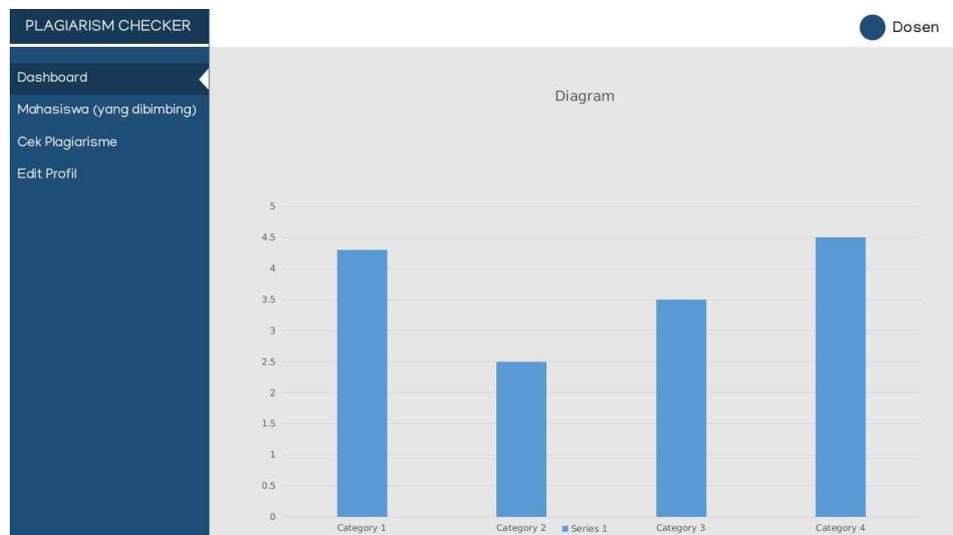
Halaman data mahasiswa menampilkan data mahasiswa aktif yang mengambil sks tugas akhir / skripsi pada tahun ajaran tersebut. Dapat dilihat pada Gambar 3.24



Gambar 3. 24 Rancangan Data Mahasiswa Pada Admin

### 4. Halaman *Dashboard* Dosen

Halaman *dashboard* dosen sebagai halaman utama dosen menampilkan mahasiswa yang sedang dibimbing. Dapat dilihat pada Gambar 3.25



Gambar 3. 25 Halaman *dashboard* Dosen

## 5. Halaman Cek *Plagiarisme*

Halaman ini berfungsi sebagai tempat dosen melakukan cek *plagiarisme* pada dokumen milik mahasiswa yang sedang dibimbing. Pada tabel bagian bawah berisi dokumen pembanding Naskah Publikasi. Dapat dilihat pada Gambar 3.26

| DOKUMEN PEMBANDING                  |  |
|-------------------------------------|--|
| <input type="checkbox"/>            |  |
| <input checked="" type="checkbox"/> |  |
| <input checked="" type="checkbox"/> |  |
| <input type="checkbox"/>            |  |
| <input checked="" type="checkbox"/> |  |
| <input type="checkbox"/>            |  |
| <input type="checkbox"/>            |  |
| <input type="checkbox"/>            |  |

