PENERAPAN ALGORITMA *COSINE SIMILARITY* DALAM EFEKTIFITAS PENGACAKAN SOAL UJIAN *ONLINE*

SKRIPSI

Diajukan sebagai Salah Satu Syarat untuk Menyusun Skripsi
Program Studi Informatika

AIDHIL PRIMA ABDIGUNA
1058411100920

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH MAKASSAR
2025

PENGESAHAN

Skripsi atas nama **Aidhil Prima Abdiguna** dengan nomor induk Mahasiswa **105 84 11009 20**, dinyatakan diterima dan disahkan oleh Panitia Ujian Tugas Akhir/Skripsi sesuai dengan Surat Keputusan Dekan Fakultas Teknik Universitas Muhammadiyah Makassar Nomor : 001/SK-Y/55202/091004/2025, sebagai salah satu syarat guna memperoleh gelar Sarjana Teknik pada Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar pada hari Senin, 17 Februari 2025.

Panitia Ujian :

Makassar, 18 Syaban 1446 H / 17 Februari 2025 M

1. Pengawas Umum
   a. Rektor Universitas Muhammadiyah Makassar
      Dr. Ir. H. Abd. Rakhim Nanda, ST.,MT.,IPU : .....
   b. Dekan Fakultas Teknik Universitas Hasanuddin
      Prof. Dr. Eng. Muhammad Isran Ramli, S.T., M.T.,ASEAN.,Eng : .....
2. Penguji
   a. Ketua : Desi Anggreani, S.Kom., MT. : .....
   b. Sekertaris : Titin Wahyuni, S.Pd., MT. : .....
3. Anggota : 1. Fahrim Irhamma Rahman, S.Kom., M.T : .....
   2. Muhyiddin A.M. Hayat, S.Kom, M.T : .....
   3. Chyquitha Danuputri, S.Kom., M.T : .....

Mengetahui :

Pembimbing I	Pembimbing II
Lukman, S.Kom, MT.	**Rizki Yusliana Bakti, S.T., M.T.**

Dekan

Dr. Ir. Hj. Nurnawaty, S.T., M.T.,IPM
NBM : 795 108

HALAMAN PENGESAHAN

Tugas Akhir ini diajukan untuk memenuhi syarat ujian guna memperoleh gelar Sarjana Komputer (S.Kom) Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar.

Judul Skripsi : **PENERAPAN ALGORIMA COSINE SIMILARITY DALAM EFEKTIFITAS PENGACAKAN SOAL UJIAN ONLINE**

Nama : AIDHIL PRIMA ABDIGUNA

Stambuk : 105 84 11009 20

Makassar, 18 Februari 2025

Telah Diperiksa dan Disetujui
Oleh Dosen Pembimbing;

Pembimbing I	Pembimbing II
Lukman, S.Kom, MT.	**Rizki Yusliana Bakti, S.T., M.T.**

Mengetahui,
Ketua Prodi Informatika

Muhyiddin A M Hayat, S.Kom., M.T.
NBM : 1504 577

ABSTRAK

AIDHIL PRIMA ABDI GUNA. Penerapan Algoritma *Cosine Similarity* dalam Efektifitas Pengacakan Soal Ujian *Online* (dibimbing oleh Rizki Yusliana Bakti, S.T., M.T dan Lukman, SKM, S.Kom, MT).

Pengacakan soal ujian *online* yang efektif merupakan tantangan penting dalam memastikan keadilan dan keakuratan dalam distribusi soal. Penelitian ini bertujuan untuk mengetahui bagaimana algoritma *Cosine Similarity* dapat diterapkan dalam sistem pengacakan soal ujian online serta mengevaluasi efektifitasnya dalam pendistribusian soal. Metode *Term Frequency-Inverse Document Frequency* (TF-IDF) untuk merepresentasikan soal dalam bentuk vektor numerik sebelum dilakukan perhitungan nilai kesamaan oleh algoritma *Cosine Similarity*, serta metode *Mean Absolute Error* (MAE) dan *Root Mean Squared Error* (RMSE) untuk memvalidasi efektifitas hasil pengacakan. Hasil serta kesimpulan dari penelitian menunjukkan bahwa penerapan algoritma *Cosine Similarity* dalam sistem pengacakan soal dapat dilakukan dengan sebelumnya menerapkan tahap *preprocessing data* dan *Term Frequency-Inverse Document Frequency* serta hanya digunakan sebelum tahap pengacakan, dan efektifitas algoritma ini dalam distribusi dinilai efektif karena selisih nilai aktual dan nilai prediksi yang kecil. Dengan demikian, algoritma ini dapat menjadi solusi efektif dalam membantu sistem pengacakan soal ujian online namun masih memerlukan perbaikan untuk mencapai hasil yang lebih optimal.

Kata Kunci: *cosine similarity, mean absolute error*, *root mean squared error,* soal ujian *online*, *term frequency-inverse document frequency.*

ABSTRAC

AIDHIL PRIMA ABDI GUNA. Penerapan Algoritma *Cosine Similarity* dalam Efektifitas Pengacakan Soal Ujian *Online* (dibimbing oleh Rizki Yusliana Bakti, S.T., M.T dan Lukman, SKM, S.Kom, MT).

Effective randomization of online exam questions is an important challenge in ensuring fairness and accuracy in question distribution. This study aims to determine how the Cosine Similarity algorithm can be applied in an online exam question randomization system and evaluate its effectiveness in distributing questions. The Term Frequency-Inverse Document Frequency (TF-IDF) method to represent questions in the form of a numeric vector before calculating the similarity value by the Cosine Similarity algorithm, and the Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) methods to validate the effectiveness of the randomization results. The results and conclusions of the study indicate that the application of the Cosine Similarity algorithm in the question randomization system can be done by previously implementing the data preprocessing stage and Term Frequency-Inverse Document Frequency and only used before the randomization stage, and the effectiveness of this algorithm in distribution is effective because a small difference between the actual value and the predicted value. Thus, this algorithm can be an effective solution in helping the online exam question randomization system but still needs improvement to achieve more optimal results.

Keywords: *cosine similarity, mean absolute error, root mean squared error, randomization of online exam questions, term frequency-inverse document frequency.*

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Dengan penuh rasa syukur atas kehadirat Allah Subhanallahu Wa Ta'ala, atas nikmat iman, Islam, dan juga nikmat Kesehatan yang senantiasa terlimpahkan. Sehingga penulis dapat menyelesaikan Laporan Tugas Akhir yang berjudul **"Penerapan Algoritma *Cosine Similarity* dalam Efektifitas Pengacakan Soal Ujian *Online*"**. Tak lupa pula sholawat serta salam kepada junjungan Nabi Muhammad SAW, sang pembawa rahmat bagi semesta alam. Yang telah membawa kita dari Zaman jahiliyah menuju Zaman yang penuh berkah seperti sekarang ini.

Ucapan terima kasih yang tidak terhingga penulis sampaikan kepada semua yang telah membantu dan memberikan dukungan dalam penyusunan Laporan Tugas Akhir ini, khususnya:

1. Kepada kedua Orang Tua saya tercinta, Bapak **Ahman Yani** dan Ibu **Memoriana**, persembahan terima kasih yang tak terhingga atas esegala pengorbanan, kasih sayang, dan bimbingan, yang telah diberikan.
2. Ibu **Dr.Ir.Hj Nurnawati, S.T., M.T., I.P.M,** selaku Dekan Fakultas Teknik.
3. Bapak **Muh. Syafaat S Kuba, S.T., M.T,** selaku Wakil Dekan Fakultas Teknik.
4. Bapak **Muhyiddin A M Hayat S.Kom., M.T,** selaku Ketua Prodi Informatika.
5. Bapak **Lukman, S.Kom, MT,** selaku Dosen Pembimbing 1 Proposal.
6. Ibu **Rizki Yusliana Bakti, S.T., M.T,** Selaku Dosen Pembimbing 2 Proposal.
7. Dosen dan Staff Fakultas Teknik Universitas Muhammadiyah Makassar.
8. Teman-teman khususnya Angkatan 2020 Fakultas Teknik Universitas Muhammadiyah Makassar.
9. Teman-teman kelas A Angkatan 2020 Program Studi Informatika Universitas Muhammadiyah Makassar.
10. Senior sejurusan saya Kak **Andi Agung Dwi Arya, S.Kom,** atas bantuannya sejak awal saya masuk ke jurusan ini.

11. Teman baik saya **Ahmad Wildan Dzakki Adam**, yang telah banyak mmeberikan bantuan maupun beban sebelum, saat, maupun setelah pembuatan proposal ini.
12. Dan juga orang-orang yang tidak sempat saya sebutkan, terima kasih atas segala bantuan, semangat dan do'anya.

