DETEKSI PLAGIARISME TUGAS AKHIR MAHASISWA DENGAN MENGGUNAKAN METODE COSINE SIMILIARITY

LAPORAN TUGAS AKHIR

Laporan ini Disusun untuk Memenuhi Salah Satu Syarat Memperoleh Gelar Sarjana Strata 1 (S1) Pada Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang

DISUSUN OLEH :

YUSTIAN DIKMA EKA PUTRA

NIM : 32601500994

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ISLAM SULTAN AGUNG SEMARANG

2022

FINAL PROJECT

Thesis Plagiarism Detection Using Cosine Similarity Method

Proposed to complete the requirement to obtain a bachelor's degree (S-1)
at Informatics Engineering Department of Industrial Technology Faculty Sultan Agung Islamic University

Arranged By:

YUSTIAN DIKMA EKA PUTRA

NIM : 32601500994

MAJORING OF INFORMATICS ENGINEERING
INDUSTRIAL TECHONLOGY FACULTY
SULTAN AGUNG ISLAMIC UNIVERSITY
SEMARANG

2022

LEMBAR PENGESAHAN PEMBIMBING

Laporan Tugas Akhir dengan judul **"DETEKSI PLAGIARISME TUGAS AKHIR MAHASISWA DENGAN MENGGUNAKAN METODE COSINE SIMILIARITY"** ini disusun oleh :

Nama : Yustian Dikma Eka Putra

NIM : 32601500994

Program Studi : Teknik Informatika

Telah disahkan oleh dosen pembimbing pada :

Hari : Senin

Tanggal : 03 Oktober 2022

Mengesahkan,

Pembimbing I	Pembimbing II
(Imam Much Ibnu Subroto, ST, M.Sc, Ph.D)	**(Sam Farisa Chaerul Haviana, ST, M.Kom)**
NIDN. 0613037301	NIDN. 0628028602

Mengetahui,

Ketua Program Studi Teknik Informatika
Fakultas Teknologi Industri
Universitas Islam Sultan Agung

Ir. Sri Mulyono, M.Eng
NIDN. 0626066601

LEMBAR PENGESAHAN PENGUJI

Laporan tugas akhir dengan judul **"Deteksi Plagiarisme Tugas Akhir Mahasiswa Dengan Menggunakan Metode CosineSimiliarity"** ini telah dipertahankan di depan dosen penguji Tugas Akhir pada :

Hari : Senin

Tanggal : 03 Oktober 2022

TIM PENGUJI

Anggota I

Dedy Kurniadi, ST, M.Kom.
NIDN. 0622058802

Anggota II

Asih Widi Harini, Ssi.MT
NIDN. 0617087002

Ketua Penguji

Bagus Satrio WP, S.Kom.M.Cs
NIDN. 210616051

SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan dibawah ini :

Nama : Yustian Dikma Eka Putra

NIM : 32601500994

Judul Tugas Akhir : DETEKSI PLAGIARISME TUGAS AKHIR MAHASISWA DENGAN MENGGUNAKAN METODE COSINE SIMILIARITY

Dengan bahwa ini saya menyatakan bahwa judul dan isi Tugas Akhir yang saya buat dalam rangka menyelesaikan Pendidikan Strata Satu (S1) Teknik Informatika tersebut adalah asli dan belum pernah diangkat, ditulis ataupun dipublikasikan oleh siapapun baik keseluruhan maupun sebagian, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka, dan apa bila di kemudian hari ternyata terbukti bahwa judul Tugas Akhir tersebut pernah diangkat, ditulis ataupun dipublikasikan, maka saya bersedia dikenakan sanksi akademis. Demikian surat pernyataan ini saya buat dengan sadar dan penuh tanggung jawab.

Semarang, 03 Oktober 2022

Yang Menyatakan,

Yustian Dikma Eka Putra

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH

Saya yang bertanda tangan dibawah ini :

Nama	: Yustian Dikma Eka Putra
NIM	: 32601500994
Program Studi	: Teknik Informatika
Fakultas	: Teknologi industri
Alamat Asal	: Krajan Banyubiru Rt.06 Rw.01

Dengan ini menyatakan Karya Ilmiah berupa Tugas akhir dengan Judul : **"DETEKSI PLAGIARISME TUGAS AKHIR MAHASISWA DENGAN MENGGUNAKAN METODE COSINE SIMILIARITY** Menyetujui menjadi hak milik Universitas Islam Sultan Agung serta memberikan Hak bebas Royalti Non-Eksklusif untuk disimpan, dialihmediakan, dikelola dan pangkalan data dan dipublikasikan diinternet dan media lain untuk kepentingan akademis selama tetap menyantumkan nama penulis sebagai pemilik hak cipta. Pernyataan ini saya buat dengan sungguh-sungguh. Apabila dikemudian hari terbukti ada pelanggaran Hak Cipta/Plagiarisme dalam karya ilmiah ini, maka segala bentuk tuntutan hukum yang timbul akan saya tanggung secara pribadi tanpa melibatkan Universitas Islam Sultan Agung.

Semarang, 03 Oktober 2022

Yang menyatakan,

Yustian Dikma Eka Putra

KATA PENGANTAR

Dengan mengucap syukur alhamdulillah atas kehadirat Allah SWT yang telah memberikan rahmat dan karunianya kepada penulis, sehingga dapat menyelesaikan Tugas Akhir dengan judul "Deteksi Plagiarisme Tugas Akhir Mahasiswa Dengan Menggunakan Metode Cosine Similiarity" ini untuk memenuhi salah satu syarat menyelesaikan studi serta dalam rangka memperoleh gelar sarjana (S-1) pada Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang.

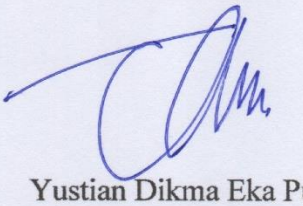
Tugas Akhir ini disusun dan dibuat dengan adanya bantuan dari berbagai pihak, materi maupun teknis, oleh karena itu saya selaku penulis mengucapkan terima kasih kepada :

1. Ibunda Ayahanda, dan adik saya tercinta yang telah banyak memberikan semangat, doa keselamatan dan keberhasilan selama menempuh ujian.
2. Bapak Imam Much Ibnu Subroto,ST, M.Sc, Ph.D Selaku Dosen Pembimbing I yang telah meluangkan waktu dan memberi ilmu kepada penulis.
3. Bapak Sam Farisa Chaerul Haviana, ST., M.Kom Selaku Dosen Pembimbing II yang telah meluangkan waktu dan memberi ilmu kepada penulis.
4. Para Dosen FTI Universitas Islam Sultan Agung yang telah memberikan banyak ilmu yang bermanfaat.

Dengan segala kerendahan hati, penulis menyadari masih banyak terdapat banyak kekurangan dari segi kualitas atau kuantitas maupun dari ilmu pengetahuan dalam penyusunan laporan, sehingga penulis mengharapkan adanya saran dan kritikan yang bersifat membangun demi kesempurnaan laporan ini dan masa mendatang.

Semarang, 03 Oktober 2022

Penulis



Yustian Dikma Eka Putra

5.

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

ABSTRAK

Tugas akhir (TA) atau tesis adalah sebuah mahakarya tertulis berupa tulisan-tulisan ilmiah yang mempresentasikan hasil penelitian guna membahas suatu masalah di dalam bidang keilmuan tertentu dengan menggunakan kaidah-kaidah penulisan yang berlaku di suatu bidang ilmu pengetahuan tertentu. Sementara dalam pembuatan tugas akhir sendiri sering ditemui tindak plagiarisme, tindakan ini sendiri bertujuan untuk mencuri hasil pikiran orang lain. Metode *cosine similiarity* digunakan untuk menghitung *similiarity* atau kemiripan dokumen dari tugas akhir dengan tujuan untuk menghitung seberapa besar tingkat *similiarity*-nya sehingga nantinya dapat digunakan sebagai salah satu cara mengidentifikasi apakah dokumen tersebut plagiat atau tidak. Pada perhitungan metode kosine semakin mendekati angka 1 maka dokumen tersebut bisa dianggap plagiat dan sebaliknya jika mendekati angka 0 maka dinyatakan sebaliknya.

Kata Kunci : Tugas akhir, Plagiarisme, *Cosine Similiarity*, Teks Preprocessing

Abstract

The final project or thesis is a written masterpiece in the form of scientific writings that present the results of research in order to discuss a problem in a particular scientific field by using the rules of writing that apply in a particular field of science. While in the making of the final project itself, plagiarism is often encountered, this action itself aims to steal the thoughts of others. The cosine similiarity method is used to calculate the similiarity or similarity of documents from the final project with the aim of calculating how much the similiarity level is so that later it can be used as a way to identify whether the document is plagiarized or not. In the calculation of the cosine method, the closer to 1, the document can be considered plagiarism and vice versa, if it is close to 0, the similiarity or similarity of the document is low.

Keywords: Final Project, Plagiarism, Cosine Similiarity, Text Preprocessing

BAB I

PENDAHULUAN

1.1.Latar Belakang

Tugas akhir (TA) atau tesis adalah karya tulis ilmiah berupa presentasi hasil penelitian yang membahas suatu masalah dalam bidang tertentu dengan menggunakan kaidah penulisan yang berlaku di bidang Ilmu Pengetahuan. Penelitian tugas akhir juga merupakan salah satu syarat kelulusan di setiap perguruan tinggi (S1), serta di Teknik Informatika UNISSULA, yang pelaksanaannya dilakukan selama satu semester, diambil pada tingkat semester terakhir.

Plagiarisme sendiri merupakan salah satu bentuk pelanggaran akademik dikarenakan mengandung unsur pencurian ide dan gagasan ilmiah tanpa mencantumkan sumbernya, dengan bertujuan untuk mengutip sebagian dari karya ilmiah maupun seluruh karya ilmiah orang lain dan diakuinya sebagai hasil pemikiran dan kerja keras orang tersebut, tanpa menyebutkan sumber dimana orang tersebut mendapatkan pengetahuan atau ilmu yang diambil secara jelas, akurat dan lengkap. Hal ini tentu saja sangat bertentangan dengan prinsip pendidikan yang bertujuan ingin menciptakan sumber daya manusia yang mempunyai akhlak yang mulia. Plagiarisme merusak moral dan martabat seorang penulis dan mempengaruhi integritas *civitas* akademik (Wibowo, 2012).

Plagiarisme di dalam pembuatan tugas akhir atau skripsi merupakan realitas yang sering terjadi di kalangan masyarakat terutama terjadi pada kalangan mahasiswa, bahkan ada sebuah kasus di mana seorang guru besar pun dapat terbukti melakukan tindakan plagiarisme, sehingga tidak mengherankan bahwasanya seseorang dapat meraih beberapa gelar sekaligus dalam waktu yang terlampau cepat, bukan hanya skripsi ataupun tesis, namun hal ini juga berlaku pada plagiarisme jawaban tugas sehari-hari yang dosen berikan kepada mahasiswa selama masa perkuliahan berlangsung, terjadinya praktik salin tempel sendiri merupakan hal lumrah yang sering dijalani oleh mahasiswa dalam

menjalani keseharian tugas semasa kehidupan perkuliahan mereka, yang dimana tindakan salin tempel tersebut sebenarnya merupakan salah satu bagian dari tindakan plagiarisme (Prihantini & Indudewi, 2017). Disisi lain plagiarisme berasal dari Bahasa latin *Plagiari(us)* atau *Plagi(um)* yang memiliki arti menculik, membajak atau juga bisa diartikan sebagai merampok. Definisi plagiarisme sendiri sangat banyak, sebagai salah satu contohnya menurut Kamus Besar Bahasa Indonesia (KBBI) dimana "Plagiat merupakan tindakan mengambil karangan (pendapat, dsb) orang lain lalu menyiarkannya sebagai karangan (pendapat, dsb) sendiri".

Dalam praktik plagiarisme sendiri memiliki banyak definisi tentang tindak perilaku plagiarisme, dimana tindakan tersebut berupa menjadikan buah hasil karya tulisan orang lain untuk dijadikan menjadi milik diri sendiri, menyalin kalimat-kalimat dan pola pikir tertentu yang bersumber dari buah pikir orang lain tanpa mencantumkan sumber dari orang tersebut, memberikan informasi dan pengetahuan yang salah mengenai sumber kutipan ilmu pengetahuan yang diambil, mengubah kata, kalimat dan paragraf namun tetap mempertahankan struktur kalimat dari suatu sumber tanpa memberikan *credit* sumber terkait (Maurer dkk., 2006).

Dalam beberapa dekade terakhir plagiarisme telah diklasifikasikan sebagai fenomena ketidakjujuran berlapis yang terjadi pada dunia pendidikan, sejumlah makalah penelitian telah mengidentifikasi faktor-faktor seperti jenis kelamin, peningkatan efisiensi, motivasi belajar, atau akses mudah media elektronik seperti internet dan teknologi-teknologi baru sebagai alasan melakukan tindak plagiarisme (Jereb dkk., 2018).

Plagiarisme sendiri diatur dalam UU NO.28 Tahun 2014 mengenai hak cipta, dan sudah diatur secara jelas dan lengkap. Menurut UU ini, sebuah hak cipta atau *copyright* merupakan sebuah hak eksklusif bagi para pencipta yang timbul secara otomatis berdasarkan sebuah prinsip deklaratif setelah suatu karya atau hasil ciptaan berhasil diwujudkan dalam sebuah bentuk nyata tanpa mengurangi pembatasan-pembatasan sesuai ketentuan perundang-undangan yang berlaku.

1.2.Perumusan Masalah

Dari latar belakang yang telah dituliskan di atas maka rumusan masalah pada penelitian ini adalah bagaimana mengimplementasikan metode *cosine similiarity* pada sebuah sistem yang digunakan untuk mengukur tingkat kemiripan pada tugas akhir mahasiswa, dikarenakan semakin tinggi tingkat kesamaan pada suatu dokumen maka persentase plagiatnya juga semakin besar.

1.3.Pembatasan Masalah

Batasan masalah pada penelitian ini yang berupa sistem pendeteksi plagiarisme tugas akhir adalah yaitu :

1. Dokumen yang digunakan berbentuk file tugas akhir mahasiswa dan format yang digunakan adalah PDF.
2. Sistem deteksi plagiarisme yang dibuat hanya pada deteksi tugas akhir mahasiswa.
3. Penelitian ini hanya membahas seberapa besar tingkat kesamaan dokumen atau kemiripan dari suatu tugas akhir, sedangkan untuk menentukan plagiat atau tidaknya kembali ke peraturan instansi user yang menggunakannya.
4. Mengukur tingkat kesamaan dari tugas akhir mahasiswa yang berbahasa Bahasa Indonesia .

