SISTEM PENCARIAN TREND JUDUL TUGAS AKHIR MAHASISWA TEKNIK INFORMATIKA UNISSULA MENGGUNAKAN METODE KEYWORD EXTRACTION

LAPORAN TUGAS AKHIR

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

DISUSUN OLEH:

CHOLID FAJAR SUPARDI
NIM 32601900009

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM SULTAN AGUNG
SEMARANG
2023

FINAL PROJECT

TREND SEARCH SYSTEM FINAL PROJECT TITLE OF UNISSULA INFORMATICS ENGINEERING STUDENTS USING KEYWORD EXTRACTION

Proposed to complete the requirement to obtain a bachelor's degree (S1)
at Informatics Engineering Departement of Industrial Technology Faculty
Sultan Agung Islamic University

ARRANGED BY:

CHOLID FAJAR SUPARDI

NIM 326019000009

MAJORING OF INFORMATICS ENGINEERING
INDUSTRIAL TECHNOLOGY FACULTY
SULTAN AGUNG ISLAMIC UNIVERSITY
SEMARANG
2023

LEMBAR PENGESAHAN PEMBIMBING

Laporan Tugas Akhir dengan judul **"SISTEM PENCARIAN TREND JUDUL TUGAS AKHIR MAHASISWA TEKNIK INFORMATIKA UNISSU MENGGUNAKAN METODE KEYWORD EXTRACTION"**
ini disusun oleh :

Nama : Cholid Fajar Supardi
NIM 32601900009
Program Studi : Teknik Informatika
Telah disahkan oleh dosen pembimbing pada :
Hari : Kamis
Tanggal : 10-08-2023

Mengesahkan,

Pembimbing I	Pembimbing II
Sam Farisa Chaerul H, S.T., M.Kom	Badrul ah, ST., M.Kom
NIDN. 0628028602	NIDN. 0619018701

Mengetahui,

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

Ir. Sri Mulyono, M.Eng
NIDN. 0626066601

iii

LEMBAR PENGESAHAN PENGUJI

Laporan tugas akhir dengan judul "**SISTEM PENCARIAN TREND JUDUL TUGAS AKHIR MAHASISWA TEKNIK INFORMATIKA UNISSULA MENGGUNAKAN METODE KEYWORD EXTRACTION**"
ini telah dipertahankan di depan dosen penguji Tugas Akhir pada:

Hari : Kamis

Tanggal : 10-08-2023

TIM PENGUJI

Ketua Penguji	Anggota II
Andi Riansyah. ST., M.Kom	Bagus Satrio WP S.Kom.M.Cs
NIDN.0609108802	NIDN. 1027118301

SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan dibawah ini :

Nama	: Cholid Fajar Supardi
NIM	: 32601900009
Judul Tugas Akhir	: SISTEM PENCARIAN TREND JUDUL TUGAS AKHIR MAHASISWA TEKNIK INFORMATIKA UNISSULA MENGGUNAKAN KEYWORD EXTRACTION

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

Semarang, 10 Agustus 2023

Yang Menyatakan,

Cholid Fajar Supardi

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH

Saya yang bertanda tangan dibawah ini :

Nama	: Cholid Fajar Supardi
NIM	: 32601900009
Program Studi	: Teknik Informatika
Fakultas	: Teknologi Industri
Alamat Asal	: Jl.SA.MAULANA GG.BAMBU No.76 Kab.Berau Kalimantan Timur

Dengan ini menyatakan Karya Ilmiah berupa Tugas akhir dengan Judul : **SISTEM PENCARIAN TREND JUDUL TUGAS AKHIR MAHASISWA TEKNIK INFORMATIKA UNISSULA MENGGUNAKAN METODE KEYWORD EXTRACTION**

Menyetujui menjadi hak milik Universitas Islam Sultan Agung serta memberikan Hakbebas Royalti Non-Eksklusif untuk disimpan, dialihmediakan, dikelola dan pangkalandata dan dipublikasikan diinternet dan media lain untuk kepentingan akademis selamatetap menyantumkan nama penulis sebagai pemilik hak cipta. Pernyataan ini saya buat dengan sungguh-sungguh. Apabila dikemudian hari terbukti ada pelanggaran Hak Cipta/Plagiarisme dalam karya ilmiah ini, maka segala bentuk tuntutan hukum yang timbul akan saya tanggung secara pribadi tanpa melibatkan Universitas Islam Sultan agung.

Semarang, 10 Agustus 2023

Yang menyatakan,

Cholid Fajar Supardi

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

KATA PENGANTAR

Dengan mengucap syukur alhamdulillah atas kehadirat Allah SWT yang telah memberikan rahmat dan karunianya kepada penulis, sehingga dapat menyelesaikan Tugas Akhir dengan judul "Sistem Pencarian Trend Judul Tugas Akhir Mahasiswa Teknik Informatika Unissula Menggunakan Metode *Keyword Extraction*" ini untuk memenuhi salah satu syarat menyelesaikan studi serta dalam rangka memperoleh gelar sarjana (S-1) pada Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang. Tugas Akhir ini disusun dan dibuat dengan adanya bantuan dari berbagai pihak, materi maupun teknis, oleh karena itu saya selaku penulis mengucapkan terima kasih kepada:

1. Rektor UNISSULA Bapak Prof. Dr. H. Gunarto, S.H., M.H yang mengizinkan penulis menimba ilmu di kampus ini.
2. Dekan Fakultas Teknologi Industri Ibu Dr. Novi Marlyana, S.T., M.T.
3. Dosen pembimbing I penulis Bapak Sam Farisa Chaerul Haviana, S.T., M.Kom yang telah meluangkan waktu dan memberi ilmu.
4. Dosen pembimbing II penulis Bapak Ir. Sri Mulyono, M.Eng yang memberikan banyak nasehat dan saran.
5. Orang tua penulis yang senantiasa memberikan semangat serta doa agar tugas akhir ini berjalan dengan lancar,
6. Dan kepada semua pihak yang tidak dapat saya sebutkan satu persatu.

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

Semarang, 10-08-2023

Cholid Fajar Supardi

ABSTRAK

Skripsi adalah sebuah karya ilmiah yang dibuat berdasarkan pengetahuan khusus dan fakta yang jelas. Seorang Mahasiswa diwajibkan menulis skripsi sebagai syarat kelulusan, dimana tujuannya adalah menyelesaikan suatu masalah dengan menerapkan metode tertentu.Oleh karena itu, dibutuhkan sebuah sistem yang dapat menampilkan trend metode skripsi sehingga dapat dilihat metode apa saja yang paling banyak digunakan mahasiswa Teknik Informatika Unissula dalam menulis skripsi. Metode *Keyword Extraction* digunakan untuk mengekstrak *keyword* yang ada pada abstrak skripsi, TF-IDF mengekstrak abstrak tersebut berdasarkan frekuensi kemunculan kata *term*, semakin tingg frekuensi kata maka semakin besar kemungkinan *keyword* tersebut muncul. Kemudian hasil ekstraksi *keyword* dilakukan validasi untuk pencocokan antara *keyword* hasil ekstraksi dengan *keyword* asli dari dokuemen skripsi menggunakan algoritma *Cosine Similarity*. Alhasil didapat hasil *similarity* tertinggi 0, 437248.
Kata Kunci : Skripsi, *Keyword Extraction*, TF-IDF, *Cosine Similarity*

ABSTRACT

Thesis is a scientific work created based on specific knowledge and clear facts. It is prepared as a requirement for a student's graduation, designed to solve a particular problem using a specific method.. Therefore, a system is needed to display the trends of thesis methods, allowing us to see which methods are most commonly used by Computer Science students at Unissula in writing their theses.The Keyword Extraction method is used to extract keywords from thesis abstracts. TF-IDF is then used to extract abstracts based on the term frequency of words. The higher the word frequency, the greater the likelihood of it being a keyword. Subsequently, the extracted keywords are validated by matching them with the original keywords from the thesis documents using the Cosine Similarity algorithm. As a result, the highest similarity obtained was 0.437248.
Keyword: Thesis, Keyword Extraction, TF-IDF, *Cosine Similarity.*

BAB I
PENDAHULUAN

1.1 Latar Belakang

Teknologi yang terus berkembang dengan pesat serta hampir mempengaruhi segala aspek kehidupan manusia. Di era revolusi 4.0, muncul berbagai teknologi yang menggunakan sistem pembelajaran mesin (*machine learning)* untuk membantu manusia dalam melakukan pencarian sesuatu secara cepat dan tepat. Dalam hal ini teknologi di zaman sekarang dijadikan sarana pencarian sumber rujukan dalam penulisan ,pembuatan jurnal dan karya tulis ilmiah. Pada penelitian ini, sistem pencarian trend judul skripsi akan dibuat dengan tujuan untuk menampilkan metode yang paling banyak diangkat oleh mahasiswa Teknik Informatika Unissula dalam pembuatan skripsi.

Pada penelitian ini, fokus utama adalah mengambil studi kasus dengan mengambil data Skripsi dari Situs Repository Unissula dalam rangka penyusunan Proposal Tugas Akhir. Skripsi adalah hasil dari penelitian sarjana S1 yang mengangkat metode permasalahan dalam bidang tertentu dan mengikuti aturan-aturan penulisan tertentu.

Penelitian yang dilakukan untuk menciptakan sebuah aplikasi pencarian karya ilmiah.Penelitian tersebut melibatkan setiap karya ilmiah yang di temukan serta menggunakan algoritma TF-IDF. Hasil penelitian tersebut menghasilkan aplikasi pencarian karya ilmiah yang dapat menghasilkan bobot setiap karya ilmiah dan dapat menampilkan karya ilmiah sesuai kata kunci yang dicari serta memverifikasi dokumen yang mengandung kata kunci tersebut.(Azis Maarif 2015)

Penelitian yang dilakukan untuk meringkas dokumen berbahasa indonesia. Penelitian tersebut menggunakan dokumen non-fiksi dan fiksi serta menggunakanalgoritma TF-IDF. Hasil dari penelitian tersebut menunjukan durasi rata-rata selama 68,25 detik pada seluruh dokumen uji menghasilkan seleksi fitur bobot kata cukup baik, dan lebih cocok digunakan pada dokumen non-fiksi.(Widyasanti, Darma Putra, and Dwi Rusjayanthi 2018)

Pada penelitian ini, dapat disimpulkan bahwa penggunaan metode TF-IDF dalam penentuan bobot kata atau meringkas suatu kata cukup baik, mengingat telah dilakukan penelitian-penelitian sebelumnya yang menghasilkan hasil yang memadai menggunakan metode tersebut. Oleh karena itu, pada penelitian ini bertujuan untuk membuat sistem yang mampu menampilkan sebuah tampilan informasi trend judul skripsi yang memperlihatkan tema skripsi apa yang sering diangkat oleh mahasiswa Teknik Informatika Unissula dalam menulis skripsi.

