Jurnal Elektronik Ilmu Komputer Udayana
Volume 13, No 2. November 2024

p-ISSN: 2301-5373
e-ISSN: 2654-5101

Perbandingan Kesamaan Tugas Mahasiswa Berbasis *Text Summarization* Menggunakan Metode *Cosine Similarity*

I Gusti Ayu Purnami Pinatih[a1], I Gede Santi Astawa[a2] , Ngurah Agus Sanjaya ER[a3],
Ida Ayu Gde Suwiprabayanti Putra[a4]

[a]Universitas Udayana, Program Studi Informatika
Jimbaran, Bali
[1]purnamipinatih029@student.unud.ac.id
[2]santiastawa@unud.ac.id
[3]agus_sanjaya@unud.ac.id
[4]iagsuwiprabayantiputra@unud.ac.id

Abstract

The manual process of checking student assignments for similarities can be time-consuming and labor-intensive. Implementing text summarization allows for the extraction of important information from lengthy student assignment texts, enabling the identification of similarities in submitted assignments. Therefore, this study applies text summarization methods to reduce the length of document answers, and then compares the summary results to expedite the assessment process. The data used was obtained from assignment archives of a particular course, consisting of 90 documents with 4 essay questions. Summarization is performed by ranking word weights generated using TF-IDF weighting according to the highest weights. The summary results are then compared using cosine similarity. The research results indicate that the system is capable of generating summaries consisting of the highest-weighted words, with evaluation results showing an accuracy of 94.4%. This means that the compared summaries have a fairly high degree of similarity. Meanwhile, the document similarity evaluation by experts shows that out of 105 data comparisons, 67 were found to be consistent, equating to 63.80%. This discrepancy is due to the system only comparing based on the words present in the summary, not based on their meaning.
Keywords: *summary, similarity, tf-idf, cosine similarity*

1. Pendahuluan

Seiring dengan perkembangan zaman mekanisme penilaian pembelajaran juga semakin beragam. Mengutip dari Permenristekdikti No. 44 Tahun 2015 tentang standar nasional pendidikan tinggi pada bagian kelima mengenai Standar penilaian pembelajaran pada pasal 21 ayat (1) terdapat beberapa mekanisme penilaian mulai dari observasi, tes tertulis, tes lisan dan angket. Pemberian tugas dalam bentuk pilihan ganda dan jawaban uraian menjadi metode penilaian tes tertulis. Terkait dengan adanya kebutuhan untuk mengevaluasi kualitas tugas mahasiswa dalam suatu mata kuliah, terutama pada mata kuliah yang menerapkan penugasan berupa jawaban uraian yang cukup panjang dan kompleks. Dalam proses penilaian, untuk memeriksa jawaban uraian memerlukan waktu yang jauh lebih lama dibandingkan memeriksa jawaban pilihan ganda. Ketika mahasiswa mengumpulkan tugas dalam sistem pembelajaran *online* tugas akan dikumpulkan dalam bentuk file yang akan di evaluasi oleh tenaga pendidik atau dosen pengampu dari mata kuliah yang bersangkutan. Evaluasi ini dapat dilakukan dengan memeriksa kesamaan pada tugas yang dikumpulkan oleh mahasiswa. Namun, proses memeriksa kesamaan pada tugas mahasiswa secara manual dapat memakan waktu dan tenaga yang banyak, terutama jika jumlah tugas yang harus diperiksa cukup banyak. Oleh karena itu, diperlukan suatu metode yang lebih efisien dan efektif untuk melakukan evaluasi tugas mahasiswa tersebut. Untuk itu diperlukannya sebuah alat yang bisa digunakan untuk merangkum tugas tanpa mengurangi informasi penting didalamnya dan juga menghemat waktu dalam mengevaluasi.

Salah satu solusi yang dapat digunakan adalah dengan mengimplementasikan metode *text summarization*. Peringkasan teks otomatis (*Automatic Text Summarization*) merupakan teks yang dihasilkan dari satu atau lebih dokumen, yang mana hasil teks tersebut memberikan informasi penting dari sumber dokumen asli, serta secara otomatis hasil teks tersebut tidak lebih panjang dari setengah

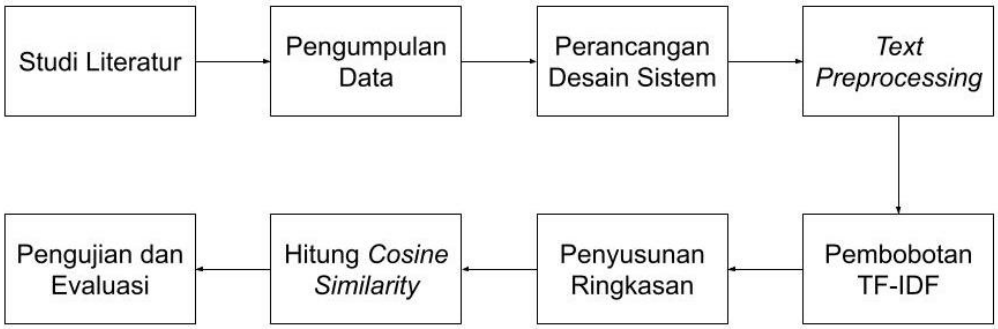
sumber dokumen aslinya [1]. Metode ini memungkinkan pengambilan informasi penting dari teks tugas mahasiswa yang panjang, sehingga dapat diidentifikasi apakah terdapat kesamaan atau plagiarisme pada tugas yang dikumpulkan oleh mahasiswa. Penelitian mengenai membandingkan kemiripan antara dua dokumen yang pernah dilakukan seperti penelitian mengenai Analisis Metode *Cosine Similarity* Pada Aplikasi Ujian *Online* Esai Otomatis (Studi Kasus Jti Polinema)[2]. Pada penelitian ini dilakukan pembuatan sistem ujian esai *online* dengan penilaian kemiripan jawaban menggunakan metode *Cosine similarity* dan persamaan *Term Frequency* (*TF*) untuk menyamakan frekuensi setiap kata yang terdapat dalam kalimat. Untuk pengujian akurasi metode dilakukan pengujian *Precision*, *Recall*, dan *f-measure* dan berdasarkan hasil analisis dengan menggunakan metode yang telah dicoba diperoleh rata-rata 81%. Kemudian terdapat penelitian yang dilakukan untuk melakukan deteksi plagiarisme pada artikel junal menggunakan metode *cosine similarity* [3]. Pada penelitian ini dilakukan pengecekan plagiarisme terhadap suatu artikel yang kemudian isi artikel tersebut dilakukan perbandingan kemiripannya terhadap dokumen repositori untuk mengetahui nilai kemiripan dari artikel yang ingin di cek plagiarisme nya. Namum dalam penelitian yang disebutkan diatas hanya membandingkan kemiripan dokumen tanpa melalui proses peringkasan teks.

Penelitian mengenai peringkasan teks yang pernah dilakukan yaitu peringkasan artikel berbahasa Indonesia dengan menerapkan algoritma lexrank. Pada penelitian ini dilakukan peringkasan sebanyak 300 artikel yang terdiri dari beberapa topik. Berdasarkan hasil pengujian pada penelitian tersebut hasil pengujian ringkasan yang terbentuk dipengaruhi oleh tingkat kompresi yang diterapkan semakin kecil tingkat kompresinya maka hasil pengujian nya semakin kecil pula. Dalam penelitian ini dilakukan pengujian dengan kompresi 50% dengan rata-rata skor *f-measure* 67,53% dan pada kompresi 30% menghasilkan rata-rata skor *f-measure* sebesar 55,82% [4]. Berikutnya dilakukan penelitian mengenai penerapan peringkas teks otomatis terhadap modul pembelajaran bahasa Indonesia menggunakan metode *cross latent semantic analysis*. Penelitian ini meringkas 10 buah modul pembelajaran yang menghasilkan rata-rata akurasi *f-measure* sebesar 38,53% pada *compression rate* 20% [5].

