LAPORAN PROJEK AKHIR NLP

INTEGRASI *TF-IDF* DAN ALGORITMA *COSINE SIMILARITY* UNTUK DETEKSI TINGKAT KEMIRIPAN JUDUL DAN DESKRIPSI PENELITIAN

Disusun Oleh :

Yosep Adriana Fauzi Ramdani 227006033

JURUSAN INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS SILIWANGI

2025

1. LATAR BELAKANG PROJEK

Dalam dunia akademik khususnya pada tingkat perguruan tinggi, plagiarisme dan kemiripan konten seperti skripsi, tesis, dan disertasi menjadi isu yang krusial untuk diangkat. Pada penelitan [1] telah dilakukan deteksi tingkat kemiripan judul penelitian, namun jika hanya judul saja yang dideteksi pastinya kita hanya mengetahui sebatas judul saja yang mirip sedangkan kita tidak tahu apakah isinya juga mirip? Bisa saja tidak, maka dari itu menambahkan deteksi deskripsi penelitian (dalam hal ini adalah abstrak) sangat bisa meningkatkan akurasi model dalam mendeteksi plagiarisme atau kemiripan satu penelitian dengan penelitian lain.

Plagiarisme menjadi topik yang sering dibahas dan krusial dilingkungan akademik khususnya di perguruan tinggi [2]. Tindakan menjiplak atau menyalin karya orang lain tanpa memberikan pengakuan yang semestinya tidak hanya melanggar etika akademik, tetapi juga merusak esensi dari proses pendidikan itu sendiri, yaitu mendorong kemampuan berfikir kritis, analisis dan orisinalitas. Ketika plagiarisme dibiarkan, hal ini akan menyebabkan kemunduran dan menurunkan kualitas di lingkungan Pendidikan itu sendiri dan nantinya akan berdampak pada lulusan yang tidak benar-benar menguasai bidang studinya. Dengan kemudahan akses terhadap berbagai sumber informasi digital, peluang terjadinya proses plagiarisme semakin besar. Oleh karena itu penting untuk mengembangkan system deteksi dan pencegahan plagiarisme yang akurat dan komprehensif, termasuk yang dapat mendeteksi tidak hanya dari judul tapi juga dari isi seperti abstrak. Upaya ini menjadi bagian dari komitmen menjaga integritas akademik dan mendorong terciptanya budaya ilmiah yang sehat dan bertanggung jawab.

Beberapa studi sebelumnya telah memanfaatkan pendekatan seperti *TF-IDF (Term Frequency-Inverse Document Frequency)* dan algoritma *Cosine Similarity* untuk mendeteksi tingkat kemiripan antar dokumen [3]. Pendekatan-pendekatan ini telah terbukti dalam mengukur sejauh mana dua dokumen teks memiliki kesamaan berdasarkan representasi vektornya. Namun, dari beberapa referensi yang menjadi rujukan masih terbatas pada analisis judul saja, sementara itu abstrak juga merupakan bagian penting yang berisi informasi esensial yang menggambarkan inti dari penelitian itu sendiri.

2. ANALISIS PROJEK

2.1. KEBAHARUAN PROYEK DARI REFERENSI SEBELUMNYA

Penelitian terdahulu yang menjadi acuan dalam proyek ini hanya berfokus pada deteksi tingkat kemiripan berdasarkan judul penelitian. Hal tersebut dilakukan karena tujuan utamanya adalah untuk mengidentifikasi kemungkinan plagiarisme pada judul semata. Namun, dalam praktiknya, terdapat kasus di mana dua penelitian memiliki judul yang berbeda tetapi isi atau substansi penelitiannya sangat mirip. Kondisi ini dapat terjadi karena beberapa penulis tidak mencantumkan metode atau aspek penting lainnya dalam judul, sehingga kemiripan tidak dapat terdeteksi hanya dengan menganalisis judul. Oleh karena itu, pada proyek ini dilakukan pengembangan dengan menambahkan analisis terhadap deskripsi penelitian (dalam konteks projek kali ini adalah abstrak). Abstrak dipilih karena umumnya memuat gambaran menyeluruh mengenai latar belakang, tujuan, metode, dan hasil penelitian, sehingga memungkinkan sistem mendeteksi tingkat kemiripan secara lebih akurat dan komprehensif.

Pada jurnal rujukan proyek ini, telah dilakukan deteksi kemiripan judul penelitian dengan menggunakan pendekatan TF-IDF dan algoritma Cosine Similarity. Integrasi antara kedua pendekatan dan algoritma tersebut terbukti ampuh dalam mendeteksi kemiripan dua dokumen

berdasarkan representasi vector nya. Pada jurnal rujukan sebelumnya telah didapatkan akurasi sebesar 89.7%.

Berdasarkan pada latar belakang, projek kali ini melakukan pengembangan model deteksi kemiripan tidak hanya berdasarkan judul, tetapi juga mencakup bagian abstrak. Dengan menambahkan analisis pada abstrak, diharapkan hasil deteksi menjadi lebih akurat dan menyeluruh karena mencerminkan baik permukaan (judul) maupun konten inti (abstrak) dari sebuah karya ilmiah. Dataset yang digunakan pada projek ini merupakan data skripsi yang di *scrapping* dari website Repository resmi milik Institut Teknologi Sepeluh Nopember (ITS) yang berjumlah 168 data. Data skripsi yang diambil yaitu skripsi dengan tema *Machine Learning* dan keinformatikaan. Model ini diharapkan dapat menjadi Solusi awal dalam membantu institusi Pendidikan dalam proses evaluasi orisinalitas topik dan mencegah potensi duplikasi atau plagiarisme dalam penyusunan tugas akhir mahasiswa. Kemudian dengan pendekatan dan algoritma yang sama seperti pada jurnal rujukan yaitu TF-IDF dan Cosine Similarity, projek ini juga diharapkan dapat meningkatkan akurasi model dalam menangkap tingkat kemiripan sebuah penelitian.

2.2. ANALISIS KEBUTUHAN PROJEK

2.2.1. Kebutuhan fungsional

Berikut merupakan kebutuhan fungsional yang dibutuhkan projek.

a. Sistem dapat mengubah teks judul dan abstrak menjadi representasi vector menggunakan pendekatan TF-IDF.
b. Sistem dapat menghitung tingkat kemiripan antar dokumen menggunakan Cosine Similarity.
c. Sistem dapat membandingkan judul antar skripsi, abstrak antar skripsi ataupun kombinasi keduanya, dan menampilkan tingkat kemiripan dalam persentase.
d. Sistem dapat menampilkan daftar pasangan antar judul dan abstrak dalam bentuk tabel.

2.2.2. Kebutuhan non fungsional

Berikut merupakan kebutuhan non-fungsional yang dibutuhkan projek.

a. Sistem harus mampu memberikan hasil kemiripan yang representatif dan konsisten berdasarkan data yang telah diproses.
b. Reusability, komponen preprocessing dapat digunakan ulang untuk proses data baru.
c. Maintainability, struktur folder yang rapi dan komentar pada bagian kode yang penting.

2.2.3. Kebutuhan perangkat lunak dan perangkat keras

Berikut merupakan kebutuhan perangkat lunak dan keras yang dibutuhkan proyek.

a. Perangkat lunak
   - Visual Studio Code
   - Python 3.11.9
   - Library: scikit-learn, pandas, numpy, bs4, nltk, sastrawi,
b. Perangkat keras
   - Laptop dengan minimal RAM 4GB
   - Penyimpanan yang cukup untuk menyimpan source-code proyek dan dataset nya.
   - Browser: Chrome atau lainnya.