Gambar 3. 26 Halaman Cek Plagiarisme

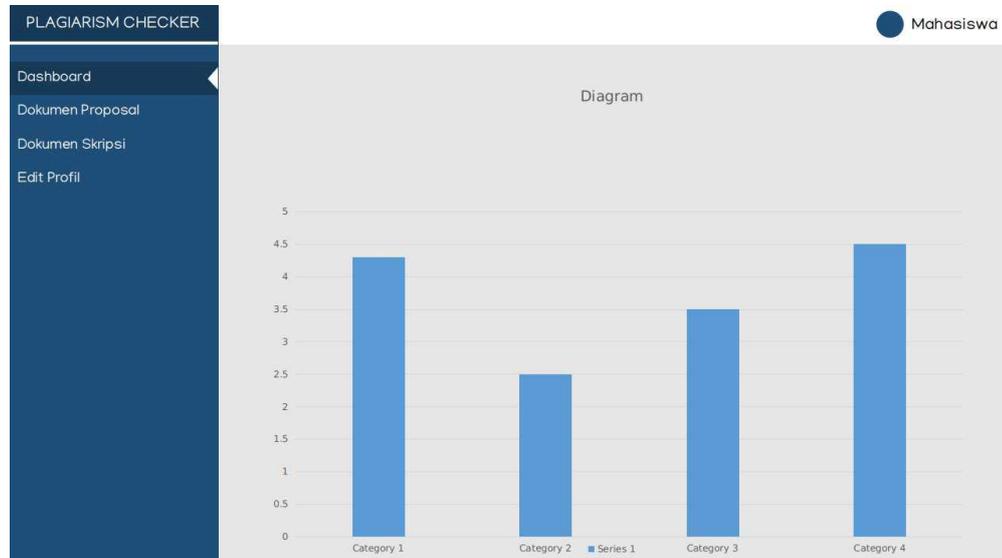
## 6. Halaman Login

Halaman login digunakan untuk menampilkan menu login yang akan digunakan admin, mahasiswa, dosen untuk login kedalam sistem. Dapat dilihat pada Gambar 3.27.

Gambar 3. 27 Halaman Login

## 7. Halaman *Dashboard* Mahasiswa

Halaman ini digunakan untuk halaman utama mahasiswa yang berisi data berapa kali melakukan revisi pada kegiatan karya tulis ilmiahnya. Dapat dilihat rancangannya pada Gambar 3.28



Gambar 3. 28 Halaman Dashboard Mahasiswa

## 8. Halaman Data Proposal

Halaman ini adalah untuk menampilkan halaman upload dokumen proposal. Dapat dilihat pada Gambar 3.29.

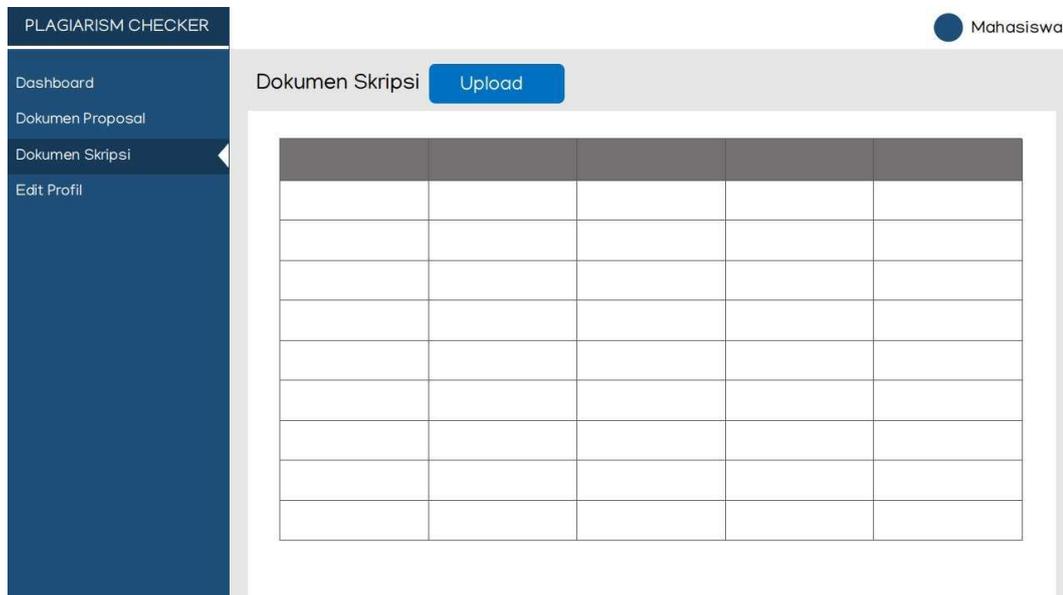
The screenshot shows a dashboard for a student. On the left is a dark blue sidebar with the text 'PLAGIARISM CHECKER' at the top. Below it are menu items: 'Dashboard', 'Dokumen Proposal', 'Dokumen Skripsi', and 'Edit Profil'. The main content area is light gray and contains a section titled 'Dokumen Proposal' with a blue 'Upload' button. Below this is a table with a dark gray header row and 10 empty rows. In the top right corner of the dashboard, there is a user profile icon and the name 'Mahasiswa'.