Hal ini dapat membantu mahasiswa dalam penulisan skripsi yang wajib dikerjakan sebagai syarat lulus jenjang S1. Oleh karena itu, pada penelitian ini bertujuan untuk membuat sitem pencarian trend judul skripsi mahasiswa Teknik Informatika Universitas Islam Sultan Agung yang dapat menampilkan tema skripsi apa yang banyak di buat oleh alumni mahasiswa Teknik Informatika Unissula sebelumnya.

1.2 Perumusan Masalah

Belum adanya sistem trend judul skripsi yang dapat menampilkan trend skripsi atau banyaknya metode yang diangkat oleh mahasiswa Teknik Informatika Unissula.

1.3 Pembatasan Masalah

Adapun batasan masalah dari penulisan proposal adalah sebagai berikut:

1. Dataset yang digunakan merupakan hasil *scrapping* yang dilakukan dengan cara *scrapping* website menggunakan *extensions web scrapper* pada website Repository Unissula yang digunakan sebagai bahan penelitian, dengan link berikut,http://repository.unissula.ac.id/view/divisions/jur=5Finformatika dimana terdapat kumpulan skripsi mahasiswa Teknik informatika dari tahun 2018 – 2022.

2. Metode yang digunakan dalam system ini adalah *Keyword Extraction* TF-IDF.

1.4 Tujuan

Tujuan tugas akhir adalah perancangan sistem yang dapat menampilkan seberapa banyak metode yang diangkat oleh mahasiswa Teknik Informatika Unissula dalam pembuatan skripsi.

1.5 Manfaat

Adapun manfaat dari pembuatan sistem ini adalah diharapkan memudahkan dalam mencari metode apa saja yang yang paling banyak di angkat oleh mahasiswa dalam menulis skripsi, khususnya mahasiswa Prodi Teknik Informatika Unissula.

1.6 Sistematika Penulisan

Adapun sistematika penulisan yang akan dipakai penulis dalam pembuatan laporan tugas akhir ini adalah seperti pada tabel 1.1:

Tabel 1. 1 Sistematika Penulisan

BAB I	:	PENDAHULUAN
		Pada bab ini penulis mengutarakan latar belakang pemilihan judul, rumusan masalah, batasan masalah, tujuan penelitian, metodologi penelitian, serta sistematika penulisan.
BAB II	:	TINJAUAN PUSTAKA DAN DASAR TEORI
		Bab ini memuat penelitian-penelitian sebelumnya dan dasar teori untuk membantu penulis memahami bagaimana teori yang berhubungan dengan metode *Keyword Extraction* dan *Cosine Similarity* untuk penelitian ini.

BAB III	:	METODE PENELITIAN
		Bab ini mengungkapkan proses tahapan-tahapan penelitian dimulai dari mendapatkan data hingga proses pengolahan data yang ada.
BAB IV	:	HASIL DAN ANALISIS
		Pada bab ini penulis mengungkapkan hasil penelitian yaitu hasil *Keyword Extraction* menggunakan TF-IDF dan *Cosine Similarity*.
BAB V	:	KESIMPULAN DAN SARAN
		Bab ini penulis memaparkan kesimpulan proses penelitian dari awal hingga akhir

BAB II
TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Skripsi adalah karya ilmiah yang dituliskan berdasarkan pengetahuan khusus dan fakta-fakta yang jelas dan kemudian dirangkai menjadi masalah umum dengan bukti. Bagi mahasiswa, ini adalah Langkah terakhir dalam mendapatkan pendidikan adalah menyusun skripsi. Skripsi adalah syarat kelulusan bagi mahasiswa

Proses penyusunan skripsi ini dilakukan secara mandiri, dengan harapan mahasiswa tersebut dapat menyelesaikan masalah penelitian yang dibahas dalam skripsi mereka sendiri. Jika dilakukan secara mandiri, diharapkan setiap mahasiswa dapat menggunakan pengetahuan yang mereka peroleh selama kuliah unutk meningkatkan kapasitas ilmu pengetahuan mereka sendiri. Pada akhirya, diharapkan bahwa pengetahuan yang mereka peroleh akan mebantu dalam Menyusun skripsi..(Dewi 2018)

Dalam penyusunan skripsi terdapat banyak masalah yang sering dihadapi oleh mahasiswa khususnya mahasiswa Teknik Informatika Unissula. Berbagai masalah sering dilalui adalah dalam menentukan judul yang sulit dengan metode yang sesuai dan selaras dengan judul yang dimiliki. Oleh karena itu, dengan membuat sistem informasi trend judul skripsi yang dapat menampilkan trend judul skripsi atau metode apa saja yang banyak diangkat oleh mahasiswa Teknik Informatika mungkin dapat sedikit membantu memberikan refrensi metode bagi para mahasiswa yang sedang mengalami kesulitan.

Penelitian ini bertujuan untuk mengekstraksi kata kunci dari 50 artikel dan menggunakan algoritma Textrank. Hasilnya menunjukan bahwa algoritma Textrank memperoleh nilai akurasi terdiri dari *Precision, Recall*, dan *F-Measure*, dengan nilai rata-rata 20%.(Shiddiq 2019)

Penelitian yang dilakukan untuk meringkas dokumen berbahasa indonesia. Penelitian tersebut menggunakan dokumen non-fiksi dan fiksi serta menggunakan algoritma TF-IDF.Hasil penelitian menunjukan bahwa fitur bobot kata cukup baik untuk dokumen non-fiksi dengan durasi rata-rata 68,25 detik.(Widyasanti, Darma Putra, and Dwi Rusjayanthi 2018)

Penelitian yang dilakukan ini bertujuan membuat sistem untuk mengukur kesamaan dokumen. Kumpulan dokumen harus terdiri dari tiga dokumen, dan menggunakan algoritma TF-IDF. Hasil uji korelasi menunjukan bahwa ada korelasi kuat antara jumlah karakter pada dokumen PDF.(Andayani and Ryansyah 2017)

Penelitian ini bertujuan untuk pembobotan dengan algortima TF-7IDF dan WIDF dan menggunakan algoritma Nazief Adriani, KNN, dan *Cosine Similarity*. Teks dokumen berbahasa indonesia juga digunakan. Hasil uji coba menunjukkan bahwa pembobotan TF-IDF memiliki rasio akurasi 70,7%, sedangkan untuk pembobotan dengan menggunakan algoritma WIDF memiliki rasio akurasi 63,1%. Ini menunjukkan bahwa pembobotan algoritma TF-IDF lebih baik daripada pembobotan menggunakan algortima WIDF.(Susandi and Sholahudin 2017)

Penelitian ini bertujuan untuk membuat sistem temu kembali informasi untuk Syarah Hadits dengan menggunakan Hadist Shahih Bukhar Muslim dan menggunakan metode *Cosine Similarity* dan TF-IDF. *Tokenizing, removal of stopwords*, dan *stemming* adalah bidang-bidang yang hasil uji coba dapat diterapkan dengan baik serta mendapatkan nilai Recall 88,7%, Precision (Keakuratan) 100%, Accuracy (Keakuratan) 88,73%, dan Error Rate (Kesalahan rata-rata) 11,27%. (Amrizal 2018)

Dari kelima penelitian diatas dapat disimpulkan bahwa untuk membuat sistem pencarian trend judul skripsi menggunakan metode TF-IDF

sangatlah tepat yang mana dalam hal pembobotan TF-IDF memiliki *precision* yang terbilang cukup akurat. Dalam sistem yang akan dibuat, *Keyword Extraction* berperan sebagai pengolahan data mentah yang dari hasil *scrapping* website *repository* unissula yang memuat banyak dokumen skripsi mahasiswa. Metode *Cosine similarity* dalam hal ini digunakan untuk validasi antara kecocokan *keyword* yang dihasilkan oleh TF-IDF dengan *keyword* yang sudah ada didalam jurnal skripsi.

2.2 Dasar Teori

2.2.1 *Web Scraper*

Web Scrapper adalah *Tools* gratis yang disediakan oleh Google Chrome sebagai alat ekstraksi yang mudah dan digunakan untuk semua orang. *Tools* ini dikembangkan oleh websraper.io dan dapat di unduh di menu *Chrome Web Store* secara gratis. *Web Scraper* ini memiliki keunggulan yaitu dapat melakukan pengambilan data dengan mudah dan tidak perlu melalukan pemrograman dengan Pyhton, PHP, atau Javascript untuk memulai proses *scraping*.

2.2.2 *Natural Languange Processing* (NLP)

NLP atau bisa disebut juga dengan *Natural Language Processing* adalah fokus dari bidang ilmu computer dan linguistic yang dikenal sebagai pemrosesan bahasa alami (NLP). NLP biasa disebut dengan Komputasional Linguistik serta memiliki segmentasi tuturan (*segmentation of speech*) dan penandaan kelas kata(*part of speech tagging*) adalah beberapa pengembangannya. (Herwin 2019)

Saat ini, *natural language processing* (NLP) banyak digunakan dalam aplikasi sehari-hari meliputi Siri dan Google *Assistens* sebagai asisten virtual secara pribadi. Didunia Teknologi Industri saat ini menggunakan NLP adalah Langkah yang sangat penting untuk mendapatkan keuntungan dalam kompetitif. NLP dapat membantu dalam berbagai bidang kehidupan dalam menganalisis nilai dari data yang tidak terstruktur.(Rumaisa et al. 2021)

Dalam hal ini, terdapat dua metode yang digunakan, yaitu *pharsing* kalimat dan *lemmatization*.Pertama-tama, *pharsing* kalimat digunakan untuk memecah kalimat menjadi beberapa bagian yang terstruktur. Setelah itu, *lemmatization* dapat digunakan sebagai proses untuk mengidentifikasi kata kunci dari setiap kalimat. Proses ini sangat berguna untuk menghilangkan infleksi atau variasi kata yang mungkin muncul pada sebuah kalimat. Setelah kata kunci dasar ditemukan dengan sukses, pencarian dilakukan untuk menemukan jawaban yang sesuai dengan pertanyaan yang diajukan oleh pengguna.(Khoirunisa 2020)

2.2.3 *Keyword Extraction*

Keyword Extraction adalah metode dengan Teknik pengambilan kata kunci atau istilah penting dari sebuah teks atau dokumen dengan menggunakan algortima dan pemrosesan Bahasa alami (*Natural Language Processing*). Metode ini bertujuan untuk mengidentifikasi kata-kata atau frasa-frasa yang paling mewakili topik atau isi dokumen tersebut.(Pratama 2015)

Salah satu langkah dalam *Keyword Extraction* adalah TF-IDF, yang berarti frekuensi inversi kata dalam dokumen. Proses ini membandingkan frekuensi kata dalam dokumen dengan frekuensi kata yang sama di dokumen lain dalam koleksi yang sama.(Widyasanti, Darma Putra, and Dwi Rusjayanthi 2018)

Metode ini adalah suatu algoritma yang bertujuan untuk meningkatkan jumlah dokumen yang dapat ditemukan kembali dan dianggap relevan dengan menggabungkan konsep kata dalam sebuah dokumen dan mengandalkan frekuensi kehadiran kata tersebut dalam dokumen-dokumen lain. Kata-kata yang paling sesuai adalah kata-kata yang sering muncul dalam satu dokumen, namun jarang ditemukan di dokumen-dokumen lainnya.(Nurjannah and Fitri Astuti 2013)

Metode ini menganggap bahwa setiap kata memiliki nilai penting yang sebanding dengan jumlah kali kata tersebut muncul dalam teks. Permasaan berikut menunjukkan nilai *term* t pada teks d:

$$\boldsymbol{W(d,t) = TF(d,t)} \quad (1)$$

Dalam rumus di atas, terdapat representasi dari TF (*Term Frequency*), yang diwakili oleh TF(d,t), mengindikasikan frekuensi kemunculan term t dalam teks d. Penggunaan *Term Frequency* (TF) dalam information retrieval dapat meningkatkan *recall*, namun tidak selalu meningkatkan *precision*. Hal ini disebabkan oleh fakta bahwa term yang sering muncul cenderung memiliki daya pembeda yang rendah karena kemunculannya tersebar di banyak teks. Oleh karena itu, untuk mengatasi permasalahan ini, disarankan untuk menghapus term dengan frekuensi tinggi dari kumpulan *term* yang digunakan.