Penulis menyadari bahwa Laporan Tugas Akhir ini masih jauh dari kesempurnaan. Oleh karena itu, kritik dan saran yang membangun sangat penulis harapkan demi penyempurnaan laporan ini di masa depan. Harapan penulis, semoga Laporan Tugas Akhir ini dapat memberikan manfaat bagi penyandang disabilitas tunanetra dalam meningkatkan kemandirian dan kualitas hidup mereka. Akhir kata, tunanetra dalam meningkatkan kemandirian dan kualitas hidup mereka. Akhir kata, penulis mohon maaf atas segala kekurangan dan kekhilafan yang terdapat dalam Laporan Tugas Akhir ini.

Billahi fisabililhaq, fastabiqul khairat.
Wassalamualaikum Wr.Wb.

Makassar, Februari 2025
Penulis,

AIDHIL PRIMA ABDIGUNA

DAFTAR ISI

ABSTRAK ..... **i**
ABSTRAC ..... **ii**
KATA PENGANTAR ..... **iii**
DAFTAR ISI ..... **v**
DAFTAR GAMBAR ..... **vii**
DAFTAR TABEL ..... **ix**
DAFTAR LAMPIRAN ..... **x**
DAFTAR ISTILAH ..... **xi**
BAB I PENDAHULUAN **1**
A. Latar Belakang 1
B. Rumusan Masalah 3
C. Tujuan Penelitian 3
D. Manfaat Penelitian 3
E. Ruang Lingkup Penelitian 4
F. Sistematika Penulisan 4
BAB II TINJAUAN PUSTAKA **5**
A. Landasan Teori 5
B. Penelitian Terkait 11
C. Kerangka Pikir 15
BAB III METODE PENELITIAN **16**
A. Tempat dan Waktu Penelitian 16
B. Alat dan Bahan 16
C. Perancangan Sistem 17
D. Teknik Pengujian Sistem 19
E. Teknik Analisis Data 20
BAB IV HASIL DAN PEMBAHASAN **23**
A. Pengambilan Data 23
B. Pengujian dan Hasil Model 24
BAB V PENUTUP **43**
A. Kesimpulan 43

B. Saran 44
DAFTAR PUSTAKA 45
LAMPIRAN 48

DAFTAR GAMBAR

Gambar 1. Kerangka Pikir 15
Gambar 2. Flowchart Sistem 17
Gambar 3. Flowchart Representasi dan Perhitungan Nilai Soal 18
Gambar 4. Dataset soal di dalam MySQL 23
Gambar 5. Output Koneksi Database 24
Gambar 6. Output Dataset Soal dari Database 25
Gambar 7. Output Hasil Preprocessing Data 26
Gambar 8. Output Hasil TF-IDF Soal 27
Gambar 9. Output Hasil TF-IDF Kategori 27
Gambar 10. Output Dataframe Matrix Cosine Similarity Soal 29
Gambar 11. Output Dataframe Matrix Cosine Similarity Kategori 29
Gambar 12. Output Nilai Cosine Similarity Terendah 30
Gambar 13. Output Nilai Cosine Similarity Tertinggi 31
Gambar 14. Output Soal-soal Identik 32
Gambar 15. Output Soal-soal Tidak Identik 32
Gambar 16. Output Hasil Pengacakan Cosine Similarity Ke-1 (Kategori) 33
Gambar 17. Output Hasil Pengacakan Cosine Similarity Ke-1 (Soal dan Rata-rata Nilai) 34
Gambar 18. Output Hasil Pengacakan Cosine Similarity Ke-10 (Kategori) 34
Gambar 19. Output Hasil Pengacakan Cosine Similarity Ke-10 (Soal dan Rata-rata Nilai) 34
Gambar 20. Output Distribusi Data Aktual dan Ideal serta MAE dan RMSE per Soal 36
Gambar 21. Data Aktual dan Ideal serta MAE dan RMSE untuk 10x pengacakan dan 50 soal 37
Gambar 22. Data Aktual dan Ideal serta MAE dan RMSE untuk 10x pengacakan dan 100 soal 38
Gambar 23. Data Aktual dan Ideal serta MAE dan RMSE untuk 50x pengacakan dan 50 soal 39

Gambar 24. Data Aktual dan Ideal serta MAE dan RMSE untuk 50x pengacakan dan 100 soal 40
Gambar 25. Data Aktual dan Ideal serta MAE dan RMSE untuk 100x pengacakan dan 50 soal 41
Gambar 26. Data Aktual dan Ideal serta MAE dan RMSE untuk 100x pengacakan dan 100 soal 42

DAFTAR TABEL

Table 1. Penelitian Terkait 13
Table 2. Jumlah Data Soal 24

DAFTAR LAMPIRAN

Lampiran 1. Source Code..... 48

DAFTAR ISTILAH

Computer-Based Test	Ujian yang dilakukan secara elektronik menggunakan komputer sebagai *platform* untuk menyajikan soal dan mencatat jawaban peserta ujian.
Paper-Based Test	Ujian yang dilakukan secara konvensional dengan menggunakan kertas sebagai media untuk soal dan jawaban.
Online	Istilah ini mengacu pada keadaan di mana suatu sistem atau perangkat terhubung ke jaringan internet, sehingga memungkinkan akses atau komunikasi secara real-time melalui jaringan tersebut. Dalam konteks ujian *online*, istilah ini mengacu pada pelaksanaan ujian melalui platform berbasis web yang membutuhkan koneksi ke internet.
Cosine Similarity	Algoritma yang digunakan untuk mengukur kesamaan antara dua vektor dalam ruang dimensi, sering diterapkan dalam teks untuk membandingkan kesamaan antar dokumen atau soal.
TF-IDF	*Term Frequency-Inverse Document Frequency*, Metode untuk menghitung kepentingan suatu kata dalam sebuah dokumen berdasarkan seberapa sering kata tersebut muncul di dokumen dan seberapa jarang muncul di dokumen lain.
Mean Absolute Error	MAE adalah metrik yang digunakan untuk menghitung selisih absolut rata-rata antara nilai aktual dan prediksi.

Root Mean Squared Error	RMSE adalah metrik yang dapat digunakan untuk menghitung akar rata-rata kuadrat selisih antara nilai aktual dan prediksi.
Hardware	Perangkat fisik yang digunakan dalam sistem komputer, termasuk prosesor, memori, dan perangkat penyimpanan yang mendukung operasi perangkat lunak.
Software	Program komputer yang berisi instruksi untuk menjalankan tugas-tugas spesifik, seperti aplikasi untuk mengelola ujian *online*.
Dataset	Sekumpulan data yang terorganisir, sering digunakan dalam penelitian atau pengembangan sistem untuk tujuan analisis dan pengujian algoritma.
Flowchart	Diagram yang digunakan untuk memvisualisasikan alur kerja atau langkah-langkah dalam suatu proses, membantu dalam pemahaman sistem atau algoritma.
Database	Struktur terorganisir yang digunakan untuk menyimpan, mengelola, dan mengambil data secara efisien. *Database* biasanya digunakan untuk menyimpan soal ujian *online*.
MySQL	Sistem manajemen basis data relasional yang menggunakan *SQL* untuk mengakses dan mengelola data, sering digunakan dalam pengembangan aplikasi web.
Jupyter Notebook	Alat interaktif untuk menulis kode, menjalankan analisis, dan mendokumentasikan hasil, sering digunakan dalam penelitian data ilmiah dan pengembangan algoritma.

SQLalchemy	Pustaka *Python* yang menggunakan konsep *Object-Relational Mapping* (ORM) untuk menghubungkan aplikasi ke database.
Preprocessing	Proses persiapan data sebelum digunakan untuk model pembelajaran mesin atau analisis.
Random Shuffle	Merupakan proses pengacakan yang dilakukan secara acak untuk menyusun ulang urutan elemen dalam suatu data, tanpa memperhatikan urutan sebelumnya.
Dataframe	Dataframe adalah struktur data dua dimensi dalam pustaka pandas yang menyerupai tabel dengan baris dan kolom, digunakan untuk menyimpan, mengelola, dan menganalisis data secara efisien.

BAB I
PENDAHULUAN

A. Latar Belakang

Salah satu bidang yang memainkan peran yang sangat penting dalam pembangunan suatu negara adalah pendidikan. Berbagai aspek pendidikan telah berubah sebagai hasil dari kemajuan teknologi informasi, termasuk salah satunya metode ujian. Ujian berbasis kertas atau *Paper-Based Test* yang sejak dulu telah menjadi metode dalam pelaksanaan ujian perlahan-lahan mulai teralihkan dengan menggunakan metode ujian berbasis komputer atau *Computer-based Test* (Setiawan et al., 2022). Ujian yang menggunakan metode berbasis komputer ataupun perangkat lainnya biasanya dilakukan secara daring atau *online* menawarkan berbagai keuntungan, termasuk diantaranya lebih mudah dalam pendistribusian soal, lebih efisien dalam hal waktu yang dihabiskan, dan lebih murah untuk biaya operasi karena tidak harus mengeluarkan biaya tambahan untuk kertas dan juga untuk biaya lainnya seperti percetakan (Wardani, 2021).

