**SKRIPSI**

**SISTEM PENDETEKSI PLAGIARISME PROPOSAL TUGAS AKHIR MAHASISWA DENGAN ALGORITMA *RABIN KARP*.**

**(Studi Kasus: Jurusan Teknik Informatika Universitas Halu Oleo)**

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Sarjana Teknik

****

# NUR’AZIZA TADJUDDIN

**E1E1 17 023**

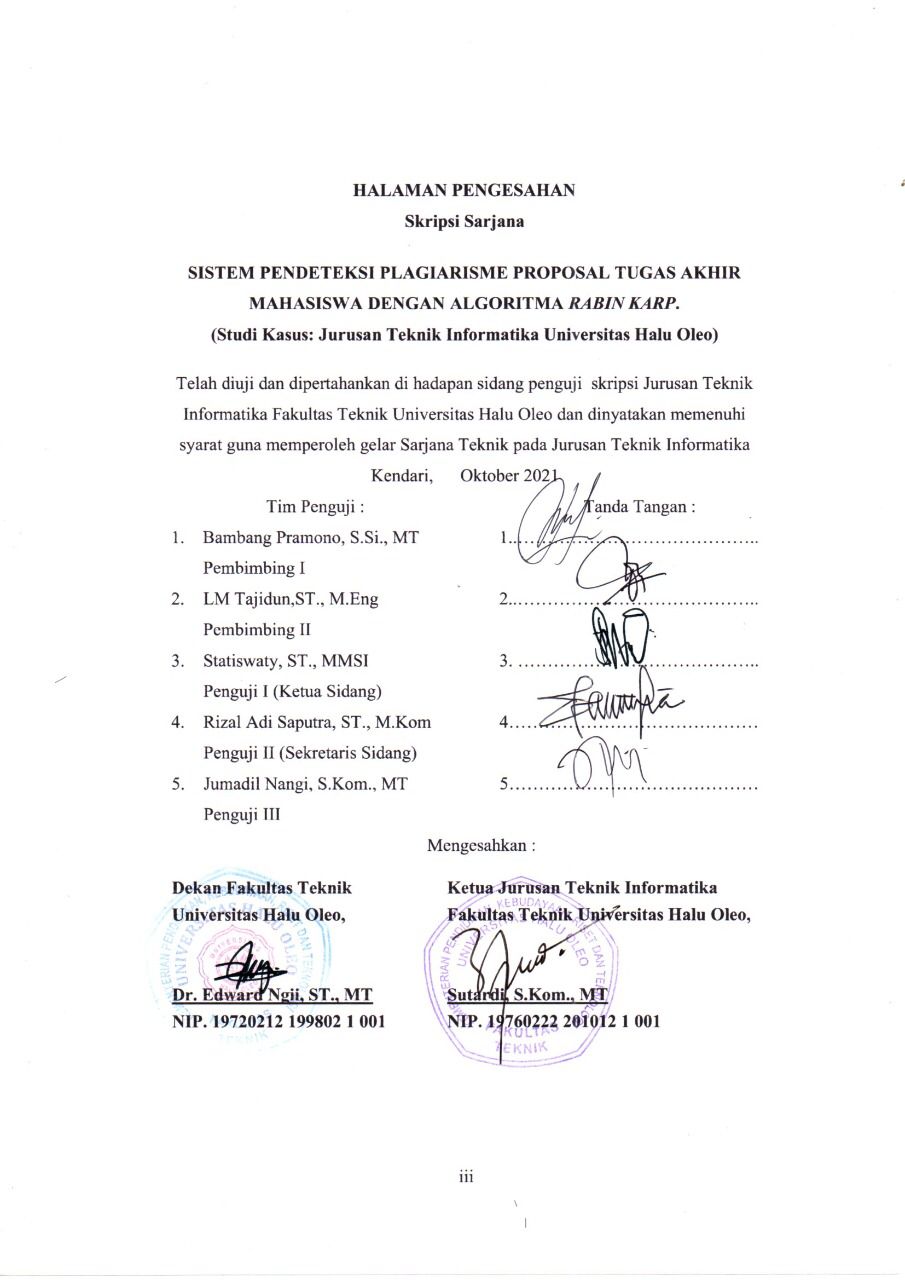
**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS HALU OLEO**

**2021**

# HALAMAN PENGESAHAN

**HALAMAN PENGESAHAN**

# PERNYATAAN

# INTISARI

Nur’ Aziza Tadjuddin, E1E117023

**SISTEM PENDETEKSI PLAGIARISME PADA PROPOSAL TUGAS AKHIR MAHASISWA DENGAN ALGORITMA *RABIN KARP*.**

Skripsi, Fakultas Teknik, 2021.

Kata kunci : Proposal tugas akhir dan Algoritma *Rabin-Karp.*

Plagiarisme sering di jumpai dalam sektor akademis maupun non akademis, Dalam sektor akademis, plagiarisme di anggap sebagai tindak pidana serius karena di anggap pengambilan karangan, pendapat, ide dan gagasan orang lain. Salah satu plagiarisme yang sering terjadi di kalangan mahasiswa adalah plagiarisme di dalam penulisan Proposal tugas akhir. Proposal tugas akhir mahasiswa bertujuan untuk menyampaikan rancangan penelitiannya kepada dosen pembimbing agar disetujui sebelum membuat skripsi sehingga sistem ini dapat di gunakan oleh dosen pembimbing untuk mendeteksi tingkat plagiarisme proposal mahasiswa yang melakukan bimbingan sebelum melakukan ujian seminar proposal. Tindakan plagiarisme secara perlahan harus di cegah dan dihilangkan dengan melakukan pendeteksian plagiat secara manual maupun dengan memanfaatkan metode pencocokan *string*. Dengan demikian melakukan pendeteksian plagiarisme secara manual sangat tidak efektif sehingga Algoritma *Rabin Karp* dapat menjadi solusi untuk menyelesaikan hal tersebut. Algoritma *Rabin Karp* merupakan algoritma pencarian *string* yang menggunakan fungsi hashing untuk membandingkan *string* yang dicari (m) dengan *string* yang dibandingkan (n).

# *ABSTRACT*

Nur’ Aziza Tadjuddin, E1E117023

**PLAGIARISM DETECTION SYSTEM ON STUDENT FINAL PROPOSAL WITH RABIN KARP ALGORITHM.**

*Thesis, Faculty of Engineering, 2021*

Keywords: Final project proposal and Rabin-Karp Algorithm.

           Plagiarism is often encountered in the academic and non-academic sectors. In the academic sector, plagiarism is considered a serious crime because it is considered taking other people's essays, opinions, ideas, and ideas. One of the plagiarism that often occurs among students is plagiarism in writing the final project proposal. The student's final project proposal aims to submit the research design to the supervisor for approval before making the thesis so that this system can be used by the supervisor to detect the level of plagiarism in student proposals who conduct guidance before conducting the proposal seminar exam. Plagiarism must be slowly prevented and eliminated by detecting plagiarism manually or by using string matching methods. Thus, manual plagiarism detection is not very effective, so the Rabin Karp Algorithm can be a solution to solve this problem. Rabin Karp algorithm is a string search algorithm that uses a hashing function to compare the searched string (m) with the compared string (n).

# HALAMAN PERSEMBAHAN



Alhamdulillah puji syukur kehadirat Allah SWT yang telah memberikan rahmat dan hidayah-Nya kepada penyusun, sehingga berkat ridho-Nyalah laporan Tugas Akhir yang berjudul “Sistem Pendeteksi Plagiarisme Pada Proposal Tugas Akhir Mahasiswa Dengan Algoritma *Rabin Karp* (Studi Kasus : Jurusan Teknik Informatika Fakultas Teknik Universitas Halu Oleo)” dapat diselesaikan dengan baik. Laporan tugas akhir ini di susun sebagai syarat menempuh jenjang Strata-1 pada Jurusan S1 Teknik Informatika, Fakultas Teknik Universitas Halu Oleo.

Tugas akhir ini merupakan sarana bagi mahasiswa untuk mengaplikasikan ilmu dan pengetahuan yang telah di dapat selama mengikuti perkuliahan di Jurusan S1 Teknik Informatika, Fakultas Teknik Universitas Halu Oleo untuk mendapatkan satu pengetahuan baru dari hasil penelitian yang dilakukan.

Untuk dapat menyelesaikan laporan Tugas Akhir ini, tentunya tidak lepas dari hambatan dan rintangan, namun berkat bantuan moril maupun materil dari berbagai pihak, akhirnya Tugas Akhir ini dapat diselesaikan dengan baik. Untuk itu penulis mengucapkan terima kasih kepada orang-orang yang sangat saya kasihi dan sayangi, terkhusus kepada kedua orang tua saya Bapak H. Tadjuddin, S.Pd sosok panutan dalam keluarga sifat baik dan dermawannya yang selalu bapak tunjukkan kepada anak-anaknya, selalu memberikan semangat dan dukungan dalam setiap Langkah aziza dan ummi Hj. Normah, sosok perempuan terbaik yang saya kenal di dunia ini, cinta dan kasihnya kepada anak-anaknya yang tiada batas,ketulusan dari dalam dirinya yang tidak pernah saya temukan pada diri orang lain . karena do’a dan pengorbanan dari bapak dan ummi aziza bisa sampai pada tahap ini. . Saudara saudari saya Nur Fadillah Tadjuddin,S.Kep Ners sosok kakak perempuuan yang selalu memberikan dukungan dan semangat kepada adik-adiknya dengan sifat loyalnya yang sangat besar Bersama dengan kakak ipar saya Ahmad Syahid Ramadan,S.Si terimakasih selalu menjadi kakak-kakak yang baik dan selalu membantu aziza. Adik saya Nur Fadli Tadjuddin yang sifatnya malas pusing tapi asli perhatian dengan semua orang khususnya pada keluarga. Tante saya Normiati,S.Pdi dan Om Ririn, terimakasih telah menjadi orang tua kedua untuk aziza selama kurang lebih 2 tahun selama kuliah dan menjadi tempat pulang paling nyaman setelah Lelah yang Panjang yang dirasakan di luar rumah. Sepupu saya Maya Magfirah yang menjadi partner terbaik selama 3 tahun, partner bercerita, parter jalan partner begadang partner gabut gabutan pokoknya argh makasih siss. Mami alias kk uni yang selalu menjadi tempat weekendnya aziza dan maya makasih mami dan semua keluaga yang tidak saya sebutkan namanya.

Untuk Onsight 017, terimakasih telah menjadi sobat-sobat yang luar biasa selama 4 tahun, Selalu membantu, dan saling mendukung. Ayok semangat yaa untuk menyelesaikan tugas akhir ini. Ciwi ciwiku, marlina, argghh nda tau mau bilang apa untuk deskripsikan anak ini dia yang tau semua sifat baik buruknya aziza tapi tetap ada dalam setiap keadaan eakk yang selalu ayok kalau di ajak kemanapun terimakasih marlin. Unii, si aslab yang baikk hati tidak sombong tapi malas menabung karena ngemill adalah nomor satu haha terimakasih unycan. Ismi si gadis pendiam tapi sekali bicara kadang nyeri sedap haha makasi ismi. Ice si ibu owner rumgas gadis yg super duper sibuk tapi loyal dan baik hatiii makasih cee. Ayuu partner seperskripsian yang selalu menangkan azizah saat panik selamat untuk kita berdua yaah. Pitii alias fitnawati yang paling cerewet kalau liat kita malas malasan atau tunda-tunda kerja TA makasih piti. Dillah si gadis baik yang selalu menawarkan bantuannya makasih dilla. Indra yang di tengah tengah sibuknya kerja TA video call adalah solusi penghilang capeknya haha. Iklil si baik hati sahabatnya ayu dari lahir haha. Annisa, atry, yang selalu berbagi cerita yang bisa di jadikan pelajaran hidup, sindi, cindii, milwat, vina, gita, mustika, anggi, ratna, terimakasih atas semua bantuannya ciwi-ciwi. Tim CV.Konsultas Indonesia, Ikbal yang sangat membantu dalam menyelesaikan TA ini maaf selalu saya teror kalau ada erornya sistem wkwk, makasih ball. Fadel, makasih del banyak membantu selama 4 tahun ini semoga selalu menjadi fadel yang baik hati dan tidak sombong, pak dirut danil yang selalu motifasi biar selesakan TA secepatnya walaupun dengan cara yang pareare haha. Gagas si designernya kindo yang paling suka izin kalau dipanggil kumpul sm pak kur wkwk. Pokoknya Terimakasih untuk semua orang yang ada dalam proses ini.

# KATA PENGANTAR

Puji syukur kami panjatkan kehadirat Allah SWT. Karena atas rahmat dan ridho-Nya lah penulis dapat menyelesaikan skripsi ini sebagai persyaratan dalam menyelesaikan studi S-1 pada Jurusan Teknik Informatika Fakultas Teknik Universitas Halu Oleo dengan judul “Sistem Pendeteksi Plagiarisme Pada Proposal Tugas Akhir Mahasiswa Dengan Algoritma *Rabin Karp* ”. Dalam penelitian dan penyusunan skripsi, penulis mendapatkan bantuan baik secara teknis maupun non teknis berupa bimbingan, arahan maupun bantuan lainnya dari berbagai pihak. Untuk itu, penulis ingin mengucapkan terima kasih kepada :

1. Bapak Dr. Muhammad Zamrun Firihu, S.Si. M.Si., M.Sc. selaku Rektor Universitas Halu Oleo.
2. Bapak Dr. Edward Ngii, S.T., M.T. selaku Dekan Fakultas Teknik Universitas Halu Oleo.
3. Bapak Sutardi, S.Kom., M.T. selaku Ketua Jurusan S-1 Teknik Informatika Fakultas Teknik Universitas Halu Oleo.
4. Bapak Bambang Pramono, S.Si., MT. sebagai Pembimbing I yang telah banyak memberikan saran, petunjuk, ilmu pengetahuan, dukungan dan motivasi sehingga skripsi ini dapat terselesaikan.
5. Bapak LM Tajidun, ST., M.Eng. sebagai Pembimbing II yang telah banyak memberikan saran, petunjuk, ilmu pengetahuan, dukungan dan motivasi sehingga skripsi ini dapat terselesaikan.
6. Ibu Statiswaty, ST., MSI. selaku Dosen Penguji atas koreksi dan sarannya mulai dari ujian proposal, ujian seminar hasil dan ujian akhir serta bapak Rizal Adi Saputra, S.T., M.Kom. selaku Dosen Penguji atas koreksi dan sarannya saat seminar proposal hingga ujian skripsi dan Bapak Jumadil Nangi, S.Kom., MT. selaku Dosen Penguji atas koreksi dan sarannya mulai dari ujian seminar hasil sampai ujian akhir (skripsi).
7. Dosen serta para staf Fakultas Teknik Universitas Halu Oleo, khususnya pada Jurusan Teknik Informatika atas bimbingan dan bantuannya.
8. Kedua orang tuaku, yang selalu mendoakan dan memberikan dukungan serta kasih sayang yang tiada tara.
9. Kakak, Adik dan semua keluarga saya yang selalu memberikan dukungan dan semangat dalam menyelesaikan tugas akhir ini.
10. Teman - teman Teknik Informatika angkatan 2017 yang telah memberikan semangat, motivasi, bantuan dan doanya.
11. Serta kakak-kakak Angkatan 2015, 2016 dan adik-adik 2018, 2019 dan 2020 yang telah memberikan bantuan dan doanya.

Penulis memohon maaf atas segala kekurangan yang terdapat di dalam skripsi ini. Semoga penulisan skripsi ini bermanfaat untuk pengembangan ilmu pengetahuan dan untuk semua pihak yang bersangkutan.

Kendari, Oktober 2021

Penulis,

Nur Aziza Tadjuddin

# DAFTAR ISI

[NUR’AZIZA TADJUDDIN i](#_Toc84866149)

[HALAMAN PENGESAHAN ii](#_Toc84866150)

[PERNYATAAN iv](#_Toc84866151)

[INTISARI v](#_Toc84866152)

[*ABSTRACT* vi](#_Toc84866153)

[HALAMAN PERSEMBAHAN vii](#_Toc84866154)

[KATA PENGANTAR ix](#_Toc84866155)

[DAFTAR ISI xi](#_Toc84866156)

[DAFTAR TABEL xiii](#_Toc84866157)

[DAFTAR GAMBAR xiv](#_Toc84866158)

[BAB I PENDAHULUAN 1](#_Toc84866159)

[1.1 Latar Belakang 1](#_Toc84866160)

[1.2 Rumusan masalah 2](#_Toc84866161)

[1.3 Batasan Masalah 2](#_Toc84866162)

[1.4 Tujuan penelitian 3](#_Toc84866163)

[1.5 Manfaat penelitian 3](#_Toc84866164)

[1.6 Sistematika penulisan 3](#_Toc84866165)

[1.7 Tinjauan pustaka 4](#_Toc84866166)

[BAB II LANDASAN TEORI 7](#_Toc84866167)

[2.1 Proposal Tugas Akhir 7](#_Toc84866168)

[2.2 Pengertian plagiarisme 8](#_Toc84866169)

[2.2.1 Tipe-tipe Plagiarisme 9](#_Toc84866170)

[2.2.2 Metode Pendeteksi Plagiarisme 9](#_Toc84866171)

[2.3 *Information Retrieval* (IR) 10](#_Toc84866172)

[2.4 *American Standard Code for Information Interchange* ( ASCII ) 11](#_Toc84866173)

[2.5 Algoritma *Rabin Karp* 12](#_Toc84866174)

[2.6 *HyperText Markup Language* (HTML) 15](#_Toc84866175)

[2.7 *Hypertext Preprocessor* (PHP) 15](#_Toc84866176)

[2.8 MySQL 16](#_Toc84866177)

[2.9 *Database Management System* (DBMS) 17](#_Toc84866178)

[2.10 *Unified Modeling Language* (UML) 17](#_Toc84866179)

[2.11 *Entity Relational Diagram* (ERD) 23](#_Toc84866180)

[2.12 Rancangan Pengujian *Black Box* 24](#_Toc84866181)

[BAB III METODOLOGI PENELITIAN 25](#_Toc84866182)

[3.1 Waktu dan Tempat Penelitian 25](#_Toc84866183)

[3.1.1 Waktu 25](#_Toc84866184)

[3.1.2 Tempat Penelitian 25](#_Toc84866185)

[3.2 Metode Pengumpulan Data 25](#_Toc84866186)

[3.3 Metode Pengembangan Sistem 26](#_Toc84866187)

[3.3.1 Permulaan *(Inception)* 26](#_Toc84866188)

[3.3.2 Perluasaan / Perencanaan *(Elaboration)* 26](#_Toc84866189)

[3.3.3 Konstruksi *(Construction)* 26](#_Toc84866190)

[3.3.4 Transisi *(Transition)* 26](#_Toc84866191)

[BAB IV ANALISIS DAN PERANCANGAN SISTEM 27](#_Toc84866192)

[4.1 Analisis Kebutuhan Sistem 27](#_Toc84866193)

[4.1.1 Kebutuhan Fungsional 27](#_Toc84866194)

[4.1.2 Kebutuhan Nonfungsional 27](#_Toc84866195)

[4.2 Analisis Perancangan Sistem 29](#_Toc84866196)

[4.2.1 *Entity Relationship* *Diagram* 29](#_Toc84866197)

[4.2.2 *Unified Modeling Language* (UML) 31](#_Toc84866198)

[4.2.3 Perancangan Antarmuka (*Interface*) 46](#_Toc84866199)

[BAB V IMPLEMENTASI DAN PEMBAHASAN 53](#_Toc84866200)

[5.1 Implementasi sistem 53](#_Toc84866201)

[5.2 Implementasi *Interface* 53](#_Toc84866202)

[5.3 Pengujian Sistem 57](#_Toc84866203)

[5.3.1 Perhitungan Manual Algoritma Rabin Karp 57](#_Toc84866204)

[5.3.2 Pengujian Algoritma *Rabin Karp* pada sistem 63](#_Toc84866205)

[5.3.3 Implementasi source code Algoritma Rabin karp 88](#_Toc84866206)

[5.4 Pengujian *black box* 89](#_Toc84866207)

[BAB VI PENUTUP 91](#_Toc84866208)

[6.1 Kesimpulan 91](#_Toc84866209)

[6.2 Saran 91](#_Toc84866211)

[DAFTAR PUSTAKA 92](#_Toc84866213)

# DAFTAR TABEL

[Tabel 2.1 ASCII 11](#_Toc83300294)

[Tabel 2.2 Komponen-komponen use *Case* diagram 18](#_Toc83300295)

[Tabel 2.3 Komponen-komponen *activity diagram* 20](#_Toc83300296)

[Tabel 2.4 Simbol-Simbol *Class Diagram* 20](#_Toc83300297)

[Tabel 2.5 Simbol-Simbol Sequence Diagram 22](#_Toc83300298)

[Tabel 2.6 Entity Relationship Standard 23](#_Toc83300299)

[Tabel 5.2 data uji 4 kata & data uji 7 kata 63](#_Toc84866410)

[Tabel 5.3 data asli 840 kata & data uji 708 kata. 64](#_Toc84866411)

[Tabel 5. 4 uji data 1 E1E117001 65](#_Toc84866412)

[Tabel 5. 5 uji data 2 E1E117011 66](#_Toc84866413)

[Tabel 5. 6 uji data 3 E1E117038 67](#_Toc84866414)

[Tabel 5. 7 uji data 4 E1E117053 69](#_Toc84866415)

[Tabel 5. 8 uji data 5 E1E117035 70](#_Toc84866416)

[Tabel 5. 9 Uji Data 6 E1E117049 72](#_Toc84866417)

[Tabel 5. 10 uji data 7 E1E117018 73](#_Toc84866418)

[Tabel 5. 11 uji data 8 E1E117016 75](#_Toc84866419)

[Tabel 5. 12 uji data 9 E1E117054 76](#_Toc84866420)

[Tabel 5. 13 uji data 10 E1E117065 78](#_Toc84866421)

[Tabel 5. 14 uji data 11 E1E117056 79](#_Toc84866422)

[Tabel 5. 15 uji data 12 E1E117039 81](#_Toc84866423)

[Tabel 5. 16 uji data 13 E1E117013 82](#_Toc84866424)

[Tabel 5. 17 uji data 14 E1E117064 84](#_Toc84866425)

[Tabel 5. 18 uji data 15 E1E117022 85](#_Toc84866426)

[Tabel 5. 19 uji data 15 E1E117021 87](#_Toc84866427)

[Tabel 5.20 Pengujian *Black box* 90](#_Toc84866428)

# DAFTAR GAMBAR

[Gambar 4. 1 *Entity* Diagram Hak Akses 29](#_Toc83299920)

[Gambar 4. 2 Entity Relationship Diagram Utama 30](#_Toc83299921)

[Gambar 4. 3 Physical Model dari ERD hak akses 30](#_Toc83299922)

[Gambar 4. 4 Physical Model dari ERD menu utama 31](#_Toc83299923)

[Gambar 4. 5 *Use Case Diagram* 32](#_Toc83299924)

[Gambar 4. 6 *Activity diagram login* 36](#_Toc83299925)

[Gambar 4. 7 *Activity* Diagram Mahasiswa 37](#_Toc83299926)

[Gambar 4. 8 *Activity* Diagram Dosen 38](#_Toc83299927)

[Gambar 4. 9 *Activity Diagram* Proposal 39](#_Toc83299928)

[Gambar 4. 10 *Activity Diagram* Plagiarisme 40](#_Toc83299929)

[Gambar 4. 11 *Class diagram* 41](#_Toc83299930)

[Gambar 4. 12 *Sequence Diagram Login* 42](#_Toc83299931)

[Gambar 4. 13 *Sequence Diagram* Mahasiswa 43](#_Toc83299932)

[Gambar 4. 14 *Sequence Diagram* Dosen 44](#_Toc83299933)

[Gambar 4. 15 *Sequence Diagram* Proposal 45](#_Toc83299934)

[Gambar 4. 16 *Sequence Diagram* Cek Plagiarisme 46](#_Toc83299935)

[Gambar 4. 17 Menu Login 47](#_Toc83299936)

[Gambar 4. 18 Menu *Dashboard* 48](#_Toc83299937)

[Gambar 4. 19 Mahasiswa 49](#_Toc83299938)

[Gambar 4. 20 Menu Dosen 50](#_Toc83299939)

[Gambar 4. 21 Proposal 51](#_Toc83299940)

[Gambar 4. 22 Cek Plagiarisme 52](#_Toc83299941)

[Gambar 5. 1 Tampilan *Login* (Admin, Mahasiswa dan dosen) 53](#_Toc83299910)

[Gambar 5. 2 Tampilan Halaman Beranda (Admin) 54](#_Toc83299911)

[Gambar 5. 3 Tampilan Halaman Mahasiswa (Admin) 54](#_Toc83299912)

[Gambar 5. 4 Tampilan Halaman Dosen (Admin) 55](#_Toc83299913)

[Gambar 5. 5 Tampilan Halaman Proposal (Admin dan mahasiswa) 55](#_Toc83299914)

[Gambar 5. 6 Tampilan Halaman Cek Plagiarisme (Admin dan Dosen) 56](#_Toc83299915)

[Gambar 5. 7 Tampilan Hasil Cek Plagiarisme (Admin dan Dosen)…..………...56](#_Toc83299915)

[Gambar 5. 8 Algoritma Rabin Karp untuk data asli 89](#_Toc83299916)

[Gambar 5. 9 Algoritma Rabin Karp untuk data uji 89](#_Toc83299917)

# BAB I PENDAHULUAN

## **Latar Belakang**

Informasi dapat beredar dengan cepat dan luas dari berbagai belahan dunia melalui Internet. Kemudahan akses internet yang sudah bisa dinikmati di mana saja merupakan salah satu faktor utama yang menyebabkan informasi dapat dengan mudah diperoleh. Namun, dikarenakan kemudahan informasi yang didapat tersebut, tidak sedikit tindak kejahatan yang terjadi di dunia maya contohnya adalah plagiarisme.