Jika *term frequency* berfokus pada kemunculan istilah dalam sebuah teks, yang menjadi focus *Inverse Document Frequency* (IDF). Kemunculan istilah tersebut diseluruh kumpulan *teks*. Di IDF, istilah yang jarang muncul lebih dihargai. Nilai IDF untuk term t adalah sebagai berikut: Nilai kepentingan tiap istilah diasumsikan berbanding terbalik dengan jumlah teks yang mengandung istilah tersebut.

$$\boldsymbol{IDF\ (t) = \log\left(\frac{N}{df(t)}\right)} \quad (2)$$

Dalam rumus IDF di atas, jumlah total teks atau dokumen dalam koleksi adalah N, dan jumlah dokumen yang mengandung term t adalah df(t). Penelitian terbaru telah berhasil menggabungkan TF dan IDF untuk menghitung term, yang menunjukkan bahwa gabungan keduanya menghasilkan hasil ekstraksi yang lebih baik.(Susandi & Sholahudin, 2017) Berikut ini adalah deskripsi kombinasi bobot dari istilah term t pada teks d:

$$\boldsymbol{TF - IDF(d,t) = TF(d,t).IDF(t)} \quad (3)$$

Pada Rumus diatas adalah Rumus kombinasi TF-IDF, yang dimana kombinasi TF-IDF mengalikan nilai TF dengan nilai IDF untuk mendapatkan skor akhir yang mencerminkan kepentingan relatif suatu kata dalam dokumen dan koleksi dokumen secara keseluruhan. Skor TF-IDF yang lebih tinggi menunjukan kata yang lebih penting atau spesifikasi dalam dokumen tersebut.

2.2.4 *Text Preprocessing*

Text Preprocessing adalah langkah yang dilakukan sebelum proses pengklasifikasian, dimana langkah tersebut bertujuan untuk membersihkan, menghapus, dan mengubah data, termasuk karakter non-alfabet dan kata-kata yang tidak relevan bagi sistem. (Muttaqin and Bachtiar 2016)

Tujuannya adalah untuk memastikan data yang diperoleh menjadi lebih bersih dan optimal, sehingga menghasilkan output yang optimal pula. Prosesing data juga bertujuan untuk mengolah data awal yang acak menjadi data yang terstruktur, sehingga dapat diterapkan dalam proses ekstraksi dengan baik..(H et al. n.d.)

Text Processing merupakan sebuah tahapan dalam data mining yang diperlukan untuk membuat kinerja menjadi maksimal dalam algoritma pengklasifikasian. Pada umumnya tahap dalam *Processing* antara lain adalah:

a) *Data Cleaning*

Proses membersihkan data dari gangguan dan ketidakkonsistenan dikenal sebagai data cleaning.Tahap ini membersihkan data dari tanda baca , symbol atau karakter yang tidak diperlukan seperti (!, @, #, *, %) dan sebagainya. (Sulastri and Gufroni 2017)

b) *Case Folding*

Case Folding adalah proses yang mengubah semua huruf dalam teks menjadi huruf kecil atau huruf besar, dengan tujuan untuk menyamakan kata-kata yang sebenarnya sama namun memiliki perbedaan kapitalisasi. Hal ini dilakukan karena ketika suatu kata mengandung arti yang sama namun

berbeda nama kapitalisasi akan dianggap berbeda sehingga diperlukannya tahapan case folding untuk menyamakan kapitalisasi dari kata tersebut.(Pratiwi 2022)

c) *Tokenizing*

Tokenizing atau tokenisasi adalah proses membagi atau memecah suatu dokumen menjadi bagian-bagian yang dikenal sebagai token.. Tokenisasi juga dapat melibatkan Langkah-langkah tambahan, seperti menghapus tanda baca, menggabungkan kata-kata terpisah sebagai token terpisah. Contohnya , kata "berlari" dan "berlari-lari" dapat dianggap sebagai dua token terpisah dalam proses tokenisasi. (Alita and Isnain 2020)

d) *Stopword Removal*

Dalam data *preprocessing*, penghapusan kata-kata umum yang dianggap tidak signifikan adalah tahap yang disebut *Stopword.* Dalam *Stopword Removal* yang biasanya terdiri dari kata-kata seperti "dan", "atau", "yang", "di" dan sebagainya akan dihapus dari teks. Dengan menghilangkan kata-kata tersebut kita dapat melakukan analisis teks yang lebih akurat dan efesien. Tujuan utama dari *Stopword Removal* adalah untuk menghilangkan kata-kata yang sering muncul dalam teks yang tidak memiliki makna yang signifikan. (K. and R. 2016)

BAB III
METODOLOGI PENELITIAN

3.1 Metode Penelitian

Dalam penelitian ini, metode yang digunakan adlaah *Keyword Extraction* TF-IDF. *Keyword Extraction* digunakan untuk mengambil kata kunci dengan mengekstrak *keyword* dari setiap abstrak dalam dokumen. Kemudian hasil *keyword* ekstraksi tersebut di hitung tingkat *similarity*-nya dengan *keyword* asli dari abstrak skripsi dengan menggunakan *Cosine Similarity* guna untuk mengetahui seberapa bagus *Keyword Extraction TF-IDF* dalam mengekstraksi *keyword* dalam dokumen. Adapun tahapan yang harus dilakukan dalam penelitian ini , antara lain:

3.1.1 Studi Literatur

Peneliti mempelajari teori dan praktik mengenai *Natural Language Processing (NLP) , Keyword Extraction,* algoritma TF-IDF dari beberapa sumber belajar. Meliputi dari e-book, artikel, jurnal, serta hasil dari penelitian terdahulu. Serta mempelajari karakteristik bagaimana cara menggunakan algoritma TF-IDF dalam melakukan ekstraksi *keyword* dalam sebuah dokumen.

3.1.2 *Data Collecting*

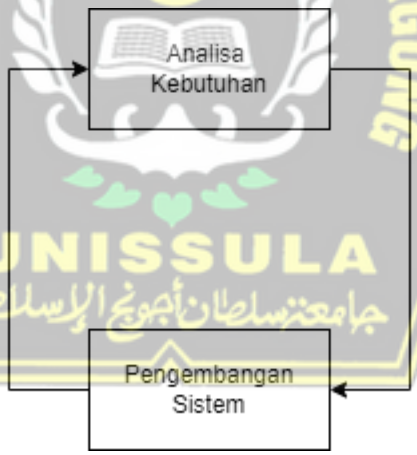
Teknik yang digunakan dalam tahap ini adalah dengan melakukan web scrapping pada website *Repository* Unissula dengan memilih opsi Fakultas Teknologi Industri -> Teknik Informatika dengan link sebagai berikut: http://repository.unissula.ac.id/view/divisions/jur=5Finformatika/ . Teknik web *scrapping* berfokus pada pengambilan dan ekstraksi data untuk mendapatkan informasi tertentu dari suatu website secara cepat dan otomatis tanpa harus mengambilnya secara manual. (A. Yani, Pratiwi, and Muhardi 2019) Dalam melakukan web *scrapping* peneliti menggunakan *tools* browser *extension* bernama *Web Scrapper* kemudian data yang diperoleh diekspor menjadi File Excel. Data skripsi yang diambil dari tahun 2018 – 2022. Data skripsi yang

diambil berupa Judul, penulis, tahun , dan abstrak. Jumlah data yang digunakan sekitar dari 216 data skripsi. Data tersebut dilakukan proses *Keyword Extraction* menggunakan algoritma TF-IDF untuk dapat mengambil kata kunci dari abstrak. Kemudia dilakukan proses *Cosine Similarity* untuk mencocokan hasil ekstraksi TF-IDF dengan keyword Asli dari Abstrak.

3.1.3 Perancangan Model Arsitektur Sistem

Sistem Pencarian Trend Judul Tugas Akhir ini akan dibuat dengan mengimplementasikan model *prototype*. Model *Prototype* adalah teknik pengembangan perangkat lunak yang menghasilkan model fisik kerja sistem untuk membantu pengembang dan pengguna berinteraksi selama proses pengembangan sistem informasi.(Firmansyah, Maulana, and Maulana 2021)

Model *Prototype* ini memiliki beberapa tahapan sebagai berikut:



Gambar 3. 1 Alur Perancangan Sistem dengan Metode *Prototype*

3.1.4 Tahapan Perancangan Model

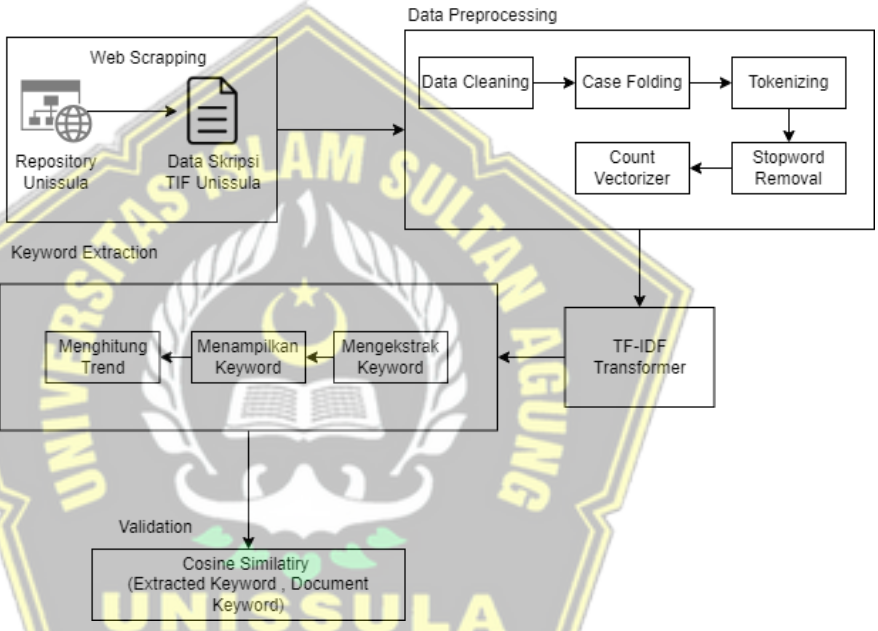
Metode *Prototype* memiliki tahapan-tahapan sebagai berikut:

1) Analisa kebutuhan

Pada tahap ini pengembang melakukan identifikasi perangkat lunak dan semua kebutuhan sistem yang akan dibuat.