Berdasarkan penelitian-penelitian sebelumnya, *text summarization* menunjukkan performa yang cukup baik dan dapat mempermudah dalam membaca isi dokumen. Sehingga pada penelitian ini penulis akan menerapkan *text summarization* dan melakukan perbandingan hasil ringkasan menggunakan metode *cosine similarity* untuk melakukan evaluasi tugas mahasiswa yang berupa jawaban teks uraian.

2. Metodologi Penelitian

Dalam penelitian ini terdapat beberapa tahapan yang terdiri dari pengumpulan data, perancangan desain sistem, kemudian dilanjutkan dengan data *preprocessing*, selanjutnya melakukan pembobotan kata dengan menggunkan *TF-IDF*, lalu dilanjutkan dengan penyusunan ringkasan. Setelah hasil ringkasan terbentuk akan dilanjutkan dengan proses perbandingan kesamaan antar tugas menggunakan *cosine similarity* dan diikuti dengan evaluasi serta analisis hasil penelitian.



Gambar 1. Alur Penelitian

Pada grafik yang dapat dilihat pada gambar 1 menunjukkan tahapan-tahapan dari alur penelitian yang dilaksanakan dalam penelitian ini.

2.1 Data dan Pengumpulan Data

Data yang digunakan pada penelitian ini berasal dari arsip tugas mahasiswa pada Mata Kuliah Matematika Diskrit Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Udayana. Jumlah data yang akan digunakan dalam penelitian adalah sebanyak 90 file tugas mahasiswa yang masing- masing file terdiri dari 4 soal jawaban teks uraian. Arsip data tugas mahasiswa berupa kumpulan file yang dikumpulkan dalam satu folder. Data tersebut berupa file jawaban

p-ISSN: 2301-5373
e-ISSN: 2654-5101

teks uraian dengan panjang kata dikisaran 20 – 250 kata. Data tersebut akan dibagi menjadi dua bagian yaitu data latih dan data uji, dimana 75 dokumen untuk melakukan pelatihan data dan 15 dokumen untuk melakukan pengujian.

2.2 Perancangan Desain Sistem

Pada tahap ini dilakukan proses perancangan terhadap sistem yang akan dibuat. Dilakukan beberapa langkah diantaranya analisis kebutuhan sistem, *text preprocessing*, pembobotan kata, penyusunan ringkasan, perhitungan *cosine similarity* dan implementasi sistem.

a. Analisis Kebutuhan Sistem

Analisis kebutuhan sistem yang dilakukan pada penelitian ini dibagi menjadi analisis kebutuhan fungsional dan analisis kebutuhan non fungsional.

1. **Kebutuhan Fungsional**

   Analisis kebutuhan Fungsional meliputi kegunaan dari sistem. Pada kebutuhan fungsional sistem ini meliputi :

Tabel 1. Analisis Kebutuhan Fungsional

Preprocessing	Sistem harus mampu mengolah dokumen dalam format csv. Sistem mampu melakukan *Preprocessing* terhadap dataset.
Pembobotan kata	Sistem dapat menghitung skor *tf-idf* untuk setiap kata dalam dokumen.
Peringkasan dukumen	Mampu menghasilkan ringkasan dokumen berdasarkan skor *tf-idf* tertinggi hingga terendah
Perhitungan kesamaan dokumen	Sistem dapat menghitung kesamaan antara dokumen secara berpasangan untuk masing-masing soal.

   Pada tabel 1 dapat dilihat bahwa tabel menunjukkan jenis kebutuhan dari sistem yang terdiri dari empat proses utama yang dilakukan sistem.

2. **Kebutuhan Non Fungsional**

   Analisis kebutuhan non-fungsional meliputi komponen pendukung yang digunakan dalam menunjang penelitian ini.

Tabel 2. Analisis Kebutuhan Non-fungsional

Perangkat	**Penjelasan**
Windows 10	Sistem operasi
Visual Studio Code	*Code editor* yang digunakan dalam pembuatan sistem berbasis *website*
Python	Bahasa pemrograman komputer yang digunakan untuk pengembangan sistem
Flask	*Framework python* untuk pengembangan aplikasi berbasis *website*

   Pada tabel 2 dapat dilihat bahwa jenis kebutuhan dari sistem yang dari empat komponen yang digunakan dalam membangun sistem nantinya.

b. *Text Preprocessing*

Text Preprocessing merupakan tahapan dari proses awal terhadap teks untuk mempersiapkan teks menjadi data yang akan diolah lebih lanjut. Suatu teks tidak dapat diproses langsung oleh algoritma pencarian, oleh karena itu dibutuhkan *preprocessing text* untuk mengubah teks menjadi data numerik [6]. *Text Preprocessing* yang dilakukan pada penelitian ini melalui beberapa tahapan diantaranya :

1. *Case folding* : mengubah semua huruf dalam dokumen menjadi huruf kecil. Hanya huruf a sampai dengan z yang diterima [7].
2. *Cleaning* : menghilangkan karakter khusus dalam dokumen seperti tanda baca.
3. *Tokenization* : memisahkan kata dalam kalimat atau paragraf menjadi bentuk kata atau token[8].

4. *Stopword Removal* : menghilangkan kata-kata umum yang tidak memiliki makna khusus seperti kata penghubung[9].
5. *Negation Handling* : menangani kata kata yang memiliki makna negasi. Menghubungkan kata negasi dengan kata setelahnya dan dijadikan sebagai satu kata.

c. Pembobotan Kata

Metode pembobotan kata yang digunakan dalam penelitian ini adalah *tf-idf*. Metode *TF-IDF* merupakan metode untuk menghitung bobot suatu kata (*term*) terhadap dokumen. Metode ini menggabungkan dua konsep untuk perhitungan bobot, yaitu frekuensi kemunculan sebuah kata didalam sebuah dokumen tertentu dan *inverse* frekuensi dokumen yang mengandung kata tersebut [8]. Frekuensi kemunculan kata di dalam dokumen yang diberikan menunjukan seberapa penting kata tersebut di dalam dokumen [10]. Metode ini akan menghitung bobot setiap *term* di dokumen dengan menghitung nilai *tf* yang diperoleh dari :

$$tf_d = \frac{jumlah\ munculnya\ kata\ t\ dalam\ dokumen}{total\ jumlah\ seluruh\ kata\ dalam\ dokumen}$$

Kemudian nilai *tf* dikalikan dengan nilai *idf* yang diperoleh dari:

$$IDF = log_2 \frac{d}{df} + 1$$

Sehingga diperoleh persamaan dari perhitungan *tf-idf* adalah seperti yang ditunjukkan pada persamaan 1.

$$W_d = t\,f * IDF \quad (1)$$

Keterangan :

W = bobot dokumen ke –d
d = dokumen
t = kata kunci
tf = *term*s *frequency* (jumlah kemunculan kata)
df = jumlah dokumen yang mengandung kata kunci.
IDF = *Inverse Document Frequency*

Data yang telah melalui proses *Preprocessing* akan dihitung bobotnya untuk tiap kata didalam masing masing dokumen di setiap soal. Contoh kata “matematika” muncul sebanyak 2 kali pada dokumen D dengan jumlah dokumen sebanyak 15 dokumen, serta terdapat 5 dokumen yang mengandung kata tersebut. Frekuensi kemunculan kata di tiap dokumen dibagi dengan total jumlah kata dalam dokumen. Maka nilai *tf-idf* untuk kata “matematika” pada dokumen D adalah sebagai berikut :