Salah satu tantangan utama dalam metode ujian berbasis komputer adalah dalam pengacakan soal untuk memastikan keadilan dan mengurangi kemungkinan kecurangan ketika ujian terebut sedang berlangsung. Pengacakan soal yang efektif, tidak hanya harus memastikan pendistribusian soal yang diberikan tetapi juga harus mempertimbangkan kesamarataan antara soal-soal yang diberikan (Hanif Ridwannulloh, 2021).

Dalam penelitian ini, algoritma *cosine similarity* diterapkan sebagai metode untuk pengacakan soal yang menawarkan keunggulan tertentu dibandingkan algoritma pengacakan lainnya. Misalnya algoritma pengacakan tradisional, seperti *Fisher-Yates Shuffle*, mengacak soal secara acak tanpa memperhatikan kesamaan antar soal, tetapi algoritma *cosine similarity* memungkinkan pengacakan yang lebih baik dengan mempertimbangkan tingkat kesamaan antara soal-soal yang diberikan (Qhorifadillah et al., 2022).

Algoritma *cosine similarity*, yang biasanya digunakan untuk mengukur kesamaan antara dua vektor, diadaptasi untuk mengukur kesamaan antara soal-soal ujian (Rusdiyanto et al., 2022). Dalam penerapan algoritma *cosine* similarity tidak

lepas dari penggunaan metode *Term Frequency-Inverse Document Frequency* (TF-IDF). Yang dimana metode TF-IDF adalah kunci untuk mengukur kesamaan antara dua vektor dalam penggunaan algoritma *cosine similarity* (Andriani & Wibowo, 2021). Sebelum perhitungan kesamaan dapat dilakukan, setiap soal ujian harus direpresentasi menjadi vektorisasi numerik, di mana TF-IDF digunakan untuk menghitung frekuensi setiap kata antar soal. Dengan mengidentifikasi frekuensi kata-kata yang muncul dalam soal, metode TF-IDF ini memungkinkan perhitungan kesamaan antar soal yang lebih akurat dalam *cosine similarity* (Darwis et al., 2020).

Langkah penting dalam mengevaluasi kinerja algoritma dalam sistem pengacakan soal ujian yang menggunakan algoritma *Cosine Similarity* adalah memastikan bahwa data prediksi dan data aktual secara konsisten mendekati. Untuk mengetahui tingkat kesalahan prediksi, diadaptasi metode validasi *Mean Absolute Error* (MAE) dan *Root Mean Squared Error* (RMSE) (Sukmaningtyas et al., 2024). MAE menghitung rata-rata kesalahan absolut antara data aktual dan data prediksi, memberikan gambaran sederhana tentang jarak rata-rata kesalahan. RMSE, di sisi lain, memberikan bobot lebih besar pada kesalahan besar dengan menghitung akar kuadrat rata-rata dari kuadrat selisih, sehingga lebih sensitif terhadap anomali. Validasi ini memastikan bahwa soal-soal yang dibuat memenuhi standar pengacakan yang ideal dengan mempertahankan tingkat kesamaan sesuai tujuan algoritma. Ini membantu meningkatkan efektifitas pengacakan soal untuk ujian yang lebih adil dan variatif (Mulyana & Marjuki, 2022).

Berdasarkan latar belakang yang telah diuraikan, maka penulis mengambil judul "Penerapan Algoritma *Cosine Similarity* dalam Efektifitas Pengacakan Soal Ujian *Online*" yang dimana algoritma dan juga metode yang digunakan dalam penelitian ini tidak hanya akan melakukan pengacakan soal, namun memungkinkan pengacakan yang lebih variatif dan tidak ada soal yang sama sehingga memberikan hasil pengacakan soal yang lebih seimbang.

B. Rumusan Masalah

1. Bagaimana cara algoritma *cosine similarity* bisa diterapkan dalam pengacakan soal ujian *online*?
2. Apakah penerapan algoritma *cosine similarity* efektif digunakan dalam sistem pengacakan soal ujian online?

C. Tujuan Penelitian

1. Untuk mengetahui bagaimana cara algoritma *cosine similarity* bisa diterapkan dalam pengacakan soal ujian *online*.
2. Untuk mengetahui apakah penerapan algoritma *cosine similarity* efektif digunakan dalam sistem pengacakan soal ujian *online*.

D. Manfaat Penelitian

Manfaat dari penelitian ini adalah:

1. Bagi penulis, penelitian ini akan memberikan pemahaman yang mendalam tentang algoritma *cosine similarity* dan bagaimana penerapannya dalam pengacakan soal ujian, serta memberikan pemahaman dalam menerapkan teknologi ke dalam sistem pendidikan.
2. Bagi Universitas, hasil penelitian ini dapat meningkatkan reputasi Universitas sebagai Lembaga yang mendukung inovasi dalam pengacakan soal ujian dan juga dapat sebagai sumber referensi untuk penelitian selanjutnya yang berkaitan dengan cosine similarity maupun pengacakan soal ujian *online*.
3. Bagi pembaca, memberikan pemahaman tentang algoritma *cosine* similarity dan bagaimana penerapan algoritma ini bekerja ketika digunakan dalam sistem pengacakan soal ujian *online*, juga sebagai referensi untuk mengeksplorasi lebih lanjut tentang penelitian ini.

E. Ruang Lingkup Penelitian

Agar penelitian tetap berada pada jalur yang telah ditentukan, maka penelitian ini dibuatkan ruang lingkup penelitian, diantaranya yaitu:

1. Cara penerapan algoritma *cosine similarity* dalam pengacakan soal ujian *online*.
2. Mengevaluasi efektifitas penerapan algoritma *cosine similarity* ketika digunakan dalam sistem pengacakan soal ujian *online*

F. Sistematika Penulisan

Secara garis besar, penulisan dari laporan tugas akhir ini dibagi ke beberapa bab yang tersusun, yaitu:

BAB I PENDAHULUAN

Pada bab ini dijelaskan tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Pada bab ini dijelaskan tentang teori-teori yang menjadi landasan penulisan dalam pelaksanakan skirpsi.

BAB III METODE PENELITIAN

Pada bab ini membahas tentang metode penelitian dan alat yang digunakan untuk pembuatan sistem.

BAB IV HASIL PENELITIAN

Pada bab ini dijelaskan hasil penelitian yang telah dilakukan sebelumnya, di bab ini dijelaskan tentang hasil penelitian juga pengujian.

BAB V PENUTUP

Pada bab ini menjelaskan kesimpulan dari hasil penelitian yang telah dilakukan.

BAB II
TINJAUAN PUSTAKA

A. Landasan Teori

1. Ujian

Ujian ialah cara untuk mengukur kemampuan peserta didik, ujian konvensional, yang banyak digunakan saat ini, membutuhkan banyak waktu, biaya, dan tenaga kerja untuk menyiapkan ujian dengan benar, dan membutuhkan banyak kertas. Tahap yang paling sulit dari persiapan ujian konvensional adalah menggandakan lembar jawab kertas (LJK) sebanyak jumlah mata pelajaran di setiap tingkat, yang membutuhkan banyak kertas dan biaya (Daulay & Ekadiansyah, 2024).

Ujian *online* sendiri ialah suatu metode penilaian di mana peserta dapat mengikuti ujian melalui internet. Peserta dapat melakukannya di mana saja, asalkan mereka memiliki perangkat yang dapat terhubung ke internet. Sistem ujian *online* biasanya melibatkan serangkaian pertanyaan yang harus dijawab dalam jangka waktu tertentu, dan sistem otomatis dapat menilai dan mendistribusikan hasil langsung (Prakarsa et al., 2020).

2. Algoritma *Cosine Similarity*

Salah satu cara untuk menunjukkan bahwa dua atau lebih objek hampir sama adalah dengan menggunakan algoritma *cosine similarity*. Metode ini bergantung pada pengukuran kemiripan ruang vektor. Masing-masing vektor mewakili dua bagian atau lebih dari satu dokumen, dan tingkat kemiripannya dapat dihitung dengan menggunakan kata kunci (Qhorifadillah et al., 2022).

Salah satu metode untuk mengukur kesamaan antara 2 vektor dalam algoritma ini adalah dengan menghitung cosinus sudut di antara keduanya. Nilai *cosine similarity* berkisar antara 0 hingga 1, di mana:

a. 1 berarti vektor-vektor tersebut identik.

b. 0 berarti vektor-vektor tersebut tidak ada kesamaan.

Formula *cosine similarity* antara dua vektor A dan B adalah:

$$Cosine\ Similarity = \cos\theta = \frac{A \cdot B}{\|A\| \times \|B\|} \text{.....(1)}$$

Di mana:

a. $A \cdot B$ = hasil kali dot antara vektor A dan B.
b. $\|A\|$ dan $\|B\|$ = magnitudo dari vektor A dan B.
c. θ = sudut antara 2 vektor.