1.4.Tujuan

Tujuan dari penelitian ini adalah untuk membangun sebuah sistem guna mendeteksi sejauh mana tingkat kesamaan dalam tugas akhir mahasiswa sehingga dapat mengetahui seberapa persen tingkat plagiarisme suatu tugas akhir.

1.5.Manfaat

Berdasarkan dengan permasalahan dan tujuan penelitian, makan penulis mengharapkan penelitian ini dapat memberikan manfaat yang besar bagi penggunanya guna memudahkan untuk mengetahui seberapa besar tingkat plagiarisme dari dokumen tugas akhir mahasiswa.

1.6.Sistematika Penulisan

Sistematika penulisan dari penelitian ini yaitu:

BAB 1 : PENDAHULUAN, pada tahap pendahuluan ini bertujuan untuk menampilkan isi dari latar belakang, perumusan masalah, pembatasan masalah, tujuan, manfaat dan sistematika penulisan.

BAB 2 : TINJAUAN PUSTAKA DAN DASAR TEORI, pada tahap ini menampilkan penjelasan mengenai konsep, teori dan prinsip dasar, guna untuk memecahkan permasalahan pada tugas akhir dengan bersumber dari berbagai referensi yang ada secara relevan sesuai dengan penelitian yang dilakukan dan didukung dengan adanya indeks atau notasi-notasi keterangan sumber referensi yang didapat.

BAB 3 : METODE PENELITIAN, pada bab ini disampaikan metode-metode yang digunakan untuk melakukan perancangan sistem serta pendekatan guna mendapatkan solusi dari permasalahan yang ada. Solusi ini dapat berupa langkah-langkah metode yang harus ditempuh, waktu penelitian, sumber data, cara mengolah data tersebut, perhitungan, simulasi dalam komputer dan desain sistem yang nantinya akan dibuat.

BAB 4 : HASIL DAN ANALISIS PENELITIAN, memuat hasil penelitian yang telah dilakukan dan pengujian sistem dari data dan hasil penelitian yang dilakukan maupun data yang sudah dibuat.

BAB 5 : KESIMPULAN DAN SARAN, memuat kesimpulan-kesimpulan dari keseluruhan uraian bab sebelumnya beserta saran dari hasil yang diperoleh dan harapan dari pemanfaatan pengembangan sistem yang telah dilakukan demi pengembangan selanjutnya.

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1.Tinjauan Pustaka

Mengukur kemiripan pada *file* dokumen teks yang berbahasa Indonesia menggunakan metode *cosine*, dimana sistem ini dibangun guna mengukur tingkat kemiripan antar suatu dokumen yang berbahasa Indonesia dan dibangun dengan menggunakan algoritma *cosine similiarity*, cosine similiarity merupakan ukuran kesamaan antara dua buah vektor yang terdapat pada sebuah ruang dimensi yang didapat dari nilai kosinus yang merupakan hasil dari perkalian dua buah vektor yang dibandingkan, dikarenakan nilai kosinus dari 0^0 sendiri adalah 1 dan kurang dari 1 untuk nilai sudut yang lain, nilai sebuah kemiripan atau similiarity dikatakan mirip ketika nilai dari *cosine similiarity* adalah 1.

. Data yang digunakan dalam penelitian ini merupakan data *dummy* dimana data yang digunakan untuk perhitungan merupakan data tugas akhir dari universitas Sam Ratulangi Manado dimana data yang digunakan telah dapat digunakan untuk mengukur tingkat kemiripan dokumen (Ariantini dkk., 2016).

Analisa performa pengujian kesamaan dokumen biasa digunakan menggunakan dua metode sekaligus yaitu metode *jaccard* dan metode *cosine*, dimana kedua teknik tersebut digunakan pada sebuah *system* guna menguji kesamaan antar dokumen. Sistem ini digunakan untuk mengukur kemiripan abstrak dari sebuah dokumen, dimana keduanya memiliki performa yang tinggi dan dapat digunakan sebagai algoritma pembanding kesamaan dokumen, namun setiap metode memiliki tingkat keakurasiannya sendiri-sendiri. Seperti algoritma *cosine similiarity* yang dimana merupakan algoritma yang memiliki tingkat akurasi yang lebih baik dari pada algoritma *jaccard* hal ini terbukti bahwa algoritma *cosine similiarity*

sendiri memiliki tingkat akurasi kesamaan dokumen sebesar 0,949808, sementara algoritma pembandingnya yaitu algoritma jaccard sendiri memiliki tingkat akurasi kesamaan dokumen sebesar 0,949077, yang dimana rata-rata ini diambil dari perbandingan data 550 skripsi mahasiswa dengan bentuk judul dan abstrak(Sugiyamto dkk., 2014).

Di jaman yang digital seperti sekarang ini mengakibatkan banyaknya arsip-arsip dokumen skripsi yang terkumpul dalam bentuk *soft file* dan dokumen *soft file* tersebut tidak terklasifikasi dengan baik sehingga mengakibatkan proses untuk mencari kembali menjadi sulit, dimana mengakses informasi yang berada dalam dokumen-dokumen tersebut memerlukan banyak waktu, apalagi jika dokumen tersebut disimpan dalam sebuah *folder database* yang sama, maka dari itu diperlukan sebuah *database* yang mampu mengklasifikasikan dokumen-dokumen tersebut secara otomatis ke dalam suatu folder berbeda pada sebuah *database*, hal ini bertujuan agar lebih mudah dalam pengelolaan dokumen-dokumen yang ada. Hal yang menjadi tujuan dari penelitian ini adalah untuk membangun sistem yang mampu mengklasifikasikan dokumen secara otomatis dengan menggunakan metode *cosine similiarity*, dimana objek penelitian ini menggunakan dokumen-dokumen skripsi dalam bentuk elektronik (*soft file*), yang nantinya dokumen-dokumen ini akan secara otomatis diklasifikasikan ke dalam kategori-kategori yang berbeda sehingga nantinya diharapkan sistem yang dihasilkan oleh penelitian tersebut dapat membantu kegiatan pengarsipan dokumen ke depanya (Wahyuni, R. T., Prastiyanto, D., & Supraptono, 2017).

Deteksi plagiarisme bisa juga dilakukan memalui pemanfaatan daftar Pustaka, penelitian ini dilakukan dengan menggunakan judul sebuah dokumen lalu pada bagian daftar pustaka dimanfaatkan dalam pencarian kemiripan tema menggunakan *cosine similiarity*. Sebuah studi kasus yang dilaksanakan pada universitas Muhammadiyah Magelang didasari karena pada universitas tersebut belum memiliki sebuah sistem pendeteksi plagiarisme yang terkomputerisasi, maka pada akhirnya penelitian tersebut

dimaksudkan untuk membuat sebuah sistem yang mampu mendeteksi plagiarisme dengan menggunakan *cosine similiarity*. Sistem ini sendiri memanfaatkan daftar pustaka untuk menemukan kemiripan dalam tema agar dengan cepat dan tepat mampu mencari kemiripan dokumen secara cepat, dengan menggunakan batas awal 0,6 ditentukan sebagai nilai terkecil dalam kemiripan dan nilai 1 ditentukan untuk kemiripan paling tinggi. Dimana sebuah tugas akhir dikatakan plagiat atau bukan merupakan hasil keputusan dari instansi dimana *user* tersebut berada, lalu untuk mengambil tindakan selanjutnya dari hasil dokumen yang telah diketahui memiliki kemiripan yang tinggi tersebut apakah masuk kategori plagiat atau tidak berdasar keputusan instansi tersebut berada (Sejati dkk., 2018).

Pada tahun 2019 dilakukan sebuah penelitian yang menggunakan metode similiaritas kosine dan metode TF-IDF juga digunakan guna mendeteksi kemiripan pada suatu dokumen. Metode TF-IDF merupakan sebuah metode untuk menghitung bobot suatu kata(*term*) terhadap suatu dokumen. Metode ini merupakan metode yang efisien, mudah dan menghasilkan hasil yang akurat dalam mendeteksi kemiripan dokumen. Metode TF-IDF menggunakan penggabungan dua buah konsep untuk perhitungan bobotnya, yang pertama adalah menghitung bobot frekuensi kemunculan sebuah kata pada dokumen tertentu frekuensi dokumen yang mengandung kata tersebut. Frekuensi kemunculan kata dalam sebuah dokumen memberikan petunjuk seberapa penting kata tersebut di dalam sebuah dokumen. Sementara metode *cosine similiarity* digunakan untuk melakukan perhitungan kesamaan dari dokumen tersebut, penelitian ini menggunakan dokumen uji berupa teks abstrak tugas akhir Institut Teknologi Telkom Purwokerto, kumpulan dari dokumen tersebut dijadikan sebagai bahan untuk membandingkan kemiripan antar dokumen. Hasil dari pengujian dan analisis penelitian yang dilakukan dapat disimpulkan dalam penelitian tersebut adalah algoritma *cosine similiarity* dan pembobotan TF-IDF berhasil digunakan untuk mendeteksi kemiripan suatu dokumen. Dalam penelitian ini proses *stemming* pada proses *preprocessing* sangat

berpengaruh dalam hasil akhir nilai kemiripan dokumen, hasil menunjukkan nilai rata-rata perbedaan nilai kemiripan saat dilakukan proses *stemming* menunjukkan nilai yang lebih tinggi. Nilai yang didapat ketika proses *stemming* dilakukan dan tidak dilakukan adalah sebesar 10%. Kekurangan pada penelitian ini ketika proses *stemming* adalah waktu proses data yang lebih lama dibandingkan ketika tidak menggunakan proses *stemming* tersebut (Naf'an dkk., 2019).

Pada penelitian yang dilakukan pada tahun 2013 mengenai jarak ukuran kesamaan kosinus pada klasifikasi teks menyebutkan bahwa di dalam model ruang vektor , *cosine* banyak digunakan untuk mengukur kesamaan antara dua vektor dikarenakan perhitungannya sangat efisien terutama untuk jarak antar vektor. Sebagai komponen fundamental *cosine similiarity* sering kali diterapkan dalam menyelesaikan masalah *text mining*, seperti klasifikasi teks, peringkasan teks, pencarian informasi, menjawab pertanyaan mengenai kemiripan antar dua buah vektor dan sebagainya. Meskipun popular *cosine similarity* memiliki beberapa kekurangan seperti bias yang terjadi ketika menguji sampel yang diakibatkan oleh fitur nilai yang lebih tinggi. (Li & Han, 2013).

Tabel 2. 1. Kesimpulan daftar pustaka

No.	Judul Pustaka	Kesimpulan
1.	Pengukuran kemiripan dokumen teks bahasa indonesia menggunakan metode *cosine similarity*	*Test similiaity* antar dokumen dikatakan sama ketika *vector* menunjukkan 0^0, sementara kemiripan dikatakan berbeda jauh ketika sudut vectornya 180^0
2.	Analisa performa metode *cosine* dan *jacard* pada pengujian kesamaan dokumen	algoritma *cosine similiarity* memiliki tingkat akurasi kesamaan dokumen sebesar 0,949808, sementara

		algoritma pembandingnya algoritma *jaccard* memiliki tingkat akurasi kesamaan dokumen sebesar 0,949077
3.	Deteksi plagiarisme karya ilmiah dengan pemanfaatan daftar pustaka dalam pencarian kemiripan tema menggunakan metode *cosine similarity* (studi kasus: di Universitas Muhammadiyah Magelang)	Tujuan daripada penelitian kali ini adalah guna membangun sistem yang mampu untuk mengklasifikasikan dokumen secara otomatis dengan menggunakan metode *cosine similiarity*
4.	Penerapan algoritma *cosine similarity* dan pembobotan tf-idf pada sistem klasifikasi dokumen skripsi	Menggunakan metode berupa TF-IDF yang merupakan sebuah metode untuk menghitung bobot suatu kata(term) terhadap suatu dokumen. Metode ini terkenal efisien, mudah dan menghasilkan hasil yang akurat dalam mendeteksi kemiripan dokumen
5.	Distance weighted cosine similarity for text classification	Sebagai komponen fundamental *cosine similiarity* sering kali diterapkan dalam menyelesaikan masalah *text mining*, seperti klasifikasi teks, peringkasan teks, pencarian informasi, menjawab pertanyaan mengenai

		kemiripan diantar dua buah vektor dan sebagainya

Pada tabel 2.1 kesimpulan tinjauan Pustaka, merupakan kumpulan kesimpulan dari seluruh daftar Pustaka yang ada pada laporan ini.

2.2.Dasar Teori

2.2.1. Garba Rujukan Digital

Penelitian menggunakan ini menggunakan sumber data yang semuanya diunduh dari (GARUDA(Garba Rujukan Digital). GARUDA sendiri merupakan situs portal yang menjadi pusat referensi ilmiah di Indonesia. Pada mulanya situs ini Bernama RII atau singkatan dari Referensi Ilmiah Indonesia yang pertama kali dirilis pada 2 Mei 2010 oleh DEPDIKNAS (Departemen Pendidikan Nasional). Rujukan Ilmiah yang ada dalam portal GARUDA sendiri meliputi : dokumen skripsi, dokumen tesis, artikel jurnal ilmiah, makalah, *prosiding* atau *paper* akademis dan sebagainya.

2.2.2. Teks

Menurut A.Purba teks merupakan sebuah ungkapan dari Bahasa yang menurutkan isi, sintaksis, dan pragmatis merupakan satu kesatuan. Teks sendiri berasal dari kata berbahasa inggris yang berbunyi "*text"* yang berarti 'tenunan' (Purba & Situmorang, 2017). Pada penelitian tahun 2013 mengenai pembelajaran Bahasa Indonesia berbasis teks, teks sendiri merupakan sebuah ungkapan dari hasil pikiran manusia yang lekap, yang di dalamnya ada situasi dan konteksnya (Sufanti, 2013). Dari pengertian tersebut diartikan bahwasanya teks merupakan suatu kesatuan Bahasa yang mempunyai isi dan bentuk, baik itu lisan maupun tulisan yang disampaikan oleh pengirim kepada orang yang menerima untuk menyampaikan sebuah pesan tertentu (Permadi, 2006).

2.2.3. Text Mining

Menurut witten *text mining* merupakan sebuah pengembangan dari ilmu pengetahuan yang bertujuan untuk mendapatkan informasi asli (teks asli) (Ivory, 2004). Secara mudah hal ini dapat berarti mengekstrak suatu

dokumen atau teks guna mendapatkan informasi yang berguna demi tujuan tertentu. *Teks mining* sendiri juga sering disebut dengan penambangan data teks dengan tujuan menemukan data pengetahuan dari tekstural *database* hal ini mengacu pada proses ekstraksi pola data yang diambil dari dokumen teks (Tan, 1999). Dalam analisis *big data*, *text mining* merupakan alat yang ampuh yang memanfaatkan data tekstural yang tidak terstruktur dengan menganalisisnya dan mengekstrak pengetahuan-pengetahuan baru untuk mengidentifikasi pola yang signifikan dan korelasi yang signifikan dalam data (Hassani dkk., 2020).