*TF***IDF*=2*log 15/5 + 1 = 1,954

Dengan menerapkan persamaan (1) diatas maka ditemukan nilai *tf-idf* untuk kata “matematika” pada dokumen D adalah 1,954.

d. Penyusunan Ringkasan

Setelah mendapatkan bobot untuk setiap kata, bobot tersebut diurutkan berdasarkan bobot tertinggi ke bobot terendah. Pembentukan ringkasan dilakukan dengan mengurangi jumlah kata dalam dokumen. Secara umum teks dikatakan ringkasan apabila teks lebih pendek dari teks aslinya dan jga memuat informasi penting dari teks aslinya. *Compression ratio* dihasilkan dari pembagian antara panjang kata dalam ringkasan dengan panjang kata dalam teks aslinya. [1]. Rasio retensi menentukan seberapa banyak informasi yang disimpan. Ringkasan yang baik adalah ringkasan yang memiliki rasio retensi tinggi dan *compression ratio* rendah [11].Pada penelitian ini jumlah kata yang dikurangi atau biasa disebut *compression ratio* untuk menghasilkan ringkasan adalah sebanyak setengah dari jumlah kata pada dokumen asli. Setelah menentukan *compression ratio* maka akan dibentuk ringkasan menurut bobot tertinggi hingga terendah. *Compression ratio* dapat dihitung dengan mengikuti persamaan 3 sebagai berikut.

$$compression\ ratio = \frac{panjang\ \ kata\ ringkasan}{total\ kata\ keseluruhan} \quad (2)$$

Sebagai contoh jika total kata dalam dokumen asli adalah 30 kata dan panjang ringkasan adalah 15 kata maka *compression rasio* dari ringkasan tersebut adalah ½.

e. Perhitungan *Cosine Similarity*

Metode *cosine similarity* merupakan metode yang digunakan untuk menghitung *similarity* (tingkat kesamaan) antar dua buah objek. Dalam kasus analisis perbandingan kesamaan tugas mahasiswa, *cosine similarity* dapat digunakan untuk membandingkan kesamaan antara tugas mahasiswa yang telah diubah menjadi vektor representasi kata-kata.Secara umum penghitungan metode ini didasarkan pada *vector space similarity measure*. Metode *cosine similarity* ini menghitung *similarity* antara dua buah objek yang dinyatakan dalam dua buah vektor dengan menggunakan *keyword*s (kata kunci) dari sebuah dokumen sebagai ukuran[12]. Adapun rumus dari perhitungan *cosine similarity* dapat dilihat pada persamaan (3) dibawah ini.

$$Cos\ a = \frac{A \cdot B}{|A||B|} = \frac{\sum_{i=1}^{n} A_i \times B}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}} \quad (3)$$

Keterangan :
A = Vektor A, yang akan dibandingkan kemiripan
B = Vektor B, yang akan dibandingkan kemiripan
A·B = *dot product* antara vektor A dan vektor B
A.B = *dot product* antara vektor A dan vektor B
|A|= Panjang vektor A
|B|= Panjang vektor B
|A||B|= *cross product* antara |A| dan |B|

Perhitungan *cosine similarity* diawali dengan menghitung *dot product* dari vektor frekuensi data untuk dua dokumen. *Dot product* merupakan jumlah perkalian antara frekuensi kata yang sama di kedua dokumen. Kemudian menghitung panjang (magnitude) vektor frekuensi kata untuk masing-masing dokumen. *Cosine similarity* dihitung sebagai *dot product* dibagi dengan hasil kali panjang vektor. Jika salah satu dokumen memiliki panjang vektor 0 (dokumen kosong) maka nilai *cosine* diatur menjadi 0 untuk menghindari pembagian dengan nol. Perhitungan dilakukan untuk seluruh pasangan dokumen yang kemudian disimpan kedalam tabel *similarity*, selain itu untuk pasangan dokumen yang memiliki nilai *similarity* terbesar akan ditampilkan setelah perhitungan kemiripan untuk setiap soal selesai dilakukan.

Sebagai contoh jika terdapat dokumen “saya suka makan ayam” dan “saya suka makan” jika dihitung mengikuti persamaan 3 akan menghasilkan nilai sebagai berikut.

term	**saya**	**suka**	**makan**	**ayam**
doc 1 (A)	1	1	1	1
doc 2 (B)	1	1	0	1

Maka *dot product* dari vektor A dan B adalah

$$A \cdot B = (1 \times 1) + (1 \times 1) + (1 \times 0) + (1 \times 1) = 3$$

Dengan panjang vektor A dan B

$$||A|| = \sqrt{(1^2 + 1^2 + 1^2 + 1^2} = \sqrt{4} = 2$$
$$||B|| = \sqrt{(1^2 + 1^2 + 0^2 + 1^2} = \sqrt{3} = 1.732$$

Kemudian nilai *cosine similarity* dari dua dokumen tersebut adalah

$$cosine\ similarity = \frac{3}{2 \times 1.732} = \frac{3}{3.464} = 0.866$$

Berdasarkan persamaan 2 diperoleh nilai *cosine similarity* 0.866 yang menunjukkan kedua dokumen memiliki kemiripan yang cukup tinggi karena nilai kemiripannya mendekati 1.

f. Implementasi Sistem

Setelah melalui proses perancangan, pada tahap implementasi ini sistem yang dibangun merupakan sebuah aplikasi sederhana berbasis *website*. Implementasi dilakukan dengan menggunakan bahasa pemrograman *Python* menggunakan *Visual Studio Code* sebagai *code editor* dan *framework Flask*. *Flask* adalah sebuah *framework* dalam *python* untuk pengembangan aplikasi *website*. Sistem yang dibangun akan menerima input berupa file dengan format csv yang memuat beberapa kolom berisi teks uraian. Kemudian sistem akan memproses data dan menghasilkan output berupa tabel yang memuat hasil perbandingan kemiripan keseluruhan dokumen.

2.3 Desain Evaluasi

Evaluasi hasil ringkasan akan dilakukan dibedakan menjadi dua bagian yaitu evaluasi hasil ringkasan dan evaluasi kemiripan jawaban oleh pakar.

a. Evaluasi Hasil Ringkasan Sistem

Untuk mengetahui hasil ringkasan yang dihasilkan sistem dilakukan evaluasi dengan cara membandingkan ringkasan yang dihasilkan sistem (tanpa *library*) dengan ringkasan yang dihasilkan dengan menggunakan *library*. Perbandingan hasil ringkasan dengan *library* berfungsi untuk mengetahui apakah fungsi yang dibuat sendiri ini sudah mampu menghasilkan hasil yang mirip atau sama dengan fungsi yang sudah ada. Performa hasil ringkasan diukur dari nilai *cosine similarity* yang dihasilkan antara hasil ringkasan dengan dan tanpa bantuan *library*. Jika nilai *cosine similarity* yang dihasilkan mendekati atau sama dengan satu (1) maka dokumen yang dibandingkan tersebut memiliki kesamaan. Sebaliknya jika nilai kemiripannya mendekati nol (0) maka dokumen yang dibandingkan tersebut tidak memiliki kemiripan. Setelah mendapatkan nilai kemiripan maka dapat dihitung akurasi ringkasannya dengan persamaan 4.

$$akurasi = \frac{total\ jumlah\ nilai\ kemiripan}{total\ jumlah\ dokumen\ untuk\ seluruh\ soal} \times 100 \quad (4)$$

b. Evaluasi Perbandingan Kesamaan Jawaban

Ringkasan yang sudah dihasilkan sistem kemudian dilakukan perbandingan kesamaan jawaban antara seluruh dokumen di tiap soal. Perbandingan kesamaan jawaban ini dilakukan dengan tujuan untuk mengetahui seberapa tingkat kemiripan jawaban-jawaban yang telah dihasilkan dari seluruh dataset. Pengujian dilakukan dengan membandingkan similaritas antar dokumen. Dari hasil perhitungan *cosine similarity* yang sudah dihasilkan kemudian diurutkan dari kemiripan tertinggi hingga terendah. Apabila total similaritas yang didapatkan adalah mendekati nol (0) maka dokumen yang diolah tidak memiliki kesamaan dan jika nilai yang didapatkan maksimal adalah 1 maka dokumen tersebut memiliki kemiripan.