Dalam konteks pengacakan soal ujian *online*, penerapan *cosine similarity* dapat dilakukan dengan perhitungan kesamaan nilainya, *cosine similarity* digunakan untuk mengukur tingkat kesamaan yang ada antara masing-masing pasangan soal dan kategori. Ini membantu memastikan bahwa soal-soal yang dipilih dari *database* tidak terlalu mirip satu sama lain.

3. Metode *Term Frequency-Inverse Document Frequency* (TF-IDF)

Term Frequency-Inverse Document Frequency atau TF-IDF adalah metode yang digunakan dalam teks mining untuk menilai seberapa penting suatu kata dalam sebuah dokumen, relatif terhadap sekumpulan dokumen (Azmi, 2022). TF-IDF digunakan untuk merepresentasikan kata-kata dalam bentuk vektor numerik, dan merupakan dasar dalam algoritma pengukuran kesamaan, seperti cosine similarity (Darwis et al., 2020).

Disini (t) diibaratkan kata sedangkan (d) adalah soal

a. Rumus TF-IDF:

$$TF - IDF(t,d) = TF(t,d) \times IDF(t,D) \text{ .....(1)}$$

1) TF(t, d) = Frekuensi kata t dalam soal d. Ini menunjukkan seberapa sering suatu kata muncul dalam soal tertentu.
2) IDF(t, D) = Invers frekuensi dokumen yang mengandung kata t. Ini menunjukkan seberapa jarang kata tersebut muncul di seluruh soal dalam kategori yang sama.

b. TF, mengukur frekuensi kemunculan suatu kata dalam soal tertentu. Jika sebuah kata sering muncul dalam soal, nilainya akan lebih tinggi.

$$\mathrm{TF(t,d)} = \frac{Jumlah\ kemunculan\ kata\ t\ dalam\ soal\ d}{Total\ kata\ dalam\ soal\ d} \text{.....(2)}$$

c. IDF, mengukur pentingnya suatu kata dalam kumpulan soal. Semakin banyak soal yang mengandung kata tersebut, semakin rendah nilai IDF-nya, karena kata tersebut dianggap umum.

$$\text{IDF}(\text{t}, \text{D}) = log\left(\frac{N}{1+df(t)}\right) \text{.....(3)}$$

1) N = jumlah total soal dalam kategori
2) df(t) = jumlah soal yang mengandung kata t

Adapun cara kerja dari metode ini ialah:

a. Hitung nilai TF dan IDF untuk setiap kata.
b. Kalikan nilai TF dan IDF untuk setiap kata.
c. Hasilnya adalah bobot setiap kata yang menunjukkan seberapa penting kata tersebut dalam sebuah soal.

Metode TF-IDF dapat digunakan untuk memvektorisasi tiap kata dalam *database* soal ke dalam bentuk numerik dan menganalisis soal ujian berdasarkan frekuensi kata yang muncul di dalamnya sebelum dihitung nilai kesamaannya oleh algoritma *cosine similarity*.

4. *Mean Absolut Error (MAE) dan Root Mean Squared Error (RMSE)*

Mean Absolute Error dan *Root Mean Squared Error* adalah dua metode untuk mengevaluasi evaluasi tingkat kesalahan dalam prediksi suatu model terhadap data aktual (Mulyana & Marjuki, 2022).

a. MAE

MAE adalah rata-rata dari nilai absolut selisih antara hasil prediksi dan nilai aktual, rumusnya:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|yi - \widehat{yi}| \text{.....(1)}$$

Di mana yi adalah nilai aktual, $\widehat{yi}$ adalah nilai prediksi, dan nnn adalah jumlah data. MAE memberikan gambaran langsung tentang tingkat kesalahan prediksi rata-rata tanpa tanda.

b. RMSE

RMSE adalah akar kuadrat dari rata-rata kuadrat selisih antara hasil prediksi dan nilai aktual, umusnya:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(yi - \widehat{yi})^2} \quad (1)$$

Karena kuadrat selisihnya, RMSE lebih rentan terhadap kesalahan besar, jadi perhatikan outlier atau prediksi yang sangat meleset.

Dalam penelitian ini, metode MAE dan RMSE digunakan dalam pengacakan soal ujian menggunakan algoritma *Cosine Similarity* untuk memvalidasi selisih antara hasil prediksi dibuat oleh algoritma dan data aktual. Validasi ini membantu mengukur seberapa jauh hasil pengacakan mendekati target yang diinginkan, memastikan pengacakan yang tepat, dan mengurangi kemungkinan pengacakan yang salah.

Untuk menentukan apakah hasil dari pengacakan dengan algoritma *cosine similarity* efektif atau tidak, kita perlu memvalidasinya, jika selisih hasil data prediksi dan data aktualnya, jika selisihnya berada diantara angka 0-1 maka bisa dikatakan pengacakan dengan algoritma tersebut efektif, jika berada diantara angka 1-3 maka bisa dikatakan kurang efektif, dan jika selisihnya lebih dari angka 3 maka bisa disimpulkan kalau algoritma tersebut tidak efektif digunakan dikarenakan selisih yang banyak (Hodson, 2022).

5. *Random Shuffle*

Random Shuffle atau fungsi *random* adalah bagian penting dari proses pengacakan ujian. Algoritma ini mengacak urutan indeks catatan atau tabel, teknik ini mirip dengan mengacak kartu, di mana setiap kartu diacak sedemikian rupa sehingga urutannya terdistribusi secara acak (Askar et al., 2023).

Proses pengacakan dengan menggunakan fungsi random menghasilkan pendistribusian secara acak, fungsi ini dapat digunakan untuk:

a. Mengacak urutan sehingga tidak ada pola yang jelas dalam distribusi soal,
b. Memilih secara acak dari *database* yang tersedia untuk menciptakan set yang unik.

c. Memastikan variasi, dengan mempertimbangkan faktor kesamaan atau kesetaraan tingkat kesulitan, dalam hal ini mengacu pada kategori.

Pengacakan digunakan dalam penelitian ini untuk menghasilkan variasi soal yang tidak dapat diprediksi oleh peserta, dan memastikan bahwa soal didistribusikan secara acak dan berbeda kepada tiap-tiap peserta. Namun, fungsi random memiliki batasan ketika digunakan secara murni tanpa metode tambahan. Dalam pengacakan soal berbasis algoritma, algoritma lain sering digunakan bersama dengan fungsi random yang dalam penelitian ini dikombinasikan dengan algoritma *cosine similarity* untuk mengontrol variasi soal agar tidak ada soal yang terlalu mirip satu sama lain ketika diambil dari *database*. Untuk sistem pengacakan dalam penelitian ini, digunakan sistem kombinasi, yaitu sistem yang mengambil objek tanpa memperhatikan urutannya namun nantinya akan divalidasi oleh algoritma *cosine similarity* jika objek atau soal yang diambil memiliki kemiripan satu sama lain.

6. Peluang

Peluang, juga dikenal sebagai probabilitas, adalah ukuran seberapa mungkin suatu peristiwa terjadi dalam sebuah eksperimen atau proses acak. Nilai 0 untuk peluang menunjukkan bahwa peristiwa tidak mungkin terjadi, sedangkan nilai 1 menunjukkan bahwa peristiwa pasti terjadi (Dafitri et al., 2023).

Rumus dasar peluang adalah:

$$P(A) = \frac{n(A)}{n(S)} \quad \text{.....(1)}$$

Keterangan:

n(A) = banyaknya anggota dalam himpunan kejadian A.

n(S) = banyaknya anggota dalam himpunan kejadian S.

Peluang dalam pengacakan soal ujian online dapat dimisalkan sebagai berikut, jika jumlah soal ada sebanyak 50 dan di dalam *database* terdapat soal dengan 7 kategori berbeda yang akan di distribusi secara merata ke 50 soal,

maka per kategori adakan mendapatkan jatah 7 soal dan ada salah satu dari 7 kategori tersebut yang memiliki jumlah soal sebanyak 8. Dan untuk memvalidasi apakah pengacakan soal itu merata pada tiap kategori soal, digunakanlah metode validasi MAE dan RMSE untuk melihat selisih antara data aktual dan data prediksinya.

7. Populasi dan Ruang Sampel

Populasi adalah himpunan entitas atau data yang digunakan untuk mengambil sampel secara acak. Populasi dapat didefinisikan sebagai keseluruhan komponen penelitian yang mencakup objek dan subjek yang memiliki karakteristik dan atribut tertentu (Candra Susanto et al., 2024).

Sampel secara sederhana adalah bagian dari populasi yang berfungsi sebagai sumber data penelitian. Dengan kata lain, sampel adalah sebagian dari populasi untuk menggambarkan seluruh populasi. Sampel selalu berasal dari ruang sampel. oleh karena itu, ruang sampel merupakan sumber utama dari mana sampel dihasilkan (Suriani et al., 2023).