2) Pengembangan Sistem

Jika *prototype* disetujui maka akan mulai pengembangan sistem dengan Bahasa pemrograman yang sesuai.



Gambar 3. 2 Metodologi perancangan alur sistem

Pada tahap pengembangan sistem dilakukan serangkaian tahapan dibawah:

a) Melakukan *Scrapping* website *repository* unissula dan mendapatkan data skripsi dari mahasiswa TIF Unissula
b) Pada tahapan *Data prepocessing* dilakukan *Data Cleaning* pada Abstrak untuk menghapus objek-objek yang tidak diperlukan.
c) Lalu, melakukan *Case Folding* untuk merubah huruf kapital menjadi huruf kecil.
d) Melakukan *Tekonizing* yaitu proses memecah teks menjadi kata per kata.

e) Membuat *Stopword Removal* atau menghapus kata-kata atau *keyword* yang sering muncul yang tidak memiliki makna signifikan.
f) Pada tahap *Count Vectorizer* yaitu mengubah setiap kata menjadi vektor atau angka.
g) Selanjutnya, tahap TF-IDF *Transformer* adalah merubah angka vektor menjadi skor TF-IDF.
h) Pada tahapan *Keyword Extraction*, peneliti mengekstrak *keyword* yaitu menampilkan *keyword* yang dihasilkan beserta vektornya.
i) Lalu, menampilkan *Keyword* hasil ektraksi.
j) Menghitung trend, untuk melihat seberapa banyak frekuensi kata tersebut muncul dalam dokumen.
k) Pada tahap *Validation*, yaitu validasi *Keyword* hasil ekstraksi dengan *Keyword* asli dari dokumen menggunakan *Cosine Similatiry* untuk dilihat sebarapa similarity TF-IDF dalam mengekstrak *Keyword*.

3.1.5 Gambaran Sistem

Penelitian ini akan mengembangkan sistem berbasis web untuk menampilkan trend metode skripsi. Pada output yang ditampilkan adalah kumpulan trend metode skripsi mahasiswa Teknik Informatika Unissula dari tahun 2018-2022. Terdapat menu chart yang menampilkan trend metode dari rentan waktu 5 tahun serta dapat menampilkan trend metode dari setiap tahunnya. Data Skripsi didapat dari Repository Unissula dengan cara *scrapping* menggunakan *extensions web scrapper* dengan link berikut, http://repository.unissula.ac.id/view/divisions/jur=5Finformatika . Data skripsi tersebut diolah didalam data *preprocessing* meliputi *Data Cleaning, Case Folding, Tokenizing, Stopword Removal, Count Vectorizer.* Lalu setelah data di olah dalam *preprocessing* masuk kedalam proses TF-IDF *Transformer* yaitu mengubah *vector* menjadi Skor TF-IDF untuk diolah kedalam proses *Keyword*

Extraction yaitu mengekstrak *keyword*, menampilkan *keyword*, menghitug trend. Lalu pada tahapan terakhir yaitu *validation* adalah tahap validasi tingkat *similarity keyword* hasil ekstraksi TF-IDF dengan *keyword* asli dari abstrak.

3.1.6 Indentifikasi Perangkat Lunak

Pada tahap pengembangan , peneliti menganalisis semua perangkat lunak yang diperlukan untuk mengembangkan aplikasi. Berikut adalah perangkat lunak yang digunakan dalam pengembangan sistem adalah :

1. Pyhton 3.11.33

   Pyhton adalah salah satu Bahasa pemrograman yang sering diadopsi oleh perusahaan-perusahaan besar. Python adalah Bahasa pemrograman tingkat tinggi yang sangat popular untuk digunakan dalam pengembangan perangkat lunak, kecerdasar buatan, pengolahan dan analisis data, dan berbagai program lainnya. (Muhammad Romzi and Kurniawan 2020)

2. *Library* Pandas 2.0.1

   Pandas adalah salah satu *Library* yang cukup popular dalam pemrograman menggunakan Python. Pandas digunakan untuk analisis data dan manipulasi data. Library ini menyediakan berbagai fungsi yang kuat untuk mempermudah pengolahan data. Fitur utama dari *Library* pandas adalah *Dataframe*, *Dataseries*, Membaca dan menulis Data, *Indexing* dan *Slicing*, Manipulasi data, Operasi Statistik, Visualisasi data.

3. *Scikit-learn* 1.2.2

   Scikit-learn adalah library yang popular dalam pemrograman menggunakan bahasa Pyhton untuk kegunaan *Machine Learning*. *Scikit-learn* bisa disebut dengan modul python yang mengintegrasikan berbagai algoritma pembelajaran mesin (*Machine Learning*). Library ini menyediakan berbagai algoritma untuk tugas-tugas seperti klasifikasi, pengelompokan, pemrosesan data, pengurangan dimensi, dan masih banyak lagi. (Riadi Silitonga and Munawar 2019)

4. Google Colab

   Google Colab adalah sebuah IDE untuk bahasa pemrograman Pyhton yang diproses oleh server Google yang sangat cepat. *Library* yang ditawarkannya mencakup Matplotib, Pandas, dan Numpy. (Gelar Guntara 2023)

5. Streamlit 1.23.1

   Streamlit merupakan sebuah kerangka kerja *open-source* yang digunakan untuk menciptakan aplikasi web yang interaktif menggunakan bahasa pemrograman Pyhton. *Framework* ini memiliki desain yang khusus untuk pengembangan aplikasi bidang data *science* dan *machine learning* yang dapat dibangun dan dijalan dengan mudah. Streamlit memiliki beberapa keunggulan yaitu: Sederhana dan cepat, *Automatic Rerun* , Interaktif, Integrasi dengan *Library* lain yang popular, Deployment yang mudah.(Prasetyo and Laksana 2022)

6. Visual Studio Code

   Visual Studio Code adalah software gratis yang dikembangkan oleh Microsoft dan tersedia untuk Windows, Linux, dan MacOS. Visual Studio Code memungkikan untuk seorang *developer* atau pengembang sistem melakukan *debugging* , control git dan Github. (Agustini and Kurniawan 2019)

3.1.7 Perancangan *User Interface*

Berikut ini merupakan rancangan desain dari sistem yang akan digunakan pada penelitian ini :

1. Halaman utama

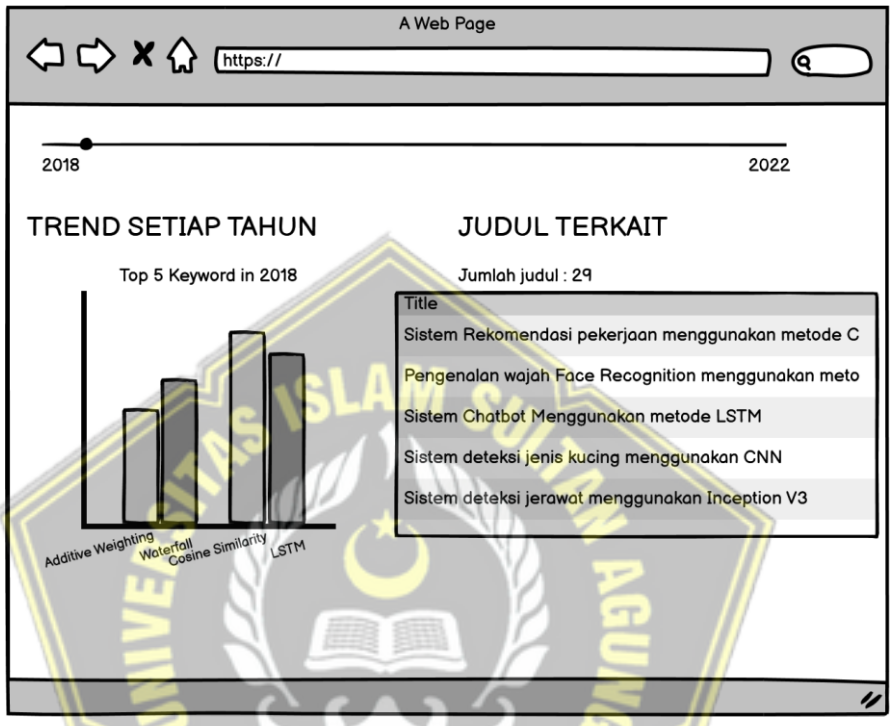
   Pada gambar 3. 3 Merupakan desain antar muka untuk halaman utama dimana halaman ini akan dilihat pertama kali oleh pengguna.

Gambar 3. 3 Halaman utama

Gambar 3.3 adalah rancangan halaman utama yang pertama kali akan dilihat oleh pengguna. Pada halaman ini aka nada header bertuliskan “TREND SKRIPSI INFORMATIKA” , serta dibawahnya terdapat 2 informasi menu yang sebelah kiri informasi mengenai Tahun dan jumal Skripsinya, serta pada bagian kanan adalah informasi Total Judul dari tahun 2018-2022. Dibawahnya juga terdapat menu *slider* yang bisa di arahkan ke kanan dan kiri sesuai keinginan pengguna , Ketika *slider* tersebut diarahkan ke tahun yang berbeda akan berubah pula tampilan *keyword* trend metode dibawahnya serta tampilan judulnya.

2. Halaman Trend & Judul

Pada gambar 3. 4 adalah halaman yang menampilkan trend *keyword* dari setiap tahun dan menampilkan judul yang terkait dalam trend metode tersebut.