Menurut Ronen Feldman *text mining* dapat diidentifikasikan sebagai cara atau proses untuk menggali suatu informasi dimana seorang pengguna berinteraksi dengan kumpulan dokumen dengan menggunakan peralatan analisis yang mendukung proses dan kegiatan *data mining* (Feldman & Sanger, 2007).

Terdapat beberapa proses dalam melakukan *text* mining untuk mengetahui kemiripan dokumen adalah sebagai berikut :

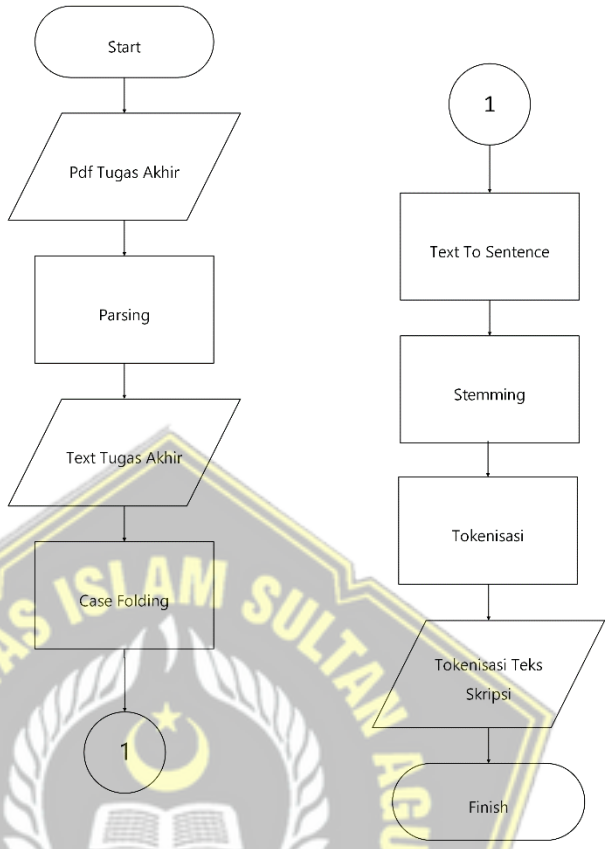
1. *Case Folding* yaitu merupakan tahapan pertama dalam proses penggalian data teks dimana mengubah semua huruf menjadi huruf kecil lalu dilanjutkan dengan menghilangkan karakter selain huruf dan angka.
2. *Tokenizing*/*Parsing* Merupakan tahapan kedua dalam proses penggalian data teks , proses ini berupa pemecahan kalimat menjadi kata tunggal atau frasa *(parsing)*.
3. *Stopwords Removal* merupakan tahapan ke tiga dalam proses *text mining*, tahap ini berupa pengambilan kata penting lalu dilajutkan dengan menghilangkan *stoplist/stopword. Stopword* sendiri merupakan kata penghubung dan kata pengganti, bisa juga merupakan kata preposisi seperti kata “dan”, “atau”, “dia”, “di”, “yang”, “dari” dan lain-lain .
4. *Stemming* merupakan tahapan proses terakhir yang gunanya untuk mengubah kata dari hasil *stopwords removal* menjadi bentuk kata dasar (root word).

2.2.4. Text Preprocessing

Dikarenakan ke tidak teraturan struktur pada dokumen teks , mengakibatkan proses menemukan data tersebut menjadi lama, mengakibatkan proses *text mining* memerlukan beberapa langkah untuk melakukannya, langkah-langkah ini pada dasarnya dimaksudkan guna mempersiapkan agar teks dapat diubah menjadi lebih terstruktur dan rapi. Dalam persiapan dokumen teks untuk proses *text preprocessing* merupakan sebuah tahapan proses yang sangat penting dalam menentukan kualitas tahap selanjutnya (Rahayu & RMS, 2018). Teks preprosessing sendiri memiliki tujuan untuk mempersiapkan dokumen yang tidak agar menjadi data yang terstruktur sehingga siap digunakan untuk menuju Langkah proses selanjutnya. *Text Preprocessing* merupakan sebuah tahapan awal dari proses dokumen teks guna mempersiapkan teks menjadi data yang siap untuk diolah guna proses yang lebih lanjut (Feldman & Sanger, 2007).

Pada proses teks preprosessing bertujuan sebagai tahapan awal suatu dokumen sebelum teks diolah untuk tahap selanjutnya, dimana tahapan proses seleksi data yang kemudian dilakukan pada setiap dokumen terkait (Nugroho, 2011).

Berikut merupakan langkah-langkah dari *text prepossessing*.
Gambar 2.1 merupakan flowchart dari proses tahapan *preprosessing*.



Gambar 2. 1. Flowchart Tahapan Preprosessing

1. Parsing PDF : Dalam ilmu komputer dan *linguistic*, *parsing* atau lebih formal disebut dengan *sysntactic analysis* yang merupakan sebuah proses menganalisis teks yang tersusun dari urutuan *token* (sebagai contoh : sebuah kata yang digunakan untuk menentukan apakah struktur gramatikal seperti apa yang akan digunakan (bisa saja lebih atau malah kurang)). Dokumen sendiri dapat tersusun dari beberapa bagian seperti Bahasa atau format tertentu, penguraian atau *parsing* berfungsi untuk memecah rangkaian dokumen menjadi komponen terpisah, pada tahap ini *parsing* mengambil data dari dokumen PDF lalu mengubannya menjadi bentuk teks.

2. Case Folding : Merupakan langkah dari urutan *text preprocesing* yang dimana tujuan dari *case folding* itu sendiri adalah untuk mengubah semua

huruf dalam ada di dalam suatu dokumen menjadi huruf kecil dimana hanya huruf ‘a’ sampai ‘z’ yang diterima, dalam *case folding* sendiri beberapa cara yang pergunakan seperti mengubah *text* menjadi *lowercase* dimana keseluruhan *text* diubah menjadi huruf kecil. Menghapus angka proses ini dilakukan jika angka tidak relevan dengan objek yang sedang dianalisis dalam hal ini bisa dicontohkan seperti penomoran nomor rumah, nomor telepon atau ponsel dll. Menghapus tanda baca, sama halnya dengan angka tanda baca dalam kalimat sendiri tidak akan memiliki pengaruh besar dalam proses *test preprocessing*, penghapusan tanda baca dilakukan dengan menghapus karakter seperti [!”#$%&’()*+,-./:;<=>?@[\]^_`{|}~]. Dan Langkah *case folding* yang terakhir adalah guna menghapus spasi di awal dan di akhir kalimat atau dokumen.

3. *Text to sentence* : sebuah proses dimana teks utuh hasil proses sebelumnya kemudian dipecah manjadi kalimat-kalimat penyusun teks tersebut.

4. *Stemming* : Merupakan sebuah teknik yang dilakukan guna memperkecil jumlah indeks yang berbeda suatu dokumen, juga dapat diklasifikasikan sebagai cara mengelompokkan kata-kata lain yang akar dan maknanya sama tetapi memiliki bentuk atau bentuk yang berbeda karena memiliki imbuhan yang berbeda, sebagai contoh *stemming* sebagai proses menghilangkan kata menjadi bentuk dasar yang dimana bentuk kata dasar tersebut tidak akan lagi berarti sama dengan akar kata (*root word*), sebagai contoh *stemming* adalah kata “mendengarkan”, “dengarkan”, “didengarkan” akan ditransformasikan menjadi kata “dengar”.

5. *Tokenisasi* : *Tokenisasi* Merupakan sebuah proses untuk membagi teks menjadi token *token* atau bagian tertentu. Sebagai contoh tokenisasi kalimat yang memecah menjadi kata seperti “Aku”, “Baru”, “Pergi”,

biasanya yang menjadi acuan pemisah antar *token* adalah spasi dan tanda baca.

2.2.5. Model Ruang Vektor

Model ruang vektor merupakan model yang populer saat ini dalam sistem temu kembali informasi. Model ini menunjukkan hasil pemulihan secara berurutan (Imran & Sharan, 2009). Ruang vektor sendiri tidak memerlukan sistem komputer yang redundansi sehingga waktu eksekusi lebih cepat dan efisien (Ramadhany, 2008).

Karakteristik yang dimiliki model ruang vektor :

1. Model vektor yang berdasarkan *keytrem*.
2. Model vektor yang mendukung *partial matching* (sebagian sesuai) dan penentuan peringkat dokumen.
3. Prinsip dasar model vektor adalah sebagai berikut :
   a. Dokumen dipresentasikan dengan menggunakan vektor *keyterm*.
   b. Ruang dimensi yang ada dalam model vektor ditentukan oleh keyterms.
   c. *Query* dipresentasikan dengan menggunakan vektor *keyterm.*
   d. Kesamaan dokumen *keytrem* dihitung berdasarkan jarak antar vektor.
4. Model ruang vektor sendiri memerlukan beberapa ketentuan :
   a. Bobot *keyterm* untuk vektor dokumen.
   b. Normalisasi *keyterm* untuk vektor *query*.
   c. Perhitungan jarak untuk vektor dokumen *keyterm.*
5. Kinerja Ruang Vektor
   a. Kinerja ruang vektor lebih efisien.
   b. Kinerja ruang vektor mudah digunakan dalam representasi.
   c. Kinerja ruang vektor dapat juga diimplementasikan pada dokumen *matching* dan *partial matching.*

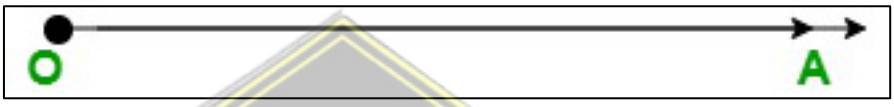
Prosedur model ruang vektor dapat dikeompokkan menjadi tiga tahap yaitu :

1. Pengindeksan dokumen.

2. Pembobotan indeks, untuk menghasil hasil dokumen yang relevan.
3. Memberikan peringkat dokumen berdasarkan ukuran kesamaan (*similiarity measure*).

Prinsip utamanya adalah bahwa *query* ditransformasikan menjadi vektor *query*, dan dokumen dalam kumpulan dokumen diubah menjadi vektor dokumen (Salton & McGill, 1983).

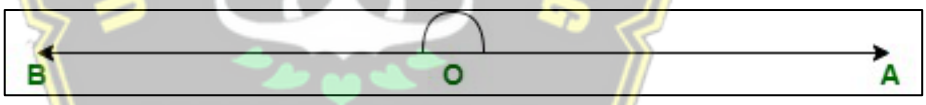
Menurut konsep mengenai aljabar *liner*, nilai dari sim(a,b) ketika sama adalah 0^0, seperti pada gambar 2.2



Gambar 2. 2 Zero Angle (00)

Zero angle merupakan sudut yang terbentuk ketika 2 buah *vector* saling tumpang tindih, hal ini mengakibatkan terbentuknya sudut 0^0, dalam *similiarity* sendiri sudut 0^0 berarti dokumen yang di bandingkan sama sehingga garis yang timbul hanya 1 garis lurus dari titik O menuju A.

Menurut konsep mengenai aljabar *liner*, nilai dari sim(a,b) adalah 180^0, seperti pada gambar 2.3



Gambar 2. 3. *Straight Angle* (180^0)

Staight angle merupakan sudut yang terbentuk ketika 2 buah *vector* saling bertolak belakang, hal ini mengakibatkan terbentuknya sudut 180^0 yang dapat diartikan bahwa 2 dokumen yang dibandingkan sama sekali tidak mirip.

2.2.6. Pembobotan Kata (*Term)*

Pembobotan kata atau istilah memiliki dampak besar pada kesamaan antara dokumen dan *query* pencarian. Jika bobot setiap kata dapat ditentukan dengan benar, hasil komputasi diharapkan dapat memberikan nilai kemiripan yang lebih baik dari kemiripan teks (Polettini, 2004). *Term frequency* (tf) merupakan istilah pengertian standar frekuensi dalam *corpus-based natural language processing*(NPL) yang berfungsi

menghitung berapa kali satu jenis (istilah/kata/ngram) muncul dalam korpus kemudian dipinjam untuk proses temu kembali informasi , proses tersebut setidaknya menghitung frekuensi dari suatu kata meski hanya muncul sekali dalam dokumen (Yamamoto & Church, 2001).

2.2.7. Metode *Cosine*

Metode *cosinus* sendiri menggambarkan suatu metode untuk menghitung kemiripan atau kemiripan dokumen, dimana Tf atau persamaan frekuensi term digunakan untuk menghitung kemiripan. (Fitri & Asyikin, 2015). Secara umum, fungsi *similarity* adalah fungsi yang mengambil dua objek dan mengembalikan kemiripan (*similarity*) antara kedua objek tersebut (Sejati dkk., 2018).

Kemiripan kosinus adalah ukuran kemiripan antara dua vektor dalam suatu ruang dimensional yang diperoleh dari nilai kosinus sudut yang diperoleh dari perkalian dua vektor yang dibandingkan. Ini karena kosinus dari 00 adalah 1 dan kosinus dari nilai lainnya kurang dari 1. Untuk sudut, jika nilai kemiripan kosinusnya adalah 1, maka nilai kemiripan dua buah vektor dikatakan mirip. Kesamaan kosinus digunakan dalam ruang positif di mana hasilnya dibatasi antara nilai 0 dan 1. Jika nilainya 1 dokumen dianggap serupa dan jika hasilnya 0 nilainya dianggap berbeda. Perhatikan bahwa pembatasan ini berlaku untuk rentang dimensi, dan kesamaan kosinus paling sering digunakan dalam ruang positif berdimensi tinggi. Misalnya, temu kembali informasi mengasumsikan bahwa setiap istilah memiliki dimensi yang berbeda dan menandai dokumen dengan vektor di mana nilai setiap dimensi sesuai dengan jumlah istilah dalam dokumen (Ariantini dkk., 2016).

Pada penelitian yang dilakukan pada tahun 2009 mengenai konsep klastering dokumen menggunakan konsep kesamaan ruang dan metode *cosine* telah menghasilkan hasil yang signifikan tentang dimensi matriks dengan referensi mengenai *k-rank*(*total number of pattern*) atau jumlah total pola dengan rata-rata menghasilkan kinerja yang tinggi dengan *f-*

measure sekitar 0,91 dan entropi sebesar 0.51. Ini merupakan peningkatan yang sangat signifikan ketika diterapkan dalam volume data yang besar (*multi7* dan *multi10 dataset*) sampai tingkat yang lebih dari 50% (Muflikhah & Baharudin, 2009).