Plagiarisme sering di jumpai dalam sektor akademis maupun *non* akademis. Dalam sektor akademis, plagiarisme di anggap sebagai tindak pidana serius karena di anggap pengambilan karangan, pendapat, ide dan gagasan orang lain. Plagiarisme atau penjiplakan karya tulis orang lain dalam dunia akademik dipicu oleh banyak faktor. Salah satu faktor pemicunya adalah penulis (mahasiswa) ingin segera menyelesaikan skripsinya agar bisa meraih gelar akademik secepatnya tanpa harus bekerja keras sesuai proses riset dan penulisan ilmiah yang benar. Salah satu plagiarisme yang sering terjadi di kalangan mahasiswa adalah plagiarisme di dalam penulisan Proposal tugas akhir.

Proposal merupakan sebuah rancangan dan rencana kerja yang sistematis dan terperinci. Pada umumnya sebuah proposal bertujuan untuk menyampaikan rencana kegiatan pada pihak terkait, sehingga kegiatan tersebut dapat diterima dengan tujuan mendapatkan dukungan, mendapatkan izin, memperoleh dana dan sponsor, dan sebagainya. Sama halnya dengan proposal skripsi, yang mana mahasiswa bertujuan untuk menyampaikan rancangan penelitiannya kepada dosen pembimbing agar disetujui sebelum membuat skripsi sehingga sistem ini dapat di gunakan oleh dosen pembimbing untuk mendeteksi tingkat plagiarisme proposal mahasiswa yang melakukan bimbingan sebelum melakukan ujian seminar proposal.

Tindakan plagiarisme secara perlahan harus dicegah dan dihilangkan dengan melakukan pendeteksian plagiat secara manual maupun dengan memanfaatkan metode pencocokan *string*. Dengan demikian melakukan pendeteksian plagiarisme secara manual sangat tidak efektif sehingga Algoritma *Rabin Karp* dapat menjadi solusi untuk menyelesaikan hal tersebut.

Algoritma *Rabin Karp* merupakan algoritma pencarian *string* yang ditemukan oleh Michael Rabin dan Richard Karp. Algoritma *Rabin Karp* merupakan algoritma pencarian *string* yang menggunakan fungsi hashing untuk membandingkan *string* yang dicari (m) dengan *string* yang dibandingkan (n). Fungsi *hash* adalah fungsi matematis yang digunakan untuk mengubah data menjadi bilangan bulat yang relatif kecil yang dapat berfungsi sebagai indeks pada *array*. Fungsi *hashing* merupakan proses inti pada perhitungan algoritma *Rabin Karp*. *Hashing* merupakan representasi ASCII yang menggantikan atau mentransformasikan karakter atau tanda baca menjadi sebuah nilai atau angka (Suryati dkk., 2018).

Berdasarkan uraian diatas, peneliti akan melakukan implementasi Algoritma *Rabin Karp* ke dalam sistem berbasis web agar dapat mendeteksi seberapa besar persentase plagiarisme atau kemiripan proposal tugas akhir mahasiswa.

## **Rumusan masalah**

Berdasarkan latar belakang masalah yang diuraikan sebelumnya maka rumusan masalah penelitian ini adalah:

1. Bagaimana mendesain dan mengembangkan Sistem pendeteksi plagiarisme proposal tugas akhir?
2. Bagaimana menerapkan algoritma *Rabin Karp* yang akan digunakan dalam pengembangan sistem pendeteksian plagiarisme?

## **Batasan Masalah**

Adapun batasan masalah dalam sistem ini adalah:

1. Pembuatan sistem pendeteksi plagiarisme proposal skripsi menggunakan Algoritma *Rabin-Karp*.
2. Data yang di uji merupakan proposal tugas akhir yang menggunakan bahasa indonesia.
3. Dokumen yang di uji pada proposal tugas akhir yaitu pada bagian latar belakang proposal.
4. Mendeteksi file teks yang bersifat plain teks yaitu huruf, tidak berupa gambar dan diagram.

## **Tujuan penelitian**

Adapun tujuan dari penelitian ini adalah:

1. Mendesain dan membangun sistem pendeteksi plagiarisme dalam proposal tugas akhir mahasiswa dengan Algoritma *Rabin Karp*.
2. Mengurangi tingkat plagiarisme pada Proposal Tugas akhir mahasiswa.
3. Mendeteksi tingkat plagiarisme pada Proposal Tugas akhir mahasiswa.

dengan *Algoritma* *Rabin Karp*.

## **Manfaat penelitian**

Adapun manfaat dari penelitian ini adalah:

1. Memudahkan dosen pembimbing untuk mendeteksi plagiarisme proposal mahasiswa yang melakukan bimbingan.
2. Dapat membantu sebagai bahan pertimbangan dalam menentukan plagiarisme.
3. Dapat menentukan *persentase* kemiripan (*similarity*) antara dokumen yang diuji dengan dokumen lainnya oleh sistem.

## **Sistematika penulisan**

Sistematika penulisan tugas akhir ini terdiri atas beberapa bagian utama sebagai berikut:

**BAB I Pendahuluan**

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

**BAB II Landasan teori**

Pada bab ini membahas teori-teori pendukung yang berkaitan dengan skripsi yang akan di buat. Teori yang akan di angkat yaitu mengenai deteksi plagiarisme dengan Algoritma *Rabin Karp*.

**BAB III Metodologi penelitian**

Pada bab ini membahas tentang tahapan penelitian, tahapan pengumpulan data, perancangan perangkat lunak, implementasi pengujian sistem dan analisa akhir.

**BAB IV Analisis Dan Perancangan Sistem**

Pada bagian ini diuraikan analisis sistem yang akan dibuat dan kebutuhan sistem yang meliputi kebutuhan fungsional, kebutuhan non fungsional sistem. Rancangan sistem meliputi rancangan arsitektur sistem, rancangan proses, rancangan prosedural, rancangan data, dan rancangan *user interface*.

**BAB V Implementasi Sistem**

Bagian ini menyajikan secara lengkap setiap langkah yang dilakukan dalam implementasi sistem, analisis hasil dan pembahasan hasil penelitian.

**BAB VI Penutup**

Bagian ini berisi tentang kesimpulan dan saran-saran mengenai tugas akhir pendeteksi plagiarisme pada proposal tugas akhir mahasiswa dengan Algoritma *Rabin Karp*.

## **Tinjauan pustaka**

Menurut penelitian (Filcha dan Hayaty, 2019) yang berjudul “Implementasi Algoritma *Rabin-Karp* untuk pendeteksi pada dokumen tugas Mahasiswa” menjelaskan Berdasarkan hasil dan pembahasan maka peneliti dapat mengambil kesimpulan diantaranya Algoritma *Rabin-Karp* berhasil diimplementasikan pada sistem pendeteksi plagarisme dokumen tugas mahasiswa. Sistem ini berhasil menampilkan persentase kemiripan dokumen tugas antar mahasiswa. Hasil perhitungan akurasi dengan *confusion matrix* pada sistem pendeteksi plagiarisme dokumen tugas mahasiswa adalah 90% yang diperoleh dari 20 perbandingan dokumen tugas mahasiswa. Algoritma yang digunakan pada sistem pendeteksi plagiarisme dokumen tugas mahasiswa tidak memiliki perbedaan persentase saat urutan perbandingan diubah.

Menurut penelitian (Putra dan Sularno, 2019) yang berjudul “Penerapan Algoritma *Rabin Karp* Dengan Pendekatan *Synonym Recognition* Sebagai Antisipasi Plagiarisme Pada Penulisan Skripsi” menjelaskan, Berdasarkan hasil analisa dan penerapan yang sudah dilakukan pada bab sebelumnya, maka dapat disimpulkan bahwa Telah dibuat suatu sistem berbasiskan web yang dapat digunakan untuk pendeteksian plagiarisme terhadap suatu dokumen teks dengan menggunakan Algoritma *Rabin Karp.* Tindakan plagiarisme dapat dilakukan dengan mengubah beberapa bagian bahkan mungkin secara keseluruhan dengan cara mengubah kata-kata dengan sinonim daripada kata-kata tersebut. Algoritma *Rabin Karp* yang menggunakan metode *Synonym Recognition* memilki akurasi penentuan nilai persentase *Similarity* yang lebih akurat bila dibandingkan dengan algoritma *Rabin Karp* tanpa *Synonym Recognition*. Karena walaupun kata pada dokumen uji diganti sedemikian rupa namun memiliki makna yang sama akan terdeteksi oleh sistem ini. Hasil dari *Synonym Recognition* sangat bergantung pada banyaknya sinonim yang diinputkan ke dalam kamus sinonim pada *database*. Nilai *K-gram* yang lebih kecil akan menghasilkan akurasi nilai similarity yang lebih baik dibandingkan dengan nilai *K-gram* yang lebih besar. Untuk nilai basis, tidak semua angka bisa digunakan. Karena di beberapa kasus nilai basis yang salah akan mengakibatkan nilai hash yang dihasilkan akan sama dengan nilai hash lain yang memilki kata berbeda. Dokumen uji dinyatakan terduplikasi apabila nilai persentase *Similarity* nya diatas 50%.

Menurut penelitian (Uji Cahyono, 2018) yang berjudul “Aplikasi Deteksi Dini Plagiarisme Judul Tugas Akhir Mahasiswa Sekolah Tinggi Ilmu Kesehatan Yayasan Rs. Islam Surabaya Dengan Algoritma *Rabin-Karp* Draja” menjelaskan, beberapa kesimpulan yang dapat diambil pada penelitian ini antara lain : Aplikasi deteksi dini plagiarisme tugas akhir sudah cukup mudah untuk digunakan. Berdasarkan hasil evaluasi uji coba sistem dapat diketahui bahwa performa hasil persentase similaritas dari Algoritma *Rabin-Karp* memiliki ketergantungan dengan nilai *K-gram* yang diberikan. Berdasarkan hasil pengujian akurasi pada pendeteksian similaritas judul tugas akhir diperoleh hasil rata-rata akurasi similaritas adalah sebesar 63,19 %. Algoritma Rabin-Karp cukup layak untuk digunakan sebagai suatu metode sederhana untuk mendeteksi secara dini kemungkinan terjadinya plagiarisme. Akan tetapi, aplikasi ini belum dapat digunakan secara sempurna untuk memastikan kemungkinan akan terjadinya tindakan plagiarisme tugas akhir.

Menurut penelitian (Adi, 2018) yang berjudul “Penerapan Algoritma *Rabin Karp* Untuk Mendeteksi Kemiripan Judul Skripsi” menjelaskan, Beberapa kesimpulan yang dapat diambil pada penelitian ini antara lain : Sistem dalam membandingkan file memberikan hasil berupa presentase *similarity.* *Preprocessing* sangat membantu sekali dalam proses *Rabin Karp*. Karena dengan melakukan *preprocessing* maka dapat mengurangi volume dataset tanpa mengurangi esensi nilai dataset tersebut, sehingga proses *parsing* jumlah k-nya menjadi lebih sedikit. Jika file tidak mengalami proses *preprocessing*, maka waktu yang diperlukan semakin kecil namun nilai similaritas-nya berkurang.

Menurut penelitian (Pratomo dan Suryotomo, 2020) yang berjudul “Sistem Pencegahan Plagiarism Tugas Akhir Menggunakan Algoritma *Rabin-Karp*” menjelaskan beberapa kesimpulan yang dapat di ambil pada penelitian ini antara lain : Sistem ini dibangun untuk digunakan oleh mahasiswa, dosen dan Koordinator skripsi untuk mempermudah dalam mengajukan, memproses serta memberi penilaian terhadap proposal skripsi yang diajukan. Dari hasil penelitian diketahui pemilihan n-gram yang semakin kecil akan memperoleh nilai persentase kemiripan yang besar (76% dan 89%). Hal ini terjadi karena pada n-gram yang lebih sedikit, string yang dipotong lebih kecil sehingga kemungkinan untuk ditemukannya rangkaian karakter yang sama semakin besar. Semakin besarnya n-gram, maka mengandung karakter yang lebih banyak dibandingkan dengan n-gram yang lebih kecil sehingga menyebabkan rangkaian karakter yang ditemukan semakin berkurang sehingga menurunkan nilai similaritas (hingga 57% dan 68%).

# BAB II LANDASAN TEORI

## **Proposal Tugas Akhir**

Proposal adalah rencana yang dituangkan dalam bentuk rancangan kerja, perencanaan sistematis, matang, dan teliti yang dibuat oleh peneliti sebelum melaksanakan penelitian. dari sudut pandang dunia ilmiah, pengertian proposal adalah rancangan dari suatu usulan sebuah penelitian yang kemudian dilaksanakan oleh peneliti terhadap bahan penelitiannya. Sehingga proposal sama halnya dengan sebuah usulan kegiatan yang memerlukan persetujuan dari pihak lain. Tujuan proposal adalah untuk mendapatkan persetujuan dari suatu pihak mengenai rencana atau rancangan yang akan dilakukan. Di dalam proposal berisi kelayakan dari proposal untuk di lanjutkan menjadi tugas akhir. Adapun isi dari proposal tugas akhir di sesuaikan dengan jenis tugas akhir, yaitu tugas akhir penelitian dan tugas akhir rekayasa sistem. Salah satu isi dari proposal tugas akhir penelitian adalah latar belakang. Pada latar belakang berisi alasan dipilihnya judul penelitian, yang biasanya berasal dari fenomena atau kejadian empirris yang akan diangkat sebagai topik penelitian. Kemukaan uraian tentang masalah yang menarik minat untuk diteliti. Selain itu memuat uraian atau penjelasan mengenai alasan-alasan sehingga permasalahan yang di ajukan merupakan permasalahan yang layak untuk diteliti dan di temukan cara untuk menyelesaikannya. Penelitian yang di lakukan hendaknya asli, dalam arti bahwa masalah yang dipilih belum pernah di teliti oleh peneliti sebelumnya atau pengambilan karya orang lain (plagiarisme) sehingga harus di nyatakan dengan tegas perbedaan penelitian yang di lakukan dengan penelitian yang sudah pernah di lakukan sebelumnya.

Salah satu contoh jenis proposal yaitu Proposal Penelitian,jenis proposal ini lebih sering digunakan di bidang akademisi misalnya penelitian untuk pembuatan skripsi, tesis dan lainnya. Isi dari proposal ini adalah pengajuan kegiatan penelitan. Kemampuan menyusun proposal penelitian sangat penting untuk merencanakan dan mengusulkan suatu kegiatan atau proyek penelitian. Setelah penelitian dilakukan, disusun laporan untuk disampaikan kepada pihak-pihak yang

berkepentingan. Oleh karena itu selain kemampuan menyusun proposal juga perlu dimiliki kemampuan menulis laporan penelitian (Hasyim dan Nugrahanto, 2014).

## **Pengertian plagiarisme**

Plagiarisme atau sering disebut plagiat adalah penjiplakan atau pengambilan karangan, pendapat, dan sebagainya dari orang lain dan menjadikannya seolah karangan dan pendapat sendiri. Plagiat dapat dianggap sebagai tindak pidana karena mencuri hak cipta orang lain tanpa meminta izin dan menyertakan sumber yang dicatutnya. Di dunia pendidikan, pelaku plagiarisme dapat mendapat hukuman berat seperti dikeluarkan dari sekolah / universitas, denda berupa uang dan bahkan hukuman penjara. Pelaku plagiat disebut sebagai plagiator. Sejak abad ke-19, plagiat atau plagiarisme telah menjadi masalah serius dalam dunia pendidikan yang masih berlangsung sampai saat ini. Ini tentu memerlukan pertimbangan khusus karena memiliki dampak yang tidak sehat dalam dunia pendidikan. (Aziz dkk., 2015).

Berikut ini adalah hal-hal yang tergolong kedalam tindakan plagiarisme, antara lain:

1. Mengakui tulisan orang lain sebagai tulisan sendiri.
2. Mengakui gagasan orang lain sebagai pemikiran sendiri.
3. Mengakui temuan orang lain sebagai kepunyaan sendiri.
4. Mengakui karya kelompok sebagai kepunyaan atau hasil sendiri.
5. Menyajikan tulisan yang sama dalam kesempatan yang berbeda tanpa menyebutkan asal-usulnya.
6. Meringkas dan memparafrasekan (mengutip tak langsung) tanpa menyebutkan sumbernya.
7. Meringkas dan memparafrasekan dengan menyebut sumbernya, tetapi rangkaian kalimat dan pilihan katanya masih terlalu sama dengan sumbernya.

### **2.2.1 Tipe-tipe Plagiarisme**

Plagiarisme tidak selalu disengaja atau mencuri sesuatu dari orang lain. Praktik ini dapat bersifat tidak disengaja, kebetulan, dan dapat mencakup pencurian sendiri (*self stealing*). Berikut ini beberapa tipe plagiarisme.

1. Kebetulan (*accidental*)

Praktik plagiarisme ini dapat terjadi karena kurangnya pengetahuan akan plagiarisme dan pemahaman mengenai penulisan referensi.

1. Tidak disengaja (*unintentional*)

Ketersediaan informasi dalam jumlah yang sangat besar mempengaruhi pemikiran sehingga ide yang sama dapat dihasilkan secara tertulis maupun lisan sebagai milik pribadi.

1. Disengaja (*intentional*)

Tindakan menyalin sebagian atau keseluruhan hasil karya orang lain secara sengaja tanpa mengikutsertakan nama pemilik hasil karya.

1. Diri sendiri (*self* *plagiarism*)

Penggunaan hasil karya yang dibuat diri sendiri dalam bentuk lain tanpa menunjuk hasil karya asli.

### **Metode Pendeteksi Plagiarisme**

Metode Pendeteksi Plagiarisme dibagi menjadi tiga bagian yaitu metode perbandingan teks lengkap, metode dokumen *fingerprinting*, dan metode kesamaan kata kunci. Metode pendeteksi plagiarisme dapat digambarkan sebagai berikut :

1. Perbandingan Teks Lengkap

Metode ini diterapkan dengan membandingkan semua isi dokumen.Dapat diterapkan untuk dokumen yang besar. Pendekatan ini membutuhkan waktu yang lama tetapi cukup efektif, karena kumpulan dokumen yang diperbandingkan adalah dokumen yang disimpan pada penyimpanan lokal. Metode perbandingan teks lengkap tidak dapat diterapkan untuk kumpulan dokumen yang tidak terdapat pada dokumen lokal. Algoritma yang digunakan pada metode ini adalah algoritma *Brute-Force*, Algoritma *Edit Distance*, Algoritma *Boyer Moore* dan Algoritma *Lavenshtein Distance*.

1. Dokumen *Fingerprinting*

Dokumen *fingerprinting* merupakan metode yang digunakan untuk mendeteksi keakuratan salinan antar dokumen, baik semua teks yang terdapat di dalam dokumen atau hanya sebagian teks saja. Prinsip kerja dari metode dokumen *fingerprinting* ini adalah dengan menggunakan teknik *hashing*. Teknik *hashing* adalah sebuah fungsi yang mengkonversi setiap *string* menjadi bilangan. Misalnya *Rabin Karp*, *Winnowing dan Manber*.

1. Kesamaan Kata Kunci

Prinsip dari metode ini adalah mengekstrak kata kunci dari dokumen dan kemudian dibandingkan dengan kata kunci pada dokumen yang lain. Pendekatan yang digunakan pada metode ini adalah teknik dot.