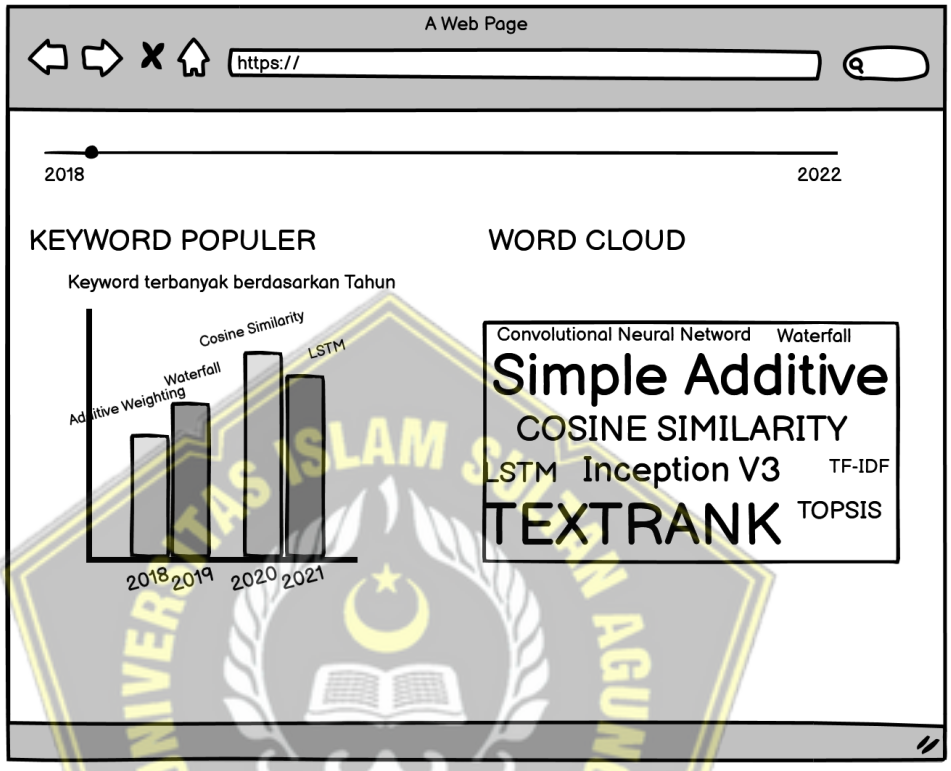


Gambar 3. 4 Halaman Trend & Judul

Gambar 3.4 adalah halaman yang menampilkan trend metode berserta frekuensinya skripsi pada tahun tertentu tergantung menu *slider* diatas di arahkan ke tahun berapa. Pada bagian kanannya adalah menu yang dapat menampilkan beberapa judul yang terkait pada chart trend metode skripsi tersebut.

3. Halaman *Keyword* populer & *WordCloud*

Pada gambar 3.5 adalah desain halaman untuk menampilkan *Keyword* populer & *WordCloud*



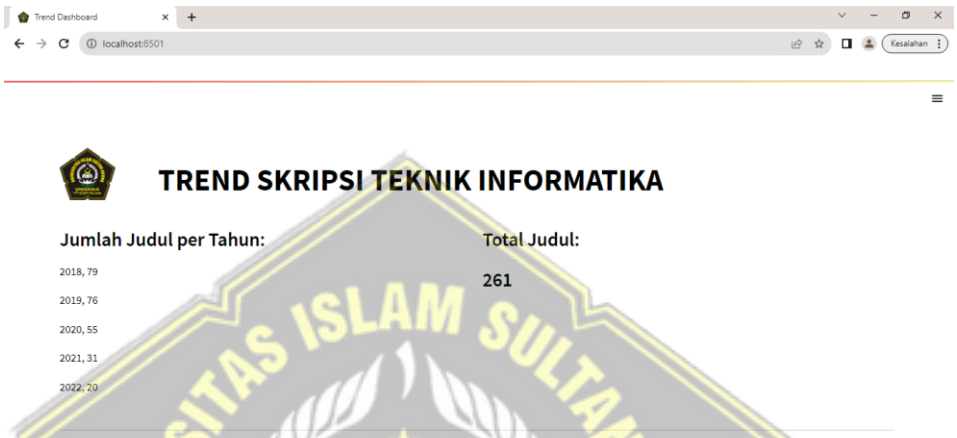
Gambar 3. 5 Halaman *Keyword* popular & *WordCloud*

Pada gambar 3.5 adalah desain halaman yang dapat menampilkan *Keyword* Populer & *Wordcloud*. *Keyword* Populer akan menampilkan metode apa yang paling banyak digunakan dalam setiap tahunnya. Serta pada bagian *wordcloud* akan memperlihatkan frekuensi *Keyword* metode yang muncul. Semakin besar *Keyword* yang ditampilkan *wordcloud* itu adalah *Keyword* dengan frekuensi yang banyak.

BAB IV
HASIL DAN ANALISIS

4.1 Hasil Implementasi Sistem

1. Halaman Utama



Gambar 4. 1 Halaman Utama

Gambar 4.1 merupakan tampilan halaman utama dimana pada halaman ini lah yang akan dilihat pertama kali oleh pengguna. Terdapat judul dari sistemnya "TREND SKRIPSI TEKNIK INFORMATIKA" . Didalam sistem tersebut juga terdapat Informasi "Jumlah Judul Per Tahun" yang didalamnya berisikan informasi jumlah judul skripsi dari tahun 2018-2022.

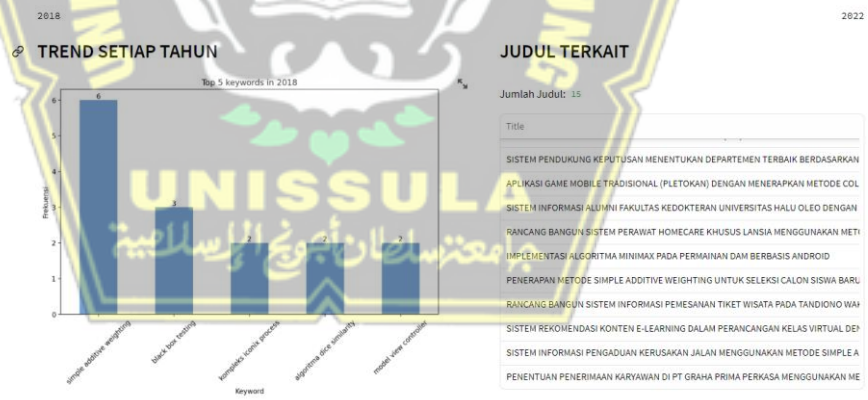
Masing-masing didalamnya terdapat pada tabel berikut:

Tabel 4. 1 Tahun dan Jumlah Skripsi

2018	79
2019	76
2020	55
2021	31
2022	20

Pada Tabel 4.1 adalah 5 tahun terakhir dari dataset skripsi yang tersedia, yaitu pada tahun 2018 memiliki 79 Judul Skripsi, tahun 2019 memiliki 76 judul skripsi, tahun 2020 memiliki 55 judul skripsi, tahun 2021 memiliki 31 judul skripsi, dan tahun 2022 memiliki 20 judul skripsi serta pada bagian "Total Judul" yang memiliki 261 Judul .

2. Halaman Trend setiap tahun 2018-2022
   a) Trend pada tahun 2018



Gambar 4. 2 Trend tahun 2018

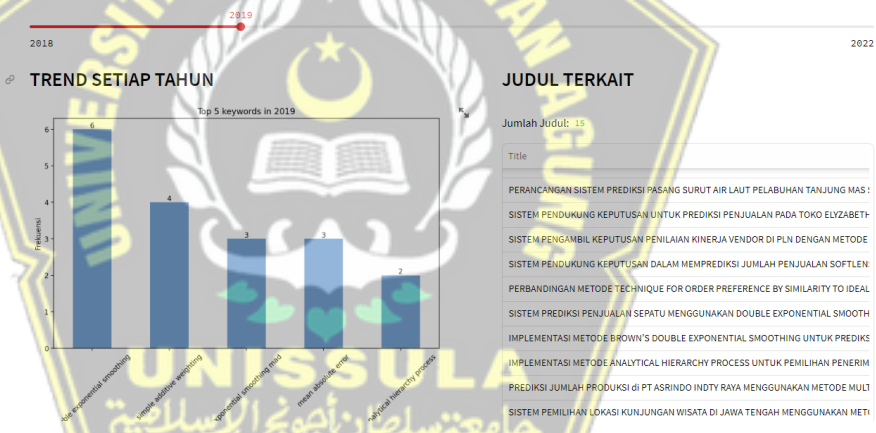
Gambar 4.2 adalah halaman Trend Skripsi pada tahun 2018 yang memiliki data trend pada tahun 2018 serta pada bagian Judul Terkait adalah judul yang terkait dalam trend pada tahun 2018.

Tabel 4. 2 Metode & Frekuensi tahun 2018

Metode	Frekuensi
Simple Additive Weighting	6
Black Box Testing	3
Komplex Iconic Process	2
Algoritma Dice Similarity	2
Model View Controller	2

Pada Tabel 4.2 adalah hasil output dari gambar 4.2 , metode *Simple Additive Weighting* memiliki frekuensi lebih tinggi daripada metode-metode lainnya.

b) Trend pada tahun 2019



Gambar 4. 3 Trend tahun 2019

Gambar 4. 3 adalah halaman Trend Skripsi pada tahun 2019 yang memiliki data trend pada tahun 2019 serta pada bagian Judul Terkait adalah judul yang terkait dalam trend pada tahun 2019.

Tabel 4. 3 Metode & Frekuensi tahun 2019

Metode	Frekuensi
Double Exponential Smoothing	6
Simple Additive Weighting	4
Exponential Smoothing mad	3
Mean absolute error	3
Analytical Hiearchy Process	2

Pada Tabel 1.4 adalah hasil output dari gambar 4.3 , metode *Double Exponential Smoothing* memiliki frekuensi lebih tinggi dengan nilai 6 yang artinya lebih tinggi daripada metode-metode lainnya.

c) Trend pada tahun 2020

Gambar 4. 4 Trend tahun 2020

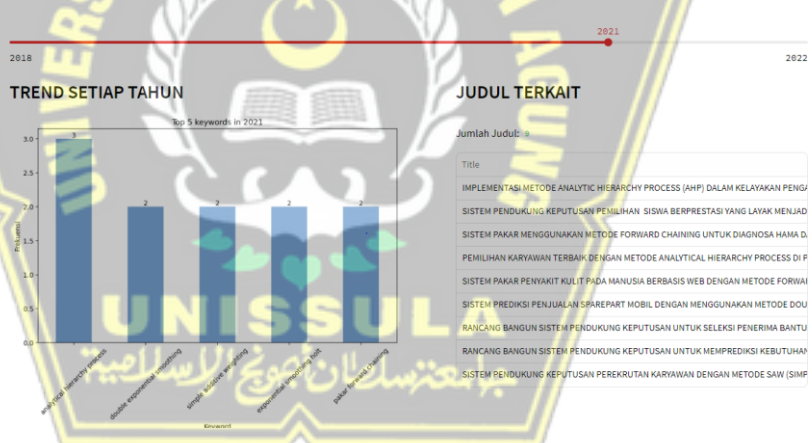
Gambar 4. 4 adalah halaman Trend Skripsi pada tahun 2020 yang memiliki data trend pada tahun 2020 serta pada bagian Judul Terkait adalah judul yang terkait dalam trend pada tahun 2020.