Berikut adalah rumus *cosine similarity*.

$$Similiarity = \frac{A.B}{\|A\|.\|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \quad (1)$$

Keterangan : A_i = bobot *term* i dalam blok A

B_i = bobot *term* i dalam blok B

i = jumlah *term* pada kalimat

n = jumlah vektor

BAB III
METODE PENELITIAN

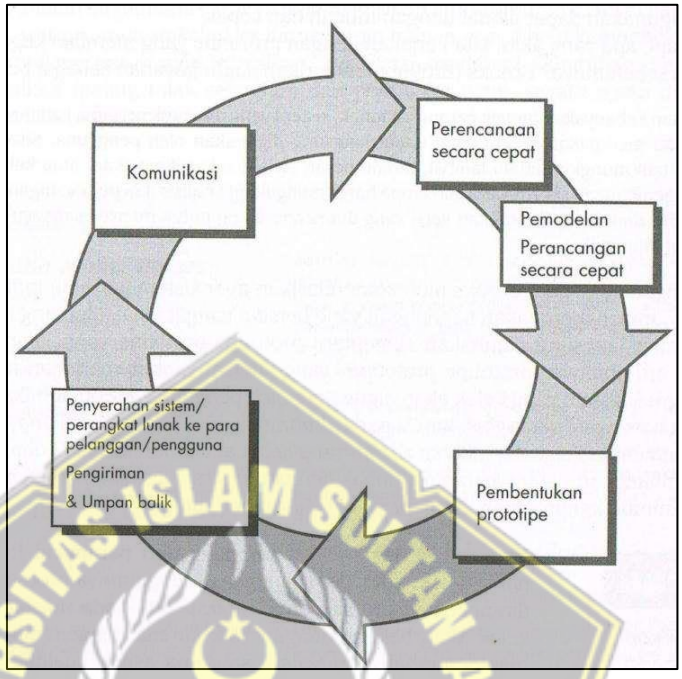
3.1.Tahapan Pengumpulan Data

Pada tahap ini yaitu tahapan pengumpulan data , data yang akan digunakan adalah data dokumen tugas akhir mahasiswa atau skripsi dengan format PDF dengan total berjumlah 15 buah dokumen yang diambil dari situs http://garuda .ristekbrin.go.id/ sebagai data acuannya dan untuk proses *training* menggunakan 10 dokumen dari keseluruhan data dan 5 sisanya digunakan untuk pengujian seberapa besar tingkat kemiripan-nya.

3.2.Metode Pengembangan Sistem

Pada penelitian ini metode yang digunakan untuk mengembangkan aplikasi ini adalah metode prototipe. Metode *prototype* sendiri merupakan salah satu metode pengembangan perangkat lunak yang memungkinkan adanya interaksi antara developer atau pengembang dengan *user* atau penggunanya , sehingga dapat mengatasi *bug* atau *error* dengan cepat antara pengembang dan pengguna.

Gambar 3.1 merupakan model pengembangan *prototype*



Gambar 3. 1. Model Pengembangan *prototype*

1. Komunikasi : Komunikasi sendiri merupakan tahapan awal dari model *prototype* yang berguna untuk mengidentifikasi permasalahan-permasalahan yang ada , serta informasi yang berguna demi membangun sebuah *system*.
2. Perencanaan : Pada tahap perencanaan sendiri merupakan tahapan yang dikerjakan sebagai tahap untuk menentukan suberdaya, spesifikasi yang berguna demi perkembangan dan kebutuhan *system*.
3. Permodelan : Tahap pemodelan merupakan representasi atau representasi dari model sistem yang akan dikembangkan. Pada tahap ini, prototipe yang dibangun dengan menggunakan sistem desain tentatif dievaluasi oleh pengguna atau pengguna apakah sesuai dengan apa yang diinginkan atau apakah diperlukan evaluasi di masa mendatang. Setelah memastikan bahwa sistem telah sesuai dengan keinginan pengguna, tahap pengkodean atau pembuatan aplikasi dimulai dengan perancangan sistem yang dibuat

menggunakan bahasa pemrograman PHP dan *framework CodeIgniter* bawaan dan *database MySQL*.

4. Konstruksi : Pada tahap konstruksi sendiri merupakan tahap dimana prototipe dibangun dan sistem yang dibangun diuji. Proses instalasi dan pemberian dukungan pengguna juga dilakukan untuk memastikan sistem bekerja dengan baik.
5. Penyerahan : Tahap penyerahan sendiri merupakan tahapan paling terakhir yang dibutuhkan untuk mendapatkan hasil *feedback*, dari hasil yang didapat *feedback* tersebut dapat digunakan sebagai bahan evaluasi untuk tahapan selanjutnya dan juga tahap pembuatan sistem.

3.3.Gambaran sistem

Platform yang digunakan untuk aplikasi ini adalah berupa *website*, dikarenakan sebuah *website* memiliki kemudahan akses tanpa harus mengistal apa pun. sistem ini dibuat dengan menggunakan Bahasa pemrograman PHP dikarenakan Bahasa pemrograman ini bersifat open *source* , lebih fleksibel dikarenakan mudah dikombinasikan dengan fungsi Bahasa pemrograman lainya, memiliki banyak bantuan dan *library support*, memiliki tingkat pemuatan data ke *database* yang cepat dan stabil.

Framework Codeigniter4 digunakan untuk membangun *system* ini, kelebihan codeigniter4 sendiri adalah sudah menggunakan versi PHP versi 7.2, memiliki performa eksekusi yang sangat cepat, minim konfigurasi sehingga memudahkan konfigurasi, dokumentasi yang lengkap , *maintance* yang mudah .

3.4.Identifikasi perangkat lunak

Identifikasi perangkat lunak yang digunakan dalam pembuatan *system* ini adalah :

1. PHP 7.3.4

Bahasa pemrograman yang digunakan dalam penelitian ini adalah Bahasa PHP dengan versi 7.3.4 digunakan dikarenakan Bahasa ini mudah digunakan untuk membangun *website*.

2. Codeigniter4 *Framework*

Untuk mengimplementasikan model ke dalam aplikasi berbasis *website*, penelitian ini menggunakan *framework* Codeigniter4 dikarenakan *framework* tersebut sangat ringan dan mudah digunakan, serta memiliki banyak pilihan *library* yang mendukung.

3. *Visual Studio Code*

Visual Studio Code dipilih sebagai *text editor* pada pengembangan aplikasi dan penelitian ini, *Visual Studio Code* dipilih dikarenakan mendukung banyak Bahasa pemrograman dan *framework*, *multi* platform, performa yang sangat cepat, mempunyai banyak *extensions* yang dapat mempermudah proses peng-codingan *website*.

4. Xampp

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri, yang terdiri atas program Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP.

5. *Library spatie/pdf-to-text*

Library Spatie/pdf-to-text dipilih sebagai metode *parsing* dimana metode ini memproses data berbentuk PDF agar dapat dipecah dan diubah menjadi teks.

6. Library Myth-Auth

Library Myth-Auth digunakan dalam pembuatan aplikasi ini, library ini memungkinkan membuatfitur login dengan mudah dan cepat sehingga memudahkan proses pembuatan website.

3.5.Aktor

Sistem ini terdapat 2 aktor yaitu mahasiswa dan admin yang akan mengoperasikan sistem ini :

1. Admin

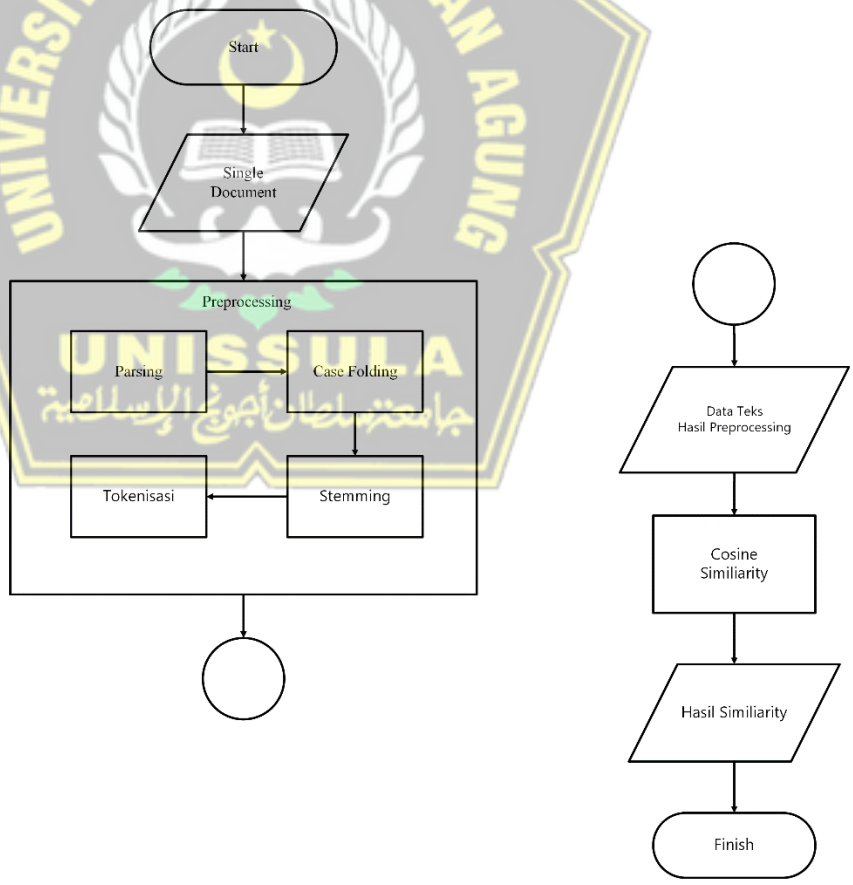
   Admin yaitu *user* yang dapat melakukan pengecekan kemiripan dan menambah data dari *corpus*.

2. Mahasiswa

   Mahasiswa merupakan *user* yang hanya dapat melakukan pengecekan kemiripan dan tidak bisa melakukan penambahan data pada *corpus*.

3.6.Perancangan arsitektur sistem

Dalam perancangan arsitektur sebuah *system*, diperlukan *flowchar* yang *menunjukan* langkah-langkah bagaimana sebuah *system* akan berjalan. Gambar 3.2 merupakan *flowchart* teks preprocessing aplikasi yang akan dibangun :



Gambar 3. 2 Flowchart teks *preprocessing*

Gambar 3.2 merupakan *flowchart* teks *preprocessing system* yang akan berjalan dimana tahapannya adalah :

1. *Parsing* untuk tahap yang pertama adalah *parsing*, pada tahap ini *file* PDF yang masuk akan diekstrak menjadi teks. Sebagai contoh tahapan *parsing* adalah sebagai berikut : *file* Teknik informatika fakultas teknik industri universitas islam sultan agung.pdf *file* ini berisi sebuah kalimat yang berisi "Teknik Informatika Fakultas Teknik Industri Universitas Islam., Sultan Agung!" dan ketika di *parsing* maka akan menjadi sebuah *file* teks biasa yang isinya kalimat "Teknik Informatika Fakultas Teknik Industri Universitas Islam., Sultan Agung!" yang tidak lagi berformat PDF.
2. *Case Folding* tahap *case folding* adalah tahap dimana semua huruf diubah menjadi huruf kecil serta melakukan proses *cleaning* dokumen dari karakter-karakter lain selain huruf dan angka. Sebagai contoh tahapan *case folding* adalah sebagai berikut. Dimana kalimat awal dari tahapan *parsing* adalah "Teknik Informatika Fakultas Teknik Industri Universitas Islam., Sultan Agung!" maka setelah di proses dalam tahap *case folding* maka akan menjadi "teknik informatika fakultas Teknik industri universitas islam sultan agung".
3. *Stemming* proses *stemming* merupakan proses dimana menghapus semua imbuhan pada kata sehingga hanya tersisa kata dasar (*root words*). Contoh *stemming* sendiri dapat di contohkan sebagai berikut : kalimat awal yang berupa "Sesungguhnya kuingin kamu terus ada di sampingku, tetapi aku tak bisa menghentikanmu" akan menjadi "sungguh ingin kamu terus ada di samping".
4. Tokenisasi proses terakhir dadi alur teks preprosessing dimana kalimat diproses sehingga membentuk token atau dipisah per kata. Sebagai contoh dalam tahap preprosessing adalah akan menjadi "sungguh", "ingin", "kamu", "terus", "ada", "di", "samping".

3.7.Perancangan Database

Tabel 3. 1.Docs

Atribut	Tipe	Keterangan
id	int(11)	AUTO_INCREMENT
file_name	varchar(255)	Nama File Yang Di *upload*
author	varchar(255)	Nama author yang di *upload*
release_year	varchar(23)	Tahun rilis yang di *upload*

Pada tabel 3.1 merupakan tabel docs dimana tabel ini digunakan untuk menyimpan nama dokumen, author dokumen dan tahun rilis dokumen yang di *upload* . Terdapat 4 kolom pada tabel docs yang memiliki fungsi penyimpanan berbeda-beda.

Tabel 3. 2.Docs_sentence

Atribut	Tipe	Keterangan
sentence_id	int(11)	AUTO_INCREMENT
doc_id	varchar(255)	Berisi *id* dokumen yang di *upload* pada tabel Docs kolom id.
text_sentence	varchar(255)	Berisi teks berbentuk kalimat hasil dari proses *text to sentence*.
key_sentence	varchar(250)	Berisi tentang urutan kalimat yang masuk per dokumen

Pada tabel 3.2 Docs_sentence dimana tabel ini mempunyai 3 kolom dan berfungsi untuk menyimpan *id* dokumen, kalimat dan urutan kalimat yang disimpan.

Tabel 3. 3.Docs_tokens

Atribut	Tipe	Keterangan
tokens_id	int(11)	AUTO_INCREMENT
doc_id	varchar(255)	Berisi *id* dari token berdasarkan *id* dari yang di upload pada tabel *docs*
token	varchar(255)	Berisi token dari hasil tokenisasi

Pada tabel 3.3.Docs_token tabel ini mempunyai 3 kolom yang berfungsi sebagai penyimpanan *token* dan *id* dari *sentence* yang di *upload*.

Tabel 3. 4.Corpus_Docs

Atribut	Tipe	Keterangan
id	int(11)	AUTO_INCREMENT
file_name	varchar(255)	Nama File Yang Di *upload*
author	varchar(255)	Nama author yang di *upload*
release_year	varchar(23)	Tahun rilis yang di *upload*

Pada tabel 3.4 merupakan tabel docs_corpus dimana tabel ini digunakan untuk menyimpan nama dokumen, *author* dokumen dan tahun rilis dokumen yang di *upload* . Terdapat 4 kolom pada tabel *docs* yang memiliki fungsi penyimpanan berbeda-beda. Perbedaan dari tabel *docs* adalah tabel docs_corpus merupakan tabel yang gunakan sebagai acuan perbandingan teks.