| Dokumen Proposal |  |  |  |  |
|------------------|--|--|--|--|
|                  |  |  |  |  |
|                  |  |  |  |  |
|                  |  |  |  |  |
|                  |  |  |  |  |
|                  |  |  |  |  |
|                  |  |  |  |  |
|                  |  |  |  |  |
|                  |  |  |  |  |
|                  |  |  |  |  |
|                  |  |  |  |  |

Gambar 3. 29 Halaman Data Proposal

## 9. Halaman Data Skripsi

Halaman ini berfungsi untuk menampilkan menu upload Naskah publikasi bagi mahasiswa yang sudah mengikuti seminar pendadaran dan sudah revisi. Dapat dilihat pada Gambar 3.30



Gambar 3. 30 Halaman Data Skripsi

## BAB VI

### KESIMPULAN DAN SARAN

#### A. Kesimpulan

Dari pembahasan yang sudah diuraikan maka dapat disimpulkan sebagai berikut :

1. Pada pengujian dengan menggunakan data real dokumen yang memiliki perbedaan, dari dokumen 1 dengan *count words* berjumlah 629 dan dokumen 2 dengan *counts words* 128 diketahui nilai kesamaan 13,50% dan nilai perbedaan 86,50%.
2. Pada pengujian efektifitas kecepatan perhitungan, dokumen sumber dengan *count words* 4405 diuji dengan 3 dokumen pembanding dengan total *count words* 13465 memiliki kecepatan menghitung selama 3,57 detik dan jika jumlah dokumen ditambah maka lama perhitungan juga akan naik.
3. Kecepatan perhitungan algoritma *Levenshtein Distance* tergantung pada jumlah total *words* yang terkandung dalam dokumen sumber dan dokumen pembanding. Semakin banyak *words* yang terkandung maka akan membutuhkan waktu yang sedikit lama.
4. Setelah menerapkan algoritma *Levenshtein Distance* pada sistem, proses *parsing* dokumen berbentuk **.pdf** ke dalam bentuk *string* mempengaruhi kecepatan waktu perhitungan cek *similarity*.

#### B. Saran

Dari hasil kesimpulan diatas, penggunaan Sistem Pendeteksi Dini *Plagiarisme* untuk mengetahui tingkat persentase kesamaan dan perbedaan isi dokumen memiliki potensi besar untuk dapat dikembangkan lagi agar lebih baik, oleh karena itu beberapa saran yang dapat digunakan sebagai pertimbangan yang sekiranya dapat berguna untuk masing-masing sistem :

1. File dokumen yang diupload tidak hanya berformat **.pdf**, tapi bisa ditambahkan format **.doc**, **.docx** agar lebih cepat melakukan perhitungan tanpa harus melalui tahap *parsing* dokumen.

- 
2. Hasil perhitungan tidak hanya menampilkan besar persentase kesamaan dan perbedaan dari isi dokumen, melainkan juga dapat mengetahui kata atau kalimat mana saja yang memiliki kesamaan.