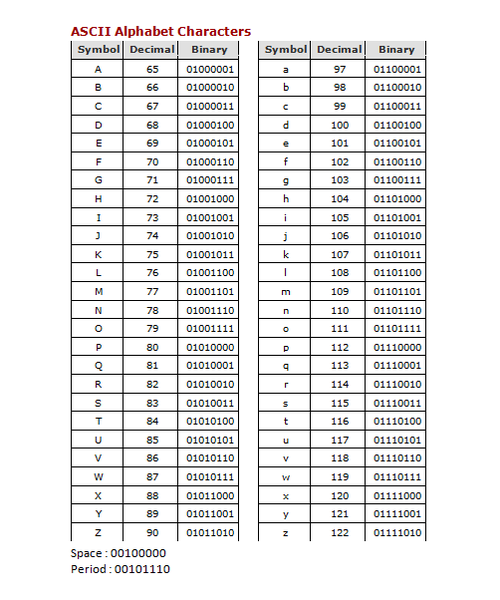
* 1. ***Information Retrieval* (IR)**

*Information retrieval* adalah studi tentang sistem pengindeksan, pencarian, dan mengingat data, khususnya teks atau bentuk tidak teratur lainnya. Pada proses tahap indexing didalam *Information retrieval* terdapat proses *stemming* yaitu proses mengubah suatu kata bentukan menjadi kata dasar. Proses *stemming* sangat tergantung kepada bahasa dari kata yang akan di stem. Hal ini dikarnakan, dalam melakukan proses stemming harus mengaplikasikan aturan morfologikal dari suatu bahasa. Bahasa Indonesia memiliki kata berimbuhan yang lebih kompleks dibandingkan dengan Bahasa Lainnya. Terdapat dua pendekatan dalam proses stemming yaitu: *Light Stemming* dan *Dictionary Base Stemming*. pendekatan dengan proses *Dictionary Base Stemming* dapat memberikan solusi untuk men-stemm kata berimbuhan dalam Bahasa Indonesia, karena menggunakan struktur morfologi untuk mengekstrak kata berimbuhan menjadi kata dasar (*root word*). Stemming adalah metote yang digunakan untuk menghilangkan imbuhan dari sebuah kata. *Stemming* merupakan salah satu tahapan dalam *pre processing*. Memiliki pengaruh terhadap tingkat relevansi temu kembali informasi seperti dalam pencarian informasi berdasarkan *query*. Untuk mendapatkan kata dasar beberapa imbuhan yang dihilangkan adalah awalan, akhiran dan sisipan (Magriyanti, 2018).

* 1. ***American Standard Code for Information Interchange* ( ASCII )**

Kode Standar Amerika untuk Pertukaran Informasi atau ASCII (*American Standard Code for Information Interchange*) merupakan suatu standar internasional dalam kode huruf dan simbol seperti *Hex* dan *Unicode* tetapi ASCII lebih bersifat universal, contohnya 124 adalah untuk karakter "|". Itu selalu digunakan oleh komputer dan alat komunikasi lain untuk menunjukkan teks. Kode ASCII sebenarnya memiliki komposisi bilangan biner sebanyak 7 bit. Namun, ASCII disimpan sebagai sandi 8 bit dengan menambakan satu angka 0 sebagai bit significant paling tinggi. Bit tambahan ini sering digunakan untuk uji prioritas. Karakter control pada ASCII dibedakan menjadi 5 kelompok sesuai dengan penggunaan yaitu berturut-turut meliputi *logical communication, device control, information separator, code extention, dan physical communication*. Jumlah kode ASCII adalah 255 kode. (Tantoni and Zaen, 2018).

Tabel 2.1 ASCII



## **Algoritma *Rabin Karp***

Algoritma *Rabin Karp* adalah algoritma pencarian kata yang mencari sebuah pola berupa *substring* dalam sebuah teks menggunakan *hashing*. Algoritma ini sangat efektif untuk pencocokan kata dengan pola banyak. Salah satu aplikasi praktis dari Algoritma *Rabin Karp* adalah dalam pendeteksian plagiarisme. Dalam ilmu komputer, Algoritma *Rabin-Karp* adalah algoritma pencarian *string* yang dibuat oleh Michael O. Rabin dan Richard M. Karp pada tahun 1987 yang menggunakan *hashing* untuk menemukan salah satu dari satu set *string* pola dalam teks.

Algoritma *Rabin Karp* merupakan algoritma pencarian *string* yang menggunakan fungsi hashing untuk membandingkan *string* yang dicari (m) dengan *string* yang dibandingkan (n). Fungsi hash adalah fungsi matematis yang digunakan untuk mengubah data menjadi bilangan bulat yang relatif kecil yang dapat berfungsi sebagai indeks pada array. Fungsi *hashing* merupakan proses inti pada perhitungan algoritma *Rabin Karp*. *Hashing* merupakan representasi ASCII yang menggantikan atau mentransformasikan karakter atau tanda baca menjadi sebuah nilai atau angka (Suryati, Wibisono and Wihardi, 2018).

Adapun tahapan proses sistem deteksi kesamaan dokumen dengan metode *rabin karp* adalah sebagai berikut:

1. **Tahap *Input* Parameter**

Proses sistem yang akan dibuat dimulai dari memasukkan dokumen yaitu dokumen asli dan dokumen uji. Kemudian akan dilakukan pengujian melalui tahapan-tahapan *preprocessing*, yaitu menghilangkan tanda baca, merubah huruf ke huruf kecil, menghilangkan kata-kata yang kurang penting, kemudian pembentukan rangkaian gram panjang k, lalu menghitung nilai hashing. Dari pengujian tersebut akan didapatkan hasil akhirnya berupa persentase kemiripan (*similarity*). Dalam menentukan standar persentase kemiripan terdapat 3 klasifikasi yang digunakan dalam sistem pendeteksi plagiarism tugas akhir , yaitu:

* + - 1. Plagiarisme ringan, jika tingkat kesamaannya kurang dari 30%.
      2. Plagiarisme sedang jika tingkat kesamaannya lebih dari 30% hingga 70%.
      3. Plagiarism berat, jika tingkat kesamaannya lebih dari 70%.

1. **Tahapan *Preprocessing***

Tahap *preprocessing* harus dilalui untuk menentukan *keyword* pada kedua dokumen yang akan dilakukan pengujian, yaitu dokumen asli dan dokumen uji. Pada tahap *preprocessing* terdapat beberapa proses yang dilakukan oleh sistem terhadap dokumen yang diinputkan. Proses-proses tersebut antara lain *Case folding, filtering, dan tokenizing.*

1. Sub Proses *Case Folding*

Proses *Case folding* (mengubah huruf kapital menjadi huruf kecil) merupakan tahap pertama yang akan dilakukan dari rangkaian tahapan yang terdapat pada *preprocessing.*

1. Sub Proses *Filtering*

Setelah proses *Case folding*, dokumen selanjutnya masuk ketahap *filtering*, yaitu proses membuang kata yang tidak penting (stop word) misalnya kata “yang”, “adalah” dan “dari”. dan membuang tanda baca (“ “), ( , ), ( . ).

1. *Stemming*

*Stemming* adalah proses untuk merubah kata menjadi kata dasar. Selain itu, *Stemming* adalah proses ekstraksi kata dari imbuhannya untuk mendapatkan kata dasar. Hasil dari proses *stemming* disebut dengan stem.

1. Sub Proses *Tokenizing*

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

1. Proses *Parsing* *K-gram*

Langkah selanjutnya adalah *parsing* *k-gram*, *k-gram* adalah rangkaian *terms* dengan panjang K. Kebanyakan yang digunakan sebagai *terms* adalah kata, dimana pada proses ini kata dipecah menjadi potongan-potongan dimana setiap potongan mengandung karakter sebanyak k. Nilai *k-gram* sangat berpengaruh terhadap nilai persentase suatu kemiripan dokumen. pemilihan *k-gram* yang semakin kecil akan memperoleh nilai persentase kemiripan yang besar. Hal ini terjadi karena pada *k-gram* yang lebih sedikit, string yang dipotong lebih kecil sehingga kemungkinan untuk ditemukannya rangkaian karakter yang sama semakin besar. Semakin besarnya *k-gram*, maka mengandung karakter yang lebih banyak dibandingkan dengan *k-gram* yang lebih kecil sehingga menyebabkan rangkaian karakter yang ditemukan semakin berkurang sehingga menurunkan nilai similaritas (Pratomo and Suryotomo, 2020). Nilai *k-gram* di dalam sistem ini nilai *k-gram* nya telah di tetapkan yaitu nilai k= 4.

1. Proses *Hashing*

Setelah melakukan proses *preprocessing* dan *tokenizing* *k-gram* maka langkah selanjutnya adalah melakukan proses *hashing* terhadap pecahan string yang telah dibagi menjadi *k-gram*. Fungsi ini akan mengganti atau men-transpose-kan data untuk menciptakan *fingerprint* yang biasa disebut *hash value*. Fungsi *hash* yang baik adalah yang menghasilkan sedikit *hash collision*. Solusi yang digunakan untuk mengatasi *hash collision* adalah menggunakan *Rolling Hash*. Dalam prosesnya digunakan basis yang biasanya adalah bilangan prima yang cukup besar, dengan tujuan agar meminimalkan terjadinya tabrakan. Alasan kenapa Basis bilangan (b) harus dipilih bilangan prima yang cukup besar, adalah untuk mengurangi kemungkinan adanya dua *hash value* yang sama. Proses *hashing* berfungsi untuk mengkonvert setiap *string* menjadi bilangan. Dengan cara mengalikan nilai ASCII hasil huruf hasil *k-gram* dengan basis bilangan tertentu.

Pada sistem ini proses *hashing* memanfaatkan tabel ASCII dengan rumus *Hash* sebagai berikut :

H(C1 …. CK) =C1.b(K-1)+c2.b(K-2)+ ….. +C(k-1).b(k)+ck

Keterangan :

H= Nilai *hash*

c = Nilai asciikarakter (desimal)

k = Banyak karakter (indeks karakter)

b = Basis Bilangan (nilai dari basis bilangan harus bilangan prima).

1. Deteksi Kesamaan

Setelah diketahui nilai hashnya, jumlah hash pada dokumen asli 24, jumlah hash pada dokumen uji 26 dan hash yang sanma pada kedua dokumen 14. Proses selanjutnya adalah menghitung *similarity* yaitu tingkat kesamaan dua dokumen dengan menggunakan persamaan, yaitu barapa persen tingkat kasamaannya. *Similarity* didapat dari hasil *hash* yang sama di bagi dengan jumlah *hash* kedua dokumen dikali dengan seratus persen.

***Similarity* (asli, uji) = x 100 %**

## ***HyperText Markup Language* (HTML)**

HTML yang merupakan singkatan dari *HyperText Markup Language* adalah serangkaian kode program yang merupakan dasar dari representasi visual sebuah halaman Web. Didalamnya berisi kumpulan informasi yang disimpan dalam tag-tag tertentu, dimana tag-tag tersebut digunakan untuk melakukan format terhadap informasi yang dimaksud. Berbagai pengembangan telah dilakukan terhadap kode HTML dan telah melahirkan teknologi-teknologi baru di dalam dunia pemrograman web. Kendati demikian, sampai sekarang HTML tetap berdiri kokoh sebagai dasar dari bahasa web seperti PHP, ASP, JSP dan lainnya. Bahkan secara umum, mayoritas situs web yang ada di Internet pun masih tetap menggunakan HTML sebagai teknologi utama (Pahlevi, Mulyani and Khoir, 2018).

## ***Hypertext Preprocessor* (PHP)**

PHP Kepanjangan dari PHP adalah "*Hypertext Preprocessor*" (ini merupakan singkatan rekursif). PHP adalah bahasa *scriptingweb* HTML-*embedded*. Ini berarti kode PHP dapat disisipkan ke dalam HTML halaman Web. Ketika sebuah halaman PHP diakses, kode PHP dibaca atau"diurai" oleh server. Output dari fungsi PHP pada halaman biasanya dikembalikan sebagai kode HTML, yang dapat dibaca oleh *browser*. Karena kode PHP diubah menjadi HTML sebelum halaman dibuka, pengguna tidak dapat melihat kode PHP pada halaman. Ini membuat halaman PHP cukup aman untuk mengakses database dan informasi aman lainnya. Banyak sintaks PHP yang hasil adaptasi dari bahasa lain seperti bahasa C, Java dan Perl. Namun, PHP memiliki sejumlah fitur unik dan fungsi tertentu juga. Tujuan dari bahasa pemrograman PHP adalah untuk memungkinkan pengemban web untuk menulis halaman yang dihasilkan secara dinamis dengan cepat dan mudah. PHP juga bagus untuk menciptakan situs Web *database-driven.* Jika Anda ingin mempelajari lebih lanjut tentang PHP, situs resminya yaitu PHP.net. (Ferdianto, 2013).

Beberapa kelebihan PHP dari bahasa pemrograman web, antara lain:

1. Bahasa pemrograman PHP adalah sebuah bahasa script yang tidak melakukan sebuah kompilasi dalam penggunaanya.
2. Web Server yang mendukung PHP dapat ditemukan dimana - mana dari mulai *apache,* IIS*, Lighttpd*, hingga *Xitami* dengan konfigurasi yang relatif mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis - milis dan developer yang siap membantu dalam pengembangan.
4. Dalam sisi pemahamanan, PHP adalah bahasa *scripting* yang paling mudah karena memiliki referensi yang banyak.
5. PHP adalah bahasa *open source* yang dapat digunakan di berbagai mesin (*Linux, Unix, Macintosh, Windows*) dan dapat dijalankan secara runtime melalui console serta juga dapat menjalankan perintah-perintah system.

## **MySQL**

*SQL* *(Structured Query Language)* merupakan sebuah bahasa relational yang berisi pernyataan yang digunakan untuk memasukkan, mengubah, menghapus, memilih dan melindungi data (Prihatna, 2005). SQL bukan *database* aplikasi, tetapi lebih berarti dengan suatu bahasa yang digunakan untuk mengajukan pertanyaan ke dalam *database* berupa pengguna SQL. *Database* sistem yang memiliki konsep sama dengan *SQL*, adalah Postgres dan MySQL, dimana *database* tersebut bisa didapatkan gratis atau dengan harga yang murah. MySQL adalah *server multithreaded*, sehingga sangat memungkinkan daemon untuk menghandle permintaan layanan secara stimultan. Model koneksi dengan protocol TCP-IP membuat akses ke server *database* lebih cepat jika dibandingkan dengan menggunakan *mapping drive* (Pahlevi dkk., 2018).

* 1. ***Database Management System* (DBMS)**

*Database Management**System* (DBMS) merupakan perangkat lunak untuk mengendalikan pembuatan, pemeliharaan, pengolahan, dan penggunaan data yang berskala besar. Penggunaan DBMS saat ini merupakan hal yang sangat penting dalam segala aspek, baik itu dalam skala yang besar atau kecil. Sebagai contoh media sosial facebook menggunakan DBMS untuk menyimpan data-data pengguna facebook yang sangat banyak kedalam DBMS MySQL. Beberapa DBMS yang digunakan adalah MySQL dan MariaDB. Berdasarkan survey yang dilakukan, MySQL dan MariaDB merupakan DBMS yang banyak digunakan sebagai contoh survey yang terdapat pada *db-engines.com,* *db-Engines Ranking* menempatkan MySQL pada posisi ke-2 sedangkan MariaDB pada posisi ke-20 namun pada *survey* yang terdapat di *serverwatch*.com *Top 10 Enterprise Database Sistem Of 2016*, MariaDB menempati posisi ke-6 dan MySQL menempati posisi ke-7 (Wulan Nafesa Septine, S.T., 2019).

1. *Table*

*Table* atau Tabel adalah sekumpulan data dengan struktur yang sedemikian rupa, terbentuk dari *record* dan *field.* Istilah Tabel disini berbeda dengan istilah Tabel pada HTML, walaupun secara visual hampir sama.

1. *Record*

*Record* adalah sekumpulan *field* yang membentuk suatu objek tertentu.

1. *Field*

*Field* adalah atribut dari objek yang memiliki tipe data tertentu.

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

[*Unified Modeling Language*](http://fadhlyashary.blogspot.com/2012/06/pengertian-uml-unified-modeling.html) (UML) adalah bahasa permodelan untuk menentukan visualisasi, kontruksi, dan merancang sistem berbasis *object-oriented.* [*Unified modeling language*](http://fadhlyashary.blogspot.com/2012/06/pengertian-uml-unified-modeling.html) juga digunakan untuk pengembangan sebuah sistem yang dapat menyampaikan alur kerja sistem dan menjelaskan tugas setiap *user* dalam sebuah sistem (Aziz, Lulu A., Ana I., 2015). Ada beberapa jenis diagram dalam UML (Iii, Jl dan Soepomo, 2010) , yaitu:

1. ***Use Case Diagram***

*Use Case* adalah kegiatan atau urutan interaksi yang saling berkaitan antara sistem dan actor, *Use Case* bekerja dengan cara mendeskripsikan tipe interaksi antara *user* sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Adapun komponen-komponen *use Case* diagram yaitu pada tabel 2.2 sebagai berikut:

Tabel 2.2 Komponen-komponen *Use Case Diagram*

|  |  |  |
| --- | --- | --- |
| Simbol | Nama | Keterangan |
|  | *Actor* | Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan *use Case.* |
|  | *Dependency* | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*dependent*) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*). |
|  | *Generalization* | Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada  diatasnya objek induk (*ancestor*). |
|  | *Include* | Menspesifikasikan bahwa *use Case* sumber secara *eksplisit*. |
|  | *Extend* | Menspesifikasikan bahwa *use Case* target memperluas perilaku dari *use Case* sumber pada suatu titik yang  diberikan. |
|  | *Association* | Apa yang menghubungkan antara objek  satu dengan objek lainnya. |
|  | *System* | Menspesifikasikan paket yang menampilkan sistem secara terbatas. |
|  | *Use Case* | Deskripsi dari uraian aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu *actor.* |
|  | *Collaboration* | Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah  dan elemen-elemennya (sinergi). |
|  | *Note* | Elemen fisik yang eksis saat aplikasi  dijalankan dan mencerminkan suatu  sumber daya komputasi. |

Sumber: (Kiswanto, 2016).

1. ***Activity Diagram***

*Activity diagram* ini menggambarkan tentang aktifitas yang terjadi pada sistem. Dari mulai sampai selesai, diagram ini menunjukkan langkah-langkah dalam proses kerja sistem yang dibuat (Shasi Aprilia Windiyani, 2020).

Adapun komponen-komponen yang terdapat pada diagram ini yaitu pada tabel 2.3 yaitu sebagai berikut:

Tabel 2.3 Komponen-komponen *Activity Diagram*

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Keterrangan** |
|  | *Activity* | Memperlihatkan bagaimana masing- masing kelas antarmuka saling berinteraksi satu sama lain. |
|  | *Action* | *State* dari sistem yang mencerminkan  eksekusi dari suatu aksi. |
|  | *Initial Node* | Bagaimana objek dibentuk atau diawali  (*start flow*). |
|  | *Activity Final*  *Node* | Bagaimana objek dibentuk dan diakhiri  (*end flow*). |
|  | *Fork Node* | Satu aliran yang pada tahap tertentu  berubah menjadi beberapa alur. |

Sumber: (Kiswanto, 2016).

1. ***Class Diagram***

*Class diagram* atau diagram kelas menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Diagram kelas dibuat agar *programmer* dapat membuat kelas-kelas didalam *program* perangkat lunak sesuai dengan perancangan diagram kelas (Kiswanto, 2016). Ada beberapa simbol-simbol yang terdapat dalam diagram kelas yaitu pada tabel 2.4, yaitu:

Tabel 2.4 Simbol-Simbol *Class Diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Simbol** | **Nama** | **Keterangan** |
| 1 | |  | | --- | | + atribut | | + operasi() | | Kelas | Kelas pada struktur sistem |
| 2. |  | *Interface* | Semua dengan konsep *interface* dalam  pemrograman b*error* ientasi objek |
| 3. |  | *Association* | Relasi antarkelas dengan makna umum, asoasi biasanya juga disertai dengan *multiplicity* |
| 4. |  | *directed asosiasi* | Relasi antarkelas dengan makna kelas yang satu  digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan *multiplicity* |
| 5. |  | Generalisasi | Relasi antarmuka dengan makna generalisasi spesialisasi (umum khusus) |
| 6. |  | *Dependency* | Relasi antarmuka dengan makna kebergantungan antarkelas |
| 7. |  | *Aggregation* | Relasi antarkelas dengan makna semua bagian (*whole- part*) |

Sumber: (Kiswanto, 2016).

1. ***Sequence Diagram***

*Sequence diagram* menggambarkan kelakuan objek pada *use Case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan terima antar objek. Oleh karena itu untuk menggambar diagram *sequence* maka harus diketahui objek-objek yang dimiliki kelas yang diinstasiasi menjadi objek itu. Menggambar diagram  *sequence* juga dibutuhkan untuk melihat skenario yang ada pada *use Case* (Kiswanto, 2016).

Beberapa simbol-simbol yang terdapat dalam diagram kelas yaitu pada tabel 2.5 sebagai berikut :

Tabel 2.5 Simbol-Simbol *Sequence Diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Simbol** | **Nama** | **Keterangan** |
| 1. |  | Aktor | Orang proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri. |
| 2. |  | Garis hidup / *lifeline* | Menyatakan kehidupan suatu objek |
| 3. | Nama objek : nama kelas | Objek | Menyatakan objek yang berinteraksi pesan |
| 4. |  | Waktu aktif | Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya. |
| 5. |  | Pesan tipe *create* | Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat |
| 6. |  | Pesan tipe *call* | Menyatakan suatu objek memanggil opeasi/metode yang ada pada objek  lain atau dirinya sendiri |
| 7. |  | Pesan tipe *send* | Menyatakan bahwa suatu objek mengirim data / masukkan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirimi |
| 8. |  | Pesan tipe *return* | Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian |
| 9. |  | Pesan *type* *destroy* | Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri,  sebaiknya jika ada create makan ada *destroy* |

Sumber: (Kiswanto, 2016).

## ***Entity Relational Diagram* (ERD**)

*Entity Relationship Diagram* (ERD) adalah model teknik pendekatan yang menyatakan atau menggambarkan hubungan suatu model. Didalam hubungan ini tersebut dinyatakan yang utama dari ERD adalah menunjukan objek data (*entity*) dan hubungan (*relationship*), yang ada pada *entity* berikutnya. ERD digunakan untuk menggambarkan *database* untuk pemprosesan transaksi online atau Online Transaction Processing (OLTP). Proses memungkinkan analis menghasilkan struktur basis data dapat disimpan dan diambil secara efisien (Edi dan Betshani, 2012).

Simbol-simbol dalam *Entity Relationship Diagram* (ERD) yaitu pada tabel 2.6 sebagai berikut:

Tabel 2.6 Entity Relationship Standard

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Deskripsi** |
|  | Entitas | Suatu yang nyata atau abstrak yang mempunyai karakteristik dimana kita akan menyimpan data |
|  | Attribut | Ciri umum semua atau sebagian besar instansi pada entitas tertentu. |
|  | Relasi | Hubungan alamiah yang terjadi antara satu atau lebih entitas |
|  | *Link* | Garis penghubung atribut dengan kumpulan entitas dan kumpulan entitas dengan relasi |
|  | Relasi 1 : 1 | setiap entitas pada himpunan entitas pertama berhubungan dengan paling banyak satu entitas pada himpunan entitas kedua |
|  | Relasi N : N | setiap entitas pada himpunan entitas yang pertama dapat berhubungan dengan banyak entitas pada himpunan entitas yang kedua, demikian juga sebaliknya |
|  | Relasi 1 : N | Hubungan antara entitas pertama  Dengan entitas kedua adalah satu  Banding banyak atau sebaliknya. |
|  | Atribut composite | Atribut Komposit yaitu atribut yang dapat dipecah menjadi bagian-bagian yang lebih kecil dan tetap bermakna. |

* 1. **Rancangan Pengujian *Black Box***

Pengertian dari *black box testing* merupakan teknik pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak. Pengujian bekerja dengan mengabaikan struktur pada control sehingga berfokus pada informasi domain. Pengujian menggunakan *black box testing* memungkinkan pengembang sistem untuk membuat himpunan kondisi input yang akan melatih seluruh batasan-batasan fungsional pada suatu sistem.