Dalam ruang sampel, setiap elemen mewakili hasil yang mungkin dari percobaan acak. Peluang suatu peristiwa ditentukan dengan membandingkan jumlah elemen yang memenuhi kondisi peristiwa dengan jumlah total elemen dalam ruang sampel (Dafitri et al., 2023).

Dalam penelitian ini, Populasi adalah seluruh soal yang terdapat dalam *database*, sampel adalah masing-masing soal yang berada dalam database dan ruang sampel adalah *database* itu sendiri yang berisi kumpulan soal-soal yang akan digunakan dalam penelitian ini.

8. Substitusi dan Distribusi

Substitusi adalah ide yang digunakan untuk mengganti elemen tertentu dengan elemen lain yang setara atau relevan dalam situasi tertentu. Dalam matematika, substitusi sering digunakan untuk menyederhanakan persamaan, menghitung integral, atau melakukan transformasi tertentu dalam fungsi. Distribusi adalah pola atau penyebaran elemen dalam kumpulan data. Dalam

matematika, distribusi sering dikaitkan dengan probabilitas, yang menunjukkan kemungkinan terjadinya suatu peristiwa (Fanani et al., 2024).

Substitusi dapat digunakan untuk mengganti soal yang memiliki nilai *cosine similarity* yang sama, sedangkan distribusi digunakan untuk mendistribusikan pengambilan soal dari *database* serta mendistribusikan hasil pengecekan dan pengacakan nilai kesamaan dari algoritma *cosine similarity* untuk ditampilkan.

B. Penelitian Terkait

Hasil penelitian yang terkait mencakup ringkasan isi penelitian yang telah diterbitkan sebelumnya, baik dari segi kelebihan maupun kekurangan. Hasil ini dikumpulkan dari jurnal, makalah, atau penelitian dan dimaksudkan untuk memberikan masukan kepada peneliti saat mereka membuat kerangka kerja penelitian mereka.

1. Aplikasi Ujian *Online* Dan Penerapan Algoritma LCG Untuk Proses Pengacakan Soal Ujian Di Smk Negeri Tugumulyo (Rusdiyanto et al., 2022). Dari penelitian ini mengambil hasil kesimpulan sebagai berikut, yaitu Metode LCG dapat mengacak soal ujian serta memiliki berbagai variabel, dan juga menghasilkan pengacakan soal ujian yang berbeda, tetapi akan menghasilkan nilai pengacakan yang sama untuk variabel yang sama.
2. Implementasi Algoritma *Multiply With Carry Generator* (MWCG) Dalam Pengacakan Soal Ujian Semester Berbasis Web Pada SMKN 1 Kendari (Saputra et al., 2022). Menguji keacakan Algoritma *Multiply With Carry Generator* (MWCG) pada 10 siswa dengan 20 soal secara bersamaan, pengujian *Run Test* menghasilkan bentuk soal acak yang diterima. Hal ini menunjukkan bahwa Algoritma *Multiply With Carry Generator* dapat digunakan untuk mengacak soal.
3. Perancangan Aplikasi Bank Soal Berbasis Website dengan Algoritma *Fisher Yates Shuffle* dan *Cosine Similarity* (Qhorifadillah et al., 2022). Menarik kesimpulan bahwa rancangan aplikasi berjalan dengan baik

melalui pengujian blackbox yang memperoleh hasil 100%, dari 30 responden mendapatkan hasil 77,3% layak dan dari hasil uji pengguna mendapatkan hasil 84,5%.

4. Implementasi Metode *Multiplicative Random Number Generator* (MRNG) Pada Aplikasi Ujian Sekolah Berbasis Komputer (Prasetyadi & Budi Nugroho, 2020). Berdasarkan hasil analisa menunjukkan bahwa pembuatan sebuah aplikasi web yang memudahkan peserta ujian di SMK Nurcahaya Medan dapat menyelesaikan masalah pengacakan soal ujian dengan menggunakan algoritma *Multiplicative Random Number Generator* dan Simulasi. Dimulai dengan perancangan dan pembuatan sebuah aplikasi yang dapat membantu SMP Negeri 1 Pancurbatu Medan dalam pengacakan soal ujian, dilanjutkan dengan pengkodean dan penerapan metode *Multiplicative Random Number Generator* ke dalam pemrograman berbasis web. Kemudian dilakukan ujicoba terhadap beberapa siswa di SMK Nurcahaya Medan untuk mengevaluasi sejauh mana sistem yang dibangun memenuhi kebutuhan yang dibutuhkan.
5. Implementation of TF-IDF Algorithm and K-mean Clustering Method to Predict Words or Topics on Twitter (Darwis et al., 2020). Algoritma TF-IDF dan metode K-mean dapat digunakan untuk mengelompokkan topik dan tweet yang disampaikan banyak orang dalam satu waktu. Hasil clustering tersebut digunakan sebagai prediksi untuk mengetahui kemungkinan topik atau kata yang mungkin menjadi tren dalam percakapan publik di Twitter. Metode K-mean dilakukan dengan memanfaatkan metode dan modul mengenai data mining yang terdapat pada bahasa pemrograman python, sehingga cukup cepat dan akurat.

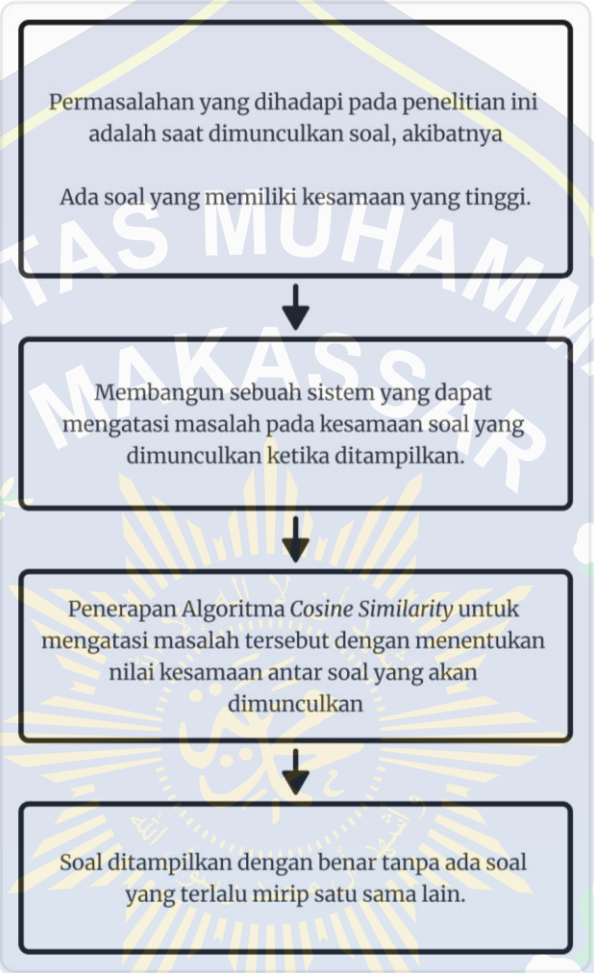
Dari hasil tersebut, peneliti memperhatikan beberapa keunggulan maupun kekurangan dari metode yang digunakan ringkasan penelitian sebelumnya meliputi:

Table 1. Penelitian Terkait

Judul Penelitian	Metode/Algoritma	Keunggulan	Kekurangan
Aplikasi Ujian *Online* Dan Penerapan Algoritma LCG Untuk Proses Pengacakan Soal Ujian Di Smk Negeri Tugumulyo (Rusdiyanto et al., 2022).	*Linear Congruential Generators* (LCG)	Sederhana dan cepat dalam menghasilkan bilangan acak.	Periode bilangan acak yang terbatas dan kurangnya kerandoman yang kuat, terutama ketika digunakan untuk aplikasi yang membutuhkan keamanan tinggi.
Implementasi Algoritma *Multiply With Carry Generator* (MWCG) Dalam Pengacakan Soal Ujian Semester Berbasis Web Pada SMKN 1 Kendari (Saputra et al., 2022).	*Multiply With Carry Generator* (MWCG)	Kemampuannya untuk menghasilkan bilangan acak yang tersebar secara luas dan dalam jangka waktu yang lama.	Tingkat kompleksitas yang lebih tinggi saat digunakan dibandingkan dengan algoritma acak biasa, dan kemungkinan pola berulang jika parameter tidak dipilih dengan benar.