2.3. DATASET

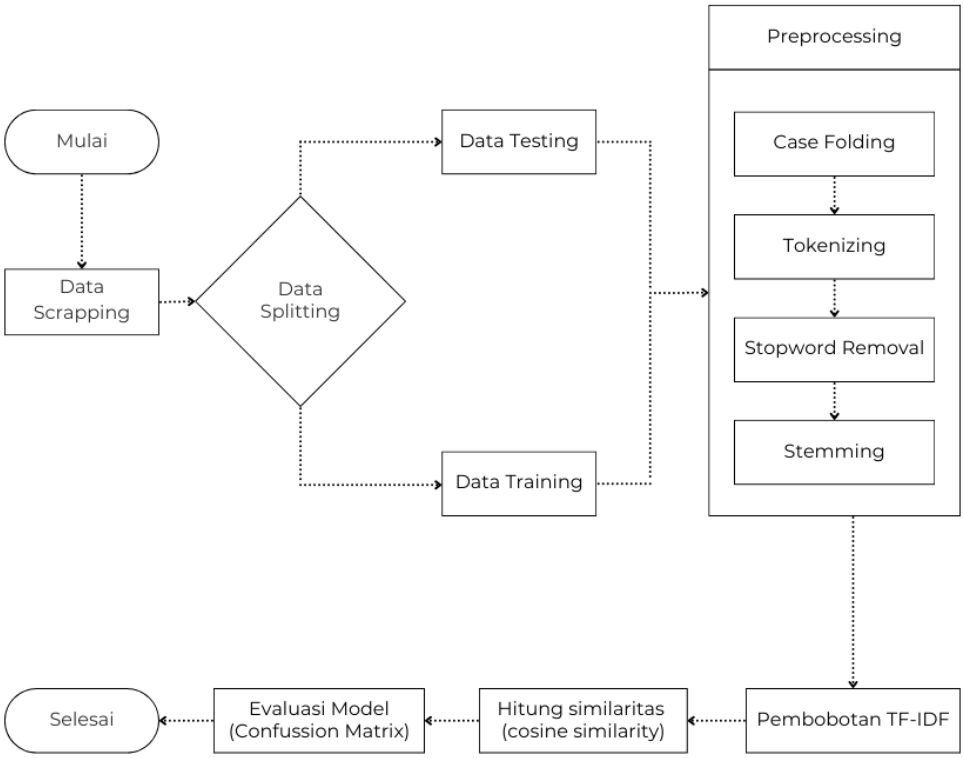
Dataset diambil dari website repository Institut Teknologi Sepuluh Nopember Surabaya (ITS) dengan proses *scrapping* yang memanfaatkan komponen *BeautifulSoup* dari library bs4. Dataset yang diambil merupakan data Judul, penulis, link dan abstrak dari skripsi mahasiswa Institut Teknologi Sepuluh Nopember Surabaya. Proses *scrapping* berhasil mengambil sejumlah 168 data skripsi yang mencakup tema *Machine Learning* dan Keinformatikaan. Data-data yang berhasil di scrap kemudian diubah menjadi file CSV *(Comma Separated Value)*. Dataset dilakukan *splitting* 30/70, 30% untuk data uji dan 70% untuk data latih.

	Judul	Penulis	Link	Abstrak
0	Klasifikasi Aritmia Sinyal ECG Menggunakan Tra...	Gusnam, Mu'thiana	http://repository.its.ac.id/95953/	Pengenalan kelainan aritmia seseorang diketahu...
1	Klasifikasi Jenis Tumor Otak Meningioma, Gliom...	Hajjanto, Ariq Dreiki	http://repository.its.ac.id/110912/	Berdasarkan Global Cancer Statistic pada tahun...
2	Pembuatan Sistem Visual Question Answering Ber...	Hanifah, Asiyah	http://repository.its.ac.id/102392/	Seiring pesatnya perkembangan teknologi, Indon...
3	Analisis Prediksi Faktor Intensitas Tegangan P...	Hardian, Muhammad Akbar	http://repository.its.ac.id/98455/	Berdasarkan data SKK migas pada tahun 2016, 54...
4	Surrogate-Assisted Model Untuk Prediksi Umur K...	Hardian, Muhammad Akbar	http://repository.its.ac.id/108251/	Berdasarkan informasi yang disampaikan dalam p...

Gambar 1, Head Dataframe dari Dataset

3. PEMODELAN/SISTEM/APLIKASI

3.1. ILUSTRASI ATAU ARSITEKTUR PROJEK



Gambar 2, Arsitektur proyek

Gambar 2 merupakan ilustrasi atau arsitektur projek, projek dimulai dengan proses pencarian dataset sebuah penelitian dan mengambil data judul dan abstrak, dataset berhasil didapatkan dengan teknik scrapping dari website repository Institut Teknologi Sepuluh Nopember. Tahap selanjutnya dataset dibagi menjadi data testing dan data training dengan proporsi 30/70. Kedua

dataset selanjutnya dilakukan preprocessing dengan tahapan case folding, tokenizing, stopword removal dan stemming. Setelah dataset di preprocessing, kemudian dilakukan proses pembobotan kata menjadi representasi vector dengan menggunakan pendekatan TF-IDF, setelah melalui pembobotan langkah selanjutnya adalah menghitung tingkat similaritas antar vector kata dengan menggunakan algoritma Cosine Similarity, kemudian langkah terakhir yaitu evaluasi model dengan menggunakan confussion matrix.

3.2. TAHAPAN

3.2.1. Data gathering

Dataset diambil dari repository Institut Teknologi Sepuluh Nopember menggunakan Teknik scrapping dengan memanfaatkan komponen *BeautifulSoap* dari library python bs4. Data yang didapatkan berupa Judul, penulis, link dan abstrak dari penelitian atau skripsi dan 168 data berhasil di scrap. Berikut proses data scrapping-nya.

1. Mengambil judul

```
# Cari semua link
for a_tag in soup.find_all('a', href=True):
    href = a_tag.get('href')
    em_tag = a_tag.find('em')

    if not em_tag:
        continue  # Hanya proses link yang ada <em> (berarti link ke judul)

    if href.startswith('/'):
        link = base_url + href
    elif href.startswith('http'):
        link = href
    else:
        continue  # Skip kalau href aneh

    title = em_tag.text.strip()  # Ambil text dari <em>
```

Gambar 3, Dokumentasi program

Karena pada website ini judul nya memiliki atribut link atau ‘href’ dan tag nya adalah ‘em’ maka program akan mencoba mencari komponen html ber-atribut ‘href’ dengan tag ‘em’ dan menyimpannya dalam variable ‘title’.

2. Mengambil penulis, link dan abstrak

```python
    # Masuk ke halaman detail
    try:
        detail_response = requests.get(link, headers=headers)
        detail_soup = BeautifulSoup(detail_response.content, 'html.parser')

        # Ambil penulis
        author_tag = detail_soup.find('span', class_='person_name')
        author = author_tag.text.strip() if author_tag else 'Tidak ditemukan'

        # Ambil abstrak
        abstract_tag = detail_soup.find('p', class_='ep_field_para')
        abstract = abstract_tag.text.strip() if abstract_tag else 'Tidak ditemukan'

        # Simpan hasil
        data.append({
            'Judul': title,
            'Link': link,
            'Penulis': author,
            'Abstrak': abstract
        })

        print(f"Sukses ambil: {title}")  # Status biar kelihatan progres

        time.sleep(1)  # Delay biar aman

    except Exception as e:
        print(f"Error saat ambil detail {link}: {e}")
        continue

df = pd.DataFrame(data)
```

Gambar 4, Dokumentasi program

Kemudian dari link tersebut, program akan mencoba masuk ke halaman detail untuk mengambil informasi lebih rinci yaitu penulis, link dan juga abstrak kemudian menyimpannya pada variable author untuk penulis dan abstract untuk abstrak. Kemudian setelah keempat variable tersebut telah didapatkan maka tahap selanjutnya adalah menyimpannya dalam bentuk data frame.

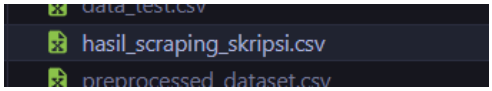
3. Menyimpan data frame dalam bentuk csv

```python
# Simpan ke CSV
df.to_csv('hasil_scraping_skripsi.csv', index=False, columns=['Judul', 'Penulis', 'Link', 'Abstrak'])
df = df[['Judul', 'Penulis', 'Link', 'Abstrak']]

print("✅ Data berhasil disimpan ke 'hasil_scraping_skripsi.csv'!")
# Cetak hasil
for item in data:
    print(f"Judul: {item['Judul']}")
    print(f"Link: {item['Link']}")
    print(f"Penulis: {item['Penulis']}")
    print(f"Abstrak: {item['Abstrak']}")
    print('---')
```

Gambar 5, Dokumentasi program

Setelah variabel-variabel berhasil disimpan dalam bentuk dataframe, maka tahap selanjutnya dataframe disimpan dalam bentuk csv.



