

**RANCANG BANGUN APLIKASI PENDETEKSI
KESAMAAN DOKUMEN MENGGUNAKAN METODE
*DICE SIMILARITY***

SKRIPSI



ARIS AKHYAR ABDILLAH

H071171505

PROGRAM STUDI SISTEM INFORMASI

DEPARTEMEN MATEMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS HASANUDDIN

MAKASSAR

APRIL 2023

**RANCANG BANGUN APLIKASI PENDETEKSI
KESAMAAN DOKUMEN MENGGUNAKAN METODE
*DICE SIMILARITY***

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana pada Program
Studi Sistem Informasi Departemen Matematika Fakultas Matematika dan Ilmu
Pengetahuan Alam Universitas Hasanuddin

ARIS AKHYAR ABDILLAH

H071171505

**PROGRAM STUDI SISTEM INFORMASI
DEPARTEMEN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS HASANUDDIN
MAKASSAR
APRIL 2023**

LEMBAR PERNYATAAN KEOTENTIKAN

Saya yang bertanda tangan di bawah ini menyatakan dengan sungguh – sungguh
bahwa skripsi yang saya buat dengan judul :

**RANCANG BANGUN APLIKASI PENDETEKSI KESAMAAN
DOKUMEN MENGGUNAKAN METODE *DICE SIMILARITY***

Adalah benar hasil karya saya sendiri, bukan hasil plagiat dan belum pernah
dipublikasikan dalam bentuk apa pun

Makassar, ... April 2023

Aris Akhyar Abdillah

NIM. H071171505

**RANCANG BANGUN APLIKASI PENDETEKSI KESAMAAN
DOKUMEN MENGGUNAKAN METODE *DICE SIMILARITY***

Disetujui Oleh :

Pembimbing Utama,

Pembimbing Pertama

Dr. Hendra S.Si., M. Kom.
NIP. 19760102 200312 1 001

Edy Saputra R, S. Si., M. Si.
NIP. 19910410 202005 3 001

Ketua Program Studi

Dr. Hendra S.Si., M. Kom.
NIP. 19760102 200312 1 001

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama : Aris Akhyar Abdillah

NIM : H071171505

Program Studi : Sistem Informasi

Judul Skripsi : Rancang Bangun Aplikasi Pendeteksi Kesamaan Dokumen
Menggunakan Metode *Dice Similarity*

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Sains pada Program Studi Sistem Informasi Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin

DEWAN PENGUJI

1. Ketua : Dr. Hendra, S.Si, M.Kom. (.....)

2. Sekretaris : Edy Saputra Rusdi, S.Si, M.Si (.....)

3. Anggota : Dr. Muhammad Hasbi, M.Sc (.....)

4. Anggota : Ir. Eliyah Acantha Manapa Sampetoding, (.....)

S.Kom, M.Kom.

Ditetapkan di : Makassar

Tanggal : ... April 2023

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Segala puji bagi Allah Subhanallahu Wa Ta'ala atas segala nikmat serta limpahan rahmat dan hidayah-Nya yang telah diberikan kepada penulis sampai dapat menuntaskan penulisan skripsi ini. Shalawat dan salam serta doa senantiasa terlimpahkan kepada baginda Nabi besar, Nabi Muhammad Shallallahu 'Alaihi Wa sallam. Dengan izin dan kemurahan hatinya yang diberikan oleh Allah Subhanallahu Wa Ta'ala, penulis dapat menyelesaikan skripsi yang berjudul **“Rancang Bangun Aplikasi Pendeteksi Kesamaan Dokumen Menggunakan Metode Dice Similarity”** sebagai salah satu syarat memperoleh gelar sarjana pada Program Studi Sistem Informasi Departemen Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin.

Dalam menyelesaikan skripsi ini, tentunya penulis mengalami banyak tantangan dan permasalahan selama proses penulisan skripsi ini. Hanya dengan karunia dan kehendak-Nya serta bimbingan dan pertolongan dari berbagai pihak baik secara langsung maupun tidak langsung, maka skripsi ini dapat diselesaikan. Untuk itu, penulis mengucapkan terima kasih yang sebesar-besarnya dan penghormatan yang setinggi-tingginya kepada berbagai pihak terkhusus kepada kedua orang tua penulis yaitu Ayahanda **Arham Aras** dan Ibunda **Ernawati Haruna** dan juga saudara satu-satunya yaitu **Nur Izzah Khairani**. Penghargaan yang tulus dan ucapan terima kasih dengan penuh keikhlasan juga penulis ucapkan kepada:

1. **Bapak Prof. Dr. Ir. Jamaluddin Jompa, M.Sc.**, selaku Rektor Universitas Hasanuddin beserta seluruh jajarannya.
2. **Bapak Dr. Eng. Amiruddin, S.Si., M.Si.**, yang menjabat sebagai Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin beserta seluruh jajarannya.
3. **Bapak Prof. Dr. Nurdin, S.Si., M.Si.**, yang menjabat sebagai Ketua Departemen Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin beserta seluruh jajarannya.

4. **Bapak Dr. Hendra S.Si., M. Kom.**, yang menjabat sebagai Ketua Program Studi Sistem Informasi dan juga sebagai pembimbing utama saya yang telah memberikan saya arahan, pengetahuan, motivasi dan bimbingan dengan sabar dan penuh perhatian di tengah kesibukannya.
5. **Bapak Edy Saputra R, S. Si., M. Si.**, yang menjabat sebagai Penasihat Akademik dan juga sebagai pembimbing pertama saya yang telah memberikan saya arahan, pengetahuan, motivasi dan bimbingan dengan sabar dan penuh perhatian ditengah kesibukannya.
6. **Bapak Dr. Muhammad Hasbi, M.Sc.**, dan **Bapak Ir. Eliyah Acantha Manapa Sampetoding, S. Kom, M. Kom.**, yang menjabat sebagai tim penguji yang telah memberikan saya saran dan kritikan yang konstruktif dalam penyempurnaan penyusunan tugas akhir ini.
7. **Bapak/Ibu Dosen** Departemen Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin yang telah memberikan saya ilmu dan pengetahuan selama saya belajar di sini.
8. Seluruh warga **KM FMIPA Unhas** dan **Himatika FMIPA Unhas** yang telah menjadi keluarga kedua saya yang memberikan saya kenangan, serta pelajaran hidup yang sangat berharga. Khususnya, seluruh saudara-saudara **DISKRIT 2017** dan **MIPA 2017**, terima kasih atas kebersamaan, loyalitas, pengalaman, dan semangat yang sangat berarti bagi saya.
9. Teman-teman **Sistem Informasi 2017** yang menjadi teman pertama dikampus yang telah memberi dukungan dan kerja samanya dalam menemani saya selama perkuliahan, yang banyak membantu saya dalam hal apa pun serta berbagi suka-duka dalam menuntut ilmu.
10. Kepada seluruh pihak yang tidak dapat saya sebutkan satu persatu, terima kasih sebesar-besarnya untuk segala dukungan dan partisipasi yang diberikan kepada saya semoga bernilai ibadah di sisi Allah Subhanahu Wa Ta'ala.

Penulis mengakui bahwa skripsi ini masih memiliki banyak kekurangan, untuk itu dengan segala kerendahan hati penulis meminta maaf atas kekurangan tersebut. Penutup, semoga tulisan ini dapat memberikan banyak manfaat untuk pembaca. *Wassalamu'alaikum Warahmatullahi Wabarakatuh.*

**PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR
UNTUK KEPENTINGAN AKADEMIK**

Sebagai sivitas akademik Universitas Hasanuddin, saya yang bertanda tangan di bawah ini :

Nama : Aris Akhyar Abdillah
NIM : H071171505
Program Studi : Sistem Informasi
Departemen : Matematika
Fakultas : Matematika dan Ilmu Pengetahuan Alam
Jenis Karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Hasanuddin **Hak Bebas Royalti Noneklusif** (*Non-exclusive Royalty-Free Right*) atas karya ilmiah yang berjudul:

**“Rancang Bangun Aplikasi Pendeteksi Kesamaan Dokumen
Menggunakan Metode *Dice Similarity*”**

Beserta perangkat yang ada (jika diperlukan). Terkait dengan hal di atas, maka pihak universitas berhak menyimpan, mengalih-media/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di Makassar Pada tanggal ... April 2023

Yang Menyatakan
(Aris Akhyar Abdillah)

ABSTRAK

Dengan adanya teknologi di saat ini, dapat dikatakan pekerjaan manusia semakin mudah. Seperti untuk akses informasi, sekarang sudah banyak sekali informasi yang tersebar di internet dan hampir semua yang dicari dapat ditemukan. Dengan semua kemudahan yang diberikan teknologi seperti contoh di atas, timbul suatu masalah yaitu mudahnya dilakukan plagiarisme. Plagiarisme berasal dari bahasa Latin yaitu “plagiare” yang berarti mencuri. Plagiarisme berasal dari kata plagiat yang berarti pengambilan karangan (pendapat dan sebagainya) orang lain dan menjadikannya seolah-olah karangan (pendapat dan sebagainya) sendiri, misalnya menerbitkan karya tulis orang lain atas nama dirinya sendiri. Untuk mengurangi masalah tersebut dibuatlah Aplikasi Pendeteksi Kesamaan Dokumen Menggunakan Metode *Dice Similarity*. Aplikasi ini dibangun dengan menggunakan metode *Prototype* yaitu salah satu metode untuk merancang aplikasi di mana aplikasi di analisa kebutuhannya baik secara kebutuhan fungsional dan non fungsional, kemudian merancang aplikasi tersebut kemudian diuji. Aplikasi ini juga diuji menggunakan metode *blackbox* di mana aplikasi diuji berdasarkan sudut pandang pengguna. Selain itu, aplikasi ini menggunakan arsitektur *Model View Controller* (MVC) yang mana mempartisi kode aplikasi ini menjadi beberapa bagian. Aplikasi ini telah berhasil diuji dengan hasil yang sesuai dengan pengujian.

Kata Kunci : aplikasi, plagiarisme, *prototype*, MVC, *blackbox*,

ABSTRACT

With the existence of technology in the present, it can be said that human work is easier. For example, for access to information, there are now a lot of information scattered on the internet and almost everything that is searched can be found. With all the conveniences provided by technology such as the example above, a problem arises, namely the ease of plagiarism. Plagiarism comes from Latin, namely “plagiare” which means stealing. Plagiarism comes from the word plagiarism which means taking someone else’s work (opinions and so on) and making it as if it were one’s own work (opinions and so on), for example publishing someone else’s written work in one’s own name. To reduce this problem, an Application for Detecting Document Similarity Using the Dice Similarity Method was created. This application is built using the Prototype method which is one of the methods for designing applications where the application needs are analyzed both in terms of functional and non-functional requirements, then designing the application and then testing it. This application is also tested using the blackbox method where the application is tested based on the user’s point of view. In addition, this application uses the Model View Controller (MVC) architecture which partitions the application code into several parts. This application has been successfully tested with results that match the testing.

Keywords : application, plagiarism, prototype, MVC, blackbox

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN SAMPUL	ii
LEMBAR PERNYATAAN KEOTENTIKAN.....	iii
HALAMAN PERSETUJUAN PEMBIMBING	iv
HALAMAN PENGESAHAN.....	v
KATA PENGANTAR	vi
PERSETUJUAN PUBLIKASI TUGAS AKHIR	viii
ABSTRAK	ix
ABSTRACT.....	x
DAFTAR ISI.....	xi
DAFTAR GAMBAR	xiii
DAFTAR TABEL.....	xiv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan Penelitian	3
1.4 Manfaat Penelitian	3
1.5 Batasan Masalah	3
BAB II TINJAUAN PUSTAKA.....	4
2.1 Penelitian Terkait.....	4
2.2 Landasan Teori	6
2.2.1 <i>Website</i>	6
2.2.2 <i>Hypertext Transfer Protocol Secure (HTTPS)</i>	6
2.2.3 Javascript	7
2.2.3.1 Karakteristik Javascript.....	7
2.2.3.2 NodeJS	9
2.2.3.3 ExpressJS	10
2.2.4 <i>Database</i>	10
2.2.4.1 SQL dan NoSQL	10
2.2.4.2 MongoDB.....	11
2.2.5 Jurnal.....	11
2.2.6 Plagiarisme.....	12
2.2.7 <i>Dice Similarity</i>	13
2.2.7.1 Implementasi <i>Dice Similarity</i>	14
BAB III METODE PENELITIAN.....	18

3.1 Prosedur Penelitian.....	18
3.2 <i>Timeline</i> Penelitian.....	18
3.3 Rancangan Sistem	19
3.3.1 Rancangan Antar Muka (<i>Interface</i>).....	19
3.4 Metode Penelitian.....	21
3.4.1 Metode Prototype.....	21
3.5 Instrumen Penelitian.....	22
BAB IV HASIL DAN PEMBAHASAN	23
4.1. <i>Requirements Gatehering and Analysis</i> (Analisis Kebutuhan).....	23
4.2 <i>Quick Design</i> (Desain Cepat).....	24
4.2.1 <i>Use Case Diagram</i>	24
4.2.2 <i>Activity Diagram</i>	25
4.2.3 Entity Relationship Diagram (ERD).....	26
4.2.4 Tampilan Antar Muka (<i>Interface</i>).....	27
4.2.5 Desain Sistem	31
4.2.5.1 Algoritma Input Dokumen ke dalam <i>Database</i>	31
4.2.5.2 Algoritma Perbandingan Dokumen.....	32
4.2.5.3 Algoritma <i>Dice Similarity</i>	34
4.3 <i>Build Prototype</i> (Bangun Prototipe).....	37
4.3.1 Model View Controller (MVC)	37
4.3.2 Implementasi Pembuatan Sistem.....	37
4.3.3 Sintaks <i>Dice Similarity</i>	44
4.4 <i>User Evaluation</i> (Evaluasi Pengguna).....	48
4.4.1 Rencana Pengujian.....	48
4.4.2 Tahap Pengujian	51
4.5 <i>Refining Prototype</i> (Memperbaiki Prototipe).....	58
4.6 <i>Implement Product and Maintain</i> (Implementasi dan Pemeliharaan).....	58
BAB V KESIMPULAN DAN SARAN.....	59
5.1 Kesimpulan.....	59
5.2 Saran.....	59
DAFTAR PUSTAKA	60
LAMPIRAN	62

DAFTAR GAMBAR

Gambar 2.1 Contoh Pembuatan Variabel.....	8
Gambar 2.2 Contoh Penggunaan Tipe Data.....	8
Gambar 2.3 Contoh Penggunaan <i>Synchronous</i> dan <i>Asynchronous</i>	9
Gambar 2.4 Contoh Code MongoDB	11
Gambar 3.1 Rancangan Antar Muka Halaman Utama.....	19
Gambar 3.2 Rancangan Antar Muka Halaman <i>Login</i>	20
Gambar 3.3 Rancangan Antar Muka Halaman Beranda	20
Gambar 3.4 Rancangan Antar Muka Halaman <i>Admin</i>	21
Gambar 4.1 <i>Use Case Diagram</i>	25
Gambar 4.2 <i>Activity Diagram</i>	26
Gambar 4.3 <i>Entity Relation Diagram</i>	27
Gambar 4.4 Tampilan Halaman Utama	28
Gambar 4.5 Tampilan Halaman <i>Login</i>	28
Gambar 4.6 Tampilan Halaman Beranda	29
Gambar 4.7 Tampilan Halaman Admin	30
Gambar 4.8 Algoritma <i>Input</i> Dokumen ke dalam <i>Database</i>	31
Gambar 4.9 Algoritma Perbandingan Dokumen.....	32
Gambar 4.10 Algoritma <i>Dice Similarity</i>	35
Gambar 4.11 Contoh Sintaks <i>Model</i>	38
Gambar 4.12 Contoh Sintaks <i>View</i>	39
Gambar 4.13 Contoh Sintaks <i>Controller</i>	41
Gambar 4.14 Contoh Sintaks <i>Middleware</i>	42
Gambar 4.15 Contoh Sintaks <i>Routes</i>	43
Gambar 4.16 Contoh Sintaks <i>Utils</i>	44
Gambar 4.17 Sintaks <i>Input</i> Dokumen ke Dalam <i>Database</i>	45
Gambar 4.18 Sintaks <i>Dice Similarity</i>	46
Gambar 4.19 Pengujian Langsung Menggunakan Jurnal	57

DAFTAR TABEL

Tabel 2.1 Perbandingan Data X dan Data Y	15
Tabel 2.2 Kamus Data.....	15
Tabel 2.3 Perhitungan Dalam Kamus Data.....	16
Tabel 3.1 <i>Timeline</i> Penelitian.....	18
Tabel 4.1 Pengujian Pengguna <i>Guest</i>	48
Tabel 4.2 Pengujian Pengguna <i>User</i>	49
Tabel 4.3 Pengujian Pengguna Admin.....	50
Tabel 4.4 Hasil Pengujian Pengguna <i>Guest</i>	52
Tabel 4.5 Hasil Pengujian Pengguna <i>User</i>	52
Tabel 4.6 Hasil Pengujian Pengguna <i>Admin</i>	54

BAB I PENDAHULUAN

1.1 Latar Belakang

Teknologi merupakan sarana untuk menyediakan barang-barang yang diperlukan untuk mempermudah kegiatan atau kehidupan manusia. Penggunaan teknologi oleh manusia dimulai dengan pengubahan sumber daya alam menjadi alat-alat sederhana. Penemuan di masa lampau tentang kemampuan mengendalikan api telah menaikkan ketersediaan sumber-sumber pangan, sedangkan penciptaan roda telah membantu manusia dalam bepergian dari satu tempat ke tempat lainnya. Perkembangan teknologi terbaru dimasa sekarang seperti mesin cetak, telepon, dan internet telah mengubah manusia untuk bekerja maupun berinteraksi dalam menjalankan kehidupannya.

Dengan adanya teknologi di saat ini, dapat dikatakan pekerjaan manusia semakin mudah. Seperti untuk akses informasi, sekarang sudah banyak sekali informasi yang tersebar di internet dan hampir semua yang dicari dapat ditemukan. Tidak seperti dahulu, untuk mengakses informasi, orang biasa pergi ke perpustakaan untuk mendapatkan informasi yang diinginkan. Hal itu memungkinkan karena adanya digitalisasi dokumen yang dulunya hanya berupa *hardcopy* (salinan cetak) menjadi *softcopy* (salinan digital). Dengan hal tersebut kita bisa mengakses semua dokumen kapan saja dan di mana saja. Dengan semua kemudahan yang diberikan teknologi seperti contoh di atas, timbul suatu masalah yaitu mudahnya dilakukan plagiarisme yang merupakan suatu tindakan yang tidak baik.

Plagiarisme berasal dari bahasa Latin yaitu "*plagiare*" yang berarti mencuri. Plagiarisme berasal dari kata plagiat yang berarti pengambilan karangan (pendapat dan sebagainya) orang lain dan menjadikannya seolah-olah karangan (pendapat dan sebagainya) sendiri, misalnya menerbitkan karya tulis orang lain atas nama dirinya sendiri. Sehingga dapat diartikan plagiarisme merupakan tindakan mencuri gagasan hasil penelitian orang lain, untuk kemudian disajikan seolah-olah milik sendiri (Ridhatillah, 2003). Plagiarisme yang marak dilakukan di kalangan mahasiswa membuat mahasiswa menjadi malas berpikir dan mengembangkan kemampuan

sebagai kaum intelektual. Moral mahasiswa akan luntur karena dengan melakukan plagiarisme pemikiran mereka tidak dapat berkembang dengan maksimal. Sebab mahasiswa cenderung mencari kemudahan dengan mengambil karya orang lain dan mengakui sebagai karya pribadi (Arista & Listyani, 2015). Tindakan plagiarisme merupakan salah satu tindakan yang melanggar hak cipta. Hak Cipta itu sendiri merupakan hak eksklusif untuk Pencipta ataupun penerima hak buat mengumumkan ataupun perbanyak Ciptaannya ataupun membagikan izin buat itu dengan tidak kurangi pembatasan- pembatasan bagi peraturan perundang-undangan yang berlaku. Jika terjadi pelanggaran tersebut, dapat dikenai pelanggaran hak cipta di Pasal 72 ayat UUHC dengan dipidana dengan pidana penjara pendek selama 1 bulan serta / ataupun denda sangat sedikitnya Rp1.000.000,00, ataupun pidana penjara lama 7 tahun serta / ataupun denda sebanyak Rp5.000.000.000,00 (Lopes, 2013)

Untuk mengurangi hal tersebut, dapat dilakukan beberapa metode untuk mencegah plagiarisme seperti menggunakan cara manual ataupun menggunakan teknologi yang ada. Jika menggunakan cara manual, dilakukan dengan membandingkan sebuah dokumen yang akan dicek dengan dokumen yang akan dijadikan perbandingan dengan melihat kata yang digunakan ataupun tata penulisan dokumen tersebut. Cara ini kurang efektif karena jumlah dokumen yang akan dijadikan perbandingan jumlah tidak sedikit bisa mencapai ratusan bahkan ribuan dokumen lebih. Sedangkan untuk cepat menggunakan mesin sebagai pembanding.

Ada beberapa metode yang dapat digunakan seperti *Cosine Similarity*, TF-IDF, *Jaccard Similarity*, *Dice Similarity*, *Word2Vec* dan sebagainya. Dengan menggunakan salah satu metode di atas, dapat temukan persentase kesamaan sebuah dokumen. Salah satu metode yang dapat digunakan yaitu dengan metode *Dice Similarity*. *Dice Similarity* merupakan salah satu metode perbandingan antara dokumen dengan membandingkan kata yang ada di antara dokumen tersebut. Melihat permasalahan di atas, maka disusunlah tugas akhir ini dengan judul “RANCANG BANGUN APLIKASI PENDETEKSI KESAMAAN DOKUMEN MENGGUNAKAN METODE DICE SIMILARITY”. Dengan adanya penelitian ini, diharapkan dapat membangun sebuah aplikasi Web yang mampu memberikan persentase kesamaan dari sebuah dokumen terhadap dokumen lainnya yang

terdapat di dalam *database* guna mengurangi tindak plagiarisme yang sering terjadi saat ini.

1.2 Rumusan Masalah

Rumusan masalah dalam penelitian ini yaitu bagaimana cara merancang sebuah aplikasi pendeteksi kesamaan teks berbasis *website* dengan mengimplementasikan metode *Dice Similarity* serta bagaimana cara menguji aplikasi tersebut menggunakan sebuah dokumen ?

1.3 Tujuan Penelitian

Tujuan penelitian dalam penelitian yaitu merancang sebuah aplikasi pendeteksi kesamaan teks berbasis *website* dengan mengimplementasikan metode *Dice Similarity* serta menguji aplikasi tersebut menggunakan sebuah dokumen.

1.4 Manfaat Penelitian

Penelitian ini merupakan pengimplementasian metode *Dice Similarity* pada sebuah aplikasi berbasis web guna mengetahui persentase kesamaan sebuah dokumen dengan dokumen lainnya sehingga dapat digunakan untuk mengurangi tindak plagiarisme dengan mudah.

1.5 Batasan Masalah

Adapun beberapa batasan masalah pada penelitian ini yaitu :

1. Sistem ini dibuat menggunakan bahasa pemrograman Javascript dengan *runtime environment* Nodejs, *framework* Express JS, dengan database MongoDB.
2. Menggunakan data berupa jurnal yang diinput oleh user dan memiliki ekstensi pdf (*Portable Document Format*).

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Dalam Penelitian Mufti Ari Bianto, dkk (2018) dengan judul *Perancangan Sistem Pendeteksi Plagiarisme Terhadap Topik Penelitian Menggunakan Metode K-Means Clustering dan Model Bayesian*, hasil kemiripan diuji berdasarkan kesamaan pola kata dalam kalimat, dokumen dengan nilai *similarity* tertinggi diperoleh satu dokumen dengan persentase kemiripan sebanyak 100 %, dan jika berdasarkan kesamaan *term*, berpengaruh terhadap hasil dokumen mirip yang dihasilkan, sehingga diperoleh dua dokumen mirip. Meskipun uji berdasarkan kesamaan *term*, menghasilkan dokumen mirip yang lebih banyak namun belum cukup akurat menunjukkan adanya plagiasi, karena dalam menentukan plagiasi kesamaan rangkaian kalimat merupakan hal yang penting untuk diperhatikan (Bianto, Rahayu, Huda, & Kusriani, 2018).

Pada penelitian Joko Priambodo (2018) dengan judul *Pendeteksian Plagiarisme Menggunakan Algoritma Rabin - Karb dengan Meotde Rolling Hash*, pendeteksian plagiarisme menggunakan algoritma *Rabin-Karp* dengan metode *rolling hash* dari hasil pengujian 30 dokumen teks yang menghasilkan tingkat akurasi yang terbesar yaitu 47.58 %. Hasil persentase tersebut termasuk dalam kategori tingkat plagiat 15 - 50 %, berarti menandakan dokumen tersebut termasuk plagiat tingkat sedang. Sedangkan tingkat akurasi yang terkecil yaitu 19.28 %, berarti menandakan dokumen tersebut termasuk plagiat tingkat sedang. Selain itu, berdasarkan analisis proses pendeteksian tingkat plagiarisme menggunakan algoritma *Rabin-Karp* dengan metode *Rolling Hash* bisa membaca karakter berupa huruf, simbol seperti titik (.), koma (,), dan lain-lain (Priambodo, 2018).

Kemudian dipenelitian Pavel Stefanovic (2019) dengan judul *The N-Grams Based Text Similarity Detection Approach Using Self-Organizing Maps and Similarity Measures*, dengan pendekatan berdasarkan teks yang dipecah menjadi n-gram dan mengevaluasinya menggunakan *Self-Organizing Maps* (SOM) dan *Similarity Measure*. Deteksi teks serupa dilakukan dalam tiga langkah: (1) konversi kumpulan data teks ke numerik ekspresi menggunakan n-gram; (2) perhitungan

ukuran kesamaan; (3) visualisasi dataset teks menggunakan SOM dan representasi kesamaan di atasnya. Pada langkah pertama, fokus utamanya adalah membuat grup n -gram dari semua dataset. Berbagai jumlah kata dalam n -gram dianalisis. Selain itu, filter yang berbeda diterapkan seperti penghapusan angka dan tanda baca, frekuensi kata, transformasi huruf besar, *stemming* algoritma, dll. Analisis menunjukkan filter dan ukuran n -gram mempengaruhi hasil akhir. Untuk ini dataset, ukuran n -gram dipilih dan sama dengan tiga untuk penyelidikan eksperimental. Pada langkah kedua, empat ukuran kesamaan dihitung: *Cosinus*, *Dice*, *Extends Jaccard*, dan *Overlap*. Hasil akhir menunjukkan bahwa persentase kemiripan tertinggi diperoleh dengan menggunakan *overlap*. Tiga nilai ukuran lainnya selalu sama dan lebih kecil. Penggunaan SOM menunjukkan bahwa SOM membantu untuk melihat hasil ringkasan kesamaan semua teks dalam bentuk visual dengan cepat. Dia sangat mudah untuk memahami teks mana yang mirip satu sama lain atau tidak. Dalam kasus kumpulan data yang dianalisis, SOM membantu mendeteksi kesamaan, dan *cluster* yang terbentuk dikorelasikan dengan kategori yang diberikan deskripsi kumpulan data (Stefanovi, Kurasova, & Štrimaitis, 2019).

Sedangkan, pada penelitian Uswatun Hasanah, dkk (2019) dengan judul *Perbandingan Metode Cosine Similarity dan Jaccard Similarity Untuk Penilaian Otomatis Jawaban Pendek*, dinilai belum mampu memberikan jawaban yang memuaskan i dikarenakan kedua metode hanya menilai kemiripan berdasarkan susunan leksikalnya. Sementara itu, jawaban mahasiswa juga sangat bervariasi dan menggunakan kata-kata yang jauh berbeda dari jawaban kunci, walaupun pada dasarnya memiliki makna semantik yang sama. Oleh karena itu, pada penelitian selanjutnya diperlukan metode lain yang mampu menangani makna semantik pada jawaban. Bagaimanapun, metode *Cosine Similarity* dan *Jaccard Similarity* masih dapat dipertimbangkan untuk menilai jawaban pendek secara otomatis, dengan batasan bahwa pertanyaan yang digunakan mengharuskan jawaban dalam format *keyword* sehingga tidak memunculkan kata-kata lain yang mampu menurunkan nilai kemiripan (Hasanah & Mutiara, 2019).

2.2 Landasan Teori

2.2.1 Website

Website adalah kumpulan halaman web yang saling terhubung dan seluruh file saling terkait. Web terdiri dari *page* atau halaman dan kumpulan halaman yang dinamakan *homepage*. *Homepage* berada pada posisi teratas dengan halaman-halaman terkait berada di bawahnya. Biasanya, setiap halaman di bawah *homepage* (*child page*) berisi *hyperlink* ke halaman lain dalam web (Agung, 2000).

Sejarah *website* dimulai oleh Tim Berners Lee, yang mengembangkan *World Wide Web* (WWW) pada tahun 1989. Pada bulan Oktober tahun 1990, Tim menggagas tiga teknologi dasar untuk membangun sebuah *website*, yaitu HTML, URL, dan HTTP. Awalnya, *website* dirancang hanya untuk berbagi informasi di kalangan para ilmuwan di *Centre Europeen pour la Recherche Nucleaire* (CERN). Sampai akhirnya, dia melihat potensi *website* sebagai sarana yang bisa digunakan oleh siapa saja untuk berbagai tujuan. Kemudian, dia merilisnya secara resmi pada 6 Agustus 1991.

Website yang pertama dirilis saat itu baru sebatas menampilkan teks sederhana tanpa variasi *font*, video, maupun gambar. Saat ini, *website* sudah berkembang sangat pesat dan menjadikan *website* sebagai salah satu kebutuhan penting baik individu, organisasi, dan perusahaan di seluruh dunia hingga saat ini. Saat ini, *website* berkembang sangat pesat dengan muncul berbagai jenis *website* seperti pribadi, *e-commerce*, *blog*, dan media sosial.

2.2.2 Hypertext Transfer Protocol Secure (HTTPS)

Hypertext Transfer Protocol Secure (HTTPS) adalah sebuah *protocol* komunikasi dalam suatu jaringan internet dengan keamanan yang lebih terjamin. Disebut lebih aman karena suatu perintah atau data yang dikirim melalui HTTPS ini dilindungi dengan sistem enkripsi sehingga menyulitkan *hacker* untuk membobol atau mencurinya (Firmansyah, 2019). HTTPS merupakan tingkatan dari HTTP di mana yang membedakan di HTTPS terdapat (*Secure Socket Layer*) SSL dan (*Transport Layer Security*) TLS yang digunakan untuk mengamankan data yang disimpan atau yang akan dikirim. Menurut SSL Labs pada April 2018, 33.2% dari 1.000.000 situs web teratas Alexa menggunakan HTTPS sebagai *default*,

57.1% dari 137.971 situs web paling populer di Internet memiliki implementasi HTTPS yang aman, dan 70% dari pemuatan halaman (diukur oleh *Firefox Telemetry*) menggunakan HTTPS. Jadi saat ini *website* sekarang sudah mulai beralih dari sebelumnya HTTP ke HTTPS karena di HTTPS lebih aman daripada HTTP yang sudah lama.

2.2.3 Javascript

Javascript adalah sebuah bahasa pemrograman tingkat tinggi yang dinamis, *scripting*, *untyped*, dan *interpreter*. Javascript sendiri dibuat oleh Brendan Eich dari perusahaan Netscape pada tahun 1994 yang diberi nama Mocha pada saat itu, kemudian berganti menjadi Livescript. Karena saat itu browser yang populer adalah Netscape, Microsoft berusaha untuk mengalahkan popularitas browser tersebut dengan Internet Explorer dan melakukan *Reverse Engineering* terhadap Livescript dan terciptalah JScript pada tahun 1996. Karena terdapat dua browser yang besar yang berbeda, maka dibuatlah satu standar agar mempermudah pembuatan website saat itu dan dibuatlah ECMAScript. Menurut Douglas Rockford, "*Javascript, JScript, and ECMAScript 3 Silly Name for 1 Silly Language*" yang berarti bahwa ketiga nama tersebut adalah bahasa yang sama yaitu Javascript.

2.2.3.1 Karakteristik Javascript

Untuk Ekstensi dari file Javascript menggunakan ekstensi *.js. Contoh namaFile.js. Setiap akhir dari kode Javascript dapat menggunakan ";" ataupun tidak. Sedangkan *block scope* di Javascript menggunakan { }.

Javascript merupakan salah satu bahasa pemrogram yang *untyped* / *dynamicly typed* yang berarti tidak mendefinisikan terlebih dahulu tipe variabel yang akan didefinisikan. Untuk penamaan variabel, terdapat beberapa *keyword* yaitu var, let, dan const. Untuk var, dan let, nilai variabel dapat berubah atau diisi ulang, sedangkan const nilai variabelnya tidak dapat diubah. Perbedaan var dan let sendiri terletak pada *hoisting*. Sedangkan untuk penamaan variabel, hampir sama dengan beberapa bahasa pemrograman lainnya seperti karakter pertama variabel tidak boleh angka, menggunakan penulisan *camelCase*, dan untuk *keyword* const yang merupakan variabel const biasanya menggunakan *snake_case* dan semuanya huruf kapital. Untuk contoh pembuatan variabel dapat dilihat pada Gambar 2.1.

```
var value = 10 // Contoh Penggunaan var
let nilai = 'Delapan' // Contoh Penggunaan let
const PI = 3.14 // Contoh Penggunaan const
```

Gambar 2.1 Contoh Pembuatan Variabel

Javascript memiliki beberapa tipe data seperti *String*, *Number* atau *Integer*, *Boolean*, *Array*, *Object*, dan *Undefined*. Untuk mengetahui tipe data dari sebuah variabel dapat menggunakan *keyword typeof*. Untuk contoh penggunaan tipe data dapat dilihat pada Gambar 2.2.

```
let name = 'Aris Akhyar Abdillah' // String
let age = 22 // Number atau Integer
let male = true // Boolean
let family = ['Father', 'Mother', 'Son', 'Girl'] // Array
let detail = {
  fullName : 'Aris Akhyar Addillah',
  nim : 'H071171505'
} // Object
let car = undefined // Undefined
```

Gambar 2.2 Contoh Penggunaan Tipe Data

Synchronous dan *Asynchronous*, Secara sederhana, *Synchronous* dan *Asynchronous* merupakan tahapan dalam mengeksekusi sebuah kode di mana *Synchronous* mengeksekusi sebuah kode per baris sesuai urutan kode yang dituliskan. Sedangkan *Asynchronous*, tidak selalu seperti *Synchronous*, tapi melihat waktu proses dari kode tersebut. Penggunaan *Asynchronous* tidak akan menunggu suatu kode selesai dijalankan, tetapi berlanjut ke kode selanjutnya. Untuk contoh penggunaan *Synchronous* dan *Asynchronous* dapat dilihat pada Gambar 2.3.

```

// Menggunakan Synchronous
console.log('What Is Your Name ?')
console.log('Hello World')
console.log('My Name is Aris')

// Output :
// What Is Your Name ?
// Hello World
// My Name is Aris

// Menggunakan Asynchronous
setTimeout(function() {
    console.log('What Is Your Name ?')
}, 1500)
console.log('Hello World')
setTimeout(function() {
    console.log('My Name is Aris')
}, 1000)

// Output :
// Hello World
// My Name is Aris
// What Is Your Name ?