Tabel 4. 4 Metode & Frekuensi tahun 2020

Metode	Frekuensi
Preference Similarity Solution	5
Technique Prefrence Similarity	4
Similarity Solution Topsis	3
Simple Additive Weighting	3
Double Exponential Smoothing	3

Pada Tabel 4.4 adalah hasil output dari gambar 4.4 , metode *Preference Similarity Solution* memiliki frekuensi lebih tinggi dengan nilai 5 yang artinya lebih tinggi daripada metode-metode lainnya.

d) Trend pada tahun 2021



Gambar 4. 5 Trend pada tahun 2021

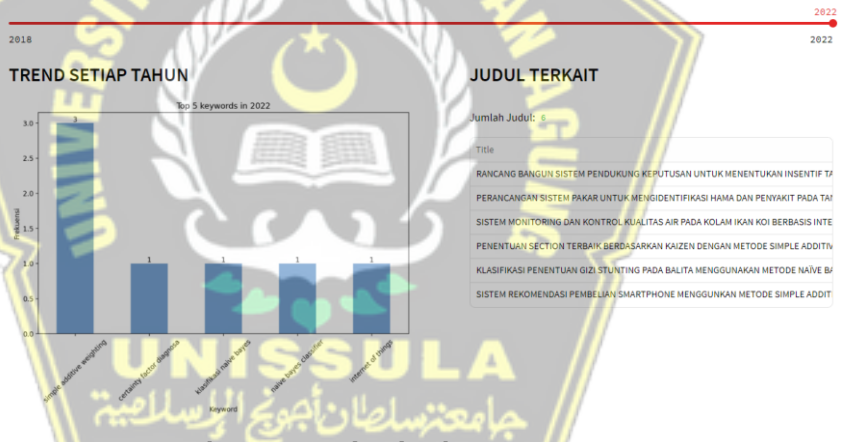
Gambar 4.5 adalah halaman Trend Skripsi pada tahun 2021 yang memiliki data trend pada tahun 2021 serta pada bagian Judul Terkait adalah judul yang terkait dalam trend pada tahun 2021.

Tabel 4. 5 Metode & Frekuensi tahun 2021

Metode	Frekuensi
Analytical Hierarcy Process	3
Double Exponention Smoothing	2
Simple Additive Weighting	2
Exponential Smoothing Holt	2
Forward Chaining	2

Pada Tabel 4.5 adalah hasil output dari gambar 4.5 , metode *Analytical Hierarcy Process* memiliki frekuensi lebih tinggi dengan nilai 3 yang artinya lebih tinggi daripada metode-metode lainnya.

e) Trend pada tahun 2022



Gambar 4. 6 Trend Pada tahun 2022

Gambar 4.6 adalah halaman Trend Skripsi pada tahun 2021 yang memiliki data trend pada tahun 2021 serta pada bagian Judul Terkait adalah judul yang terkait dalam trend pada tahun 2021.

Tabel 4. 6 Metode & Frekuensi tahun 2021

Metode	Frekuensi
Simple Additive Weighting	3
Certainty Factor Diagnosa	1
Klasifikasi Naïve Bayes	1
Naïve Bayes Classifier	1
Internet of Things	1

Pada Tabel 4.6 adalah hasil output dari gambar 4.6 , metode *Simple Additive Weighting* memiliki frekuensi lebih tinggi dengan nilai 3 yang artinya lebih tinggi daripada metode-metode lainnya.

3. Halaman Trend dari keseluruhan tahun 2018-2022

Gambar 4. 7 Trend Keseluruhan

Gambar 4.7 adalah halaman trend keseluruhan dari tahun 2018-2022 terlihat masing-masing tahun memiliki trend metode skripsi yang berbeda-beda. Pada bagian kanan terdapat gambar WordCloud yang menampilkan seluruh metode. Semakin besar suatu keyword dalam WordCloud itu memiliki frekuensi yang tinggi.

Tabel 4. 7 Trend Metode tahun 2018-2022

Tahun	Metode	Frekuensi
2018	*Simple Additive Weighting*	6
2019	*Double Exponential Smooting*	6
2020	*Preference Similarity Solution*	5
2021	*Analytical Hierarcy Process*	3
2022	*Simple Additive Weighting*	3

Pada tabel 4.7 adalah output dari trend judul skripsi meliputi dari tahun 2018 – 2022. Pada tampilan ini hanya menampilkan 1 Metode Skripsi yang paling tinggi frekuensinya di tahun tersebut.

Pada tahun 2018 metode *Simple Additive Weigthing* memiliki frekuensi 6, pada tahun 2019 terdapat metode *Double Exponential Smooting* yang memiliki frekuensi 6, tahun 2020 metode *Preference Similarity Solution* memiliki frekuensi 5, pada tahun 2021 terdapat metode *Analytical Hierarcy Process* memiliki frekuensi 3, sedangkan pada tahun 2022 metode *Simple Additive Weighting* memiliki frekuensi 3.

Pada Bagian *WordCloud* juga menampilkan banyak metode yang dipakai oleh mahasiswa Teknik Informatika Unissula berdasarkan banyaknya frekuensi kemunculan kata. Terlihat metode yang paling banyak dipakaiadalah *Similarity, Haversine,* dan topsis.

4.2 Implementasi *Keyword Extraction*

1. *Case Folding & Data Cleaning*

Gambar 4.8 dibawah adalah proses *Case Folding & Data Cleaning* untuk membuat semua *keyword* menjadi kecil dan membersihkan dari karakter-karakter yang tidak dibutuhkan.

```
import re
def pre_process(text):

    # lowercase
    text=text.lower()

    #remove tags
    text=re.sub("</?.*?>"," <> ",text)

    # remove special characters and digits
    text=re.sub("(\\d|\\W)+"," ",text)

    return text
#membuat kolom baru bernama text di sebelah kolom abstrak sebagai duplikat dari kolom abstrak untuk
diolah supaya kolom abstrak tetap utuh
df_idf['text'] = df_idf['Abstract']
#lambda adalah fungsi anonimus untuk menerapkan fungsi pre prosess pada kolom text meliputi
lowercasing, remove tags, remove digits
df_idf['text'] = df_idf['text'].apply(lambda x:pre_process(x))

#show the first 'text'
df_idf['text'][2]
```

Gambar 4. 8 *Data Cleaning & Casefolding*

Gambar 4.8 adalah sebuah Program Pyhton pada *Preprocessing* yaitu melakukan *data cleaning* pada teks yang ada di kolom "Abstract" dari *dataframe* "df_idf". Tahapaman *Preprocessing* ini terdiri dari Langkah-langkah berikut:

1.) Mengimport Library "re" yaitu "*Regular Expression*" disingkat Regex untuk memanipulasi teks.

2.) Mendefinisikan fungi "*pre_process(text)",* yang berisi *preprocessing* teks. Pertama menghilangkan perbedaan huruf kecil dan huruf besar yaitu menjadikannya semua huruf kecil. Selanjutnya, *Library* Regex mencari tag HTML atau XML dalam mencari teks dan mengganti dengan string kosong.

3.) Selanjutnya, Regex digunakan untuk menghapus karakter khusus dan angka dari teks dan menggantinya dengan spasi.

4.) Setelah mendefinisikan fungi *"pre_process*", program membuat kolom baru dengan nama '*text*' di *dataframe* 'df_idf' dan mengisi nilainya dengan isi kolom 'Abstract'. Dengan demikian, teks dari kolom 'Abstract' akan disimpan juga di 'Kolom *'text'*.

5.) Terakhir program menerapkan fungsi '*pre_process'* pada setiap baris teks di kolom *'text'* menggunakan fungsi '*apply()*', sehingga teks pada kolom '*text'* telah melalui proses *preprocessing*.

2. Tokenisasi , *Stopword Removal* , *Countvectorizer*

Gambar 4.9 adalah proses Tokenisasi Membuat *Stopword Removal* dan melakukan *CountVectorizer.*

```
from sklearn.feature_extraction.text import CountVectorizer
import re

with open('stopwords.txt', 'r') as file:
    stopwords = [word.strip() for word in file.readlines()]

#get the text column
docs=df_idf['text'].tolist()

#create a vocabulary of words,
#ignore words that appear in 85% of documents,
#eliminate stop words
cv=CountVectorizer(stop_words=stopwords,
                   ngram_range=(3,3),
                   min_df=0.003,
                   max_features=120
                   )
word_count_vector=cv.fit_transform(docs)
word_count_vector.shape
```

Gambar 4. 9 Tokenisasi , *Stopword Removal, CountVectorizer*

Gambar 4. 9 adalah proses Tokenisasi , *Stopword Removal* , *CountVectorizer* yang dilakukan dalam Langkah-langkah berikut:

1.) Tokenisasi dilakukan secara otomatis oleh fungsi '*Count Vectorizer*' dari Library '*sk.learn.feature_extraction.text'*. Membaca file bernama 'stopword.text' yang berisi daftar kata *stopwords.*

2.) Didalam objek *CountVectorizer* memiliki beberapa konfigurasi,'stop_words=stopwords' yaitu menggunakan *stopwords* yang telah dibaca sebelumnya untuk dihilangkan dari data teks. Membuat n-grams dengan Panjang 3. Serta mengabaikan kata-kata yang muncul dalam kurang dari 0.3% atau 0.003 dari seluruh dokumen. Membatasi jumlah kata yang diekstrak, hanya mengambil 120 kata dengan frekuensi tertinggi.

3. TF-IDF *Transformer*

Gambar 4. 10 adalah proses TF-IDF *Transformer* yaitu merubah *keyword* yang sudah dipecah-pecah dan di jadikan vektor / angka menjadi skor TF-IDF.

```
#mengimport Tfidftransformer dari modul feature_extraction.text dari library sklrean
from sklearn.feature_extraction.text import TfidfTransformer

#smooth_idf menghindari pembagian dengan 0
tfidf_transformer=TfidfTransformer(smooth_idf=True,use_idf=True)
#menerapkan tfidf transformer kedalam WCV
tfidf_transformer.fit(word_count_vector)
```

Gambar 4. 10 Tf-Idf *Transformer*

Gambar 4.10 adalah tahapan untuk mentransformasi matriks TF yang telah dibuat sebelumnya menrjadi matriks TF-IDF menggunakan 'TFidfTransformer' dari *library* scikit-learn. TF-IDF Transformer akan memberikan bobot yang lebih tinggi pada kata-kata yang lebih jarang muncul diseluruh koleksi dokumen dan memberikan bobot yang lebih

rendah pada kata-kata yang muncul dalam banyak dokumen. ‘smooth_idf=True’ adalah argument yang menambahkan nilai 1 ke frekuensi dokumen untuk menghindari pembagian dengan nol saat menghitung IDF. ‘use_idf=True’ adalah argument yang menentukan apakah ingin menggunakan IDF dalam perhitungan TF-IDF, jika disetel False, itu akan menghitung TF saja.