Gambar 6, File hasil scrapping

3.2.2. Preprocessing

Preprocessing merupakan proses untuk menyiapkan data agar siap untuk diproses di tahap selanjutnya [4]. Pada projek kali ini, preprocessing melalui 4 tahapan yaitu *Case Folding, Tokenizing, Stopword Removal,* dan *Stemming*.

a. *Case Folding*

Case Folding merupakan tahapan mengubah semua huruf dalam dokumen menjadi kecil atau besar (biasanya kecil) secara seragam dan menghapus □ystem. Proses ini dapat bermanfaat untuk mengurangi jumlah token *unique* pada sebuah dokumen sehingga membuat pemrosesan roken menjadi lebih hemat memori dan komputasi.

```
#Case folding, mengubah semua huruf menjadi kecil dan menghapus simbol.
def case_folding(text):
    text = text.lower()

    text = text.translate(str.maketrans('', '', string.punctuation))

    return text

df['casefolding_judul'] = df['Judul'].apply(case_folding)
df['casefolding_abstrak'] = df['Abstrak'].apply(case_folding)


#df.to_csv('casefolded_dataset.csv', index=False, encoding='utf-8-sig')
```

Gambar 7, Dokumentasi program

Casefolding	
Judul	**Abstrak**
klasifikasi aritmia sinyal ecg menggunakan transformasi wavelet dan □ystem□ statistic	pengenalan kelainan aritmia seseorang diketahui dengan melakukan rekam aktivitas jantung menggunakan electrocardiogram ecg rekaman ecg detak jantung dibagi menjadi gelombang p qrs dan t yang menunjukkan aktivitas kelistrikan jantung seperti depolarisasi atrium dari gelombang p depolarisasi ventrikel dari kompleks qrs dan repolarisasi ventrikel maupun atrium dari segmen st
klasifikasi jenis tumor otak meningioma glioma dan □ystem□ia berbasis hybrid vgg16 dan svm	berdasarkan global cancer statistic pada tahun 2020 kasus baru tumor otak dan cns mencapai 308102 dengan kematian mencapai 251329 di seluruh dunia di □ystem□ia sendiri estimasi kejadian dan kematian pada tahun 2016 mencapai 6337 dan 5405 kasus banyak jenis tumor otak dengan variasi □ystem ukuran dan tingkat keganasan membuat melokalisasi dan klasifikasi tumor kompleks bagi ahli medis secara konvensional menyebabkan kesalahan dalam

untuk diagnosis praoperasi	penentuan jenis tumor otak karena perlu membaca hasil citra dalam jumlah yang sangat banyak keakuratan klasifikasi konvensional dapat dipengaruhi oleh beberapa □ystem seperti perbedaan subjektivitas individu dalam mengenali □ystem tumor waktu ketelitian kelelahan dan □ystem manusia lainnya maka dari itu dibutuhkan suatu metode dalam menghasilkan diagnosis tumor yang akurat dengan machine learning akan tetapi penelitian yang menggunakan pendekatan machine learning sangat rentan akan overfitting disebabkan kurangnya dataset ataupun model arsitektur yang digunakan dan juga lamanya proses komputasi yang dibutuhkan oleh sebab itu pada penelitian ini diusulkan □ystem klasifikasi hibrida dengan bantuan machine learning yaitu menggunakan arsitekturmodel vgg16 dan support vector machine svm vgg16 memiliki keunggulan dalam ekstraksi fitur hierarkis dan invariansi spasial yang memungkinkan identifikasi tumor dengan akurasi lebih tinggi output fitur jenis tumor otak dari vgg16 direduksi menggunakan principal component analysis pca lalu diklasifikasi dengan bantuan svm serta dioptimalkan dengan pengujian kombinasi kernel dan hyperparameter performa arsitektur dievaluasi menggunakan performance metrics dan komparasi model sebelumnya yang memungkinkan penilaian objektif terhadap hasil yang dicapai hasil penelitian memberikan hasil untuk masingmasing metrik akurasi presisi recall¸f1score dan spesifisitas secara berturutturut sebesar 969 973 9667 9667 dan 9997 dengan menggunakan kernel polynomial dengan hyperparameter c degree dan coef0 sebesar 10 3 dan 05.
Pembuatan □ystem visual question answering berbasis web untuk mendukung pembelajaran visual anak tk berbahasa □ystem□ia menggunakan deep learning	seiring pesatnya perkembangan teknologi □ystem□ia semakin gencar melakukan persiapan transformasi digital untuk menghadapi perubahan teknologi salah satunya adalah implementasi elearning di berbagai sektor termasuk □ystem□ia□ elearning telah diterapkan dalam pembelajaran taman kanakkanak termasuk pembelajaran visual bentuk pembelajaran visual pada elearning dapat dibuat dengan □ystem□ia□n □ystem visual question answering beberapa penelitian telah dibuat untuk □ystem□ia□n □ystem visual question answering dan berhasil membuat □ystem visual question answering dengan ilmu patologi dalam □ystem inggris dan dataset objek di sekitar monas dalam □ystem □ystem□ia oleh karena itu dilakukan pengajuan pembuatan □ystem visual question answering dengan dataset yang lebih umum dan dapat dikenali oleh anak tk berbahasa □ystem□ia dadanya penelitian ini akan dapat membantu tenaga pendidik dalam kegiatan belajar mengajar yang lebih interaktif dalam elearning penelitian ini menggunakan model bootstrapping languageimage pretraining blip untuk proses pembuatan □ystem visual question answering dan mengimplementasikan model no language left behind nllb pada inputoutput pertanyaan untuk menerjemahkan □ystem yang digunakan hasil implementasi kedua model blip dan nllb berhasil membangun system visual question answering berbahasa ystem□ia berdasarkan hasil pengujiannya dari beberapa pertanyaan yang

	mengandung 6 jenis jawaban yatidak kata benda kata kerja kata sifat kata keterangan dan numeral ystem ini berhasil menjawab tepat untuk jenis jawaban yatidak kata benda kata kerja dan kata keterangan dengan nilai ketepatan jawaban yatidak 100 kata benda 100 kata kerja 100 dan kata keterangan 875

Tabel 1, Case Folding

b. *Tokenizing*

Tokenizing merupakan proses untuk memecah serangkaian teks dan mengubah kata-kata yang telah dipecah menjadi sebuah token. Proses ini bertujuan agar teks yang semula berupa satu string panjang bisa dianalisis secara komputasional dalam bentuk satuan yang lebih kecil yaitu kata-kata.

```
#Tokenizing, tahap untuk memisahkan atau memecah teks menjadi bagian-bagian kata yang disebut token.
def tokenizing(text):
    tokens = word_tokenize(text)

    return tokens

df['tokenizing_judul'] = df['casefolding_judul'].apply(tokenizing)
df['tokenizing_abstrak'] = df['casefolding_abstrak'].apply(tokenizing)

df.to_csv('tokenized_dataset.csv', index=False, encoding='utf-8-sig')
```