3. Hasil dan Pembahasan

3.1 Pengumpulan Data

Data yang digunakan dalam penelitian ini merupakan data arsip tugas mahasiswa yang terdiri dari 4 soal dengan jawaban teks uraian. File tugas tersebut dikumpulkan kedalam beberapa folder sesuai kelas dengan jumlah 90 dokumen. Kemudian data tersebut dikelompokkan menjadi data latih sejumlah 75 file dan 15 file untuk data uji kedalam format file csv yang Adapun soal dan contoh jawaban data yang diperoleh dari file jawaban tugas mahasiswa adalah sebagai berikut :

Tabel 3. Contoh soal

Q1	Q2	Q3	Q4
Apa yang dimaksud himpunan kosong?	Jelaskan pemahaman fungsi rekursif!	Jelaskan mengenai proporsi majemuk!	Bagaimana algoritma konversi bilangan hexsadesimal ke oktal?

Pada tabel 3 memuat pertanyaan dalam dataset dimana Q melambangkan *Question*.

Tabel 4. Contoh jawaban

A1	A2	A3	A4
Himpunan kosong adalah himpunan yang tidak memiliki anggota satu pun, atau sama sekali tidak memiliki anggota. Dalam notasi himpunan, himpunan kosong biasanya dilambangkan dengan simbol ? atau {}.	Fungsi rekursif adalah jenis fungsi pada pemrograman yang memanggil dirinya sendiri untuk menyelesaikan masalah yang lebih besar. Fungsi ini biasanya digunakan untuk menghitung pangkat, faktorial dan fibonnaci.	Proposisi majemuk merupakan gabungan dari beberapa proposisi atomik yang menghasilkan sebuah nilai kebenaran baru. Proposisi atomik dapat digabungkan dengan menggunakan operator logika.	Langkah awal yaitu dengan mengkonversi bilangan heksadesimal ke desimal terlebih dahulu. Kemudian kita hitung total nilai decimal tersebut, Langkah kedua yaitu konversi dari desimal ke oktal, dengan membagi hasil desimal yang sudah kita dapatkan dengan 8.

p-ISSN: 2301-5373
e-ISSN: 2654-5101

Pada tabel 4 memuat contoh jawaban yang terdapat dalam dataset dimana A melambangkan *answer* atau jawaban.

3.2 Data *Preprocessing*

Pada tahap *preprocessing* ini data akan melewati proses perubahan ke huruf kecil, *tokenization*, *cleaning* data, menghapus tanda baca, menghapus angka, menghilangkan karakter khusus, penghapusan *stopword* dan penanganan negasi. Berikut merupakan contoh perbandingan data sebelum dan setelah melalui *preprocessing* ditunjukkan pada tabel 5.

Tabel 5. Perbandingan Sebelum dan Setelah *Preprocessing*

Sebelum	Sesudah
Himpunan kosong adalah himpunan yang tidak memiliki anggota satu pun, atau sama sekali tidak memiliki anggota. Dalam notasi himpunan, himpunan kosong biasanya dilambangkan dengan simbol ? atau {}.	himpunan kosong himpunan tidakmemiliki anggota tidakmemiliki anggota notasi himpunan himpunan kosong lambang simbol

Pada tabel 5 menunjukkan salah satu jawaban dalam dataset setelah melalui tahap *preprocessing*.

3.3 Pembobotan Kata

Data yang sudah melalui tahap *preprocessing* kemudian dilakukan pembobotan kata untuk menentukan bobot tertinggi hingga terendahnya. Pembobotan kata ini menggunakan *tf-idf* untuk mengubah data teks menjadi data numerik. Pada tahap ini dilakukan perhitungan *tf* dan *idf* dilakukan menggunakan fungsi yang dibuat sendiri mengikuti persamaan (1) yang kemudian dikalikan sehingga menghasilkan representasi numerik dari data.

	algoritma	konversi	bilangan	hexadesimal	oktal	mengubah	biner	pisahkan	angka	koversi
0	0.300105	0.172884	0.669392	0.425867	0.267757	0.272042	0.323883	NaN	NaN	NaN
1	NaN	NaN	0.096435	0.153379	0.192869	NaN	0.349947	0.195956	0.538320	0.557922
2	NaN	0.131982	0.204410	0.081278	0.255512	NaN	0.309071	NaN	NaN	NaN
3	NaN	0.053351	0.371829	NaN	0.041314	NaN	0.299847	0.083951	0.076875	NaN
4	NaN	0.070922	0.384448	NaN	0.109842	NaN	NaN	NaN	NaN	NaN
5	NaN	NaN	0.238968	NaN	NaN	0.060698	NaN	NaN	0.111164	NaN
6	NaN	0.301767	NaN	NaN	0.233683	NaN	NaN	NaN	NaN	NaN
7	NaN	0.091247	0.565282	0.112385	0.070660	NaN	0.170944	NaN	NaN	NaN
8	NaN	0.298286	0.115494	0.183693	0.115494	NaN	0.139703	NaN	NaN	NaN
9	NaN	0.140462	0.543857	NaN	0.108771	NaN	0.219286	NaN	NaN	NaN
10	NaN	0.181912	0.563477	0.336078	0.211304	NaN	0.340795	NaN	0.131060	NaN
11	NaN	NaN	0.138544	NaN	0.138544	0.281523	NaN	NaN	NaN	NaN
12	0.091567	0.052750	0.285939	0.064969	0.204242	NaN	0.098822	0.083004	NaN	NaN
13	NaN	NaN	0.356636	NaN	0.178318	NaN	0.215696	NaN	NaN	NaN
14	0.098234	NaN	0.306760	NaN	0.219114	0.089048	0.106018	0.178097	0.081543	NaN

15 rows × 124 columns

Gambar 2 Hasil *Tf-idf*

Pada gambar 2 dapat dilihat hasil dari pembobotan kata menggunakan *tf-idf* yang menunjukkan bobot dari tiap kata dalam dokumen. NaN dalam data berarti kata tersebut tidak muncul dalam dokumen bersangkutan sehingga tidak memuat nilai didalamnya. Sebagai perbandingan, pembobotan kata *tf-idf* juga dilakukan dengan menggunakan fungsi *TfidfVectorizer* dari pustaka *sklearn* dengan hasil yang ditunjukkan pada gambar 3.

	algoritma	angka	baca	bagiannya	basis	bentuk	bergerak	bersisa	berurutan	bilang
0	0.300105	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
1	0.000000	0.538320	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
2	0.000000	0.000000	0.128362	0.000000	0.256725	0.147826	0.000000	0.000000	0.000000	0.000000
3	0.000000	0.076875	0.000000	0.119512	0.000000	0.000000	0.000000	0.119512	0.000000	0.119512
4	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.158873	0.000000
5	0.000000	0.111164	0.000000	0.000000	0.000000	0.000000	0.086409	0.000000	0.000000	0.000000
6	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
7	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
8	0.000000	0.000000	0.000000	0.000000	0.580209	0.000000	0.000000	0.000000	0.000000	0.000000
9	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
10	0.000000	0.131060	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
11	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
12	0.091567	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
13	0.000000	0.000000	0.223956	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
14	0.098234	0.081543	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

15 rows × 124 columns

Gambar 3 Hasil *Tf-idf* (*sklearn*)

Berdasarkan gambar 2 dan 3 dapat dilihat bahwa terdapat perbedaan bobot antara hasil *tf-idf* dengan menggunakan *library* dan fungsi yang dibuat sendiri. Perbedaan ini bisa disebabkan karena *library sklearn* memiliki tokenisasi yang lebih canggih sedangkan fungsi yang dibuat sendiri hanya menerapkan '*split()*' sederhana yang memisahkan kata berdasarkan spasi. Serta dalam *library sklearn* menerapkan normalisasi yang secara otomatis menangani beberapa kasus tepi dan normalisasi tambahan yang tidak ada dalam fungsi buatan sendiri sehingga terdapat perbedaan bobot yang dihasilkan.