Keuntungan dalam menggunakan metode *black box testing* adalah pengujian yang tidak perlu memiliki pengetahuan tentang bahasa pemrograman tertentu. Pengujian dilakukan dari sudut pandang pengguna sehingga programmer dan tester keduanya saling bergantung satu sama lain (Hidayat dan Putri, 2019).

# BAB III METODOLOGI PENELITIAN

## **Waktu dan Tempat Penelitian**

### **Waktu**

Waktu pelaksanaan penelitian tugas akhir dilaksanakan mulai dari bulan Mei 2021 sampai dengan September 2021. Rincian kegiatan dapat dilihat pada Tabel 3.1 berikut:

Tabel 3.1 *Gannt Chart* Waktu Penelitian

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Waktu (2021)** | | | | | | | | | | | | | | | | | | |
| **No** | **Uraian** | **Mei** | | | **Juni** | | | | **Juli** | | | | **Agustus** | | | | **September** | | | |
| **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** |
| **1** | ***Inception*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **2** | ***Elaboration*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **3** | ***Construction*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **4** | ***Transition*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

## **Tempat Penelitian**

Adapun tempat penelitian tugas akhir yang akan dilakukan yaitu di Jurusan Teknik Informatika Fakultas Teknik Universitas Halu Oleo.

## **Metode Pengumpulan Data**

Dalam penelitian ini metode pengumpulan data yang digunakan pada perancangan aplikasi adalah kepustakaan, yang dilakukan dengan cara mengumpulkan referensi dengan objek penelitian yang dirujuk dalam penelitian ini berkaitan dengan kajian data yang dilakukan dengan cara membaca berbagai literatur, dokumen-dokumen dan sumber lainnya.

## **Metode Pengembangan Sistem**

Metode pengembangan sistem yang digunakan dalam sistem ini adalah metode *Rational Unified Process* (RUP). Dalam metode ini, terdapat empat tahap pengembangan perangkat lunak, yaitu:

### **Permulaan *(Inception)***

Pada fase ini dilakukan proses pengidentifikasian sistem, dilakukan dengan analisis kebutuhan akan sistem, melakukan kajian terhadap penelitian yang terkait dengan metode *Rabin Karp.*

### **Perluasaan / Perencanaan *(Elaboration)***

Setelah menentukan ruang lingkup penelitian, tahap ini akan dilakukan perancangan dan analisis sistem menggunakan flowchart meliputi flowchart Sistem pendeteksi plagiarisme, dan flowchart metode *Rabin Karp*. Pada perancangan ini, digunakan juga *UML* (*Unified Modelling Language*) yang meliputi *use Case diagram, activity diagram, class diagram* dan *sequence diagram.*

### **Konstruksi *(Construction)***

Proses yang dilakukan pada tahap ini yaitu membangun aplikasi dengan perancangan yang telah dilakukan sebelumnya, mulai dari tampilan *interface* sampai implementasi rancangan UML.

### **Transisi *(Transition)***

Pada tahap ini difokuskan untuk melakukan pengujian terhadap sistem pencarian. Memperbaiki masalah yang muncul saat pembuatan dan setelah pengujian aplikasi.

# BAB IV ANALISIS DAN PERANCANGAN SISTEM

* 1. **Analisis Kebutuhan Sistem**

Analisis kebutuhan sistem bertujuan untuk mengidentifikasi permasalahan-permasalahan yang ada pada sistem dimana sistem dibangun, meliputi perangkat lunak, dan hasil analisis terhadap sistem serta elemen-elemen sistem. Pembahasan yang ada pada analisis sistem ini yaitu analisis kebutuhan fungsional dan analisis kebutuhan nonfungsional.

### **Kebutuhan Fungsional**

Analisis kebutuhan fungsional adalah segala bentuk data yang dibutuhkan oleh sistem agar sistem dapat berjalan sesuai dengan prosedur yang dibangun melalui perancangan sistem. Adapun kebutuhan fungsional dari sistem yang akan dibangun yaitu Perancangan diagram sistem menggunakan bahasa peModelan UML (*Unified Modeling Language*) yang meliputi *use Case diagram, activity diagram, class diagram, serta sequence diagram* dan Desain *Interface* sistem. Informasi data mahasiswa Universitas Halu Oleo.

### **Kebutuhan Nonfungsional**

Analisis kebutuhan nonfungsional adalah sebuah langkah dimana seorang pembangun aplikasi menganalisis sumber daya yang dibutuhkan untuk membangun aplikasi yang akan dibangun. Analisis kebutuhan nonfungsional yang dilakukan dibagi dalam dua tahap, yaitu analisis kebutuhan perangkat keras dan analisis kebutuhan perangkat lunak.

* + - 1. Kebutuhan Perangkat Keras

Perangkat keras yang digunakan pada pembangunan sistem ini yaitu pada tabel 4.1 sebagai berikut

Tabel 4.1 Spesifikasi Perangkat

|  |  |  |
| --- | --- | --- |
| **No** | **Nama perangkat** | **Spesifikasi** |
| 1. | *PC* | *Acer* |
| 2. | *Processor* | *Intel dual core* |
| 3. | *RAM* | *4 GB* |
| 4. | *Harddisk* | *128 GB SSD* |
| 5. | *Monitor* | *Monitor 14 Inch* |

* + - 1. Kebutuhan Perangkat Lunak

Perangkat lunak yang digunakan pada pembangunan sistem ini yaitu pada tabel 4.2 sebagai berikut:

Tabel 4.2 Spesifikasi Perangkat Lunak

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Nama Perangkat** | **Fungsi** | **Spesifikasi** |
| 1. | *Windows* | *Operating System* | *Windows 10 Home* |
| 2. | *Xampp* | *Universal development environment* | *Xampp v3.2.2* |
| 3. | *Apache* | *Web Server* | *Apache 2.4 win64-VC14* |
| 4. | *Mysql* | *Database Management* | *Mysql 5.7* |
| 5. | *PHP* | *Web development* | *PHP 7.3* |
| 6. | *Chrome* | *Web browser* | *Chrome version 76.0* |
| 7. | *Visual Studio Code* | *Code Editor* | *VS Code v1.38* |

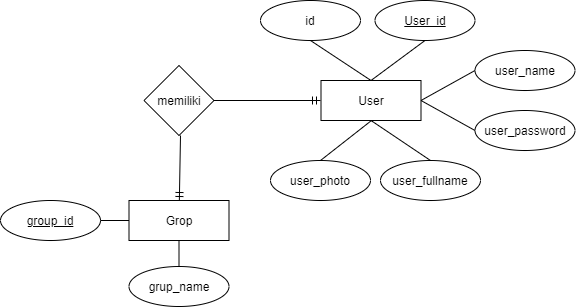
* 1. **Analisis Perancangan Sistem**

Perancangan sistem yang akan dibangun yaitu perancangan *Entity Relationship Diagram*, UML dan perancangan *user interface* sistem*.*

### ***Entity Relationship* *Diagram***

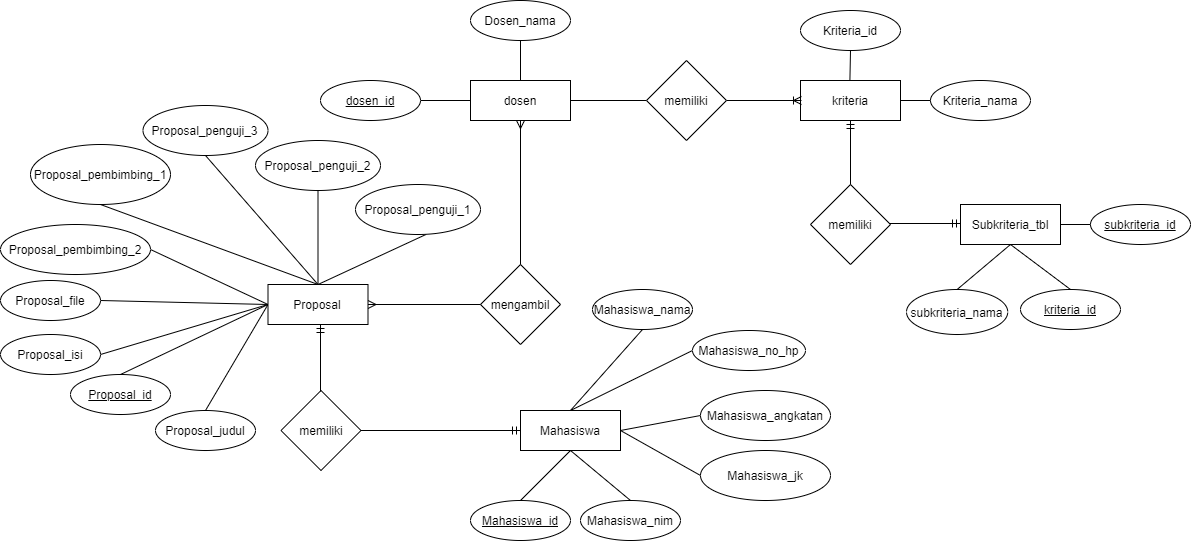
Untuk perancangan *database* dari sistem yang akan dikembangkan, pengembang menggunakan *entity relationship* diagram sebagai Model *entity relationship* *diagram* dari sistem.

*Logical Model*



Gambar 4.1 *Entity Diagram* Hak Akses

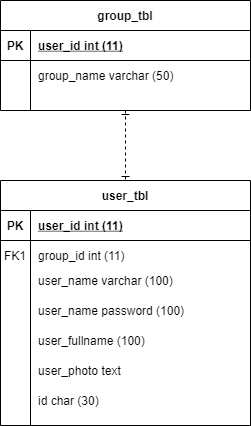
Pada gambar 4.1 merupakan diagram *logical Model* untuk tabel hak akses berupa *user* dan *group*.



Gambar 4.2 *Entity Relationship Diagram* Utama

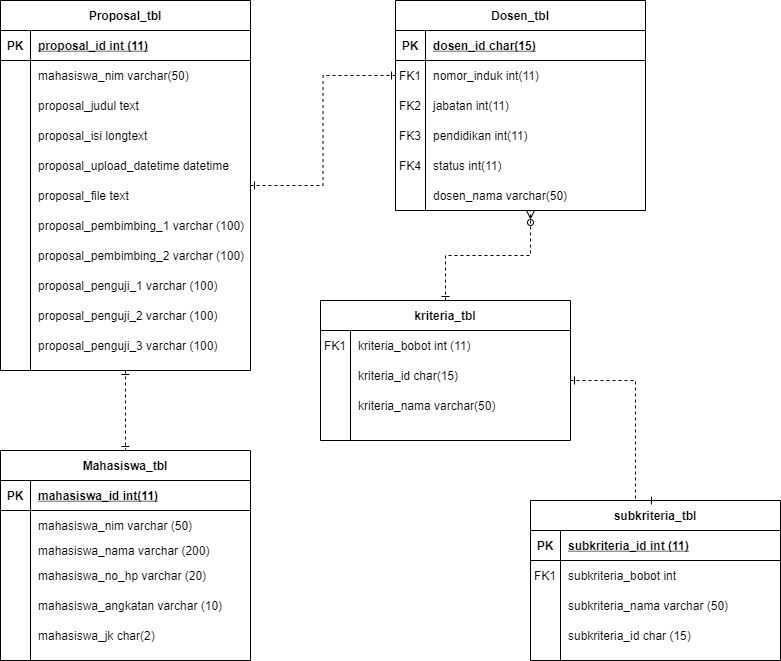
Pada gambar 4.2 merupakan diagram *logical Model* yang utama berupa proposal, dosen, mahasiswa, kriteria, dan subkriteria.

*Physical Model*



Gambar 4.3 *Physical Model* dari ERD hak akses

Pada gambar 4.3 merupakan *physical Model* dari *entity relationship* diagram. Model inilah yang selanjutnya diimplementasikan pada *database* sistem



Gambar 4.4 *Physical* *Model* dari ERD menu utama

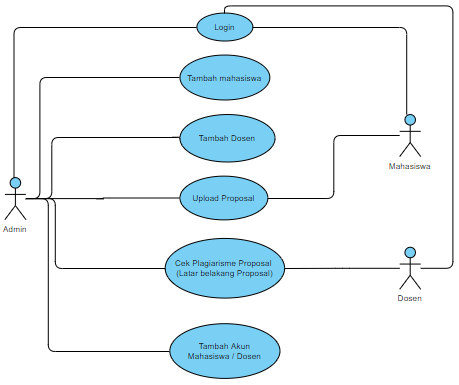
Pada gambar 4.4 merupakan *physical Model* dari *entity relationship* diagram menu. Model inilah yang selanjutnya diimplementasikan pada *database* bagian menu sistem.

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

Aplikasi dibangun dengan menggunakan *Unified Modeling Language* (UML). UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram yang terdiri dari *Use Case Diagram, Activity Diagram, Class Diagram dan Sequence Diagram.*

1. ***Use Case Diagram***

*Use Case* *Diagram* adalah sebuah diagram yang dapat merepresentasikan interaksi yang terjadi antara user dengan sistem. *Use Case Diagram* ini mendeskripsikan siapa saja yang menggunakan sistem dan bagaimana cara mereka berinteraksi dengan sistem. *Use Case* *Diagram* dari sistem yang akan dibangun yaitu pada gambar 4.5 sebagai berikut.

****

Gambar 4.5 *Use Case Diagram*

Tabel 4.3 Deskripsi *Case Login (Admin,* Dosen dan Mahasiswa)

|  |  |
| --- | --- |
| **Nama** | ***Login*** |
| Deskripsi Singkat: | Proses Keamanan login untuk *Admin* dan Mahasiswa |
| Persyaratan | *Admin,* Dosen, dan Mahasiswa harus memasukkan *username* dan *password* |
| Kondisi Akhir | Masuk untuk manage sistem |
| Situasi Kesalahan | * + 1. Yang masuk bukan *Admin,* Dosen, dan Mahasiswa     2. Salah memasukkan *username* dan *password* |
| Status sistem dalam  saat terjadi kesalahan | 1. Menampilkan pesan *error* 2. Tidak dapat masuk kedalam sistem |
| Aktor | *Admin,* Dosen, dan Mahasiswa |
| Trigger | Keamanan *login* untuk *Admin* |
| Proses Standar | 1. *Admin,* Dosen, dan Mahasiswa masuk ke sistem 2. *Admin,* Dosen, dan Mahasiswa memasukkan *username* 3. *Admin,* Dosen, dan Mahasiswa memasukkan *password* |
| Proses Alternatif | Mengecek *username* dan *password* |

Tabel 4.4 Deskripsi *Case* Mahasiswa (Admin)

|  |  |
| --- | --- |
| **Nama** | **Mahasiswa** |
| Deskripsi Singkat: | *Admin* ingin melihat atau Menambah data mahasiswa |
| Persyaratan | *Admin* harus melakukan *login* terlebih dahulu |
| Kondisi Akhir | Data Mahasiswa di tampilkan |
| Situasi Kesalahan | - |
| Status sistem dalam saat terjadi kesalahan | - |
| Aktor | *Admin* |
| Trigger | *Admin* ingin menambahkan, menghapus atau memperbarui data mahasiswa |
| Proses Standar | 1. *Admin* masuk ke sistem dan telah melakukan proses *login* 2. *Admin* melihat data Mahasiswa 3. *Admin* menambahkan Data Mahasiswa 4. *Admin* menghapus data Mahasiswa 5. *Admin* mengedit data Mahasiswa |
| Proses Alternatif | - |

Tabel 4.5 Deskripsi Dosen (admin)

|  |  |
| --- | --- |
| **Nama** | **dosen** |
| Deskripsi Singkat: | *Admin* ingin melihat atau Menambah data Dosen |
| Persyaratan | *Admin* harus melakukan *login* terlebih dahulu |
| Kondisi Akhir | Data Dosen di tampilkan |
| Situasi Kesalahan | - |
| Status sistem dalam saat terjadi kesalahan | - |
| Aktor | *Admin* |
| Trigger | *Admin* ingin menambahkan, menghapus atau memperbarui data Dosen |
| Proses Standar | 1. *Admin* masuk ke sistem dan telah melakukan proses *login* 2. *Admin* melihat data Dosen 3. *Admin* menambahkan Data Dosen 4. *Admin* menghapus data Dosen 5. *Admin* mengedit data Dosen |
| Proses Alternatif | - |

Tabel 4.6 Deskripsi *Case* Proposal (Admin dan mahasiswa)

|  |  |
| --- | --- |
| **Nama** | **Proposal** |
| Deskripsi Singkat: | Admin dan Mahasiswa ingin mengecek menambahkan / *upload* data proposal |
| Persyaratan | Admin dan Mahasiswa harus masuk kesistem terlebih dahulu |
| Kondisi Akhir | Data Proposal Tersimpan |
| Situasi Kesalahan | - |
| Status sistem dalam saat terjadi kesalahan | - |
| Aktor | Admin dan Mahasiswa |
| Trigger |  |
| Proses Standar | 1. Admin dan Mahasiswa masuk kesistem 2. Admin dan Mahasiswa input Latar belakang proposal 3. Data proposal di tampilkan |
| Proses Alternatif | *-* |

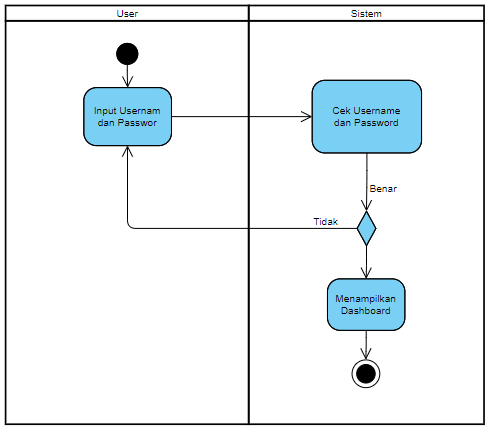
Tabel 4.7 Deskripsi Cek Plagiarisme

|  |  |
| --- | --- |
| **Nama** | **Plagiarisme** |
| Deskripsi Singkat: | Admin dan Dosen ingin mengecek plagiarisme |
| Persyaratan | Admin dan Dosen harus masuk kesistem terlebih dahulu |
| Kondisi Akhir | Data Latar belakang Proposal persentase Plagiarisme harus di tampilkan |
| Situasi Kesalahan | - |
| Status sistem dalam saat terjadi kesalahan | - |
| Aktor | Admin dan Dosen |
| Trigger |  |
| Proses Standar | 1. Admin dan Dosen masuk kesistem 2. Admin dan Dosen input Latar belakang proposal 3. Latar belakang proposal dan persentase plagiarismenya ditampilkan |
| Proses Alternatif | *-* |

1. ***Activity diagram***

*Activity Diagram* menggambarkan berbagai alir aktivitas dalam sistem yang sedang di rancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Berikut ini adalah *activity* diagram yang akan menggambarkan alir aktivitas sistem.

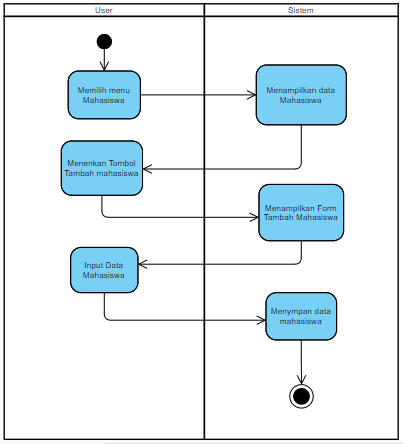
1. *Activity Diagram Login*



Gambar 4.6 *Activity diagram login*

Pada Gambar 4.6 merupakan diagram aktivitas yang menunjukkan aktivitas *admin* dan *user* ketika memasukkan *username* dan *password*, lalu sistem akan menampilkan Dashboard.

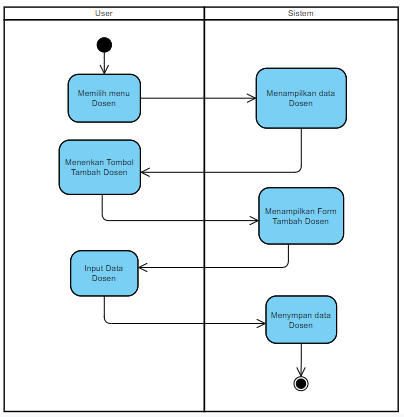
1. *Activity Diagram* Mahasiswa



Gambar 4.7 *Activity Diagram* Mahasiswa

Pada Gambar 4.7 merupakan diagram aktivitas yang menunjukkan aktivitas *admin* memilih menu Mahasiswa untuk menambahkan Data Mahasiswa agar Mahasiswa dapat melakukan Upload data Proposal.

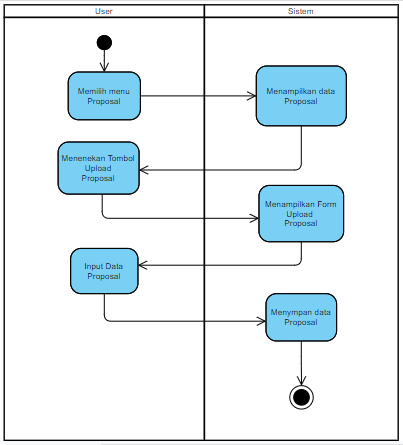
1. *Activity Diagram* Dosen



Gambar 4. 8 *Activity* Diagram Dosen

Pada Gambar 4.8 merupakan diagram aktivitas yang menunjukkan aktivitas admin Menambah Data Dosen agar dapat di masukkan dalam data proposal.

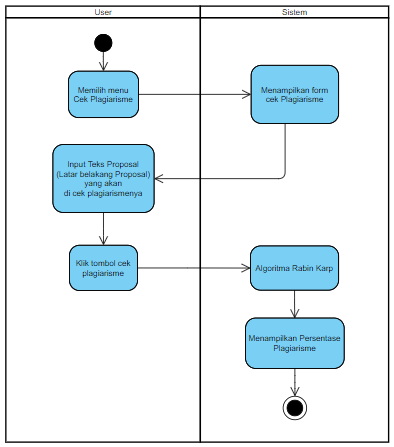
1. *Activity Diagram* Proposal



Gambar 4.9 *Activity Diagram* Proposal

Pada Gambar 4.9 merupakan diagram aktivitas yang menunjukkan aktivitas *admin* memilih menu Proposal untuk menambahkan / *upload* data proposal.

1. *Activity diagram* Cek Plagiarisme

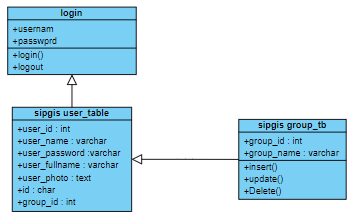


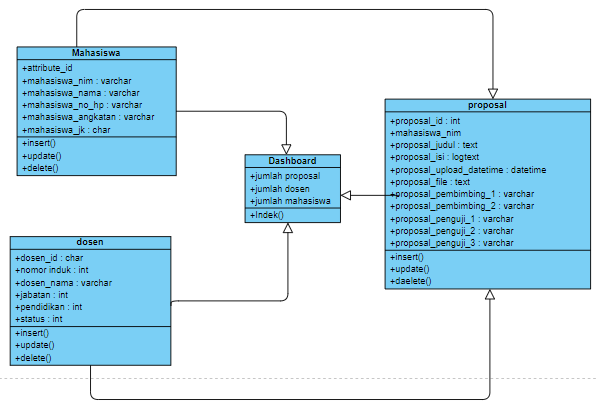
Gambar 4.10 *Activity Diagram* Plagiarisme

Pada Gambar 4.10 merupakan diagram aktivitas yang menunjukkan aktivitas Admin dan dosen melakukan cek plagiarisme pada latar belakang teks proposal.