Tabel 3. 5.Corpus_tokens

Atribut	Tipe	Keterangan
token_id	int(11)	AUTO_INCREMENT
doc_id	varchar(255)	Berisi *id* dari token berdasarkan *id* dari yang di upload pada tabel *docs*
token	varchar(255)	Berisi token dari hasil tokenisasi

Pada tabel 3.5.Corpus_token tabel ini mempunyai 3 kolom yang berfungsi sebagai penyimpanan token dan *id* dari doc_id yang di *upload*. Perbedaan dari tabel docs_tokens adalah tabel corpus_tokens merupakan tabel yang gunakan sebagai acuan perbandingan teks.

Tabel 3. 6 docs_result_similiarity

Atribut	Tipe	Keterangan
id	int(11)	AUTO_INCREMENT
file_name	varchar(255)	Nama file Yang di *upload*
doc_id	varchar(255)	Id dokumen yang didapat pada tabel docs
result	decimal(6,2)	Hasil similiarity
key_token	int(255)	Hasil *similiarity* dari dokumen terkait

Pada tabel 3.6 docs_result_similiarity merupakan tabel yang berfungsi untuk menyimpan *id*, nama doc_id , *result* dan hasil key_token dari dokumen terkait yang di *upload* untuk dicek *similiarity*-nya.

Tabel 3. 7 User

Atribut	Tipe	Keterangan
id	int(11)	Berisi *id* dari *user* yang terdaftar di tabel *user*
email	varchar(255)	Berisi email yang didaftarkan
username	varchar(30)	Berisi *user* yang didaftarkan
password_hash	varchar(255)	Berisi *password* yang di enkripsi dengan metode *hash*
reset_hash	varchar(255)	Berisi *password hash* yang digunakan untuk *mereset password*

reset_at	datetime	Berisi tanggal reset *password* terakhir
reset_expires	datetime	Berisi tanggal *kadaluarsa* reset *password*
activate_hash	varchar(255)	Berisi *password hash* yang aktif
status	varchar(255)	Status verifikasi pengguna
status_message	varchar(255)	Berisi status pesan yang di tautkan
active	tinyint(1)	Berisi status aktif
force_pass_reset	tinyint(1)	Mewajibkan pengguna mengganti *password* ketika *login*
created_at	datetime	Berisi tanggal data user dibuat
updated_at	datetime	Berisi tanggal data user diupdate
deleted_at	datetime	Berisi tanggal data user dihapus

Tabel 3. 7 user merupakan tabel yang memuat data pengguna yang dapat mengakses sistem, data pengguna meliputi data admin.

Tabel 3. 8 auth_logins

Atribut	**Tipe**	**Keterangan**
id	int(11)	Berisi *id* dari *user* yang terdaftar di tabel auth_logins
ip_address	varchar(255)	Berisi *ip addres* yang mengakses *login* aplikasi

email	varchar(255)	Berisi email user yang didaftarkan
user_id	varchar(255)	Berisi urutan *user id* yang tersimpan
date	int(11)	Berisi tanggal dan waktu *user login*
success	tinyint(1)	Berisi status sukses *login user* yang *login*, jika menunjukkan hasil 1 maka *login* berhasil, dan jika menunjukkan hasil 0 maka *login* gagal

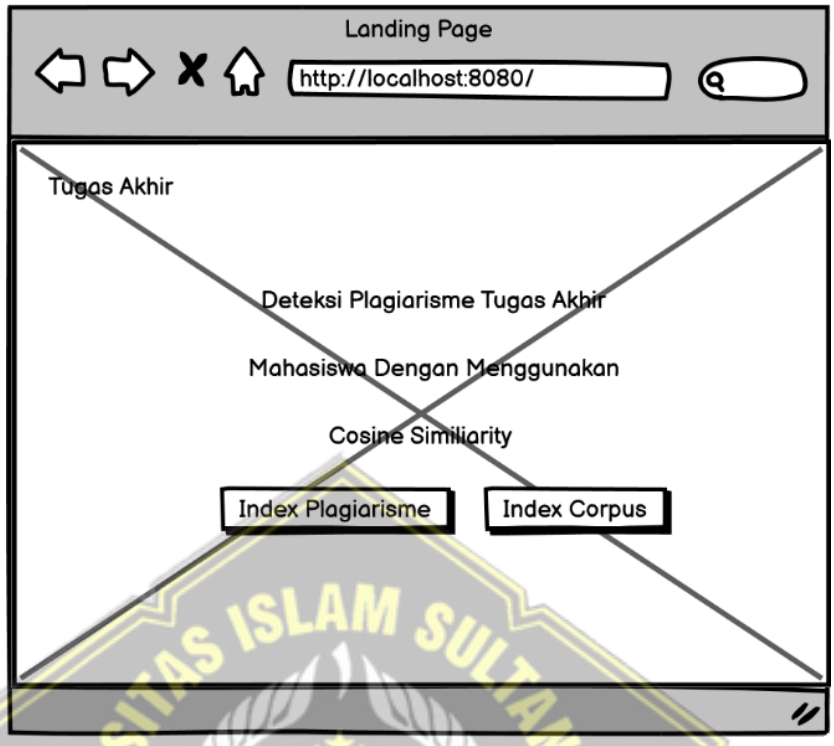
Tabel 3.8 auth_logins merupakan tabel yang berguna untuk menyimpan data *login user* , data ini berupa data *ip*, e-*mail*, waktu mengakses dan status sukses atau tidak *user* tersebut *login*

*3.8.*Perancangan *User Interface*

Berikut ini merupakan rancangan / *design system* yang digunakan pada penelitian ini :

1. Halaman *Landing Page*

   Halaman 3.3 merupakan perancangan antar muka untuk halaman *landing page* :

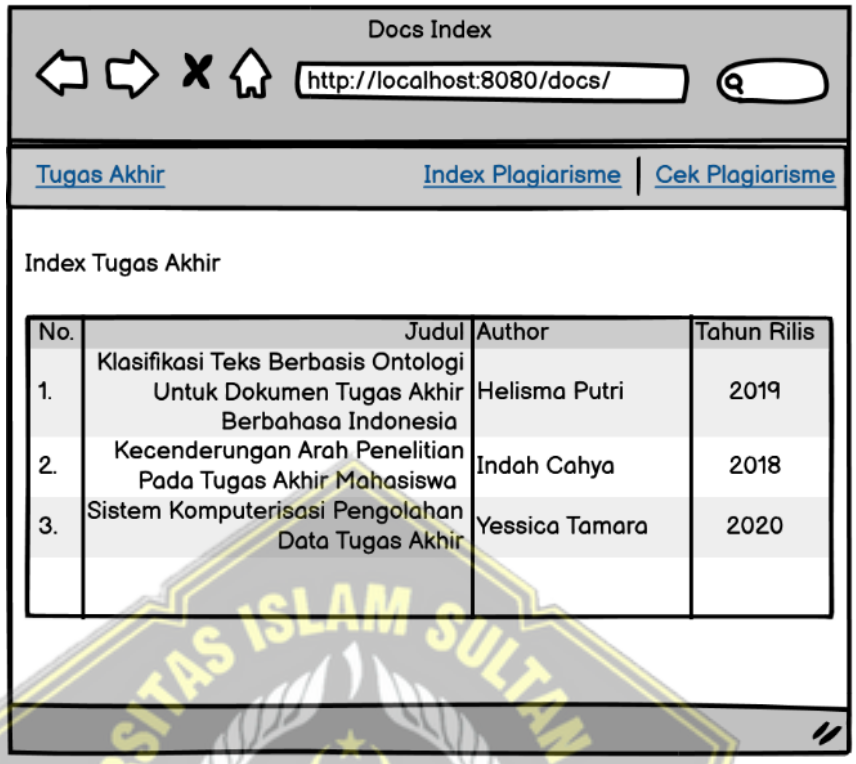


Gambar 3. 3. Halaman *Landing Page*

Gambar 3.3 merupakan gambaran rancangan untuk halaman *landing page*, *landing page* sendiri merupakan halaman yang dibuat dengan tujuan spesifik untuk mengarahkan *user* dengan fitur yang disediakan, untuk halaman *landing page* ini berisi mengenai judul tugas akhir dan *link* menuju halaman *docs index* dan halaman *database corpus.*

2. Halaman *Docs Index*

   Gambar 3.4 merupakan rancangan halaman antar muka untuk halaman *docs index.*

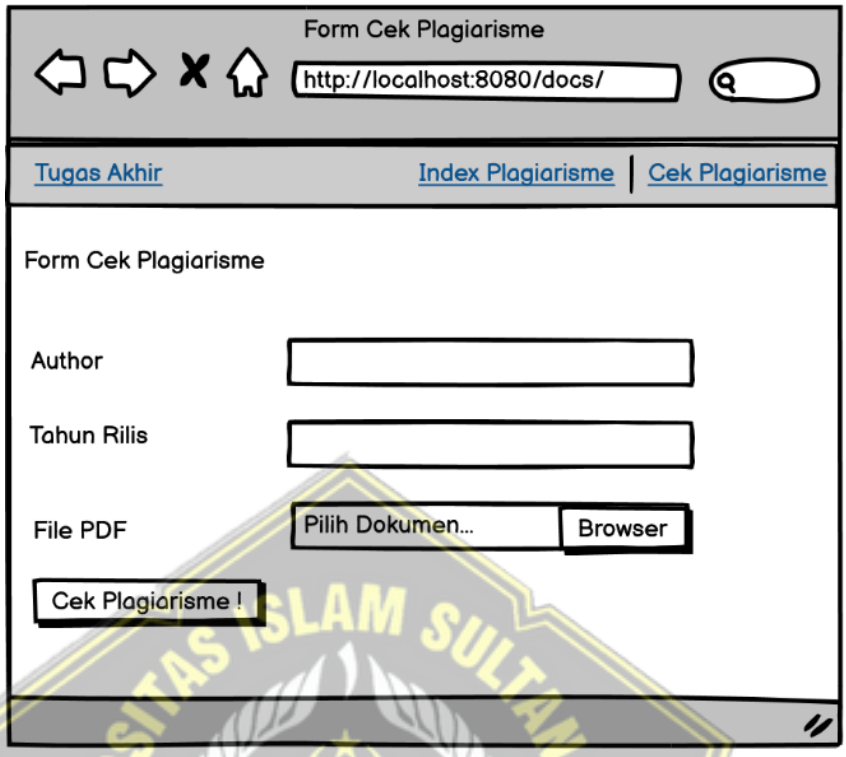


Gambar 3. 4. Halaman *Docs Index*

Gambar 3.4 merupakan rancangan halaman antar muka untuk halaman *docs index*, pada halaman ini berisi tentang judul dokumen, *author* tugas akhir, dan tahun rilis dokumen tugas akhir halaman ini akan terisi otomatis ketika pengguna melakukan pengecekan dokumen dan mengisi *form* yang ada pada halaman *form* cek plagiarisme mahasiswa.

3. Halaman *Form* Cek Plagiarisme

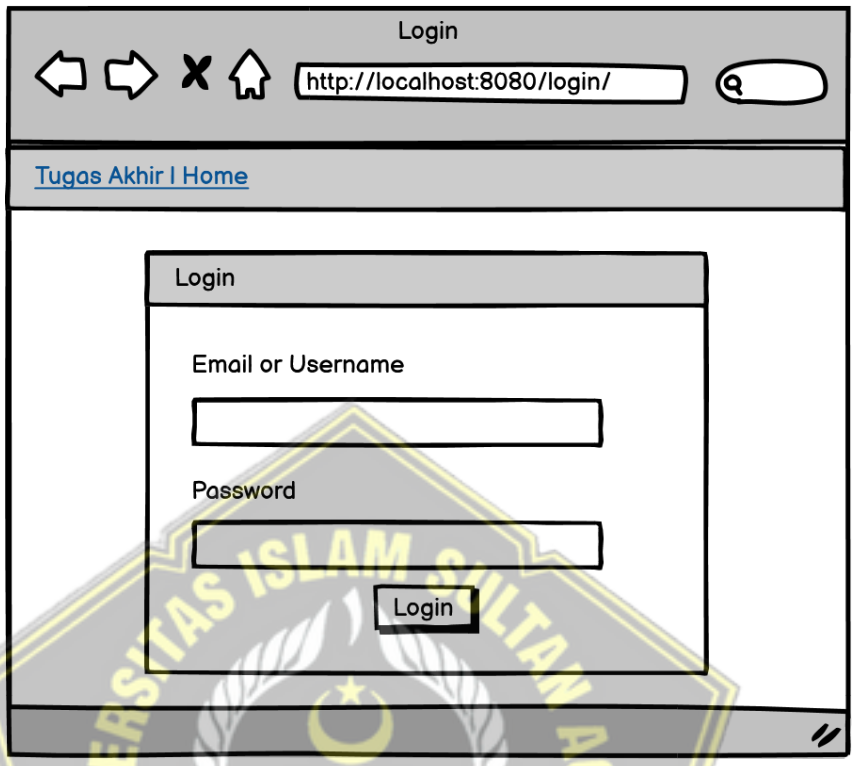
   Gambar 3.5 merupakan halaman rancangan untuk halaman *form* cek plagiarisme



Gambar 3. 5. Halaman *Form* Cek Plagiarisme

Gambar 3.5 merupakan halaman rancangan untuk halaman *form* cek plagiarisme, pada halaman ini *user* harus mengisi *form author* yang difungsikan sebagai *form* untuk pembuat dokumen tugas akhir tersebut, tahun rilis berisi tahun rilis tugas akhir yang akan di cek plagiarismenya , *form file* pdf berguna sebagai tempat *upload* dokumen tugas akhir yang akan di cek pagiarismenya.