Perancangan Aplikasi Bank Soal Berbasis Website dengan Algoritma *Fisher Yates Shuffle* dan *Cosine Similarity* (Qhorifadillah et al., 2022).	*Fisher Yates Shuffle* dan *Cosine Similarity*	Kemampuannya untuk mengacak data secara efisien dan acak dengan kompleksitas waktu.	kepekaan terhadap implementasi yang salah, yang dapat menyebabkan pengacakan yang tidak akurat.
Implementasi Metode *Multiplicative Random Number Generator* (MRNG) Pada Aplikasi Ujian Sekolah Berbasis Komputer (Prasetyadi & Budi Nugroho, 2020).	*Multiplicative Random Number Generator* (MRNG)	Sederhananya dan kecepatan dalam menghasilkan bilangan acak dengan sedikit sumber daya komputasi.	Periode yang terbatas dan rentan terhadap pola berulang, terutama jika parameter tidak dipilih dengan benar.
Implementation of TF-IDF Algorithm and K-mean Clustering Method to Predict Words or Topics on Twitter (Darwis et al., 2020).	TF-IDF dan *K-Mean Clustering*	Efektif dalam pemrosesan teks serta sederhana, cepat dan fleksibel.	Perlu penyesuaian pada koleksi dokumen besar serta harus menentukan jumlah *cluster* sebelumnya.

C. Kerangka Pikir



Gambar 1. Kerangka Pikir

BAB III
METODE PENELITIAN

A. Tempat dan Waktu Penelitian

1. Tempat Penelitian

Penelitian ini dilakukan di Markas PMI Kota Makassar

2. Waktu Penelitian

Adapun pelaksanaan penelitian ini dilakukan selama bulan Agustus – Oktober 2024.

B. Alat dan Bahan

Adapun alat dan bahan yang akan digunakan dalam penelitian ini adalah sebagai berikut:

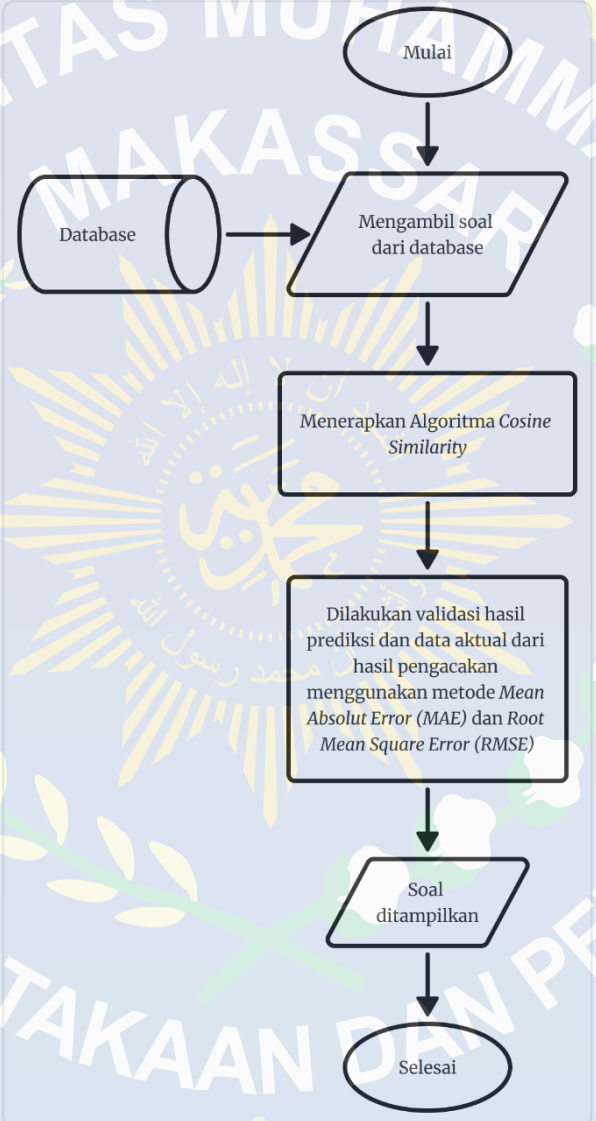
1. Kebutuhan *Hardware* (Perangkat Keras)
   a. Laptop, untuk spesifikasi yang digunakan dalam penelitian ini yaitu menggunakan merk Acer Swift 3 SF314-41 dengan *processor AMD Ryzen 5 3500U with Radeon Vega Mobile Gfx 2.10 GHz,* RAM 8 GB dan SSD Internal 128 GB.
2. Kebutuhan *Sofware* (Perangkat Lunak)
   a. *Jupyter Notebook* (sebagai *software* utama)
   b. *MySQL* (sebagai *database*)
3. Kebutuhan *Dataset*

   Dataset yang digunakan berupa soal lomba PMR yang diambil dari Markas PMI Kota Makassar.

C. Perancangan Sistem

1. *Flowchart* Sistem

Untuk memudahkan proses perancangan, peneliti menggunakan *flowchart* untuk membuat perancangan sistem, seperti yang ditunjukkan pada gambar berikut:



Gambar 2. Flowchart Sistem

Pada saat setelah sistem dijalankan, sistem mengambil soal yang tersedia pada *database* yang selanjutnya diterapkan algoritma *cosine similarity* untuk membandingkan kesamaan antar soal. Kemudian dilakukan validasi menggunakan metode MAE dan RMSE yang melihat selisih antara data

prediksi dan data aktual dari hasil pengacakan, dan setelah itu soal kemudian ditampilkan.

2. *Flowchart Cosine Similarity* dan metode TF-IDF dalam perhitungan nilai kesamaan antar soal

Untuk lebih mudah memahami bagaimana proses perhitungan nilai *cosine similarity* dengan metode TF-IDF, peneliti menggunakan *flowchart* seperti yang ditunjukkan pada gambar berikut:

Gambar 3. Flowchart Representasi dan Perhitungan Nilai Soal

Pada tahapan proses algoritma *cosine similarity* setelah soal diambil dari *database,* digunakan metode TF-IDF untuk merepresentasi soal menjadi vektorisasi numerik agar algoritma *cosine similarity* dapat lebih mudah melakukan perhitungan nilai kesamaan. Kemudian algoritma *cosine similarity* melakukan perhitungan kesamaan antar soal dan antar kategori dedngan kisaran 0 hingga 1, semakin nilai soal mendekati 0 maka soal tersebut semakin berbeda, dan jika nilai soal mendekati 1 maka soal tersebut semakin mirip satu sama lain. Kemudian akan dilakukan pengurutan soal dimulai dari nilai kesamaan terendah yang lalu soal-soal dengan nilai kesamaan identik akan dikelompokkan, begitu juga dengan soal-soal dengan nilai kesamaan yang tidak identik, setelah itu dilakukan pengacakan soal dilihat dari nilai cosine similarity yang telah ditentukan ambang batas kemiripannya, guna memastikan tidak ada soal yang terlalu mirip atau sama dalam tiap pengacakan. langkah terakhir adalah mengirimkan data soal yang sudah diacak ke sistem ujian online, sehingga peserta dapat melihat soal.

D. Teknik Pengujian Sistem

Metode yang dikenal sebagai Algoritma *Cosine Similarity* dapat digunakan untuk menentukan seberapa mirip dua bagian dalam ruang. Dalam penelitian ini, diadaptasi untuk dapat digunakan dalam pengacakan soal ujian *online* untuk memastikan bahwa soal-soal yang diberikan kepada peserta memiliki tingkat pengacakan yang merata, yang memungkinkan soal-soal dalam ujian menjadi lebih adil dan merata.

Namun untuk melakukan pengacakan menggunakan algoritma ini, diperlukan sebuah metode bernama TF-IDF (Term Frequency-Inverse Document Frequency) yang memiliki fungsi untuk merepresentasikan soal dan kategori ke dalam bentuk vektor numerik agar algoritma *cosine similarity* dapat melakukan perhitungan nilai kesamaan.

Cosine similarity melakukan perhitungan nilai antar soal dan antar kategori untuk mengetahui seberapa mirip satu soal dengan yang lain. Ini mencakup

perhitungan produk dot dari dua vektor dibagi dengan magnitudo, atau hasil kali panjang dari kedua vektor tersebut.

Semua soal diurutkan berdasarkan nilai *cosine similarity* diurutkan dari yang terendah hingga yang tertingi dalam masing-masing kategori. Setelah soal-soal diurutkan berdasarkan nilainya, soal-soal kemudian diacak berdasarkan nilai *cosine similarity* terendah dari tiap kategori untuk memastikan tidak ada soal yang terlalu mirip atau sama dalam masing-masing pengacakan.

Kemudian digunakan metode MAE dan RMSE, untuk memvalidasi data prediksi dan data aktual soal untuk memastikan bahwa soal-soal yang dipilih dapat memenuhi tujuan penelitian yang diharapkan.