```

Gambar 2.3 Contoh Penggunaan *Synchronous* dan *Asynchronous*

Jika melihat output dari dua program di atas antara menggunakan *Synchronous* dan *Asynchronous* terdapat perbedaan urutan dari hasil *output*, di mana *Synchronous* memiliki *output* sesuai dari urutan baris kode sedangkan *Asynchronous* berbeda di mana *output* yang dihasilkan berdasarkan jumlah waktu pengekseskuan dari kode tersebut yang disimulasikan menggunakan *setTimeout()* dari pengekseskuan paling cepat hingga yang paling lambat.

2.2.3.2 NodeJS

Node.js adalah *runtime environment* untuk Javascript yang bersifat *open-source* dan *cross-platform*. Dengan Node.js kita dapat menjalankan kode Javascript di mana pun, tidak hanya terbatas pada lingkungan *browser*. Seperti diketahui bahwa Javascript hanya dapat berjalan pada sebuah *web browser*, kemudian Ryan Dahl,

membuat sebuah *runtime environment* dengan mengeluarkan *engine* Javascript dari Chrome yaitu *V8 Javascript Engine* menggunakan bahasa C agar Javascript dapat dijalankan diluar *browser*. Akhirnya tercipta NodeJS pada tahun 2009. Dengan Begitu, Javascript yang sebelumnya hanya bisa di *client side* dengan adanya NodeJS bisa juga di *server side*. Beberapa fitur yang terdapat di NodeJS seperti *Asynchronous & Event-driven, Single Threaded but Highly Scalable* (Anonymous, NodeJS, 2022).

2.2.3.3 ExpressJS

Menurut website resmi dari Express JS adalah “*Fast, unopinionated, minimalist web framework for Nodejs*” (Anonymous, Express JS, 2022). Express JS sendiri merupakan salah satu *web framework* khusus untuk NodeJS di mana kita dapat membuat sebuah *website* yang cepat, sederhana, dengan struktur yang tidak ditentukan atau tergantung dari pengguna ExpressJS sendiri.

2.2.4 Database

Database atau basis data adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data (Andaru, 2018). Berdasarkan definisi *database*, *database* berfungsi untuk menyimpan catatan atau sebuah data pada sebuah penyimpanan yang kemudian akan digunakan pada waktu lainnya. Pada umumnya, database sudah pasti memiliki *key* dan *value*, walaupun istilah ini berbeda di tiap-tiap jenis database yang ada. Adapun beberapa jenis database seperti *Operational Database*, *Database Warehouse*, *Distributed Database*, *Relational Database*, dan *End User Database*.

2.2.4.1 SQL dan NoSQL

SQL dan NoSQL merupakan salah satu contoh dari *Relational Database* yang populer saat ini. SQL (*Structured Query Language*) merupakan bahasa yang digunakan untuk mengelola data secara *relational*. SQL sendiri memiliki ciri yaitu memiliki tabel yang terdiri dengan *row* atau *record* dan *field* yang bisa memiliki relasi dengan tabel lainnya. Untuk tiap data yang ada di dalam tabel, harus memiliki skema yang sama untuk tiap recordnya. Contoh dari SQL seperti MySQL, PostgreSQL, dan MariaDB. Berbeda dengan NoSQL (*Not Only Structured Query*

Language) yang merupakan bahasa untuk mengelola data secara *Non Relational*. Berbeda dengan SQL yang menggunakan tabel untuk menyimpan data. NoSQL memiliki banyak jenis tempat untuk menyimpan data seperti *Document Database*, *Key-Value Database*, dan *Graph Database*. NoSQL juga tidak memiliki skema sehingga untuk menyimpan data bisa secara flexibel. Contoh dari NoSQL seperti MongoDB, Redis, Neo4j, dan Cassandra.

2.2.4.2 MongoDB

MongoDB merupakan salah satu contoh dari NoSQL yang dikembangkan menggunakan bahasa pemrograman C++ yang rilis pertama kali pada tanggal 11 Februari 2009. MongoDB adalah salah satu contoh *Document Database* yang di mana tiap - tiap datanya merupakan sebuah JSON (*Javascript Object Notation*) atau dalam MongoDB disebut BSON (*Binary JSON*). Hingga saat ini, MongoDB sudah digunakan lebih dari 85 Juta pengguna di seluruh dunia dan sudah banyak perusahaan besar yang menggunakan database ini seperti EBay, Google, Adobe, dan EA (Anonymous, What Is MongoDB ?, 2022). Salah satu contoh data yang terdapat di MongoDB :

```
{
  _id : '91829jw1j000jo2',
  firstName : 'Aris',
  lastName : 'Akhyar',
  age : 20
  hobbies : ['eat', 'drink']
}
```

Gambar 2.4 Contoh Code Mongoddb

2.2.5 Jurnal

Jurnal merupakan bagian dari jenis terbitan berseri yang ada diperpustakaan, adapun pengertian jurnal menurut *High Beam* “*Journal is the collection and periodic publication or transmission of news and the result of research through media*”, artinya bahwa jurnal merupakan suatu koleksi dan terbitan berkala atau transmisi mengenai berita dan hasil-hasil penelitian mengenai media. Jurnal sendiri terbagi atas dua format yaitu tercetak dan digital (*e-journal*). Untuk format digital

jurnal dikemas dalam dua format , yaitu bentuk CD-ROM dan dalam bentuk akses secara *online* melalui *internet*. *E-Journal* dipahami sebagai publikasi ilmiah dalam format elektronik dan mempunyai ISSN (*International Standard Serial Number*) yang format dokumennya menggunakan pdf (Rusydi, 2014).

Penggunaan kata jurnal untuk berbagai bidang juga memberi arti yang bervariasi, misalnya jurnal dalam bidang ekonomi menunjukkan sistem pembukuan rangkap. Jurnal dalam bidang pelayaran diartikan sebagai *logbook* berarti buku untuk mencatat semua kejadian selama pelayaran. Jurnal sebenarnya merupakan publikasi ilmiah yang memuat informasi tentang hasil kegiatan dalam bidang ilmu pengetahuan dan teknologi minimal harus mencakup kumpulan atau kumulasi pengetahuan baru, pengamatan empiris dan pengembangan gagasan atau usulan. Dengan demikian jurnal merupakan representasi dari pengetahuan baru tentang perkembangan ilmu pengetahuan yang dilaksanakan secara empiris dan biasanya merupakan gagasan yang terbaru.

2.2.6 Plagiarisme

Plagiarisme berasal dari bahasa Latin “*plagiare*” yang berarti mencuri. Plagiarisme berasal dari kata plagiat yang berarti pengambilan karangan (pendapat dan sebagainya) orang lain dan menjadikannya seolah-olah karangan (pendapat dan sebagainya) sendiri, misalnya menerbitkan karya tulis orang lain atas nama dirinya sendiri. Sehingga dapat diartikan plagiarisme merupakan tindakan mencuri gagasan hasil penelitian orang lain, untuk kemudian disajikan seolah-olah milik sendiri (Ridhatillah, 2003).

Tindakan plagiarisme merupakan salah satu tindakan yang melanggar hak cipta. Hak Cipta itu sendiri merupakan hak eksklusif untuk pencipta ataupun penerima hak buat mengumumkan ataupun perbanyak Ciptaannya ataupun membagikan izin buat itu dengan tidak mengurangi pembatasan-pembatasan bagi peraturan perundang-undangan yang berlaku. Jika terjadi pelanggaran tersebut, dapat dikenai pelanggaran hak cipta di Pasal 72 ayat UUHC dengan dipidana dengan pidana penjara pendek selama 1 bulan serta / ataupun denda sangat sedikitnya Rp1.000.000,00, ataupun pidana penjara lama 7 tahun serta / ataupun denda sebanyak Rp5.000.000.000,00 (Lopes, 2013).

2.2.7 Dice Similarity

Dice Similarity atau *Sørensen–Dice coefficient* merupakan salah satu algoritma yang mengukur kesamaan antara dua set data (Khontoro, Andjarwirawan, & Yulia, 2021). Algoritma ini banyak digunakan dalam validasi algoritma segmentasi gambar yang dibuat dengan *Artificial Intelligence* atau AI, tetapi ini adalah konsep yang jauh lebih umum yang dapat diterapkan pada kumpulan data untuk berbagai aplikasi termasuk *Natural Language Processing* (NLP). Metode ini sering digunakan untuk mengukur kemiripan antara dua dokumen teks dengan menggunakan pendekatan N-gram. N-gram adalah cara untuk membagi dokumen menjadi potongan-potongan kecil yang terdiri dari N karakter kemudian divisualisasikan menjadi vektor.

Dice Similarity memiliki kesamaan terhadap *Jaccard Similarity* di mana keduanya merupakan cara untuk membandingkan kedua data set atau vektor menggunakan pendekatan matematika. Untuk rumus dari *Jaccard Similarity* dengan X dan Y sebagai vektor yaitu :

$$J = \frac{|X.Y|}{|X|^2 + |Y|^2 - |X.Y|} \quad (2.1)$$

Untuk memperoleh rumus dari *Dice Similarity*, dapat menggunakan persamaan yaitu :

$$DSC = \frac{2J}{J + 1} \quad (2.2)$$

Dengan DSC adalah *Dice Similarity* dan J adalah persamaan dari *Jaccard Similarity* yang terdapat pada persamaan 2.1. Kemudian, nilai dari variabel J diganti dengan persamaan *Jaccard Similarity* sebelumnya sehingga menjadi :

$$DSC = \frac{2 \frac{|X.Y|}{|X|^2 + |Y|^2 - |X.Y|}}{\frac{|X.Y|}{|X|^2 + |Y|^2 - |X.Y|} + 1} \quad (2.3)$$

$$DSC = \frac{2 \frac{|X.Y|}{|X|^2 + |Y|^2 - |X.Y|}}{\frac{|X.Y| + |X|^2 + |Y|^2 - |X.Y|}{|X|^2 + |Y|^2 - |X.Y|}} \quad (2.4)$$

$$DSC = \frac{2 \frac{|X.Y|}{|X|^2 + |Y|^2 - |X.Y|}}{\frac{|X|^2 + |Y|^2}{|X|^2 + |Y|^2 - |X.Y|}} \quad (2.5)$$

$$DSC = \frac{2 \frac{|X.Y|}{|X|^2 + |Y|^2 - |X.Y|}}{\frac{|X|^2 + |Y|^2}{|X|^2 + |Y|^2 - |X.Y|}} \quad (2.6)$$

$$DSC = \frac{2 |X.Y|}{|X|^2 + |Y|^2} \quad (2.7)$$

Sehingga diperoleh persamaan dari *Dice Similarity* menggunakan data *vektor*, dengan X dan Y sebagai *vektor*, maka persamaan akhirnya yaitu sesuai dengan persamaan 2.7. Metode *Dice Similarity* memiliki kelebihan yaitu tidak membutuhkan komputasi yang terlalu berat, namun memiliki kekurangan yaitu sensitif terhadap perbedaan panjang dokumen.

2.2.7.1 Implementasi *Dice Similarity*

Untuk pengimplementasian dari metode *Dice Similarity*, diambil contoh dari definisi dari komputer menurut para ahli yaitu Menurut Robert H. Blissmer, “komputer adalah suatu alat elektronik yang mampu melakukan beberapa tugas seperti menerima input, memproses input tadi sesuai dengan programnya, menyimpan perintah-perintah dan hasil pengolahan, serta menyediakan output dalam bentuk informasi”. Sedangkan menurut Arief Susanto, “komputer adalah sekelompok alat elektronik yang terdiri atas perintah input, alat yang mengolah input, dan peralatan output yang memberikan informasi serta bekerja secara otomatis” (Susanto, 2009).

Setelah mendapatkan kedua data di atas, kemudian dilakukan pembagian. Untuk pengertian dari Robert H. Blissmer akan menjadi data dari *database* sebagai X, kemudian untuk pengertian dari Arief Susanto akan menjadi data yang akan dibandingkan dengan data dari *database* sebagai Y. Sebelum diolah, kedua data di atas akan dilakukan penghapusan *Stopword* dan karakter yang tidak berguna dalam perhitungan seperti pada Tabel 2.1.

Tabel 2.1 Perbandingan Data X dan Data Y

Data X	Data Y
alat elektronik tugas menerima input memproses input sesuai programnya menyimpan perintah - perintah hasil pengolahan menyediakan output bentuk informasi	sekelompok alat elektronik perintah input alat mengolah input peralatan output informasi otomatis

Setelah dilakukan penghapusan *Stopword* dan karakter yang tidak berguna, kemudian dilakukan pembuatan kamus kata lalu dilakukan perhitungan jumlah kata terhadap kamus data yang ada seperti pada Tabel 2.2.

Tabel 2.2 Kamus Data

No	Kamus Data	Data X	Data Y
1	alat	2	1
2	bentuk	0	1
3	elektronik	1	1
4	hasil	0	1
5	informasi	1	1
6	input	2	2
7	memproses	0	1
8	menerima	0	1
9	mengolah	1	0
10	menyediakan	0	1
11	menyimpan	0	1
12	otomatis	1	0
13	output	1	1
14	pengolahan	0	1
15	perintah	1	2
16	perlatan	1	0
17	programnya	0	1
18	sekelompok	1	0
19	sesuai	0	1
20	tugas	0	1

Setelah itu, data di atas akan diubah menjadi vektor satu dimensi sehingga menjadi :

$$X = \{2, 0, 1, 0, 1, 2, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0\} \quad (2.8)$$

$$Y = \{1, 1, 1, 1, 1, 2, 1, 1, 0, 1, 1, 0, 1, 1, 2, 0, 1, 0, 1, 1\} \quad (2.9)$$

Kemudian kedua vektor di atas dimasukkan ke dalam rumus *Dice Similarity* pada persamaan 2.7. Untuk mempermudah perhitungan, dilakukan beberapa penambahan *row* dan *field* sehingga menjadi :

Tabel 2.3 Perhitungan Dalam Kamus Data

No	Kamus Data	Data X	Data Y	X . Y	X ²	Y ²
1	alat	2	1	2	4	1
2	bentuk	0	1	0	0	1
3	elektronik	1	1	1	1	1
4	hasil	0	1	0	0	1
5	informasi	1	1	1	1	1
6	input	2	2	4	4	4
7	memproses	0	1	0	0	1
8	menerima	0	1	0	0	1
9	mengolah	1	0	0	1	0
10	menyediakan	0	1	0	0	1
11	menyimpan	0	1	0	0	1
12	otomatis	1	0	0	1	0
13	output	1	1	1	1	1
14	pengolahan	0	1	0	0	1
15	perintah	1	2	2	1	4
16	perlatan	1	0	0	1	0
17	programnya	0	1	0	0	1
18	sekelompok	1	0	0	1	0
19	sesuai	0	1	0	0	1
20	tugas	0	1	0	0	1
Total		12	18	11	16	22

Setelah memperoleh hasil seperti di Tabel 2.3, kemudian nilai tersebut dimasukkan ke dalam rumus *Dice Similarity* sehingga menjadi :

$$DSC = \frac{2|X.Y|}{|X|^2 + |Y|^2} \quad (2.10)$$

$$DSC = \frac{2|11|}{16 + 22} \quad (2.11)$$

$$DSC = \frac{22}{38} \quad (2.12)$$

$$DSC = 0.5789473684 \quad (2.13)$$

Sehingga diperoleh nilai *Dice Similarity* adalah 0.5789473684 atau jika dipersentasekan menjadi 57.8 %. Jadi, persentase kesamaan antara Data X dengan Data Y adalah sebesar 57.8 %

BAB III

METODE PENELITIAN

3.1 Prosedur Penelitian

Prosedur Penelitian merupakan serangkaian kegiatan yang dilaksanakan secara sistematis serta teratur agar mencapai tujuan dari penelitian ini. Adapun prosedur penelitian sebagai berikut :

1. Analisa Kebutuhan Sistem

Dalam tahap ini, dilakukan analisa terhadap kebutuhan sebuah sistem seperti bagaimana tampilan sistem, bagaimana cara kerja sistem, ataupun penggunaan bahasa pemrograman yang tepat.

2. Studi Literatur

Dalam tahap ini, dilakukan pengumpulan teori-teori ataupun landasan dasar dalam proses membangun sebuah sistem yang akan dibuat.

3. Perancangan Sistem

Dalam tahap ini, dilakukan pembangunan Sistem Pendeteksi Kesamaan Teks Menggunakan Metode *Dice Similarity* dengan menggunakan metode, teori, ataupun bahasa pemrograman yang telah ditentukan sebelumnya.

4. Pengujian Sistem

Dalam tahap ini, dilakukan *testing* atau pengujian terhadap sistem.

3.2 Timeline Penelitian

Timeline Penelitian adalah gambaran visual yang menggambarkan peristiwa atau tahapan yang akan terjadi dalam penelitian ini. Untuk *Timeline* penelitian dapat dilihat pada Tabel 3.1.

Tabel 3.1 *Timeline* Penelitian

No	Prosedur Penelitian	Waktu															
		Oktober				November				Desember				Januari			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Analisa Kebutuhan Sistem																
2	Studi Literatur																
3	Perancangan Sistem																
4	Pengujian Sistem																

Pada *timeline* penelitian ini, terdapat beberapa prosedur yang akan dilakukan dengan rentang waktu yang ada seperti analisa kebutuhan sistem, studi literatur, perancangan sistem, dan pengujian sistem dengan rentang waktu yang ada. Dengan adanya *timeline* penelitian ini, dapat mengendalikan pelaksanaan penelitian secara menyeluruh agar pelaksanaan penelitian tersebut berjalan dengan lancar.

3.3 Rancangan Sistem

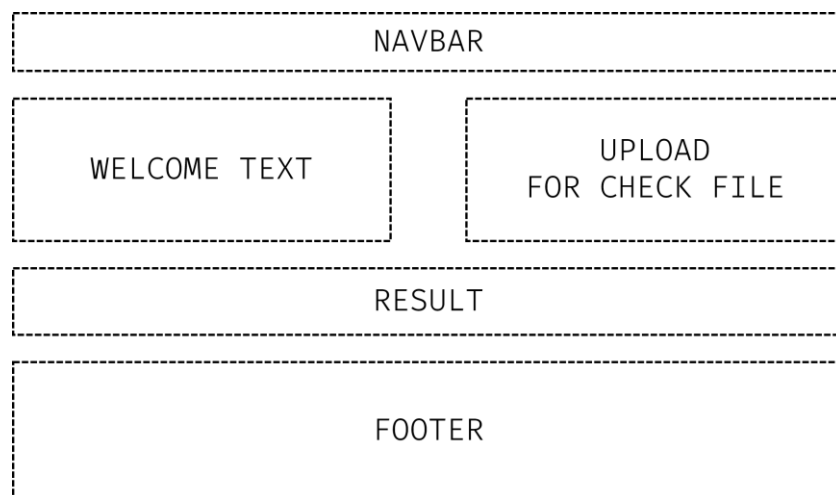
3.3.1 Rancangan Antar Muka (*Interface*)

Rancangan antar muka adalah suatu proses dalam menyusun dan menganalisis sistem secara fisik maupun non fisik untuk dapat dimanfaatkan ke depannya. Perancangan antarmuka terdiri dari beberapa proses mulai dari membuat fungsi yang ada pada sistem, informasi masalah, tahapan membuat *prototype*, melakukan implementasi terhadap perancangan yang dibuat dan evaluasi hasil (Sridevi, 2014).

Untuk aplikasi terdapat beberapa rancangan *interface* seperti berikut :

a. Halaman Utama

Halaman utama merupakan halaman awal yang akan diakses pertama kali saat membuka *website* ini. Untuk rancangan antar muka pada halaman utama dapat dilihat pada Gambar 3.1.

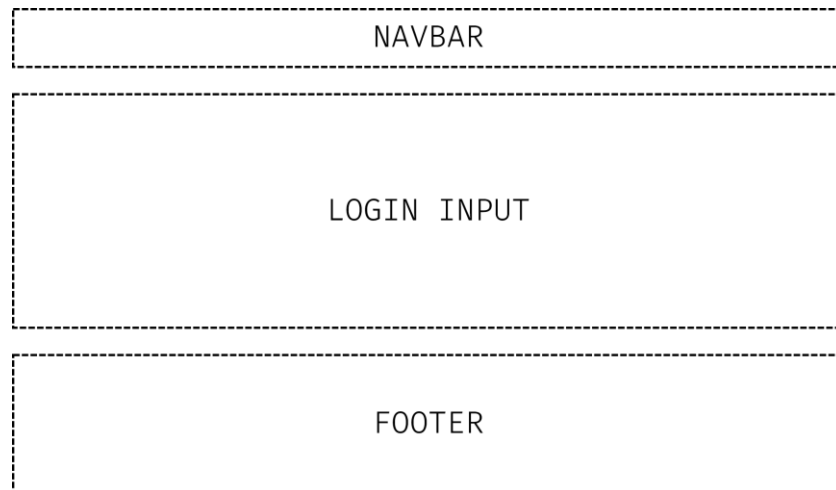


Gambar 3.1 Rancangan Antar Muka Halaman Utama

b. Halaman *Login*

Halaman *login* merupakan halaman untuk masuk ke dalam *website*. Pada halaman ini terdapat input berupa input *username* dan *password*, serta

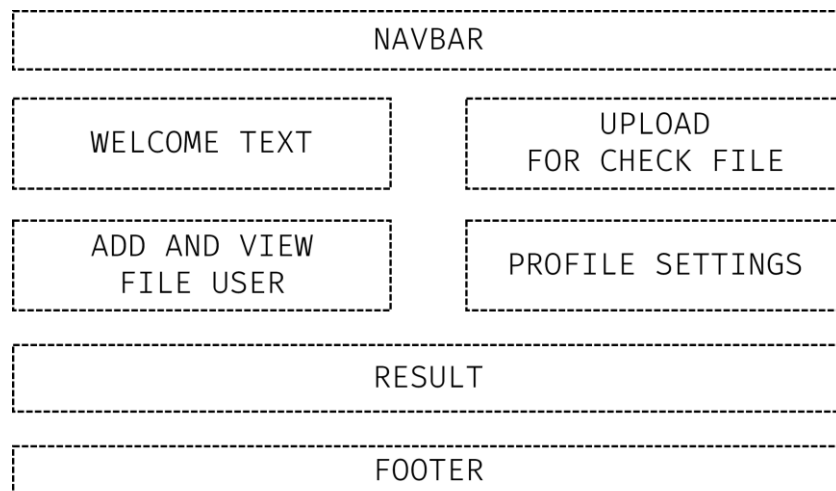
tombol *login*. Untuk rancangan antar muka pada halaman login dapat dilihat pada Gambar 3.2.



Gambar 3.2 Rancangan Antar Muka Halaman *Login*

c. Halaman Beranda

Halaman beranda merupakan halaman yang akan didapatkan setelah berhasil *login* ke dalam *website* sebagai user. Pada halaman ini, terdapat beberapa fitur seperti tambah *file* dan ubah pengaturan *profile*. Untuk rancangan antar muka pada halaman beranda dapat dilihat pada Gambar 3.3.

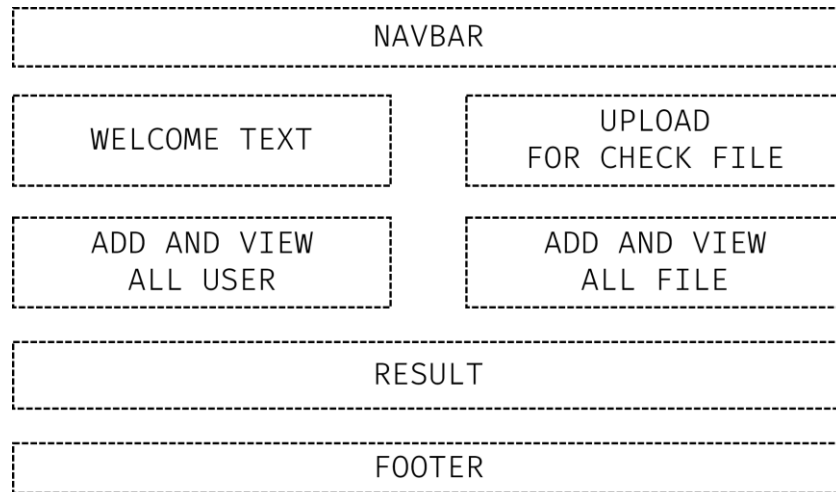


Gambar 3.3 Rancangan Antar Muka Halaman Beranda

d. Halaman *Admin*

Halaman *admin* merupakan halaman yang didapatkan setelah berhasil *login* sebagai *admin*. Untuk halaman ini, terdapat lebih banyak tambahan fitur

dibandingkan dengan halaman beranda seperti dapat menambah atau menghapus *user* dan *files*. Untuk rancangan antar muka pada halaman admin dapat dilihat pada Gambar 3.4.



Gambar 3.4 Rancangan Antar Muka Halaman *Admin*

3.4 Metode Penelitian

Dalam sistem ini, metode perhitungan yang digunakan dalam metode *Dice Similarity* dengan membandingkan data teks yang diinput pengguna dengan teks yang terdapat di dalam *database* atau repositori teks.

3.4.1 Metode Prototype

Metode *Prototype* adalah sebuah metode pengembangan *software* yang cukup banyak digunakan karena Teknik ini mengembangkan sistem yang menggunakan *prototype* untuk menggambarkan sistem sehingga klien atau pemilik sistem mempunyai gambaran jelas pada sistem yang akan dibangun oleh tim pengembang. Dengan metode ini, pengembang dan pengguna bisa saling berinteraksi selama proses pengembangan *software*. Hal ini tentu sangat menguntungkan dan semakin memudahkan dalam pembuatan perangkat lunak.

Saat menggunakan model jenis ini, kesalahan biasanya dapat dideteksi lebih cepat dan umpan balik pengguna yang lebih cepat tersedia untuk menghasilkan solusi yang lebih baik. Dalam metodologi ini model kerja dari sistem disediakan, pengguna mendapatkan pemahaman yang lebih baik tentang sistem yang sedang dikembangkan. Developer bisa bekerja menentukan kebutuhan klien dengan baik,

Efisiensi waktu tinggi dalam pengembangan sistem serta Lebih mudah dalam penerapannya karena klien mengetahui apa yang dibutuhkan.

Dalam penelitian ini dilakukan sesuai tahapan yang ada pada model *prototype* untuk mempermudah memahami proses dari awal sampai sistem selesai dan implementasi.

3.5 Instrumen Penelitian

Instrumen yang digunakan dalam penelitian ini adalah terbagi dua yaitu *hardware* dan *software*. Untuk *hardware*, menggunakan laptop dengan *processor* AMD Athlon 300U with Radeon Vega Mobile Gfx 2.40 GHz dan RAM sebanyak 8 GB serta sistem operasi Windows 10 Home 64 bit. Sedangkan *software* yang digunakan yaitu *Visual Studio Code* dan *Microsoft Edge*.

BAB IV HASIL DAN PEMBAHASAN

4.1. *Requirements Gatehering and Analysis (Analisis Kebutuhan)*

Dalam tahap ini kebutuhan sistem didefinisikan dengan rinci. Dalam prosesnya, peneliti akan menganalisa kebutuhan apa saja yang diperlukan demi pembuatan sistem yang sesuai dengan keinginan agar aplikasi dapat berjalan sebagaimana mestinya. Kebutuhan yang akan dianalisis mulai dari sistem yang akan dibangun, bahasa yang digunakan, *database* yang sesuai dengan aplikasi serta infrastruktur yang digunakan dalam membangun aplikasi tersebut. Analisis kebutuhan fungsional sistem terdiri dari beberapa fungsi utama yang saling berkaitan dan saling mendukung satu sama lain. Sedangkan kebutuhan non fungsional sistem memiliki fungsi sebagai sarana pendukung agar kelancaran dari fungsi utama beroperasi sesuai dengan harapan.

Analisis kebutuhan fungsional pada sistem yaitu terdapat beberapa jenis pengguna yaitu *guest*, *user*, dan *admin* di mana semua jenis pengguna ini memiliki interaksi yang berbeda-beda sehingga memiliki ruang lingkup yang berbeda juga. Aktivitas yang akan terjadi pada pengguna berbeda-beda namun berhubungan. Untuk *guest* atau pengguna biasa hanya dapat melakukan pengecekan dokumen. Pada *user*, selain bisa melakukan apa yang bisa dilakukan oleh *guest*, pengguna juga bisa melakukan manajemen dokumen yang akan dijadikan sebagai pembanding dalam sistem seperti menambah, mengubah, atau menghapus dokumen tersebut. Sedangkan untuk Admin, selain pengguna dapat melakukan apa yang dapat dilakukan oleh *guest* dan *user*, pengguna juga dapat melakukan manajemen *user* seperti menambah, mengubah, atau menghapus *user*.

Analisis kebutuhan non fungsional yaitu meliputi kebutuhan akan spesifikasi seperti *hardware* dan *software* yang mendukung jalannya sistem. Sistem ini membutuhkan pemrosesan data yang cepat karena akan mengolah banyak data berupa dokumen. Selain itu, *database* yang digunakan harus bisa menangani data yang besar. Maka berdasarkan hal tersebut, maka kebutuhan non fungsional yang akan dibutuhkan yaitu :

A. Hardware

Spesifikasi perangkat keras (*hardware*) yang digunakan agar sistem dapat berjalan yaitu :

Processor : AMD Athlon 300U with Radeon Vega Mobile Gfx
2.40 GHz
Memory (RAM) : 8 Gb
Storage : Solid State Drive (SSD) 240 Gb

B. Software

Spesifikasi perangkat lunak (*software*) yang digunakan agar sistem dapat berjalan yaitu :

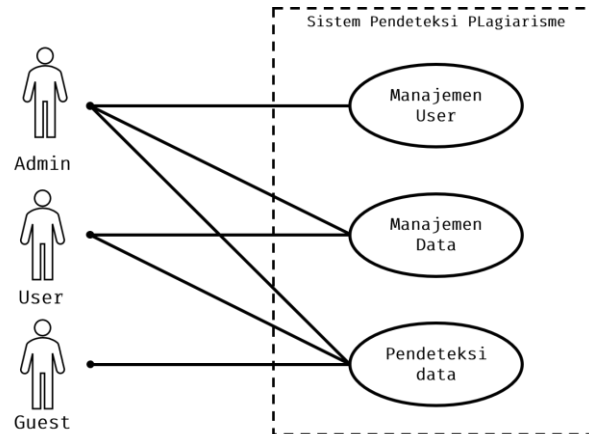
Sistem Operasi : Windows 10 Home Single Language
Server : NodeJs
Bahasa Pemrograman : Javascript
Framework : ExpressJs, Tailwindcss 3
Database : MongoDB
Text Editor : Visual Studio Code
Web Browser : Microsoft Edge

4.2 Quick Design (Desain Cepat)

Tahap selanjutnya adalah pembuatan desain sederhana yang akan memberi gambaran singkat tentang sistem yang ingin dibuat. Desain yang dibuat pada tahap ini berdasarkan pada tahap sebelumnya.

4.2.1 Use Case Diagram

Use Case Diagram adalah suatu jenis diagram yang menggambarkan hubungan atau interaksi dari sebuah sistem dengan pengguna. Dengan adanya *Use Case Diagram*, kita dapat mendeskripsikan tipe atau jenis yang dilakukan pengguna terhadap sebuah sistem. Untuk *use case diagram* dari Sistem Pendeteksi Kesamaan Teks Menggunakan Metode *Dice Similarity* dapat dilihat pada Gambar 4.1.

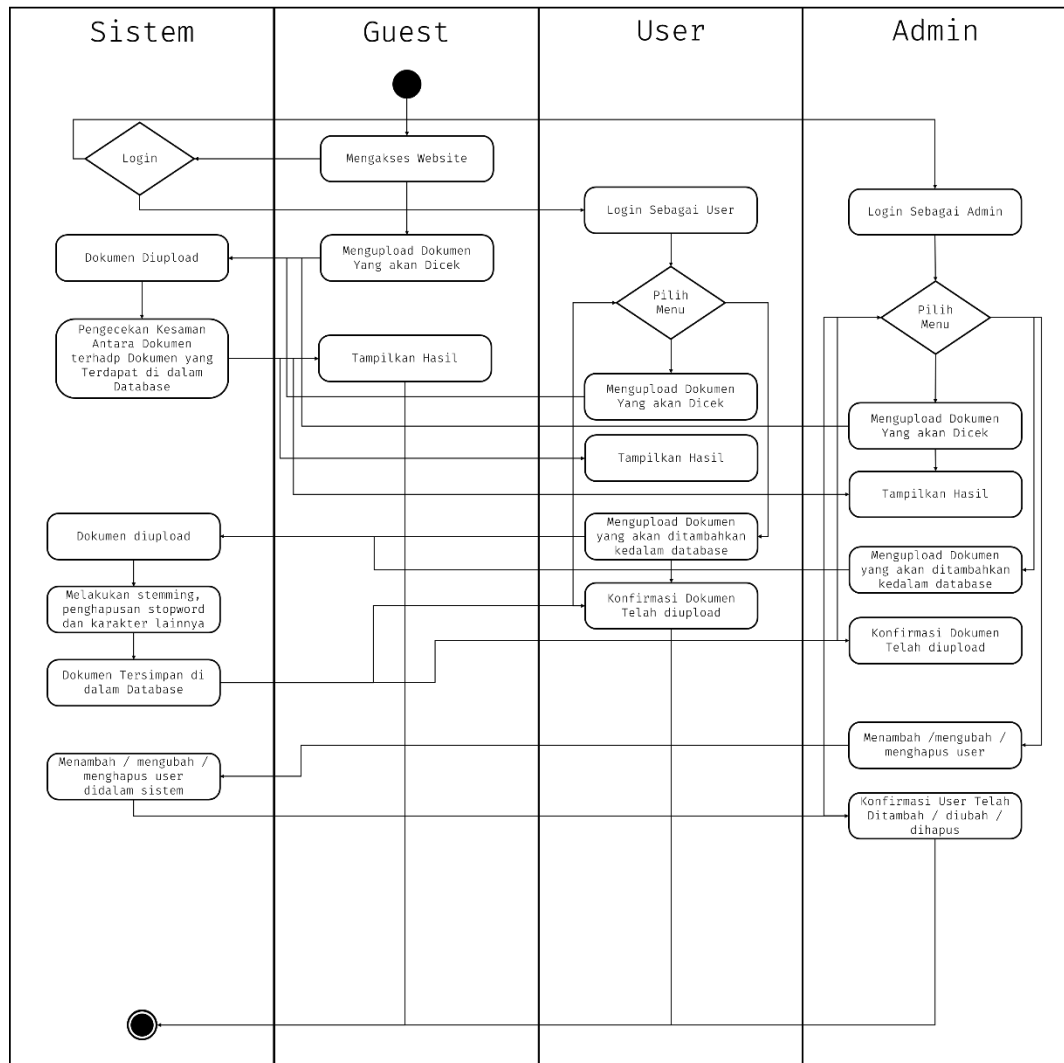


Gambar 4.1 Use Case Diagram

Dalam *use case diagram* pada Gambar 3.1, terdapat 3 jenis pengakses yang akan menggunakan sistem ini yaitu *Admin*, *User*, dan *Guest*. *Admin* memiliki akses terhadap semua sistem baik manajemen dari *user*, dan juga manajemen data yang disimpan dalam *database*. Untuk *user*, hanya memiliki akses terhadap data yang disimpan ke dalam *database* nantinya berdasarkan yang dimiliki oleh *user*. Selain itu, *user* juga dapat menggunakan sistem pendeteksi kesamaan dokumen yang ada dalam sistem ini. Sedangkan *guest* atau tamu hanya bisa melakukan pendeteksi kesamaan dokumen dan tidak bisa melakukan manajemen data yang ada di dalam *database*.