4. **Mengekstrak *Keyword***

```
# read test docs into a dataframe and concatenate product and desc
df_test=pd.read_excel('repo.xlsx')
df_test['text'] = df_test['Title'] + df_test['Abstract']
df_test['text'] = df_test['text'].apply(lambda x:pre_process(x))

# get test docs into a list
docs_test=df_test['Abstract'].tolist()

def sort_coo(coo_matrix):
    tuples = zip(coo_matrix.col, coo_matrix.data)
    return sorted(tuples, key=lambda x: (x[1], x[0]), reverse=True)

def extract_topn_from_vector(feature_names, sorted_items, topn=10):
    """get the feature names and tf-idf score of top n items"""

    #use only topn items from vector
    sorted_items = sorted_items[:topn]

    score_vals = []
    feature_vals = []

    for idx, score in sorted_items:
        fname = feature_names[idx]

        #keep track of feature name and its corresponding score
        score_vals.append(round(score, 3))
        feature_vals.append(feature_names[idx])

    #create a tuples of feature,score
    #results = zip(feature_vals,score_vals)
    results= {}
    for idx in range(len(feature_vals)):
        results[feature_vals[idx]]=score_vals[idx]

    return results

  # you only needs to do this once
feature_names=cv.get_feature_names_out()

# get the document that we want to extract keywords from
doc=docs_test[0]

#generate tf-idf for the given document
tf_idf_vector=tfidf_transformer.transform(cv.transform([doc]))

#sort the tf-idf vectors by descending order of scores
sorted_items=sort_coo(tf_idf_vector.tocoo())

#extract only the top n; n here is 10
keywords=extract_topn_from_vector(feature_names,sorted_items,10)
```

Gambar 4. 11 Mengekstrak *Keyword*

Pada Gambar 4.11 adalah proses untuk melakukan ekstraksi *keyword* yang dilakukan pada beberapa Langkah sebagai berikut:

a) Langkah pertama

```
df_test = pd.read_excel('repo.xlsx')
df_test['text']       =       df_test['Title']       +
df_test['Abstract']
df_test['text']   =   df_test['text'].apply(lambda   x:
pre_process(x))
docs_test = df_test['Abstract'].tolist()
```

Kode diatas adalah Langkah untuk membaca file Excel ‘repo.xlsx’ dan memuatnya ke dalam DataFrame ‘df_test’. Menggabungkan dua kolom Title dan ‘Abstract’ menjadi satu kolom baru bernama ‘text’ dalam DataFrame ‘df_test’. Menggunakan fungsi ‘pre_process’ pada setiap elemen dalam kolom ‘text’.

Kemudian mengambil kolom ‘abstract’ dari DataFrame ‘df_test’ dan mengonversi menjadi list yang berisi teks dari setiap abstrak.

b) Langkah kedua

```
def sort_coo(coo_matrix):
    tuples = zip(coo_matrix.col, coo_matrix.data)
    return sorted(tuples, key=lambda x: (x[1], x[0]),
reverse=True)
def          extract_topn_from_vector(feature_names,
sorted_items, topn=10):
    # ...
feature_names = cv.get_feature_names_out()
doc = docs_test[0]
tf_idf_vector                                        =
tfidf_transformer.transform(cv.transform([doc]))
sorted_items = sort_coo(tf_idf_vector.tocoo())
keywords   =   extract_topn_from_vector(feature_names,
sorted_items, 10)
```

Kode diatas adalah Langkah untuk mendefinisikan fungsi ‘sort_coo’ yang digunakan untuk mengurutkan sebuah sparse matrix dalam format COOrdinate (COO). Spare matrix ini adalah hasil ekstraksi fitur dengan metode TF-IDF. Menggunakan fungsi ‘extract_top_from_vector’ yang digunakan untuk mengekstrak top-n fitur beserta nilai TF-IDF dari hasil perhitungan TF-IDF. Kemudian menggunakan CountVectorizer pada cv untuk mendapatkan nama-nama fitur yang digunakan dalam proses ekstraksi fitur sebelumnya. Kemudian selanjutnya, memilih dokumen pertama dari daftar ‘docs_test’, menerapkan transformasi TF-IDF pada dokumen tersebut menggunakan ‘tfidf_transformer’, yang dibuat dengan objek ‘cv’. Mengurutkan vector TF-IDF secara menurun dan menyimpan hasilnya dalam ‘sorted_item, serta menggunakan fungsi ‘extract_top_from_vector’ untuk mengekstrak 10 kata kunci teratas dari dokumen tersebut dan menyimpannya dalam variable ‘keywords’.

5. Menghitung Trends

Gambar 4.12 adalah kode untuk melakukan perhitungan Trend dengan *keyword* yang dihasilkan menggunakan *Count*.

```
grouped = dft.groupby(['tahun', 'keywords']).size().reset_index(name='count')
grouped.sort_values(by='count', ascending=False)
```

Gambar 4. 12 Menghitung Trends

Gambar 4. 12 adalah tahapan menghitung trend dengan menggunakan metode *count* yang memiliki Langkah-langkah sebagai berikut:

1) ‘*grouped = dft.groupby*(['tahun' '*keywords*']).*size()*.reset_index(*name='count'*):’

Pada langkah ini, *DataFrame* dft yang telah diubah ke dalam bentuk tabular dan dibersihkan akan dikelompokkan berdasarkan kolom 'tahun' dan

'keywords'. Kemudian, metode *size*() digunakan untuk menghitung jumlah kemunculan (bukan nilai unik) dari setiap pasangan tahun dan kata kunci. Hasil dari perhitungan ini akan menghasilkan *Series* yang berisi jumlah kemunculan untuk setiap pasangan tahun dan kata kunci. Selanjutnya, dengan menggunakan metode reset_index(), Series tersebut diubah kembali menjadi DataFrame dengan kolom 'tahun', '*keywords*', dan '*size*'. Kolom '*size*' akan diberi nama ulang menjadi '*count*' menggunakan argumen *name='count'*.

2) '*grouped.sort_values(by='count', ascending=False):'*

Setelah *DataFrame grouped* dihasilkan, langkah ini akan mengurutkan *DataFrame* tersebut berdasarkan nilai *'count*' secara menurun (dari besar ke kecil). Kata kunci yang memiliki jumlah kemunculan paling tinggi akan muncul di bagian atas *DataFrame*, sedangkan kata kunci dengan jumlah kemunculan paling rendah akan muncul di bagian bawah.

4.3 Validasi Implementasi Algoritma

Dalam penelitian ini menggunakan validasi *cosine similarity.* Tujuannya agar dapat mengukur hasil klasifikasi, termasuk kemampuan model dalam menemukan semua dokumen yang mirip, dan bagaimana kemiripian hasil ekstraksi algoritma TF-IDF dengan keyword asli dari dokumen Tugas Akhir.

```
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
import numpy as np
from numpy.linalg import norm

# Combine all to make single corpus of text (i.e. list of sentences)
corpus = pd.concat([matching['kata_kunci'], matching['keywords']], axis=0, ignore_index=True).to_list()
# print(corpus)  # Display list of sentences

# Vectorization using basic Bag of Words (BoW) approach
vectorizer = CountVectorizer()
X = vectorizer.fit_transform(corpus)
# print(vectorizer.get_feature_names_out())  # Display features
vect_sents = X.toarray()

cosine_sim_scores = []
# Iterate over each vectorised sentence in the A-B pairs from the original dataframe
for A_vect, B_vect in zip(vect_sents, vect_sents[int(len(vect_sents)/2):]):
    # Calculate cosine similarity and store result
    cosine_sim_scores.append(np.dot(A_vect, B_vect)/(norm(A_vect)*norm(B_vect)))
# Append results to original dataframe
matching.insert(2, 'cosine_sim', cosine_sim_scores)
matching
```

Gambar 4. 13 Validasi *keyword* menggunakan *Cosine Similarity*

Gambar 4.13 adalah kode untuk melakukan proses validasi *keyword* asli dengan *keyword* hasil ekstraksi. Proses validasi memiliki Langkah-langkah sebagai berikut:

1. Mengimport *Library* yang diperlukan

```
import pandas as pd
from sklearn.feature_extraction.text import
CountVectorizer
import numpy as np
from numpy.linalg import norm
```

2. Menggabungkan dua kolom

```
corpus = pd.concat([matching['kata_kunci'],
matching['keywords']], axis=0,
ignore_index=True).to_list()
```

   Menggabungkan dua kolom ‘kata_kunci’ dan ‘keywords’ dari DataFrame ‘matching’ menjadi satu korpus.

3. Melakukan vektorisasi pada korpus

```
vectorizer = CountVectorizer()
X = vectorizer.fit_transform(corpus)
vect_sents = X.toarray()
```

Objek ‘vectorizer’ digunakan untuk mengubah data teks menjadi format numerik yang mempresentasikan frekuensi setiap kata pada setiap kalimat. Variabel X berisi hasil vektorisasi, dan ‘vec_sents’ berisi kalimat dalam bentuk array numpy.

4. Menghitung kesamaan cosine untuk setiap pasangan kalimat dalam korpus

```
cosine_sim_scores = []
for A_vect, B_vect in zip(vect_sents,
vect_sents[int(len(vect_sents)/2):]):
    cosine_sim_scores.append(np.dot(A_vect,
B_vect)/(norm(A_vect)*norm(B_vect)))
```

Kode diatas melakukan iterasi untuk setiap pasangan kalimat A_vect dan B_vect dari array vec_sents. Fungsi np.dot() digunakan untuk menghitung hasil perkalian antara dua vector, dan fungsi norm() menghitung norma Euclidean dari setiap vector. Kemudian, kesamaan kosinus anatara dua kalimat dihitung sebagai hasil perkalian dot dibagi dengan hasil perkalian norma keduanya.

5. Menambahkan sor kesamaan kosinus yang dihitung kedalam DataFrame

```
matching.insert(2, 'cosine_sim', cosine_sim_scores)
```

Menambahkan kolom baru dengan nama ‘cosine_sim’ pada DataFrame yang berisi skor kesamaan kosinus. DataFrame berisi dua kolom ‘kata_kunci’ dan ‘keywords’ yang berisi data teks yang akan dibandingkan kesamaanya. Lalu skornya ditambahkan sebagai kolom baru ‘cosine_sim’.

Tabel 4. 8 Ouput Validasi

Kata_kunci	Keywords	Cosine_sim
Sistem pendukung keputusan intensif tahunan	*Simple additive weighting*	0.577350
Sistem estimasi persediaan oli regresi	Regresi linier berganda, *mean absolute percentage*	0.293171
Sistem pakar certainty factor bawang	Pakar *certainty*, pakar *certainty factor*	0.585540
Prediksi masa studi *case based reasoning*	*Cased based reasoning*	0.61372
Sistem pendukung keputusan penilaian karyawan	*Support analitical hierarchy*	0.447214
Logika fuzzy metode mamdani fuzzy	Logika fuzzy mamdani	0.800641
Internet of things penyiraman real time	Internet of things	0.577350
Sistem informasi penjualan griya safira	Griya safira ungaran, safira ungaran	0.707107
Jalan raya simple additive weighting	*Simple additive weighting*	0.547723
Penerimaan karyawan metode simple additive	*Simple additive weighting*	0.7070107

6. Menampilkan statistic dari DataFrame

```
Matching.describe().T
```

Kode diatas adalah untuk melihat hasil statistik persamaan kosinus dari rangkaian kode diatas untuk melihat nilai rata-ratabya.