3.4 Pembentukan Ringkasan

Pembentukan ringkasan dilakukan dengan menentukan terlebih dahulu berapa *compression rate* yang digunakan kemudian dikalikan dengan total jumlah kata dalam dokumen. Berikutnya bobot *tf-idf* yang telah diperoleh dalam tahap sebelumnya akan diurutkan dari bobot tertinggi hingga terendah. Adapun hasil ringkasan yang terbentuk ditunjukkan pada tabel 6.

Tabel 6 Contoh Hasil Ringkasan

Nomor soal	**Hasil Ringkasan**
A1	himpunan dinotasikan kosong dimana anggota
A1	himpunan dimana dilambangkan anggota
A2	solusi pemrogramman dihentikan berdasarkan tercapai konteks menentukan mencapai
A2	dimana penggunaan penyetop tidaklagi solusinya diperoleh menerapkan langkah menggiring penyetopnya mencerminkan fungsi rekursif
A3	kebenaran berdasarkan proposisi premis nilai kalimat kebenarannya dianggap true false pemahaman definisi menentukan kesimpulan majemuk memiliki dimana bernilai gabungan
A3	hasil pengkombinasian
A4	bilangan hexadesimal biner algoritma mengubah oktal konversi
A4	koversi angka biner kedalam pisahkan oktal gabungkan konversikan hexadesimal

Tabel 6 menunjukkan contoh hasil ringkasan yang terbentuk dari bobot *tf-idf* yang dibuat menggunakan fungsi manual tanpa menggunakan *library*. Berdasarkan hasil ringkasan yang terbentuk dapat dilihat bahwa hasil ringkasan yang dihasilkan tidak membentuk satu kalimat utuh. Hal tersebut bisa terjadi karena ringkasan dibentuk dengan mengurutkan bobot tertinggi ke terendah tidak memerhatikan urutan kata dalam kalimat. Sebagai perbandingan akan dibandingkan dengan hasil ringkasan *library* apakah hasil ringkasan yang dihasilkan sudah sesuai atau berbeda.

p-ISSN: 2301-5373
e-ISSN: 2654-5101

Tabel 7 Contoh Hasil Ringkasan (*tf-idf sklearn*)

Nomor soal	Hasil Ringkasan
A1	himpunan dinotasikan kosong dimana anggota
A1	himpunan dimana dilambangkan anggota
A2	solusi berdasarkan pemrogramman dihentikan tercapai mencapai konteks menentukan
A2	dimana solusinya mencerminkan menerapkan menggiring diperoleh penggunaan penyetop penyetopnya langkah tidaklagi rekursif fungsi
A3	berdasarkan kebenaran proposisi premis nilai kalimat definisi false kesimpulan pemahaman menentukan kebenarannya dianggap true majemuk dimana memiliki bernilai operator
A3	hasil pengkombinasian
A4	bilangan hexadesimal biner algoritma mengubah oktal konversi
A4	koversi angka biner kedalam pisahkan oktal konversikan gabungkan hexadesimal

Tabel 7 menampilkan hasil ringkasan menggunakan library. Berdasarkan hasil ringkasan yang dihasilkan dapat dilihat bahwa hasil ringkasan yang dihasilkan oleh *tf-idf* dengan fungsi buatan sendiri pada tabel 6 dan *tf-idf* menggunakan *library* pada tabel 7 sedikit memiliki perbedaan, hal tersebut dapat terjadi karena bobot kata yang dihasilkan pada gambar 2 dan 3 juga memiliki perbedaan. Ringkasan yang dihasilkan tidak membentuk satu kalimat yang utuh. Hal tersebut dikarenakan ringkasan yang disusun berdasarkan bobot tertinggi ke bobot terendah tanpa memedulikan susunan kata dalam kalimat.

3.5 Perhitungan *Cosine Similarity*

Hasil ringkasan yang telah terbentuk pada tahap sebelumnya kemudian dilakukan perbandingan antara dokumen satu dengan yang lainnya untuk mengetahui tingkat kemiripan antar dokumen. Dokumen dari tiap pertanyaan akan dibandingkan berpasangan antara dokumen A dengan B, dokumen A dengan C, hingga seterusnya. Berdasarkan hasil perbandingan dokumen diperoleh hasil seperti yang ditunjukkan pada gambar 4.

doc	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	0,89	0,83	0,84	0,76	0,84	0,89	0,53	0,68	0,86	0,87	0,85	0,81	0,82	0,86
2	0,89	0	0,76	0,81	0,7	0,73	0,91	0,54	0,58	0,78	0,79	0,77	0,78	0,83	0,8
3	0,83	0,76	0	0,8	0,7	0,8	0,8	0,48	0,67	0,77	0,78	0,83	0,75	0,73	0,81
4	0,84	0,81	0,8	0	0,73	0,81	0,83	0,54	0,63	0,78	0,79	0,84	0,75	0,83	0,85
5	0,76	0,7	0,7	0,73	0	0,81	0,77	0,44	0,68	0,7	0,71	0,79	0,74	0,64	0,74
6	0,84	0,73	0,8	0,81	0,81	0	0,82	0,48	0,74	0,78	0,79	0,86	0,78	0,7	0,86
7	0,89	0,91	0,8	0,83	0,77	0,82	0	0,54	0,67	0,82	0,84	0,88	0,82	0,83	0,83
8	0,53	0,54	0,48	0,54	0,44	0,48	0,54	0	0,37	0,49	0,5	0,49	0,47	0,52	0,58
9	0,68	0,58	0,67	0,63	0,68	0,74	0,67	0,37	0	0,71	0,73	0,74	0,64	0,6	0,65
10	0,86	0,78	0,77	0,78	0,7	0,78	0,82	0,49	0,71	0	0,99	0,78	0,77	0,8	0,82
11	0,87	0,79	0,78	0,79	0,71	0,79	0,84	0,5	0,73	0,99	0	0,8	0,78	0,81	0,83
12	0,85	0,77	0,83	0,84	0,79	0,86	0,88	0,49	0,74	0,78	0,8	0	0,81	0,79	0,82
13	0,81	0,78	0,75	0,75	0,74	0,78	0,82	0,47	0,64	0,77	0,78	0,81	0	0,72	0,79
14	0,82	0,83	0,73	0,83	0,64	0,7	0,83	0,52	0,6	0,8	0,81	0,79	0,72	0	0,81
15	0,86	0,8	0,81	0,85	0,74	0,86	0,83	0,58	0,65	0,82	0,83	0,82	0,79	0,81	0

Gambar 4 Matriks Nilai Kesamaan Jawaban Nomor 1

Pada gambar 4 menunjukkan hasil perbandingan kemiripan jawaban untuk jawaban nomor 1 dapat diketahui bahwa untuk jawaban nomor 1 diantara dokumen 1 – 15 memiliki nilai kemiripan yang beragam. Untuk melihat nilai kemiripan terbesar dari seluruh pasangan jawaban dapat dilihat pada tabel 8.