4.2.2 Activity Diagram

Activity diagram adalah diagram yang dapat memodelkan proses-proses yang terjadi pada sebuah sistem. Runtutan proses dari suatu sistem digambarkan secara vertikal. *Activity diagram* merupakan pengembangan dari *Use Case* yang memiliki alur aktivitas. Alur atau aktivitas berupa bisa berupa runtutan menu-menu atau proses bisnis yang terdapat di dalam sistem tersebut. Untuk *activity diagram* pada Sistem Pendeteksi Kesamaan Teks Menggunakan Metode *Dice Similarity* dapat dilihat pada Gambar 4.2.



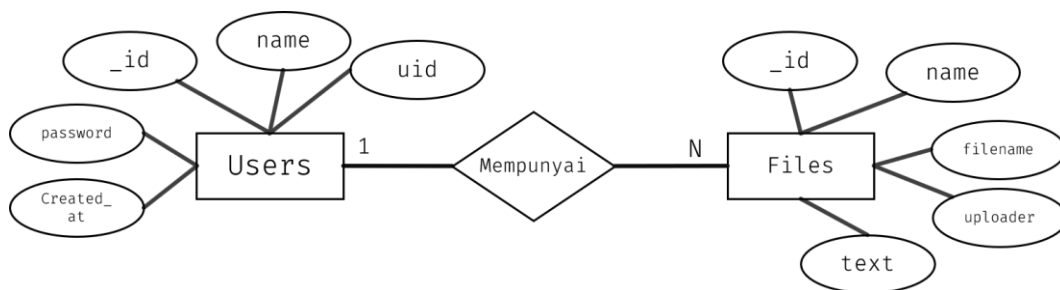
Gambar 4.2 *Activity Diagram*

Pada Gambar 4.2 dapat dilihat ada beberapa aktivitas yang dapat dilakukan pada sistem ini berdasarkan *role* atau jenis pengguna pada sistem ini seperti *guest* yang hanya terbatas hanya pada pengecekan kesamaan dokumen saja. Untuk pengguna user, selain bisa melakukan pengecekan dokumen, juga bisa menambah dokumen yang akan dijadikan perbandingan pada saat pengecekan nantinya. Selain itu, terdapat admin yang bisa melakukan semua yang dapat dilakukan oleh user dan *guest*. Selain itu admin dapat melakukan manajemen user seperti menambah, mengubah, dan menghapus user yang terdapat dalam sistem.

4.2.3 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah suatu jenis diagram yang digunakan untuk merancang suatu basis data (*database*). Dalam ERD, terdapat beberapa

komponen seperti entitas, relasi, atribut, dan garis penghubung di mana digunakan untuk memperlihatkan hubungan atau relasi antar entitas atau objek yang terlihat beserta atributnya. Untuk ERD pada Sistem Pendeteksi Kesamaan Teks Menggunakan Metode *Dice Similarity* dapat dilihat pada Gambar 4.3.



Gambar 4.3 *Entity Relation Diagram*

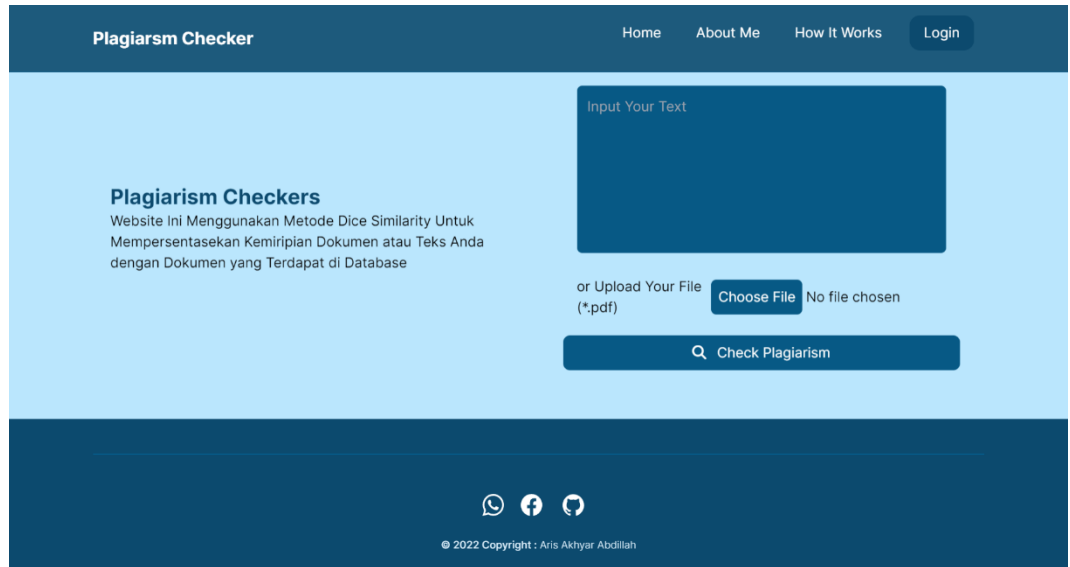
Berdasarkan ERD pada Gambar 4.3, terdapat dua entitas yaitu *Users* dan *Files*, di mana *Users* memiliki beberapa atribut yaitu *_id*, *name*, *password*, *uid*, *password*, dan *created_at*. Sedangkan *Files* memiliki beberapa atribut juga seperti *_id*, *name*, *filename*, *uploader*, dan *text*. *Collections Users* mempunyai relasi *One to Many* terhadap Tabel *Files*, sedangkan sebaliknya yaitu *Collections Files* memiliki relasi *One to One* terhadap *Collections Users*.

4.2.4 Tampilan Antar Muka (*Interface*)

Tampilan antar muka merupakan tampilan yang akan muncul saat pengguna mengakses *website* yang akan dibuat. Tampilan antarmuka ini dibuat berdasarkan rancangan antar muka yang telah rancang sebelumnya. Untuk tampilan antar muka dapat dilihat sebagai berikut :

a. Halaman Utama

Halaman utama merupakan halaman yang diakses pertama kali oleh user. Berdasarkan rancangan antar muka sebelumnya, maka tampilan antar muka halaman utama dapat dilihat pada Gambar 4.4.

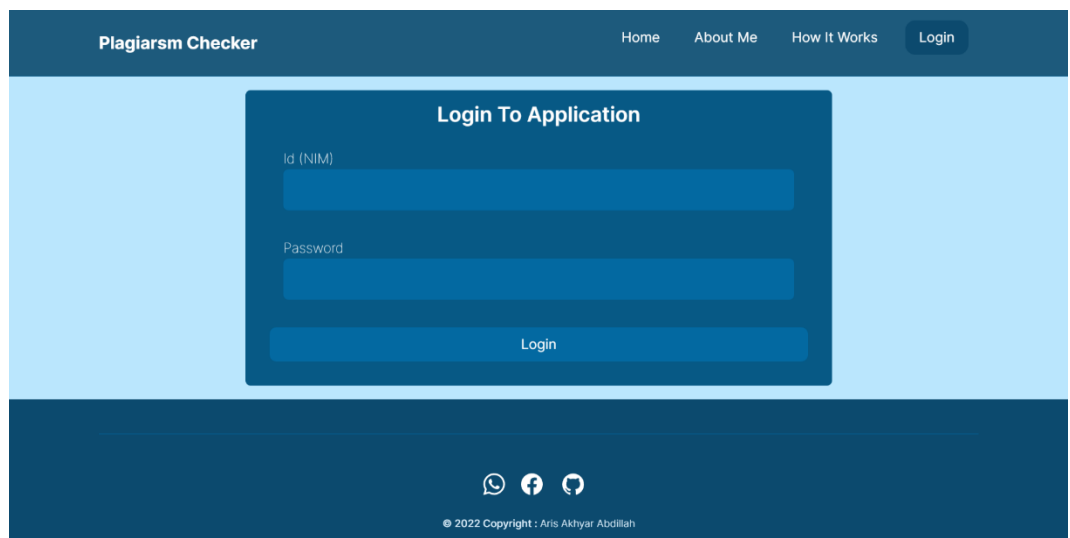


Gambar 4.4 Tampilan Halaman Utama

Dapat dilihat pada Gambar 4.4, tampilan *website* ini berdasarkan rancangan antar muka terdapat beberapa bagian seperti *navbar*, *welcome text*, *upload for check file*, serta *footer*.

b. Halaman Login

Halaman login merupakan halaman yang diakses ketika pengguna ingin mengakses *website* baik sebagai user, ataupun sebagai admin. Berdasarkan rancangan antar muka sebelumnya, maka tampilan antar muka halaman utama dapat dilihat pada Gambar 4.5.

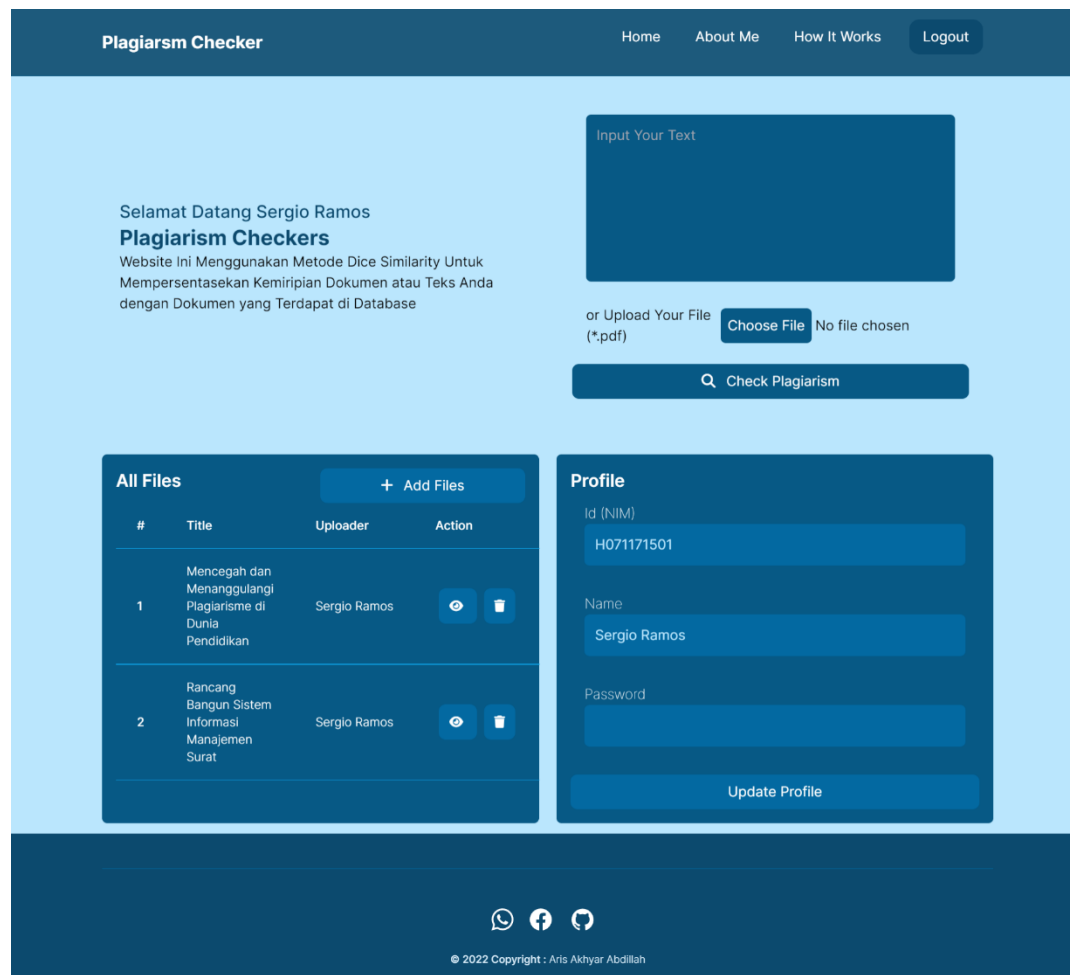


Gambar 4.5 Tampilan Halaman *Login*

Pada Gambar 4.5, tampilan pada halaman *login* ini berdasarkan rancangan antar muka yang memiliki beberapa bagian seperti *navbar*, *login input*, dan *footer*.

c. Halaman Beranda

Halaman beranda merupakan halaman yang didapat pengguna ketika berhasil *login* ke dalam sistem sebagai user. Berdasarkan rancangan antar muka sebelumnya, maka tampilan antar muka halaman utama dapat dilihat pada Gambar 4.6.

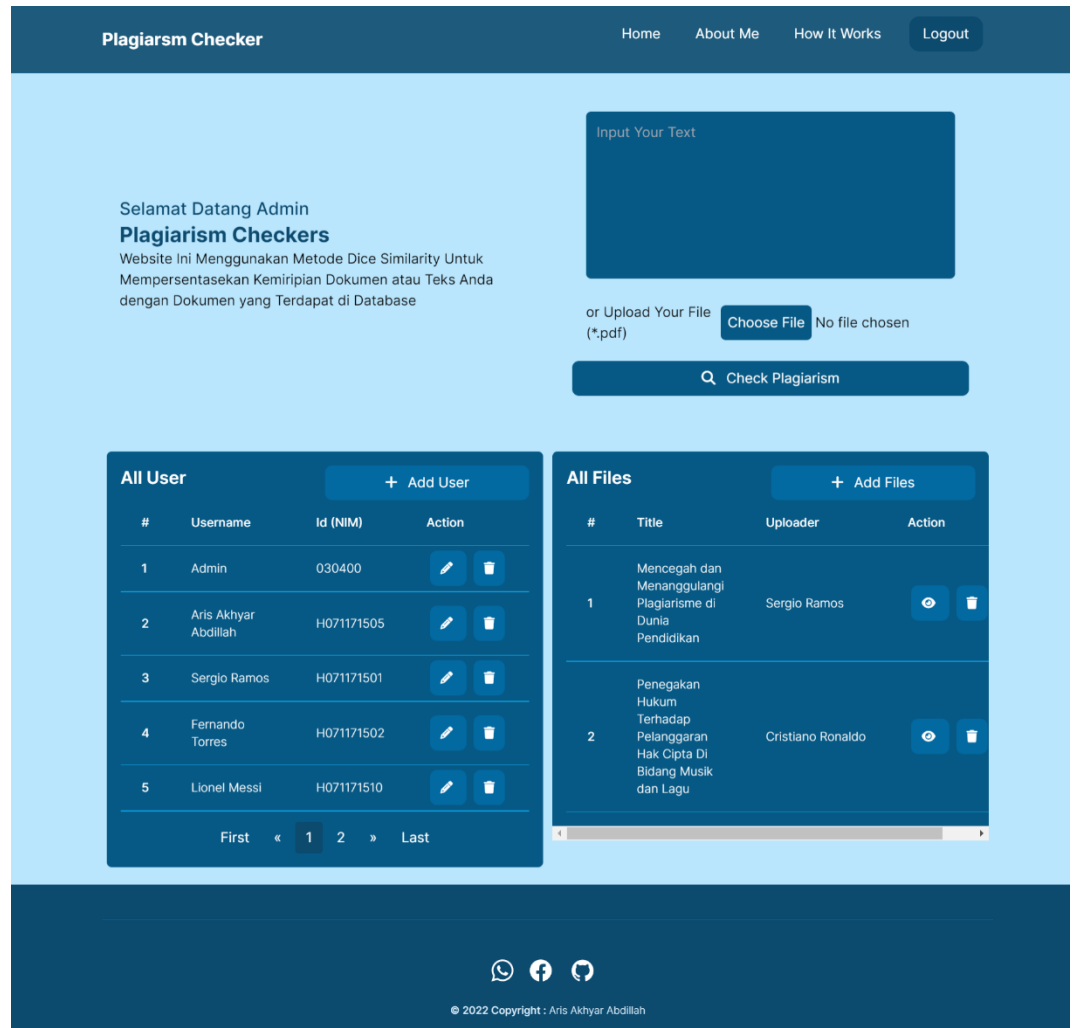


Gambar 4.6 Tampilan Halaman Beranda

Pada Gambar 4.6, tampilan pada halaman beranda ini berdasarkan rancangan antar muka yang memiliki beberapa bagian seperti *navbar*, *welcome text*, *upload for check file*, *add and view file user*, *profile settings*, dan *footer*.

d. Halaman Admin

Halaman admin merupakan halaman yang didapat pengguna ketika login sebagai admin ke dalam sistem. Berdasarkan rancangan antar muka sebelumnya, maka tampilan antar muka halaman utama dapat dilihat pada Gambar 4.7.



Gambar 4.7 Tampilan Halaman Admin

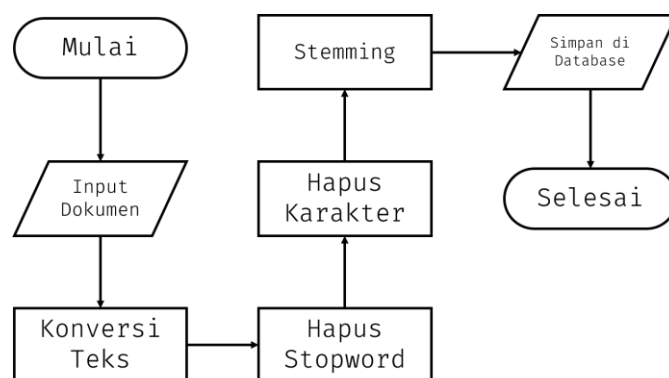
Pada Gambar 4.7, tampilan pada halaman beranda ini berdasarkan rancangan antar muka yang memiliki beberapa bagian seperti *navbar*, *welcome text*, *upload for check file*, *add and view file all user*, *add and view file all files*, *profile settings*, dan *footer*.

4.2.5 Desain Sistem

Pada bagian ini, akan menjelaskan bagaimana desain suatu sistem yang diimplementasikan. Dengan adanya desain sistem ini, dapat memberikan gambaran rancang bangun yang lengkap terhadap pengguna dari sistem ini.

4.2.5.1 Algoritma Input Dokumen ke dalam Database

Algoritma *input* dokumen ke dalam *database* merupakan salah satu tahapan dalam sistem ini di mana dokumen akan dimasukkan ke dalam *database*. *Flowchart* untuk tahapan ini dapat dilihat pada Gambar 4.8.



Gambar 4.8 Algoritma *Input* Dokumen ke dalam *Database*

Keterangan :

1. *Input* Dokumen.

Sebelum membandingkan teks yang akan ditentukan kesamaannya, terlebih dahulu dibuat sebuah repositori atau database teks yang akan digunakan sebagai pembanding nantinya. Teks yang diinput pada penelitian ini berupa *file* yang berformat pdf (*Portable Document Format*).

2. Konversi Teks.

Teks yang telah diinput, kemudian diubah ke dalam bentuk tipe data *String* agar dapat diolah nantinya di dalam sistem.

3. Hapus *Stopword*.

Stopword merupakan kata yang diabaikan karena memiliki frekuensi kemunculan yang sangat tinggi dan tidak mempunyai arti. Beberapa contoh *Stopword* seperti, "atau", "dan", dan "tapi". Dengan menghapus *Stopword*,

sistem dapat bekerja lebih cepat karena *Stopword* akan menghapus kata-kata yang tidak relevan atau yang abaikan (Rahutomo & Ririd, 2018).

4. Hapus Karakter.

Selain *Stopword*, karakter seperti ",", ".", "?", dll yang hanya berupa karakter yang tidak memiliki arti dan fungsi karena sistem hanya memeriksa kata.

5. *Stemming*

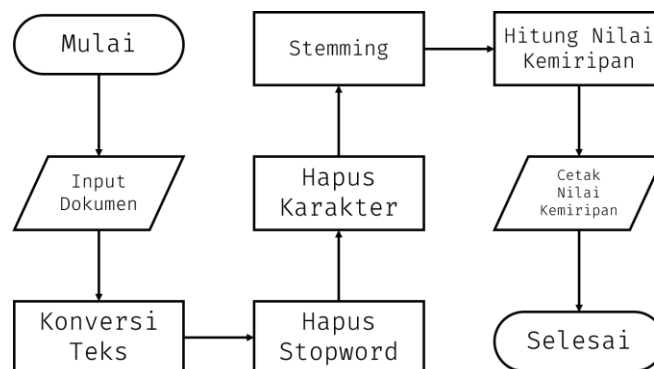
Stemming adalah proses yang dilakukan untuk mengembalikan kata berimbuhan ke bentuk dasarnya dengan cara menghilangkan imbuhan. Imbuhan tersebut berupa awalan, akhiran, awalan dan akhiran, dan sisipan (Winarti, Titin, Jati, & Sunny, 2017). Dengan adanya *stemming*, sistem dapat menghilangkan kata imbuhan menjadi kata dasar sehingga dapat mempermudah perhitungan.

6. Simpan di *Database*.

Setelah teks telah melewati tahap sebelumnya, maka teks akan dimasukkan ke dalam *database* yang nantinya akan digunakan saat ingin mendapatkan persentase kemiripan dari teks yang akan dibandingkan nantinya.

4.2.5.2 Algoritma Perbandingan Dokumen

Algoritma perbandingan dokumen merupakan salah satu tahapan dalam sistem ini di mana dokumen akan dibandingkan dengan dokumen lainnya yang sudah berada di dalam *database*. *Flowchart* untuk tahapan ini dapat dilihat pada Gambar 4.9.



Gambar 4.9 Algoritma Perbandingan Dokumen

Keterangan :

1. Input Dokumen.

Input teks yang akan dibandingkan ke dalam sistem yang nantinya akan dibandingkan terhadap teks yang terdapat dalam *database* atau repositori. Teks yang di input berupa teks yang berformat pdf (*Portable Document Format*).

2. Konversi Teks.

Teks yang telah diinput, kemudian diubah ke dalam bentuk tipe data *String* agar dapat diolah nantinya di dalam sistem.

3. Hapus *Stopword*.

Stopword merupakan kata yang diabaikan karena memiliki frekuensi kemunculan yang sangat tinggi dan tidak mempunyai arti. Beberapa contoh *Stopword* seperti, "atau", "dan", dan "tapi". Dengan menghapus *Stopword*, sistem dapat bekerja lebih cepat karena *Stopword* akan menghapus kata-kata yang tidak relevan atau yang abaikan (Rahutomo & Ririd, 2018). Dengan adanya *stemming*, sistem dapat menghilangkan kata imbuhan menjadi kata dasar sehingga dapat mempermudah perhitungan.

4. Hapus Karakter.

Selain *Stopword*, karakter seperti ",", ".", "?", dll yang hanya berupa karakter yang tidak memiliki arti dan fungsi karena sistem hanya memeriksa kata.

5. *Stemming*

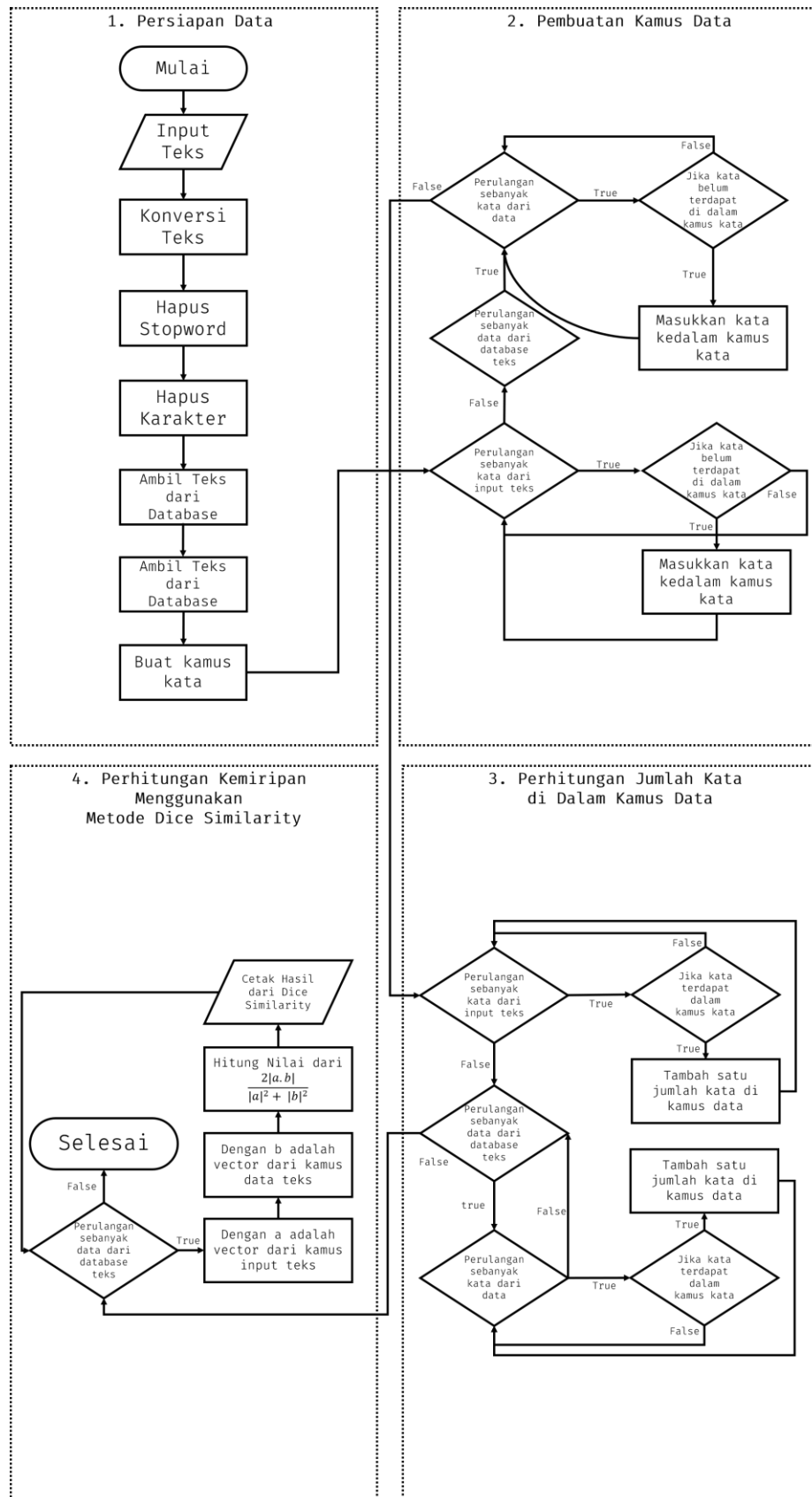
Stemming adalah proses yang dilakukan untuk mengembalikan kata berimbuhan ke bentuk dasarnya dengan cara menghilangkan imbuhan. Imbuhan tersebut berupa awalan, akhiran, awalan dan akhiran, dan sisipan (Winarti, Titin, Jati, & Sunny, 2017).

6. Hitung Nilai Kemiripan.

Setelah teks melewati tahap sebelumnya, maka teks akan dibandingkan dengan semua teks yang terdapat di dalam *database* atau repositori. Teks akan dibandingkan dengan semua data teks dengan menggunakan metode *Dice Similarity*. Kemudian hasil kemiripan antar teks akan muncul sebanyak data yang terdapat dalam *database* atau repositori teks.

4.2.5.3 Algoritma *Dice Similarity*

Algoritma perhitungan persentase kemiripan antar teks dapat dilihat pada Gambar 4.10.



Gambar 4.10 Algoritma *Dice Similarity*

1. Persiapan Data.

Sebelum dilakukan perhitungan, data disiapkan terlebih dahulu, di mana data *input* teks merupakan data yang akan dibandingkan dengan data teks yang terdapat di dalam *database* atau repositori.

2. Pembuatan Kamus Data.

Untuk mempermudah perhitungan *Dice Similarity*, dilakukan pembuatan kamus data yang berisi kata yang terdapat di dalam *input* teks maupun data teks yang berasal dari *database* atau repositori. Pada awal tahap ini, dilakukan perulangan sebanyak kata yang terdapat di dalam input teks kemudian dilakukan pengkondisian jika di dalam kamus kata tidak terdapat indeks yang sama dengan kata, maka kata tersebut ditambahkan ke dalam kamus. Setelah itu, dilakukan perulangan terhadap data dalam *database* atau repositori kemudian dilakukan perulangan kembali sebanyak kata yang terdapat di dalam teks dan dilakukan hal yang sama dengan *input* teks di awal.

3. Perhitungan Jumlah Kata di Dalam Kamus Data.

Sebelum dilakukan perhitungan, terlebih dahulu ditentukan jumlah kata yang terdapat di dalam teks untuk data teks *input* dan data teks dari *database*. Pada awal tahap ini, dilakukan perulangan sebanyak kata yang terdapat di dalam input teks kemudian dilakukan pengkondisian jika di dalam kamus kata terdapat kata yang sama dengan kata yang terdapat di kamus kata, maka kata untuk indeks *input* teks akan bertambah satu. dilakukan perulangan terhadap data dalam *database* atau repositori kemudian dilakukan perulangan kembali sebanyak kata yang terdapat di dalam teks dan dilakukan hal yang sama dengan *input* teks di awal.

4. Perhitungan Kemiripan Menggunakan Metode *Dice Similarity*.

Setelah memperoleh kamus data, maka akan dilakukan perhitungan untuk memberikan persentase kemiripan antara *input* teks dan data teks yang berasal dari *database*. Ditahap ini dilakukan perulangan sebanyak data dalam *database*. Kemudian dilakukan perhitungan sesuai dengan rumus dari *Dice Similarity* kemudian hasilnya akan ditampilkan.

4.3 Build Prototype (Bangun Prototipe)

Setelah melalui tahapan sebelumnya, maka selanjutnya adalah pembangunan prototipe untuk pembuatan program atau aplikasi berdasarkan *quick desain* sebelumnya. Untuk pembuatan *website* ini menggunakan metode *Model View Controller* (MVC).

4.3.1 Model View Controller (MVC)

Model View Controller (MVC) adalah sebuah konsep yang diperkenalkan oleh penemu *Smalltalk* (Trygve Reenskaug) untuk membuat satu jenis paket data jaringan menjadi jenis data lainya bersama dengan pemrosesan (*model*), dari proses manipulasi (*controller*) dan tampilan (*view*) untuk dipresentasikan pada sebuah *user interface* (Deacon, 2009). Untuk pembuatan aplikasi menggunakan metode MVC membagi menjadi beberapa bagian yaitu *model*, *view*, dan *controller*.

Model merupakan *code* atau sintaks yang menghubungkan antara aplikasi ke dalam *database*. *Model* biasanya berisi deklarasi variabel yang mempunyai hubungan dengan *database* seperti deklarasi tabel, koneksi, dan semacamnya. *View* merupakan *code* atau sintaks yang menampilkan bagian depan dari aplikasi. Dalam *view* terdapat stintaks yang merupakan hasil dari rancangan antar muka yang telah dibuat sebelumnya yang kemudian diimplementasikan dalam bentuk sintaks atau *code*. Sedangkan *controller* merupakan *code* atau sintaks yang berisi fungsi yang akan mengatur sistem serta sebagai penghubung antara *model* dan *view* nantinya. Dalam *controller* biasanya terdapat logika yang mengatur sistem ataupun logika yang akan menampilkan *view* dari aplikasi.

4.3.2 Implementasi Pembuatan Sistem

Implementasi pembuatan sistem ini akan menggunakan *arsitektur Model View Controller* (MVC) dengan beberapa tambahan penyesuaian untuk membuat sistem. Pembagian sistem akan dibagi menjadi beberapa bagian seperti pada konsep *Model View Controller* (MVC) dengan tambahan beberapa *folder* tambahan seperti *config*, *middleware*, *routes*, *utils*, dan sebagainya. Untuk pembagiannya dibagi sebagai berikut :

1. Model

Model merupakan *code* atau sintaks yang menghubungkan antara aplikasi ke dalam *database*. Contoh *code* atau sintaks salah satu model dalam sistem sebagai dapat dilihat pada Gambar 4.11.

```
const mongoose = require('mongoose')
const mongoosePaginate = require('mongoose-paginate-v2');
const bcrypt = require('bcryptjs')

require('../utils/db')

const userSchema = new mongoose.Schema( {
  name : {
    type : String,
    required : true,
  },
  uid : {
    type : String,
    required : true,
  },
  password : {
    type : String,
    required : true,
  },
})
userSchema.plugin(mongoosePaginate)
const User = mongoose.model('User', userSchema )
const salt = bcrypt.genSaltSync(10)

const verifyPassword = (password, h) => {
  return bcrypt.compare(password, h)
}

const createPassword = (password) => {
  return bcrypt.hashSync(password, salt)
}

module.exports = {
  User,
  userSchema,
  verifyPassword,
  createPassword
}
```

Gambar 4.11 Contoh Sintaks *Model*

Pada Gambar Gambar 4.11 merupakan *code* atau sintaks dari salah satu contoh model yaitu model User. Pada baris awal merupakan inisialisasi *library* yang digunakan yaitu *mongoose*. *Mongoose* merupakan *library* yang digunakan untuk membuat sebuah model untuk *database* MongoDB. Kemudian pada baris selanjutnya merupakan pembuatan dari model User di

mana terdapat beberapa *field* yaitu *name*, *uid*, dan *password*. Setelah mendefinisikan *field* pada model nantinya, kemudian di inisialisasi pada variabel *User* yang akan dipakai pada *controller*. Kemudian selanjutnya merupakan modul tambahan untuk mengenkripsi *password* yang akan disimpan pada *database*. Untuk baris akhir adalah kode untuk *mengexport code* agar dapat digunakan pada *code* atau *file* lain.

2. View

View merupakan *code* atau sintaks yang menampilkan bagian depan dari aplikasi. Contoh *code* atau sintaks salah satu model dalam sistem dapat dilihat pada Gambar 4.12.

```
<tr class="border-b border-sky-500">
  <td class="px-6 py-4 whitespace-nowrap text-sm font-
medium"><%= i + 1 %> </td>
  <td class="text-sm font-light px-6 py-4">
    <%= u.name %>
  </td>
  <td class="text-sm font-light px-6 py-4 whitespace-nowrap">
    <%= u.uid %>
  </td>a
  <td class="text-sm font-light px-6 py-4 whitespace-nowrap">
    <a href="" data-id="<%= u._id %> " data-uid="<%= u.uid
%>" data-name="<%= u.name %>"
      class="w-full mx-1 bg-sky-700 p-3 rounded-lg text-
white hover:bg-yellow-500 user-change"
      data-bs-toggle="modal" data-bs-
target="#addUserModal">
      <i class="fa-solid fa-pencil"></i>
    </a>
    <a href="" class="w-full mx-1 bg-sky-700 p-3 rounded-lg
text-white hover:bg-red-500 delete-user"
      data-id="<%= u._id %>" data-name="<%= u.name %> "
data-bs-toggle="modal" data-bs-target="#deleteUserModal">
      <i class="fa-solid fa-trash"></i>
    </a>
  </td>
</tr>
```

Gambar 4.12 Contoh Sintaks *View*

Pada Gambar 4.12, merupakan *code* atau sintaks dari salah satu contoh *view* yaitu *view* dari *table* *User* pada Halaman Admin. Dalam salah satu contoh

view diatas merupakan sintaks dari *file* yang memiliki format *ejs*, yaitu salah satu *templating engine* di mana dalam filenya terdapat *markup* HTML dan Javascript. Pada baris awal merupakan inisialisasi pada *table* kemudian dibuat *tag row* `<tr>`. Selanjutnya untuk kolom menggunakan *tag coloumn* `<td>`. Untuk *class* yang digunakan berasal dari *utility class* TailwindCss. Karena memiliki format *.ejs*, dalam HTML ini dapat memasukkan kode dari Javascript menggunakan tag `<%= %>` didalamnya.