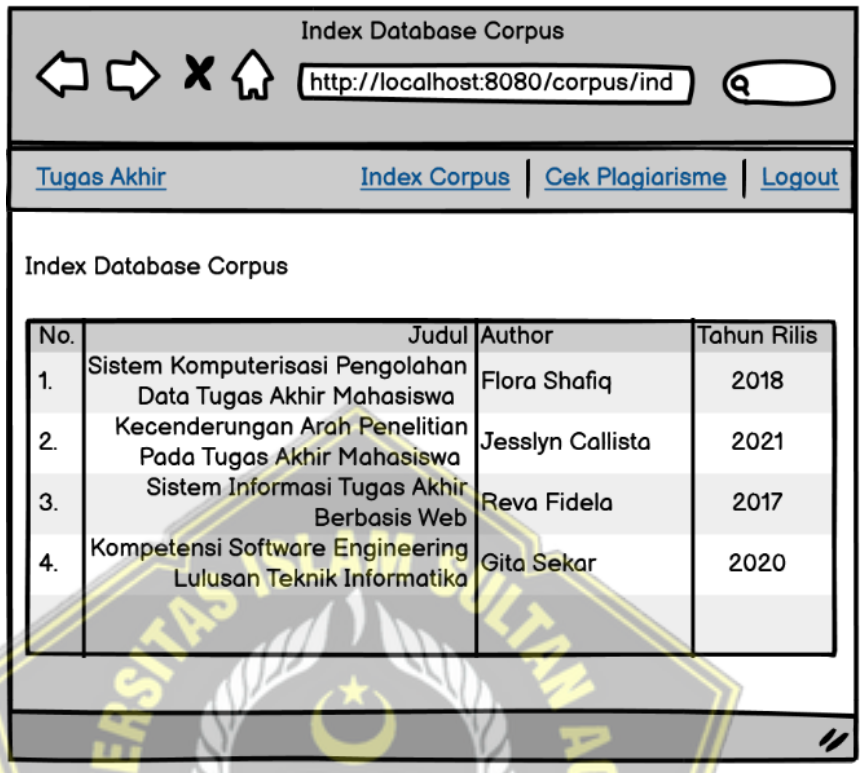
4. Halaman Login



Gambar 3. 6. Halaman Login

Gambar 3.6 merupakan halaman *login*, halaman ini difungsikan untuk admin dikarenakan halaman ini difungsikan untuk mengakses data corpus.

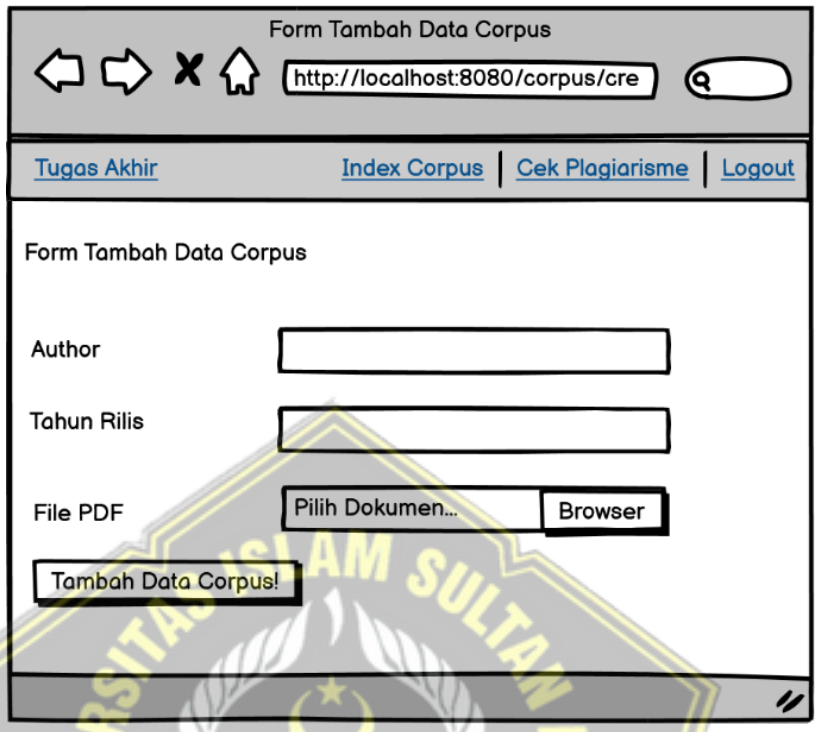
5. Halaman Index Database Corpus



Gambar 3. 7. Halaman Index Database Corpus

Gambar 3.7 merupakan rancangan halaman antar muka untuk halaman *index daabase corpus*, pada halaman ini berisi tentang judul dokumen, *author* tugas akhir, dan tahun rilis dokumen tugas akhir halaman ini akan terisi otomatis ketika admin mengisi data *corpus* sebagai data acuan nantinya.

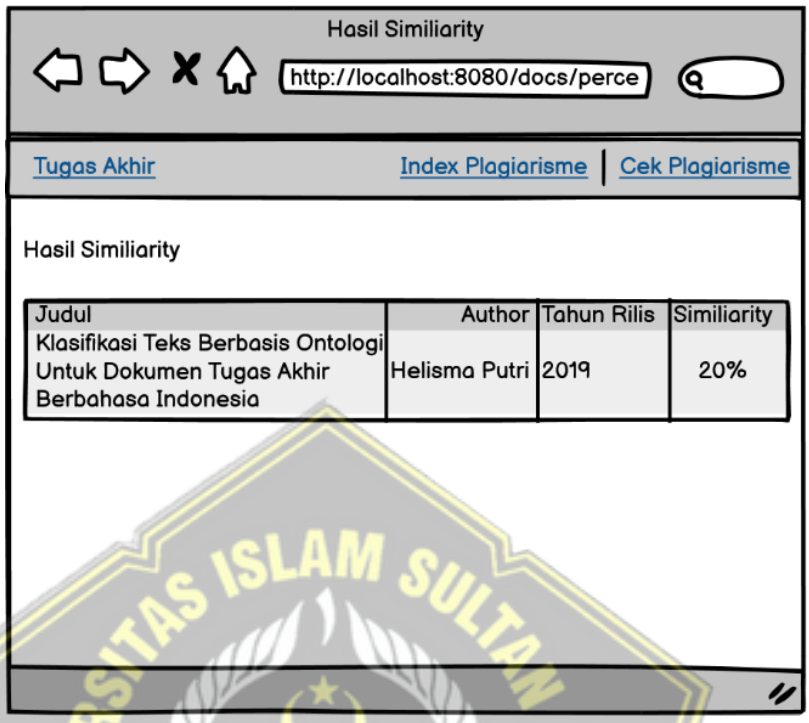
6. Halaman Tambah Data Corpus



Gambar 3. 8. Halaman Form Tambah Data Corpus

Gambar 3.8 merupakan halaman rancangan untuk halaman *form* tambah data *corpus*, pada halaman ini *admin* harus mengisi *form author* yang difungsikan sebagai *form* untuk pembuat dokumen tugas akhir tersebut, tahun rilis berisi tahun rilis tugas akhir yang akan di cek plagiarismenya , *form file* pdf berguna sebagai tempat *upload* dokumen tugas akhir yang akan di tambahkan ke *database corpus.*

7. Halaman Hasil Cek Similiarity



Gambar 3. 9 Halaman Hasil Cek *Similiarity*

Gambar 3.9 Halaman hasil cek *similiarity* merupakan halaman yang berguna untuk menampikan informasi dokumen yang di cek *similiarity*-nya dan juga menampikan hasil *similiarity* pada dokumen tersebut.

BAB IV
HASIL DAN ANALISI PENELITIAN

4.1. Hasil Perhitungan

Di dalam perhitungan untuk memperoleh hasil kemiripan pada suatu dokumen menggunakan metode *cosine similiarity*. Dimana nilai yang dihasilkan pada suatu dokumen digunakan sebagai bahan acuan pertimbangan suatu dokumen tersebut terindikasi melakukan plagiarisme atau tidak.

4.2.Sampel Dokumen

Sampel merupakan sebagian objek dari populasi yang diambil untuk dijadikan sumber data penelitian , sampel yang diambil merupakan data tugas akhir mahasiswa yang diunduh dari *website* http://garuda .ristekbrin.go.id/ sebagai data acuannya sejumlah 15 dokumen . Data yang sudah diunduh kemudian 10 dokumen digunakan sebagai dokumen *corpus* dan 5 dokumen sisanya digunakan sebagai dokumen yang dibandingkan.

4.3.Implementasi Teks Preprocessing

1. *Parsing*

```
1  //memulai library pdftotext
2   $path = 'c:/Program Files/Git/mingw64/bin/pdftotext';
3   $pdfttxt  = strtolower(Pdf::getText($fileDokumen, $path));
4
```

Gambar 4. 1 Parsing

Gambar 4.1 Parsing merupakan tangkapan layar code untuk parsing dimana untuk memulai library-nya yaitu library pdf to text dilakukan dengan cara memanggil *path* atau tempat dimana library tersebut disimpan , lalu dilanjutkan dengan memanggil *code library nya* yang di kombinasikan dengan *file* dokumen yang di upload dan path yang sudah di sebutkan di atas tadi.

Dari hasil *coding* di atas maka akan mendapatkan hasil berupa teks yang di *parsing* dari *file* pdf menjadi : “Jurnal Sarjana Teknik Informatika Vol. 6, No. 2, Juni 2018, Pp. 43-52 E-ISSN 2338-5197 43 Rancang Bangun Aplikasi Pengecekan Kemiripan Judul Skripsi Dengan Metode Cosine Similarity… Jurnalsarjana@tif.uad.ac.id”

2. *Case Folding*

   Gambar 4.1 merupakan gambaran tangkap layar dari proses parsing maka selanjutnya dilanjutkan pada tahap case folding yang akan di jelaskan pada gambar 4.2

Gambar 4. 2 case Folding

Gambar 4. 2 case Folding merupakan hasil tangkapan layar proses *coding case folding*, proses ini masih menjadi proses lanjutan dari *parsing*, dimana pdf yang di *parsing* menjadi teks lalu diubah menjadi huruf kecil dengan metode strtlower, lalu dilanjutkan dengan proses

cleaning teks dari karakter selain huruf dan angka menggunakan preg_replace ('/[^\p{L}\p{N}.]/', " ", $pdfttxt). Proses ini akan menggantikan semua karakter non huruf alphabet dan angka dengan spasi. Dari hasil *coding* diatas maka hasil yang didapatkan adalah " jurnal sarjana teknik informatika vol 6 no 2 juni 20063 pp 43 52 e issn 2338 5197 43 rancang bangun aplikasi pengecekan kemiripan judul skripsi dengan metode cosine similarity studi kasus program studi teknik informatika uad ibnu abdullah apriyantoa 1 eko aribowo a 2 a program studi teknik informatika universitas ahmad dahlan prof ".

3. *Stemming*

Pada gambar 4.2 case folding mendapatkan hasil berupa teks yang sudah bersih dari karakter-karakter non *alphabet* dan angka setelah itu maka dilanjutkan dengan proses *stemming*

```
1  //stemming
2          $stemmerFactory =
3              new \Sastrawi\Stemmer\StemmerFactory();
4      $stemmer  = $stemmerFactory->createStemmer();
5      $output = $stemmer->stem($pdfttxtCsfdg);
```

Gambar 4. 3 Stemming

Gambar 4.3 Stemming merupakan tangkapan layar dari proses *stemming*, setelah teks di *case folding* maka selanjutnya dimulai dengan proses *stemming* yang pada awalnya harus memanggil *path* dari *library* sastrawi *stemmer* , lalu dilanjutkan dengan *code stemming* dan di akhiri dengan *output* yang nantinya akan menghasilkan teks yang sudah di *stemming*. Dari *coding* di atas maka akan menghasilkan data berupa : "jurnal sarjana teknik informatika vol ", " 6 no ", " 2 juni 20063 pp ",

“43 52 e issn 2338 5197 43 rancang bangun aplikasi pengecekan kemiripan judul skripsi dengan metode cosine similarity studi kasus program studi teknik informatika uad ibnu abdullah apriyantoa 1 eko aribowo a 2 a program studi teknik informatika universitas ahmad dahlan prof dr”.

4. Tokenisasi

Pada gambar 4.3 yang menjelaskan proses *stemming* maka dilanjutkan dengan proses tokenisasi di 4.4.

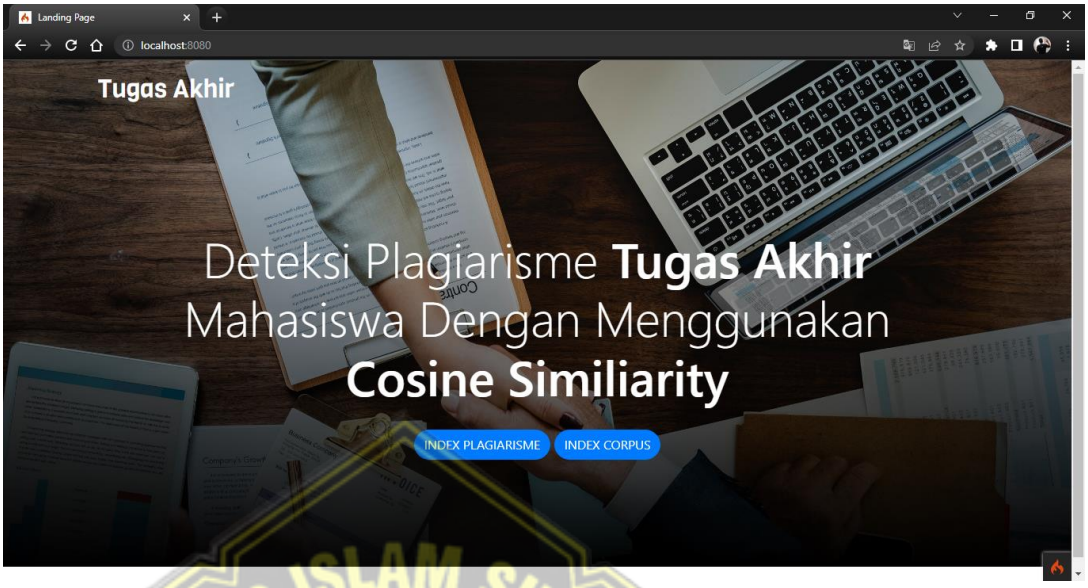
Gambar 4. 4 tokenisasi

Gambar 4.4 tokenisasi merupakan tangkapan layar dari proses dengan nama yang sama yaitu tokenisasi, dimana setelah proses *stemming* selesai lalu teks di ubah menjadi bentuk token atau kata. Dari *coding* di atas maka akan menghasilkan hasil seperti "jurnal", "sarjana", "teknik", "informatika", "vol".