E. Teknik Analisis Data

Analisis data adalah proses pencarian dan pengaturan hasil dan bahan yang dikumpulkan secara sistematis dalam upaya meningkatkan pemahaman dan presentasi temuan. Berikut adalah langkah-langkah yang digunakan dalam penelitian untuk menganalisis data:

1. Menyiapkan *Dataset*

Langkah pertama ialah menyiapkan *dataset*, data yang dipakai berupa soal-soal lomba dari Markas PMI Kota Makssar, yang dibagi berdasarkan kategori-kategori tertentu. Dalam hal ini sebanyak 7 kategori soal dengan jumlah ±30 hingga 50 soal dari tiap kategori. Data-data soal yang digunakan kemudian disimpan ke dalam *database MySQL*.

2. *Preprocessing Data*

Selanjutnya adalah tahap pemrosesan awal data. Tujuannya adalah untuk membuat data soal ujian siap untuk digunakan. Proses-proses ini termasuk

a. mengubah teks menjadi huruf kecil,
b. mengubah tanda "_" menjadi spasi pada kolom kategori,
c. menghapus tanda baca, angka maupun karakter khusus,

d. menghapus spasi berlebih di awal dan akhir soal.

3. Penggunaan metode TF-IDF

Pada bagian ini soal-soal serta kategori yang telah diproses kemudian direpresentasikan kedalam bentuk vektorisasi numerik menggunakan metode TF-IDF, agar kemudian bisa digunakan dalam perhitungan nilai kesamaan *cosine similarity*.

4. Penerapan *Cosine Similarity*

Pada titik ini, algoritma *cosine similarity* digunakan untuk menghitung nilai kesamaan antar soal dan antar kategori melalui hasil vektorisasi numerik dari metode TF-IDF sebelumnya. setelah selesai dihitung nilai kesamaannya kemudian diurutkan berdasarkan nilai *cosine similarity* terendah hingga tertinggi.

Setelah diurutkan, soal kemudian dikelompokan sesuai nilai *cosine similarity* nya saat dibandingkan satu sama lain, kemudian diacak menggunakan fungsi *random shuffle* dengan melihat berdasarkan nilai *cosine similarity* yang telah diberi batas nilai kesamaannya untuk memastikan tidak ada soal yang terlalu mirip atau sama lain, dalam tiap pengacakan diberikan ambang batas nilai kesamaan dibawah 0.9. Dalam pengacakan ini akan dilakukan tes dalam *jupyter notebook* untuk kemudian dipakai sebagai baha pengujian efektifitas algoritma ini.

5. Penerapan Metode Validasi *Mean Absolut Error (MAE) dan Root Mean Squared Error (RMSE)* untuk Pengecekan Efektifitas

Setelah hasil pengacakan dari algoritma *cosine similarity* selesai, metode MAE dan RMSE digunakan untuk memvalidasi nilai antara data prediksi dan data aktual berdasarkan kategori yang tertampil dari hasil pengacakan. Tujuannya adalah untuk memastikan apakah penerapan algoritma *cosine similarity* sesuai dalam membantu efektifitas sistem pengacakan.

6. Evaluasi Hasil

Untuk membuat kesimpulan tentang efektifitas algoritma *cosine similarity*, evaluasi hasil melibatkan interpretasi data yang telah dianalisis dengan metode MAE dan RMSE dalam melihat hasil distribusi per kategorinya. Dalam proses evaluasi ini, dilakukan pengujian dengan pengacakan sebanyak 10x, 50x, dan 100x, serta terdapat sebanyak 50 dan 100 soal di masing-masing pengacakan.

Pada titik ini, akan dibuatkan evaluasi dari hasil pengujian yang telah dilakukan lalu membuat kesimpulan untuk mengetahui apakah tujuan penelitian tercapai dan apakah algoritma tersebut efektif dalam pengacakan soal ujian *online*, kesimpulan akhir ditarik.

BAB IV
HASIL DAN PEMBAHASAN

A. Pengambilan Data

Pada proses pengambilan data untuk penerapan algoritma *cosine similarity* dalam efektifitas pengacakan soal ujian *online* ini, *dataset* soal diambil dari Markas PMI Kota Makassar.

kategori	teks_soal
sejarah_gerakan	Pada tanggal bulan dan tahun berapakah NERKAI terb...
pertolongan_pertama	Tanda apa saja yang perlu kita temukan saat melaku...
kepemimpinan	Sebutkan jenis–jenis komunikasi ?
donor_darah	Jelaskan apa yang dimaksud dengan donor darah suka...
remaja_sehat_peduli_sesama	Lina berumur 27 tahun, tinggi badan 161 cm dan ber...
sejarah_gerakan	Sebutkan 3 syarat terbentuknya suatu perhimpunan n...
kepemimpinan	Sebutkan 4 hal yang mendukung komunikasi ?
remaja_sehat_peduli_sesama	Adi umur 17 tahun, tinggi badan 152 cm dan berat b...
pendidikan_remaja_sebaya	Sebutkan perubahan yang terjadi selama tumbuh kemb...
ayo_siaga_bencana	Sebutkan 3 penyebab bencana akibat ulah manusia ya...
pertolongan_pertama	Peragakan cara penanganan terkilir pada pergelanga...
sejarah_gerakan	Pada tanggal, bulan dan tahun berapakah panitia li...
pertolongan_pertama	Secara umum cedera otot rangka dapat berupa ?
kepemimpinan	Sebutkan yang termasuk dalam unsur-unsur komunikas...

Gambar 4. Dataset soal di dalam MySQL

Dapat dilihat pada gambar diatas yang merupakan *dataset* soal yang berada di *database MySQL*, terdapat 2 kolom yaitu yang pertama kolom “kategori” yang berisi kategori dari tiap-siap soal yang ada di dalam database dan kolom “teks_soal” yang berisi soal-soal dari tiap kategori yang ada di dalam database.

Table 2. Jumlah Data Soal

Kategori Soal	Total Soal
ayo_siaga_bencana	36
donor_darah	24
kepemimpinan	37
pendidikan_remaja_sebaya	56
pertolongan_pertama	64
remaja_sehat_peduli_sesa	37
sejarah_gerakan	49
Total Keseluruhan	**303**

Tabel diatas merupakan total seluruh soal yang berada di *databse* berdasarkan kategori yang ada, terdapat 7 kategori berbeda untuk tiap soal yang diambil dari 7 materi pokok Kepalangmerahan PMI.

B. Pengujian dan Hasil Model

1. Preprocessing Data

Sebelum memasuki tahapan *preprocessing* data dilakukan, dilakukan koneksi *jupyter notebook* ke *database MySQL* dengan hasil output sebagai berikut:

```
Koneksi ke database berhasil!
```

Gambar 5. Output Koneksi Database

Setelah koneksi berhasil disambungkan selanjutnya yaitu mengimpor modul untuk pengolahan dan menampilkan data, dengan menggunakan modul *create_engine* dari *sqlalchemy* untuk lebih memudahkan pengolahan data langsung dari *database* ke *jupyter notebook* serta membuat koneksi antara *sqlalchemy* dan *database* dan kemudian menampilkannya dengan hasil output sebagai berikut:

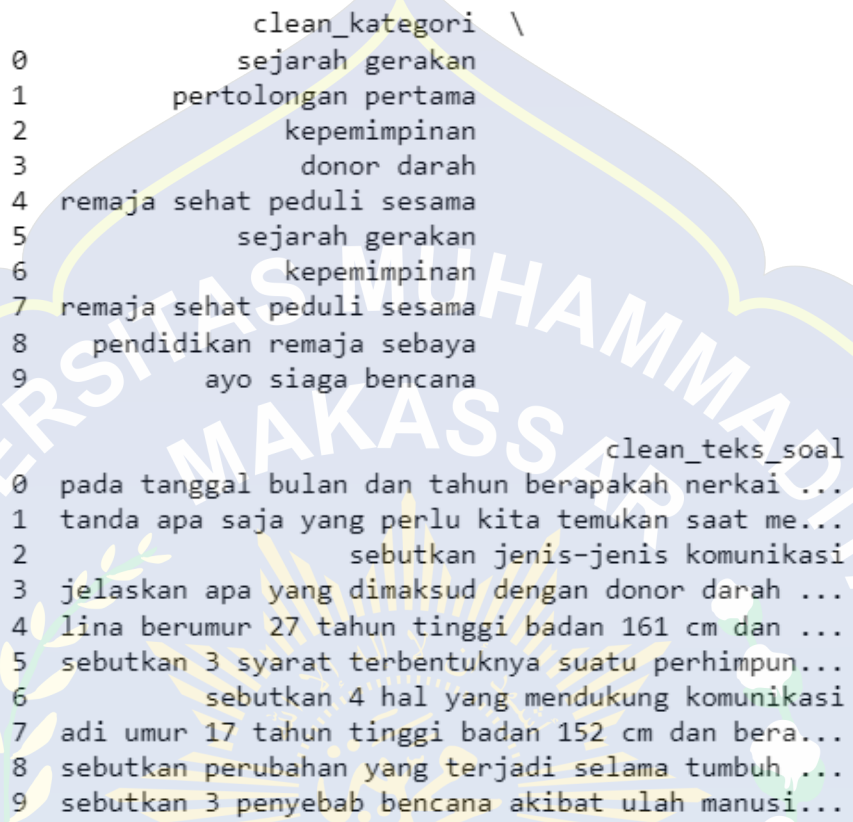
```
Sampel data :
                    kategori  \
0            sejarah_gerakan
1         pertolongan_pertama
2               kepemimpinan
3                donor_darah
4  remaja_sehat_peduli_sesama
5            sejarah_gerakan
6               kepemimpinan
7  remaja_sehat_peduli_sesama
8    pendidikan_remaja_sebaya
9          ayo_siaga_bencana

                                          teks_soal
0  Pada tanggal bulan dan tahun berapakah NERKAI ...
1  Tanda apa saja yang perlu kita temukan saat me...
2                Sebutkan jenis-jenis komunikasi ?
3  Jelaskan apa yang dimaksud dengan donor darah ...
4  Lina berumur 27 tahun, tinggi badan 161 cm dan...
5  Sebutkan 3 syarat terbentuknya suatu perhimpun...
6        Sebutkan 4 hal yang mendukung komunikasi ?
7  Adi umur 17 tahun, tinggi badan 152 cm dan ber...
8  Sebutkan perubahan yang terjadi selama tumbuh ...
9  Sebutkan 3 penyebab bencana akibat ulah manusi...

                   Kategori  Total Soal
0          ayo_siaga_bencana          36
1                donor_darah          24
2               kepemimpinan          37
3    pendidikan_remaja_sebaya          56
4         pertolongan_pertama          64
5  remaja_sehat_peduli_sesama          37
6            sejarah_gerakan          49
```