3. *Controller*

Controller merupakan *code* atau sintaks yang berisi fungsi yang akan mengatur sistem serta sebagai penghubung antara *model* dan *view*. Contoh *code* atau sintaks salah satu *controller* dalam sistem dapat dilihat pada Gambar 4.13.

```

require('dotenv').config()
const {User, verifyPassword, createPassword} = require('../models/user')
const jwt = require('jsonwebtoken')
require('../utils/db')

const loginView = (req, res) => {
  res.render('../views/login-page', {
    layout : 'main',
    errors : req.flash('errors'),
  })
}

const loginAuth = async(req, res) => {
  const nim = req.body.nim
  const password = req.body.password

  if (nim === '030400' && password == '1234') {
    let admin = await User.findOne({uid : '030400'})
    req.session.userName = 'Admin'
    req.session._id = admin._id
    const token = jwt.sign({data : admin}, process.env.SECRET_KEY)
    res.cookie('x-access-token', token)
    return res.redirect('/admin')
  }
  const userLog = await User.findOne({
    uid : nim
  })

  if (!userLog) {
    req.flash('errors', 'errors')
    return res.redirect('/login')
  }

  let passwordVerify = await verifyPassword(password, userLog.password)

  if (passwordVerify) {
    const token = jwt.sign({data : userLog}, process.env.SECRET_KEY)
    res.cookie('x-access-token', token)
    return res.redirect('/')
  }
  req.flash('errors', 'errors')
  return res.redirect('/login')
}

const logout = (req, res) => {
  req.session.destroy(error => {
    res.clearCookie('x-access-token')
    res.redirect('/login')
  })
}

module.exports = {
  loginView,
  loginAuth,
  logout,
}

```

Gambar 4.13 Contoh Sintaks *Controller*

Pada Gambar 4.13 merupakan *code* atau sintaks dari salah satu contoh *controller* yaitu *controller login*. Untuk awal merupakan inisialisasi *library* yang digunakan dalam *controller* ini yaitu *JSON Web Token* (JWT) dan *model* dari *User*. Kemudian baris selanjutnya merupakan *controller* untuk *login view*, dimana fungsi ini menerima dua parameter yaitu *req* (*request*)

dan *res* (*result*). Untuk baris selanjutnya merupakan sintaks untuk menampilkan *view* dari *login-page*.

4. *Middleware*

Middleware merupakan perangkat lunak yang digunakan oleh aplikasi yang berbeda untuk berkomunikasi dengan satu sama lain. *Middleware* menyediakan fungsionalitas untuk menghubungkan aplikasi secara cerdas dan efisien sehingga. Contoh *code* atau *sintaks* dari *middleware* dalam sistem dapat dilihat pada Gambar 4.14.

```
require('dotenv').config()
const jwt = require('jsonwebtoken')

const adminVerify = (req, res, next) => {
  const token = req.cookies['x-access-token']
  if (!token) return res.redirect('/')
  const data = jwt.verify(token, process.env.SECRET_KEY)
  if (data.data.uid === '030400') return next()
  return res.redirect('/')
}

module.exports = {
  adminVerify
}
```

Gambar 4.14 Contoh Sintaks *Middleware*

Pada Gambar 4.14 merupakan *code* atau sintaks dari salah satu contoh *middleware* yaitu *middleware* admin. Untuk baris awal merupakan fungsi untuk melakukan pengecekan terhadap user apakah dia admin atau tidak. Jika dia adalah admin maka proses akan berlanjut dan jika pengguna bukan admin maka akan dikembalikan ke halaman utama atau halaman *login*.

5. *Routes*

Routes merupakan arah atau tempat yang akan dituju pada sebuah *website* yang akan diakses. Dalam *routes* ini akan memiliki beberapa rute yang nantinya akan mengatur aplikasi sesuai dengan. Contoh *code* atau sintaks dari *routes* dalam sistem dapat dilihat pada Gambar 4.15.


```

const express = require('express')
const multer = require('multer')
const fileUpload = require('express-fileupload')

const {home} = require('../controllers/homeController')
const {loginView, loginAuth, logout} = require('../controllers/loginController')
const {admin, addUser, deleteUser, updateUser, updateUserByUser, pagination,
paginationUser, paginationFile} = require('../controllers/adminController')
const {uploadPdf, deletePdf, result, resultNone} =
require('../controllers/fileController')

const {toLogin, toAddUser, toUpdateUser, toUpdateUserByUser, verifyToken} =
require('../middleware/user')
const {adminVerify} = require('../middleware/admin')
const { toAddFile } = require('../middleware/file')
const { fileStorage, fileFilter, fileStorageRes } =
require('../config/fileConfig')

const uploadFile = multer({
  storage : fileStorage,
  fileFilter : fileFilter
}).single('upload')

const uploadFileRes = multer({
  storage : fileStorageRes,
  fileFilter : fileFilter
}).single('upload-res')

const router = express.Router()

router.get('/', home)
router.post('/result', uploadFileRes, result)
router.post('/result-none', fileUpload(), resultNone)
router.put('/user', toUpdateUserByUser, updateUserByUser)

router.get('/login', loginView)
router.post('/login', toLogin, loginAuth)
router.get('/logout', logout)

router.post('/pdf', uploadFile, toAddFile, uploadPdf)
router.delete('/pdf', deletePdf)

router.get('/user', paginationUser)
router.get('/file', paginationFile)

router.use(adminVerify)
router.get('/admin', admin)
router.post('/admin-user', toAddUser, addUser)
router.put('/admin-user', toUpdateUser, updateUser)
router.delete('/admin-user', deleteUser)

module.exports = router

```

Gambar 4.15 Contoh Sintaks *Routes*

Pada Gambar 4.15 merupakan *code* atau sintaks dari salah satu contoh routes. Pada baris awal merupakan proses *import controller* dari folder controller yang ada dalam sistem. Untuk baris selanjutnya merupakan proses pembagian *routes* berdasarkan pada *controller* yang telah dibuat sebelumnya sesuai dengan *route* yang akan dihubungkan dengan *controller*.

6. *Utils*

Utils merupakan komponen tambahan yang nantinya akan dibutuhkan dalam sebuah aplikasi. Contoh *code* atau sintaks dari *utils* pada sistem dalam dilihat pada Gambar 4.16.

```
require('dotenv').config()
const mongoose = require('mongoose')
const URI = process.env.MONGODB_URL
const DB = 'latihan'

mongoose.connect(`${URI}/${DB}`, {
  useNewUrlParser : true,
  useUnifiedTopology : true,
})
```

Gambar 4.16 Contoh Sintaks *Utils*

Pada Gambar 4.16 merupakan *code* atau sintaks dari salah satu contoh *utils* yaitu *utils db*. Pada sintaks ini merupakan sintaks untuk menghubungkan *website* dengan *database*.

4.3.3 Sintaks *Dice Similarity*

Dice Similarity merupakan algoritma yang akan digunakan untuk membandingkan dokumen untuk mengetahui tingkat kesamaannya antar dokumen yang terdapat di dalam *database*. Sebelum melakukan perbandingan, sesuai dengan desain system yang telah dibuat, terlebih dahulu dilakukan pemasukan dokumen ke dalam database. Untuk sintaks dari input dokumen ke dalam database dapat dilihat pada Gambar 4.17.

```

const uploadPdf = async (req, res) => {
  const location = `${req.file.destination}/${req.file.filename}`
  const text = (await pdfParse(location)).text
  const result = sanitizePdf(text)
  const token = req.cookies['x-access-token']
  const data = jwt.verify(token, process.env.SECRET_KEY)

  const newFile = new File({
    name : req.body.title,
    fileName : req.file.filename,
    uploader : data.data._id,
    text : result,
  })

  newFile.save()
  req.flash('success-add-file', 'File Has Been Added')
  if (req.session.userName !== 'Admin') return res.redirect('/')
  return res.redirect('/admin')
}

const sanitizePdf = (text) => {
  const textFix = tokenizer.tokenize(text)
  const newData = []
  textFix.forEach(e => {
    if (e.length > 2 && !stopWord.includes(e)) newData.push(stemmer.stem(e))
  })
  return newData.join(' ')
}

```

Gambar 4.17 Sintaks *Input* Dokumen ke Dalam *Database*

Pada gambar 4.17 merupakan sintaks untuk menginput dokumen ke dalam *database*. Untuk baris awal merupakan pembuatan fungsi dari *uploadPdf* di mana fungsi ini yang akan dipanggil ketika sebuah dokumen berhasil diupload. Untuk baris selanjutnya yaitu proses mengubah *file* pdf menjadi string yang akan simpan ke dalam *database* dengan nama variabel *text*. Baris selanjutnya yaitu proses menormalkan teks dengan melakukan *stemming* dan penghapusan *stopword* menggunakan fungsi *sanitizePdf*. Pada fungsi *sanitizePdf*, dibaris awalnya merupakan proses *stemming* terhadap *text*. Selanjutnya dilakukan perulangan terhadap jumlah kata yang terdapat dalam teks kemudian jika kata tersebut termasuk *stopword* maka dihapus, jika tidak maka akan dimasukkan kedalam variabel *newData*. Jika perulangan selesai maka fungsi *sanitizePdf* akan mengembalikan *newData*. Setelah melakukan *sanitizePdf*, maka data tersebut dimasukkan ke dalam *database*.

Setelah dilakukan *input* dokumen ke dalam *database*, selanjutnya dilakukan perbandingan dokumen tersebut dengan menggunakan metode *Dice Similarity*. Untuk sintaks dari *Dice Similarity* dapat dilihat pada Gambar 4.18.

```

const data = res.data
const trainData = res.file
const user = res.user

let allData = {}
let all = []

addDict(data, all)
trainData.forEach(data => {
  addDict(data.text, all)
})
all = [...new Set(all)]
all.forEach(e => allData[e] = {})

data.split(' ').forEach(e => {
  allData[e]['s'] = (allData[e]['s'] || 0) + 1
})

trainData.forEach((str, i) => {
  str.text.split(' ').forEach(f => {
    allData[f]['s' + (i + 1)] = (allData[f]['s' + (i + 1)] || 0) + 1
  })
})

let resultAll = []
let averageSimilarity = 0
const keys = Object.keys(allData)

for (let i = 0; i < trainData.length; i++) {
  let e = 's' + (i + 1)
  let aTimesB = 0
  let aa = 0
  let bb = 0

  keys.forEach(f => {
    const a = allData[f]['s'] || 0
    const b = allData[f][e] || 0
    aTimesB += a * b
    aa += a * a
    bb += b * b
  })

  const result = (2 * aTimesB) / (aa + bb)
  resultAll.push({
    file: trainData[i],
    result: Math.round(result * 100)
  })
  averageSimilarity += Math.round(result * 100)
}
averageSimilarity /= trainData.length
averageSimilarity = Math.round(averageSimilarity)

```

Gambar 4.18 Sintaks *Dice Similarity*

Pada Gambar 4.18 merupakan sintaks untuk *Dice Similarity*. Untuk baris awal merupakan data yang akan digunakan sebagai pembanding dan data dari user yang telah diupload. Untuk variabel data merupakan *data* dari *user* yang telah diupload untuk variabel *trainData* merupakan data yang diambil dari *database* sistem. Pada baris selanjutnya merupakan variabel yang digunakan untuk menyimpan data yang akan dihitung kemiripannya di mana variabel *allData* akan menyimpan semua data

yang akan dibandingkan sedangkan variabel *all* untuk menyimpan semua kata yang akan dihitung terhadap semua data yang terdapat dalam *database*.

Untuk fungsi *addDict* merupakan fungsi untuk memasukkan kata ke dalam variabel *all*. Setelah itu merupakan baris untuk menghapus kata yang berulang dengan membuatnya ke dalam *Set* sehingga tidak ada kata yang terulang dalam variabel *all*. Setelah itu, variabel *all* akan menjadi *key* pada variabel *allData*. Tahap selanjutnya yaitu menghitung jumlah kata yang terdapat di dalam variabel *data* dan *trainData*. Setelah telah kata dari variabel *data* dan *trainData* telah dihitung maka akan dilakukan perhitungan sesuai dengan metode *Dice Similarity* pada Persamaan 2.1.

Untuk perhitungan, terlebih dahulu dilakukan perulangan sebanyak jumlah data yang terdapat pada variabel *trainData*. Kemudian dilakukan perbandingan terhadap data yang diupload oleh user dan data yang berasal dari *database*. Untuk mempermudah menghitung nilai *Dice Similarity*, dibuat terlebih dahulu beberapa variabel yaitu variabel *e*, *aTimesB*, *aa*, dan *bb*. Untuk variabel *e* merupakan urutan dari data dari variabel *dataTrain* yang akan bertambah sesuai dengan perulangannya. Selanjutnya, variabel *aTimesB* merupakan variabel untuk menyimpan nilai dari hasil perkalian vektor *a* dan *b*. Sedangkan untuk variabel *aa* dan *bb* untuk menyimpan hasil kali antara *a* dan *a*, dan juga untuk *b* dan *b*.

Setelah membuat variabel untuk menyimpan nilainya, dilakukan perhitungan *Dice Similarity* dengan cara melakukan perulangan terhadap *key* yang terdapat pada *trainData*. Untuk nilai *a* merupakan jumlah kata yang terdapat pada variabel *data* yang di awal, kemudian untuk *b* adalah variabel yang mempunyai nilai berupa jumlah kata yang sama dengan *a*. Misalkan untuk kata “rumah”, maka variabel *a* akan mempunyai nilai berupa jumlah kata rumah pada variabel *data*, sedangkan variabel *b* akan mempunyai nilai berupa jumlah kata rumah yang terdapat pada sebuah data dalam *trainData*. Selanjutnya hasil dari perhitungan akan disimpan pada variabel *resultAll* berupa hasil nilai *Dice Similarity* kemudian dikalikan dengan 100 lalu dibulatkan untuk mendapatkan nilai kemiripan antara dua data tersebut. Untuk mendapatkan nilai rata-ratanya yaitu dengan cara menjumlahkan semua nilai *Dice Similarity* tiap *trainData* kemudian dibagi dengan jumlah

trainData. Hasil *Dice Similarity* tiap-tiap dokumen nantinya akan ditampilkan ke dalam website.

4.4 User Evaluation (Evaluasi Pengguna)

Di tahap ini, sistem yang telah dibuat dalam bentuk prototipe akan dilakukan pengujian terhadap sistem yang telah dibuat. Untuk pengujian yang digunakan yaitu menggunakan metode pengujian *Black Box*. *Black Box* adalah pengujian perangkat lunak dari segi spesifikasi fungsional tanpa menguji kode atau sisi internal programnya. Artinya, hanya sisi fungsi, antarmuka, dan alurnya saja yang diuji tanpa menyentuh kode atau *script* dari perangkat lunak.

4.4.1 Rencana Pengujian

Tahap pengujian menggunakan metode Black Box akan menguji *website* yang akan dibuat dengan beberapa pengujian berdasarkan jenis pengguna yang akan mengakses *website* tersebut. Rencana pengujian secara spesifik dapat dilihat pada Tabel 4.1, Tabel 4.2, dan Tabel 4.3.

Tabel 4.1 Pengujian Pengguna *Guest*

No	Requirement Yang Diuji	Indikator	Pengujian
1	Akses Halaman Utama	Pengguna dapat mengakses halaman utama	Pengguna dapat mengakses <i>website</i>
2	Check Plagiarisme	Pengguna dapat melakukan pengecekan plagiarisme dalam <i>website</i>	Pengguna dapat melakukan pengecekan plagiarisme dengan <i>file</i> yang memiliki format <i>pdf</i>
			Pengguna tidak dapat melakukan pengecekan plagiarisme dengan <i>file</i> yang memiliki format selain <i>pdf</i>

Tabel 4.2 Pengujian Pengguna *User*

No	Requirement Yang Diuji	Indikator	Pengujian
1	Akses Halaman Utama	Pengguna (<i>User</i>) dapat mengakses halaman utama	Pengguna (<i>User</i>) dapat mengakses <i>website</i>
2	Melakukan <i>Login</i>	Pengguna (<i>User</i>) dapat melakukan <i>login</i> ke dalam <i>website</i>	Pengguna (<i>User</i>) dapat melakukan proses <i>login</i> dengan NIM dan <i>Password</i> yang benar
			Pengguna (<i>User</i>) tidak dapat melakukan proses <i>login</i> dengan NIM dan <i>Password</i> yang salah
3	Check Plagiarisme	Pengguna (<i>User</i>) dapat melakukan pengecekan plagiarisme dalam <i>website</i>	Pengguna (<i>User</i>) dapat melakukan pengecekan plagiarisme dengan <i>file</i> yang memiliki format <i>pdf</i>
			Pengguna (<i>User</i>) tidak dapat melakukan pengecekan plagiarisme dengan <i>file</i> yang memiliki format selain <i>pdf</i>
4	Manajemen Dokumen	Pengguna (<i>User</i>) dapat melakukan manajemen dokumen yang akan digunakan sebagai perbandingan	Pengguna (<i>User</i>) dapat menambahkan dokumen yang memiliki format <i>pdf</i>
			Pengguna (<i>User</i>) tidak dapat menambahkan dokumen yang memiliki format selain <i>pdf</i>
			Pengguna (<i>User</i>) dapat menghapus

No	Requirement Yang Diuji	Indikator	Pengujian
			dokumen yang telah ditambahkan sebelumnya

Tabel 4.3 Pengujian Pengguna Admin

No	Requirement Yang Diuji	Indikator	Pengujian
1	Akses Halaman Utama	Pengguna (<i>Admin</i>) dapat mengakses halaman utama	Pengguna (<i>Admin</i>) dapat mengakses <i>website</i>
2	Melakukan <i>Login</i>	Pengguna (<i>Admin</i>) dapat melakukan <i>login</i> ke dalam <i>website</i>	Pengguna (<i>Admin</i>) dapat melakukan proses <i>login</i> dengan NIM dan <i>Password</i> yang benar
			Pengguna (<i>Admin</i>) tidak dapat melakukan proses <i>login</i> dengan NIM dan <i>Password</i> yang salah
3	Check Plagiarisme	Pengguna (<i>Admin</i>) dapat melakukan pengecekan plagiarisme dalam <i>website</i>	Pengguna (<i>Admin</i>) dapat melakukan pengecekan plagiarisme dengan <i>file</i> yang memiliki format <i>pdf</i>
			Pengguna (<i>Admin</i>) tidak dapat melakukan pengecekan plagiarisme dengan <i>file</i> yang memiliki format selain <i>pdf</i>
4	Manajemen Dokumen	Pengguna (<i>Admin</i>) dapat melakukan manajemen dokumen	Pengguna (<i>Admin</i>) dapat menambahkan dokumen yang

No	Requirement Yang Diuji	Indikator	Pengujian
		yang akan digunakan sebagai perbandingan	memiliki format <i>pdf</i>
			Pengguna (Admin) tidak dapat menambahkan dokumen yang memiliki format selain <i>pdf</i>
			Pengguna (Admin) dapat menghapus dokumen yang telah ditambahkan sebelumnya
5	Manajemen User	Pengguna (<i>Admin</i>) dapat melakukan manajemen <i>user</i>	Pengguna (Admin) dapat membuat <i>user</i> baru
			Pengguna (Admin) tidak dapat membuat <i>user</i> baru sesuai ketentuan
			Pengguna (Admin) dapat mengubah <i>user</i> yang sudah ada
			Pengguna (Admin) dapat menghapus <i>user</i> yang sudah ada

4.4.2 Tahap Pengujian

Setelah membuat rencana pengujian, selanjutnya dilakukan pengujian berdasarkan rencana pengujian yang telah dibuat. Untuk pengujian *website* dilakukan secara bertahap sesuai dengan rencana pengujian yang sudah ada. Dalam pengujian, nantinya akan didapat hasil berdasarkan pengujian apakah sesuai dengan pengujian yang dilakukan atau tidak. Jika tidak sesuai, nantinya akan dilakukan evaluasi sesuai rencana pengujiannya dan akan dilakukan perbaikan agar hasil pengujian yang dilakukan sesuai dengan yang diharapkan.

Untuk pengujian pertama yaitu pengujian pada pengguna *guest* yang sesuai pada Tabel 4.1. Berikut hasil pengujian untuk pengguna *guest* pada Tabel 4.4

Tabel 4.4 Hasil Pengujian Pengguna *Guest*

No	Requirement Yang Diuji	Indikator	Pengujian	Hasil Pengujian
1	Akses Halaman Utama	Pengguna dapat mengakses halaman utama	Pengguna dapat mengakses <i>website</i>	Sesuai
2	Check Plagiarisme	Pengguna dapat melakukan pengecekan plagiarisme dalam <i>website</i>	Pengguna dapat melakukan pengecekan plagiarisme dengan <i>file</i> yang memiliki format <i>pdf</i>	Sesuai
			Pengguna tidak dapat melakukan pengecekan plagiarisme dengan <i>file</i> yang memiliki format selain <i>pdf</i>	Sesuai

Pengujian selanjutnya yaitu pengujian pada pengguna *User* yang sesuai dengan pada Tabel 4.2. Berikut hasil pengujian untuk pengguna *User* pada Tabel 4.5.

Tabel 4.5 Hasil Pengujian Pengguna *User*

No	Requirement Yang Diuji	Indikator	Pengujian	Hasil Pengujian
1	Akses Halaman Utama	Pengguna (<i>User</i>) dapat mengakses halaman utama	Pengguna (<i>User</i>) dapat mengakses <i>website</i>	Sesuai
2	Melakukan <i>Login</i>	Pengguna (<i>User</i>) dapat	Pengguna (<i>User</i>) dapat	Sesuai

No	Requirement Yang Diuji	Indikator	Pengujian	Hasil Pengujian
		melakukan <i>login</i> ke dalam <i>website</i>	melakukan proses <i>login</i> dengan NIM dan <i>Password</i> yang benar	
			Pengguna (User) tidak dapat melakukan proses <i>login</i> dengan NIM dan <i>Password</i> yang salah	Sesuai
3	Check Plagiarisme	Pengguna (User) dapat melakukan pengecekan plagiarisme dalam <i>website</i>	Pengguna (User) dapat melakukan pengecekan plagiarisme dengan <i>file</i> yang memiliki format <i>pdf</i>	Sesuai
			Pengguna (User) tidak dapat melakukan pengecekan plagiarisme dengan <i>file</i> yang memiliki format selain <i>pdf</i>	Sesuai
4	Manajemen Dokumen	Pengguna (User) dapat melakukan manajemen dokumen yang akan digunakan sebagai perbandingan	Pengguna (User) dapat menambahkan dokumen yang memiliki format <i>pdf</i>	Sesuai
			Pengguna (User) tidak dapat menambahkan dokumen yang	Sesuai

No	Requirement Yang Diuji	Indikator	Pengujian	Hasil Pengujian
			memiliki format selain <i>pdf</i>	
			Pengguna (User) dapat menghapus dokumen yang telah ditambahkan sebelumnya	Sesuai

Pengujian terakhir yaitu pengujian pada pengguna *Admin* yang sesuai dengan pada Tabel 4.3. Berikut hasil pengujian untuk pengguna *Admin* pada Tabel 4.6.

Tabel 4.6 Hasil Pengujian Pengguna *Admin*

No	Requirement Yang Diuji	Indikator	Pengujian	Hasil Pengujian
1	Akses Halaman Utama	Pengguna (<i>Admin</i>) dapat mengakses halaman utama	Pengguna (<i>Admin</i>) dapat mengakses <i>website</i>	Sesuai
2	Melakukan <i>Login</i>	Pengguna (<i>Admin</i>) dapat melakukan <i>login</i> ke dalam <i>website</i>	Pengguna (<i>Admin</i>) dapat melakukan proses <i>login</i> dengan NIM dan <i>Password</i> yang benar	Sesuai
			Pengguna (<i>Admin</i>) tidak dapat melakukan proses <i>login</i> dengan NIM dan <i>Password</i> yang salah	Sesuai
3	<i>Check Plagiarisme</i>	Pengguna (<i>Admin</i>) dapat melakukan pengecekan	Pengguna (<i>Admin</i>) dapat melakukan pengecekan plagiarisme	Sesuai

No	Requirement Yang Diuji	Indikator	Pengujian	Hasil Pengujian
		plagiarisme dalam <i>website</i>	dengan <i>file</i> yang memiliki format <i>pdf</i>	
			Pengguna (Admin) tidak dapat melakukan pengecekan plagiarisme dengan <i>file</i> yang memiliki format selain <i>pdf</i>	Sesuai
4	Manajemen Dokumen	Pengguna (Admin) dapat melakukan manajemen dokumen yang akan digunakan sebagai perbandingan	Pengguna (Admin) dapat menambahkan dokumen yang memiliki format <i>pdf</i>	Sesuai
			Pengguna (Admin) tidak dapat menambahkan dokumen yang memiliki format selain <i>pdf</i>	Sesuai
			Pengguna (Admin) dapat menghapus dokumen yang telah ditambahkan sebelumnya	Sesuai
5	Manajemen User	Pengguna (Admin) dapat melakukan manajemen <i>user</i>	Pengguna (Admin) dapat membuat <i>user</i> baru	Sesuai

No	Requirement Yang Diuji	Indikator	Pengujian	Hasil Pengujian
			Pengguna (Admin) tidak dapat membuat <i>user</i> baru sesuai ketentuan	Sesuai
			Pengguna (Admin) dapat mengubah <i>user</i> yang sudah ada	Sesuai
			Pengguna (Admin) dapat menghapus <i>user</i> yang sudah ada	Sesuai

Setelah diuji berdasarkan rencana pengujian, didapatkan hasil yang sesuai dengan hasil pengujian yang diharapkan. Selanjutnya yaitu pengujian secara langsung di dalam *website*. Dalam pengujian ini, menggunakan jurnal yang dimasukkan ke dalam sistem sebanyak 40 jurnal yang memiliki judul atau tema mengenai teknologi yang akan digunakan sebagai pembandingan. Untuk jurnal yang digunakan dapat dilihat pada lempira. Sedangkan jurnal yang digunakan untuk dibandingkan yaitu jurnal dengan judul “*Implementasi Deteksi Plagiarisme Menggunakan Metode N-Gram dan Jaccard Similarity Terhadap Algoritma Winnowing*” oleh Sunardi, Anton Yudhanam dan Lif Alfiatul Mukaromah dari Universitas Ahmad Dahlan . Untuk jurnal yang digunakan sebagai pembandingan dapat dilihat dalam lampiran. Untuk hasil pengujian dapat dilihat pada gambar 4.19

Plagiarsm Checker

Home About Me How It Works Login

Plagiarism Checkers

Website Ini Menggunakan Metode Dice Similarity Untuk Mempersentasikan Kemiripan Dokumen atau Teks Anda dengan Dokumen yang Terdapat di Database

Input Your Text

or Upload Your File (*.pdf) Choose File [[Testing]] Im...innowing.pdf

Check Plagiarism

Result

1 of 6

TRANSMISI, 26, (1), JULI 2018, p-ISSN 1411-8814 e-ISSN 2487-6422

IMPLEMENTASI DETEKSI PLAGIARISME MENGGUNAKAN METODE N-GRAM DAN JACCARD SIMILARITY TERHADAP ALGORITMA WINNOWING

Samar¹⁾, Anton Yudhan²⁾, dan Sri Alifanul Mubandani³⁾

Program Studi Teknik Elektro, Universitas Ahmad Dahlan^{1,2)}
Program Studi Magister Teknik Informatika, Universitas Ahmad Dahlan³⁾
Kampus 1 UIN 26, Pleris, Yogyakarta, 55161, Indonesia 1, 2, 3
E-Mail: Samar@uii.ac.id¹⁾, ayudhan@uii.ac.id²⁾, alifan100@gmail.com³⁾

Abstrak

Penelitian ini bertujuan untuk mendeteksi plagiarisme pada dokumen yang telah diunggah ke sistem yang berbasis web. Dokumen yang telah diunggah ke sistem akan dibandingkan dengan dokumen yang telah ada di database. Dokumen yang telah diunggah ke sistem akan dibandingkan dengan dokumen yang telah ada di database. Dokumen yang telah diunggah ke sistem akan dibandingkan dengan dokumen yang telah ada di database.

Kata Kunci: N-gram, Jaccard Similarity, Algoritma Winnowing

Details

File Name : [[Testing]] Implementasi Deteksi Plagiarisme Menggunakan Metode N-Gram dan Jaccard Similarity Terhadap Algoritma Winnowing.pdf

File Size : 886.07 Kb

Character : 23.821

Words : 4.706

Plagism Percentage

Unique : 87 % Plagiarism : 13 %

87 % 13 %

File With High Similarity

#	Title Pdf	Uploader	Similarity	Show
1	Rancang Bangun Website untuk Memeriksa Plagiat E-Journal	Lionel Messi	32 %	Show
2	[Random] Blackbox Testing Pada Boundary Value Analysis Pada Aplikasi Submission System	Sergio Ramos	19 %	Show
3	Pengembangan Instrumen Pengukuran Aliran Air Berbasis Internet Oof Things (IOT)	Cristiano Ronaldo	18 %	Show
4	Portal Publikasi Jurnal Karya Ilmiah Program Studi	Lionel Messi	17 %	Show
5	Penerapan Metode Prototipe Pada Perancangan Sistem Informasi Penggajian Karyawan (Persis Gawan) Berbasis Web	Sergio Ramos	17 %	Show
6	Pengujian Black Box pada Aplikasi Sistem Informasi Akademik Menggunakan Teknik Equivalence Partitioning	Sergio Ramos	16 %	Show
7	Rancang Bangun Sistem Informasi Evaluasi Siswa dan Kehadiran Guru Berbasis Web (Studi Kasus di SMK Nusa Putra Kota Tangerang)	Lionel Messi	16 %	Show
8	Aplikasi Mobile Driver Online Berbasis Android Untuk Perusahaan Rental Kendaraan	Admin	16 %	Show
9	Pengujian Black Box pada Aplikasi Sistem Seleksi Sales Terbaik Menggunakan Teknik Equivalence Partitions	Sergio Ramos	15 %	Show
10	Pengujian Black Box Testing Pada Aplikasi Action & Strategy Berbasis Android dengan Teknologi Phonegap	Sergio Ramos	14 %	Show

© 2023 Copyright : Aris Akhyar Abdillah

Gambar 4.19 Pengujian Langsung Menggunakan Jurnal

57

Pada gambar 4.19, dapat dilihat langsung hasil perhitungan yang dilakukan menggunakan metode *Dice Similarity*. Pada bagian kiri yaitu dokumen yang akan dibandingkan kemudian selanjutnya di sebelah kanan yaitu detail tentang dokumen yang telah diupload. Pada bagian ini, dapat dilihat tingkat *similarity* dari dokumen yang dibandingkan terhadap dokumen yang terdapat dalam sistem yaitu sebesar 13%. Untuk bagian selanjutnya yaitu dokumen yang memiliki kemiripan paling tinggi dengan dokumen yang diupload. Untuk dokumen yang ditampilkan berjumlah 10 dokumen yang memiliki tingkat kesamaan yang tinggi dengan dokumen yang diupload oleh *user*.

4.5 Refining Prototype (Memperbaiki Prototipe)

Setelah melakukan tahap sebelumnya, maka selanjutnya akan dilakukan perbaikan prototipe jika masih terdapat pengujian yang tidak sesuai dengan apa yang diharapkan. Proses ini akan terus dilakukan sampai sudah tidak ada lagi pengujian yang tidak sesuai dengan yang diharapkan.

Berdasarkan hasil pengujian pada Tabel 4.4, Tabel 4.5, dan Tabel 4.6, hasil pengujian yang diperoleh sesuai dengan rencana pengujian yang dibuat sehingga untuk memperbaiki prototipe ini tidak dilakukan perbaikan. Sehingga pada tahap ini tidak dilakukan.

4.6 Implement Product and Maintain (Implementasi dan Pemeliharaan)

Pada fase akhir ini, aplikasi akan segera diselesaikan berdasarkan prototipe akhir. Selanjutnya adalah fase pemeliharaan agar sistem berjalan lancar tanpa kendala pada waktu ke depannya.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan dan pembahasan sebelumnya, dapat diambil kesimpulan yaitu Aplikasi Pendeteksi Kesamaan Teks Berbasis *Website* dengan Mengimplementasikan Metode *Dice Similarity* telah selesai dirancang menggunakan metode *Prototype* kemudian telah diuji menggunakan beberapa dokumen dengan hasil yang sesuai.

5.2 Saran

Berdasarkan penelitian yang dilakukan, maka untuk penelitian selanjutnya agar dapat dilakukan pengecekan dokumen berdasarkan jenis jurnal yang terdapat di dalam sistem untuk memperoleh kesamaan yang lebih akurat serta waktu perhitungan yang digunakan lebih sedikit.

DAFTAR PUSTAKA

- Agung, G. (2000). *Microsoft Frontpage 2000 Webbot*. Jakarta: PT. Elex Media Komputindo.
- Andaru, A. (2018). Pengertian Database Secara Umum. *Fakultas Komputer Umitra*.
- Anonymous. (2022, July 15). *Express JS*. Diambil kembali dari Express - Node.js web application framework: <https://expressjs.com/>
- Anonymous. (2022, July 15). *NodeJS*. Diambil kembali dari <https://nodejs.dev/>
- Anonymous. (2022, July 15). *What Is MongoDB ?* Diambil kembali dari MongoDB: <https://www.mongodb.com/what-is-mongodb>
- Arista, R. F., & Listyani, R. H. (2015). Plagiarisme di Kalangan Mahasiswa. *Paradigma. Volume 03 Nomor 02*.
- Bianto, M. A., Rahayu, S., Huda, M., & Kusriani. (2018). Perancangan Sistem Pendeteksi Plagiarisme Terhadap Topik Penelitian Menggunakan Metode K-Means Clustering dan Model Bayesian. *Seminar Nasional Teknologi Informasi dan Multimedia*.
- Deacon, j. (2009). Model-View-Controller (MVC) Architecture. *JOHN DEACON Computer Systems Development, Consulting & Training*.
- Dhamayanti, & Sari, L. P. (2019). Aplikasi Pendeteksi Plagiasi pada Universitas Indo Global Mandiri Berbasis Web. *Jurnal Ilmiah Informatika Global Volume 10 No. 02*.
- Fikri, A. D. (2019). Perbandingan Metode Dice Similarity Dengan Cosine Similarity Menggunakan Query Expansion Pada Pencarian Ayatul Ahkam Dalam Terjemah Al Quran Berbahasa Indonesia .
- Firmansyah, F. A. (2019, September 22). *NESABAMEDIA*. Diambil kembali dari Pengertian HTTPS Beserta Fungsi dan Perbedaannya dengan HTTP: <https://www.nesabamedia.com/pengertian-https/>
- Hasanah, U., & Mutiara, D. A. (2019). Perbandingan Metode Cosine Similarity dan Jaccard Similarity Untuk Penilaian Otomatis Jawaban Pendek. *Seminar Nasional Sistem Informasi dan Teknik Informatika SENSITif*.
- Khontoro, C., Andjarwirawan, J., & Yulia. (2021). Penerapan Algoritma TextRank dan Dice Similarity Untuk Verifikasi Berita Hoax. *Jurnal Infra Vol 9, No 1*.
- Lopes, F. M. (2013). Penegakan Hukum Pelanggaran Terhadap Hak Cipta Dibiidang Musik dan Lagu. *Lex Privatum, Vol.I/No.2/Apr-Jun/2013*.
- Pratama, R. P. (2018). *Aplikasi Pendeteksi Plagiarisme Menggunakan Cosine Similarity*. Universitas Islam Negeri Maulana Malik Ibrahim.