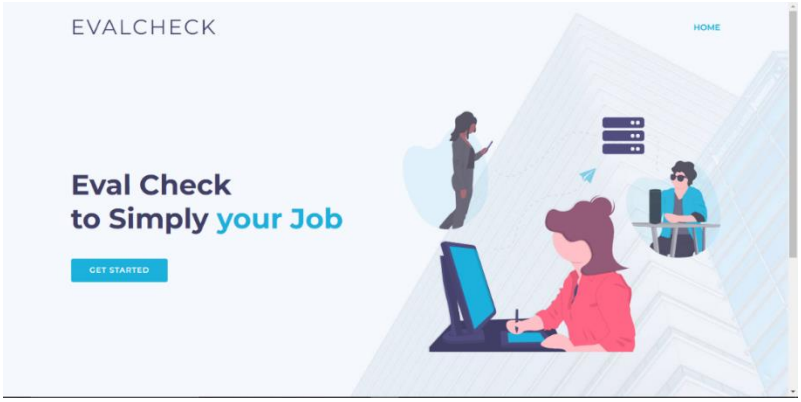
Tabel 8 Kemiripan Hasil Jawaban

Kolom Soal	Dokumen 1	Dokumen 2	Kemiripan Terbesar
A1	10	11	0.985184
A2	3	15	0.861892
A3	8	9	0.799305
A4	1	11	0.910191

Setelah mengetahui nilai kemiripan dari masing masing dokumen maka dapat dilihat pada tabel 8 menunjukkan pasangan dokumen dari masing masing soal yang memiliki nilai kemiripan terbesar. Dimana kemiripan terbesar dari jawaban nomor 1 terdapat pada pasangan dokumen 10 dan 11 sebesar 98% yang artnya ddokumen tersebut sangat mirip.

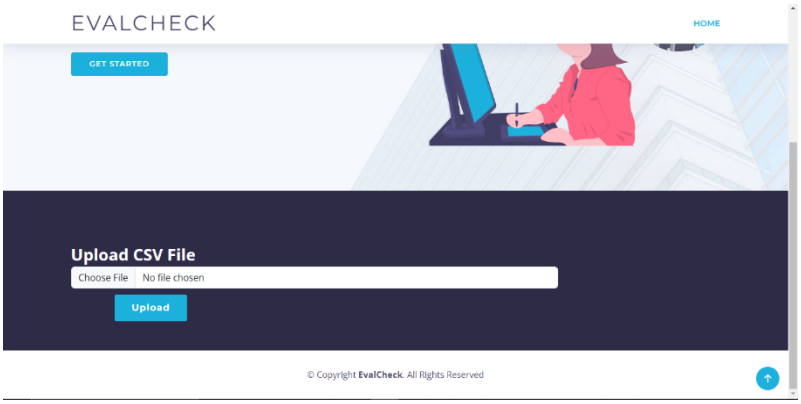
3.6 Implementasi Sistem

Sistem yang dikembangkan adalah sistem aplikasi sederhana berbasis *website* yang bernama EvalCheck. Sistem ini dikembangkan dengan menggunakan *code editor VS Code* dengan bahasa pemrograman *Python* dan *framework Flask* yang terdiri dari dua halaman dan memiliki satu fitur utama yaitu *upload* file csv. Dengan tampilan yang ditunnjukkan pada gambar 4.



Gambar 5 Tampilan Awal Sistem

Gambar 5 merupakan halaman pertama dalam sistem yang memuat tampilan awal dari sistem yang dikembangkan. Dalam halaman ini terdapat sebuah tombol '*Get Started*' yang berfungsi untuk mengarahkan pengguna ke menu utama dari sistem EvalCheck.

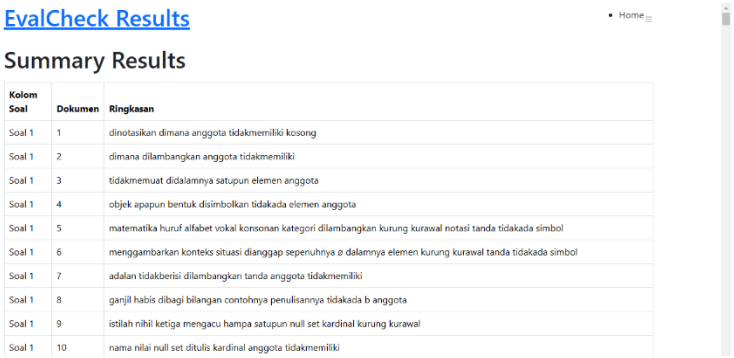


Gambar 6 Tampilan menu utama sistem

Berikutnya pada gambar 6 menampilkan fitur utama dari sistem EvalCheck. Pada halaman ini pengguna dapat mengunggah file berupa csv kedalam sistem dengan menekan tombol '*Upload*'. Kemudian pengguna dapat memilih file dataset csv yang dimiliki. Setelah pengguna

p-ISSN: 2301-5373
e-ISSN: 2654-5101

memilih file yang diinginkan pengguna dapat menekan tombol *upload*. Selanjutnya pengguna akan diarahkan ke halaman berikutnya.


EvalCheck Results

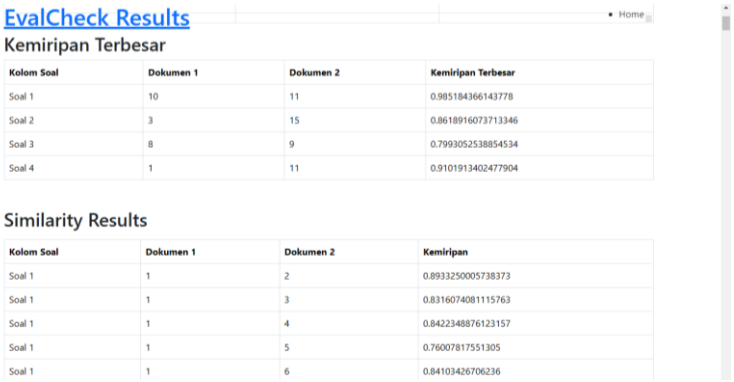
• Home

Summary Results

Kolom Soal	Dokumen	Ringkasan
Soal 1	1	dinotasikan dimana anggota tidakmemiliki kosong
Soal 1	2	dimana dilambangkan anggota tidakmemiliki
Soal 1	3	tidakmemuat didalamnya satupun elemen anggota
Soal 1	4	objek apapun bentuk disimbolkan tidakada elemen anggota
Soal 1	5	matematika huruf alfabet vokal konsonan kategori dilambangkan kurung kurawal notasi tanda tidakada simbol
Soal 1	6	menggambarkan konteks situasi dianggap sepenuhnya ø dalamnya elemen kurung kurawal tanda tidakada simbol
Soal 1	7	adalan tidakberisi dilambangkan tanda anggota tidakmemiliki
Soal 1	8	ganjil habis dibagi bilangan contohnya penulisannya tidakada b anggota
Soal 1	9	istilah nihil ketiga mengacu hampa satupun null set kardinal kurung kurawal
Soal 1	10	nama nilai null set ditulis kardinal anggota tidakmemiliki

Gambar 7 Halaman Hasil Ringkasan

Gambar 7 menunjukkan halaman hasil ringkasan. File dataset yang telah di unggah kedalam sistem akan diproses dan menghasilkan output berupa tabel dengan rincian terdapat atribut kolom soal, dokumen dan ringkasan.