Gambar 8, Dokumentasi program

Tokenizing	
Judul	Abstrak
['klasifikasi', 'aritmia', 'sinyal', 'ecg', 'menggunakan', 'transformasi', 'wavelet', 'dan', 'analisa', 'statistik']	['pengenalan', 'kelainan', 'aritmia', 'seseorang', 'diketahui', 'dengan', 'melakukan', 'rekam', 'aktivitas', 'jantung', 'menggunakan', 'electrocardiogram', 'ecg', 'rekaman', 'ecg', 'detak', 'jantung', 'dibagi', 'menjadi', 'gelombang', 'p', 'qrs', 'dan', 't', 'yang', 'menunjukkan', 'aktivitas', 'kelistrikan', 'jantung', 'seperti', 'depolarisasi', 'atrium', 'dari', 'gelombang', 'p', 'depolarisasi', 'ventrikel', 'dari', 'kompleks', 'qrs', 'dan', 'repolarisasi', 'ventrikel', 'maupun', 'atrium', 'dari', 'segmen', 'st']
['klasifikasi', 'jenis', 'tumor', 'otak', 'meningioma', 'glioma', 'dan', 'pituitari', 'berbasis', 'hybrid', 'vgg16', 'dan', 'svm', 'untuk', 'diagnosis', 'praoperasi']	['berdasarkan', 'global', 'cancer', 'statistic', 'pada', 'tahun', '2020', 'kasus', 'baru', 'tumor', 'otak', 'dan', 'cns', 'mencapai', '308102', 'dengan', 'kematian', 'mencapai', '251329', 'di', 'seluruh', 'dunia', 'di', 'indonesia', 'sendiri', 'estimasi', 'kejadian', 'dan', 'kematian', 'pada', 'tahun', '2016', 'mencapai', '6337', 'dan', '5405', 'kasus', 'banyak', 'jenis', 'tumor', 'otak', 'dengan', 'variasi', 'lokasi', 'ukuran', 'dan', 'tingkat', 'keganasan', 'membuat', 'melokalisasi', 'dan', 'klasifikasi', 'tumor', 'kompleks', 'bagi', 'ahli', 'medis', 'secara', 'konvensional', 'menyebabkan', 'kesalahan', 'dalam', 'penentuan', 'jenis', 'tumor', 'otak', 'karena', 'perlu', 'membaca', 'hasil', 'citra', 'dalam', 'jumlah', 'yang', 'sangat', 'banyak', 'keakuratan', 'klasifikasi', 'konvensional', 'dapat', 'dipengaruhi', 'oleh', 'beberapa', 'faktor', 'seperti', 'perbedaan', 'subjektivitas', 'individu', 'dalam', 'mengenali', 'lokasi', 'tumor', 'waktu', 'ketelitian', 'kelelahan', 'dan',

	'faktor', 'manusia', 'lainnya', 'maka', 'dari', 'itu', 'dibutuhkan', 'suatu', 'metode', 'dalam', 'menghasilkan', 'diagnosis', 'tumor', 'yang', 'akurat', 'dengan', 'machine', 'learning', 'akan', 'tetapi', 'penelitian', 'yang', 'menggunakan', 'pendekatan', 'machine', 'learning', 'sangat', 'rentan', 'akan', 'overfitting', 'disebabkan', 'kurangnya', 'dataset', 'ataupun', 'model', 'arsitektur', 'yang', 'digunakan', 'dan', 'juga', 'lamanya', 'proses', 'komputasi', 'yang', 'dibutuhkan', 'oleh', 'sebab', 'itu', 'pada', 'penelitian', 'ini', 'diusulkan', 'sistem', 'klasifikasi', 'hibrida', 'dengan', 'bantuan', 'machine', 'learning', 'yaitu', 'menggunakan', 'arsitekturmodel', 'vgg16', 'dan', 'support', 'vector', 'machine', 'svm', 'vgg16', 'memiliki', 'keunggulan', 'dalam', 'ekstraksi', 'fitur', 'hierarkis', 'dan', 'invariansi', 'spasial', 'yang', 'memungkinkan', 'identifikasi', 'tumor', 'dengan', 'akurasi', 'lebih', 'tinggi', 'output', 'fitur', 'jenis', 'tumor', 'otak', 'dari', 'vgg16', 'direduksi', 'menggunakan', 'principal', 'component', 'analysis', 'pca', 'lalu', 'diklasifikasi', 'dengan', 'bantuan', 'svm', 'serta', 'dioptimalkan', 'dengan', 'pengujian', 'kombinasi', 'kernel', 'dan', 'hyperparameter', 'performa', 'arsitektur', 'dievaluasi', 'menggunakan', 'performance', 'metrics', 'dan', 'komparasi', 'model', 'sebelumnya', 'yang', 'memungkinkan', 'penilaian', 'objektif', 'terhadap', 'hasil', 'yang', 'dicapai', 'hasil', 'penelitian', 'memberikan', 'hasil', 'untuk', 'masingmasing', 'metrik', 'akurasi', 'presisi', 'recall¸f1score', 'dan', 'spesifisitas', 'secara', 'berturutturut', 'sebesar', '969', '973', '9667', '9667', 'dan', '9997', 'dengan', 'menggunakan', 'kernel', 'polynomial', 'dengan', 'hyperparameter', 'c', 'degree', 'dan', 'coef0', 'sebesar', '10', '3', 'dan']
['pembuatan', 'sistem', 'visual', 'question', 'answering', 'berbasis', 'web', 'untuk', 'mendukung', 'pembelajaran', 'visual', 'anak', 'tk', 'berbahasa', 'indonesia', 'menggunakan', 'deep', 'learning']	['seiring', 'pesatnya', 'perkembangan', 'teknologi', 'indonesia', 'semakin', 'gencar', 'melakukan', 'persiapan', 'transformasi', 'digital', 'untuk', 'menghadapi', 'perubahan', 'teknologi', 'salah', 'satunya', 'adalah', 'implementasi', 'elearning', 'di', 'berbagai', 'sektor', 'termasuk', 'pendidikan', 'elearning', 'telah', 'diterapkan', 'dalam', 'pembelajaran', 'taman', 'kanakkanak', 'termasuk', 'pembelajaran', 'visual', 'bentuk', 'pembelajaran', 'visual', 'pada', 'elearning', 'dapat', 'dibuat', 'dengan', 'pembangunan', 'sistem', 'visual', 'question', 'answering', 'beberapa', 'penelitian', 'telah', 'dibuat', 'untuk', 'pembangunan', 'sistem', 'visual', 'question', 'answering', 'dan', 'berhasil', 'membuat', 'sistem', 'visual', 'question', 'answering', 'dengan', 'ilmu', 'patologi', 'dalam', 'bahasa', 'inggris', 'dan', 'dataset', 'objek', 'di', 'sekitar', 'monas', 'dalam', 'bahasa', 'indonesia', 'oleh', 'karena', 'itu', 'dilakukan', 'pengajuan', 'pembuatan', 'sistem', 'visual', 'question', 'answering', 'dengan', 'dataset', 'yang', 'lebih', 'umum', 'dan', 'dapat', 'dikenali', 'oleh', 'anak', 'tk', 'berbahasa', 'indonesia', 'dadanya', 'penelitian', 'ini', 'akan', 'dapat', 'membantu', 'tenaga', 'pendidik', 'dalam', 'kegiatan', 'belajar', 'mengajar', 'yang', 'lebih', 'interaktif', 'dalam', 'elearning', 'penelitian', 'ini', 'menggunakan', 'model', 'bootstrapping', 'languageimage', 'pretraining', 'blip', 'untuk', 'proses', 'pembuatan', 'sistem', 'visual', 'question', 'answering', 'dan', 'mengimplementasikan', 'model', 'no', 'language', 'left', 'behind', 'nllb', 'pada', 'inputoutput', 'pertanyaan', 'untuk', 'menerjemahkan', 'bahasa', 'yang', 'digunakan', 'hasil', 'implementasi', 'kedua', 'model', 'blip', 'dan', 'nllb', 'berhasil', 'membangun', 'sistem', 'visual', 'question', 'answering', 'berbahasa', 'indonesia', 'berdasarkan', 'hasil', 'pengujiannya', 'dari',

	'beberapa', 'pertanyaan', 'yang', 'mengandung', '6', 'jenis', 'jawaban', 'yatidak', 'kata', 'benda', 'kata', 'kerja', 'kata', 'sifat', 'kata', 'keterangan', 'dan', 'numeral', 'sistem', 'ini', 'berhasil', 'menjawab', 'tepat', 'untuk', 'jenis', 'jawaban', 'yatidak', 'kata', 'benda', 'kata', 'kerja', 'dan', 'kata', 'keterangan', 'dengan', 'nilai', 'ketepatan', 'jawaban', 'yatidak', '100', 'kata', 'benda', '100', 'kata', 'kerja', '100', 'dan', 'kata', 'keterangan']