- Pratama, R. P., Faisal, M., & Hanani, A. (2019). Deteksi Plagiarisme Pada Artikel Jurnal Menggunakan Metode Cosine Similarity . *SMARTICS Journal*, Vol.5 No. 1 .
- Priambodo, J. (2018). Pendeteksian Plagiarisme Menggunakan Algoritma Rabin - Karp dengan Metode Rolling Hash. *Jurnal Informatika Universitas Pamulang* Vol. 3, No. 1.
- Rahutomo, F., & Ririd, A. R. (2018). Evaluasi Daftar Stopword Bahasa Indonesia. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*.
- Ridhatillah, A. (2003). Dealing with Plagiarism in the Information System Research Community: A Look at Factors That Drive Plagiarism and Ways to Address Them. *MIS Quarterly*, Vol.27, No. 4, p. 511-532.
- Rusydi, I. (2014). Pemanfaatan E-Journal Sebagai Media Informasi Digital. *Jurnal Iqra'* 8(2).
- Sasongko, J., & Diartono, D. A. (2009). Rancang Bangun Sistem Informasi Manajemen Surat. *Jurnal Teknologi Informasi Dinamik*.
- Sridevi, S. (2014). User Interface Design. *International Journal of Computer Science and Information Technology Research*.
- Stefanovi, P., Kurasova, O., & Štrimaitis, R. (2019). The N-Grams Based Text Similarity Detection Approach Using Self-Organizing Maps and Similarity Measures. *Applied Science*.
- Sunardi, Yudhana, A., & Mukaromah, I. A. (2018). Implementasi Deteksi Plagiarisme Menggunakan Metode N-Gram dan Jaccard Similarity Terhadap Algoritma Winnowing. *TRANSMISI*, 20, (3).
- Susanto, A. (2009). *Pengenalan Komputer*. Bekasi: IlmuKomputer.com.
- Tala, F. Z. (2003). A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia. *Master of Logic Project Institute for Logic, Language and Computation Universiteit van Amsterdam The Netherlands*.
- Wibowo, A. (2012). Mencegah dan Menaggulangi Plagiarisme di Dunia Pendidikan. *Jurnal Kesehatan Masyarakat Nasional* Vol. 6, No. 5.
- Winarti, Titin, Jati, K., & Sunny, A. (2017). Determining Term on Text Document Determining Term on Text Document. *International Journal of Computer Application* 157 (9):6.

LAMPIRAN***Lampiran 1 (Source Code)***config/fileConfig.js

```

const multer = require('multer')
const fileStorage = multer.diskStorage({
  destination : (req, file, cb) => {
    cb(null, 'public/docs')
  },
  filename : (req, file, cb) => {
    cb(null, new Date().getTime() + '-' + file.originalname)
  }
})
const fileStorageRes = multer.diskStorage({
  destination : (req, file, cb) => {
    cb(null, 'public/result')
  },
  filename : (req, file, cb) => {
    cb(null, new Date().getTime() + '-' + file.originalname)
  }
})

const fileFilter = (req, file, cb) => {
  // console.log(file.mimetype)
  if (file.mimetype == 'application/pdf') cb(null, true)
  else cb(null, false)
}

module.exports = {
  fileStorage,
  fileStorageRes,
  fileFilter
}

```

-controllers/adminController.js

```

require('../utils/db')
const jwt = require('jsonwebtoken')
const {User, createPassword} = require('../models/user')
const {File} = require('../models/files')

```

```

const admin = async (req, res) => {
  const allUser = await User.find()
  const allFile = await File.find()
  // res.send(allUser)
  res.render('../views/admin-page', {
    layout : 'main',
    user : req.session.userName,
    allUser : allUser,
    allFile : allFile,
    errorsUser : req.flash('errors-add-user'),

```

```

        successUser : req.flash('success-add-user'),
        errorsFile : req.flash('errors-add-file'),
        successFile : req.flash('success-add-file'),
    })
}

const addUser = async (req, res) => {
    const nim = req.body.nim
    const userName = req.body.name
    const password = req.body.password
    const newUser = await new User({
        uid : nim,
        name : userName,
        password : createPassword(password)
    })
    newUser.save()
    req.flash('success-add-user', 'New User Has Been Added')
    return res.redirect('/admin')
}

const updateUser = async (req, res) => {
    const name = req.body.name
    const uid = req.body.nim
    // const user = await User.findOne({uid : uid})

    await User.updateOne({
        uid : uid
    },
    {
        $set : {
            uid : uid,
            name : name,
        }
    })

    req.flash('success-add-user', 'User Has Been Updated')
    return res.redirect('/admin')
}

const updateUserByUser = async (req, res) => {
    const name = req.body.name
    const uid = req.body.nim
    const password = req.body.password
    // return res.send(req.body)
    // const user = await User.findOne({uid : uid})
    const update = {
        uid : uid,
        name : name,
    }
    if (password.length !== 0) update.password =
    createPassword(password)
}

```

```

    await User.updateOne({
      _id : req.body._id
    },
    {
      $set : update
    })

    const newUserUpdate = await User.findOne({_id : req.body._id})
    req.session.details = newUserUpdate

    req.flash('success-update-user', 'User Has Been Updated')
    return res.redirect('/')
  }

const deleteUser = async (req, res) => {
  // return res.send(req.body)
  const id = req.body.id
  const name = req.body.name
  await User.deleteOne({_id : id})
  req.flash('success-add-user', `User ${name} Has Been Deleted`)
  return res.redirect('/admin')
}

const paginationUser = async (req, res) => {
  const {limit : lim, pages} = req.query
  const {limit, offset} = getPagination(pages, lim)

  const data = await User.paginate({}, {offset, limit})
  const result = {
    totalItems : data.totalDocs,
    data : data.docs,
    totalPages : data.totalPages,
    currentPage : data.page - 1,
  }
  return res.send(result)
}

const allUser = async (req, res) => {
  const allUser = await User.find({}, {_id : 1, name : 1})

  return res.send(allUser)
}

const paginationFile = async (req, res) => {
  let data
  const {limit : lim, pages} = req.query
  const {limit, offset} = getPagination(pages, lim)

  const token = req.cookies['x-access-token']
  const dataUser = jwt.verify(token, process.env.SECRET_KEY)

```

```

    // console.log(dataUser.data)
    if (dataUser.data.uid === '030400') data = await
File.paginate({}, {offset, limit})
    else data = await File.paginate({uploader : dataUser.data._id},
{offset, limit})

    const result = {
      totalItems : data.totalDocs,
      data : data.docs,
      totalPages : data.totalPages,
      currentPage : data.page - 1,
    }
    return res.send(result)
  }

const getPagination = (page, size) => {
  const limit = size ? +size : 3
  const offset = page ? page * limit : 0
  return { limit, offset }
}

module.exports = {
  admin,
  addUser,
  updateUser,
  deleteUser,
  updateUserByUser,
  paginationUser,
  paginationFile,
  allUser
}

controllers/fileController.js
require('dotenv').config()
const pdfParse = require('pdf-parse')
const {File, sanitizePdf} = require('../models/files')
const {promisify} = require('util')
const fs = require('fs')
const { User } = require('../models/user')
const jwt = require('jsonwebtoken')

const allChar = ["-", " ", ""]

const unlink = promisify(fs.unlink)

const uploadPdf = async (req, res) => {
  const location =
`${req.file.destination}/${req.file.filename}`
  const text = (await pdfParse(location)).text

```

```

const result = sanitizePdf(text)
const token = req.cookies['x-access-token']
const data = jwt.verify(token, process.env.SECRET_KEY)

const newFile = new File({
  name : req.body.title,
  fileName : req.file.filename,
  uploader : data.data._id,
  text : result,
})

await newFile.save()
req.flash('success-add-file', 'File Has Been Added')
if (req.session.userName !== 'Admin') return res.redirect('/')
return res.redirect('/admin')
}

const deletePdf = async (req, res) => {
  // res.send(req.body)
  const id = req.body.id
  const oldFile = await File.findOne({_id : id})
  // return res.send(oldFile.fileName)
  fs.unlink(`public/docs/${oldFile.fileName}`, async(err) => {
    await File.deleteOne({_id : id})
    req.flash('success-add-file', 'File Has Been Deleted')
    return res.redirect('/admin')
  })
}

const resultNone = async (req, res) => {
  const allUser = await User.find({}, {_id : 1, name : 1})
  const allFile = await File.find({}, {name : 1, text : 1,
uploader : 1, fileName : 1})

  const location = req.files['upload-res']
  const text = (await pdfParse(location)).text
  const resultText = sanitizePdf(text)
  // console.log(resultText)
  const dataFile = {
    name : req.files['upload-res'].name,
    char : text.length,
    word : text.split(' ').length,
    size : req.files['upload-res'].size,
  }

  const result = {
    fileUpload : dataFile,
    data : resultText,
    user : allUser,
    file : allFile,
  }
}

```



```

    res.send(result)
  }

const result = async (req, res) => {
  const allUser = await User.find({}, {_id : 1, name : 1})
  const allFile = await File.find({}, {name : 1, text : 1,
uploader : 1, fileName : 1})
  let result
  if (req.body.text !== '') {
    result = sanitizePdf(req.body.text)
  } else {
    const location = req.file.path
    const text = (await pdfParse(location)).text
    result = sanitizePdf(text)
  }

  let allData = {}

  const retirve = {
    fileInformation : req.file,
    resultAll : [],
  }

  retirve.fileInformation.length = separator(result.length)
  retirve.fileInformation.word = separator(result.split('
').length)

  addDict(result, allData)
  allFile.forEach(files => {
    addDict(files.text, allData)
  })

  for (const prop in allData) {
    if (result.split(' ').includes(prop)) {
      let index = 0
      result.split(' ').forEach(x => {
        if (prop === x) index += 1
      })
      allData[prop]["s"] = index
    } else {
      allData[prop]["s"] = 0
    }
  }

  allFile.forEach((e, i) => {
    if (e.text.split(' ').includes(prop)) {
      let index = 0
      e.text.split(' ').forEach(x => {
        if (prop === x) index += 1
      })
      allData[prop]["s" + (i + 1)] = index
    } else {

```

```

        allData[prop]["s" + (i + 1)] = 0
      }
    })
  }

  const objectData = Object.keys(allData)[0]
  const varObject = Object.keys(allData[objectData])
  varObject.shift()

  varObject.forEach((e, i) => {
    allFile[i].text = ''
    let aTimesB = 0
    let aa = 0
    let bb = 0

    Object.keys(allData).forEach(f => {
      aTimesB += allData[f]['s'] * allData[f][e]
      aa += allData[f]['s'] * allData[f]['s']
      bb += allData[f][e] * allData[f][e]
    })

    const resultSimilarity = (2 * aTimesB) / (aa + bb)
    // console.log(`Similarity Data ${i + 1} Is
    ${Math.round(resultSimilarity * 100)} %`)
    const resForRetervie = {
      file : allFile[i],
      result : Math.round(resultSimilarity * 100)
    }
    retirve.resultAll.push(resForRetervie)

  })

  retirve.resultAll.sort((a, b) => (a.result < b.result) ? 1 : -
1)

  let resultPlag = 0
  retirve.resultAll.forEach(e => {
    resultPlag += e.result
  })
  retirve.plag = Math.round(resultPlag /
retirve.resultAll.length)
  retirve.resultAll = retirve.resultAll.filter(file => {
    return file.result > 50
  })

  await unlink(location)

```

```

    res.render('../views/result-page', {
      layout : 'main',
      user : allUser,
      result : retire
    })
  }

function addDict(datax, to) {
  datax.toLowerCase().split(' ').forEach(e => {
    e.trim()
    if (!to.hasOwnProperty(e)) to[e] = {}
  })
}

function separator(num) {
  var str = num.toString().split('.');
  str[0] = str[0].replace(/\B(?=(\d{3})+(?!\d))/g, '.');
  return str.join('.');
}

module.exports = {
  uploadPdf,
  deletePdf,
  result,
  resultNone,
}

controllers/homeController.js
const { File } = require('../models/files')
const jwt = require('jsonwebtoken')
require('dotenv').config()

require('../utils/db')

const home = async (req, res) => {
  let user, fileUser
  const token = req.cookies['x-access-token']
  if (token) user = jwt.verify(token, process.env.SECRET_KEY)
  if (user?.data?._id) fileUser = await File.find({uploader :
user?.data?._id})
  res.render('../views/home-page', {
    layout : 'main',
    user : user?.data?.name,
    details : user?.data,
    errorsUpdate : req.flash('errors-update-user'),
    successUpdate : req.flash('success-update-user'),
    errorsFile : req.flash('errors-add-file'),
    successFile : req.flash('success-add-file'),
    file : fileUser
  })
}

```

```

}

module.exports = {
  home
}

-controllers/loginController.js
require('dotenv').config()
const {User, verifyPassword, createPassword} =
require('../models/user')
const jwt = require('jsonwebtoken')
require('../utils/db')

const loginView = (req, res) => {
  return res.render('../views/login-page', {
    layout : 'main',
    errors : req.flash('errors'),
  })
}

const loginAuth = async(req, res) => {
  const nim = req.body.nim
  const password = req.body.password

  if (nim === '030400' && password == '1234') {
    let admin = await User.findOne({uid : '030400'})
    if (!admin) {
      const newAdmin = new User({
        uid : '030400',
        name : 'admin',
        password : createPassword('1234')
      })
      await newAdmin.save()
      admin = await User.findOne({uid : '030400'})
    }
    req.session.userName = 'Admin'
    req.session._id = admin._id
    const token = jwt.sign({data : admin},
process.env.SECRET_KEY)
    res.cookie('x-access-token', token)
    return res.redirect('/admin')
  }
  const userLog = await User.findOne({
    uid : nim
  })

  if (!userLog) {
    req.flash('errors', 'errors')
    return res.redirect('/login')
  }
}

```

```

    let passwordVerify = await verifyPassword(password,
userLog.password)

    if (passwordVerify) {
        const token = jwt.sign({data : userLog},
process.env.SECRET_KEY)
        res.cookie('x-access-token', token)
        return res.redirect('/')
    }
    req.flash('errors', 'errors')
    return res.redirect('/login')
}

const logout = (req, res) => {
    req.session.destroy(error => {
        res.clearCookie('x-access-token')
        res.redirect('/login')
    })
}

module.exports = {
    loginView,
    loginAuth,
    logout,
}

```

middleware/admin.js

```

require('dotenv').config()
const jwt = require('jsonwebtoken')

const adminVerify = (req, res, next) => {
    const token = req.cookies['x-access-token']
    if (!token) return res.redirect('/')
    const data = jwt.verify(token, process.env.SECRET_KEY)
    if (data.data.uid === '030400') return next()
    return res.redirect('/')
}

module.exports = {
    adminVerify
}

-middleware/file.js
const {check, validationResult} = require('express-validator')
const fs = require('fs')

const toAddFile = [
    check('title').isLength({min : 4}),
    (req, res, next) => {
        const errors = validationResult(req)

```

```

        if (errors.isEmpty()) return next()
        req.flash('errors-add-file', 'Failed to Add New File')
        if (typeof req.file === 'undefined') {
            return res.redirect('/admin')
        }
        fs.unlink(`public/docs/${req.file.filename}`, (error) => {
            return res.redirect('/admin')
        })
    }
}

module.exports = {
    toAddFile
}

-middlewares/user.js
require('dotenv').config()
const jwt = require('jsonwebtoken')
const {check, validationResult} = require('express-validator')
const {User} = require('../models/user')

const toLogin = [
    [
        check('nim').isLength({min : 10}),
        check('password').isLength({min : 5}),
        (req, res, next) => {
            const errors = validationResult(req)
            if (errors.isEmpty()) return next()
            if (errors.array()[0].value === '030400' &&
errors.array()[0].param === 'nim') return next()
            req.flash('errors', 'Your Nim or Password is Wrong')
            return res.redirect('/login')
        }
    ],
]

const toAddUser = [
    check('nim').isLength({min : 4}).custom(async (value) => {
        const isNimExist = await User.findOne({uid : value})
        if (isNimExist) throw new Error('NIM is Already exist')
        return true
    }),
    check('name').isLength({min : 4}),
    check('password').isLength({min : 5}),
    (req, res, next) => {
        const errors = validationResult(req)
        if (errors.isEmpty()) return next()
        req.flash('errors-add-user', 'Failed to Add New User')
        return res.redirect('/admin')
    }
]

```

```

const toUpdateUser = [
  check('name').isLength({min : 4}),
  // check('nim').isLength({min : 4}).custom(async value => {
  //   const isNimExist = await User.findOne({uid : value})
  //   if (isNimExist) throw new Error('NIM is Already exist')
  //   return true
  // })),
  (req, res, next) => {
    const errors = validationResult(req)
    if (errors.isEmpty()) return next()
    req.flash('errors-add-user', 'Failed to Update User')
    return res.redirect('/admin')
  }
]

const toUpdateUserByUser = [
  check('nim').isLength({min : 4}).custom(async (value, {req}) => {
    const id = req.body._id
    const isNimExist = await User.findOne({uid : value})
    if (isNimExist && isNimExist._id.toHexString().slice(0, 23)
    !== id.slice(0, 23)) throw new Error('NIM is Already exist')
    return true
  })),
  check('name').isLength({min : 4}),
  check('password').custom((val) => {
    if (val.length !== 0 && val.length < 5) throw new
Error('Minimum Password Length is 4')
    return true
  })),
  (req, res, next) => {
    const errors = validationResult(req)
    if (errors.isEmpty()) return next()
    req.flash('errors-update-user', 'Failed to Update User')
    return res.redirect('/')
  }
]

const verifyToken = (req, res, next) => {
  const token = req.cookies['x-access-token']
  // if (!token) res.redirect('/')
  console.log(jwt.verify(token, process.env.SECRET_KEY))
  next()
}

module.exports = {
  toLogin,
  toAddUser,
  toUpdateUser,

```

```

    toUpdateUserByUser,
    verifyToken,
  }

```

models/admin.js

```
require('../utils/db')
```

```
const { User, createPassword } = require('../models/user')
```

```
const admin = new User({
  name : 'Admin',
  uid : '030400',
  password : createPassword('1234')
})
```

```
console.log(admin)
```

```
admin.save()
```

-models/files.js

```
require('../utils/db')
```

```
const mongoose = require('mongoose')
```

```
const mongoosePaginate = require('mongoose-paginate-v2');
```

```
const {stopWord} = require('../utils/stopWord')
```

```
const {userSchema} = require('../models/user')
```

```
const sastrawi = require('sastrawijs')
```

```
const allChar = ["-", " ", ""]
```

```
const stemmer = new sastrawi.Stemmer()
```

```
const tokenizer = new sastrawi.Tokenizer()
```

```
const fileSchema = new mongoose.Schema({
  name : {
    type : String,
    required : true,
  },
  fileName : {
    type : String,
    required : true
  },
  uploader : {
    type : String,
    required : true
  },
  text : {
    type : String,
    required : true,
  },
})
```

```
fileSchema.plugin(mongoosePaginate)
```



```

const File = mongoose.model('File', fileSchema)

const sanitizePdf2 = (text) => {
  const textFix = text.toLowerCase().replace(/^[a-z0-9 ]/gi, ' ')
  const newData = []
  // remove Stopword
  textFix.toLowerCase().split(' ').forEach(e => {
    e.trim()
    if (e.length > 2 && !stopWord.includes(e) &&
!allChar.includes(e)) newData.push(e)
  })
  return newData.join(' ')
}

const sanitizePdf = (text) => {
  const textFix = tokenizer.tokenize(text)
  const newData = []
  textFix.forEach(e => {
    // e.trim()
    if (e.length > 2 && !stopWord.includes(e))
newData.push(stemmer.stem(e))
  })
  return newData.join(' ')
}

module.exports = {
  File,
  sanitizePdf
}

```

models/user.js

```

const mongoose = require('mongoose')
const mongoosePaginate = require('mongoose-paginate-v2');
const bcrypt = require('bcryptjs')

require('../utils/db')

const userSchema = new mongoose.Schema( {
  name : {
    type : String,
    required : true,
  },
  uid : {
    type : String,
    required : true,
  },
  password : {
    type : String,
    required : true,
  },
})
userSchema.plugin(mongoosePaginate)

```

```

const User = mongoose.model('User', userSchema )

const salt = bcrypt.genSaltSync(10)

const verifyPassword = (password, h) => {
  return bcrypt.compare(password, h)
}

const createPassword = (password) => {
  return bcrypt.hashSync(password, salt)
}

module.exports = {
  User,
  userSchema,
  verifyPassword,
  createPassword
}

```

routes/activity.js

```

const express = require('express')
const multer = require('multer')
const fileUpload = require('express-fileupload')

const {home} = require('../controllers/homeController')
const {loginView, loginAuth, logout} =
require('../controllers/loginController')
const {admin, addUser, deleteUser, updateUser, updateUserByUser,
pagination, paginationUser, paginationFile, allUser} =
require('../controllers/adminController')
const {uploadPdf, deletePdf, result, resultNone} =
require('../controllers/fileController')

const {toLogin, toAddUser, toUpdateUser, toUpdateUserByUser,
verifyToken} = require('../middleware/user')
const {adminVerify} = require('../middleware/admin')
const { toAddFile } = require('../middleware/file')
const { fileStorage, fileFilter, fileStorageRes } =
require('../config/fileConfig')

const uploadFile = multer({
  storage : fileStorage,
  fileFilter : fileFilter
}).single('upload')

const uploadFileRes = multer({
  storage : fileStorageRes,
  fileFilter : fileFilter
}).single('upload-res')

const router = express.Router()

```

```

router.get('/', home)
router.post('/result', uploadFileRes, result)
router.post('/result-none', fileUpload(), resultNone)
router.put('/user', toUpdateUserByUser, updateUserByUser)

router.get('/login', loginView)
router.post('/login', toLogin, loginAuth)
router.get('/logout', logout)

router.post('/pdf', uploadFile, toAddFile, uploadPdf)
router.delete('/pdf', deletePdf)

router.get('/user', paginationUser)
router.get('/file', paginationFile)
router.get('/all-user', allUser)

router.use(adminVerify)
router.get('/admin', admin)
router.post('/admin-user', toAddUser, addUser)
router.put('/admin-user', toUpdateUser, updateUser)
router.delete('/admin-user', deleteUser)

```

```
module.exports = router
```

```
utils/db.js
```

```

require('dotenv').config()
const mongoose = require('mongoose')
const URI = process.env.MONGODB_URL
const DB = 'latihan'

```

```

mongoose.connect(URI, {
// mongoose.connect(`${URI}/${DB}`, {
  useNewUrlParser : true,
  useUnifiedTopology : true,
})

```

```
utils/stopWord.js
```

```

const stopWord = ["ada", "adalah", "adanya", "adapun", "agak",
"agaknya", "agar", "akan", "akankah", "akhir", "akhiri",
"akhirnya", "aku", "akulah", "amat", "amatlah", "anda", "andalah",
"antar", "antara", "antaranya", "apa", "apaan", "apabila",
"apakah", "apalagi", "apatah", "artinya", "asal", "asalkan",
"atas", "atau", "ataukah", "ataupun", "awal", "awalnya", "bagai",
"bagaikan", "bagaimana", "bagaimanakah", "bagaimanapun", "bagi",
"bagian", "bahkan", "bahwa", "bahwasanya", "baik", "bakal",
"bakalan", "balik", "banyak", "bapak", "baru", "bawah", "beberapa",
"begini", "beginian", "beginikah", "beginilah", "begitu",
"begitukah", "begitulah", "begitupun", "bekerja", "belakang",

```

"belakangan", "belum", "belumlah", "benar", "benarkah", "benarlah",
 "berada", "berakhir", "berakhirilah", "berakhirnya", "berapa",
 "berapakah", "berapalah", "berapapun", "berarti", "berawal",
 "berbagai", "berdatangan", "beri", "berikan", "berikut",
 "berikutnya", "berjumlah", "berkali-kali", "berkata",
 "berkehendak", "berkeinginan", "berkenaan", "berlainan", "berlalu",
 "berlangsung", "berlebihan", "bermacam", "bermacam-macam",
 "bermaksud", "bermula", "bersama", "bersama-sama", "bersiap",
 "bersiap-siap", "bertanya", "bertanya-tanya", "berturut",
 "berturut-turut", "bertutur", "berujar", "berupa", "besar",
 "betul", "betulkah", "biasa", "biasanya", "bila", "bilakah",
 "bisa", "bisakah", "boleh", "bolehkah", "bolehlah", "buat",
 "bukan", "bukankah", "bukanlah", "bukannya", "bulan", "bung",
 "cara", "caranya", "cukup", "cukupkah", "cukuplah", "cuma",
 "dahulu", "dalam", "dan", "dapat", "dari", "daripada", "datang",
 "dekat", "demi", "demikian", "demikianlah", "dengan", "depan",
 "di", "dia", "diakhiri", "diakhirinya", "dialah", "diantara",
 "diantaranya", "diberi", "diberikan", "diberikannya", "dibuat",
 "dibuatnya", "didapat", "didatangkan", "digunakan", "diibaratkan",
 "diibaratkannya", "diingat", "diingatkan", "diinginkan", "dijawab",
 "dijelaskan", "dijelaskannya", "dikarenakan", "dikatakan",
 "dikatakannya", "dikerjakan", "diketahui", "diketahuinya",
 "dikira", "dilakukan", "dilalui", "dilihat", "dimaksud",
 "dimaksudkan", "dimaksudkannya", "dimaksudnya", "diminta",
 "dimintai", "dimisalkan", "dimulai", "dimulailah", "dimulainya",
 "dimungkinkan", "dini", "dipastikan", "diperbuat", "diperbuatnya",
 "dipergunakan", "diperkirakan", "diperlihatkan", "diperlukan",
 "diperlukannya", "dipersoalkan", "dipertanyakan", "dipunyai",
 "diri", "dirinya", "disampaikan", "disebut", "disebutkan",
 "disebutkannya", "disini", "disinilah", "ditambahkan",
 "ditandaskan", "ditanya", "ditanyai", "ditanyakan", "ditegaskan",
 "ditunjukan", "ditunjuk", "ditunjuki", "ditunjukkan",
 "ditunjukkannya", "ditunjuknya", "dituturkan", "dituturkannya",
 "diucapkan", "diucapkannya", "diungkapkan", "dong", "dua", "dulu",
 "empat", "enggak", "enggaknya", "entah", "entahlah", "guna",
 "gunakan", "hal", "hampir", "hanya", "hanyalah", "hari", "harus",
 "haruslah", "harusnya", "hendak", "hendaklah", "hendaknya",
 "hingga", "ia", "ialah", "ibarat", "ibaratkan", "ibaratnya", "ibu",
 "ikut", "ingat", "ingat-ingat", "ingin", "inginkah", "inginkan",
 "ini", "inikah", "inilah", "itu", "itukah", "itulah", "jadi",
 "jadilah", "jadinya", "jangan", "jangankan", "janganlah", "jauh",
 "jawab", "jawaban", "jawabnya", "jelas", "jelaskan", "jelaslah",
 "jelasnya", "jika", "jikalau", "juga", "jumlah", "jumlahnya",
 "justru", "kala", "kalau", "kalaulah", "kalaupun", "kalian",
 "kami", "kamilah", "kamu", "kamulah", "kan", "kapan", "kapankah",
 "kapanpun", "karena", "karenanya", "kasus", "kata", "katakan",
 "katakanlah", "katanya", "ke", "keadaan", "kebetulan", "kecil",
 "kedua", "keduanya", "keinginan", "kelamaan", "kelihatan",
 "kelihatannya", "kelima", "keluar", "kembali", "kemudian",
 "kemungkinan", "kemungkinannya", "kenapa", "kepada", "kepadanya",
 "kesampaian", "keseluruhan", "keseluruhannya", "keterlaluhan",

"ketika", "khususnya", "kini", "kinilah", "kira", "kira-kira", "kiranya", "kita", "kitalah", "kok", "kurang", "lagi", "lagian", "lah", "lain", "lainnya", "lalu", "lama", "lamanya", "lanjut", "lanjutnya", "lebih", "lewat", "lima", "luar", "macam", "maka", "makanya", "makin", "malah", "malahan", "mampu", "mampukah", "mana", "manakala", "manalagi", "masa", "masalah", "masalahnya", "masih", "masihkah", "masing", "masing-masing", "mau", "maupun", "melainkan", "melakukan", "melalui", "melihat", "melihatnya", "memang", "memastikan", "memberi", "memberikan", "membuat", "memerlukan", "memihak", "meminta", "memintakan", "memisalkan", "memperbuat", "mempergunakan", "memperkirakan", "memperlihatkan", "mempersiapkan", "mempersoalkan", "mempertanyakan", "mempunyai", "memulai", "memungkinkan", "menaiki", "menambahkan", "menandaskan", "menanti", "menanti-nanti", "menantikan", "menanya", "menanyai", "menanyakan", "mendapat", "mendapatkan", "mendatang", "mendatangi", "mendatangkan", "menegaskan", "mengakhiri", "mengapa", "mengatakan", "mengatakannya", "mengenai", "mengerjakan", "mengetahui", "menggunakan", "menghendaki", "mengibaratkan", "mengibaratkannya", "mengingat", "mengingatkan", "menginginkan", "mengira", "mengucapkan", "mengucapkannya", "mengungkapkan", "menjadi", "menjawab", "menjelaskan", "menuju", "menunjuk", "menunjuki", "menunjukkan", "menunjuknya", "menurut", "menuturkan", "menyampaikan", "menyangkut", "menyatakan", "menyebutkan", "menyeluruh", "menyiapkan", "merasa", "mereka", "merekalah", "merupakan", "meski", "meskipun", "meyakini", "meyakinkan", "minta", "mirip", "misal", "misalkan", "misalnya", "mula", "mulai", "mulailah", "mulanya", "mungkin", "mungkinkah", "nah", "naik", "namun", "nanti", "nantinya", "nyaris", "nyatanya", "oleh", "olehnya", "pada", "padahal", "padanya", "pak", "paling", "panjang", "pantas", "para", "pasti", "pastilah", "penting", "pentingnya", "per", "percuma", "perlu", "perlukah", "perlunya", "pernah", "persoalan", "pertama", "pertama-tama", "pertanyaan", "pertanyakan", "pihak", "pihaknya", "pukul", "pula", "pun", "punya", "rasa", "rasanya", "rata", "rupanya", "saat", "saatnya", "saja", "sajalah", "saling", "sama", "sama-sama", "sambil", "sampai", "sampai-sampai", "sampaikan", "sana", "sangat", "sangatlah", "satu", "saya", "sayalah", "se", "sebab", "sebabnya", "sebagai", "sebagaimana", "sebagainya", "sebagian", "sebaik", "sebaik-baiknya", "sebaiknya", "sebaliknya", "sebanyak", "sebegini", "sebegini", "sebelum", "sebelumnya", "sebenarnya", "seberapa", "sebesar", "sebetulnya", "sebisanya", "sebuah", "sebut", "sebutlah", "sebutnya", "secara", "secukupnya", "sedang", "sedangkan", "sedemikian", "sedikit", "sedikitnya", "seenaknya", "segala", "segalanya", "segera", "seharusnya", "sehingga", "seingat", "sejak", "sejauh", "sejenak", "sejumlah", "sekadar", "sekadarnya", "sekali", "sekali-kali", "sekalian", "sekaligus", "sekalipun", "sekarang", "sekarang", "sekecil", "seketika", "sekiranya", "sekitar", "sekitarnya", "sekurang-kurangnya", "sekurangnya", "sela", "selain", "selaku", "selalu", "selama", "selama-lamanya", "selamanya", "selanjutnya", "seluruh", "seluruhnya", "semacam", "semakin", "semampu", "semampunya",

```

"semasa", "semasih", "semata", "semata-mata", "semaunya",
"sementara", "semisal", "semisalnya", "sempat", "semua",
"semuanya", "semula", "sendiri", "sendirian", "sendirinya",
"seolah", "seolah-olah", "seorang", "sepanjang", "sepantasnya",
"sepantasnyalah", "seperlunya", "seperti", "sepertinya", "sepihak",
"sering", "seringnya", "serta", "serupa", "sesaat", "sesama",
"sesampai", "sesegera", "sesekali", "seseorang", "sesuatu",
"sesuatunya", "sesudah", "sesudahnya", "setelah", "setempat",
"setengah", "seterusnya", "setiap", "setiba", "setibanya",
"setidak-tidaknya", "setidaknya", "setinggi", "seusai", "sewaktu",
"siap", "siapa", "siapakah", "siapapun", "sini", "sinilah", "soal",
"soalnya", "suatu", "sudah", "sudahkah", "sudahlah", "supaya",
"tadi", "tadinya", "tahu", "tahun", "tak", "tambah", "tambahnya",
"tampak", "tampaknya", "tandas", "tandasnya", "tanpa", "tanya",
"tanyakan", "tanyanya", "tapi", "tegas", "tegasnya", "telah",
"tempat", "tengah", "tentang", "tentu", "tentulah", "tentunya",
"tepat", "terakhir", "terasa", "terbanyak", "terdahulu",
"terdapat", "terdiri", "terhadap", "terhadapnya", "teringat",
"teringat-ingat", "terjadi", "terjadilah", "terjadinya", "terkira",
"terlalu", "terlebih", "terlihat", "termasuk", "ternyata",
"tersampaikan", "tersebut", "tersebutlah", "tertentu", "tertuju",
"terus", "terutama", "tetap", "tetapi", "tiap", "tiba", "tiba-
tiba", "tidak", "tidakkah", "tidaklah", "tiga", "tinggi", "toh",
"tunjuk", "turut", "tutur", "tuturnya", "ucap", "ucapnya", "ujar",
"ujarnya", "umum", "umumnya", "ungkap", "ungkapnya", "untuk",
"usah", "usai", "waduh", "wah", "wahai", "waktu", "waktunya",
"walau", "walaupun", "wong", "yaitu", "yakini", "yakni", "yang"]
module.exports = {
  stopWord
}

```

```
views/partials/add-files.ejs
```

```

<div class="modal fade fixed top-0 left-0 hidden w-full h-full
outline-none overflow-x-hidden overflow-y-auto "
  id="addFilesModal" tabindex="-1" aria-
labelledby="addFilesModalTitle" aria-modal="true" role="dialog">
  <div class="modal-dialog modal-dialog-centered relative w-auto
pointer-events-none">
    <div
      class="modal-content border-none shadow-lg relative
flex flex-col w-full pointer-events-auto bg-sky-800 bg-clip-padding
rounded-md outline-none text-current backdrop-filter backdrop-blur-
lg bg-opacity-30">
      <div
        class="modal-header flex flex-shrink-0 items-
center justify-between p-4 border-b border-gray-700 rounded-t-md
backdrop-filter backdrop-blur-lg bg-opacity-30">
        <h5 class="text-xl font-medium leading-normal text-
sky-200" id="exampleModalScrollableLabel">
          Add Files