1. ***Class diagram***

*Class diagram* adalah diagram yang menjelaskan tentang pemodelan sistem berorientasi objek. *Class diagram* juga menujukan hubungan *Class* dengan sistem yang sedang dibangun dan bagaimana mereka saling berinteraksi untuk mencapai suatu tujuan. Berikut ini *Class diagram* sistem yang akan dibangun.





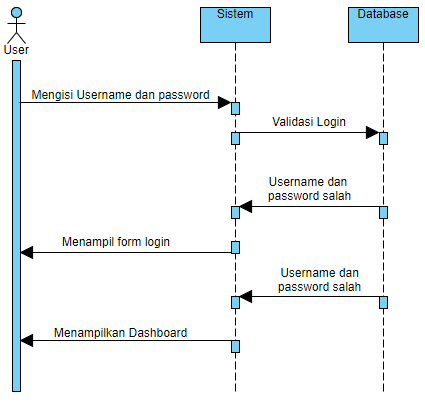
Gambar 4.11 *Class diagram*

1. ***Sequence Diagram***

*Sequence Diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem yang digambarkan terhadap waktu. Berikut ini adalah *Sequence Diagram* yang akan menggambarkan interakasi antar objek dan sistem.

1. *Sequence Diagram Login*

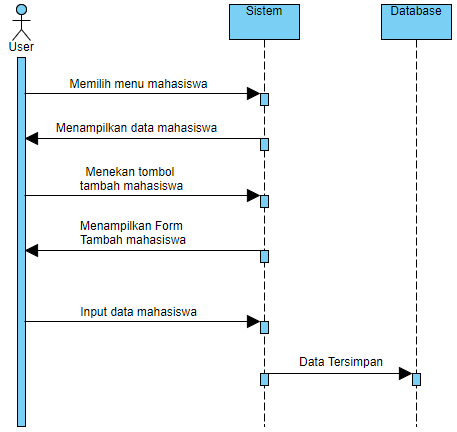
Pada Gambar 4.12 adalah *sequence* yang menunjukkan proses *login* bagi admin dan user dimana *actor* memasukkan *username* dan *password* lalu divalidasi oleh *database*.



Gambar 4.12 *Sequence Diagram Login*

1. *Sequence Diagram* Mahasiswa

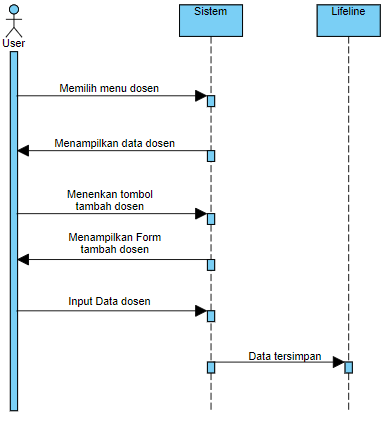
Gambar 4.13 adalah *sequence diagram* yang menjelaskan tentang Aktifitas Admin untuk menampilkan data mahasiswa dan menambahkan data mahasiswa agar dapat melakukan *upload* proposal.



Gambar 4.13 *Sequence Diagram* Mahasiswa

1. *Sequence Diagram* Dosen

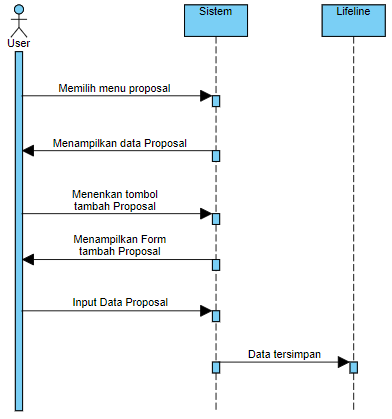
Gambar 4.14 merupakan *Sequence Diagram* yang menunjukkan aktivitas *admin* menampilkan data dosen dan menambahkan data dosen.



Gambar 4.14 *Sequence Diagram* Dosen

1. *Sequence Diagram* Proposal

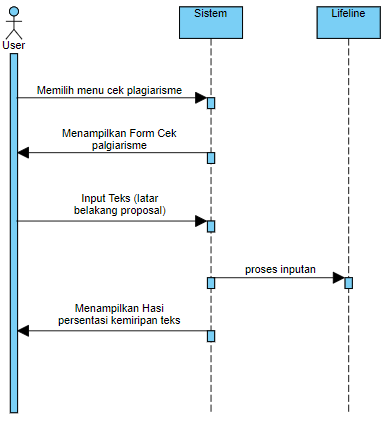
Pada Gambar 4.15 merupakan *Sequence Diagram* yang menunjukkan Proses *Admin* dan mahasiswa memilih menu Proposal untuk melakukan *upload* proposal yang di jadikan sebagai data uji.



Gambar 4.15 *Sequence Diagram* Proposal

1. *Sequence Diagram* Cek plagiarisme

Pada Gambar 4.16 merupakan *Sequence Diagram* yang menunjukkan Proses admin dan dosen memilih menu Cek plagiarisme untuk mengecek tingkat persentase kemiripan Latar belakang proposal Mahasiswa.



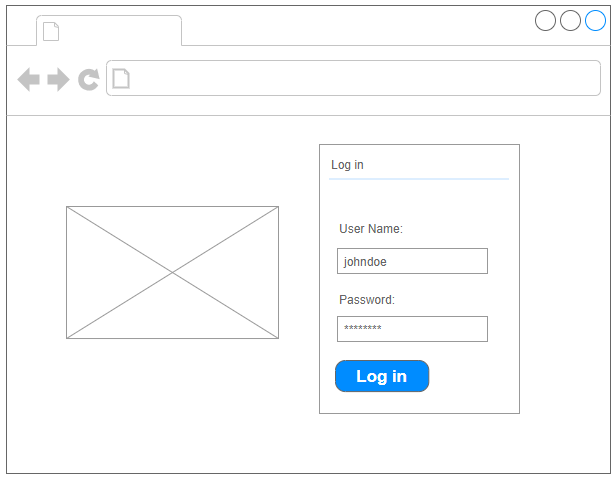
Gambar 4.16 *Sequence Diagram* Cek Plagiarisme

### **Perancangan Antarmuka (*Interface*)**

Rancangan antarmuka pengguna atau *design user interface* merupakan penggambaran tampilan yang digunakan secara langsung oleh pengguna terhadap sistem. Rancangan *user* *interface* ini dibuat sederhana agar mudah dimengerti pengguna dan tidak ada kerumitan dalam menjalankannya sehingga mencapai tujuan perangkat lunak yang *user friendly*.

* + - 1. **Menu *Login***

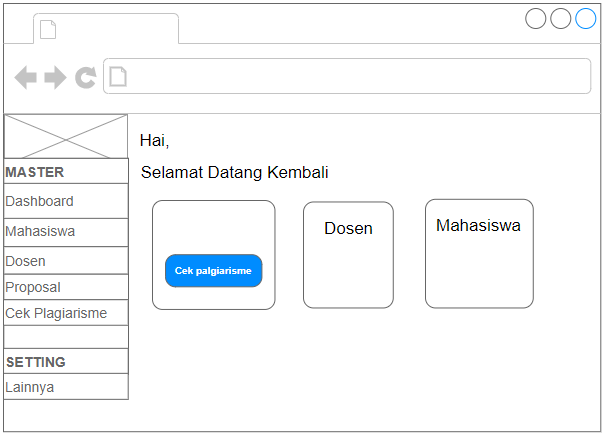
Halaman *login* pada *website* menampilkan *form login* yang terdiri dari *username* dan *password*. Setiap *field* harus diisi agar dapat melakukan *login*. Bentuk dari tampilan *login* dapat dilihat pada gambar berikut.



Gambar 4.17 Menu *Login*

* + - 1. **Menu *Dashboard***

Tampilan *Dashboard* adalah tampilan pertama yang akan digunakan pada sistem ini. Pada tampilan *Dashboard* berisi logo dan menu-menu pada sistem. Bentuk dari tampilan menu *Dashboard* dapat dilihat pada gambar berikut.

****

Gambar 4.18 Menu *Dashboard*

* + - 1. **Menu Mahasiswa**

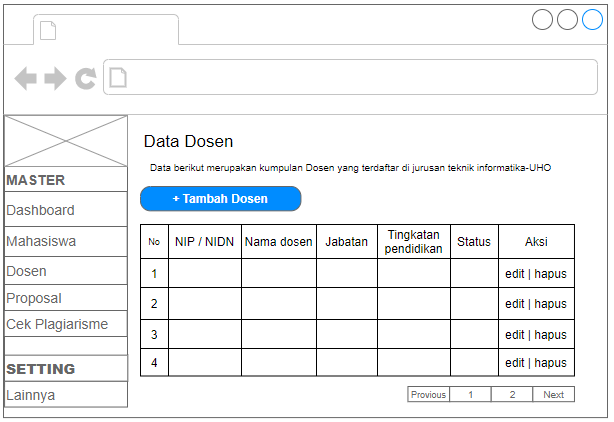
Menu mahasiswa menampilkan *form* tambah data mahasiswa agar mahasiswa yang telah di input datanya dapat melakukan tambah atau *upload* data proposal. Bentuk dari tampilan menu Mahasiswa dapat dilihat pada gambar berikut.

****

Gambar 4.19 Mahasiswa

* + - 1. **Menu Dosen**

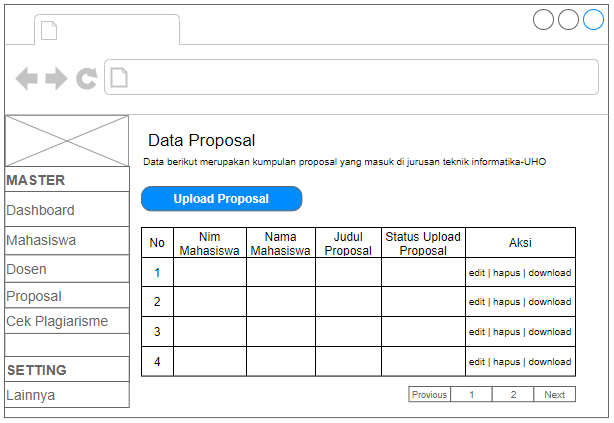
Menu dosen menampilkan *form* tambah data dosen agar data dosen yang telah di input datanya dapat di masukkan dalam proses upload data proposal. Bentuk dari tampilan menu dosen dapat dilihat pada gambar berikut.



Gambar 4.20 Menu Dosen

* + - 1. **Menu Proposal**

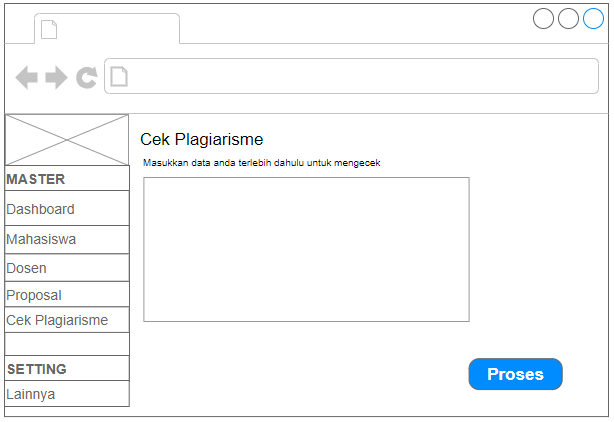
Menu Proposal menampilkan *form* untuk admin dan mahasiswa dapat melakukan Tambah. Atau *upload* proposal yang akan di cek plagiarismenya. Bentuk dari tampilan menu Proposal dapat dilihat pada gambar berikut.

****

Gambar 4.21 Proposal

* + - 1. **Menu Cek Plagiarisme**

Menu cek plagiarisme menampilkan *form* untuk *admin* dan Dosen dapat melakukan cek plagiarisme pada teks Latar belakang proposal. Bentuk dari tampilan menu cek plagiarisme dapat dilihat pada gambar berikut.

****

Gambar 4.22 Cek Plagiarisme

BAB V   
IMPLEMENTASI DAN PEMBAHASAN

* 1. **Implementasi sistem**

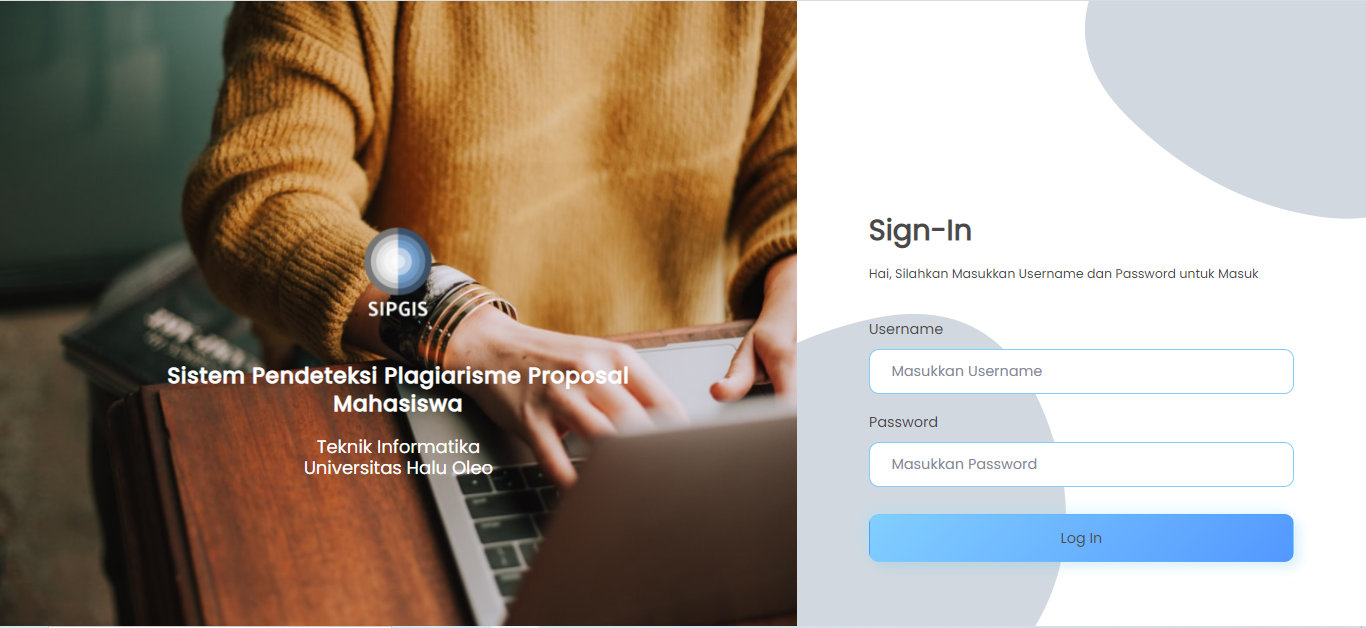
Pada tahap Implementasi merupakan tahap penerapan kode program yang dilakukan untuk membuat aplikasi berdasarkan rancangan dan desain sistem. Pada tahap ini akan dijelaskan mengenai data yang digunakan pada sistem, implementasi *interface*, implementasi Algoritma *Rabin Karp* dan pengujian sistem.

* 1. **Implementasi *Interface***

Adapun implementasi *interface* dibuat berdasarkan perancangan *user interface* yang ada yaitu sebagai berikut.

Halaman *Login*

Pada gambar 5.1 merupakan halaman *Login* dimana staf, dosen, dan mahasiswa memasukkan *username* dan *password* terlebih dahulu untuk bisa masuk kedalam sistem.



Gambar 5.1 Tampilan *Login* (Admin, Mahasiswa dan dosen)

Menu *dashboard*

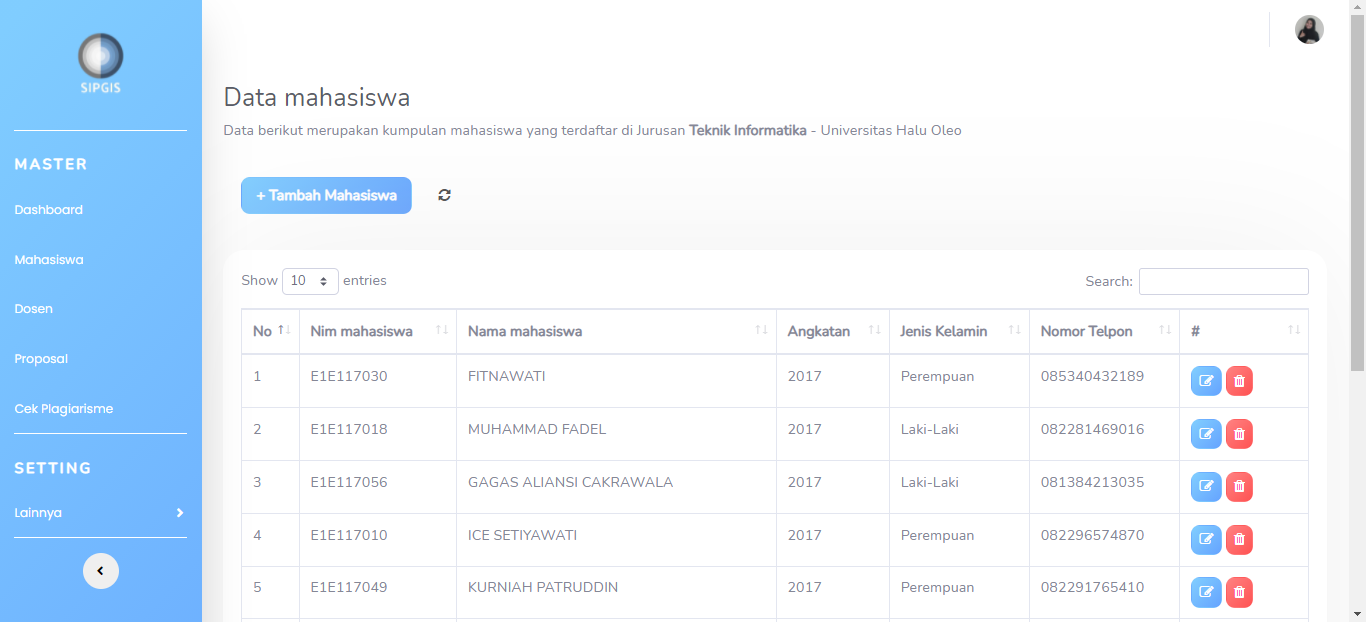
Pada gambar 5.2 merupakan halaman pertama setelah *user* berhasil melakukan *Login* kedalam sistem. Halaman ini berisi tentang jumlah Proposal, jumlah dosen, dan jumlah mahasiswa.



Gambar 5.2 Tampilan Halaman Beranda (Admin)

Menu Mahasiswa

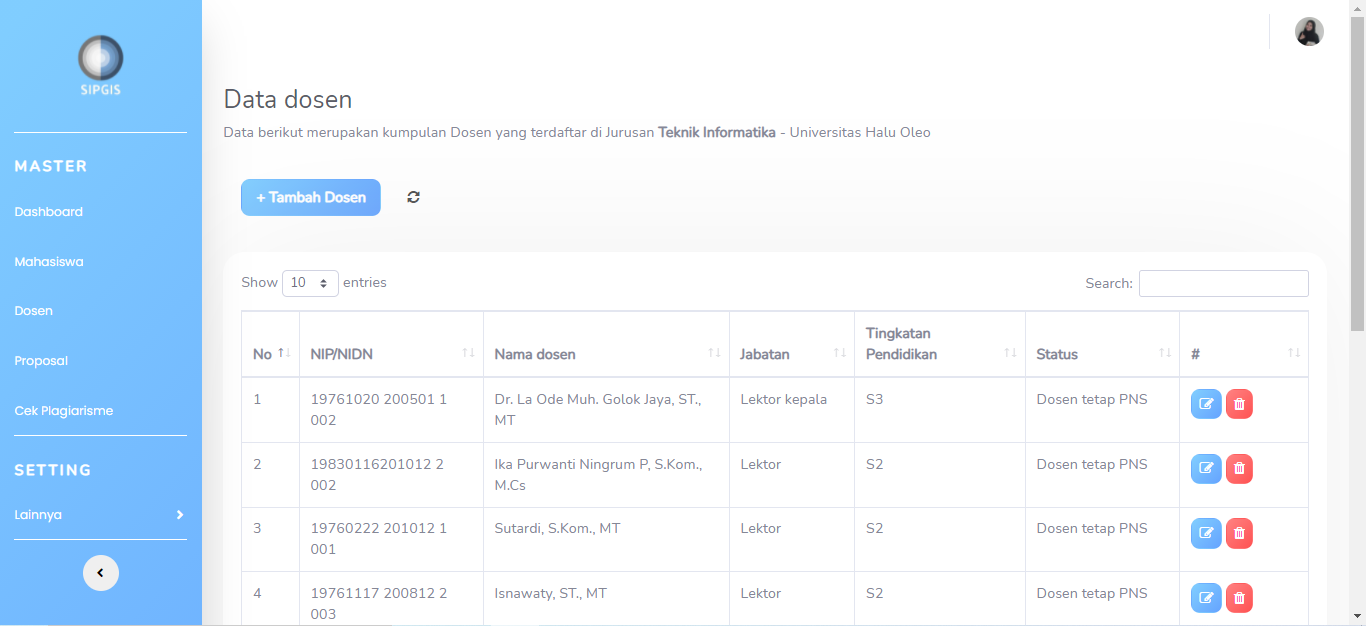
Pada gambar 5.3 merupakan halaman mahasiswa setelah admin memilih menu mahasiswa. Halaman ini berisi tentang data mahasiswa dan juga beberapa tombol yaitu tombol tambah data, hapus dan edit data mahasiswa.