Tabel 2, Tokenizing

c. *Stopword Removal*

Tahap *stopword removal* adalah tahap menghapus kata yang tidak relevan atau tidak memiliki makna didalam suatu kalimat berdasarkan daftar stopword. Karena *stopword* biasanya tidak menambah makna penting pada dokumen, maka menghapusnya akan membantu model untuk fokus pada kata-kata yang lebih bermakna seperti “deteksi”, “algoritma”, “skripsi” dan lain-lain.

```
#Stopword removal, tahap menghapus kata yang tidak relevan didalam suatu kalimat berdasarkan daftar stopword.
def stopword_removal(text):
    tokens = word_tokenize(text)
    tokens = [word for word in tokens if word not in stop_words]

    return tokens

df['stopword_removal_judul'] = df['casefolding_judul'].apply(stopword_removal)
df['stopword_removal_abstrak'] = df['casefolding_abstrak'].apply(stopword_removal)


df.to_csv('stopword_removed_dataset.csv', index=False, encoding='utf-8-sig')
```

Gambar 9, Dokumentasi program

Stopword Removal	
Judul	Abstrak
['klasifikasi', 'aritmia', 'sinyal', 'ecg', 'transformasi', 'wavelet', 'analisa', 'statistik']	['pengenalan', 'kelainan', 'aritmia', 'rekam', 'aktivitas', 'jantung', 'electrocardiogram', 'ecg', 'rekaman', 'ecg', 'detak', 'jantung', 'dibagi', 'gelombang', 'p', 'qrs', 't', 'aktivitas', 'kelistrikan', 'jantung', 'depolarisasi', 'atrium', 'gelombang', 'p', 'depolarisasi', 'ventrikel', 'kompleks', 'qrs', 'repolarisasi', 'ventrikel', 'atrium', 'segmen', 'st']
['klasifikasi', 'jenis', 'tumor', 'otak', 'meningioma', 'glioma', 'pituitari', 'berbasis', 'hybrid', 'vgg16', 'svm', 'diagnosis', 'praoperasi']	['berdasarkan', 'global', 'cancer', 'statistic', '2020', 'tumor', 'otak', 'cns', 'mencapai', '308102', 'kematian', 'mencapai', '251329', 'dunia', 'indonesia', 'estimasi', 'kejadian', 'kematian', '2016', 'mencapai', '6337', '5405', 'jenis', 'tumor', 'otak', 'variasi', 'lokasi', 'ukuran', 'tingkat', 'keganasan', 'melokalisasi', 'klasifikasi', 'tumor', 'kompleks', 'ahli', 'medis', 'konvensional', 'menyebabkan', 'kesalahan', 'penentuan', 'jenis', 'tumor', 'otak', 'membaca', 'hasil', 'citra', 'keakuratan', 'klasifikasi', 'konvensional', 'dipengaruhi', 'faktor', 'perbedaan', 'subjektivitas', 'individu', 'mengenali', 'lokasi', 'tumor', 'ketelitian', 'kelelahan', 'faktor', 'manusia', 'dibutuhkan', 'metode', 'menghasilkan', 'diagnosis', 'tumor', 'akurat', 'machine', 'learning', 'penelitian', 'pendekatan', 'machine', 'learning', 'rentan', 'overfitting', 'disebabkan', 'kurangnya', 'dataset', 'model', 'arsitektur', 'proses', 'komputasi', 'dibutuhkan', 'penelitian',

	'diusulkan', 'sistem', 'klasifikasi', 'hibrida', 'bantuan', 'machine', 'learning', 'arsitekturmodel', 'vgg16', 'support', 'vector', 'machine', 'svm', 'vgg16', 'memiliki', 'keunggulan', 'ekstraksi', 'fitur', 'hierarkis', 'invariansi', 'spasial', 'identifikasi', 'tumor', 'akurasi', 'output', 'fitur', 'jenis', 'tumor', 'otak', 'vgg16', 'direduksi', 'principal', 'component', 'analysis', 'pca', 'diklasifikasi', 'bantuan', 'svm', 'dioptimalkan', 'pengujian', 'kombinasi', 'kernel', 'hyperparameter', 'performa', 'arsitektur', 'dievaluasi', 'performance', 'metrics', 'komparasi', 'model', 'penilaian', 'objektif', 'hasil', 'dicapai', 'hasil', 'penelitian', 'hasil', 'masingmasing', 'metrik', 'akurasi', 'presisi', 'recall¸f1score', 'spesifisitas', 'berturutturut', '969', '973', '9667', '9667', '9997', 'kernel', 'polynomial', 'hyperparameter', 'c', 'degree', 'coef0', '10', '3', '05']
['pembuatan', 'sistem', 'visual', 'question', 'answering', 'berbasis', 'web', 'mendukung', 'pembelajaran', 'visual', 'anak', 'tk', 'berbahasa', 'indonesia', 'deep', 'learning']	['seiring', 'pesatnya', 'perkembangan', 'teknologi', 'indonesia', 'gencar', 'persiapan', 'transformasi', 'digital', 'menghadapi', 'perubahan', 'teknologi', 'salah', 'satunya', 'implementasi', 'elearning', 'sektor', 'pendidikan', 'elearning', 'diterapkan', 'pembelajaran', 'taman', 'kanakkanak', 'pembelajaran', 'visual', 'bentuk', 'pembelajaran', 'visual', 'elearning', 'pembangunan', 'sistem', 'visual', 'question', 'answering', 'penelitian', 'pembangunan', 'sistem', 'visual', 'question', 'answering', 'berhasil', 'sistem', 'visual', 'question', 'answering', 'ilmu', 'patologi', 'bahasa', 'inggris', 'dataset', 'objek', 'monas', 'bahasa', 'indonesia', 'pengajuan', 'pembuatan', 'sistem', 'visual', 'question', 'answering', 'dataset', 'dikenali', 'anak', 'tk', 'berbahasa', 'indonesia', 'dadanya', 'penelitian', 'membantu', 'tenaga', 'pendidik', 'kegiatan', 'belajar', 'mengajar', 'interaktif', 'elearning', 'penelitian', 'model', 'bootstrapping', 'languageimage', 'pretraining', 'blip', 'proses', 'pembuatan', 'sistem', 'visual', 'question', 'answering', 'mengimplementasikan', 'model', 'no', 'language', 'left', 'behind', 'nllb', 'inputoutput', 'menerjemahkan', 'bahasa', 'hasil', 'implementasi', 'model', 'blip', 'nllb', 'berhasil', 'membangun', 'sistem', 'visual', 'question', 'answering', 'berbahasa', 'indonesia', 'berdasarkan', 'hasil', 'pengujiannya', 'mengandung', '6', 'jenis', 'yatidak', 'benda', 'kerja', 'sifat', 'keterangan', 'numeral', 'sistem', 'berhasil', 'jenis', 'yatidak', 'benda', 'kerja', 'keterangan', 'nilai', 'ketepatan', 'yatidak', '100', 'benda', '100', 'kerja', '100', 'keterangan', '875']

Tabel 3, Stopword Removal

d. Stemming

Proses *stemming* merupakan proses mengubah kata ke bentuk dasarnya atau akarnya (stem), dengan cara menghapus imbuhan seperti awalan, akhiran, sisipan atau gabungan lainnya. Misal "berlari", "berlarian", "lari-lari", "pelari" semuanya akan dikembalikan ke bentuk dasar: "lari", proses ini bertujuan untuk membantu sistem menyadari bahwa semua bentuk tersebut mempunyai makna yang serupa yaitu "lari".

```python
#Stemming, proses yang sangat penting untuk mencari kata dasar dari sebuah kata derivatif.
def stemming(text):
    tokens = word_tokenize(text)
    tokens = [word for word in tokens if word not in stop_words]
    tokens = [stemmer.stem(word) for word in tokens]

    return tokens

df['stemming_judul'] = df['casefolding_judul'].apply(stemming)
df['stemming_abstrak'] = df['casefolding_abstrak'].apply(stemming)


df.to_csv('stemmed_dataset.csv', index=False, encoding='utf-8-sig')
```