```

```

        </h5>
        <button type="button"
            class="btn-close box-content w-4 h-4 p-1 text-
black border-none rounded-none opacity-50 focus:shadow-none
focus:outline-none focus:opacity-100 hover:text-black
hover:opacity-75 hover:no-underline"
            data-bs-dismiss="modal"                                aria-
label="Close"></button>
    </div>
    <div class="modal-body relative bg-sky-800 p-4
backdrop-filter backdrop-blur-lg bg-opacity-30">
        <form action="/pdf" method="POST"
enctype="multipart/form-data">
            <div class="w-full px-4 mb-8">
                <label for="id" class="text-base font-thin
text-white">Title</label>
                <input type="text" name="title" id="id"
                    class="w-full bg-sky-700 text-sky-200
p-3 rounded-md focus:outline-none focus:ring-sky-400 focus:ring-1"
                />
            </div>
            <div class="w-full my-5 px-4 flex items-center
justify-between text-sky-200">
                <label for="upload">Upload Your File
(*.pdf)</label>
                <input
                    class="file:bg-sky-700 file:border-0
file:text-white file:rounded-md file:p-2 file:hover:bg-sky-600"
                    type="file" name="upload">
            </div>
        </div>
    </div>
    <div
        class="modal-footer flex flex-shrink-0 flex-wrap
items-center justify-end p-4 border-t border-sky-700 rounded-b-md
bg-sky-800 backdrop-filter backdrop-blur-lg bg-opacity-30">
        <button type="submit" class="w-full bg-sky-700 py-
2 rounded-lg text-white hover:bg-sky-600">
            Add File
        </button>
    </form>
    </div>
</div>
</div>
</div>

<div class="modal fade fixed top-0 left-0 hidden w-full h-full
outline-none overflow-x-hidden overflow-y-auto "
    id="deleteFileModal"                                tabindex="-1"                                aria-
labelledby="deleteFileModalTitle" aria-modal="true" role="dialog">
    <div class="modal-dialog modal-dialog-centered relative w-auto
pointer-events-none bg-opacity-30">

```

```

        <div
            class="modal-content border-none shadow-lg relative
flex flex-col w-full pointer-events-auto bg-sky-800 bg-clip-padding
rounded-md outline-none text-current backdrop-filter backdrop-blur-
lg bg-opacity-30">
            <div
                class="modal-header flex flex-shrink-0 items-
center justify-between p-4 border-b border-gray-700 rounded-t-md
bg-opacity-30">
                <h5 class="text-xl font-medium leading-normal text-
sky-200" id="exampleModalScrollableLabel">
                    Delete File
                </h5>
                <button type="button"
                    class="btn-close box-content w-4 h-4 p-1 text-
white border-none rounded-none opacity-50 focus:shadow-none
focus:outline-none focus:opacity-100 hover:text-black
hover:opacity-75 hover:no-underline"
                    data-bs-dismiss="modal" aria-
label="Close"></button>
                </div>
                <div class="modal-body relative bg-sky-800 p-4 bg-
opacity-30">
                    <p id="text-delete-file" class="text-white text-
base"></p>
                    <form action="/pdf?_method=DELETE" method="POST">
                        <input type="hidden" id="del-file-id"
name="id" value="">
                        <input type="hidden" id="del-file-title"
name="title" value="">
                    </div>
                    <div
                        class="modal-footer flex flex-shrink-0 flex-wrap
items-center justify-end p-4 border-t border-gray-700 rounded-b-md
bg-sky-800 bg-opacity-30">
                        <button type="submit" class="w-full bg-sky-700 py-
2 rounded-lg text-white hover:bg-sky-600"
                            id="modal-sub-user">
                            Delete File
                        </button>
                    </form>
                </div>
            </div>
        </div>
    </div>
</div>

```

views/partials/add-user.ejs

```

<div class="modal fade fixed top-0 left-0 hidden w-full h-full
outline-none overflow-x-hidden overflow-y-auto "

```



```

        id="addUserModal"                tabindex="-1"                aria-
labelledby="addUserModalTitle" aria-modal="true" role="dialog">
        <div class="modal-dialog modal-dialog-centered relative w-auto
pointer-events-none bg-opacity-30">
            <div
                class="modal-content border-none shadow-lg relative
flex flex-col w-full pointer-events-auto bg-sky-800 bg-clip-padding
rounded-md outline-none text-current backdrop-filter backdrop-blur-
lg bg-opacity-30">
                    <div
                        class="modal-header flex flex-shrink-0 items-
center justify-between p-4 border-b border-sky-700 rounded-t-md
backdrop-filter backdrop-blur-lg bg-opacity-30">
                            <h5 class="text-xl font-medium leading-normal text-
sky-200" id="exampleModalScrollableLabel">
                                Add User
                            </h5>
                            <button type="button"
                                class="btn-close box-content w-4 h-4 p-1 text-
black border-none rounded-none opacity-50 focus:shadow-none
focus:outline-none focus:opacity-100 hover:text-black
hover:opacity-75 hover:no-underline"
                                data-bs-dismiss="modal"                aria-
label="Close"></button>
                        </div>
                        <div class="modal-body relative bg-sky-800 p-4
backdrop-filter backdrop-blur-lg bg-opacity-30">
                            <form id="form-user" action="/admin-user"
method="POST">
                                <input type="hidden" value="" name="id"
id="id_user">
                                    <div class="w-full px-4 mb-8">
                                        <label for="id" class="text-base font-thin
text-white">Id (NIM)</label>
                                        <input type="text" name="nim"
id="nim_user"
                                            class="w-full bg-sky-700 text-sky-200
p-3 rounded-md focus:outline-none focus:ring-sky-400 focus:ring-1"
                                        />
                                    </div>
                                    <div class="w-full px-4 mb-8">
                                        <label for="name" class="text-base font-
thin text-white">Username</label>
                                        <input type="text" name="name"
id="username_user"
                                            class="w-full bg-sky-700 text-sky-200
p-3 rounded-md focus:outline-none focus:ring-sky-400 focus:ring-1"
                                        />
                                    </div>
                                    <div class="w-full px-4 mb-8">

```

```

        <label for="password" class="text-base
font-thin text-white">Password</label>
        <input type="password" name="password"
id="password_user" class=" w-full bg-sky-700 text-sky-200 p-3
rounded-md focus:outline-none
        focus:ring-sky-400          focus:ring-1
disabled:bg-sky-500" />
    </div>
</div>
<div
    class="modal-footer flex flex-shrink-0 flex-wrap
items-center justify-end p-4 border-t border-sky-700 rounded-b-md
bg-sky-800 backdrop-filter backdrop-blur-lg bg-opacity-30">
    <button type="submit" class="w-full bg-sky-700 py-
2 rounded-lg text-white hover:bg-sky-600"
        id="modal-sub-user">
        Add User
    </button>
</form>
</div>
</div>
</div>
</div>

<div class="modal fade fixed top-0 left-0 hidden w-full h-full
outline-none overflow-x-hidden overflow-y-auto "
    id="deleteUserModal"          tabindex="-1"          aria-
labelledby="deleteUserModalTitle" aria-modal="true" role="dialog">
    <div
        class="modal-dialog modal-dialog-centered relative w-auto
pointer-events-none backdrop-filter backdrop-blur-lg bg-opacity-
30">
        <div
            class="modal-content border-none shadow-lg relative
flex flex-col w-full pointer-events-auto bg-sky-800 bg-clip-padding
rounded-md outline-none text-current backdrop-filter backdrop-blur-
lg bg-opacity-30">
            <div
                class="modal-header flex flex-shrink-0 items-
center justify-between p-4 border-b border-sky-700 rounded-t-md
backdrop-filter backdrop-blur-lg bg-opacity-30">
                <h5 class="text-xl font-medium leading-normal text-
sky-200" id="exampleModalScrollableLabel">
                Delete User
            </h5>
            <button type="button"
                class="btn-close box-content w-4 h-4 p-1 text-
white border-none rounded-none opacity-50 focus:shadow-none
focus:outline-none          focus:opacity-100          hover:text-black
hover:opacity-75 hover:no-underline"

```

```

        data-bs-dismiss="modal"                                aria-
label="Close"></button>
    </div>
    <div class="modal-body relative bg-sky-800 p-4
backdrop-filter backdrop-blur-lg bg-opacity-30">
        <p id="text-delete" class="text-white text-
base"></p>
        <form action="/admin-user?_method=DELETE"
method="POST">
            <input type="hidden" id="del-user-id"
name="id" value="">
            <input type="hidden" id="del-user-name"
name="name" value="">
        </div>
        <div
            class="modal-footer flex flex-shrink-0 flex-wrap
items-center justify-end p-4 border-t border-sky-700 rounded-b-md
bg-sky-800 backdrop-filter backdrop-blur-lg bg-opacity-30">
            <button type="submit" class="w-full bg-sky-700 py-
2 rounded-lg text-white hover:bg-sky-600"
                id="modal-sub-user">
                Delete User
            </button>
        </form>
    </div>
</div>
</div>
</div>

```

views/partials/description.ejs

```

<!-- Description Section -->
<section class="mt-10 min-h-8 lg:mt-0">
    <div class="container">
        <div class="flex flex-wrap items-center">
            <div class="p-5 w-full h-40 lg:w-1/2 lg:h-full">
                <% if (typeof user !== 'undefined') { %>
                    <h1 class="font-medium text-xl text-sky-900 my-
auto">Selamat Datang <%= user %></h1>
                <% } %>
                    <h1 class="font-bold text-2xl text-sky-900 my-
auto">Plagiarism Checkers</h1>
                <p class="text-base">Website Ini Menggunakan Metode
Dice Similarity Untuk Mempersentasekan
                    Kemiripan Dokumen atau Teks Anda
                    dengan Dokumen yang Terdapat di Database</p>
            </div>
            <div class="w-full my-4 lg:w-1/2 p-3">
                <div class="p-4">
                    <!-- <form action="/result" method="POST"
enctype="multipart/form-data"> -->

```

```

        <form action="/result-none" method="POST"
enctype="multipart/form-data">
            <div class="w-full px-4">
                <div class="alert mt-2 bg-red-300
rounded-lg py-5 px-6 mb-3 text-base text-red-700 hidden items-
center w-full alert-dismissible fade show"
                    id="error-file" role="alert">
                        Gagal Melakukan Pendeteksian
                        <button type="button"
                            class="btn-close box-content
w-4 h-4 p-1 ml-auto text-yellow-900 border-none rounded-none
opacity-50 focus:shadow-none focus:outline-none focus:opacity-100
hover:text-yellow-900 hover:opacity-75 hover:no-underline"
                            data-bs-dismiss="alert" aria-
label="Close"></button>
                    </div>
                    <textarea type="text" id="text"
                        class="w-full bg-sky-800 text-
white p-3 rounded-md h-48 focus:border-2 focus:border-sky-700
focus:outline-none resize-none"
                        name="text" placeholder="Input
Your Text"></textarea>
                    </div>
                    <div class="w-full my-5 px-4 flex items-
center justify-between">
                        <label for="upload">or Upload Your File
(*.pdf)</label>
                        <input
                            class="file:bg-sky-800
file:border-0 file:text-white file:rounded-md file:p-2
file:hover:bg-sky-700"
                            type="file" name="upload-res"
id="upload-res">
                        </div>
                        <button type="submit" id="check"
                            class="w-full bg-sky-800 py-2 rounded-
lg text-white hover:bg-sky-700">
                            <i class="fa-solid fa-magnifying-glass
mr-2"></i> Check
                                Plagiarism</button>
                        </form>
                    </div>
                </div>
            </div>
        </div>
</section>
<!-- Description Section -->

views/partials/footer.ejs

```

```

<footer class="bg-sky-900 pt-10 pb-5 mt-3">
  <div class="container">
    <div class="w-full pt-10 border-t border-sky-700">
      <div class="flex items-center justify-center mb-5">
        <!-- Whatsapp -->
        <a
          href="https://wa.me/+6282195008936"
          target="_blank"
          class="w-9 h-9 mr-2 flex justify-center items-
            center">
          <svg role="img" width="25" class="fill-white
            hover:fill-sky-700" viewBox="0 0 24 24"
            xmlns="http://www.w3.org/2000/svg">
            <title>WhatsApp</title>
            <path
              d="M17.472 14.382c-.297-.149-1.758-
                .867-2.03-.967-.273-.099-.471-.148-.67.15-.197.297-.767.966-.94
                1.164-.173.199-.347.223-.644.075-.297-.15-1.255-.463-2.39-1.475-
                .883-.788-1.48-1.761-1.653-2.059-.173-.297-.018-.458.13-.606.134-
                .133.298-.347.446-.52.149-.174.198-.298.298-.497.099-.198.05-.371-
                .025-.52-.075-.149-.669-1.612-.916-2.207-.242-.579-.487-.5-.669-
                .51-.173-.008-.371-.01-.57-.01-.198 0-.52.074-.792.372-.272.297-
                1.04 1.016-1.04 2.479 0 1.462 1.065 2.875 1.213 3.074.149.198 2.096
                3.2 5.077 4.487.709.306 1.262.489 1.694.625.712.227 1.36.195
                1.871.118.571-.085 1.758-.719 2.006-1.413.248-.694.248-1.289.173-
                1.413-.074-.124-.272-.198-.57-.347m-5.421 7.403h-.004a9.87 9.87 0
                01-5.031-1.378l-.361-.214-3.741.982.998-3.648-.235-.374a9.86 9.86
                0 01-1.51-5.26c.001-5.45 4.436-9.884 9.888-9.884 2.64 0 5.122 1.03
                6.988 2.898a9.825 9.825 0 012.893 6.994c-.003 5.45-4.437 9.884-
                9.885 9.884m8.413-18.297A11.815 11.815 0 0012.05 0C5.495 0 .16
                5.335.157 11.892c0 2.096.547 4.142 1.588 5.945L.057 24l6.305-
                1.654a11.882 11.882 0 005.683 1.448h.005c6.554 0 11.89-5.335
                11.893-11.893a11.821 11.821 0 00-3.48-8.413Z" />
            </svg>
          </a>
          <a
            href="https://www.facebook.com/profile.php?id=100000053025926"
            target="_blank"
            class="w-9 h-9 mr-3 flex justify-center items-
              center">
            <svg role="img" width="25" viewBox="0 0 24 24"
              class="fill-white hover:fill-sky-700"
              xmlns="http://www.w3.org/2000/svg">
              <title>Facebook</title>
              <path
                d="M24 12.073c0-6.627-5.373-12-12-12s-
                  12 5.373-12 12c0 5.99 4.388 10.954 10.125 11.854v-8.385H7.078v-
                  3.47h3.047V9.43c0-3.007 1.792-4.669 4.533-4.669 1.312 0 2.686.235
                  2.686.235v2.953H15.83c-1.491 0-1.956.925-1.956 1.874v2.25h3.328l-
                  .532 3.47h-2.796v8.385C19.612 23.027 24 18.062 24 12.073z" />
              </svg>
            </a>

```

```

        <a
            href="https://github.com/arxs-abd"
            target="_blank"
            class="w-9 h-9 mr-3 flex justify-center items-center">
                <svg role="img" width="25" class="fill-white
                    hover:fill-sky-700" viewBox="0 0 24 24"
                    xmlns="http://www.w3.org/2000/svg">
                    <title>GitHub</title>
                    <path
                        d="M12 .297c-6.63 0-12 5.373-12 12 0
                    5.303 3.438 9.8 8.205 11.385 6.113.82-.258.82-.577 0-.285-.01-1.04-
                    .015-2.04-3.338-7.24-4.042-1.61-4.042-1.61C4.422 18.07 3.633 17.7
                    3.633 17.7c-1.087-.744-.084-.729-.084-.729 1.205.084 1.838 1.236
                    1.838 1.236 1.07 1.835 2.809 1.305 3.495.998.108-.776.417-1.305.76-
                    1.605-2.665-.3-5.466-1.332-5.466-5.93 0-1.31.465-2.38 1.235-3.22-
                    .135-.303-.54-1.523.105-3.176 0 0 1.005-.322 3.3 1.23.96-.267 1.98-
                    .399 3-.405 1.02.006 2.04.138 3 .405 2.28-1.552 3.285-1.23 3.285-
                    1.23.645 1.653.24 2.873.12 3.176.765.84 1.23 1.91 1.23 3.22 0 4.61-
                    2.805 5.625-5.475 5.92.42.36.81 1.096.81 2.22 0 1.606-.015 2.896-
                    .015 3.286 0 .315.21.69.825.57C20.565 22.092 24 17.592 24 12.297c0-
                    6.627-5.373-12-12-12" />
                </svg>
            </a>
        </div>
        <p class="font-medium text-xs text-sky-100 text-center">
            © <script>
                document.write(new Date().getFullYear())
            </script> Copyright : <span class="font-light">Aris
            Akhyar Abdillah</span>
        </p>
    </div>
</div>
</footer>

```

views/partials/login.ejs

```

<section>
    <div class="container">
        <div class="w-3/4 my-4 lg:w-2/3 lg:mx-auto p-3 text-white
            bg-sky-800 rounded-md m-auto">
            <h2 class="font-semibold text-2xl mb-2 text-center">Login To Application</h2>
            <div class="p-4">
                <% if(errors.length !== 0) { %>
                    <div class="alert bg-red-300 rounded-lg py-5 px-6
                    mb-3 text-base text-red-700 inline-flex items-center w-full alert-
                    dismissible fade show"
                        role="alert">
                        <strong class="mx-1">NIM </strong> or

```

```

        <strong class="mx-1">Password </strong> is
Wrong
        <button type="button"
            class="btn-close box-content w-4 h-4 p-1
ml-auto text-yellow-900 border-none rounded-none opacity-50
focus:shadow-none focus:outline-none focus:opacity-100 hover:text-
yellow-900 hover:opacity-75 hover:no-underline"
            data-bs-dismiss="alert"                aria-
label="Close"></button>
        </div>
    <% } %>
    <form action="/login" method="POST">
        <div class="w-full px-4 mb-8">
            <label for="id" class="text-base font-thin
text-white">Id (NIM)</label>
            <input type="text" name="nim" id="id"
                class="w-full bg-sky-700 text-sky-200
p-3 rounded-md focus:outline-none focus:ring-sky-400 focus:ring-1"
            />
        </div>
        <div class="w-full px-4 mb-8">
            <label for="password" class="text-base
font-thin text-white">Password</label>
            <input type="password" name="password"
id="password" class=" w-full bg-sky-700 text-sky-200 p-3 rounded-
md focus:outline-none
                focus:ring-sky-400 focus:ring-1" />
        </div>
        <button type="submit"
            class="w-full bg-sky-700 py-2 rounded-lg
text-white hover:bg-sky-600">Login</button>
        </form>
    </div>
</div>
</div>
</section>

```

views/partials/navbar.ejs

```

<header
    class="bg-sky-900 top-0 left-0 w-full flex items-center p-1
backdrop-filter backdrop-blur-lg bg-opacity-90 border-b border-sky-
800">
    <!-- class="bg-sky-900 top-0 left-0 w-full flex items-center p-
1 backdrop-filter backdrop-blur-lg bg-opacity-90 border-b border-
sky-800 sticky z-10"> -->
    <div class="container">
        <div class="flex items-center justify-between">
            <div class="w-full ml-5 lg:ml-0">
                <a href="/" class="block font-bold text-xl text-
white py-4">Plagiarsm Checker</a>

```

```

        </div>
        <button id="hamburger" name="hamburger" type="button"
class="absolute right-10 block lg:hidden">
            <span class="hamburger-line origin-top-left
transition duration-300 ease-in-out"></span>
            <span class="hamburger-line transition duration-
300 ease-in-out"></span>
            <span class="hamburger-line origin-bottom-left
transition duration-300 ease-in-out"></span>
        </button>
        <nav id="nav-menu"
class="absolute right-4 hidden top-full w-full max-
w-[250px] rounded-lg bg-white py-5 shadow-lg -mt-3 lg:static
lg:block lg:max-w-full lg:rounded-none lg:bg-transparent lg:shadow-
none">
            <ul class="lg:flex lg:justify-end lg:text-white">
                <li class="mx-1 lg:mx-3 p-2 hover:bg-sky-700
hover:text-white hover:rounded-xl"><a href="/">Home</a>
                </li>
                <li class="mx-1 lg:mx-3 p-2 hover:bg-sky-700
hover:text-white hover:rounded-xl "><a
                    href="#about-me">About Me</a>
                <li class="mx-1 lg:mx-3 p-2 hover:bg-sky-700
hover:text-white hover:rounded-xl "><a
                    href="https://github.com/arxs-
abd/Main-Skripsi">How It
                    Works</a>
                </li>
                <li
                    class="mx-1 lg:mx-3 py-2 px-4 bg-sky-900
text-white rounded-xl hover:bg-sky-700 hover:text-white">
                    <% if (typeof user !== 'undefined') { %>
                    <a href="/logout">Logout</a>
                    <% } else {%>
                    <a href="/login">Login</a>
                    <% }%>
                </li>
            </ul>
        </nav>
        <!-- <nav class="text-white font-normal">
            <ul class="lg:flex hidden">
                <li class="mx-1 lg:mx-3 p-2 hover:bg-sky-700
hover:rounded-xl"><a href="/">Home</a></li>
                <li class="mx-1 lg:mx-3 p-2 hover:bg-sky-700
hover:rounded-xl "><a href="#about-me">About Me</a>
                <li class="mx-1 lg:mx-3 p-2 hover:bg-sky-700
hover:rounded-xl "><a href="#about-me">How It
                    Works</a>
                </li>
                <li class="mx-1 lg:mx-3 py-2 px-4 bg-sky-900
text-white rounded-xl hover:bg-sky-700">

```



```

        < if (typeof user !== 'undefined') { %>
        <a href="/logout">Logout</a>
        < } else {%>
        <a href="/login">Login</a>
        < }%>
      </li>
    </ul>
  </nav> -->
</div>
</div>
</header>

<script>
  document.querySelector('#hamburger').addEventListener('click',
function () {
    this.classList.toggle('hamburger-active')
    document.querySelector('#nav-
menu').classList.toggle('hidden')
  })
</script>

```

views/partials/result-none.ejs

```

<div id="spinner" class="fixed top-0 left-0 hidden items-center
justify-center w-[100%] h-[100%] bg-black opacity-50">
  <div class="spinner-border animate-spin inline-block w-8 h-8
border-4 rounded-full text-white" role="status">
    <span class="visually-hidden text-white">Loading...</span>
  </div>
  <span id="loading-number" class="ml-5 text-base text-
white">Loading</span>
</div>
<section id="result-none" class="pt-20 text-white hidden">
  <div class="container">
    <div class="w-full px-4 text-center">
      <h1 class="font-semibold text-2xl mb-2 text-sky-
900">Result</h1>
    </div>
    <div class="flex flex-wrap justify-between">
      <div class="w-full h-[45rem] lg:w-[59%] lg:h-[25rem]
bg-sky-800 rounded-md p-4">
        <embed src="" id="preview" class="w-full h-full">
      </div>
      <div class="w-full lg:w-[39%] rounded-md flex flex-
col">
        <!-- <div class="bg-red-300 rounded-lg py-5 px-6
mb-4 text-base text-red-700" role="alert">
          Your Document Is Unique
        </div> -->
        <div class="w-full bg-sky-800 rounded-md p-4 mt-5
lg:mt-0">

```

```

        <div class="flex flex-wrap justify-between">
            <h2 class="font-semibold text-xl mb-
3">Details</h2>
        </div>
        <div class="flex flex-col">
            <p class="text-base" id="file-name">File
Name : Namanya</p>
            <p class="text-base" id="file-size">File
Size : Namanya</p>
            <p class="text-base" id="file-
char">Character : Jumlahnya %></p>
            <p class="text-base" id="file-word">Words
: Katanya </p>
        </div>
    </div>

    <div class="w-full bg-sky-800 rounded-md p-4 mt-
5">
        <h2 class="font-semibold text-xl mb-
3">Plagirism Percentage</h2>
        <div class="">
            <div class="flex">
                <p class="text-base w-1/2">Unique :
<span id="text-uniq">10</span> %</p>
                <p class="text-base w-1/2">Plagiarism
: <span id="text-plag">10</span> %</p>
            </div>
            <div class="w-full mt-3 flex bg-sky-900
rounded-full">
                <div class="bg-sky-700 text-s font-
medium text-blue-100 text-center p-0.5 leading-none rounded-l-full"
id="bar-uniq" style="width: 10%;">
                    10 %
                </div>
                <div class="text-s font-medium text-
blue-100 text-center p-0.5 leading-none"
style="width: 90%;" id="bar-plag">
                    90 %
                </div>
            </div>
        </div>
    </div>
    <div class="w-full bg-sky-800 mt-5 rounded-md p-4 mb-
5">
        <h2 class="font-semibold text-xl mb-3">File With
High Similarity</h2>
        <table class="min-w-full text-white table-auto">
            <thead class="border-b border-sky-500 font-
semibold">
                <tr>

```

```

                <th scope="col" class="text-sm font-
medium px-6 py-4 text-left">
                    #
                </th>
                <th scope="col" class="text-sm font-
medium px-6 py-4 text-left">
                    Title Pdf
                </th>
                <th scope="col" class="text-sm font-
medium px-6 py-4 text-left">
                    Uploader
                </th>
                <th scope="col" class="text-sm font-
medium px-6 py-4 text-left">
                    Similarity
                </th>
                <th scope="col" class="text-sm font-
medium px-6 py-4 text-left">
                    Show
                </th>
            </tr>
        </thead>
        <tbody id="table-result">

        </tbody>
    </table>
</div>
</div>
</div>
</div>
</section>

-views/partials/result.ejs
<section class="pt-20 text-white">
    <div class="container">
        <div class="w-full px-4 text-center">
            <h1 class="font-semibold text-2xl mb-2">Result</h1>
        </div>
        <div class="flex flex-wrap justify-between">
            <div class="w-full lg:w-[49%] bg-sky-800 rounded-md p-
4">
                <h2 class="font-semibold text-xl mb-
3">Details</h2>
                <div class="flex flex-col">
                    <p class="text-base">File Name : <%=
result.fileInformation.originalname %></p>
                    <p class="text-base">Character : <%=
result.fileInformation.length %></p>
                    <p class="text-base">Words :
                        <%= result.fileInformation.word %></p>
                </div>
            </div>
        </div>
    </div>

```

```

        <div class="w-full mt-5 lg:w-[49%] lg:mt-0 bg-sky-800
rounded-md p-4">
            <h2 class="font-semibold text-xl mb-3">Plagirism
Percentage</h2>
            <div class="flex">
                <p class="text-base w-1/2">Unique : <%= `${100
- result plag}%` %></p>
                <p class="text-base w-1/2">Plagiarism : <%=
`${result plag}%` %></p>
            </div>
            <div class="w-full mt-3 flex bg-sky-900 rounded-
full">
                <div class="bg-sky-700 text-s font-medium text-
blue-100 text-center p-0.5 leading-none rounded-l-full w-[80%]"
                    style="width: <%= `${100 - result plag}%`
%>;">
                    <%= `${100 - result plag}%` %>
                </div>
                <div class="text-s font-medium text-blue-100
text-center p-0.5 leading-none"
                    style="width: <%= `${result plag}%` %>;">
                    <%= `${result plag}%` %>
                </div>
            </div>
        </div>
        <div class="w-full bg-sky-800 mt-5 rounded-md p-4">
            <h2 class="font-semibold text-xl mb-3">File With
High Similarity</h2>
            <table class="min-w-full text-white table-auto">
                <thead class="border-b border-sky-500 font-
semibold">
                    <tr>
                        <th scope="col" class="text-sm font-
medium px-6 py-4 text-left">
                            #
                        </th>
                        <th scope="col" class="text-sm font-
medium px-6 py-4 text-left">
                            Title Pdf
                        </th>
                        <th scope="col" class="text-sm font-
medium px-6 py-4 text-left">
                            Uploader
                        </th>
                        <th scope="col" class="text-sm font-
medium px-6 py-4 text-left">
                            Similarity
                        </th>
                        <th scope="col" class="text-sm font-
medium px-6 py-4 text-left">
                            Show

```

```

        </th>
      </tr>
    </thead>
    <tbody>
      <% result.resultAll.forEach((file, i) => {
%>
        <tr class="border-b border-sky-500">
          <td class="px-6 py-4 whitespace-nowrap
text-sm font-medium"><%= i + 1 %></td>
          <td class="text-sm font-light px-6 py-
4">
            <%= file.file.name %>
          </td>
          <td class="text-sm font-light px-6 py-
4">
            <% const index = user.findIndex(o
=> { %>
              <% return o._id.toHexString() ===
file.file.uploader %>
              <% }) %>
              <%= user[index].name %>
            </td>
            <td class="text-sm font-light px-6 py-
4 whitespace-nowrap">
              <%= file.result %>%
            </td>
            <td class="text-sm font-light px-6 py-
4 whitespace-nowrap">
              <a href="<%= 'docs/' +
file.file.fileName %>" target="_blank"
class="w-full bg-sky-700 p-3
rounded-lg text-white">
                Show
              </a>
            </td>
          </tr>
        <% }) %>
      </tbody>
    </table>
  </div>
</div>
</div>
</section>

views/partials/table-details.ejs

```

```

<section>
  <div class="container mb-5">
    <div class="flex flex-wrap justify-between">
      <div class="w-[95%] mx-auto bg-sky-800 lg:w-[49%] mt-5
rounded-md p-4">

```

```

        <div class="flex flex-wrap flex-row items-center
justify-between">
            <h2 class="font-semibold text-xl mb-3 text-
white">All User</h2>
            <button type="button" class="w-1/2 bg-sky-700
py-2 rounded-lg text-white hover:bg-sky-600"
                id="modal-user" data-bs-toggle="modal"
data-bs-target="#addUserModal"><i
                class="fas fa-plus mr-3"></i>Add
                User</button>
        </div>
        <% if(errorsUser.length !== 0) { %>
            <div class="alert mt-2 bg-red-300 rounded-lg py-5
px-6 mb-3 text-base text-red-700 inline-flex items-center w-full
alert-dismissible fade show"
                role="alert">
                <%= errorsUser %>
                <button type="button"
                    class="btn-close box-content w-4 h-4 p-1
ml-auto text-yellow-900 border-none rounded-none opacity-50
focus:shadow-none focus:outline-none focus:opacity-100 hover:text-
yellow-900 hover:opacity-75 hover:no-underline"
                    data-bs-dismiss="alert" aria-
label="Close"></button>
            </div>
            <% } %>

            <% if(successUser.length !== 0) { %>
                <div class="alert mt-2 bg-green-300 rounded-lg py-
5 px-6 mb-3 text-base text-green-700 inline-flex items-center w-
full alert-dismissible fade show"
                    role="alert">
                    <%= successUser %>
                    <button type="button"
                        class="btn-close box-content w-4 h-4 p-1
ml-auto text-yellow-900 border-none rounded-none opacity-50
focus:shadow-none focus:outline-none focus:opacity-100 hover:text-
yellow-900 hover:opacity-75 hover:no-underline"
                        data-bs-dismiss="alert" aria-
label="Close"></button>
                </div>
                <% } %>
            <div class="overflow-y-auto flex flex-col">
                <table class="min-w-full text-white table-
auto">
                    <thead class="border-b border-sky-500
font-semibold">
                        <tr>
                            <th scope="col" class="text-sm
font-medium px-6 py-4 text-left">
                                #

```

```

        </th>
        <th scope="col" class="text-sm
font-medium px-6 py-4 text-left">
            Username
        </th>
        <th scope="col" class="text-sm
font-medium px-6 py-4 text-left">
            Id (NIM)
        </th>
        <th scope="col" class="text-sm
font-medium px-6 py-4 text-left">
            Action
        </th>
    </tr>
</thead>
<tbody id="table-user">
</tbody>
</table>

<div class="paginaton mt-3 min-w-full">
    <div class="flex justify-center">
        <nav aria-label="Page navigation
example">
            <ul class="flex list-style-none
gap-1">
                <li class="page-item"><a
                    class="page-link
relative block py-1.5 px-3 rounded border-0 bg-transparent outline-
none transition-all duration-300 text-white hover:text-white
hover:bg-sky-900 focus:shadow-none"
                    href="#" aria-
label="First" data-page="false" data-click="false">
                        <span aria-
hidden="true">First</span>
                    </a></li>
                <li class="page-item"><a
                    class="page-link
relative block py-1.5 px-3 rounded border-0 bg-transparent outline-
none transition-all duration-300 text-white hover:text-white
hover:bg-sky-900 focus:shadow-none"
                    href="#" aria-
label="Previous" data-page="false" data-click="false">
                        <span aria-
hidden="true">&laquo;</span>
                    </a></li>
                <li class="page-item"><a
                    class="page-link
relative block py-1.5 px-3 rounded border-0 bg-sky-900 outline-none
transition-all duration-300 text-white hover:text-white hover:bg-
sky-900 focus:shadow-none"

```

```

data-click="false"
href="#">1</a></li>
    <li class="page-item"><a
      class="page-link
relative block py-1.5 px-3 rounded border-0 bg-transparent outline-
none transition-all duration-300 text-white hover:text-white
hover:bg-sky-900 focus:shadow-none"
      data-click="false"
href="#">2</a></li>
    <li class="page-item"><a
      class="page-link
relative block py-1.5 px-3 rounded border-0 bg-transparent outline-
none transition-all duration-300 text-white hover:text-white
hover:bg-sky-900 focus:shadow-none"
      data-click="false"
href="#">3</a></li>
    <li class="page-item"><a
      class="page-link
relative block py-1.5 px-3 rounded border-0 bg-transparent outline-
none transition-all duration-300 text-white hover:text-white
hover:bg-sky-900 focus:shadow-none"
      href="#"          aria-
label="Next" data-click="false">
      <span          aria-
hidden="true">&raquo;</span>
    </a></li>
    <li class="page-item"><a
      class="page-link
relative block py-1.5 px-3 rounded border-0 bg-transparent outline-
none transition-all duration-300 text-white hover:text-white
hover:bg-sky-900 focus:shadow-none"
      href="#"          aria-
label="Last" data-page="false" data-click="false">
      <span          aria-
hidden="true">Last</span>
    </a></li>
  </ul>
</nav>
</div>
</div>
</div>
</div>
</div>
<div class="w-[95%] mx-auto h-fit bg-sky-800 lg:w-[49%]
mt-5 rounded-md p-4 overflow-y-auto">
  <div class="flex flex-wrap flex-row items-center
justify-between">
    <h2 class="font-semibold text-xl mb-3 text-
white">All Files</h2>
    <button type="button" class="w-1/2 bg-sky-700
py-2 rounded-lg text-white hover:bg-sky-600"

```



```

                                data-bs-toggle="modal"                                data-bs-
target="#addFilesModal">
                                <i class="fas fa-plus mr-3"></i>Add
                                Files</button>
                                </div>
                                <% if(errorsFile.length !== 0) { %>
                                <div class="alert mt-2 bg-red-300 rounded-lg py-5
px-6 mb-3 text-base text-red-700 inline-flex items-center w-full
alert-dismissible fade show"
                                role="alert">
                                <%= errorsFile %>
                                <button type="button"
                                class="btn-close box-content w-4 h-4 p-1
ml-auto text-yellow-900 border-none rounded-none opacity-50
focus:shadow-none focus:outline-none focus:opacity-100 hover:text-
yellow-900 hover:opacity-75 hover:no-underline"
                                data-bs-dismiss="alert"                                aria-
label="Close"></button>
                                </div>
                                <% } %>

                                <% if(successFile.length !== 0) { %>
                                <div class="alert mt-2 bg-green-300 rounded-lg py-
5 px-6 mb-3 text-base text-green-700 inline-flex items-center w-
full alert-dismissible fade show"
                                role="alert">
                                <%= successFile %>
                                <button type="button"
                                class="btn-close box-content w-4 h-4 p-1
ml-auto text-yellow-900 border-none rounded-none opacity-50
focus:shadow-none focus:outline-none focus:opacity-100 hover:text-
yellow-900 hover:opacity-75 hover:no-underline"
                                data-bs-dismiss="alert"                                aria-
label="Close"></button>
                                </div>
                                <% } %>
                                <table class="min-w-full text-white table-auto ">
                                <thead class="border-b border-sky-500 font-
semibold">
                                <tr>
                                <th scope="col" class="text-sm font-
medium px-6 py-4 text-left">
                                #
                                </th>
                                <th scope="col" class="text-sm font-
medium px-6 py-4 text-left">
                                Title
                                </th>
                                <th scope="col" class="text-sm font-
medium px-6 py-4 text-left">
                                Uploader