EvalCheck Results

• Home

Kemiripan Terbesar

Kolom Soal	Dokumen 1	Dokumen 2	Kemiripan Terbesar
Soal 1	10	11	0.985184366143778
Soal 2	3	15	0.8618916073713346
Soal 3	8	9	0.7993052538854534
Soal 4	1	11	0.9101913402477904

Similarity Results

Kolom Soal	Dokumen 1	Dokumen 2	Kemiripan
Soal 1	1	2	0.8933250005738373
Soal 1	1	3	0.8316074081115763
Soal 1	1	4	0.8422348876123157
Soal 1	1	5	0.76007817551305
Soal 1	1	6	0.84103426706236

Gambar 8 Halaman Hasil Kemiripan Ringkasan

Gambar 8 menunjukkan hasil kemiripan ringkasan. Selain menampilkan hasil ringkasan, sistem juga menampilkan hasil pengecekan kemiripan terbesar dari masing masing soal. Dari seluruh dataset untuk masing masing soal dilakukan perbandingan kemiripan jawaban yang direpresentasikan kedalam bentuk tabel.

3.7 Evaluasi Sistem

Evaluasi sistem yang dilakukan yang dilakukan dalan penelitian ini melalui dua bagian. Evaluasi hasil ringkasan dilakukan untuk mengevaluasi hasil ringkasan yang dihasilkan oleh sistem kemudian untuk evaluasi hasil perbandingan kesamaan jawaban dilakukan oleh pakar.

1. **Evaluasi Hasil Ringkasan**
   Hasil ringkasan yang dihasilkan setelah mengurutkan bobot kata dari tertinggi hingga terendah kemudian dilakukan evaluasi. Evaluasi ringkasan dilakukan dengan membandingkan ringkasan yang terbentuk dari fungsi dengan *library* dan tanpa *library*

menggunakan metode *cosine similarity*. Masing-masing hasil ringkasan dibandingkan satu sama lain sehinggap memperoleh nilai kemiripan seperti ditunjukkan pada tabel 9.

Tabel 9 Nilai Kesamaan Ringkasan dengan dan tanpa *Library*

Dokumen ke	A1	A2	A3	A4
1	1,000000	1,000000	0,901111	1,000000
2	1,000000	1,000000	1,000000	1,000000
3	1,000000	1,000000	1,000000	0,920909
4	1,000000	0,905826	1,000000	0,938223
5	1,000000	1,000000	1,000000	1,000000
6	1,000000	0,901111	0,920909	0,883128
7	1,000000	0,669419	1,000000	0,799519
8	1,000000	1,000000	1,000000	1,000000
9	1,000000	1,000000	1,000000	1,000000
10	0,779915	1,000000	1,000000	0,842544
11	0,752320	1,000000	1,000000	0,867364
12	1,000000	0,858656	1,000000	0,503103
13	1,000000	1,000000	0,858656	0,846647
14	1,000000	0,835050	1,000000	1,000000
15	1,000000	1,000000	0,779915	0,883128

Tabel 9 menampilkan niai kesamaan ringkasan. Berdasarkan nilai yang tertera pada tabel 9 dapat dilihat untuk jawaban nomor 1, terdapat 13 dari 15 ringkasan sistem memiliki nilai kemiripan sebesar 1 yang berarti hasil ringkasannya sama, hanya terdapat 2 dokumen yang nilai kemiripannya mendekati 1 namun masih diatas 0,7 yang berarti dokumen masih memiliki kemiripan yang lumayan mirip. Begitu juga untuk jawaban nomor 2 terdapat 10 soal dengan nilai kemiripan 1 dan 5 lainnya dedngan nilai kemiripan diatas 0,8. Pada jawaban nomor 3 terdapat 11 jawaban nilai kemiripannya 1 dan 4 jawaban dengan nilai kemiripan diatas 0,7. Serta untuk jawaban nomor 4 terdapat 6 jawaban dengan nilai kemiripan 1 dan 9 jawaban yang nilai kemiripannya mendekati 1. Berdasarkan hasil perbandingan tersebut dapat diperoleh akurasi dari ringkasan yang dihasilkan dengan menerapan persamaan 4 dengan hasil evaluasi seperti yang ditunjukkan tabel 10.

Tabel 10 Hasil Evaluasi Ringkasan

Kode soal	Akurasi (%)
A1	96,8%
A2	94,4%
A3	96,4%
A4	89,8%
Rata-rata	**94,4%**

Berdasarkan tabel 10 dapat diketahui bahwa rata rata akurasi yang dihasilkan adalah sebesar 94,4%. Hal tersebut berarti kedua hasil ringkasan yang dibandingkan memiliki kesamaan antara satu sama lain yang cukup tinggi.

2. **Evaluasi Pakar**

Untuk selanjutnya untuk evaluasi hasil perbandingan kemiripan jawaban dilakukan dengan cara membandingkan dokumen jawaban secara manual kemudian diberikan skor kemiripan dengan skala A-E dimana nilai dengan skala A dan B artinya pasangan dokumen yang dibandingkan memiliki kemiripan (mendekati 1), serta nilai dengan skala

p-ISSN: 2301-5373
e-ISSN: 2654-5101

C, D dan E artinya pasangan dokumen tidak memiliki kemiripan (mendekati 0). Adapun skala kemiripan yang digunakan adalah sebagai berikut.
A = 0,81 - 1,00
B = 0,61 - 0,80
C = 0,41 - 0,60
D = 0,21 - 0,40
E = 0 - 0,20
Dalam hal ini pasangan dokumen dinyatakan kemiripannya dengan ketentuan jika skor kemiripan sistem bernilai minimal 0,80 dan skor kemiripan yang diberikan pakar dengan skala minimal B maka kemiripan antara dokumen dinyatakan “Sesuai”. Sebaliknya jika skor kemiripan sistem bernilai 0,80 kebawah dan skor kemiripan yang diberikan pakar adalah C, D dan E maka kemiripan antara dokumen dinyatakan “Tidak Sesuai”. Area yang diwarnai hijau menandakan kemiripan dokumen “Sesuai”. Area yang diwarnai kuning menandakan kemiripan dokumen “Tidak Sesuai”. Perbandingan nilai kemiripan ditunjukkan oleh gambar 9.