Gambar 10, Dokumentasi program

Stemming	
Judul	Abstrak
['klasifikasi', 'aritmia', 'sinyal', 'ecg', 'transformasi', 'wavelet', 'analisa', 'statistik']	['kenal', 'lain', 'aritmia', 'rekam', 'aktivitas', 'jantung', 'electrocardiogram', 'ecg', 'rekam', 'ecg', 'detak', 'jantung', 'bagi', 'gelombang', 'p', 'qrs', 't', 'aktivitas', 'listrik', 'jantung', 'depolarisasi', 'atrium', 'gelombang', 'p', 'depolarisasi', 'ventrikel', 'kompleks', 'qrs', 'repolarisasi', 'ventrikel', 'atrium', 'segmen', 'st']
['klasifikasi', 'jenis', 'tumor', 'otak', 'meningioma', 'glioma', 'pituitari', 'bas', 'hybrid', 'vgg16', 'svm', 'diagnosis', 'praoperasi']	['dasar', 'global', 'cancer', 'statistic', '2020', 'tumor', 'otak', 'cns', 'capai', '308102', 'mati', 'capai', '251329', 'dunia', 'indonesia', 'estimasi', 'jadi', 'mati', '2016', 'capai', '6337', '5405', 'jenis', 'tumor', 'otak', 'variasi', 'lokasi', 'ukur', 'tingkat', 'ganas', 'lokalisasi', 'klasifikasi', 'tumor', 'kompleks', 'ahli', 'medis', 'konvensional', 'sebab', 'salah', 'tentu', 'jenis', 'tumor', 'otak', 'baca', 'hasil', 'citra', 'akurat', 'klasifikasi', 'konvensional', 'pengaruh', 'faktor', 'beda', 'subjektivitas', 'individu', 'nali', 'lokasi', 'tumor', 'teliti', 'lelah', 'faktor', 'manusia', 'butuh', 'metode', 'hasil', 'diagnosis', 'tumor', 'akurat', 'machine', 'learning', 'teliti', 'dekat', 'machine', 'learning', 'rentan', 'overfitting', 'sebab', 'kurang', 'dataset', 'model', 'arsitektur', 'proses', 'komputasi', 'butuh', 'teliti', 'usul', 'sistem', 'klasifikasi', 'hibrida', 'bantu', 'machine', 'learning', 'arsitekturmodel', 'vgg16', 'support', 'vector', 'machine', 'svm', 'vgg16', 'milik', 'unggul', 'ekstraksi', 'fitur', 'hierarkis', 'invariansi', 'spasial', 'identifikasi', 'tumor', 'akurasi', 'output', 'fitur', 'jenis', 'tumor', 'otak', 'vgg16', 'reduksi', 'principal', 'component', 'analysis', 'pca', 'klasifikasi', 'bantu', 'svm', 'optimal', 'uji', 'kombinasi', 'kernel', 'hyperparameter', 'performa', 'arsitektur', 'evaluasi', 'performance', 'metrics', 'komparasi', 'model', 'nilai', 'objektif', 'hasil', 'capai', 'hasil', 'teliti', 'hasil', 'masingmasing', 'metrik', 'akurasi', 'presisi', 'recall f1score', 'spesifisitas', 'berturutturut', '969', '973', '9667', '9667', '9997', 'kernel', 'polynomial', 'hyperparameter', 'c', 'degree', 'coef0', '10', '3', '05’
['buat', 'sistem', 'visual', 'question', 'answering',	['iring', 'pesat', 'kembang', 'teknologi', 'indonesia', 'gencar', 'siap', 'transformasi', 'digital', 'hadap', 'ubah', 'teknologi', 'salah', 'satu', 'implementasi', 'elearning', 'sektor', 'didik', 'elearning', 'terap', 'ajar', 'taman', 'kanakkanak', 'ajar', 'visual', 'bentuk', 'ajar', 'visual', 'elearning', 'bangun',

'bas', 'web', 'dukung', 'ajar', 'visual', 'anak', 'tk', 'bahasa', 'indonesia', 'deep', 'learning']	'sistem', 'visual', 'question', 'answering', 'teliti', 'bangun', 'sistem', 'visual', 'question', 'answering', 'hasil', 'sistem', 'visual', 'question', 'answering', 'ilmu', 'patologi', 'bahasa', 'inggris', 'dataset', 'objek', 'monas', 'bahasa', 'indonesia', 'aju', 'buat', 'sistem', 'visual', 'question', 'answering', 'dataset', 'nali', 'anak', 'tk', 'bahasa', 'indonesia', 'dada', 'teliti', 'bantu', 'tenaga', 'didik', 'giat', 'ajar', 'ajar', 'interaktif', 'elearning', 'teliti', 'model', 'bootstrapping', 'languageimage', 'pretraining', 'blip', 'proses', 'buat', 'sistem', 'visual', 'question', 'answering', 'implementasi', 'model', 'no', 'language', 'left', 'behind', 'nllb', 'inputoutput', 'terjemah', 'bahasa', 'hasil', 'implementasi', 'model', 'blip', 'nllb', 'hasil', 'bangun', 'sistem', 'visual', 'question', 'answering', 'bahasa', 'indonesia', 'dasar', 'hasil', 'uji', 'kandung', '6', 'jenis', 'yatidak', 'benda', 'kerja', 'sifat', 'terang', 'numeral', 'sistem', 'hasil', 'jenis', 'yatidak', 'benda', 'kerja', 'terang', 'nilai', 'tepat', 'yatidak', '100', 'benda', '100', 'kerja', '100', 'terang', '875'

Tabel 4, Stemming

3.2.3. Implementasi TF-IDF

TF-IDF merupakan suatu proses untuk melakukan transformasi data dari data tekstual menjadi data numerik atau vector kata untuk dilakukan pembobotan pada tiap kata. *TF-IDF* ini adalah sebuah ukuran statistik yang digunakan untuk mengevaluasi seberapa penting sebuah kata di dalam sebuah dokumen [5]. Tujuan utama dari proses ini adalah menonjolkan kata-kata penting dalam dokumen tertentu, mengabaikan kata-kata umum yang sering muncul di banyak dokumen dan menghasilkan representasi vector bobot kata yang nantinya bisa digunakan untuk mengukur kemiripan antar dokumen, dalam konteks projek kali ini menggunakan *Cosine Similarity.*

a. *Term Frequency (TF)*

Singkatnya *Term Frequency (TF)* adalah seberapa sering kata muncul dalam satu dokumen yang bertujuan untuk mengetahui seberapa penting kata tersebut di dalam sebuah dokumen.

Dokumen Judul	Term (Kata)	Term Frequency
Pengembangan Smart Meter untuk Mendukung Home Energy Management System (HEMS) Mempertimbangkan Kualitas Daya Peralatan Rumah.	Listrik	8
	Rumah	6
	Energi	6
	Beban	5
	Daya	4
Integrasi Servqual Dan Quality Function Deployment Sebagai Upaya Peningkatan Pelayanan Minimarket.	Minimarket	8
	Tingkat	7
	Langgan	6
	Layan	5
	Bobot	4

Tabel 5, Term Frequency

b. *Document Frequency*

Singkatnya, *Document Frequency* adalah berapa banyak dokumen yang mengandung kata tertentu yang bertujuan untuk mengetahui seberapa umum kata tersebut.

Term (t)	Doc(d)	DF
svm	d1, d3, d5, d8, d11, d19, …	34
regresi	d12, d15, d21, d34, d39, …	8
klasifikasi	d0, d1, d5, d8, d9, d10, …	55
random	d11, d39, d40, d46, d73, …	12
machine	d1, d3, d4, d5, d8, d10, …	47

Tabel 6, Document Frequency

c. *Inverse Document Frequency (IDF)*

Singkatnya juga, *Inverse Document Frequency (IDF)* adalah *inverse* atau kebalikan dari *Document Frequency (DF).* Makin jarang muncul di banyak dokumen, makin tinggi nilai IDF.