```

```

        </th>
        <th scope="col" class="text-sm font-
medium px-6 py-4 text-left">
            Action
        </th>
    </tr>
</thead>
<tbody id="table-file">
    <% allFile.forEach((f, i) => { %>
        <%- include('table-file-row', { f : f, i :
i, user : allUser }) %>
        <% }) %>
    </tbody>
</table>
<div class="paginaton mt-3 min-w-full">
    <div class="flex justify-center">
        <nav aria-label="Page navigation example">
            <ul class="flex list-style-none gap-
1">
                <li class="page-item-file"><a
                    class="page-link relative
block py-1.5 px-3 rounded border-0 bg-transparent outline-none
transition-all duration-300 text-white hover:text-white hover:bg-
sky-900 focus:shadow-none"
                        href="#"                                aria-
label="First" data-page="false" data-click="false">
                            <span                                aria-
hidden="true">First</span>
                                </a></li>
                <li class="page-item-file"><a
                    class="page-link relative
block py-1.5 px-3 rounded border-0 bg-transparent outline-none
transition-all duration-300 text-white hover:text-white hover:bg-
sky-900 focus:shadow-none"
                        href="#"                                aria-
label="Previous" data-page="false" data-click="false">
                            <span                                aria-
hidden="true">&laquo;</span>
                                </a></li>
                <li class="page-item-file"><a
                    class="page-link relative
block py-1.5 px-3 rounded border-0 bg-sky-900 outline-none
transition-all duration-300 text-white hover:text-white hover:bg-
sky-900 focus:shadow-none"
                        data-click="false"
href="#">1</a></li>
                <li class="page-item-file"><a
                    class="page-link relative
block py-1.5 px-3 rounded border-0 bg-transparent outline-none
transition-all duration-300 text-white hover:text-white hover:bg-
sky-900 focus:shadow-none"

```

```

data-click="false"
href="#">2</a></li>
    <li class="page-item-file"><a
      class="page-link relative
block py-1.5 px-3 rounded border-0 bg-transparent outline-none
transition-all duration-300 text-white hover:text-white hover:bg-
sky-900 focus:shadow-none"
      data-click="false"
href="#">3</a></li>
    <li class="page-item-file"><a
      class="page-link relative
block py-1.5 px-3 rounded border-0 bg-transparent outline-none
transition-all duration-300 text-white hover:text-white hover:bg-
sky-900 focus:shadow-none"
      href="#" aria-label="Next"
data-click="false">
      <span
        aria-
hidden="true">&raquo;</span>
    </a></li>
    <li class="page-item-file"><a
      class="page-link relative
block py-1.5 px-3 rounded border-0 bg-transparent outline-none
transition-all duration-300 text-white hover:text-white hover:bg-
sky-900 focus:shadow-none"
      href="#" aria-label="Last"
data-page="false" data-click="false">
      <span
        aria-
hidden="true">Last</span>
    </a></li>
  </ul>
</nav>
</div>
</div>
</div>
</div>
</div>
</section>

```

views/partial/table-file-row.ejs

```

<tr class="border-b border-sky-500">
  <td class="px-6 py-4 whitespace-nowrap text-sm font-medium"><%=
i + 1 %> </td>
  <td class="text-sm font-light px-6 py-4">
    <%= f.name %>
  </td>
  <td class="text-sm font-light px-6 py-4 whitespace-nowrap">
    <% if (typeof user === 'string') { %>
    <%= user %>
    <% } else { %>
    <% const index = user.findIndex(o => { %>

```

```

        <% return o._id.toHexString() === f.uploader %>
        <% }) %>
        <%= user[index].name %>
        <% } %>

    </td>
    <td class="text-sm font-light px-6 py-4 whitespace-nowrap">
        <a href="docs/<%= f.fileName %> " target="_blank"
            class="w-full mx-1 bg-sky-700 p-3 rounded-lg text-white
            hover:bg-green-500">
            <i class="fa-solid fa-eye"></i>
        </a>
        <a href="" class="w-full mx-1 bg-sky-700 p-3 rounded-lg
            text-white hover:bg-red-500 delete-file"
            data-id="<%= f._id %>" data-name="<%= f.name %>" data-
            bs-toggle="modal" data-bs-target="#deleteFileModal">
            <i class="fa-solid fa-trash"></i>
        </a>
    </td>
</tr>

```

views/partial/table-user-row.ejs

```

<tr class="border-b border-sky-500">
    <td class="px-6 py-4 whitespace-nowrap text-sm font-medium"><%=
    i + 1 %> </td>
    <td class="text-sm font-light px-6 py-4">
        <%= u.name %>
    </td>
    <td class="text-sm font-light px-6 py-4 whitespace-nowrap">
        <%= u.uid %>
    </td>
    <td class="text-sm font-light px-6 py-4 whitespace-nowrap">
        <a href="" data-id="<%= u._id %> " data-uid="<%= u.uid %>"
        data-name="<%= u.name %>"
            class="w-full mx-1 bg-sky-700 p-3 rounded-lg text-white
            hover:bg-yellow-500 user-change"
            data-bs-toggle="modal" data-bs-target="#addUserModal">
            <i class="fa-solid fa-pencil"></i>
        </a>
        <a href="" class="w-full mx-1 bg-sky-700 p-3 rounded-lg
            text-white hover:bg-red-500 delete-user"
            data-id="<%= u._id %>" data-name="<%= u.name %> " data-
            bs-toggle="modal" data-bs-target="#deleteUserModal">
            <i class="fa-solid fa-trash"></i>
        </a>
    </td>
</tr>

```

-views/admin-page.ejs

```

<%- include('partials/description') %>

<%- include('partials/table-details') %>

<%- include('partials/add-user') %>

<%- include('partials/add-files') %>

<script defer>
  // User Modal
  const userIdInput = document.querySelector('#id_user')
  const userIdName = document.querySelector('#username_user')
  const userIdNim = document.querySelector('#nim_user')
  const userIdPass = document.querySelector('#password_user')
  // const userChange = document.querySelectorAll('.user-change')
  const modalUser = document.querySelector('#modal-user')
  const userModalButton = document.querySelector('#modal-sub-
user')
  const formUserModal = document.querySelector('#form-user')
  const modalTitle = document.querySelector('#exampleModalScrollableLabel')
  // const deleteUser = document.querySelectorAll('.delete-user')
  const idDeleteUser = document.querySelector('#del-user-id')
  const nameDeleteUser = document.querySelector('#del-user-
name')
  const deleteText = document.querySelector('#text-delete')
  // User Modal

  // File Modal
  const deleteFile = document.querySelectorAll('.delete-file')
  const idDeleteFile = document.querySelector('#del-file-id')
  const nameDeleteFile = document.querySelector('#del-file-
title')
  const deleteTextFile = document.querySelector('#text-delete-
file')
  // File Modal

  // File Modal
  deleteFile.forEach(file => {
    file.addEventListener('click', e => {
      e.preventDefault()
      idDeleteFile.value = file.getAttribute('data-id')
      nameDeleteFile.value = file.getAttribute('data-name')
      deleteTextFile.textContent =
        `Are You Sure Want Delete File With Title
${file.getAttribute('data-name')} ?`
    })
  })
  // File Modal

```

```

// User Modal
modalUser.addEventListener('click', (e) => {
  formUserModal.action = '/admin-user'
  e.preventDefault()
  userIdInput.value = ''
  userIdNim.value = ''
  userIdName.value = ''
  userIdPass.disabled = false
  modalTitle.textContent = 'Add User'
  userModalButton.textContent = 'Add User'
})

// User Modal

// User Tabel
const tableUser = document.querySelector('#table-user')
const navigationUser = document.querySelectorAll('.page-item')
let currentPage = 0
let totalPages = 0
const limit = 5
updateUserTable(currentPage)

// File Tabel
const tableFile = document.querySelector('#table-file')
const navigationFile = document.querySelectorAll('.page-item-
file')
let currentPageFile = 0
let totalPagesFile = 0
const limitFile = 5
updateFileTable(currentPageFile)

async function updateFileTable(page) {
  const result = await
getApi(`/file?limit=${limitFile}&pages=${page}`)
  const users = await getApi('/all-user')
  // return console.log(user)
  const data = result.data
  currentPageFile = result.currentPage
  totalPagesFile = result.totalPages
  tableFile.innerHTML = ''
  let number = (currentPageFile * limitFile) + 1
  for (const file of data) {
    const uploader = users.find(function (user) {
      return user._id === file.uploader
    })
    // console.log(uploader)
    tableFile.innerHTML += `
    <tr class="border-b border-sky-500">
      <td class="px-6 py-4 whitespace-nowrap text-sm
font-medium">${number}</td>

```

```

        <td class="text-sm font-light px-6 py-4 whitespace-
nowrap">${file.name}</td>
        <td class="text-sm font-light px-6 py-4 whitespace-
nowrap">${uploader.name}</td>
        <td class="text-sm font-light px-6 py-4 whitespace-
nowrap">
            <a
                href="docs/${file.fileName}"
                target="_blank"
                class="w-full mx-1 bg-sky-700 p-3 rounded-
lg text-white hover:bg-green-500">
                <i class="fa-solid fa-eye"></i>
            </a>
            <a href="" class="w-full mx-1 bg-sky-700 p-3
rounded-lg text-white hover:bg-red-500 delete-file"
                data-id="${file._id}" data-
name="${file.name}" data-bs-toggle="modal" data-bs-
target="#deleteFileModal">
                <i class="fa-solid fa-trash"></i>
            </a>
        </td>
    </tr>
    `
    number++
}
const deleteFile = document.querySelectorAll('.delete-
file')
deleteFile.forEach(file => {
    file.addEventListener('click', e => {
        e.preventDefault()
        idDeleteFile.value = file.getAttribute('data-id')
        nameDeleteFile.value = file.getAttribute('data-
name')
        deleteTextFile.textContent =
            `Are You Sure Want Delete File With Title
${file.getAttribute('data-name')} ?`
    })
})

let newCurrentPage = getNumberPage(currentPageFile,
totalPageFile)
for (const nav of navigationFile) {
    if (nav.children[0].ariaLabel === 'Previous') {
        nav.children[0].dataset.page = (currentPageFile -
1)
        if (nav.children[0].dataset.click === 'false') {
            pagingButtonListenerFile(nav.children[0],
'prev')
        }
    } else if (nav.children[0].ariaLabel === 'Next') {
        nav.children[0].dataset.page = (currentPageFile +
1)

```

```

        if (nav.children[0].dataset.click === 'false') {
            pagingButtonListenerFile(nav.children[0],
'next')
        }
    } else if (nav.children[0].ariaLabel === 'First') {
        if (nav.children[0].dataset.click === 'false') {
            nav.addEventListener('click', e => {
                if (currentPageFile === 0) return
e.preventDefault()
                updatePage(e, 1, 'file')
            })
            nav.children[0].dataset.click = 'true'
        }
    } else if (nav.children[0].ariaLabel === 'Last') {
        if (nav.children[0].dataset.click === 'false') {
            nav.addEventListener('click', e => {
                if (currentPageFile === totalPageFile - 1)
return e.preventDefault()
                updatePage(e, totalPageFile, 'file')
            })
            nav.children[0].dataset.click = 'true'
        }
    } else {
        nav.children[0].textContent = newCurrentPage
        if (currentPageFile + 1 === newCurrentPage) {
            removeIfExists(nav.children[0], 'bg-
transparent')
            addIfNotExist(nav.children[0], 'bg-sky-900')
        } else {
            removeIfExists(nav.children[0], 'bg-sky-900')
            addIfNotExist(nav.children[0], 'bg-
transparent')
        }
    }

    if (newCurrentPage > totalPageFile)
nav.style.display = 'none'
    // console.log(nav.children[0].dataset.click)
    if (nav.children[0].dataset.click === 'false') {
        pagingButtonListenerFile(nav.children[0])
    }
    newCurrentPage++
}

}

}

async function updateUserTable(page) {
    const result = await
getApi(`/user?limit=${limit}&pages=${page}`)
    const data = result.data

```



```

currentPage = result.currentPage
totalPage = result.totalPages
tableUser.innerHTML = ''
let number = (currentPage * limit) + 1
for (const user of data) {
  tableUser.innerHTML += `
    <tr class="border-b border-sky-500">
      <td class="px-6 py-4 whitespace-nowrap text-sm
font-medium">${number} </td>
      <td class="text-sm font-light px-6 py-4">
        ${user.name}
      </td>
      <td class="text-sm font-light px-6 py-4
whitespace-nowrap">
        ${user.uid}
      </td>
      <td class="text-sm font-light px-6 py-4
whitespace-nowrap">
        <a href="" data-id="${user._id}" data-
uid="${user.uid}" data-name="${user.name}"
class="w-full mx-1 bg-sky-700 p-3
rounded-lg text-white hover:bg-yellow-500 user-change"
data-bs-toggle="modal" data-bs-
target="#addUserModal">
          <i class="fa-solid fa-pencil"></i>
        </a>
        <a href="" class="w-full mx-1 bg-sky-700 p-
3 rounded-lg text-white hover:bg-red-500 delete-user"
data-id="${user._id}" data-
name="${user.name}" data-bs-toggle="modal" data-bs-
target="#deleteUserModal">
          <i class="fa-solid fa-trash"></i>
        </a>
      </td>
    </tr>
  `
  number++
}
const userChange = document.querySelectorAll('.user-
change')
userChange.forEach(user => {
  user.addEventListener('click', (e) => {
    e.preventDefault()
    formUserModal.action = '/admin-user?_method=PUT'
    userIdInput.value = user.getAttribute('data-id')
    userIdName.value = user.getAttribute('data-name')
    userIdNim.value = user.getAttribute('data-uid')
    userIdPass.disabled = true
    userModalButton.textContent = 'Update User'
    modalTitle.textContent = 'Update User'
  })
})

```

```

    })

    const deleteUser = document.querySelectorAll('.delete-
user')
    deleteUser.forEach(user => {
        user.addEventListener('click', e => {
            e.preventDefault()
            idDeleteUser.value = user.getAttribute('data-id')
            nameDeleteUser.value = user.getAttribute('data-
name')

            deleteText.textContent =
                `Are You Sure Want Delete
${user.getAttribute('data-name')} ?`

        })
    })

    let newCurrentPage = getNumberPage(currentPage, totalPage)

    for (const nav of navigationUser) {
        if (nav.children[0].ariaLabel === 'Previous') {
            nav.children[0].dataset.page = (currentPage - 1)
            if (nav.children[0].dataset.click === 'false') {
                pagingButtonListener(nav.children[0], 'prev')
            }
        } else if (nav.children[0].ariaLabel === 'Next') {
            nav.children[0].dataset.page = (currentPage + 1)
            if (nav.children[0].dataset.click === 'false') {
                pagingButtonListener(nav.children[0], 'next')
            }
        } else if (nav.children[0].ariaLabel === 'First') {
            if (nav.children[0].dataset.click === 'false') {
                nav.addEventListener('click', e => {
                    if (currentPage === 0) return
                })
            }
            e.preventDefault()
            updatePage(e, 1, 'user')
            nav.children[0].dataset.click = 'true'
        } else if (nav.children[0].ariaLabel === 'Last') {
            if (nav.children[0].dataset.click === 'false') {
                nav.addEventListener('click', e => {
                    if (currentPage === totalPage - 1) return
                })
            }
            e.preventDefault()
            updatePage(e, totalPage, 'user')
            nav.children[0].dataset.click = 'true'
        } else {
            nav.children[0].textContent = newCurrentPage
            if (currentPage + 1 === newCurrentPage) {

```

```

        removeIfExist(nav.children[0], 'bg-
transparent')
        addIfNotExist(nav.children[0], 'bg-sky-900')
    } else {
        removeIfExist(nav.children[0], 'bg-sky-900')
        addIfNotExist(nav.children[0], 'bg-
transparent')
    }

    if (newCurrentPage > totalPage) nav.style.display
= 'none'

    // console.log(nav.children[0].dataset.click)
    if (nav.children[0].dataset.click === 'false') {
        pagingButtonListener(nav.children[0])
    }
    newCurrentPage++
}

}

function pagingButtonListener(nav, option = '') {
    nav.addEventListener('click', (e) => {
        e.preventDefault()
        if (option === 'next')
            if (currentPage === (totalPage - 1)) return
        if (option === 'prev')
            if (currentPage === 0) return
        updatePage(e, (Number(nav.dataset.page) + 1) ||
Number(nav.textContent), 'user')
    })
    nav.dataset.click = 'true'
}

function pagingButtonListenerFile(nav, option = '') {
    nav.addEventListener('click', (e) => {
        e.preventDefault()
        if (option === 'next')
            if (currentPageFile === (totalPageFile - 1)) return
        if (option === 'prev')
            if (currentPageFile === 0) return
        updatePage(e, (Number(nav.dataset.page) + 1) ||
Number(nav.textContent), 'file')
    })
    nav.dataset.click = 'true'
}

function updatePage(e, page, table) {
    e.preventDefault()
    console.log(page - 1)

```

```

        if (table === 'user') updateUserTable(page - 1)
        else if (table === 'file') updateFileTable(page - 1)
    }

    async function getApi(url) {
        const response = await fetch(url)
        const result = await response.json()
        return result
    }

    function getNumberPage(currentPage, totalPage) {
        if (currentPage == 0) {
            return currentPage + 1
        }
        if (currentPage + 1 == totalPage && totalPage > 2) {
            return currentPage - 1
        }
        return currentPage
    }

    function prevent(e) {
        e.preventDefault()
    }

    function removeIfExist(removeItem, removeClass) {
        if (removeItem.classList.contains(removeClass))
        removeItem.classList.remove(removeClass)
    }

    function addIfNotExist(addItem, addClass) {
        if (!addItem.classList.contains(addClass))
        addItem.classList.add(addClass)
    }
</script>

-views/home-login.ejs
<%- include('partials/description') %>

-views/home-page.ejs
<%- include('partials/description') %>

<% if (typeof user !== 'undefined') { if (user !== 'Admin') { %>

<div class="container flex flex-wrap justify-between">
    <div class="w-full bg-sky-800 lg:w-[49%] mt-5 rounded-md p-4
overflow-y-auto">
        <div class="flex flex-wrap flex-row items-center justify-
between">
            <h2 class="font-semibold text-xl mb-3 text-white">All
Files</h2>

```

```

        <button type="button" class="w-1/2 bg-sky-700 py-2
rounded-lg text-white hover:bg-sky-600"
        data-bs-toggle="modal" data-bs-
target="#addFilesModal">
        <i class="fas fa-plus mr-3"></i>Add
Files</button>
    </div>

    <% if(typeof errorsFile !== 'undefined') {
if(errorsFile.length !== 0) { %>
        <div class="alert mt-2 bg-red-300 rounded-lg py-5 px-6 mb-
3 text-base text-red-700 inline-flex items-center w-full alert-
dismissible fade show"
        role="alert">
        <%= errorsFile %>
        <button type="button"
        class="btn-close box-content w-4 h-4 p-1 ml-auto
text-yellow-900 border-none rounded-none opacity-50 focus:shadow-
none focus:outline-none focus:opacity-100 hover:text-yellow-900
hover:opacity-75 hover:no-underline"
        data-bs-dismiss="alert" aria-
label="Close"></button>
        </div>
        <% } } %>

        <% if(typeof successFile !== 'undefined') {
if(successFile.length !== 0) { %>
        <div class="alert mt-2 bg-green-300 rounded-lg py-5 px-6
mb-3 text-base text-green-700 inline-flex items-center w-full
alert-dismissible fade show"
        role="alert">
        <%= successFile %>
        <button type="button"
        class="btn-close box-content w-4 h-4 p-1 ml-auto
text-yellow-900 border-none rounded-none opacity-50 focus:shadow-
none focus:outline-none focus:opacity-100 hover:text-yellow-900
hover:opacity-75 hover:no-underline"
        data-bs-dismiss="alert" aria-
label="Close"></button>
        </div>
        <% } } %>

        <table class="min-w-full text-white table-auto">
        <thead class="border-b border-sky-500 font-semibold">
        <tr>
        <th scope="col" class="text-sm font-medium px-
6 py-4 text-left">
        #
        </th>
        <th scope="col" class="text-sm font-medium px-
6 py-4 text-left">

```

```

        Title
      </th>
      <!-- <th scope="col" class="text-sm font-medium
px-6 py-4 text-left">
        Uploader
      </th> -->
      <th scope="col" class="text-sm font-medium px-
6 py-4 text-left">
        Action
      </th>
    </tr>
  </thead>
  <tbody id="table-file">
    <!-- <%# file.forEach((f, i) => { %> -->
    <!-- <%# include('partials/table-file-row', { f :
f, i : i }) %> -->
    <!-- <# }) %> -->
  </tbody>
</table>
<div class="paginaton mt-3 min-w-full">
  <div class="flex justify-center">
    <nav aria-label="Page navigation example">
      <ul class="flex list-style-none gap-1">
        <li class="page-item-file"><a
          class="page-link relative block
py-1.5 px-3 rounded border-0 bg-transparent outline-none
transition-all duration-300 text-white hover:text-white hover:bg-
sky-900 focus:shadow-none"
          href="#" aria-label="First" data-
page="false" data-click="false">
            <span
              aria-
hidden="true">First</span>
          </a></li>
        <li class="page-item-file"><a
          class="page-link relative block
py-1.5 px-3 rounded border-0 bg-transparent outline-none
transition-all duration-300 text-white hover:text-white hover:bg-
sky-900 focus:shadow-none"
          href="#" aria-label="Previous"
data-page="false" data-click="false">
            <span
              aria-
hidden="true">&laquo;</span>
          </a></li>
        <li class="page-item-file"><a
          class="page-link relative block
py-1.5 px-3 rounded border-0 bg-sky-900 outline-none transition-all
duration-300 text-white hover:text-white hover:bg-sky-900
focus:shadow-none"
          data-click="false"
href="#">1</a></li>
        <li class="page-item-file"><a

```

```

        class="page-link relative block
py-1.5 px-3 rounded border-0 bg-transparent outline-none
transition-all duration-300 text-white hover:text-white hover:bg-
sky-900 focus:shadow-none"
        data-click="false"
href="#">2</a></li>
        <li class="page-item-file"><a
        class="page-link relative block
py-1.5 px-3 rounded border-0 bg-transparent outline-none
transition-all duration-300 text-white hover:text-white hover:bg-
sky-900 focus:shadow-none"
        data-click="false"
href="#">3</a></li>
        <li class="page-item-file"><a
        class="page-link relative block
py-1.5 px-3 rounded border-0 bg-transparent outline-none
transition-all duration-300 text-white hover:text-white hover:bg-
sky-900 focus:shadow-none"
        href="#" aria-label="Next" data-
click="false">
        <span
        aria-
hidden="true">&raquo;</span>
        </a></li>
        <li class="page-item-file"><a
        class="page-link relative block
py-1.5 px-3 rounded border-0 bg-transparent outline-none
transition-all duration-300 text-white hover:text-white hover:bg-
sky-900 focus:shadow-none"
        href="#" aria-label="Last" data-
page="false" data-click="false">
        <span
        aria-
hidden="true">Last</span>
        </a></li>
    </ul>
</nav>
</div>
</div>
</div>
</div>

<div class="w-full h-max-[40px] bg-sky-800 lg:w-[49%] mt-5
rounded-md p-4">
    <div class="flex flex-wrap flex-row items-center justify-
between">
        <h2 class="font-semibold text-xl mb-3 text-
white">Profile</h2>
    </div>

    <% if(typeof errorsUpdate !== 'undefined') {
if(errorsUpdate.length !== 0) { %>

```

```

        <div class="alert mt-2 bg-red-300 rounded-lg py-5 px-6 mb-
3 text-base text-red-700 inline-flex items-center w-full alert-
dismissible fade show"
            role="alert">
                <%= errorsUpdate %>
                <button type="button"
                    class="btn-close box-content w-4 h-4 p-1 ml-auto
text-yellow-900 border-none rounded-none opacity-50 focus:shadow-
none focus:outline-none focus:opacity-100 hover:text-yellow-900
hover:opacity-75 hover:no-underline"
                    data-bs-dismiss="alert"                                aria-
label="Close"></button>
            </div>
            <% } } %>

            <% if(typeof successUpdate !== 'undefined') {
if(successUpdate.length !== 0) { %>
                <div class="alert mt-2 bg-green-300 rounded-lg py-5 px-6
mb-3 text-base text-green-700 inline-flex items-center w-full
alert-dismissible fade show"
                    role="alert">
                        <%= successUpdate %>
                        <button type="button"
                            class="btn-close box-content w-4 h-4 p-1 ml-auto
text-yellow-900 border-none rounded-none opacity-50 focus:shadow-
none focus:outline-none focus:opacity-100 hover:text-yellow-900
hover:opacity-75 hover:no-underline"
                            data-bs-dismiss="alert"                                aria-
label="Close"></button>
                    </div>
                    <% } } %>

            <form action="/user?_method=PUT" method="POST">
                <input type="hidden" name="_id" value="<%= details._id
%>">

                <div class="w-full px-4 mb-8">
                    <label for="id" class="text-base font-thin text-
white">Id (NIM)</label>
                    <input type="text" name="nim" id="id" value="<%=
details.uid %>"
                        class="w-full bg-sky-700 text-sky-200 p-3
rounded-md focus:outline-none focus:ring-sky-400 focus:ring-1" />
                    </div>
                    <div class="w-full px-4 mb-8">
                        <label for="id" class="text-base font-thin text-
white">Name</label>
                        <input type="text" name="name" id="name" value="<%=
details.name %>"
                            class="w-full bg-sky-700 text-sky-200 p-3
rounded-md focus:outline-none focus:ring-sky-400 focus:ring-1" />
                    </div>

```



```

        <div class="w-full px-4 mb-8">
            <label for="password" class="text-base font-thin
text-white">Password</label>
            <input type="password" name="password"
id="password" class="w-full bg-sky-700 text-sky-200 p-3 rounded-
md focus:outline-none
                focus:ring-sky-400 focus:ring-1" />
        </div>
        <button type="submit" class="w-full bg-sky-700 py-2
rounded-lg text-white hover:bg-sky-600">Update
        Profile</button>
    </form>

</div>
</div>

<div class="modal fade fixed top-0 left-0 hidden w-full h-full
outline-none overflow-x-hidden overflow-y-auto "
    id="addFilesModal" tabindex="-1" aria-
labelledby="addFilesModalTitle" aria-modal="true" role="dialog">
    <div class="modal-dialog modal-dialog-centered relative w-auto
pointer-events-none">
        <div
            class="modal-content border-none shadow-lg relative
flex flex-col w-full pointer-events-auto bg-sky-800 bg-clip-padding
rounded-md outline-none text-current backdrop-filter backdrop-blur-
lg bg-opacity-30">
            <div
                class="modal-header flex flex-shrink-0 items-
center justify-between p-4 border-b border-gray-700 rounded-t-md
backdrop-filter backdrop-blur-lg bg-opacity-30">
                <h5 class="text-xl font-medium leading-normal text-
sky-200" id="exampleModalScrollableLabel">
                    Add Files
                </h5>
                <button type="button"
                    class="btn-close box-content w-4 h-4 p-1 text-
black border-none rounded-none opacity-50 focus:shadow-none
focus:outline-none focus:opacity-100 hover:text-black
hover:opacity-75 hover:no-underline"
                    data-bs-dismiss="modal" aria-
label="Close"></button>
            </div>
            <div class="modal-body relative bg-sky-800 p-4">
                <form action="/pdf" method="POST"
enctype="multipart/form-data">
                    <div class="w-full px-4 mb-8">
                        <label for="id" class="text-base font-thin
text-white">Title</label>
                        <input type="text" name="title" id="id"

```

```

        class="w-full bg-sky-700 text-sky-200
p-3 rounded-md focus:outline-none focus:ring-sky-400 focus:ring-1"
/>
    </div>
    <div class="w-full my-5 px-4 flex items-center
justify-between text-sky-200">
        <label for="upload">Upload Your File
(*.pdf)</label>
        <input
            class="file:bg-sky-700 file:border-0
file:text-white file:rounded-md file:p-2 file:hover:bg-sky-600"
            type="file" name="upload">
    </div>
</div>
<div
    class="modal-footer flex flex-shrink-0 flex-wrap
items-center justify-end p-4 border-t border-gray-700 rounded-b-md
bg-sky-800">
    <button type="submit" class="w-full bg-sky-700 py-
2 rounded-lg text-white hover:bg-sky-600">
        Add File
    </button>
</form>
</div>
</div>
</div>
</div>

<div class="modal fade fixed top-0 left-0 hidden w-full h-full
outline-none overflow-x-hidden overflow-y-auto "
    id="deleteFileModal" tabindex="-1" aria-
labelledby="deleteFileModalTitle" aria-modal="true" role="dialog">
    <div class="modal-dialog modal-dialog-centered relative w-auto
pointer-events-none bg-opacity-30">
        <div
            class="modal-content border-none shadow-lg relative
flex flex-col w-full pointer-events-auto bg-sky-800 bg-clip-padding
rounded-md outline-none text-current backdrop-filter backdrop-blur-
lg bg-opacity-30">
            <div
                class="modal-header flex flex-shrink-0 items-
center justify-between p-4 border-b border-gray-700 rounded-t-md
backdrop-filter backdrop-blur-lg bg-opacity-30">
                <h5 class="text-xl font-medium leading-normal text-
sky-200" id="exampleModalScrollableLabel">
                    Delete File
                </h5>
                <button type="button"
                    class="btn-close box-content w-4 h-4 p-1 text-
white border-none rounded-none opacity-50 focus:shadow-none

```

```

focus:outline-none          focus:opacity-100          hover:text-black
hover:opacity-75 hover:no-underline"
          data-bs-dismiss="modal"          aria-
label="Close"></button>
</div>
<div class="modal-body relative bg-sky-800 p-4
backdrop-filter backdrop-blur-lg bg-opacity-30">
  <p id="text-delete-file" class="text-white text-
base"></p>
  <form action="/pdf?_method=DELETE" method="POST">
    <input type="hidden" id="del-file-id"
name="id" value="">
    <input type="hidden" id="del-file-title"
name="title" value="">
  </div>
  <div
    class="modal-footer flex flex-shrink-0 flex-wrap
items-center justify-end p-4 border-t border-gray-700 rounded-b-md
bg-sky-800 backdrop-filter backdrop-blur-lg bg-opacity-30">
    <button type="submit" class="w-full bg-sky-700 py-
2 rounded-lg text-white hover:bg-sky-600"
      id="modal-sub-user">
      Delete File
    </button>
  </form>
</div>
</div>
</div>
</div>

<script>
  const idDeleteFile = document.querySelector('#del-file-id')
  const nameDeleteFile = document.querySelector('#del-file-
title')
  const deleteTextFile = document.querySelector('#text-delete-
file')

  const tableFile = document.querySelector('#table-file')
  const navigationFile = document.querySelectorAll('.page-item-
file')
  let currentPageFile = 0
  let totalPageFile = 0
  const limitFile = 5
  updateFileTable(currentPageFile)

  async function updateFileTable(page) {
    const result = await
getApi(`/file?limit=${limitFile}&pages=${page}`)
    // return console.log(result)
    const data = result.data
    currentPageFile = result.currentPage

```

```

totalPageFile = result.totalPages
tableFile.innerHTML = ''
let number = (currentPageFile * limitFile) + 1
for (const file of data) {
  tableFile.innerHTML += `
    <tr class="border-b border-sky-500">
      <td class="px-6 py-4 whitespace-nowrap text-sm
font-medium">${number}</td>
      <td class="text-sm font-light px-6 py-4 whitespace-
nowrap">${file.name}</td>
      <td class="text-sm font-light px-6 py-4 whitespace-
nowrap">
        <a
          href="docs/${file.fileName}"
          target="_blank"
          class="w-full mx-1 bg-sky-700 p-3 rounded-
lg text-white hover:bg-green-500">
            <i class="fa-solid fa-eye"></i>
          </a>
          <a href="" class="w-full mx-1 bg-sky-700 p-3
rounded-lg text-white hover:bg-red-500 delete-file"
            data-id="${file._id}"
            data-name="${file.name}"
            data-bs-toggle="modal"
            data-bs-target="#deleteFileModal">
              <i class="fa-solid fa-trash"></i>
            </a>
          </td>
        </tr>
      `
  number++
}
const deleteFile = document.querySelectorAll('.delete-
file')
deleteFile.forEach(file => {
  file.addEventListener('click', e => {
    e.preventDefault()
    idDeleteFile.value = file.getAttribute('data-id')
    nameDeleteFile.value = file.getAttribute('data-
name')
    deleteTextFile.textContent =
      `Are You Sure Want Delete File With Title
${file.getAttribute('data-name')} ?`
  })
})

let newCurrentPage = getNumberPage(currentPageFile,
totalPageFile)
for (const nav of navigationFile) {
  if (nav.children[0].ariaLabel === 'Previous') {
    nav.children[0].dataset.page = (currentPageFile -
1)

    if (nav.children[0].dataset.click === 'false') {

```

```

        pagingButtonListener(nav.children[0], 'prev')
    }
    } else if (nav.children[0].ariaLabel === 'Next') {
        nav.children[0].dataset.page = (currentPageFile +
1)
        if (nav.children[0].dataset.click === 'false') {
            pagingButtonListener(nav.children[0], 'next')
        }
    } else if (nav.children[0].ariaLabel === 'First') {
        if (nav.children[0].dataset.click === 'false') {
            nav.addEventListener('click', e => {
                if (currentPageFile === 0) return
e.preventDefault()
                updatePage(e, 1)
            })
            nav.children[0].dataset.click = 'true'
        }
    } else if (nav.children[0].ariaLabel === 'Last') {
        if (nav.children[0].dataset.click === 'false') {
            nav.addEventListener('click', e => {
                if (currentPageFile === totalPageFile - 1)
return e.preventDefault()
                updatePage(e, totalPageFile)
            })
            nav.children[0].dataset.click = 'true'
        }
    } else {
        nav.children[0].textContent = newCurrentPage
        if (currentPageFile + 1 === newCurrentPage) {
            removeIfExists(nav.children[0], 'bg-
transparent')
            addIfNotExist(nav.children[0], 'bg-sky-900')
        } else {
            removeIfExists(nav.children[0], 'bg-sky-900')
            addIfNotExist(nav.children[0], 'bg-
transparent')
        }
    }

    if (newCurrentPage > totalPageFile)
nav.style.display = 'none'
    // console.log(nav.children[0].dataset.click)
    if (nav.children[0].dataset.click === 'false') {
        pagingButtonListener(nav.children[0])
    }
    newCurrentPage++
}

}

}

```

```

function pagingButtonListener(nav, option = '') {
  nav.addEventListener('click', (e) => {
    e.preventDefault()
    if (option === 'next')
      if (currentPageFile === (totalPageFile - 1)) return
    if (option === 'prev')
      if (currentPageFile === 0) return
    updatePage(e, (Number(nav.dataset.page) + 1) ||
Number(nav.textContent))
  })
  nav.dataset.click = 'true'
}

function updatePage(e, page) {
  e.preventDefault()
  updateFileTable(page - 1)
}

async function getApi(url) {
  const response = await fetch(url)
  const result = await response.json()
  return result
}

function getNumberPage(currentPage, totalPage) {
  if (currentPage == 0) {
    return currentPage + 1
  }
  if (currentPage + 1 == totalPage && totalPage > 2) {
    return currentPage - 1
  }
  return currentPage
}

function prevent(e) {
  e.preventDefault()
}

function removeIfExist(removeItem, removeClass) {
  if (removeItem.classList.contains(removeClass))
removeItem.classList.remove(removeClass)
}

function addIfNotExist(addItem, addClass) {
  if (!addItem.classList.contains(addClass))
addItem.classList.add(addClass)
}
</script>

<% } } %>