Gambar 5.3 Tampilan Halaman Mahasiswa (Admin)

Menu Dosen

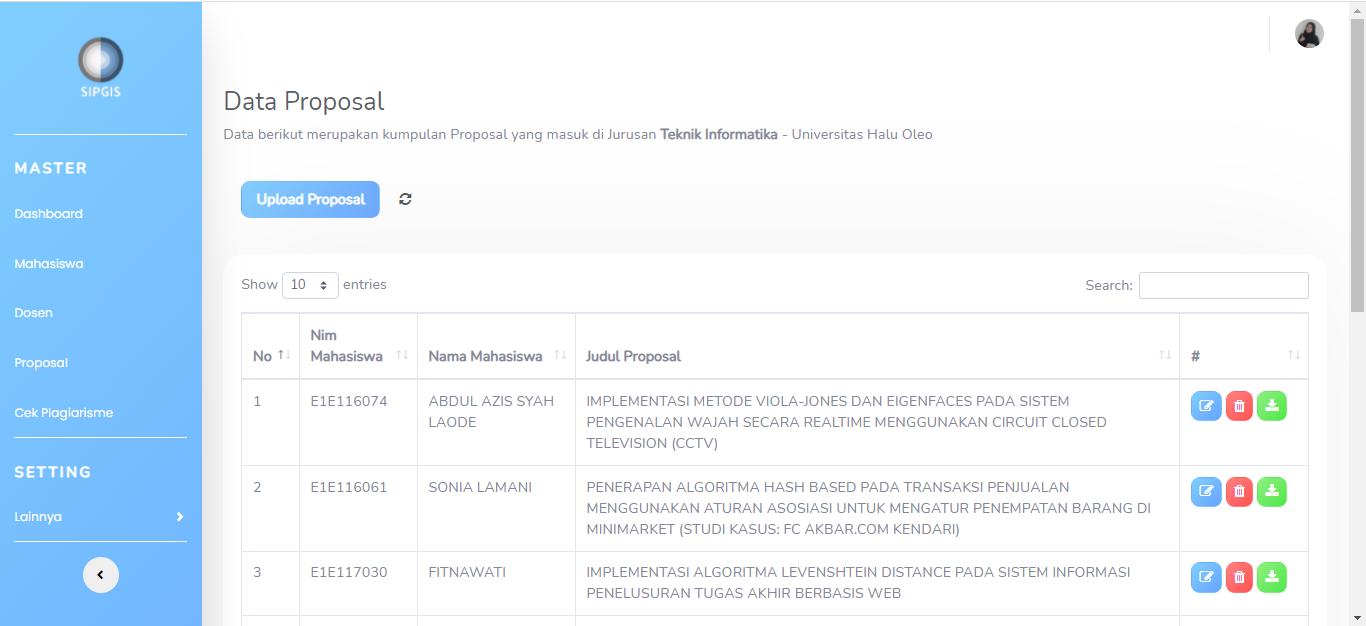
Pada gambar 5.4 merupakan halaman dosen setelah admin memilih menu dosen. Halaman ini berisi tentang data dosen dan juga beberapa tombol yaitu tombol tambah data, hapus dan edit data dosen.



Gambar 5.4 Tampilan Halaman Dosen (Admin)

Menu proposal

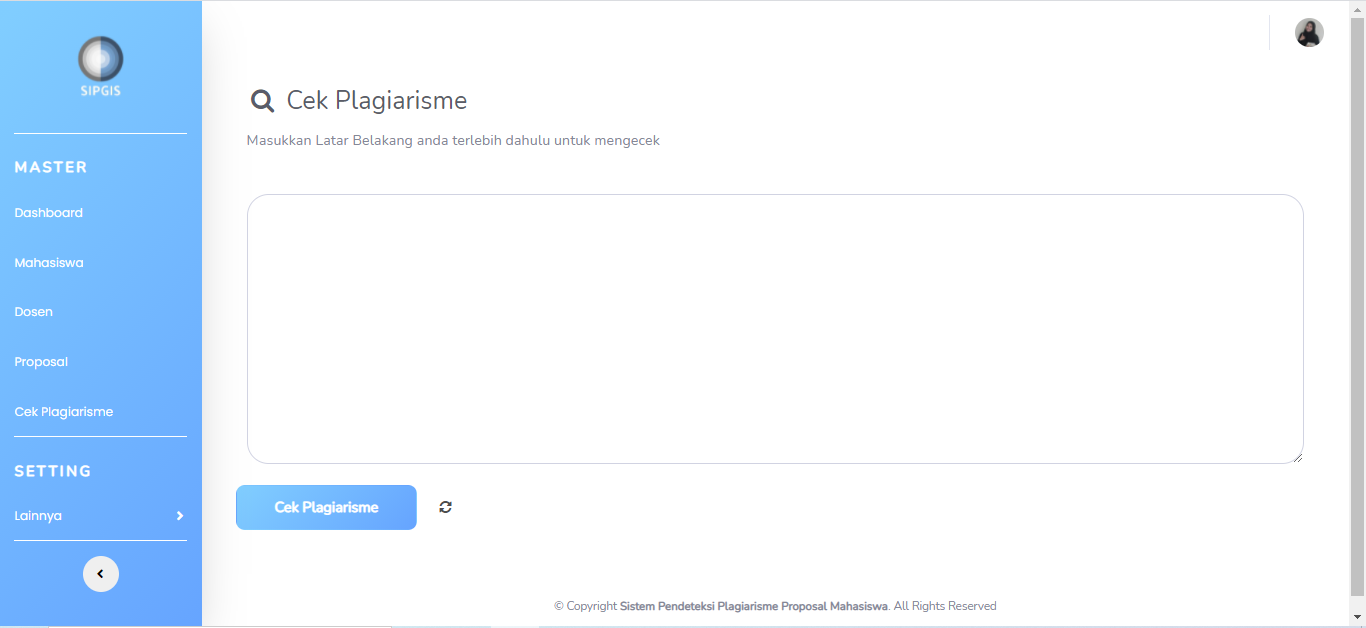
Pada gambar 5.5 merupakan halaman Proposal setelah admin memilih menu Proposal. Halaman ini berisi tentang data Proposal, admin dapat melihat data Proposal dan dapat melakukan proses edit, tambah dan hapus.



Gambar 5.5 Tampilan Halaman Proposal (Admin dan mahasiswa)

Menu Cek Plagiarisme

Pada gambar 5.6 merupakan halaman Cek Plagiarisme setelah admin memilih menu cek plagiarisme. Halaman ini berisi form untuk memasukkan teks yang akan di cek prsentase plagiarisme atau kemiripannya.



Gambar 5.6 Tampilan Halaman Cek Plagiarisme (Admin dan Dosen)

* + - 1. Hasil pengecekan plagiarisme

Berdasarkan gambar 5.7 menjelaskan bahwa contoh data yang di input yaitu data dari mahasiswa atas nama Muh. Darul zulkifli ritom menghasilkan similarity 100% dengan datanya sendiri, 56.2% dengan data Nur Hazmila, 49.1% dengan data Annisa fitri hamid, 48.8% dengan data Sonia lamani, 47.9% dengan data Atry Ilma Juni Rahim, 47.9% dengan data Muh. Saiful, 47.5% dengan data Ayu Windiarti, 47.4% dengan data Rifa’atus shalihah, 46.8% dengan data Nurvila, 46.6% dengan data Muhammad abdi fahmy.



Gambar 5.7 Tampilan Hasil Cek Plagiarisme (Admin dan Dosen)

* 1. **Pengujian Sistem**

Pengujian merupakan tahap yang utama dalam pembuatan suatu aplikasi. Hasil dari pengujian yang didapat, akan dijadikan sebagai tolak ukur dalam proses pengembangan sistem selanjutnya. Pada pengujian ini dilakukan untuk mengetahui hasil dari pembuatan sistem.

* + 1. **Perhitungan Manual Algoritma Rabin Karp**

Berikut ini adalah contoh perhitungan kesamaan dokumen yang akan di proses dengan menggunakan metode *Rabin Karp*.

* + - 1. Tahap Input Parameter

Proses sistem yang akan dibuat dimulai dari memasukkan dokumen yaitu dokumen asli dan dokumen uji.

Dokumen asli yang di input :

|  |
| --- |
| Komputer adalah perangkat “Elektronik” |

Dokumen uji yang di input :

|  |
| --- |
| Komputer adalah perangkat yang membutuhkan listrik. |

Dengan *K-gram* k= 4

Basis bilangan b= 11

* + - 1. Tahapan *Preprocessing*

Tahap *Preprocessing* harus dilalui untuk menentukan *keyword* pada kedua dokumen yang akan dilakukan pengujian, yaitu dokumen asli dan dokumen uji. Pada tahap *preprocessing* terdapat beberapa proses yang dilakukan oleh sistem terhadap dokumen yang diinputkan. Proses-proses tersebut antara lain *Case folding, filtering, Tokenizing.*

Sub Proses *Case Folding*

Proses *Case folding* (mengubah huruf kapital menjadi huruf kecil) merupakan tahap pertama yang akan dilakukan dari rangkaian tahapan yang terdapat pada *preprocessing.* Berikut adalah contoh proses *Case folding*:

Dokumen asli setelah melalui proses *Case folding* (proses 1.1)

|  |
| --- |
| komputer adalah perangkat “elektronik” |

dokumen uji setelah melalui proses *Case folding* (proses 1.1)

|  |
| --- |
| komputer adalah perangkat yang membutuhkan listrik. |

Pada contoh proses *Case folding* diatas, kalimat yang terdapat huruf kapitalnya adalah pada kata “Komputer”. Dimana pada kata tersebut setelah dilakukan proses *Case folding* akan diubah menjadi huruf kecil. Pada kata “Komputer” akan berubah menjadi “komputer”.Sehingga secara keseluruhan kata-kata pada dokumen yang terdapat huruf kapitalnya akan berubah menjadi huruf kecil semua.

* 1. Sub Proses *Filtering*

Setelah proses *Case* folding, dokumen selanjutnya masuk ketahap *filtering*, yaitu proses membuang kata yang tidak penting (stop word) dan membuang tanda baca. Berikut ini contoh proses *filtering*:

Dokumen asli setelah melalui proses *filtering* (proses 1.2)

|  |
| --- |
| komputer perangkat elektronik |

Dokumen uji setelah melalui proses *filtering* (proses1.2)

|  |
| --- |
| komputer perangkat membutuhkan listrik |

Pada contoh filtering diatas yaitu proses menghilangkan tanda baca dan menghilangkan *stop word.* Pada contoh dokumen asli tanda baca yang di hilangkan adalah tanda petik (“ “) dan Pada contoh dokumen uji tanda baca yang di hilangkan adalah tanda titik ( . ) sedangkan kata yang di anggap tidak penting dan dihilangkan pada dokumen asli yaitu kata “adalah” dan pada dokumen uji yaitu kata “adalah” dan kata “yang”.

* 1. Sub Proses *Tokenizing*

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

* + - 1. Proses Parsing *K-gram*

Langkah selanjutnya adalah parsing *k-gram*, K-gram adalah rangkaian terms dengan panjang K. Kebanyakan yang digunakan sebagai terms adalah kata, dimana pada proses ini kata dipecah menjadi potongan-potongan, setiap potongan mengandung karakter sebanyak k. Nilai *k-gram* sangat berpengaruh terhadap nilai persentase suatu kemiripan dokumen. pemilihan k-gram yang semakin kecil akan memperoleh nilai persentase kemiripan yang besar. Hal ini terjadi karena pada k-gram yang lebih sedikit, string yang dipotong lebih kecil sehingga kemungkinan untuk ditemukannya rangkaian karakter yang sama semakin besar. Semakin besarnya *k-gram*, maka mengandung karakter yang lebih banyak dibandingkan dengan *k-gram* yang lebih kecil sehingga menyebabkan rangkaian karakter yang ditemukan semakin berkurang sehingga menurunkan nilai similaritas. pada sistem ini nilai *k-gram* nya telah di tetapkan yaitu nilai *k-gram* = 4.

Hasil *K-gram* dokumen asli yang telah melalui proses tokenisasi (proses 2.1)

|  |
| --- |
| {komp} {ompu} {mput} {pute} {uter} {terp} {erpe} {rper} {pera} {eran} {rang} {angk} {ngka} {gkat} {kate} {atel} {tele} {elek} {lekt} {ektr} {ktro} {tron} {roni} {onik} |

Hasil *K-gram* dokumen uji yang telah melalui proses tokenisasi (proses 2.1)

|  |
| --- |
| {komp} {ompu} {mput} {pute} {uter} {terp} {erpe} {rper} {pera} {eran} {rang} {angk} {ngka} {gkat} {katb} {atbu} {tbut} {butu} {utuh} {tuhl} {uhli} {hlis} {list} {istr} {stri} {trik} |

Pada contoh parsing *K-gram* diatas adalah proses pembentukan pola kata dalam bentuk gram dengan panjang karakter k=4, sehinga isi kalimat pada dokumen asli dan dokumen uji dirubah dalam bentuk *k-gram*.

* + - 1. Proses *Hashing*

Kemudian dilakukan proses *hashing*, dimana pada proses ini hash berfungsi untuk mengkonvert setiap *string* menjadi bilangan. Dengan cara mengalikan nilai ASCII masing-masing huruf hasil *k-gram* dengan basis bilangan tertentu, dimana Dalam prosesnya digunakan basis yang biasanya adalah bilangan prima yang cukup besar, dengan tujuan agar meminimalkan terjadinya tabrakan. Dengan menggunakan persamaan, maka dapat dihitung hasil hashnya. Algoritma *Rabin Karp* didasarkan pada fakta jika dua buah *string* sama maka nilai hash valuenya pasti sama. Sebagai contoh kita ambil kata dari hasil *k-gram* yang pertama pada dokumen asli, yaitu kata {komp}.

Contoh proses hashing untuk menghitung nilai hash dari kata {komp}, dengan nilai **k= 4 dan b=11**

Nilai ASCII dari kata {komp}

ASCII(k)= 107

ASCII(o)= 111

ASCII(m)= 109

ASCII(p)= 112

H(C1 …. CK) =C1.b(K-1)+C2.b(K-2)+ C3.b(k-1)+ C4.b(k-1)+( C5.b(k-1))

H = (107 x 113 )+(111 x 112 )+(109 x 111 )+(112 x 110 )

H = (107 x 1331)+(111 x 121)+(109 x 11)+(112)

H = 142.417+ 13.431+ 1.199+112

H = 157.159

Contoh proses hashing untuk menghitung nilai hash dari kata {ompu}:

Nilai ASCII dari kata {ompu}

ASCII(o)= 111

ASCII(m)= 109

ASCII(p)= 112

ASCII(u)= 117

H(C1 …. CK) =C1.b(K-1)+C2.b(K-2)+ C3.b(k-1)+ C4.b(k-1)+( C5.b(k-1))

H = (111 x 113 )+(109 x 112 )+(112 x 111 )+(117 x 110 )

H = (111 x 1331)+(109 x 121)+(112 x 11)+(117)

H = 147.741 + 13.189 + 1.232 + 117

H = 162.279

Jadi nilai hash pada dokumen asli *k-gram* yang pertama adalah 157159, yang kedua 162.279, proses perhitungan nilai *hash* di ulang kembali hingga *k-gram* keseluruhan dihitung. Berikut ini adalah nilai hasil hash dari dokomen asli dan *hash* dokumen uji:

Hasil *hashing* dokumen asli:

|  |
| --- |
| {157159} {162279} {160034} {164606} {170988} {167983} {149558} {166511} {162644} {149402} {164784} {143657} {160147} {151223} {155531} {144362} {167906} {148717} {157262} {148768} {157818} {169521} {166480} {162313} |

Jumlah hash pada dokumen asli adalah= ∑ ***H***asli= 24

Hasil *hashing* dokumen uji:

|  |
| --- |
| {157159} {162279} {160034} {164606} {170988} {167983} {149558} {166511} {162644} {149402} {164784} {143657} {160147} {151223} {155528} {144338} {167657} {145988} {171154} {169805} {169604} {152762} {157834} {155060} {168460} {169452} |

Jumlah hash pada dokumen uji adalah= ∑ ***H***uji= 26

*Hash* yang sama:

|  |
| --- |
| {157159} {162279} {160034} {164606} {170988} {167983} {149558} {166511} {162644} {149402} {164784} {143657} {160147} {151223} |

Jumlah hash yang sama dari dokumen asli dan dokumen uji adalah = ∑ ***H***asliᴖ***H***uji

= 14

Setelah jumlah hash diketahui, yaitu hash pada dokumen asli sebanyak 24, hash dokumen uji sebanyak 26 dan hash yang sama sebanyak 14, maka proses selanjutnya adalah menghitung *similarity.*

* + - 1. Deteksi Kesamaan

Setelah diketahui nilai hashnya, jumlah *hash* pada dokumen asli 24, jumlah *hash* pada dokumen uji 26 dan hash yang sama pada kedua dokumen yaitu 14. Proses selanjutnya adalah menghitung *similarity* yaitu tingkat kesamaan dua dokumen dengan menggunakan persamaan yaitu barapa persen tingkat kasamaannya. *Similarity* didapat dari 2 dikali hasil *hash* yang sama di bagi dengan jumlah hash kedua dokumen dikali dengan seratus persen. Berikut ini adalah proses menghitung *similarity* dua dokumen di atas :

*Similarity* (asli, uji) = x 100 %

= x 100 %

= x 100 %

= x 100 %

= 56 %

Hasil perhitungan *similaritry* dari kedua dokumen diatas adalah 56%, dokumen yang diuji dapat di kategorikan sebagai plagiarisme sedang.

* + 1. **Pengujian Algoritma *Rabin Karp* pada sistem**

**Perbandingan *similarity* berdasarkan jumlah string data uji 4 kata & data uji 7 kata.**

Tabel 5.1 data uji 4 kata & data uji 7 kata

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | K-Gram | Basis | Waktu ( s ) | *Similarity* (%) |
| 1 | 2 | 3 | 0.743 | 81.5% |
| 2 | 2 | 7 | 0.762 | 74.1% |
| 3 | 2 | 11 | 0.491 | 74.1% |
| 4 | 3 | 3 | 0.676 | 73.1% |
| 5 | 3 | 7 | 0.733 | 61.5% |
| 6 | 3 | 11 | 0.75 | 57.7% |
| 7 | 4 | 3 | 0.992 | 56% |
| 8 | 4 | 7 | 0.866 | 56% |
| 9 | 4 | 11 | 0.896 | 56% |
| 10 | 5 | 3 | 0.999 | 58.3% |
| 11 | 5 | 7 | 0.788 | 54.2% |
| 12 | 5 | 11 | 0.911 | 54.2% |

**Perbandingan *similarity* berdasarkan jumlah string data asli 840 kata & data uji 708 kata.**

**Tabel 5.2 data asli 840 kata & data uji 708 kata.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | K-Gram | Basis | Waktu ( s ) | *Similarity* |
| 1 | 2 | 3 | 0.992 | 106.2% |
| 2 | 2 | 7 | 0.796 | 105.9% |
| 3 | 2 | 11 | 1.262 | 105.2% |
| 4 | 3 | 3 | 0.914 | 105.9% |
| 5 | 3 | 7 | 1.209 | 96.9% |
| 6 | 3 | 11 | 1.011 | 88.7% |
| 7 | 4 | 3 | 1.056 | 101.3% |
| 8 | 4 | 7 | 1.147 | 65.9% |
| 9 | 4 | 11 | 0.976 | 56.2% |
| 10 | 5 | 3 | 1.339 | 83.7% |
| 11 | 5 | 7 | 1.342 | 39.4% |
| 12 | 5 | 11 | 1.273 | 35.2% |

Berdasarkan ke dua table di atas menjelaskan bahwa :

Semakin kecil nilai k-gram maka semakin besar persentase kemiripan *(similarity*) yang dihasilkan

Nilai basis menentukan besaran tabrakan/nilai hash yang sama

Waktu yang di gunakan pengecekan plagiarisme untuk jumlah string data 4 kata dan 7 kata dengan string 450 kata dan 244 kata tidak jauh selisihnya.

Banyaknya kata dengan nilai k-gram yang di masukkan mempungaruhi akurat atau tidaknya hasil similarity data, misalnya pada jumlah string 450 dan 244 ada yang menghasilkan lebih dari 100% maka data tersebut tidak akurat.

Nilai k gram dan nilai basis yang di gunakan Pada Sistem Pendeteksi Plagiarisme Pada Proposal Tugas Akhir Mahasiswa adalah *k-gram* = 4 dan basis = 11 karena di antara kedua uji data nilai tersebut yang paling akurat.

**Hasil Uji Data Proposal Mahasiswa**

Peminatan Rpl

Tabel 5. 3 uji data 1 E1E117001

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nim Mahasiswa** | **Nama Mahasiswa** | **Judul Skripsi** | **Kemiripan (%)** |
| 13 | E1E116049 | Alfrido rahmat julianto pidani | Pengimplementasian Internet Of Things (IOT) Dalam Monitoring Kwh Meter Listrik Berbasis Mobile | 41.8% |
| 40 | E1E1 15 014 | Elsa meilan | Rancang Bangun Aplikasi Kanopi Otomatis Berbasis Android Menggunakan Logika Fuzzy Tsukamoto | 41.6% |
| 25 | E1E115096 | Muhammad abdi fahmy | Sistem Pakar Diagnosis Penyakit Hewan Sapi Dengan Metode Forward Chainin | 39.2% |
| 27 | E1E115093 | Yulina | Implementasi Intrusion Detection System (Ids) Dan Intrusion Prevention System (Ips) Menggunakan Jejaring Sosial Serta Email Sebagai Media Notifikasi | 38.6% |
| 36 | E1E115023 | Arlin | Pendeteksi Tingkat Kekeruhan Air Menggunakan Sensor Turbidity Dan Sensor Ph Berbasis Android Secara Real Time | 37.6% |
| 8 | E1E116074 | Abdul azis syah laode | Implementasi Metode Viola-Jones Dan Eigenfaces Pada Sistem Pengenalan Wajah Secara Realtime Menggunakan Circuit Closed Television (CCTV) | 37% |
| 42 | E1E115034 | Muhammad budi dharmawan p | Implementasi Latent Semantic Analiysis Method Untuk Mendeteksi Kemiripan Kalimat | 36.9% |
| 26 | E1E115095 | Muh.reza dirgantara | Implementasi Dan Pengujian Keberhasilan Email Phishing Menggunakan Teknik Social Engineering (Studi Kasus : Jurusan Teknik Informatika UHO) | 36.2% |
| 30 | E1E115054 | Ahmad khairun arsyad | Implementasi Levenshtein Distance Pada Aplikasi Pencarian Barang Di Berbagai E-Marketplace Menggunakan Teknik Web Scraping | 35.9% |
| 22 | E1E116005 | Ayu windiarti | Implementasi Metode Blum Blum Shub (BBS) Untuk Pengacakan Soal Kuis Pada Aplikasi Media Pembelajaran Ipa Tingkat Sekolah Dasar Kelas 6 Berbasis Mobile | 35.8% |

Tabel 5. 4 uji data 2 E1E117011

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nim Mahasiswa** | **Nama Mahasiswa** | **Judul Skripsi** | **Kemiripan (%)** |
| 35 | E1E115025 | La ode muhamad taufiq | Analisis Perbandingan Algoritma Penjadwalan Round Robin Dan Shortest Job Fisrt Untuk Manajemen Proses Dalam Single Processing | 38.8% |
| 19 | E1E116021 | Nurvila | Analisis Quality Of Service (Qos) Jaringan Nirkabel Pada Permainan Game Online Menggunakan Metode Peer Connection Queue (Pcq) Dengan Antrian Queue Tree | 34.6% |
| 12 | E1E116051 | Reza sanjaya | Aplikasi Perkiraan Curah Hujan Kota Kendari Menggunakan Metode Interval Type-2 Fuzzy Logic System | 35.6% |
| 4 | E1E117023 | Nur aziza tadjuddin | Data Uji 4 Kata | 4.3% |
| 20 | E1E116014 | Mardianti potto | Diagnosis Penyakit Diabetes Mellitus Dan Penentuan Pola Makan Menggunakan Metode Case Based Reasoning | 36.4% |
| 18 | E1E116023 | Pratiwi nur aisyah | Game Edukasi Berbasis Android Untuk Anak Usia Dini Menggunakan Linear Congruent Method (Lc | 37.9% |
| 9 | E1E116066 | Asrif fajar hidayat | Identifikasi Landmark Sella, Nasion Dan Menton Cephalometry Menggunakan Metode Convolutional Neural Networks (Cnn) | 31.7% |
| 2 | E1E117030 | Fitnawati | Implementasi Algoritma Levenshtein Distance Pada Sistem Informasi Penelusuran Tugas Akhir Berbasis Web | 39.7% |
| 28 | E1E115076 | Muh. Darul zulkifli ritom | Implementasi Algoritma Levenshtein Distance Pada Sistem Informasi Penelusuran Tugas Akhir Berbasis Web | 35.1% |
| 6 | E1E117005 | Annisa fitri hamid | Implementasi Algoritma Rc4 Pada Penggunaan Qr-Code Aplikasi Permohonan Pengajuan Hak Atas Tanah Milik Instansi Pemerintah Di Kabupaten Konawe Selatan | 35.9% |