Term (t)	Df	Idf
svm	34	2,223
regresi	8	3,581
klasifikasi	55	1,743
random	12	3,214
machine	47	1,907

Tabel 7, Inverse Document Frequency

3.2.4. Implementasi *Cosine Similarity*

Setelah bobot kata didapatkan dengan menggunakan perhitungan *TF-IDF* maka tahap selanjutnya adalah menghitung kemiripan antar dokumen dengan menggunakan *Cosine Similarity. Cosine Similarity* merupakan metode untuk mengukur tingkat kesamaan atau kemiripan antara dua objek (biasanya dokumen atau teks) yang direpresentasikan sebagai vektor. *Cosine Similarity* mengukur sudut *(cosine)* antara dua vector. Nilainya berada di antara:

- 1, jika dua vector identic
- 0, jika dua vector tidak mirip sama sekali
- -1, jika dua vector berlawanan arah (jarang dipakai dalam konteks teks karena vector biasanya non-negatif).

Judul tes	Judul train	Similarity Score

Sistem Prediksi Status Stunting dan Severe Stunting Menggunakan Multinomial Logistic Regression pada Anak di Indonesia.	Bayisehatkita: Aplikasi Berbasis Web Untuk Klasifikasi Stunting Pada Aud.	0,7604
Implementasi Segmentasi Pakaian Menggunakan Metode Mask R-CNN.	Klasifikasi Tumor Otak Pada Citra MRI Menggunakan en-CNN.	0,3413
Aplikasi Metode Support Vector Machine (SVM) untuk Klasifikasi Sentimen Masyarakat Terhadap E-Tilang pada Media Sosial Twitter.	Analisis Sentimen Terhadap Tweets Samsung Indonesia Menggunakan Metode Support Vector Machine.	0,4126
Rancang Bangun Dashboard Prediksi Harga Bitcoin dengan Menggunakan Long Short Term Memory (LSTM) Memory.	Implementasi IndoBERT Dalam Analisis Sentimen Berita Untuk Prediksi Harga Saham PT. Bank Rakyat Indonesia Tbk. Menggunakan Pendekatan Support Vector Regression.	0,2215
Perancangan dan Implementasi Chatbot Pembelajaran Candi Borobudur pada Metaversitas.	Pembuatan Sistem Visual Question Answering Berbasis Web Untuk Mendukung Pembelajaran Visual Anak TK Berbahasa Indonesia Menggunakan Deep Learning.	0,1211
...	…	…

Tabel 8, Hasil Cosine Similarity

3.3. HASIL DAN EVALUASI

3.3.1. Pelabelan Manual

Karena data yang digunakan untuk testing bisa dibilang tidak terlalu banyak yaitu 50 data, maka pelabelan untuk mengetahui kemiripan setiap data testing dengan data training yang telah di cocokan oleh model dilakukan secara manual. Hal ini dilakukan agar pelabelan mendapatkan hasil yang objektif.

Contoh data yang telah dilakukan pelabelan secara manual:

Judul tes	Judul train	Similarity Score	Label
Sistem Prediksi Status Stunting dan Severe Stunting Menggunakan Multinomial Logistic Regression pada Anak di Indonesia.	Bayisehatkita: Aplikasi Berbasis Web Untuk Klasifikasi Stunting Pada Aud.	0,7604	1

Implementasi Segmentasi Pakaian Menggunakan Metode Mask R-CNN.	Klasifikasi Tumor Otak Pada Citra MRI Menggunakan en-CNN.	0,3413	0
Aplikasi Metode Support Vector Machine (SVM) untuk Klasifikasi Sentimen Masyarakat Terhadap E-Tilang pada Media Sosial Twitter.	Analisis Sentimen Terhadap Tweets Samsung Indonesia Menggunakan Metode Support Vector Machine.	0,4126	1
Rancang Bangun Dashboard Prediksi Harga Bitcoin dengan Menggunakan Long Short Term Memory (LSTM) Memory.	Implementasi IndoBERT Dalam Analisis Sentimen Berita Untuk Prediksi Harga Saham PT. Bank Rakyat Indonesia Tbk. Menggunakan Pendekatan Support Vector Regression.	0,2215	0
Perancangan dan Implementasi Chatbot Pembelajaran Candi Borobudur pada Metaversitas.	Pembuatan Sistem Visual Question Answering Berbasis Web Untuk Mendukung Pembelajaran Visual Anak TK Berbahasa Indonesia Menggunakan Deep Learning.	0,1211	0
...	…	…	…

Tabel 9, Labeling manual

Keterangan: 1 Berarti judul tes mirip dengan judul train yang telah di cocokan oleh model dan sebaliknya, 0 berarti judul tes tidak mirip.

3.3.2. Tingkat similaritas judul

Judul tes	Judul train	Similaritas Judul	Label
Sistem Prediksi Status Stunting dan Severe Stunting Menggunakan Multinomial Logistic Regression pada Anak di Indonesia.	Bayisehatkita: Aplikasi Berbasis Web Untuk Klasifikasi Stunting Pada Aud.	0,3465	1
Implementasi Segmentasi Pakaian Menggunakan Metode Mask R-CNN.	Klasifikasi Tumor Otak Pada Citra MRI Menggunakan en-CNN.	0,2251	0
Aplikasi Metode Support Vector Machine (SVM) untuk Klasifikasi Sentimen Masyarakat	Analisis Sentimen Terhadap Tweets Samsung Indonesia Menggunakan Metode Support Vector Machine.	0,3052	1

Terhadap E-Tilang pada Media Sosial Twitter.			
Rancang Bangun Dashboard Prediksi Harga Bitcoin dengan Menggunakan Long Short Term Memory (LSTM) Memory.	Implementasi IndoBERT Dalam Analisis Sentimen Berita Untuk Prediksi Harga Saham PT. Bank Rakyat Indonesia Tbk. Menggunakan Pendekatan Support Vector Regression.	0,0701	0
Perancangan dan Implementasi Chatbot Pembelajaran Candi Borobudur pada Metaversitas.	Pembuatan Sistem Visual Question Answering Berbasis Web Untuk Mendukung Pembelajaran Visual Anak TK Berbahasa Indonesia Menggunakan Deep Learning.	0,0474	0
...	…	…	

Tabel 10, Tingkat similaritas judul

3.3.3. Tingkat similaritas abstrak

Judul tes	Judul train	Similaritas Abstrak	Label
Sistem Prediksi Status Stunting dan Severe Stunting Menggunakan Multinomial Logistic Regression pada Anak di Indonesia.	Bayisehatkita: Aplikasi Berbasis Web Untuk Klasifikasi Stunting Pada Aud.	0,3515	1
Implementasi Segmentasi Pakaian Menggunakan Metode Mask R-CNN.	Klasifikasi Tumor Otak Pada Citra MRI Menggunakan en-CNN.	0,4807	0
Aplikasi Metode Support Vector Machine (SVM) untuk Klasifikasi Sentimen Masyarakat Terhadap E-Tilang pada Media Sosial Twitter.	Analisis Sentimen Terhadap Tweets Samsung Indonesia Menggunakan Metode Support Vector Machine.	0,5444	1
Rancang Bangun Dashboard Prediksi Harga Bitcoin dengan Menggunakan Long Short Term Memory (LSTM) Memory.	Implementasi IndoBERT Dalam Analisis Sentimen Berita Untuk Prediksi Harga Saham PT. Bank Rakyat Indonesia Tbk. Menggunakan Pendekatan Support Vector Regression.	0,3607	0