```

```

<%- include('partials/result-none') %>

<script defer>
  "use strict"
  document.addEventListener('DOMContentLoaded', () => {
    const checkPlags = document.querySelector('#check')
    const uploadPlags = document.querySelector('#upload-res')
    const resultSection = document.querySelector('#result-
none')
    const spinner = document.querySelector('#spinner')
    const spinnerT = document.querySelector('#loading-number')

    const fileName = document.querySelector('#file-name')
    const fileSize = document.querySelector('#file-size')
    const fileChar = document.querySelector('#file-char')
    const fileWord = document.querySelector('#file-word')

    const textUniq = document.querySelector('#text-uniq')
    const textPlag = document.querySelector('#text-plag')
    const barUniq = document.querySelector('#bar-uniq')
    const barPlag = document.querySelector('#bar-plag')
    const previewPlag = document.querySelector('#preview')
    const errorDokumen = document.querySelector('#error-file')

    const tableFile = document.querySelector('#table-result')

    checkPlags.addEventListener('click', e => {
      e.preventDefault()
      if (uploadPlags.files.length === 0) return
      if (spinner.classList.contains('hidden')) {
        spinner.classList.remove('hidden')
        spinner.classList.add('flex')
      }
      if (!resultSection.classList.contains('hidden'))
resultSection.classList.add('hidden')
      const formData = new FormData()
      formData.append('upload-res', uploadPlags.files[0])
      formData.append('text',
document.querySelector('textarea').value)
      const [file] = uploadPlags.files
      if (errorDokumen.classList.contains('inline-flex')) {
        errorDokumen.classList.add('hidden')
        errorDokumen.classList.remove('inline-flex')
      }
      if (file.name.split('.')[1].toLowerCase() !== 'pdf') {
        spinner.classList.add('hidden')
        spinner.classList.remove('flex')
        if (errorDokumen.classList.contains('hidden')) {
          errorDokumen.classList.remove('hidden')
          errorDokumen.classList.add('inline-flex')
        }
      }
    })
  })

```

```

        return console.log('gagal')
    }
    //
    console.log(file.name.split('.')[1].toLowerCase() === 'pdf')
    previewPlag.src = URL.createObjectURL(file)

    fetch('/result-none', {
        method: 'post',
        body: formData
    }).then(res => {
        return res.json()
    }).then(async res => {
        resultSection.classList.remove('hidden')
        const data = res.data
        const trainData = res.file
        const user = res.user

        let allData = {}
        let all = []

        addDict(data, all)
        trainData.forEach(data => {
            addDict(data.text, all)
        })

        all = [...new Set(all)]
        all.forEach(e => allData[e] = {})

        data.split(' ').forEach(e => {
            allData[e]['s'] = (allData[e]['s'] || 0) + 1
        })

        trainData.forEach((str, i) => {
            str.text.split(' ').forEach(f => {
                allData[f]['s' + (i + 1)] = (allData[f]['s'
+
                    (i + 1)
                ] || 0) + 1
            })
        })

        let resultAll = []

        let averageSimilarity = 0
        const keys = Object.keys(allData)
        for (let i = 0; i < trainData.length; i++) {
            let e = 's' + (i + 1)
            let aTimesB = 0
            let aa = 0
            let bb = 0

```



```

        keys.forEach(f => {
            const a = allData[f]['s'] || 0
            const b = allData[f][e] || 0
            aTimesB += a * b
            aa += a * a
            bb += b * b
        })

        const result = (2 * aTimesB) / (aa + bb)
        // console.log(
        //     `Similarity Data ${i + 1} Is
        ${Math.round(result * 100)} %`
        // )
        resultAll.push({
            file: trainData[i],
            result: Math.round(result * 100)
        })
        averageSimilarity += Math.round(result * 100)
    }
    averageSimilarity /= trainData.length
    averageSimilarity = Math.round(averageSimilarity)
    spinnerT.textContent = 'Selesai'
    await sleep()
    if (spinner.classList.contains('flex')) {
        spinner.classList.remove('flex')
        spinner.classList.add('hidden')
    }
    resultAll.sort((a, b) => (a.result < b.result) ? 1
: -1)

    // console.log(resultAll, averageSimilarity)
    // resultAll = resultAll.filter(result => {
    //     return result.result > 40
    // })
    resultAll = resultAll.slice(0, 10)

    fileName.textContent = 'File Name : ' +
res.fileUpload.name
    fileSize.textContent = 'File Size : ' +
formatBytes(res.fileUpload.size)
    fileChar.textContent = 'Character : ' +
separator(res.fileUpload.char)
    fileWord.textContent = 'Words : ' +
separator(res.fileUpload.word)

    textUniq.textContent = 100 - averageSimilarity
    textPlag.textContent = averageSimilarity
    barUniq.textContent = 100 - averageSimilarity + '
%'

    barUniq.style.width = 100 - averageSimilarity + '%'
    barPlag.textContent = averageSimilarity + '%'
    barPlag.style.width = averageSimilarity + '%'

```

```

tableFile.innerHTML = ''
resultAll.forEach((res, i) => {
  const index = user.findIndex(o => {
    return o._id === res.file.uploader
  })
  const tr = document.createElement('tr')
  tr.classList.add('border-b', 'border-sky-500')

  const tdFileNumber =
document.createElement('td')
  tdFileNumber.classList.add('px-6', 'py-4',
'whitespace-nowrap',
    'text-sm', 'font-medium')
  tdFileNumber.textContent = i + 1
  tr.appendChild(tdFileNumber)

  const tdFileName =
document.createElement('td')
  tdFileName.classList.add('text-sm', 'font-
light', 'px-6',
    'py-4')
  tdFileName.textContent = res.file.name
  tr.appendChild(tdFileName)

  const tdFileUploader =
document.createElement('td')
  tdFileUploader.classList.add('text-sm', 'font-
light', 'px-6',
    'py-4')
  tdFileUploader.textContent = user[index].name
  tr.appendChild(tdFileUploader)

  const tdFileResult =
document.createElement('td')
  tdFileResult.classList.add('text-sm', 'font-
light', 'px-6',
    'py-4', 'whitespace-nowrap')
  tdFileResult.textContent = res.result + ' %'
  tr.appendChild(tdFileResult)

  const tdFileShow =
document.createElement('td')
  tdFileShow.classList.add('text-sm', 'font-
light', 'px-6',
    'py-4', 'whitespace-nowrap')
  const link = document.createElement('a')
  link.classList.add('w-full', 'bg-sky-700', 'p-
3',
    'rounded-lg', 'text-white')
  link.textContent = 'Show'

```

```

        link.href = 'docs/' + res.file.fileName
        link.target = '_blank'
        tdFileShow.appendChild(link)
        tr.appendChild(tdFileShow)

        tableFile.appendChild(tr)
    })
})

function sleep() {
    return new Promise(resolve => setTimeout(resolve, 100))
}

function formatBytes(bytes, decimals = 2) {
    if (bytes === 0) return '0 Bytes'

    const k = 1024
    const dm = decimals < 0 ? 0 : decimals
    const sizes = ['Bytes', 'Kb', 'Mb', 'Gb', 'Tb', 'Pb',
'Eb', 'Zb', 'Yb']

    const i = Math.floor(Math.log(bytes) / Math.log(k))

    return parseFloat((bytes / Math.pow(k, i)).toFixed(dm))
+ ' ' + sizes[i]
}

function addDict(datax, to) {
    to.push(...datax.toLowerCase().split(' '))
}
// function addDict(datax, to) {
//     datax.toLowerCase().split(' ').forEach(e => {
//         e.trim()
//         if (!to.hasOwnProperty(e)) to[e] = {}
//     })
// }

function separator(num) {
    let str = num.toString().split(".");
    str[0] = str[0].replace(/B(?=(\d{3})+(?!\\d))/g, ".");
    return str.join(".");
}

})
</script>

-views/index.ejs
<!DOCTYPE html>
<html lang="en">

```

```

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>PlagCheck</title>
  <link rel="stylesheet" href="csss/output.css">
  <script      src="https://kit.fontawesome.com/524dd30d0c.js"
crossorigin="anonymous"></script>
</head>

<body class="bg-dark">

  <!-- Nav Section -->
  <header
    class="bg-slate-900 top-0 left-0 w-full flex items-center
p-2 backdrop-filter backdrop-blur-lg bg-opacity-30 border-b border-
gray-800 sticky">
    <div class="container">
      <div class="flex items-center justify-between">
        <div class="px-4">
          <a href="" class="block font-bold lg:text-xl
text-white py-4">Plagiarsm Checker</a>
        </div>
        <nav class="text-white font-normal">
          <ul class="flex">
            <li class="mx-1 lg:mx-3 p-2 hover:bg-slate-
700 hover:rounded-xl">Beranda</li>
            <li class="mx-1 lg:mx-3 p-2 hover:bg-slate-
700 hover:rounded-xl ">Tentang Saya</li>
            <li class="mx-1 lg:mx-3 py-2 px-4 bg-slate-
800 text-white rounded-xl hover:bg-slate-700">Masuk
            </li>
          </ul>
        </nav>
      </div>
    </div>
  </header>
  <!-- Nav Section -->

  <!-- Description Section -->
  <section class="mt-10 text-white min-h-8 lg:mt-0">
    <div class="container">
      <div class="flex flex-wrap items-center">
        <div class="p-5 w-full h-40 lg:w-1/2 lg:h-full">
          <h1 class="font-bold text-2xl my-
auto">Plagiarism Checker</h1>
          <p class="text-base">Website Ini Menggunakan
Metode Jaccard Coefficient Untuk Mempersentasekan
Kemiripian Dokumen Anda

```

```

        dengan Dokumen yang Terdapat di
Database</p>
    </div>
    <div class="w-full my-4 lg:w-1/2 p-3">
        <div class="p-4">
            <form action="">
                <div class="w-full px-4">
                    <textarea type="text" id="text"
                        class="w-full bg-slate-800
text-white p-3 rounded-md h-48 focus:border-2 focus:border-slate-
700 focus:outline-none resize-none"
                        placeholder="Input Your
Text"></textarea>
                </div>
                <div class="w-full my-5 px-4 flex
items-center justify-between">
                    <label for="upload">or Upload Your
File (*.pdf)</label>
                    <input
                        class="file:bg-slate-800
file:border-0 file:text-white file:rounded-md file:p-2
file:hover:bg-slate-700"
                        type="file" name="upload">
                    </div>
                    <button type="submit"
                        class="w-full bg-slate-800 py-2
rounded-lg text-white hover:bg-slate-700"> <i
                        class="fa-solid fa-magnifying-
glass mr-2"></i> Check
                        Plagiarism</button>
                </form>
            </div>
        </div>
    </div>
</div>
</section>
<!-- Description Section -->

<!-- Result -->

<section class="pt-20 text-white">
    <div class="container">
        <div class="w-full px-4 text-center">
            <h1 class="font-semibold text-2xl mb-
2">Result</h1>
        </div>
        <div class="flex flex-wrap justify-between">
            <div class="w-full lg:w-[49%] bg-slate-800 rounded-
md p-4">

```

```

<h2 class="font-semibold text-xl mb-3">Details</h2>
<div class="flex">
  <p class="text-base w-1/2">Character :
2000</p>
  <p class="text-base w-1/2">Words : 50</p>
</div>
</div>
<div class="w-full mt-5 lg:w-[49%] lg:mt-0 bg-slate-800 rounded-md p-4">
  <h2 class="font-semibold text-xl mb-3">Plagirism Percentage</h2>
  <div class="flex">
    <p class="text-base w-1/2">Unique : 40%</p>
    <p class="text-base w-1/2">Plagiarism :
60%</p>
  </div>
  <div class="w-full mt-3 flex bg-slate-900 rounded-full">
    <div
      class="bg-slate-700 text-s font-medium text-blue-100 text-center p-0.5 leading-none rounded-l-full w-[60%]">
      60%
    </div>
    <div class="text-s font-medium text-blue-100 text-center p-0.5 leading-none w-[40%]">
      40%
    </div>
  </div>
</div>
<div class="w-full bg-slate-800 mt-5 rounded-md p-4">
  <h2 class="font-semibold text-xl mb-3">File With High Similarity</h2>
  <table class="min-w-full text-white table-auto">
    <thead class="border-b border-slate-500 font-semibold">
      <tr>
        <th scope="col" class="text-sm font-medium px-6 py-4 text-left">#
        </th>
        <th scope="col" class="text-sm font-medium px-6 py-4 text-left">Title Pdf
        </th>
        <th scope="col" class="text-sm font-medium px-6 py-4 text-left">Similarity
      </tr>
    </thead>
  </table>

```

```

        </th>
        <th scope="col" class="text-sm
font-medium px-6 py-4 text-left">
            Show
        </th>
    </tr>
</thead>
<tbody>
    <tr class="border-b border-slate-500">
        <td class="px-6 py-4 whitespace-
nowrap text-sm font-medium">1</td>
        <td class="text-sm font-light px-6
py-4">
            Analisa Hasil USaha Dalam
Rangka Menyambut Hari H
        </td>
        <td class="text-sm font-light px-6
py-4 whitespace-nowrap">
            50 %
        </td>
        <td class="text-sm font-light px-6
py-4 whitespace-nowrap">
            <a href="" class="w-full bg-
slate-700 p-3 rounded-lg text-white">
                Show
            </a>
        </td>
    </tr>
    <tr class="border-b border-slate-500">
        <td class="px-6 py-4 whitespace-
nowrap text-sm font-medium">2</td>
        <td class="text-sm font-light px-6
py-4">
            Analisa Hasil USaha Dalam
Rangka Menyambut Hari H
        </td>
        <td class="text-sm font-light px-6
py-4 whitespace-nowrap">
            50 %
        </td>
        <td class="text-sm font-light px-6
py-4 whitespace-nowrap">
            <a href="" class="w-full bg-
slate-700 p-3 rounded-lg text-white">
                Show
            </a>
        </td>
    </tr>
</tbody>
</table>
</div>

```

```

        </div>
    </div>
</section>

<!-- Result -->

<!-- User and File View -->

<section>
    <div class="container">
        <div class="flex flex-wrap justify-between">
            <div class="w-full bg-slate-800 lg:w-[49%] mt-5
rounded-md p-4">
                <h2 class="font-semibold text-xl mb-3 text-
white">All User</h2>
                <table class="min-w-full text-white table-
auto">
                    <thead class="border-b border-slate-500
font-semibold">
                        <tr>
                            <th scope="col" class="text-sm
font-medium px-6 py-4 text-left">
                                #
                            </th>
                            <th scope="col" class="text-sm
font-medium px-6 py-4 text-left">
                                Username
                            </th>
                            <th scope="col" class="text-sm
font-medium px-6 py-4 text-left">
                                Id (NIM)
                            </th>
                            <th scope="col" class="text-sm
font-medium px-6 py-4 text-left">
                                Action
                            </th>
                        </tr>
                    </thead>
                    <tbody>
                        <tr class="border-b border-slate-500">
                            <td class="px-6 py-4 whitespace-
nowrap text-sm font-medium">1</td>
                            <td class="text-sm font-light px-6
py-4">
                                Aris Akhyar Abdillah
                            </td>
                            <td class="text-sm font-light px-6
py-4 whitespace-nowrap">
                                H071171505
                            </td>

```



```

py-4 whitespace-nowrap">
        <td class="text-sm font-light px-6
        <a href=""
            class="w-full mx-1 bg-
slate-700 p-3 rounded-lg text-white hover:bg-yellow-500">
            <i class="fa-solid fa-
pencil"></i>
        </a>
        <a href=""
            class="w-full mx-1 bg-
slate-700 p-3 rounded-lg text-white hover:bg-red-500">
            <i class="fa-solid fa-
trash"></i>
        </a>
    </td>
</tr>
</tbody>
</table>
</div>

<div class="w-full bg-slate-800 lg:w-[49%] mt-5
rounded-md p-4">
    <h2 class="font-semibold text-xl mb-3 text-
white">All Files</h2>
    <table class="min-w-full text-white table-
auto">
        <thead class="border-b border-slate-500
font-semibold">
            <tr>
                <th scope="col" class="text-sm
font-medium px-6 py-4 text-left">
                    #
                </th>
                <th scope="col" class="text-sm
font-medium px-6 py-4 text-left">
                    Title
                </th>
                <th scope="col" class="text-sm
font-medium px-6 py-4 text-left">
                    Uploader
                </th>
                <th scope="col" class="text-sm
font-medium px-6 py-4 text-left">
                    Action
                </th>
            </tr>
        </thead>
        <tbody>
            <tr class="border-b border-slate-500">
                <td class="px-6 py-4 whitespace-
nowrap text-sm font-medium">1</td>

```

```

        <td class="text-sm font-light px-6
py-4">
            Analisa Hasil Usaha Dalam
Rangka Menyambut Hari H
        </td>
        <td class="text-sm font-light px-6
py-4 whitespace-nowrap">
            H071171505
        </td>
        <td class="text-sm font-light px-6
py-4 whitespace-nowrap">
            <a href=""
                class="w-full mx-1 bg-
slate-700 p-3 rounded-lg text-white hover:bg-green-500">
                <i class="fa-solid fa-
eye"></i>
            </a>
            <a href=""
                class="w-full mx-1 bg-
slate-700 p-3 rounded-lg text-white hover:bg-red-500">
                <i class="fa-solid fa-
trash"></i>
            </a>
        </td>
    </tr>
</tbody>
</table>
</div>
</div>
</div>
</div>
</section>

<!-- User and File View -->

<!-- Login -->
<section>
    <div class="container">
        <div class="w-full my-4 lg:w-1/2 lg:mx-auto p-3 text-
white bg-slate-800 rounded-md">
            <h2 class="font-semibold text-2xl mb-2 text-
center">Login To Application</h2>
            <div class="p-4">
                <form action="">
                    <div class="w-full px-4 mb-8">
                        <label for="id" class="text-base font-
thin text-white">Id (NIM)</label>
                        <input type="text" id="id"
                            class="w-full bg-slate-700 text-
slate-200 p-3 rounded-md focus:outline-none focus:ring-slate-400
focus:ring-1" />
                    </div>
                </form>
            </div>
        </div>
    </div>
</section>

```

```

        <div class="w-full px-4 mb-8">
            <label for="password" class="text-base
font-thin text-white">Password</label>
            <input type="password" id="password"
class=" w-full bg-slate-700 text-slate-200 p-3 rounded-md
focus:outline-none
            focus:ring-slate-400 focus:ring-1" />
        </div>
        <button type="submit"
            class="w-full bg-slate-700 py-2
rounded-lg text-white hover:bg-slate-600">Login</button>
        </form>
    </div>
</div>
</div>
</section>
<!-- Login -->

<!-- Footer -->
<footer class="bg-dark pt-10 pb-5">
    <div class="container">
        <div class="w-full pt-10 border-t border-slate-700">
            <div class="flex items-center justify-center mb-5">
                <!-- Whatsapp -->
                <a
                    href="https://wa.me/+6282195008936"
target="_blank"
                    class="w-9 h-9 mr-2 flex justify-center
items-center">
                    <svg role="img" width="25" class="fill-
white hover:fill-slate-700" viewBox="0 0 24 24"
                    xmlns="http://www.w3.org/2000/svg">
                        <title>WhatsApp</title>
                        <path
                            d="M17.472 14.382c-.297-.149-
1.758-.867-2.03-.967-.273-.099-.471-.148-.67.15-.197.297-.767.966-
.94 1.164-.173.199-.347.223-.644.075-.297-.15-1.255-.463-2.39-
1.475-.883-.788-1.48-1.761-1.653-2.059-.173-.297-.018-.458.13-
.606.134-.133.298-.347.446-.52.149-.174.198-.298.298-.497.099-
.198.05-.371-.025-.52-.075-.149-.669-1.612-.916-2.207-.242-.579-
.487-.5-.669-.51-.173-.008-.371-.01-.57-.01-.198 0-.52.074-
.792.372-.272.297-1.04 1.016-1.04 2.479 0 1.462 1.065 2.875 1.213
3.074.149.198 2.096 3.2 5.077 4.487.709.306 1.262.489
1.694.625.712.227 1.36.195 1.871.118.571-.085 1.758-.719 2.006-
1.413.248-.694.248-1.289.173-1.413-.074-.124-.272-.198-.57-.347m-
5.421 7.403h-.004a9.87 9.87 0 01-5.031-1.378l-.361-.214-
3.741.982.998-3.648-.235-.374a9.86 9.86 0 01-1.51-5.26c.001-5.45
4.436-9.884 9.888-9.884 2.64 0 5.122 1.03 6.988 2.898a9.825 9.825
0 012.893 6.994c-.003 5.45-4.437 9.884-9.885 9.884m8.413-
18.297A11.815 11.815 0 0012.05 0C5.495 0 .16 5.335.157 11.892c0
2.096.547 4.142 1.588 5.945L.057 24l6.305-1.654a11.882 11.882 0

```

```

005.683 1.448h.005c6.554 0 11.89-5.335 11.893-11.893a11.821 11.821
0 00-3.48-8.413Z" />
    </svg>
  </a>
  <a
href="https://www.facebook.com/profile.php?id=100000053025926"
target="_blank"
    class="w-9 h-9 mr-3 flex justify-center
items-center">
    <svg role="img" width="25" viewBox="0 0 24
24" class="fill-white hover:fill-slate-700"
    xmlns="http://www.w3.org/2000/svg">
    <title>Facebook</title>
    <path
    d="M24 12.073c0-6.627-5.373-12-12-12-
12s-12 5.373-12 12c0 5.99 4.388 10.954 10.125 11.854v-8.385H7.078v-
3.47h3.047V9.43c0-3.007 1.792-4.669 4.533-4.669 1.312 0 2.686.235
2.686.235v2.953H15.83c-1.491 0-1.956.925-1.956 1.874v2.25h3.328l-
.532 3.47h-2.796v8.385C19.612 23.027 24 18.062 24 12.073z" />
    </svg>
  </a>
  <a href="https://github.com/arxs-abd/arxs-abd"
target="_blank"
    class="w-9 h-9 mr-3 flex justify-center
items-center">
    <svg role="img" width="25" class="fill-
white hover:fill-slate-700" viewBox="0 0 24 24"
    xmlns="http://www.w3.org/2000/svg">
    <title>Git Hub</title>
    <path
    d="M12 .297c-6.63 0-12 5.373-12 12
0 5.303 3.438 9.8 8.205 11.385.6113.82-.258.82-.577 0-.285-.01-
1.04-.015-2.04-3.338.724-4.042-1.61-4.042-1.61C4.422 18.07 3.633
17.7 3.633 17.7c-1.087-.744.084-.729.084-.729 1.205.084 1.838 1.236
1.838 1.236 1.07 1.835 2.809 1.305 3.495.998.108-.776.417-1.305.76-
1.605-2.665-.3-5.466-1.332-5.466-5.93 0-1.31.465-2.38 1.235-3.22-
.135-.303-.54-1.523.105-3.176 0 0 1.005-.322 3.3 1.23.96-.267 1.98-
.399 3-.405 1.02.006 2.04.138 3 .405 2.28-1.552 3.285-1.23 3.285-
1.23.645 1.653.24 2.873.12 3.176.765.84 1.23 1.91 1.23 3.22 0 4.61-
2.805 5.625-5.475 5.92.42.36.81 1.096.81 2.22 0 1.606-.015 2.896-
.015 3.286 0 .315.21.69.825.57C20.565 22.092 24 17.592 24 12.297c0-
6.627-5.373-12-12-12" />
    </svg>
  </a>
</div>
<p class="font-medium text-xs text-slate-500 text-
center">
    © 2022 Copyrigh : <span class="font-
semibold">Aris Akhyar Abdillah</span>.
  </p>
</div>

```

```

    </div>
  </footer>

```

```

    <!-- Footer -->
  </body>

```

```

</html>

```

```
views/login-page.ejs
```

```
<%- include('partials/login') %>
```

```
views/main.ejs
```

```

<!DOCTYPE html>
<html lang="en">

```

```

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>PlagCheck</title>
  <link rel="stylesheet" href="css/output.css">
  <script src="https://kit.fontawesome.com/524dd30d0c.js"
crossorigin="anonymous"></script>
</head>

```

```
<body class="bg-sky-200">
```

```
  <%- include('partials/navbar') %>
```

```
  <%- body %>
```

```
  <%- include('partials/footer') %>
```

```

  <script src="js/tw/index.js"></script>
</body>

```

```
</html>
```

```
.env
```

```

MONGODB_URL=mongodb://127.0.0.1:27017/latihan
SECRET_KEY=1n15kr1ps1

```

```
index.js
```

```

require('dotenv').config()
const express = require('express')

```

```

const session = require('express-session')
const ejsLayout = require('express-ejs-layouts')
const bodyParser = require('body-parser')
const cookieParser = require('cookie-parser')
const flash = require('connect-flash')
const methodeOverride = require('method-override')

// Router
const activityRouter = require('./routes/activity')

const app = express()

app.set('view engine', 'ejs')

app.use(ejsLayout)
app.use(bodyParser.json())
app.use(bodyParser.urlencoded({
  extended : true
}))
app.use(express.static(__dirname + '/public'))
app.use(cookieParser('1n15KR1p51'))
app.use(session({
  resave: true,
  saveUninitialized: true,
  secret: '1n15KR1p51',
}))
app.use(methodeOverride('_method'))
app.use(flash())

app.use(activityRouter)

// listen on port
app.listen(3000, () => console.log('Server Running at
http://localhost:3000'))

```

package.json

```

{
  "name": "skripsi",
  "version": "1.0.0",
  "description": "Tugas Akhirku",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "node index.js",
    "dev": "nodemon -e js",
    "tailwind": "npx tailwindcss -i ./public/css/input.css -o
./public/css/output.css --watch"
  },

```

```

"keywords": [
  "skirpsi"
],
"author": "Aris Akhyar Abdillah",
"license": "ISC",
"dependencies": {
  "bcryptjs": "^2.4.3",
  "body-parser": "^1.20.0",
  "connect-flash": "^0.1.1",
  "cookie-parser": "^1.4.6",
  "dotenv": "^16.0.1",
  "ejs": "^3.1.8",
  "express": "^4.17.3",
  "express-ejs-layouts": "^2.5.1",
  "express-fileupload": "^1.3.1",
  "express-session": "^1.17.2",
  "express-validator": "^6.14.0",
  "flash": "^1.1.0",
  "jsonwebtoken": "^8.5.1",
  "method-override": "^3.0.0",
  "mongodb": "^4.4.1",
  "mongoose": "^6.2.10",
  "mongoose-paginate-v2": "^1.6.3",
  "multer": "^1.4.4",
  "nodemon": "^2.0.19",
  "pdf-parse": "^1.1.1",
  "sastrawijis": "^1.0.4",
  "tailwindcss": "^3.0.23",
  "tw-elements": "^1.0.0-alpha12"
}
}

```

Lampiran 2 (Detail Data Jurnal yang Digunakan)

NO	JUDUL JURNAL	PENULIS JURNAL	TAHUN
1	Aplikasi Budaya Toba Samosir Berbasis Android	Harni Kusniyati, Nicky Saputra Pangondian Sitanggang	2016
2	Aplikasi Mobile <i>Driver</i> Online Berbasis Android Untuk Perusahaan Rental Kendaraan	Surawijaya Surahman, Eko Budi Setiawan	2017
3	Aplikasi Penanda Lokasi Peta Digital Berbasis Mobile GIS pada <i>Smartphone</i> Android	Gunti Mustika Hati, Andri Suprayogi, Bandi Sasmito	2013
4	Media Pembelajaran Berbasis Android Pada Mata Pelajaran Sistem Operasi Jaringan Kelas XI	Joko Kuswanto, Ferri Radiansah	2018
5	Pembuatan Aplikasi Android <i>Driver Control</i> Sebagai Sarana Memonitor Anak Berkendara Secara Waktu-Nyata	Lia Lidya Roza, R. Rizal Isnanto, Eko Didik Widiyanto	2016
6	Pencarian Rumah Makan Berbasis Android	Faris Sifauttijani, Tri Listyorini, Rizkysari Meimaharani	2017
7	Pengembangan Bahan Ajar Interaktif Berbasis Android Pada Materi Jurnal Penyesuaian dan Jurnal Koreksi Untuk Kelas XII Akuntansi di SMKN 1 Surabaya	Arina Manasikana, Agung Listiadi	2017
8	Pengembangan <i>Game</i> Edukatif Berbasis Android Sebagai Media Pembelajaran Akuntansi Pada Materi Jurnal Penyesuaian Perusahaan Jasa	Ditto Rahmawan Putra, Mahendra Adhi Nugroho	2016
9	Perancangan Aplikasi Mobile <i>Learning</i> Berbasis Android di SMK Negeri 6 Padang	Rahmat Fadillah, Legiman Slamet	2019
10	Rancang Bangun Aplikasi Mobile Siska Berbasis Android	Ni Kadek Ceryna Dewi, Ida Bagus Gede Anandita, Ketut Jaya Atmaja, Putu Wirayudi Aditama	2018
11	Aplikasi <i>Internet Of Things</i> (IOT) Untuk Pemantauan Dan Pengendalian Beban Listrik Di Ruangan	Erwan Eko Prasetyo	2017
12	Implementasi <i>Internet Of Things</i> Dalam Kehidupan Sehari-Hari	Ferdy Susanto, Ni Komang Prasiani, Putu Darmawan	2022

13	<i>Internet Of Things (IOT) Sistem Pengendalian Lampu Menggunakan Raspberry PI Berbasis Mobile</i>	Yoyon Efendi	2018
14	Pengembangan Instrumen Pengukuran Aliran Air Berbasis <i>Internet of Things (IoT)</i>	M. Irfan Wahyuni, Hollanda Arief Kusuma, Sapta Nugraha	2021
15	Pengendali Pintu Gerbang Berbasis IOT (<i>Internet of Thinigs Gate Controller Berbasis IOT (Internet of Things)</i>)	Putri Dwi Lestari, Lita Karlitasari, Sufiatul Maryana	2021
16	Perancangan Sistem Kamar Kos Pintar Berbasis IoT	Ricky Novian Riyadi, Esti Wijayanti, Alif Catur Murti	2020
17	Potensi Implementasi <i>Internet Of Things (IOT)</i> Untuk Perpustakaan	Teguh Prasetyo Utomo	2019
18	Potensi <i>Internet of Things (IoT)</i> dan Ragam Sensor untuk Layanan Kesehatan	Gelmy Fitriawan, Dikpride Despa, Ika Kustiani	2019
19	Sistem Kontrol dan <i>Monitoring Smart House</i> Berbasis IoT Dengan <i>Smartphone</i> Android	Gharisa Nur Alamsyah, Slamet Winardi	2022
20	Teknologi Kontrol Lampu dan Kunci Rumah Berbasis Internet of Thinigs (IoT)	Ahmad Fitriansyah, Mochammad Riyan Suryanto	2021
21	<i>Black Box Testing Boundary Value Analysis</i> Pada Aplikasi <i>Submission System</i>	Luthfie Auditya Amarul Ma'ruf, Condro Kartiko, Citra Wiguna	2020
22	Metode <i>Rapid Application Development (RAD)</i> pada Perancangan <i>Website Inventory</i> PT. Sarana Abadi Makmur Bersama (S.A.M.B) Jakarta	Oky Irnawati, Galih Bayu Aji Listianto	2018
23	Model <i>Prototyping</i> Pada Pengembangan Sistem Informasi	Dwi Purnomo	2017
24	Penerapan Metode <i>Prototype</i> Pada Perancangan Sistem Informasi Penggajian Karyawan (Persis Gawan) Berbasis Web	Eka Wulansari Fridayanthie, Haryanto, Tsamarah Tsabitah	2021
25	Penerapan Metode <i>Prototype</i> Dalam Perancangan Apikasi SIPINJAM Berbasis <i>Website</i> pada Credit Union Canaga Antutn	Petrus Yoko, Rabiatal Adwiya, Wahyu Nugraha	2019
26	Penerapan Metode <i>Rapid Application Development (RAD)</i> Dalam Pengembangan Sistem Informasi Data Toko	Yahya Dwi Wijaya	2020

27	Penerapan Metode <i>Waterfall</i> Pada Sistem Informasi Manajemen Buku Perpustakaan Berbasis Web	Yuni Eka Achyani, Sela Saumi	2019
28	Pengujian <i>Black Box</i> pada Aplikasi Sistem Informasi Akademik Menggunakan Teknik <i>Equivalence Partitioning</i>	Dwi Bahar Muslimin, Dwi Kusmanto, Karina Femi Amilia, Muhammad Syamsul Ariffin, Siti Mardiana, Yulianti	2020
29	Pengujian <i>Black Box</i> pada Aplikasi Sistem Seleksi Sales Terbaik Menggunakan Teknik <i>Equivalence Partitions</i>	Fadhila Cahya Ningrum, Dandi Suherman, Sita Aryanti, Handika Angga Prasetya, Aries Saifudin	2019
30	Pengujian <i>Black Box</i> Testing pada Aplikasi <i>Action & Strategy</i> Berbasis Android Dengan Teknologi <i>Phonegap</i>	Wahyu Nur Cholifah, Yulianingsih, Sri Melati Sagita	2018
31	Pelatihan melalui Web Seminar terkait Publikasi Artikel untuk Menembus Jurnal Sinta 2 dan Scopus	Amrie Firmansyah, Resi Ariyasa Qadri, Amardianto Arham	2020
32	Pemanfaatan Sistem Informasi Perpustakaan Digital Berbasis <i>Website</i> Untuk Para Penulis	Agus Prayitno, Yulia Safitri	2015
33	Penerapan <i>Business Process Modeling Notation</i> (BPMN) Untuk Memodelkan Kebutuhan Sistem Proses Penyuntingan Tulisan Pada <i>Website</i> Jurnal Jtriste	Gilbert Krisantoso, Irfan AP, Mohammad Fajar	2015
34	Pengembangan <i>Website</i> Unit Penelitian dan Pengabdian Kepada Masyarakat dan Penerapan Jurnal Elektronik Berbasis <i>Open Source</i> di Politeknik Negeri Kupang	Imanuel Christian Mauko, Nicodemus Mardanus Setiohardjo, Fredrik Paulus Noach	2017
35	Pengukuran Kualitas Layanan Website Kementerian Kominfo Dengan Menggunakan Metode <i>Webqual 4.0</i>	Iman Sanjaya	2012
36	Perancangan <i>Website</i> Sebagai Media Promosi dan Informasi	Penda Sudarto Hasugian	2018
37	Portal Publikasi Jurnal Karya Ilmiah Program Studi Sistem Informasi FTK Universitas Harapan Berbasis Web dengan	Marina Elsera	2018

	Metode Fast (Framework For The Application of System Thinking)		
38	Rancang Bangun Sistem Informasi Desa Berbasis Website (Studi Kasus Desa Netpala)	Jimi Asmara	2019
39	Rancang Bangun Sistem Informasi Evaluasi Siswa dan Kehadiran Guru Berbasis Web (Studi Kasus di SMK Nusa Putra Kota Tangerang)	Ahmad Yani, Beni Saputra	2018
40	Rancang Bangun Website untuk Memeriksa Plagiat E-Journal Fakultas Kedokteran Universitas Palangka Raya	Widiatry, Nova Noor Kamala Sari	2019