## DAFTAR PUSTAKA

- DocForge. (2014). *Web Application Framework*. Diambil kembali dari DocForge:  
[http://docforge.com/wiki/Web\\_application\\_framework](http://docforge.com/wiki/Web_application_framework)
- Dwi Susanto, A. B. (2016). Deteksi Plagiat Dokumen Tugas Daring Laporan Praktikum Mata Kuliah Desain Web Menggunakan Metode Naive Bayes. *Nusantara Journal of Computers and its Applications*, 1.
- F, L. J. (2014). *PHP Metrics*. Diambil kembali dari PHP Metrics:  
<http://www.phpmetrics.org>
- Hamidillah Ajie, A. S. (2017). Aplikasi Pendeteksi Dugaan Awal Plagiarisme Pada Tugas SIswa dan Mahasiswa Berdasarkan Kemiripan Isi Teks Menggunakan Algoritma Levenshtein Distance. *Jurnal Pinter Vol. 1 No. 1 Juni 2017*, 24.
- Harlian, M. (2015, Juni 4). *Lecturer Pens*. Diambil kembali dari  
<http://iwanarif.lecturer.pens.ac.id/>:  
<http://iwanarif.lecturer.pens.ac.id/kuliah/dm/6Text%20Mining.pdf>
- KEMENRISTEKDIKTI, P. (2018, 11 01). *Grafik Jumlah Perguruan Tinggi*. Diambil kembali dari Pangkalan Data Pendidikan Tinggi Kementerian Riset, Teknologi dan Perguruan Tinggi:  
<https://forlap.ristekdikti.go.id/perguruantinggi/homegraphpt>
- LLC, M. S. (2015). *Source Code Size Metrics*. Diambil kembali dari MSquared Technologies:  
[http://msquaredtechnologies.com/m2rsm/docs/rsm\\_metrics\\_narration.htm](http://msquaredtechnologies.com/m2rsm/docs/rsm_metrics_narration.htm)
- M, A. N. (2010). Implementasi Algoritma Levenshtein Distance dan Metode Empiris untuk menampilkan saran perbaikan kesalahan pengetikan dokumen berbahasa Indonesia. *Skripsi*.
- Mufti Ari Bianto, S. R. (2018). Perancangan Sistem Pendeteksi Plagiarisme Terhadap Topik Penelitian Menggunakan Metode K-Means Clustering dan Model Bayesian. *Seminar Nasional Teknologi Informasi dan Multimedia 2018*, 19.
- Na'firul Hasna Ariyani, S. R. (2016). Aplikasi Pendeteksi Kemiripan Isi Teks Dokumen Menggunakan Metode Levenshtein Distance. *semanTIK Vol. 2, No. 1*, 279.
- Nugroho, A. (2015). *Analisis dan Perancangan Sistem Informasi dengan Metodologi Berorientasi Objek*. Bandung: Informatika.
- Panji Novantara, O. P. (2018). Implementasi Algoritma Jaro-Winkler Distance Untuk Sistem Pendeteksi Plagiarisme Pada Dokume Skripsi. *Jurnal Buffer Informatika*, 8.

- PDDIKTI, F. (2018, 11 01). *Pencarian Perguruan Tinggi*. Diambil kembali dari Pangkalan Data Pendidikan Tinggi Kementerian Riset Teknologi dan Pendidikan Tinggi: <https://forlap.ristekdikti.go.id/perguruantinggi/search>
- Rocky Yefrenes Dillak, F. L. (2016). Sistem Deteksi Dini Plagiarisme Tugas Akhir Mahasiswa Menggunakan Algoritma N-Grams dan Winnowing. *Jurnal Ilmiah Flash, [S.1]*, v. 2, n. 1, p. 12-18, 12.
- Rudianto, A. M. (2011). Pemrograman Web Dinamis Menggunakan PHP. *Andi Offset*.
- Sastroasmoro, S. (2007). Beberapa Catatan Tentang Plagiarisme. *Majalah Kedokteran Indonesia*, Volume 57.
- Sora. (2018, November 07). *Mengetahui Pengertian Dokumen dan Dokumentasi*. Diambil kembali dari Pengertianku: <http://www.pengertianku.net/2014/09/mengetahui-pengertian-dokumen-dan-dokumentasi.html>
- Tudesman, E. O. (2014). Sistem Deteksi Plagiarisme Dokumen Bahasa Indonesia Menggunakan Metode Vector Space Model. *Jurnal STMIK GI MDP*, Page 2.
- Waliyanto. (2000). Sistem Basis Data Analisis dan Pemodelan Data. *J&J Learning*.
- Widodo, P. P. (2011). UML (Unified Modeling Language). *Herlawati*.