Tabel 5. 5 uji data 3 E1E117038

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nim Mahasiswa** | **Nama Mahasiswa** | **Judul Skripsi** | **Kemiripan (%)** |
| 13 | E1E116049 | Alfrido Rahmat Julianto Pidani | Pengimplementasian Internet Of Things (Iot) Dalam Monitoring Kwh Meter Listrik Berbasis Mobile | 39% |
| 36 | E1E115023 | Arlin | Pendeteksi Tingkat Kekeruhan Air Menggunakan Sensor Turbidity Dan Sensor Ph Berbasis Android Secara Real Time | 37% |
| 27 | E1E115093 | Yulina | Implementasi Intrusion Detection System (Ids) Dan Intrusion Prevention System (Ips) Menggunakan Jejaring Sosial Serta Email Sebagai Media Notifikasi | 36% |
| 1 | E1E117040 | Muhamad Danil | Implementasi Global Navigation Satellite System (Gnss) Pada Sistem Presensi Berbasis Android Menggunakan Metode Spatial | 35.5% |
| 8 | E1E116074 | Abdul Azis Syah Laode | Implementasi Metode Viola-Jones Dan Eigenfaces Pada Sistem Pengenalan Wajah Secara Realtime Menggunakan Circuit Closed Television (Cctv) | 35.5% |
| 40 | E1E1 15 014 | Elsa Meilan | Rancang Bangun Aplikasi Kanopi Otomatis Berbasis Android Menggunakan Logika Fuzzy Tsukamoto | 35.1% |
| 23 | E1E115102 | Nurman Anshar | Sistem Pendukung Keputusan Pemilihan Hotel Di Kota Kendari Menggunakan Metode Technique For Others Preference By Similarity To Ideal Solution (Topsis) | 34.5% |
| 19 | E1E116021 | Nurvila | Analisis Quality Of Service (Qos) Jaringan Nirkabel Pada Permainan Game Online Menggunakan Metode Peer Connection Queue (Pcq) Dengan Antrian Queue Tree | 33.6% |
| 26 | E1E115095 | Muh.Reza Dirgantara | Implementasi Dan Pengujian Keberhasilan Email Phishing Menggunakan Teknik Social Engineering (Studi Kasus : Jurusan Teknik Informatika Uho) | 33.5% |
| 3 | E1E117027 | Atri Ilma Juni Rahim | Sistem Library Video Pembelajaran Applied Bahavior Analisis (Aba) Untuk Anak Autis Berbasis Mobile Menggunakan Metode Knuth Morris Pratt (Kmp) (Studi Kasus : Sekolah Luar Biasa Aksara Center) | 33.1% |

Tabel 5. 6 uji data 4 E1E117053

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nim Mahasiswa** | **Nama Mahasiswa** | **Judul Skripsi** | **Kemiripan (%)** |
| 10 | E1E116061 | Sonia Lamani | Penerapan Algoritma Hash Based Pada Transaksi Penjualan Menggunakan Aturan Asosiasi Untuk Mengatur Penempatan Barang Di Minimarket (Studi Kasus: Fc Akbar.Com Kendari) | 40.1% |
| 20 | E1E116014 | Mardianti Potto | Diagnosis Penyakit Diabetes Mellitus Dan Penentuan Pola Makan Menggunakan Metode Case Based Reasoning | 38.8% |
| 12 | E1E116051 | Reza Sanjaya | Aplikasi Perkiraan Curah Hujan Kota Kendari Menggunakan Metode Interval Type-2 Fuzzy Logic System | 38.7% |
| 39 | E1E116004 | Ayu Asriani | Penerapan Metode Autoregressive Integrated Moving Average (Arima) Pada Indeks Harga Konsumen (IHK) Dalam Inflation Forecasting System (Studi Kasus Badan Pusat Statistik Kota Kendari) | 38.3% |
| 33 | E1E115028 | Maulid | Implementasi Dan Analisis Access Point 5 Ghz Menggunakan Metode Manual Random Sampling Dan Converge Visualization | 38.1% |
| 36 | E1E115023 | Arlin | Pendeteksi Tingkat Kekeruhan Air Menggunakan Sensor Turbidity Dan Sensor Ph Berbasis Android Secara Real Time | 37.9% |
| 22 | E1E116005 | Ayu Windiarti | Implementasi Metode Blum Blum Shub (BBS) Untuk Pengacakan Soal Kuis Pada Aplikasi Media Pembelajaran Ipa Tingkat Sekolah Dasar Kelas 6 Berbasis Mobile | 37.6% |
| 37 | E1E115001 | Adnan Hidayat | Penerapan Algoritma A\* Pathfinding Dalam Pembentukan Artificial Intelligence Enemy Pada Game Ghost Adventure 3d (Studi Kasus : Pencarian Fakultas Uho) | 37.5% |
| 8 | E1E116074 | Abdul Azis Syah Laode | Implementasi Metode Viola-Jones Dan Eigenfaces Pada Sistem Pengenalan Wajah Secara Realtime Menggunakan Circuit Closed Television (CCTV) | 36% |
| 32 | E1E115036 | Nina Sularida | Implementasi Metode Multi Support Vector Machine Untuk Menentukan Tingkat Kematangan Buah Pisang Raja Dan Pisang Kepok | 35.9% |

Tabel 5. 7 uji data 5 E1E117035

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nim Mahasiswa** | **Nama Mahasiswa** | **Judul Skripsi** | **Kemiripan (%)** |
| 13 | E1E116049 | Alfrido Rahmat Julianto Pidani | Pengimplementasian Internet Of Things (Iot) Dalam Monitoring Kwh Meter Listrik Berbasis Mobile | 41.7% |
| 36 | E1E115023 | Arlin | Pendeteksi Tingkat Kekeruhan Air Menggunakan Sensor Turbidity Dan Sensor Ph Berbasis Android Secara Real Time | 41.3% |
| 22 | E1E116005 | Ayu Windiarti | Implementasi Metode Blum Blum Shub (Bbs) Untuk Pengacakan Soal Kuis Pada Aplikasi Media Pembelajaran Ipa Tingkat Sekolah Dasar Kelas 6 Berbasis Mobile | 39.6% |
| 33 | E1E115028 | Maulid | Implementasi Dan Analisis Access Point 5 Ghz Menggunakan Metode Manual Random Sampling Dan Converge Visualization | 39.5% |
| 3 | E1E117027 | Atri Ilma Juni Rahim | Sistem Library Video Pembelajaran Applied Bahavior Analisis (Aba) Untuk Anak Autis Berbasis Mobile Menggunakan Metode Knuth Morris Pratt (Kmp) (Studi Kasus : Sekolah Luar Biasa Aksara Center) | 39.4% |
| 40 | E1E1 15 014 | Elsa Meilan | Rancang Bangun Aplikasi Kanopi Otomatis Berbasis Android Menggunakan Logika Fuzzy Tsukamoto | 39.1% |
| 1 | E1E117040 | Muhamad Danil | Implementasi Global Navigation Satellite System (Gnss) Pada Sistem Presensi Berbasis Android Menggunakan Metode Spatial | 38.3% |
| 12 | E1E116051 | Reza Sanjaya | Aplikasi Perkiraan Curah Hujan Kota Kendari Menggunakan Metode Interval Type-2 Fuzzy Logic System | 38% |
| 17 | E1E116028 | Rifa’atus Shalihah | Penerapan Metode N-Gram Untuk Memperbaiki Kesalahan Penulisan Ejaan Kata Kunci (Spelling Correction) Pada Aplikasi Pencarian Hadis Riwayat Bukhari | 37.9% |
| 27 | E1E115093 | Yulina | Implementasi Intrusion Detection System (IDS) Dan Intrusion Prevention System (IPS) Menggunakan Jejaring Sosial Serta Email Sebagai Media Notifikasi | 37.8% |

Tabel 5. 8 Uji Data 6 E1E117049

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nim Mahasiswa** | **Nama Mahasiswa** | **Judul Skripsi** | **Kemiripan (%)** |
| 20 | E1E116014 | Mardianti Potto | Diagnosis Penyakit Diabetes Mellitus Dan Penentuan Pola Makan Menggunakan Metode Case Based Reasoning | 38.5% |
| 17 | E1E116028 | Rifa’atus Shalihah | Penerapan Metode N-Gram Untuk Memperbaiki Kesalahan Penulisan Ejaan Kata Kunci (Spelling Correction) Pada Aplikasi Pencarian Hadis Riwayat Bukhari | 37.8% |
| 34 | E1E115026 | Mamta Culkari. P | Perbandingan Agoritma Horspool Dan Algoritma Raita Pada Aplikasi Kamus Istilah Psikologi Berbasis Android | 37.8% |
| 15 | E1E116034 | Saskia Randawula Silondae | Penerapan Algoritma K-Medoids Dalam Penentuan Faktor Terbesar Sumber Informasi Pemilihan Jurusan Di Fakultas Teknik Universitas Halu Oleo | 37.5% |
| 10 | E1E116061 | Sonia Lamani | Penerapan Algoritma Hash Based Pada Transaksi Penjualan Menggunakan Aturan Asosiasi Untuk Mengatur Penempatan Barang Di Minimarket (Studi Kasus: Fc Akbar.Com Kendari) | 37.4% |
| 1 | E1E117040 | Muhamad Danil | Implementasi Global Navigation Satellite System (Gnss) Pada Sistem Presensi Berbasis Android Menggunakan Metode Spatial | 37.3% |
| 3 | E1E117027 | Atri Ilma Juni Rahim | Sistem Library Video Pembelajaran Applied Bahavior Analisis (Aba) Untuk Anak Autis Berbasis Mobile Menggunakan Metode Knuth Morris Pratt (Kmp) (Studi Kasus : Sekolah Luar Biasa Aksara Center) | 36.9% |
| 22 | E1E116005 | Ayu Windiarti | Implementasi Metode Blum Blum Shub (Bbs) Untuk Pengacakan Soal Kuis Pada Aplikasi Media Pembelajaran Ipa Tingkat Sekolah Dasar Kelas 6 Berbasis Mobile | 36.6% |
| 30 | E1E115054 | Ahmad Khairun Arsyad | Implementasi Levenshtein Distance Pada Aplikasi Pencarian Barang Di Berbagai E-Marketplace Menggunakan Teknik Web Scraping | 36.2% |
| 36 | E1E115023 | Arlin | Pendeteksi Tingkat Kekeruhan Air Menggunakan Sensor Turbidity Dan Sensor Ph Berbasis Android Secara Real Time | 36% |

Tabel 5. 9 uji data 7 E1E117018

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nim Mahasiswa** | **Nama Mahasiswa** | **Judul Skripsi** | **Kemiripan (%)** |
| 1 | E1E117040 | Muhamad Danil | Implementasi Global Navigation Satellite System (Gnss) Pada Sistem Presensi Berbasis Android Menggunakan Metode Spatial | 41.3% |
| 15 | E1E116034 | Saskia Randawula Silondae | Penerapan Algoritma K-Medoids Dalam Penentuan Faktor Terbesar Sumber Informasi Pemilihan Jurusan Di Fakultas Teknik Universitas Halu Oleo | 38.4% |
| 2 | E1E117030 | Fitnawati | Implementasi Algoritma Levenshtein Distance Pada Sistem Informasi Penelusuran Tugas Akhir Berbasis Web | 38.2% |
| 35 | E1E115025 | La Ode Muhamad Taufiq | Analisis Perbandingan Algoritma Penjadwalan Round Robin Dan Shortest Job Fisrt Untuk Manajemen Proses Dalam Single Processing | 38% |
| 37 | E1E115001 | Adnan Hidayat | Penerapan Algoritma A\* Pathfinding Dalam Pembentukan Artificial Intelligence Enemy Pada Game Ghost Adventure 3d (Studi Kasus : Pencarian Fakultas Uho) | 38% |
| 8 | E1E116074 | Abdul Azis Syah Laode | Implementasi Metode Viola-Jones Dan Eigenfaces Pada Sistem Pengenalan Wajah Secara Realtime Menggunakan Circuit Closed Television (Cctv) | 37.9% |
| 10 | E1E116061 | Sonia Lamani | Penerapan Algoritma Hash Based Pada Transaksi Penjualan Menggunakan Aturan Asosiasi Untuk Mengatur Penempatan Barang Di Minimarket (Studi Kasus: Fc Akbar.Com Kendari) | 37.9% |
| 41 | E1E116073 | R.M.Alfian Apriansya Diponegoro | Implementasi String Matching Algoritma Raita Untuk Pencarian Data Dalam Pembuatan Aplikasi Manajemen Arsip Pengalaman Bimbingan Lapangan (Pbl) | 37.9% |
| 22 | E1E116005 | Ayu Windiarti | Implementasi Metode Blum Blum Shub (BBS) Untuk Pengacakan Soal Kuis Pada Aplikasi Media Pembelajaran Ipa Tingkat Sekolah Dasar Kelas 6 Berbasis Mobile | 37.6% |
| 27 | E1E115093 | Yulina | Implementasi Intrusion Detection System (IDS) Dan Intrusion Prevention System (IPS) Menggunakan Jejaring Sosial Serta Email Sebagai Media Notifikasi | 37.1% |

Tabel 5. 10 uji data 8 E1E117016

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nim Mahasiswa** | **Nama Mahasiswa** | **Judul Skripsi** | **Kemiripan (%)** |
| 35 | E1E115025 | La Ode Muhamad Taufiq | Analisis Perbandingan Algoritma Penjadwalan Round Robin Dan Shortest Job Fisrt Untuk Manajemen Proses Dalam Single Processing | 45% |
| 10 | E1E116061 | Sonia Lamani | Penerapan Algoritma Hash Based Pada Transaksi Penjualan Menggunakan Aturan Asosiasi Untuk Mengatur Penempatan Barang Di Minimarket (Studi Kasus: Fc Akbar.Com Kendari) | 43.8% |
| 41 | E1E116073 | R.M.Alfian Apriansya Diponegoro | Implementasi String Matching Algoritma Raita Untuk Pencarian Data Dalam Pembuatan Aplikasi Manajemen Arsip Pengalaman Bimbingan Lapangan (Pbl) | 43.3% |
| 15 | E1E116034 | Saskia Randawula Silondae | Penerapan Algoritma K-Medoids Dalam Penentuan Faktor Terbesar Sumber Informasi Pemilihan Jurusan Di Fakultas Teknik Universitas Halu Oleo | 43.1% |
| 24 | E1E115100 | Adin Setiawan | Sistem Pendukung Keputusan Penilaian Dan Pemberian Bonus Salesman Pada Pt. Matakar Kendari Dengan Menggunakan Metode Profile Matching | 41.9% |
| 8 | E1E116074 | Abdul Azis Syah Laode | Implementasi Metode Viola-Jones Dan Eigenfaces Pada Sistem Pengenalan Wajah Secara Realtime Menggunakan Circuit Closed Television (Cctv) | 41.2% |
| 17 | E1E116028 | Rifa’atus Shalihah | Penerapan Metode N-Gram Untuk Memperbaiki Kesalahan Penulisan Ejaan Kata Kunci (Spelling Correction) Pada Aplikasi Pencarian Hadis Riwayat Bukhari | 40.9% |
| 2 | E1E117030 | Fitnawati | Implementasi Algoritma Levenshtein Distance Pada Sistem Informasi Penelusuran Tugas Akhir Berbasis Web | 40.7% |
| 29 | E1E115075 | Muh. Saiful | Implementasi Metode Rocchio Untuk Klasifikasi Jenis Pelaporan Di Laman Propam Polda Sultra | 40.5% |
| 13 | E1E116049 | Alfrido Rahmat Julianto Pidani | Pengimplementasian Internet Of Things (IOT) Dalam Monitoring Kwh Meter Listrik Berbasis Mobile | 40% |

Tabel 5. 11 uji data 9 E1E117054

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nim Mahasiswa** | **Nama Mahasiswa** | **Judul Skripsi** | **Kemiripan (%)** |
| 35 | E1E115025 | La Ode Muhamad Taufiq | Analisis Perbandingan Algoritma Penjadwalan Round Robin Dan Shortest Job Fisrt Untuk Manajemen Proses Dalam Single Processing | 40.7% |
| 19 | E1E116021 | Nurvila | Analisis Quality Of Service (Qos) Jaringan Nirkabel Pada Permainan Game Online Menggunakan Metode Peer Connection Queue (Pcq) Dengan Antrian Queue Tree | 37.8% |
| 12 | E1E116051 | Reza Sanjaya | Aplikasi Perkiraan Curah Hujan Kota Kendari Menggunakan Metode Interval Type-2 Fuzzy Logic System | 40.3% |
| 4 | E1E117023 | Nur Aziza Tadjuddin | Data Uji 4 Kata | 1.4% |
| 20 | E1E116014 | Mardianti Potto | Diagnosis Penyakit Diabetes Mellitus Dan Penentuan Pola Makan Menggunakan Metode Case Based Reasoning | 38.1% |
| 18 | E1E116023 | Pratiwi Nur Aisyah | Game Edukasi Berbasis Android Untuk Anak Usia Dini Menggunakan Linear Congruent Method (Lc | 37.3% |
| 9 | E1E116066 | Asrif Fajar Hidayat | Identifikasi Landmark Sella, Nasion Dan Menton Cephalometry Menggunakan Metode Convolutional Neural Networks (Cnn) | 33.7% |
| 2 | E1E117030 | Fitnawati | Implementasi Algoritma Levenshtein Distance Pada Sistem Informasi Penelusuran Tugas Akhir Berbasis Web | 43.9% |
| 28 | E1E115076 | Muh. Darul Zulkifli Ritom | Implementasi Algoritma Levenshtein Distance Pada Sistem Informasi Penelusuran Tugas Akhir Berbasis Web | 39.3% |
| 6 | E1E117005 | Annisa Fitri Hamid | Implementasi Algoritma Rc4 Pada Penggunaan Qr-Code Aplikasi Permohonan Pengajuan Hak Atas Tanah Milik Instansi Pemerintah Di Kabupaten Konawe Selatan | 37.6% |

Tabel 5. 12 uji data 10 E1E117065

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nim Mahasiswa** | **Nama Mahasiswa** | **Judul Skripsi** | **Kemiripan (%)** |
| 37 | E1E115001 | Adnan Hidayat | Penerapan Algoritma A\* Pathfinding Dalam Pembentukan Artificial Intelligence Enemy Pada Game Ghost Adventure 3d (Studi Kasus : Pencarian Fakultas Uho) | 43% |
| 13 | E1E116049 | Alfrido Rahmat Julianto Pidani | Pengimplementasian Internet Of Things (Iot) Dalam Monitoring Kwh Meter Listrik Berbasis Mobile | 41.4% |
| 10 | E1E116061 | Sonia Lamani | Penerapan Algoritma Hash Based Pada Transaksi Penjualan Menggunakan Aturan Asosiasi Untuk Mengatur Penempatan Barang Di Minimarket (Studi Kasus: Fc Akbar.Com Kendari) | 41.2% |
| 1 | E1E117040 | Muhamad Danil | Implementasi Global Navigation Satellite System (Gnss) Pada Sistem Presensi Berbasis Android Menggunakan Metode Spatial | 39.4% |
| 30 | E1E115054 | Ahmad Khairun Arsyad | Implementasi Levenshtein Distance Pada Aplikasi Pencarian Barang Di Berbagai E-Marketplace Menggunakan Teknik Web Scraping | 39.3% |
| 35 | E1E115025 | La Ode Muhamad Taufiq | Analisis Perbandingan Algoritma Penjadwalan Round Robin Dan Shortest Job Fisrt Untuk Manajemen Proses Dalam Single Processing | 39.3% |
| 20 | E1E116014 | Mardianti Potto | Diagnosis Penyakit Diabetes Mellitus Dan Penentuan Pola Makan Menggunakan Metode Case Based Reasoning | 38.8% |
| 3 | E1E117027 | Atri Ilma Juni Rahim | Sistem Library Video Pembelajaran Applied Bahavior Analisis (Aba) Untuk Anak Autis Berbasis Mobile Menggunakan Metode Knuth Morris Pratt (Kmp) (Studi Kasus : Sekolah Luar Biasa Aksara Center) | 38.7% |
| 12 | E1E116051 | Reza Sanjaya | Aplikasi Perkiraan Curah Hujan Kota Kendari Menggunakan Metode Interval Type-2 Fuzzy Logic System | 37.4% |
| 40 | E1E115014 | Elsa Meilan | Rancang Bangun Aplikasi Kanopi Otomatis Berbasis Android Menggunakan Logika Fuzzy Tsukamoto | 37.4% |

Peminatan Jarimgan

Tabel 5. 13 uji data 11 E1E117056

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nim Mahasiswa** | **Nama Mahasiswa** | **Judul Skripsi** | **Kemiripan (%)** |
| 13 | E1E116049 | Alfrido Rahmat Julianto Pidani | Pengimplementasian Internet Of Things (Iot) Dalam Monitoring Kwh Meter Listrik Berbasis Mobile | 38.8% |
| 8 | E1E116074 | Abdul Azis Syah Laode | Implementasi Metode Viola-Jones Dan Eigenfaces Pada Sistem Pengenalan Wajah Secara Realtime Menggunakan Circuit Closed Television (Cctv) | 37.8% |
| 39 | E1E116004 | Ayu Asriani | Penerapan Metode Autoregressive Integrated Moving Average (Arima) Pada Indeks Harga Konsumen (Ihk) Dalam Inflation Forecasting System (Studi Kasus Badan Pusat Statistik Kota Kendari) | 36.8% |
| 36 | E1E115023 | Arlin | Pendeteksi Tingkat Kekeruhan Air Menggunakan Sensor Turbidity Dan Sensor Ph Berbasis Android Secara Real Time | 36.4% |
| 11 | E1E116060 | Muhamad Fadli | Implementasi Internet Of Things (Iot) Sistem Keamanan Anti Maling Kendaraan Sepeda Motor | 36% |
| 33 | E1E115028 | Maulid | Implementasi Dan Analisis Access Point 5 Ghz Menggunakan Metode Manual Random Sampling Dan Converge Visualization | 35% |
| 15 | E1E116034 | Saskia Randawula Silondae | Penerapan Algoritma K-Medoids Dalam Penentuan Faktor Terbesar Sumber Informasi Pemilihan Jurusan Di Fakultas Teknik Universitas Halu Oleo | 34.8% |
| 17 | E1E116028 | Rifa’atus Shalihah | Penerapan Metode N-Gram Untuk Memperbaiki Kesalahan Penulisan Ejaan Kata Kunci (Spelling Correction) Pada Aplikasi Pencarian Hadis Riwayat Bukhari | 34.8% |
| 30 | E1E115054 | Ahmad Khairun Arsyad | Implementasi Levenshtein Distance Pada Aplikasi Pencarian Barang Di Berbagai E-Marketplace Menggunakan Teknik Web Scraping | 34.7% |
| 42 | E1E115034 | Muhammad Budi Dharmawan P | Implementasi Latent Semantic Analiysis Method Untuk Mendeteksi Kemiripan Kalimat | 34.5% |