4.4.Hasil Implementasi Sistem

Setelah mengumpulkan data sampel yang dibutuhkan sebagai bahan uji sistem, selanjutnya diimplementasikan di sistem. Berikut ini merupakan hasil tangkapan layar hasil implementasi sistem

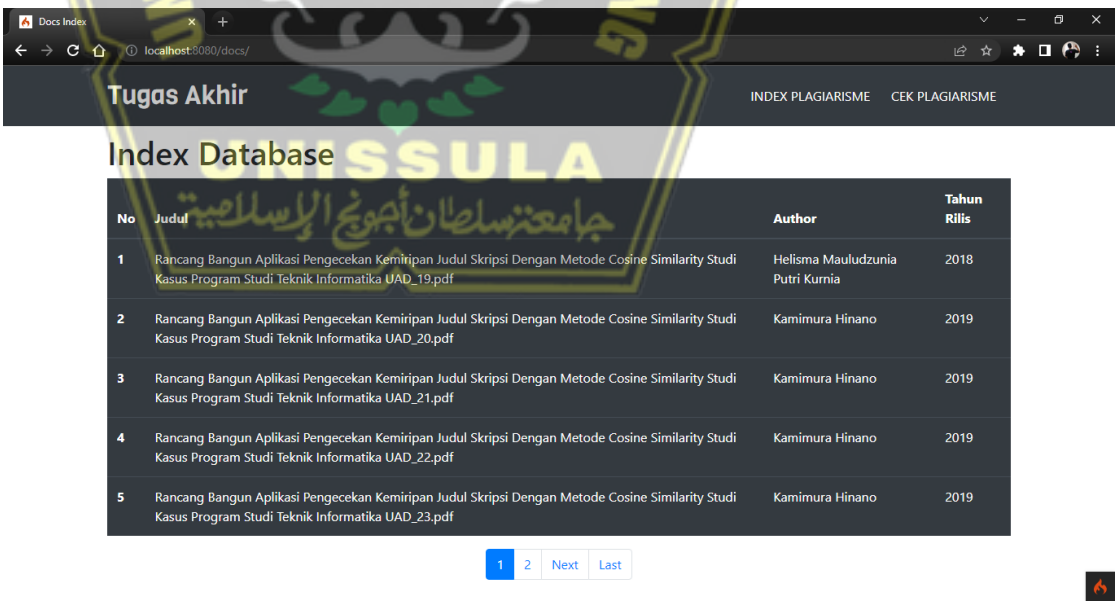
1. Halaman Landing Page



Gambar 4. 5 Halaman *Landing page*

Gambar 4.5 adalah tampilan layar yang ditampilkan pertama kali saat mengakses sistem, halaman ini berisi judul tugas akhir dan 2 *link* yang nantinya menuju *index similiarity* dan yang satunya untuk *login*.

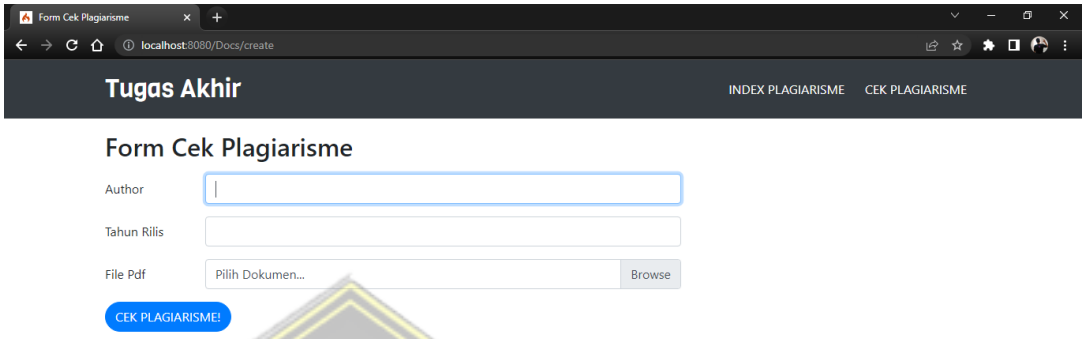
2. Halaman Docs Index



Gambar 4. 6 halaman docs indeks

Gambar 4.6 merupakan tampilan layar yang ditampilkan ketika mengakses *index plagiarism* halaman ini berisi tentang judul , *author* dan tahun rilis dokumen yang di *upload* atau di cek.

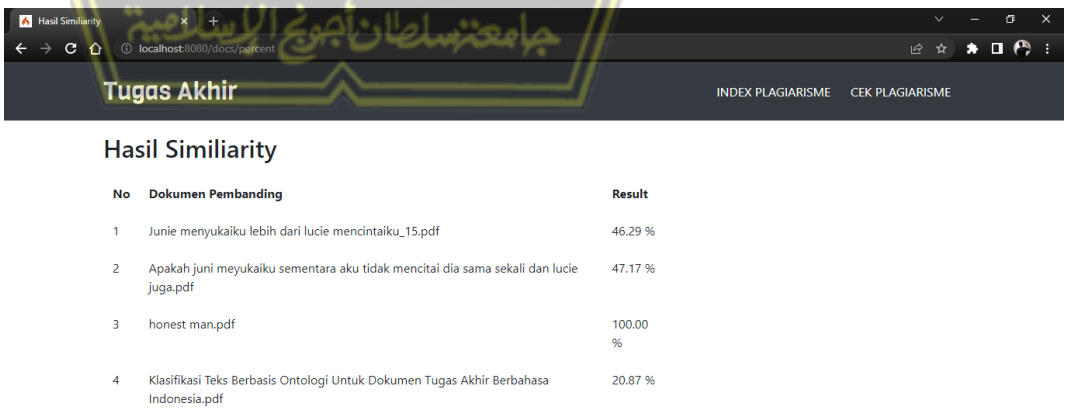
3. Halaman Form Cek *Plagiarisme*



Gambar 4. 7 Halaman *Form* Cek *Plagiarisme*

Gambar 4.7 merupakan halaman tangkapan layar untuk *form* cek *plagiarism* pada *form* ini diisi dengan nama author, tahun rilis dan *file pdf* yang akan di *upload*.

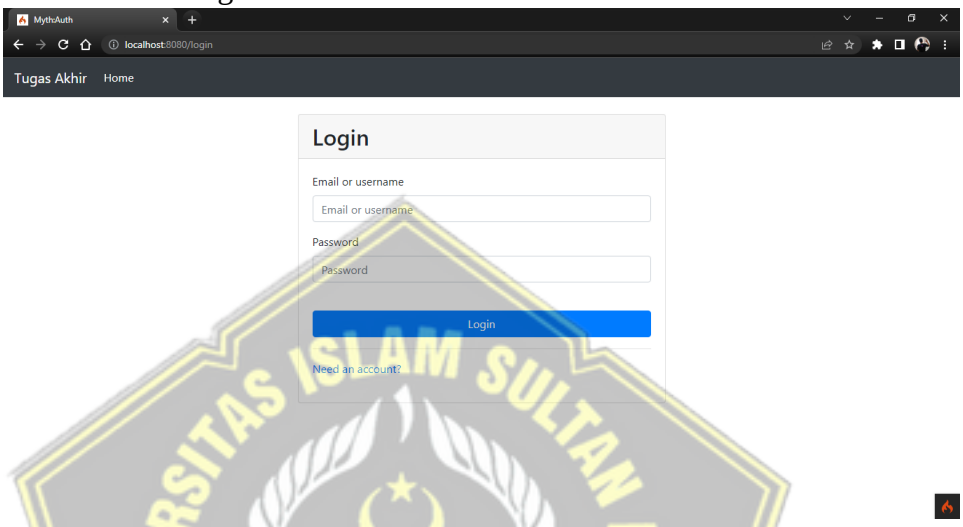
4. Halaman Hasil Cek Plagiarisme



Gambar 4. 8 Halaman Hasil Cek Plagiarisme

Gambar 4.8 merupakan tangkapan layar hasil cek *plagiarism*, di halaman ini berisi data nama *author*, *file* pdf yang di *upload* atau di cek , tahun dokumen tersebut rilis, dan hasil *similiarity* dalam bentuk persen.

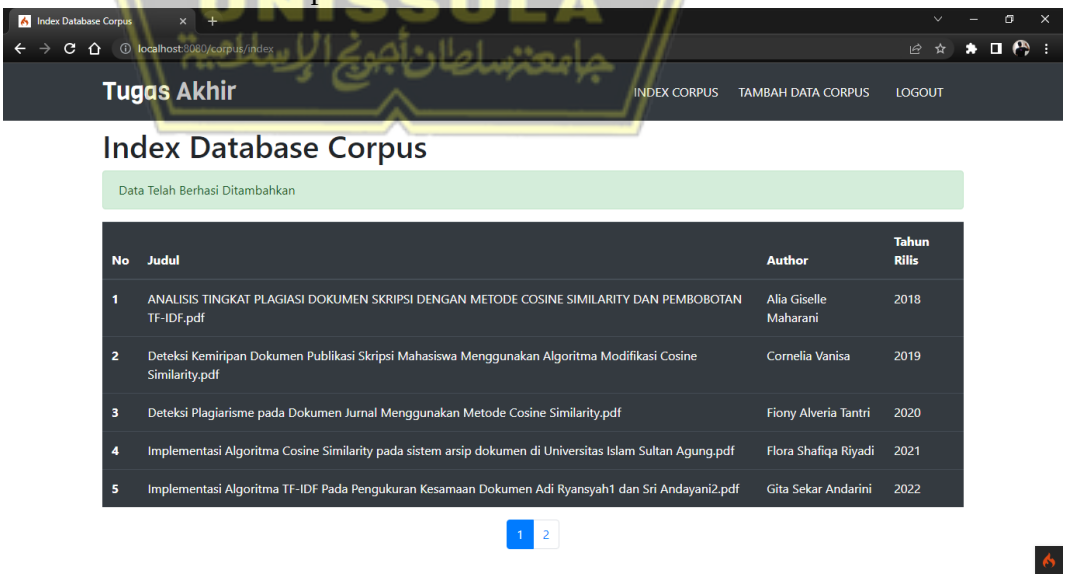
5. Halaman Login



Gambar 4. 9 Halaman *Login*

Gambar 4.9 merupakan halaman *login* , dimana pada halaman ini merupakan halaman login dimana untuk *login* sendiri menggunakan *username* atau e-mail dan *password*.

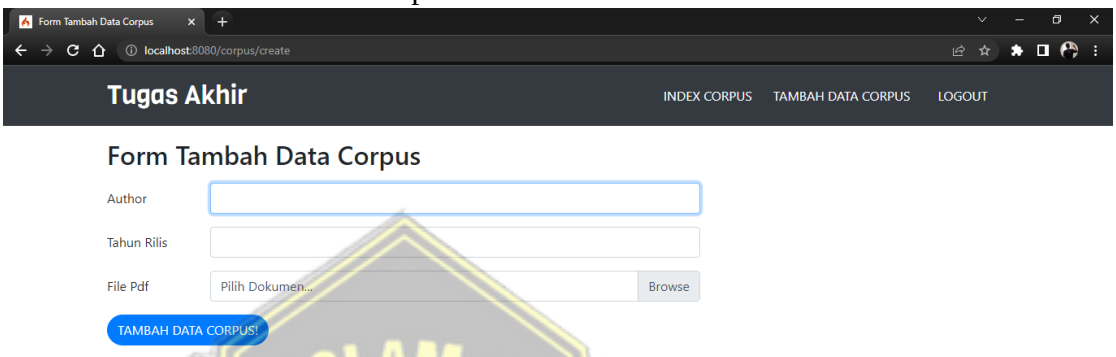
6. Halaman Index Corpus



Gambar 4. 10 Halaman *index corpus*

Gambar 4.10 merupakan tampilan layar yang ditampilkan ketika mengakses *index plagiarism* halaman ini berisi tentang judul , *author* dan tahun rilis dokumen yang di *upload* untuk sebagai dokumen acuan *corpus*.

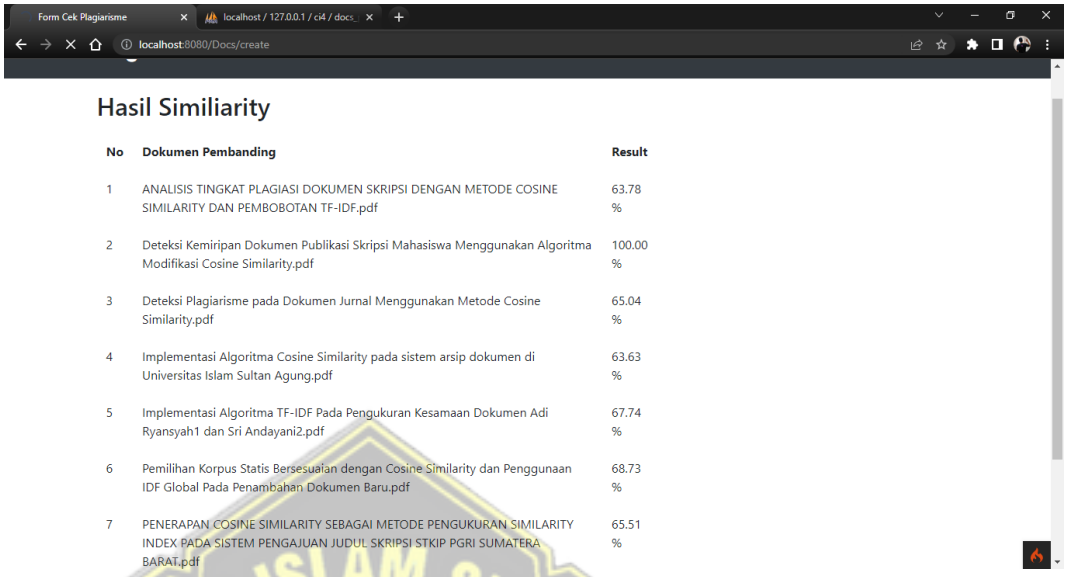
7. Halaman Tambah Data Corpus



Gambar 4. 11 halaman tambah data *corpus*

Gambar 4.11 merupakan sebuah tangkapan layar yang berisi mengenai *form* tambah data *corpus* yang nantinya akan diisi dengan *author* , tahun rilis, dan nama *file* pdf yang di *upload* , nantinya data yang dimasukkan pada *form* ini akan di jadikan sebagai bahan acuan *corpus*.

8. Halaman Hasil Cek Similiarity



Gambar 4. 12 Halaman Hasil *Cek Similiarity*

Gambar 4.12 Merupakan halaman hasil *cek similiarity* halaman ini akan langsung terbuka setelah mengisi *form* di gambar 4.8 dimana halaman tersebut merupakan halaman cek *similiarity* , kemudian akan tampil halaman seperti diatas, halaman ini berisi tentang judul dokumen yang ada di *corpus* beserta hasil similiaritynya dengan dokumen yang dibandingkan.