	Mean
Cosine_sim	0.437248

Hasil validasi menunjukan perbandingan antara *keyword* hasil ekstraksi dengan *keyword* asli dari dokumen memiliki nilai rata-rata 0.437248.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil dari penelitian ini, dapat diambil kesimpulan bahwa sistem pencarian trend judul Tugas Akhir ini bekerja dan berjalan dengan baik.Dataset yang didapat dari website *repository* unissula dengan cara *scraping* menggunakan *Tools web scraper*, *web scraper* adalah *extensions* google untuk *scraping* data tanpa perlu melakukan pemrograman. Dataset yang diambil dari tahun 2018 – 2022 yang terdiri dari Judul & Abstrak yang diolah melalui *Data Preprocessing*, TF-IDF *Transformer*, mengekstraksi *keyword*, serta melakukan validasi menggunakan Algoritma *Cosine Similarity* untuk dilihat skor kemiripan antara *keyword* Asli dengan *keyword* hasil ekstraksi menggunakan Algoritma TF-IDF. Sistem dapat menampilkan trend metode skripsi dari setiap tahun 2018-2022 dan dapat menampilkan trend metode dalam kurun waktu 5 tahun serta dapat menampilkan *wordcloud* untuk melihat seberapa besar frekuensi *keyword*-nya. Validasi sistem menggunakan *Cosine Similarity* yang memiliki nilai kemiripan *keyword* asli dan *keyword* hasil ekstraksi sebesar rata-rata 0.437248.

5.2 Saran

Saran yang dapat diterapkan untuk pengembangan sistem ini lebih lanjut nantinya adalah : Pada sistem pencarian trend judul Tugas Akhir menggunakan metode *keyword extraction* adalah dengan menambahkan fitur untuk menambahkan fitur *insert data* , agar dapat menambahkan data baru pada dataset sistem yang sudah ada.

DAFTAR PUSTAKA

A. Yani, Dhita Deviacita, Helen Sasty Pratiwi, and Hafiz Muhardi. 2019. "Implementasi Web Scraping Untuk Pengambilan Data Pada Situs Marketplace." *Jurnal Sistem dan Teknologi Informasi (JUSTIN)* 7(4): 257. http://dx.doi.org/398.29/j.powtec.2016.12.055%0A

Agustini, and Wahyu Joni Kurniawan. 2019. "Sistem E-Learning Do'a Dan Iqro' Dalam Peningkatan Proses Pembelajaran Pada TK Amal Ikhlas." *Jurnal Mahasiswa Aplikasi Teknologi Komputer dan Informasi* 1(3): 154–59. http://www.ejournal.pelitaindonesia.ac.id/JMApTeKsi/index.php/JOM/article/view/526. https://doi.org/10.1016/j.ijfatigue.2019.02.006%0A

Alita, Debby, and Auliya Rahman Isnain. 2020. "Pendeteksian Sarkasme Pada Proses Analisis Sentimen Menggunakan Random Forest Classifier." *Jurnal Komputasi* 8(2): 50–58. https://doi.org/13.9832/j.matlet.2019.04.024

Amrizal, Victor. 2018. "Penerapan Metode Term Frequency Inverse Document Frequency (Tf-Idf) Dan Cosine Similarity Pada Sistem Temu Kembali Informasi Untuk Mengetahui Syarah Hadits Berbasis Web (Studi Kasus: Hadits Shahih Bukhari-Muslim)." *Jurnal Teknik Informatika* 11(2): 149–64. https://doi.org/210.15408/jti.v11i2.8623

Andayani, Sri, and Ady Ryansyah. 2017. "Implementasi Algoritma TF-IDF Pada Pengukuran Kesamaan Dokumen." *JuSiTik : Jurnal Sistem dan Teknologi Informasi Komunikasi* 1(1): 53. https://doi.org/11.32524/jusitik.v1i1.218

Azis Maarif, Abdul. 2015. "Penerapan Algoritma Tf-Idf Untuk Pencarian Karya Ilmiah." *Universitas Dian Nuswantoro*: 4. repository.unair.ac.id/29371/3/15 BAB II.pdf.

Dewi, Nana ratna. 2018. "Kesulitan Mahasiswa Semester Akhir Dalam Menyusun Skripsi." *Journal of Materials Processing Technology* 1(1): 1–8. http://dx.doi.org/10.1016/j.cirp.2016.06.001%0Ahttp://dx.doi.o.

Firmansyah, Yoki, Reza Maulana, and Muhammad Sony Maulana. 2021. "Implementasi Metode SDLC Prototype Pada Sistem Informasi Indeks Kepuasan Masyarakat (IKM) Berbasis Website Studi Kasus Dinas Kependudukan Dan Catatan Sipil." *Jurnal Sistem dan Teknologi Informasi (Justin)* 9(3): 315. https://doi.org/90.26418/.g9i3.46964

Gelar Guntara, Rangga. 2023. "Pemanfaatan Google Colab Untuk Aplikasi Pendeteksian Masker Wajah Menggunakan Algoritma Deep Learning YOLOv7." *Jurnal Teknologi Dan Sistem Informasi Bisnis* 5(1): 55–60. https://doi.org/172.36/j.matlet.2019.127252

Gilabert, Pere L. et al. 2009. "An Efficient Combination of Digital Predistortion and Ofdm Clipping for Power Amplifiers." *International Journal of RF and Microwave Computer-Aided Engineering* 19(5): 583–91.

H, Aris Tri Jaka, Program Studi Informatika, Mencari Google, and Artikel Perkembangan. "Preprocessing Teks Untuk Meminimalisir Kata Yang TIDAK Berarti." : 1–9. https://doi.org/220.11316/j..2020.11313252

Herwin, Herwin H. 2019. "Super Agent Chatbot '3S' Sebagai Media Informasi Menggunakan Metoda Natural Language Processing(NLP)." *Jurnal Teknologi Dan Open Source* 2(1): 53–64. https://doi.org/15.7342/j.journal.2019.4572123152%A

K., Jaideepsinh, and Jatinderkumar R. 2016. "Stop-Word Removal Algorithm and Its Implementation for Sanskrit Language." *International Journal of Computer Applications* 150(2): 15–17. https://doi.org/10.4674/j.jjournal.2016.4211232720A

Khoirunisa, Rifa. 2020. "Penggunaan Natural Language Processing Pada Chatbot Untuk Media Informasi Pertanian." *Indonesian Journal of Applied Informatics* 4(2): 55.

https://doi.org/98.7263/applied.2020.6745752%8H

Muhammad Romzi, and Budi Kurniawan. 2020. "Pembelajaran Pemrograman Python Dengan Pendekatan Logika Algoritma." *JTIM: Jurnal Teknik Informatika Mahakarya* 03(2): 37–44.

https://doi.org/76.2645/j.tim.2020.153452725

Muttaqin, Firdaus Akhmad, and Adam Mukaharil Bachtiar. 2016. "Implementasi Teks Mining Pada Aplikasi Pengawasan Penggunaan Internet Anak 'Dodo Kids Browser.'" *Jurnal Ilmiah Komputer dan Informatika*: 1–8. https://doi.org/17.75765/j.kids.2016.87385372

Nurjannah, Musfiroh, and Inda Fitri Astuti. 2013. "Penerapan Algoritma Term Frequency-Inverse Document Frequency (Tf-Idf) Untuk Text Mining Mahasiswa S1 Program Studi Ilmu Komputer FMIPA Universitas Mulawarman Dosen Program Studi Ilmu Komputer FMIPA Universitas Mulawarman." *Jurnal Informatika Mulawarman* 8(3): 110–13. https://doi.org/98.2984/jtext.2013.74537

Prasetyo, Aditya Budi, and Tri Ginanjar Laksana. 2022. "Optimasi Algoritma K-Nearest Neighbors Dengan Teknik Cross Validation Dengan Streamlit (Studi Data: Penyakit Diabetes)." *Journal of Applied Informatics and Computing (JAIC)* 6(2): 194. http://jurnal.polibatam.ac.id/index.php/JAIC.

Pratama, Bagus Widya. 2015. "Analisis Efektifitas Pengukuran Keterkaitan Antar Teks Menggunakan Metode Salient Semantic Analysis Dengan TextRank for Keyword Extraction Sebagai Preprocessing." 2(2): 6665–71. https://doi.org/114.2842/.salient.2015.1245747252

Pratiwi, Inggrid Yanuar Risca. 2022. "Hoax News Identification Using Machine Learning Model from Online Media in Bahasa Indonesia." *MATRIX : Jurnal Manajemen Teknologi dan Informatika* 12(2): 58–67. https://doi.org/85.2741/j.matrix.2022.1353527252%0123

Riadi Silitonga, Yosua, and Munawar. 2019. "Sistem Pendeteksi Berita Hoax Di Media Sosial Dengan Teknik Data Mining Scikit Learn." *Jurnal Ilmu Komputer* 4: 173. www.beritasatu.com,. https://doi.org/93.7343/j.scikit.2019.1235437252%0103

Rumaisa, Fitrah et al. 2021. "Penerapan Natural Language Processing (NLP) Di Bidang Pendidikan." *Jurnal Inovasi Masyarakat* 01(03): 232–35. https://doi.org/297.2984/j.nlp.2021.12424237252%0ad

Shiddiq, Muhammad Aufa. 2019. "Ekstraksi Kata Kunci Pada Artikel

Menggunakan Metode Textrank." http://etheses.uin-malang.ac.id/17153/.

Sulastri, Heni, and Acep Irham Gufroni. 2017. "Penerapan Data Mining Dalam Pengelompokan Penderita Thalassaemia." *Jurnal Nasional Teknologi dan Sistem Informasi* 3(2): 299–305.

Susandi, Diki, and Usep Sholahudin. 2017. "Pemanfaatan Vector Space Model Pada Penerapan Algoritma Nazief Adriani, KNN Dan Fungsi Similarity Cosine Untuk Pembobotan IDF Dan WIDF Pada Prototipe Sistem Klasifikasi Teks Bahasa Indonesia." *ProTekInfo(Pengembangan Riset dan Observasi Teknik Informatika)* 3(1): 22–29. https://doi.org/10.30656/protekinfo.v3i0.54

Widyasanti, N. K., Darma Putra, I. K. G., & Dwi Rusjayanthi, N. K. (2018). Seleksi Fitur Bobot Kata dengan Metode TFIDF untuk Ringkasan Bahasa Indonesia. *Jurnal Ilmiah Merpati (Menara Penelitian Akademika Teknologi Informasi)*, *6*(2), 119. https://doi.org/126.24843/jim.2018.v06.i02.p06

Widyasanti, Ni Komang, I Ketut Gede Darma Putra, and Ni Kadek Dwi Rusjayanthi. 2018. "Seleksi Fitur Bobot Kata Dengan Metode TFIDF Untuk Ringkasan Bahasa Indonesia." *Jurnal Ilmiah Merpati (Menara Penelitian Akademika Teknologi Informasi)* 6(2): 119.