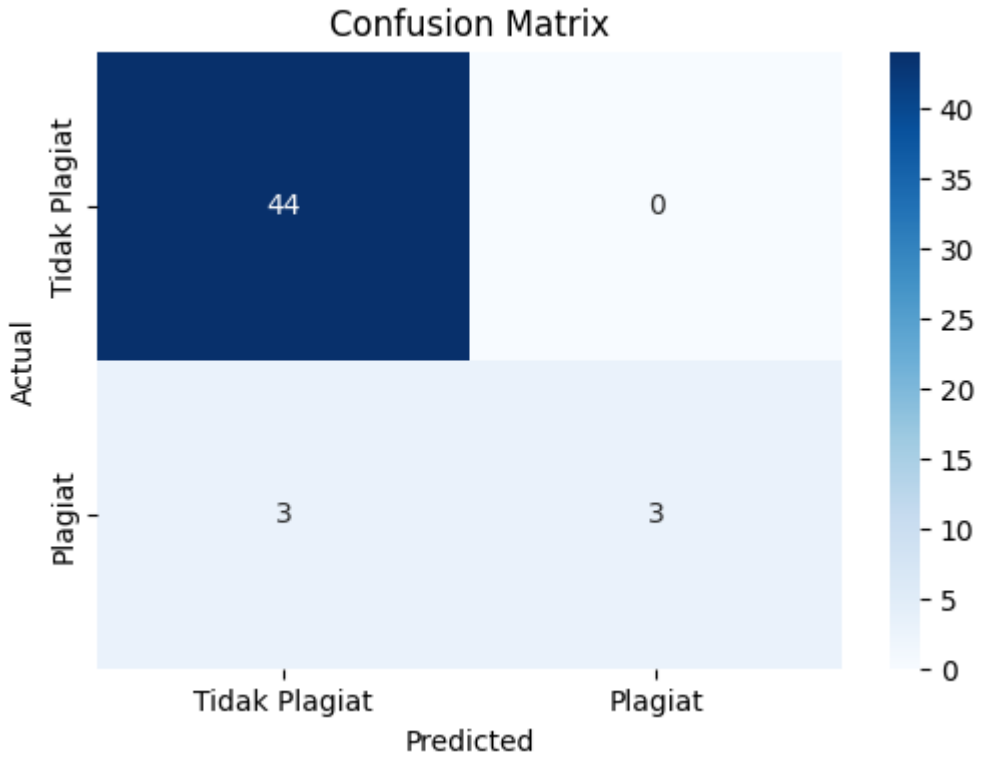
Perancangan dan Implementasi Chatbot Pembelajaran Candi Borobudur pada Metaversitas.	Pembuatan Sistem Visual Question Answering Berbasis Web Untuk Mendukung Pembelajaran Visual Anak TK Berbahasa Indonesia Menggunakan Deep Learning.	0,0689	0
...	…	…	

Tabel 11, Tingkat similaritas abstrak

3.3.4. Confussion Matrix

Dalam penelitian sebelumnya, ambang batas (threshold) yang digunakan untuk menentukan tingkat plagiarisme adalah sebesar 30%. Namun, ambang ini dinilai kurang tepat karena dapat menyebabkan sistem mendeteksi plagiarisme meskipun tingkat kemiripan antara dua dokumen masih tergolong rendah. Pada proyek ini, ambang batas tersebut ditingkatkan menjadi 50%. Hal ini didasarkan pada pertimbangan bahwa suatu dokumen baru dapat dianggap berpotensi sebagai plagiarisme apabila memiliki tingkat kemiripan yang cukup tinggi, yakni setidaknya setengah dari keseluruhan isi dokumen. Dengan menetapkan threshold sebesar 50%, sistem hanya akan mengklasifikasikan suatu dokumen sebagai plagiasi jika tingkat kemiripannya cukup signifikan, sehingga dapat meminimalisasi kesalahan deteksi pada dokumen yang sebenarnya tidak termasuk plagiarisme.

3.3.4.1. *Heatmap Confussion Matrix*



Gambar 11, Heatmap Confussion Matrix

Interpretasi tiap nilai:

- True Negative (TN = 44), artinya model berhasil mengenali 44 dokumen yang bukan *plagiarisme* dengan benar
- False Positive (FP = 0), tidak ada dokumen yang bukan *plagiarisme* tapi diprediksi plagiat, artinya model tidak ada menuduh plagiat pada dokumen yang sebenarnya bukan plagiat.
- False Negative (FN = 3), ada 3 dokumen yang sebenarnya plagiat tapi model gagal mendeteksinya, ini kurang bagus karena ada sebagian kecil plagiasi yang tidak terdeteksi.
- True Positive (TP = 3), model berhasil mendeteksi 3 dokumen yang plagiat.

Dari interpretasi nilai tersebut, bisa disimpulkan model ini sangat baik dalam mengenali dokumen non-plagiat namum model kurang sensitif dalam mengenali dokumen plagiat dikarenakan ada sebagian kecil yang tidak terdeteksi yaitu 3 dokumen.

3.3.4.2. Hasil evaluasi kinerja model

Label	Precision	Recall	F1-Score	Support
0 (Tidak Plagiat)	0,94	1,00	0,97	44
1 (Plagiat)	1.00	0,50	0,67	6
Accuracy			**0,94**	50
Macro Average	0,97	0,75	0,82	50
Weighted Average	0,94	0,94	0,93	50

Tabel 12, Hasil evaluasi kinerja model

Akurasi model pada projek ini mengalami peningkatan sebesar 4,3**%** dibandingkan dengan jurnal rujukan. Pada jurnal rujukan, model hanya mencapai akurasi 89,7**%**, sedangkan model yang dikembangkan dalam projek ini berhasil mencapai akurasi 94**%**. Peningkatan akurasi ini menunjukan bahwa penambahan komponen abstrak berkontribusi signifikan dalam proses pendeteksian tingkat kemiripan sebuah penelitan. Dengan demikian, model menjadi lebih mampu mengidentifikasi kemiripan konten secara menyeluruh, tidak hanya berdasarkan kemiripan judul, tetapi juga berdasarkan substansi dari abstrak. Hal ini menjadikan proses deteksi plagiarisme lebih akurat dan menyeluruh.

4. RENCANA PENGEMBANGAN PROJEK

Untuk pengembangan lebih lanjut, projek ini direncanakan akan ditingkatkan dari beberapa aspek, baik dari sisi data, metode maupun fitur system. Pertama, jumlah dataset akan diperluas dengan menggabungkan data dari berbagai repository skripsi atau tesis dari universitas lain guna meningkatkan generalisasi dan akurasi model. Kedua pendekatan berbasis *TF-IDF* dan *Cosine Similarity* yang saat ini digunakan akan dieksplorasi lebih lanjut dengan membandingkannya terhadap pendekatan berbasis embedding semantic seperti *Word2Vec, FastText,* atau BERT yang mampu menangkap makna kata dalam konteks lebih dalam. Dengan pengembangan ini diharapkan system deteksi kemiripan dokumen dapat digunakan lebih luas tidak hanya untuk mendeteksi plagiarisme dalam skripsi, tetapi juga pada makalah ilmiah, laporan teknis atau dokumen akademik lainnya secara lebih akurat dan efisien.

5. REFERENSI

[1] A. H. Nasrullah, “Integrasi Tf-Idf Dan Algoritma Cosine Similarity Untuk Deteksi Tingkat Kemiripan Judul Penelitian (Studi Kasus Mahasiswa Fakultas Ilmu Komputer UNISAN Gorontalo),” *INTEC Journal: Information Technology Education Journal*, vol. 3, no. 3, 2024, [Online]. Available: https://scholar.google.com/,

[2] M. Dzikry Afandi, A. Homaidi, A. Ghofur, and A. Zubairi, “Penerapan Information Retrieval dalam Sistem Analisis Kemiripan Proposal Skripsi menggunakan Cosine Similarity,” *JURNAL SWABUMI*, vol. 12, no. 1, p. 2023, 2024.

[3] J. Halim and D. Lasut, “Document Plagiarism Detection Application Using Web-Based TF-IDF and Cosine Similarity Methods,” *bit-Tech*, vol. 7, no. 2, pp. 202–213, Dec. 2024, doi: 10.32877/bt.v7i2.1697.

[4] V. Meida Hersianty, E. Larasati Amalia, D. Puspitasari, and D. Wahyu Wibowo, “PENERAPAN ALGORITMA TF-IDF DAN COSINE SIMILARITY DALAM SISTEM REKOMENDASI LOWONGAN PEKERJAAN,” 2025.

[5] “JOURNAL OF INTELLIGENT SYSTEMS AND COMPUTATION 43.” [Online]. Available: https://t.co/9WloaWpfD5