4.5.Hasil Pengujian sistem

Hasil pengujian sistem yang dilakukan pada sistem ini menggunakan pengujian kotak hitam (*black box testing*). Pengujian *black-box* adalah untuk *meng-input* dan menguji apakah fungsi-fungsi yang terdapat pada sistem yang sedang berjalan sudah sesuai dengan tujuan, dan tercermin dalam hasil keluaran. Adapun rencana yang akan dibuat yaitu *input*, hasil yang ingin di harapkan, *output* dan kesimpulan serta yang akan diuji adalah pengisian cek plagiarisme, *login* tambah data *corpus*. Berikut merupakan hasil dari pengujian :

1. Pengujian pengisian data form cek plagiarisme

Tabel 4. 1 Pengujian pengisian data form cek plagiarisme

Kasus dan Hasil Uji			
Input	Hasil yang diharapkan	*Output*	Kesimpulan
Memasukkan Author : Yustian Dikma Eka Putra Relase_year : 2022 file_name : Deteksi Plagiarisme Tugas Akhir Mahasiswa Dengan Menggunakan Metode Cosine Similiarity.php	Akan langsung berpindah menuju halaman hasil cek plagiarisme.	Begitu berpindah ke halaman hasil cek plagiarisme dan keluar hasilnya dalam persen.	Berhasil
Memasukkan Author : Yustian Dikma Eka Putra Relase_year : 2022 file_name : Deteksi Plagiarisme Tugas Akhir Mahasiswa Dengan Menggunakan Metode Cosine Similiarity.php	Akan langsung berpindah menuju halaman hasil cek plagiarisme.	Berpindah ke halaman *index* plagiarisme dan terdapat *warning* data gagal ditambahkan.	Berhasil
Memasukkan author: Relase_year :2022 file_name : Deteksi Plagiarisme Tugas Akhir Mahasiswa Dengan Menggunakan Metode Cosine Similiarity.php	Akan langsung berpindah menuju halaman hasil cek plagiarisme.	Tetap di halaman *form* pengisian dan menampilkan tulisan ‘Author harus diisi’.	Berhasil
Memasukkan Author : Yustian Dikma Eka Putra Relase_year : file_name : Deteksi Plagiarisme Tugas Akhir Mahasiswa Dengan Menggunakan	Akan langsung berpindah menuju halaman hasil cek plagiarisme.	Tetap di halaman *form* pengisian dan menampilkan tulisan ‘Tahun harus diisi ’.	Berhasil

Metode Cosine Similiarity.php			
Memasukkan author : Yustian Dikma Eka Putra Relase_year : 201q file_name : Deteksi Plagiarisme Tugas Akhir Mahasiswa Dengan Menggunakan Metode Cosine Similiarity.php	Akan langsung berpindah menuju halaman hasil cek plagiarisme.	Tetap di halaman *form* pengisian dan menampilkan tulisan 'Kolom Tahun Rilis Harus Diisi Angka'	Berhasil
Memasukkan author : Yustian Dikma Eka Putra Relase_year : 2022 file_name :	Akan langsung berpindah menuju halaman hasil cek plagiarisme.	Tetap di halaman *form* pengisian dan menampilkan tulisan 'Pilih File Dokumen Terlebih Dahulu'	Berhasil
Memasukkan author : Yustian Dikma Eka Putra Relase_year : 2022 file_name : Deteksi Plagiarisme Tugas Akhir Mahasiswa Dengan Menggunakan Metode Cosine Similiarity.php	Akan langsung berpindah menuju halaman hasil cek plagiarisme.	Tetap di halaman *form* pengisian dan menampilkan tulisan 'Ukuran File Terlalu Besar'	Berhasil
Memasukkan author : Yustian Dikma Eka Putra Relase_year : 2022 file_name : deteksi Plagiarisme tugas akhir mahasiswa menggunakan metode cosine smiliaririty.pdf	Akan langsung berpindah menuju halaman hasil cek plagiarisme.	Tetap di halaman *form* pengisian dan menampilkan tulisan 'File Yang Dipilih Bukan Berformat PDF'	Berhasil

2. Pengujian pengisian *form login*

Tabel 4. 2 Pengujian pengisian *form login*

Kasus dan Hasil Uji			
Input	Hasil yang diharapkan	*Output*	Kesimpulan
Memasukkan Username atau email : tiaaan Password : inazuma11	Akan langsung masuk ke menu *index corpus*	Dapat langsung berpindah ke *index corpus*	Berhasil
Memasukkan Username atau email : Password :	Akan langsung masuk ke menu *index corpus*	Menampilkan tulisan “The login field is required.”, “The password field is required.”	Berhasil
Memasukkan Username atau email : Password : inazuma11	Akan langsung masuk ke menu *index corpus*	Menampilkan tulisan “The login field is required.”	Berhasil
Memasukkan Username atau email : tiaaan Password :	Akan langsung masuk ke menu index corpus	Menampilkan tulisan “The password field is required.”	Berhasil

3. Pengujian pengisian data form tambah data corpus

Tabel 4. 3 Pengujian pengisian data form tambah data corpus

Kasus dan Hasil Uji			
Input	Hasil yang diharapkan	*Output*	Kesimpulan
Memasukkan author : Yustian Dikma Eka Putra Relase_year : 2022 file_name : Deteksi Plagiarisme Tugas Akhir Mahasiswa Dengan Menggunakan Metode Cosine Similiarity.php	Akan langsung berpindah menuju halaman hasil *index corpus*	Begitu berpindah ke halaman *index corpus* maka akan ada notifikasi ‘data berhasil ditambahkan’	Berhasil

Memasukkan author : Yustian Dikma Eka Putra Relase_year : 2022 file_name : Deteksi Plagiarisme Tugas Akhir Mahasiswa Dengan Menggunakan Metode Cosine Similiarity.php	Akan langsung berpindah menuju halaman hasil *index corpus*	Berpindah ke halaman tambah data *corpus* dan terdapat *warning* data gagal ditambahkan	Berhasil
Memasukkan author : Relase_year : 2022 file_name : Deteksi Plagiarisme Tugas Akhir Mahasiswa Dengan Menggunakan Metode Cosine Similiarity.php	Akan langsung berpindah menuju halaman hasil *index corpus*.	Tetap di halaman *form* pengisian dan menampilkan tulisan ‘Author harus diisi’	Berhasil
Memasukkan author : Yustian Dikma Eka Putra Relase_year : file_name : Deteksi Plagiarisme Tugas Akhir Mahasiswa Dengan Menggunakan Metode Cosine Similiarity.php	Akan langsung berpindah menuju halaman hasil *index corpus*.	Tetap di halaman *form* pengisian dan menampilkan tulisan ‘Tahun harus diisi ’	Berhasil
Memasukkan author : Yustian Dikma Eka Putra Relase_year : 201q file_name : Deteksi Plagiarisme Tugas Akhir Mahasiswa Dengan Menggunakan Metode Cosine Similiarity.php	Akan langsung berpindah menuju halaman hasil *index corpus*	Tetap di halaman *form* pengisian dan menampilkan tulisan ‘Kolom Tahun Rilis Harus Diisi Angka’	Berhasil

Memasukkan author : Yustian Dikma Eka Putra Relase_year : 2022 file_name :	Akan langsung berpindah menuju halaman hasil *index corpus.*	Tetap di halaman *form* pengisian dan menampilkan tulisan 'Pilih File Dokumen Terlebih Dahulu'	Berhasil
Memasukkan author : Yustian Dikma Eka Putra Relase_year : 2022 file_name : Deteksi Plagiarisme Tugas Akhir Mahasiswa Dengan Menggunakan Metode Cosine Similiarity.php	Akan langsung berpindah menuju halaman hasil *index corpus*	Tetap di halaman *form* pengisian dan menampilkan tulisan 'Ukuran File Terlalu Besar'	Berhasil
Memasukkan author : Yustian Dikma Eka Putra Relase_year : 2022 file_name : Deteksi Plagiarisme Tugas Akhir Mahasiswa Dengan Menggunakan Metode Cosine Similiarity.png	Akan langsung berpindah menuju halaman hasil *index corpus.*	Tetap di halaman *form* pengisian dan menampilkan tulisan 'File Yang Dipilih Bukan Berformat PDF'	Berhasil

4.6.Validasi Implementasi Algoritma

1. Konsistensi cosine

   Konsistensi metode *cosine similiarity* terbukti konsisten , dapat dibuktikan dari pengujian yang dilakukan selama 10x dengan dokumen terkait, hal ini menghasilkan angka yang sama di setiap prosesnya.

BAB V
KESIMPULAN DAN SARAN

5.1.Kesimpulan

Berdasarkan hasil dari penelitian ini, dapat diambil kesimpulan tentang deteksi plagiarisme tugas akhir mahasiswa dengan menggunakan metode *cosine similiarity* adalah sebagai berikut metode *cosine similiarity* bekerja dengan baik , hal ini terbukti dengan hasil yang akurat dan konsisten di setiap perhitungannya, dari hasil perhitungan tersebut semakin besar tingkat kemiripan sebuah dokumen semakin besar pula kemungkinan dokumen tersebut dianggap plagiat.

5.2.Saran

Saran yang dapat diterapkan guna untuk pengembang sistem yang lebih lanjut dimasa mendapatkan yaitu : Pada sistem deteksi plagiarisme tugas akhir mahasiswa dengan menggunakan metode *cosine similiarity* adalah setelah hasil didapatkan agar bisa menambahkan metode pengecualian untuk teks atau kalimat yang hampir mirip.

DAFTAR PUSTAKA

Ariantini, D. A. R., Lumenta, A. S. M., & Jacobus, A. (2016). Pengukuran Kemiripan Dokumen Teks Bahasa Indonesia Menggunakan Metode Cosine Similarity. *Jurnal Teknik Informatika*, *9*(1). https://doi.org/10.35793/jti.9.1.2016.13752

Feldman, R., & Sanger, J. (2007). *The text mining handbook: advanced approaches in analyzing unstructured data*. Cambridge university press.

Fitri, R., & Asyikin, A. N. (2015). Aplikasi penilaian ujian essay otomatis menggunakan metode cosine similarity. *Poros Teknik*, *7*(2), 88–94.

Hassani, H., Beneki, C., Unger, S., Mazinani, M. T., & Yeganegi, M. R. (2020). Text mining in big data analytics. *Big Data and Cognitive Computing*, *4*(1), 1.

Imran, H., & Sharan, A. (2009). Thesaurus and query expansion. *International journal of computer science & information Technology (IJCSIT)*, *1*(2), 89–97.

Ivory, M. Y. (2004). Website usability engineering. In *The Practical Handbook of Internet Computing* (hal. 44-1-44–17). CRC Press. https://doi.org/10.1201/9780203507223

Jereb, E., Perc, M., Lämmlein, B., Jerebic, J., Urh, M., Podbregar, I., & Šprajc, P. (2018). Factors influencing plagiarism in higher education: A comparison of German and Slovene students. *PloS one*, *13*(8), e0202252.

Li, B., & Han, L. (2013). Distance weighted cosine similarity measure for text classification. *International conference on intelligent data engineering and automated learning*, 611–618.

Maurer, H., Kappe, F., & Zaka, B. (2006). Plagiarism - A survey. *Journal of Universal Computer Science*, *12*(8), 1050–1084.

Muflikhah, L., & Baharudin, B. (2009). Document clustering using concept space and cosine similarity measurement. *2009 International Conference on Computer Technology and Development*, *1*, 58–62.

Naf'an, M. Z., Burhanuddin, A., & Riyani, A. (2019). Penerapan Cosine Similarity dan Pembobotan TF-IDF untuk Mendeteksi Kemiripan Dokumen. *Jurnal Linguistik Komputasional*, *2*(1), 23–27.

Nugroho, E. (2011). Perancangan Sistem Deteksi Plagiarisme Dokumen Teks Dengan Menggunakan Algoritma Rabin-Karp. In *Journal of Strategic Studies*. https://doi.org/10.1080/01402390.2011.569130

Permadi, T. (2006). Teks, Tekstologi, dan Kritik Teks. *Bandung: Universitas Pendidikan Indo-nesia*.

Polettini, N. (2004). The vector space model in information retrieval-term weighting problem. *Entropy*, *34*, 1–9.

Prihantini, F. N., & Indudewi, D. (2017). Kesadaran dan Perilaku Plagiarisme dikalangan Mahasiswa. *Jurnal Dinamika Sosial Budaya*, *Prihantini, F. N. and Indudewi, D. (2017) 'Kesadaran dan Perilaku Plagiarisme dikalangan Mahasiswa (Studi pada Mahasiswa Fakultas Ekonomi Jurusan Akuntansi Universitas Semarang),' Jurnal Dinamika Sosial Budaya. doi: 10.26623/jdsb.v18i1.559*. https://doi.org/10.26623/jdsb.v18i1.559

Purba, A. H., & Situmorang, Z. (2017). Analisis perbandingan algoritma rabin-karp dan levenshtein distance dalam menghitung kemiripan teks. *Jurnal Teknik Informatika UNIKA Santo Thomas*, 24–32.

Rahayu, S., & RMS, A. S. (2018). Penerapan Metode Naive Bayes Dalam Pemilihan Kualitas Jenis Rumput Taman CV. Rumput Kita Landscape. *Digital Zone: Jurnal Teknologi Informasi dan Komunikasi*, *9*(2), 162–171.

Ramadhany, T. (2008). Implementasi Kombinasi Model Ruang Vektor dan Model Probabilistik Pada Sistem Temu Balik Informasi. *Skripsi Terpublikasi. Bandung: Institut Teknologi Bandung*.

Salton, G., & McGill, M. J. (1983). *Introduction to modern information retrieval*. mcgraw-hill.

Sejati, F. B., Hendradi, P., & Pujiarto, B. (2018). Deteksi Plagiarisme Karya Ilmiah dengan Pemanfaatan Daftar Pustaka dalam Pencarian Kemiripan Tema Menggunakan Metode Cosine Similarity (Studi Kasus: Di Universitas Muhammadiyah Magelang). *Jurnal Komtika (Komputasi dan Informatika)*, *2*(2), 85–94.

Sufanti, M. (2013). *Pembelajaran bahasa indonesia berbasis teks: belajar dari ohio amerika serikat*.

Sugiyamto, S., Surarso, B., & Sugiharto, A. (2014). Analisa Performa Metode Cosine Dan Jacard Pada Pengujian Kesamaan Dokumen. *Jurnal Masyarakat Informatika*, *5*(10). https://doi.org/10.14710/jmasif.5.10.1-8

Tan, A.-H. (1999). Text mining: The state of the art and the challenges. *Proceedings of the pakdd 1999 workshop on knowledge disocovery from advanced databases*, *8*, 65–70.

Wahyuni, R. T., Prastiyanto, D., & Supraptono, E. (2017). Penerapan Algoritma Cosine Similarity dan Pembobotan TF-IDF pada Sistem Klasifikasi Dokumen Skripsi. *Jurnal Teknik Elektro*, *9*(1), 18–23.

Wibowo, A. (2012). Mencegah dan Menanggulangi Plagiarisme di Dunia Pendidikan. *Kesmas: National Public Health Journal*. https://doi.org/10.21109/kesmas.v6i5.84

Yamamoto, M., & Church, K. W. (2001). Using suffix arrays to compute term frequency and document frequency for all substrings in a corpus. *Computational Linguistics*, *27*(1), 1–30.