Tabel 5. 14 uji data 12 E1E117039

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nim Mahasiswa** | **Nama Mahasiswa** | **Judul Skripsi** | **Kemiripan (%)** |
| 11 | E1E116060 | Muhamad Fadli | Implementasi Internet Of Things (IOT) Sistem Keamanan Anti Maling Kendaraan Sepeda Motor | 51.6% |
| 31 | E1E115037 | Nur Hazmila | Implementasi Algoritma Sha-512 Dan Vigenere Cipher Untuk Pengamanan File Dari Serangan Pembajakan (Man-In-The-Middle Dan Ssh Brute Force Attack) | 46% |
| 3 | E1E117027 | Atri Ilma Juni Rahim | Sistem Library Video Pembelajaran Applied Bahavior Analisis (ABA) Untuk Anak Autis Berbasis Mobile Menggunakan Metode Knuth Morris Pratt (KMP) (Studi Kasus : Sekolah Luar Biasa Aksara Center) | 44.1% |
| 28 | E1E115076 | Muh. Darul Zulkifli Ritom | Implementasi Algoritma Levenshtein Distance Pada Sistem Informasi Penelusuran Tugas Akhir Berbasis Web | 43.5% |
| 30 | E1E115054 | Ahmad Khairun Arsyad | Implementasi Levenshtein Distance Pada Aplikasi Pencarian Barang Di Berbagai E-Marketplace Menggunakan Teknik Web Scraping | 43.5% |
| 29 | E1E115075 | Muh. Saiful | Implementasi Metode Rocchio Untuk Klasifikasi Jenis Pelaporan Di Laman Propam Polda Sultra | 43.3% |
| 8 | E1E116074 | Abdul Azis Syah Laode | Implementasi Metode Viola-Jones Dan Eigenfaces Pada Sistem Pengenalan Wajah Secara Realtime Menggunakan Circuit Closed Television (Cctv) | 43.1% |
| 6 | E1E117005 | Annisa Fitri Hamid | Implementasi Algoritma Rc4 Pada Penggunaan Qr-Code Aplikasi Permohonan Pengajuan Hak Atas Tanah Milik Instansi Pemerintah Di Kabupaten Konawe Selatan | 41.9% |
| 33 | E1E115028 | Maulid | Implementasi Dan Analisis Access Point 5 Ghz Menggunakan Metode Manual Random Sampling Dan Converge Visualization | 41.9% |
| 36 | E1E115023 | Arlin | Pendeteksi Tingkat Kekeruhan Air Menggunakan Sensor Turbidity Dan Sensor Ph Berbasis Android Secara Real Time | 41.3% |

Tabel 5. 15 uji data 13 E1E117013

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nim Mahasiswa** | **Nama Mahasiswa** | **Judul Skripsi** | **Kemiripan (%)** |
| 36 | E1E115023 | Arlin | Pendeteksi Tingkat Kekeruhan Air Menggunakan Sensor Turbidity Dan Sensor Ph Berbasis Android Secara Real Time | 45.1% |
| 13 | E1E116049 | Alfrido Rahmat Julianto Pidani | Pengimplementasian Internet Of Things (Iot) Dalam Monitoring Kwh Meter Listrik Berbasis Mobile | 45% |
| 40 | E1E1 15 014 | Elsa Meilan | Rancang Bangun Aplikasi Kanopi Otomatis Berbasis Android Menggunakan Logika Fuzzy Tsukamoto | 43.8% |
| 1 | E1E117040 | Muhamad Danil | Implementasi Global Navigation Satellite System (Gnss) Pada Sistem Presensi Berbasis Android Menggunakan Metode Spatial | 42.5% |
| 17 | E1E116028 | Rifa’atus Shalihah | Penerapan Metode N-Gram Untuk Memperbaiki Kesalahan Penulisan Ejaan Kata Kunci (Spelling Correction) Pada Aplikasi Pencarian Hadis Riwayat Bukhari | 42% |
| 10 | E1E116061 | Sonia Lamani | Penerapan Algoritma Hash Based Pada Transaksi Penjualan Menggunakan Aturan Asosiasi Untuk Mengatur Penempatan Barang Di Minimarket (Studi Kasus: Fc Akbar.Com Kendari) | 41.8% |
| 25 | E1E115096 | Muhammad Abdi Fahmy | Sistem Pakar Diagnosis Penyakit Hewan Sapi Dengan Metode Forward Chainin | 41.7% |
| 33 | E1E115028 | Maulid | Implementasi Dan Analisis Access Point 5 Ghz Menggunakan Metode Manual Random Sampling Dan Converge Visualization | 41.3% |
| 8 | E1E116074 | Abdul Azis Syah Laode | Implementasi Metode Viola-Jones Dan Eigenfaces Pada Sistem Pengenalan Wajah Secara Realtime Menggunakan Circuit Closed Television (CCTV) | 40.8% |
| 22 | E1E116005 | Ayu Windiarti | Implementasi Metode Blum Blum Shub (BBS) Untuk Pengacakan Soal Kuis Pada Aplikasi Media Pembelajaran Ipa Tingkat Sekolah Dasar Kelas 6 Berbasis Mobile | 40.5% |

Peminatan Kecerdasan Buatan

Tabel 5. 16 Tabel uji data 14 E1E117064

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nim Mahasiswa** | **Nama Mahasiswa** | **Judul Skripsi** | **Kemiripan (%)** |
| 32 | E1E115036 | Nina Sularida | Implementasi Metode Multi Support Vector Machine Untuk Menentukan Tingkat Kematangan Buah Pisang Raja Dan Pisang Kepok | 47.5% |
| 33 | E1E115028 | Maulid | Implementasi Dan Analisis Access Point 5 Ghz Menggunakan Metode Manual Random Sampling Dan Converge Visualization | 46.4% |
| 19 | E1E116021 | Nurvila | Analisis Quality Of Service (Qos) Jaringan Nirkabel Pada Permainan Game Online Menggunakan Metode Peer Connection Queue (Pcq) Dengan Antrian Queue Tree | 44.1% |
| 8 | E1E116074 | Abdul Azis Syah Laode | Implementasi Metode Viola-Jones Dan Eigenfaces Pada Sistem Pengenalan Wajah Secara Realtime Menggunakan Circuit Closed Television (Cctv) | 44% |
| 17 | E1E116028 | Rifa’atus Shalihah | Penerapan Metode N-Gram Untuk Memperbaiki Kesalahan Penulisan Ejaan Kata Kunci (Spelling Correction) Pada Aplikasi Pencarian Hadis Riwayat Bukhari | 43.9% |
| 28 | E1E115076 | Muh. Darul Zulkifli Ritom | Implementasi Algoritma Levenshtein Distance Pada Sistem Informasi Penelusuran Tugas Akhir Berbasis Web | 43.6% |
| 12 | E1E116051 | Reza Sanjaya | Aplikasi Perkiraan Curah Hujan Kota Kendari Menggunakan Metode Interval Type-2 Fuzzy Logic System | 43.4% |
| 35 | E1E115025 | La Ode Muhamad Taufiq | Analisis Perbandingan Algoritma Penjadwalan Round Robin Dan Shortest Job Fisrt Untuk Manajemen Proses Dalam Single Processing | 43.3% |
| 3 | E1E117027 | Atri Ilma Juni Rahim | Sistem Library Video Pembelajaran Applied Bahavior Analisis (Aba) Untuk Anak Autis Berbasis Mobile Menggunakan Metode Knuth Morris Pratt (Kmp) (Studi Kasus : Sekolah Luar Biasa Aksara Center) | 43.2% |
| 40 | E1E115014 | Elsa Meilan | Rancang Bangun Aplikasi Kanopi Otomatis Berbasis Android Menggunakan Logika Fuzzy Tsukamoto | 43.1% |

Tabel 5. 17 uji data 15 E1E117022

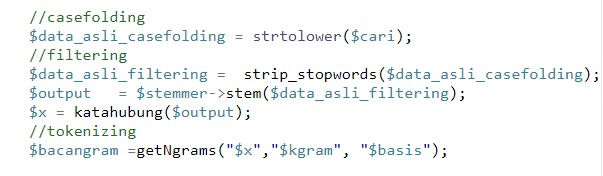
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nim Mahasiswa** | **Nama Mahasiswa** | **Judul Skripsi** | **Kemiripan (%)** |
| 8 | E1E116074 | Abdul Azis Syah Laode | Implementasi Metode Viola-Jones Dan Eigenfaces Pada Sistem Pengenalan Wajah Secara Realtime Menggunakan Circuit Closed Television (Cctv) | 38.2% |
| 32 | E1E115036 | Nina Sularida | Implementasi Metode Multi Support Vector Machine Untuk Menentukan Tingkat Kematangan Buah Pisang Raja Dan Pisang Kepok | 33.6% |
| 9 | E1E116066 | Asrif Fajar Hidayat | Identifikasi Landmark Sella, Nasion Dan Menton Cephalometry Menggunakan Metode Convolutional Neural Networks (Cnn) | 31.7% |
| 33 | E1E115028 | Maulid | Implementasi Dan Analisis Access Point 5 Ghz Menggunakan Metode Manual Random Sampling Dan Converge Visualization | 30.9% |
| 24 | E1E115100 | Adin Setiawan | Sistem Pendukung Keputusan Penilaian Dan Pemberian Bonus Salesman Pada Pt. Matakar Kendari Dengan Menggunakan Metode Profile Matching | 30.1% |
| 28 | E1E115076 | Muh. Darul Zulkifli Ritom | Implementasi Algoritma Levenshtein Distance Pada Sistem Informasi Penelusuran Tugas Akhir Berbasis Web | 29.3% |
| 10 | E1E116061 | Sonia Lamani | Penerapan Algoritma Hash Based Pada Transaksi Penjualan Menggunakan Aturan Asosiasi Untuk Mengatur Penempatan Barang Di Minimarket (Studi Kasus: Fc Akbar.Com Kendari) | 29.1% |
| 17 | E1E116028 | Rifa’atus Shalihah | Penerapan Metode N-Gram Untuk Memperbaiki Kesalahan Penulisan Ejaan Kata Kunci (Spelling Correction) Pada Aplikasi Pencarian Hadis Riwayat Bukhari | 29.1% |
| 38 | E1E1 16 084 | Wa Ode Wahyuni Makmun | Penerapan Vector Space Model (Vsm) Pada Sistem Pencarian Artikel Arkeologi (Studi Kasus: Jurusan Arkeologi Universitas Halu Oleo) | 29% |
| 34 | E1E115026 | Mamta Culkari. P | Perbandingan Agoritma Horspool Dan Algoritma Raita Pada Aplikasi Kamus Istilah Psikologi Berbasis Android | 28.9% |

Tabel 5. 18 Tabel uji data 15 E1E117021

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nim Mahasiswa** | **Nama Mahasiswa** | **Judul Skripsi** | **Kemiripan (%)** |
| 8 | E1E116074 | Abdul Azis Syah Laode | Implementasi Metode Viola-Jones Dan Eigenfaces Pada Sistem Pengenalan Wajah Secara Realtime Menggunakan Circuit Closed Television (Cctv) | 45.5% |
| 32 | E1E115036 | Nina Sularida | Implementasi Metode Multi Support Vector Machine Untuk Menentukan Tingkat Kematangan Buah Pisang Raja Dan Pisang Kepok | 45.2% |
| 17 | E1E116028 | Rifa’atus Shalihah | Penerapan Metode N-Gram Untuk Memperbaiki Kesalahan Penulisan Ejaan Kata Kunci (Spelling Correction) Pada Aplikasi Pencarian Hadis Riwayat Bukhari | 45.1% |
| 38 | E1E1 16 084 | Wa Ode Wahyuni Makmun | Penerapan Vector Space Model (Vsm) Pada Sistem Pencarian Artikel Arkeologi (Studi Kasus: Jurusan Arkeologi Universitas Halu Oleo) | 44.9% |
| 29 | E1E115075 | Muh. Saiful | Implementasi Metode Rocchio Untuk Klasifikasi Jenis Pelaporan Di Laman Propam Polda Sultra | 44.4% |
| 22 | E1E116005 | Ayu Windiarti | Implementasi Metode Blum Blum Shub (BBS) Untuk Pengacakan Soal Kuis Pada Aplikasi Media Pembelajaran Ipa Tingkat Sekolah Dasar Kelas 6 Berbasis Mobile | 42.8% |
| 39 | E1E116004 | Ayu Asriani | Penerapan Metode Autoregressive Integrated Moving Average (Arima) Pada Indeks Harga Konsumen (IHK) Dalam Inflation Forecasting System (Studi Kasus Badan Pusat Statistik Kota Kendari) | 42.8% |
| 40 | E1E115014 | Elsa Meilan | Rancang Bangun Aplikasi Kanopi Otomatis Berbasis Android Menggunakan Logika Fuzzy Tsukamoto | 42.7% |
| 37 | E1E115001 | Adnan Hidayat | Penerapan Algoritma A\* Pathfinding Dalam Pembentukan Artificial Intelligence Enemy Pada Game Ghost Adventure 3d (Studi Kasus : Pencarian Fakultas UHO) | 42.3% |
| 10 | E1E116061 | Sonia Lamani | Penerapan Algoritma Hash Based Pada Transaksi Penjualan Menggunakan Aturan Asosiasi Untuk Mengatur Penempatan Barang Di Minimarket (Studi Kasus: Fc Akbar.Com Kendari) | 42.1% |

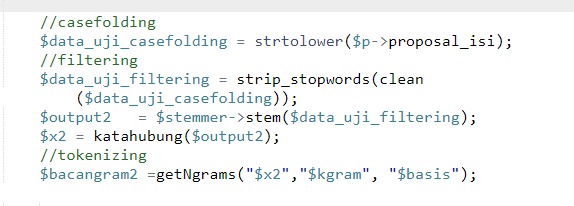
Untuk Proposal mahasiswa pada peminatan Rekayasa Perangkat Lunak memiliki kemiripan tingkat plagiarisme Sedang dengan skripsi mahasiswa yang telah lulus, hal itu di sebabkan karena skripsi mahasiswa peminatan Rekayasa Perangkat Lunak lebih banyak yang di masukkan sebagai data latih.

* + 1. **Implementasi source code Algoritma Rabin karp**
       1. Source Code Algoritma *Rabin Karp* untuk data asli.

****

Gambar 5.8 Algoritma *Rabin Karp* untuk data asli

* + - 1. Source Code Algoritma *Rabin Karp* untuk data uji

****

Gambar 5.9 Algoritma *Rabin Karp* untuk data uji

Pada gambar 5.9 merupakan salah satu implementasi dari *class diagram* ke bentuk baris kode. Berikut merupakan baris kode dari *class* Plagiarisme.

* 1. **Pengujian *black box***

Pengujian *black box* dilakukan untuk mengamati hasil eksekusi dan memeriksa fungsional dari perangkat lunak, serta mengevaluasi berdasarkan tampilan dari aplikasi, tanpa mengetahui proses detail yang terjadi. Fungsionalitas perangkat lunak yang diuji sesuai dengan *use Case* pada tahap perancangan. Pada pengujian blac box sistem ini semua menu ada di sistem ini berhasil di gunakan.

Tabel 5.19 Pengujian *Black box*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Nama *Form* | Detail Uji | *Output* | Hasil Uji |
| 1. | Pengujian *Login* | Menu *Login* | Menampilkan Menu *Login User* | Sesuai |
| 2. | Pengujian Menu Mahasiswa | Menu tambah mahasiswa | Menampilkan Halaman tambah mahasiswa | Sesuai |
|  |  | Mahasiswa  (*Edit* Mahasiswa) | Menampilkan Halaman edit mahasiswa | Sesuai |
|  |  | Mahasiswa (Hapus) | Menampilkan Halaman Hapus mahasiswa | Sesuai |
| 3. | Pengujian Menu Dosen | Dosen  (Tambah Dosen) | Menampilkan Menu Tambah Data Mahasiswa | Sesuai |
| Mahasiswa  (Edit Dosen) | Menampilkan Menu Edit Data Mahasiswa | Sesuai |
| Dosen (Hapus) | Menampilkan Menu Hapus Data Mahasiswa | Sesuai |
| 4. | Pengujian Menu Proposal | Proposal (Tambah data) | Menampilkan Data Proposal | Sesuai |
| Proposal  (Edit Proposal) | Menampilkan Menu Edit Data Proposal | Sesuai |
|  |  | Proposal  (Hapus Proposal) | Menampilkan Menu Hapus Data Proposal | Sesuai |
| 5. | Pengujian Menu Cek Plagiarisme | Cek plagiarisme | Menampilkan Hasil cek plagiarisme data Proposal | Sesuai |

# BAB VI PENUTUP

* 1. **Kesimpulan**

Berdasarkan hasil penelitian yang dilakukan mengenai sistem pendeteksi plagiarisme pada proposal tugas akhir mahasiswa dengan algoritma *rabin karp* diperoleh beberapa kesimpulan sebagai berikut :

Pada penelitian ini melakukan pengujian akurasi berdasarkan tingkatan plagiarisme yaitu plagiarisme berat atau tidak berat .

Berdasarkan hasil evaluasi uji coba sistem dapat diketahui bahwa performa hasil persentase similaritas dari algoritma *Rabin karp* memiliki ketergantungan dengan nilai *k-gram* yang diberikan.

Sistem ini dapat menampilkan persentase kemiripan Latar belakang proposal antar mahasiswa dengan menggunakan k-gram 4 dan basis 11 berdasarkan beberapa kali uji coba yang di lakukan pada sistem sehingga Algoritma *Rabin-Karp* berhasil diimplementasikan pada sistem pendeteksi plagarisme .

* 1. **Saran**

Beberapa saran yang perlu diperhatikan pada penelitian selanjutnya adalah sebagai berikut.

Aplikasi ini diharapkan dapat membantu untuk mencegah secara dini kemungkinan terjadinya kegiatan plagiarisme dalam pengerjaan tugas akhir mahasiswa.

Perlu adanya penelitian lebih lanjut yaitu Algoritma yang digunakan pada aplikasi ini dapat dikembangkan dengan algoritma atau metode lain.

# DAFTAR PUSTAKA

Adi, S. (2018) ‘Penerapan Algoritma Rabin Karp Untuk Mendeteksi Kemiripan Judul Skripsi’, *Jurnal Mantik Penuasa*, 22(1), pp. 125–130.

Aziz, Lulu A., Ana I., & A. T. K. (2015) ‘Upaya Perpustakaan Dalam Mengurangi Plagiarisme Pada Karya Ilmiah Mahasiswa (Studi Kasus Di Upt Perpustakaan Unika Soegijapranata)’, *Jurnal Ilmu Perpustakaan*, 4(3), pp. 1–13. Available at: https://www.neliti.com/id/publications/137458/upaya-perpustakaan-dalam-mengurangi-plagiarisme-pada-karya-ilmiah-mahasiswa-stud.

Edi, D. and Betshani, S. (2012) ‘Analisis Data dengan Menggunakan ERD dan Model Konseptual Data Warehouse’, *Jurnal Informatika*, 5(1), pp. 71–85.

Filcha, A. and Hayaty, M. (2019) ‘Implementasi Algoritma Rabin-Karp untuk Pendeteksi Plagiarisme pada Dokumen Tugas Mahasiswa’, *JUITA : Jurnal Informatika*, 7(1), p. 25. doi: 10.30595/juita.v7i1.4063.

Hasyim, M. and Nugrahanto, G. (2014) ‘Pelatihan Pembuatan Proposal Kegiatan Pada Remaja Dusun Ngepuh Lor, Desa Banyusidi, Pakis, Magelang, Jawa Tengah’, *Seri Pengabdian Masyarakat 2014*, 3(3), pp. 206–210.

Iii, K., Jl, U. A. D. and Soepomo, P. (2010) ‘Implementasi Algoritma Greedy Untuk Melakukan Graph Coloring: Studi Kasus Peta Propinsi Jawa Timur’, *Jurnal Informatika Ahmad Dahlan*, 4(2), p. 103610. doi: 10.26555/jifo.v4i2.a5275.

Kiswanto, C. F. (2016) ‘“Sistem Informasi Akademik Sub-Sistem : Utility Dan Epsbed”’.

Magriyanti, A. A. (2018) ‘Analisis Pengembangan Algoritma Porter Stemming Dalam Bahasa Indonesia’. doi: 10.31227/osf.io/7ge4v.

Pahlevi, O., Mulyani, A. and Khoir, M. (2018) ‘Sistem Informasi Inventori Barang Menggunakan Metode Object Oriented Di Pt. Livaza Teknologi Indonesia Jakarta’, *Jurnal PROSISKO*, 5(1).

Pratomo, A. H. and Suryotomo, A. P. (2020) ‘Implementasi Pengecekan Plagiarisme Proposal Tugas Akhir Mahasiswa Teknik Informatika Upn Veteran Yogyakarta’, *Seminar Nasional Informatika (SEMNASIF)*, 1(1), pp. 221–229. Available at: http://103.23.20.161/index.php/semnasif/article/view/4104.

Putra, N. P. and Sularno (2019) ‘Penerapan Algoritma Rabin-Karp Dengan Pendekatan Synonym Recognition Sebagai Antisipasi Plagiarisme Pada Penulisan Skripsi’, *Jurnal Teknologi Dan Sistem Informasi Bisnis*, 1(2), pp. 49–58.

Suryati, T., Wibisono, Y. and Wihardi, Y. (2018) ‘Aplikasi Deteksi Plagiarisme Dokumen Skripsi dengan Algoritma Rabin-Karp’, *JATIKOM: Jurnal Teori dan Aplikasi Ilmu Komputer*, 1(2), pp. 91–95.

Tantoni, A. and Zaen, M. T. A. (2018) ‘Implementasi Double Caesar Cipher Menggunakan Ascii’, *Jurnal Informatika dan Rekayasa Elektronik*, 1(2), p. 24. doi: 10.36595/jire.v1i2.56.

Uji Cahyono, D. (2018) ‘Aplikasi Deteksi Dini Plagiarisme Judul Tugas Akhir Mahasiswa Sekolah Tinggi Ilmu Kesehatan Yayasan Rs. Islam Surabaya Dengan Algoritma Rabin-Karp’, *Applied Technology and Computing Science Journal*, 1(1), pp. 1–10. doi: 10.33086/atcsj.v1i1.3.

Wulan Nafesa Septine, S.T., M. (2019) *DATABASE MANAGEMENT SYSTEM (DBMS) MARIA DB*, *06 Nov 2019*.