doc	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	0,81 - 1,00	0,81 - 1,00	0,81 - 1,00	0,21 - 0,40	0,61 - 0,80	0,61 - 0,80	0,41 - 0,60	0,41 - 0,60	0,61 - 0,80	0,41 - 0,60	0,61 - 0,80	0,21 - 0,40	0,81 - 1,00	0,21 - 0,40
2	0,893325	0	0,81 - 1,00	0,41 - 0,60	0,21 - 0,40	0,21 - 0,40	0,61 - 0,80	0,41 - 0,60	0,41 - 0,60	0,61 - 0,80	0,61 - 0,80	0,61 - 0,80	0,41 - 0,60	0,81 - 1,00	0,41 - 0,60
3	0,831607	0,755929	0	0,61 - 0,80	0,21 - 0,40	0,21 - 0,40	0,61 - 0,80	0,21 - 0,40	0,41 - 0,60	0,61 - 0,80	0,61 - 0,80	0,61 - 0,80	0,41 - 0,60	0,61 - 0,80	0,21 - 0,40
4	0,842235	0,808122	0,801784	0	0,41 - 0,60	0,41 - 0,60	0,81 - 1,00	0,41 - 0,60	0,41 - 0,60	0,61 - 0,80	0,81 - 1,00	0,61 - 0,80	0,41 - 0,60	0,61 - 0,80	0,41 - 0,60
5	0,760078	0,696143	0,701646	0,726651	0	0,41 - 0,60	0,41 - 0,60	0,21 - 0,40	0,21 - 0,40	0,21 - 0,40	0,21 - 0,40	0,21 - 0,40	0,61 - 0,80	0,21 - 0,40	0,41 - 0,60
6	0,841034	0,726273	0,800641	0,813125	0,814562	0	0,21 - 0,40	0,21 - 0,40	0,41 - 0,60	0,41 - 0,60	0,41 - 0,60	0,21 - 0,40	0,41 - 0,60	0,21 - 0,40	0,21 - 0,40
7	0,891338	0,908688	0,801388	0,831522	0,769015	0,815239	0	0,41 - 0,60	0,61 - 0,80	0,21 - 0,40	0,61 - 0,80	0,41 - 0,60	0,21 - 0,40	0,61 - 0,80	0,61 - 0,80
8	0,5336	0,542105	0,478091	0,543045	0,44028	0,478474	0,540899	0	0,21 - 0,40	0,41 - 0,60	0,41 - 0,60	0,41 - 0,60	0,21 - 0,40	0,41 - 0,60	0,21 - 0,40
9	0,680883	0,579066	0,667823	0,629941	0,682191	0,73598	0,666667	0,366234	0	0,41 - 0,60	0,41 - 0,60	0,61 - 0,80	0,41 - 0,60	0,41 - 0,60	0,61 - 0,80
10	0,859855	0,779194	0,768029	0,777844	0,701969	0,776736	0,823193	0,492805	0,714577	0	0,81 - 1,00	0,81 - 1,00	0,41 - 0,60	0,61 - 0,80	0,21 - 0,40
11	0,872786	0,790912	0,779579	0,789542	0,712525	0,788417	0,835573	0,500216	0,725324	0,985184	0	0,61 - 0,80	0,41 - 0,60	0,41 - 0,60	0,41 - 0,60
12	0,849548	0,774382	0,826718	0,835766	0,794435	0,863353	0,884492	0,489762	0,737076	0,784599	0,796398	0	0,21 - 0,40	0,81 - 1,00	0,41 - 0,60
13	0,811666	0,778792	0,749269	0,750939	0,739296	0,779864	0,82121	0,470162	0,640191	0,772328	0,783943	0,807957	0	0,41 - 0,60	0,41 - 0,60
14	0,815088	0,828079	0,730297	0,829515	0,640513	0,701646	0,826236	0,523723	0,602464	0,797053	0,80904	0,787499	0,718185	0	0,21 - 0,40
15	0,858137	0,796819	0,808138	0,845154	0,7396	0,860828	0,834799	0,579546	0,645975	0,818096	0,830399	0,818393	0,789799	0,80829	0

Gambar 9 Hasil Perbandingan Jawaban oleh Pakar

Gambar 9 menunjukkan hasil perbandingan jawaban yang dilakukan oleh pakar. Berdasarkan hasil evaluasi yang ditampilkan pada gambar 9 dapat dilihat bahwa dari 105 pasangan dokumen jawaban terdapat 67 pasangan jawaban yang dikatakan memiliki kemiripan dan 38 lainnya dinyatakan kurang memiliki kemiripan. Dari 105 data sistem dapat melakukan pengecekan dengan benar sebayak 67 data. Hal tersebut berarti hasil perbandingan kesamaan yang dihasilkan sistem sebesar 63,80% dari keseluruhan.

Kesimpulan

Penelitian yang dilakukan ini bertujuan untuk memudahkan dalam mengetahui perbandingan kesamaan tugas mahasiswa. Adapun kesimpulan yang diperoleh dari hasil penelitian ini adalah sebagai berikut.

1. Berdasarkan hasil penelitian sistem menghasilkan mampu menghasilkan ringkasan yang terdiri dari bobot kata tertinggi dengan hasil evaluasi menggunakan *cosine similarity* untuk membandingkan apakah ringkasan yang dihasilkan sama dengan ringkasan yang dibuat dengan menggunakan *library.* Hasil evaluasi menunjukkan nilai akurasi sebesar 94,4%. Hal tersebut berarti kedua hasil ringkasan yang dibandingkan memiliki kesamaan antara satu sama lain yang cukup tinggi.
2. Berdasarkan hasil pengujian terhadap kesamaan jawaban yang dilakukan oleh pakar diperoleh hasil bahwa sistem dapat melakukan perbandingan kesamaan dari 105 data terdapat 67 hasil perbandingan yang setara dengan 63,80%. Hal tersebut karena sistem hanya membandingkan berdasarkan kata kata yang terdapat dalam ringkasan tidak berdasarkan makna katanya.

References

[1] F. M. Kundi, M. Z. Asghar, S. R. Zahra, S. Ahmad, dan A. Khan, "A review of *text summarization*," *language*, vol. 6, no. 7, hlm. 8, 2014.
[2] E. L. Amalia, A. J. Jumadi, I. A. Mashudi, dan D. W. Wibowo, "Analisis Metode *Cosine Similarity* Pada Aplikasi Ujian *Online* Esai Otomatis (Studi Kasus JTI Polinema)," *J. Teknol. Inf. Dan Ilmu Komput. JTIIK*, vol. 8, no. 2, 2021.
[3] R. P. Pratama, M. Faisal, dan A. Hanani, "Deteksi Plagiarisme pada Dokumen Jurnal Menggunakan Metode *Cosine Similarity*," *SMARTICS J.*, vol. 5, no. 1, hlm. 22–26, 2019.
[4] S. Agustian dan S. Ramadhani, "Peringkasan teks otomatis (automated *text summarization*) pada artikel berbahasa indonesia menggunakan algoritma lexrank," *J. CoSciTech Comput. Sci. Inf. Technol.*, vol. 3, no. 3, hlm. 371–381, 2022.
[5] Y. M. Sari dan N. S. Fatonah, "Peringkasan Teks Otomatis pada Modul Pembelajaran Berbahasa Indonesia Menggunakan Metode Cross Latent Semantic Analysis (CLSA)," *JEPIN J. Edukasi Dan Penelit. Inform.*, vol. 7, no. 2, hlm. 153–159, 2021.
[6] Y. Findawati dan M. A. Rosid, "Buku Ajar *Text* Mining," *Umsida Press*, hlm. 1–123, 2020.
[7] M. A. Rosid, A. S. Fitrani, I. R. I. Astutik, N. I. Mulloh, dan H. A. Gozali, "Improving *text preprocessing* for student complaint document classification using sastrawi," dalam *IOP Conference Series: Materials Science and Engineering*, IOP Publishing, 2020, hlm. 012017.
[8] A. Tabassum dan R. R. Patil, "A survey on *text* pre-processing & feature extraction techniques in natural language processing," *Int. Res. J. Eng. Technol. IRJET*, vol. 7, no. 06, hlm. 4864–4867, 2020.
[9] S. Khomsah dan A. S. Aribowo, "*Text-preprocessing* model youtube comments in indonesian," *J. RESTI Rekayasa Sist. Dan Teknol. Inf.*, vol. 4, no. 4, hlm. 648–654, 2020.
[10] S. Robertson, "Understanding *inverse* document *frequency*: on theoretical arguments for *IDF*," *J. Doc.*, vol. 60, no. 5, hlm. 503–520, 2004.
[11] N. K. Nagwani dan S. Verma, "A frequent *term* and semantic *similarity* based single document *text summarization* algorithm," *Int. J. Comput. Appl.*, vol. 17, no. 2, hlm. 36–40, 2011.
[12] I. W. A. Setyadi, D. C. Khrisne, dan I. M. A. Suyadnya, "*Automatic Text Summarization* Menggunakan Metode Graph dan Ant Colony Optimization," *vol*, vol. 17, hlm. 124–130, 2018.