Gambar 6. Output Dataset Soal dari Database

Setelah kedua tahapan diatas selesai, selanjutnya dilakukan tahapan *preprocessing* data, pada tahapan ini kolom "kategori" dan "teks_soal" dibersihkan, maksud dari pembersihan ini yaitu agar dapat dengan mudah diproses oleh mesin dengan cara mengubahnya ke huruf kecil, mengubah tanda "_" menjadi spasi dari kolom kategori, menghapus tanda baca, serta menghapus spasi berlebih dari tiap-tiap soal. Hasil dari tahapan *preprocessing* data adalah sebagai berikut:



	clean_kategori	clean_teks_soal
0	sejarah gerakan	pada tanggal bulan dan tahun berapakah nerkai ...
1	pertolongan pertama	tanda apa saja yang perlu kita temukan saat me...
2	kepemimpinan	sebutkan jenis-jenis komunikasi
3	donor darah	jelaskan apa yang dimaksud dengan donor darah ...
4	remaja sehat peduli sesama	lina berumur 27 tahun tinggi badan 161 cm dan ...
5	sejarah gerakan	sebutkan 3 syarat terbentuknya suatu perhimpun...
6	kepemimpinan	sebutkan 4 hal yang mendukung komunikasi
7	remaja sehat peduli sesama	adi umur 17 tahun tinggi badan 152 cm dan bera...
8	pendidikan remaja sebaya	sebutkan perubahan yang terjadi selama tumbuh ...
9	ayo siaga bencana	sebutkan 3 penyebab bencana akibat ulah manusi...

Gambar 7. Output Hasil Preprocessing Data

Disini menampilkan sampel sebanyak 10 soal dan kategori dari hasil *preprocessing data* yang kemudian akan digunakan untuk tahapan selanjutnya.

2. Penerapan Metode *Term Frequency-Inverse Document Frequency*

Pada tahap ini data yang telah dibersihkan kemudian diproses dengan mengubahnya menjadi representasi vektor numerik dengan menggunakan metode TF-IDF.

mengimpor *TfidVectorizer* dari modul *sklearn* yang kemudian digunakan untuk mengubah data masing-masing soal dan kategori yang telah melewati *preprocessing* menjadi vektorisasi numerik dan menampilkannya dalam bentuk sparse. Untuk hasil output dari tahapan ini adalah sebagai berikut:

Coords	Values
(0, 461)	0.23231409655457333
(0, 625)	0.41303656663904764
(0, 132)	0.37509929323882457
(0, 147)	0.21847397491674211
(0, 621)	0.2605969545627699
(0, 105)	0.307177121841205
(0, 443)	0.46083187864276287
(0, 634)	0.46083187864276287
(1, 624)	0.3060397035872736
(1, 71)	0.2335410011154771
(1, 569)	0.26911000698135223
(1, 690)	0.13307430365829934
(1, 526)	0.37076576656982
(1, 324)	0.29131420114924633
(1, 631)	0.37076576656982
(1, 567)	0.3370557896340273
(1, 386)	0.2982670640980235
(1, 482)	0.3060397035872736
(1, 209)	0.32502417808503903
(2, 584)	0.19672015341046514
(2, 269)	0.8848503567349412
(2, 332)	0.4223044250634964
(3, 71)	0.331954826839204
(3, 690)	0.1891516145629423
(3, 267)	0.304970323413945
:	:

Gambar 8. Output Hasil TF-IDF Soal

Coords	Values
(0, 13)	0.7071067811865475
(0, 4)	0.7071067811865475
(1, 9)	0.7071067811865475
(1, 8)	0.7071067811865475
(2, 5)	1.0
(3, 3)	0.7071067811865475
(3, 2)	0.7071067811865475
(4, 10)	0.3774046894165113
(4, 12)	0.5346543121202791
(4, 6)	0.5346543121202791
(4, 14)	0.5346543121202791
(5, 13)	0.7071067811865475
(5, 4)	0.7071067811865475
(6, 5)	1.0
(7, 10)	0.3774046894165113
(7, 12)	0.5346543121202791
(7, 6)	0.5346543121202791
(7, 14)	0.5346543121202791
(8, 10)	0.4983553397779479
(8, 7)	0.6130423946656569
(8, 11)	0.6130423946656569
(9, 0)	0.5773502691896257
(9, 15)	0.5773502691896257
(9, 1)	0.5773502691896257
(10, 9)	0.7071067811865475
:	:

Gambar 9. Output Hasil TF-IDF Kategori

Disini menampilkan sebagian sampel hasil vektorisasi soal dan kategori dalam sparse matrix yang sebagian besar elemen datanya bernilai nol, ini sering terjadi pada matriks TF-IDF karena sebagian besar kata tidak muncul di dokumen tertentu.

Penjelasan:

(0, 634) 0,4608…

a. 0 = titik koordinat x atau id soal pada vektor matrix
b. 634 = titik koordinat y atau id kata dalam soal vektor matrix
c. 0,4608.. = nilai pembobotan kata dari hasil TF-IDF

3. Penerapan Algoritma *Cosine Similarity*

Pada tahap ini dilakukan pengujian penerapan algoritma *cosine similarity* dalam sistem pengacakan soal. Pada bagian ini ada beberapa tahapan yang dilakukan yaitu:

a. Menghitung nilai *cosine similarity* antar soal dan antar kategori

Mengimpor algoritma *cosine similarity* dari modul *sklearn* yang kemudian digunakan untuk menghitung nilai kesamaan antar soal dan antar kategori yang telah melewati proses TF-IDF. Setelah perhitungan nilai kesamaan selesai, hasilnya kemdian disimpan dalam *cosine_sim_matrix_teks* dan *cosine_sim_matrix_kategori* yang kemudian ditampilkan dalam matrix dengan sampel tampilan 5x5. Kemudian hasilnya disimpan dalam *dataframe cosine similarity* untuk diproses setelahnya. Untuk hasil outputnya adalah sebagai berikut:

```
Cosine Similarity Matrix untuk Teks Soal (sampel 5x5):
          0         1    2         3         4       5         6         7    \
0  1.000000  0.000000  0.0  0.000000  0.118403  0.0000  0.000000  0.117240
1  0.000000  1.000000  0.0  0.102696  0.000000  0.0000  0.033066  0.000000
2  0.000000  0.000000  1.0  0.000000  0.000000  0.0302  0.242334  0.000000
3  0.000000  0.102696  0.0  1.000000  0.000000  0.0000  0.047000  0.000000
4  0.118403  0.000000  0.0  0.000000  1.000000  0.0000  0.000000  0.349598

          8         9    ...       293       294       295       296       297  \
0  0.000000  0.000000  ...  0.046745  0.044146  0.010956  0.000000  0.040195
1  0.019013  0.019700  ...  0.130052  0.000000  0.027618  0.000000  0.057594
2  0.024845  0.025743  ...  0.032132  0.030346  0.007531  0.026018  0.000000
3  0.027025  0.094437  ...  0.000000  0.000000  0.000000  0.000000  0.081863
4  0.000000  0.000000  ...  0.026356  0.024891  0.006177  0.000000  0.000000

          298       299       300       301       302
0  0.000000  0.065632  0.024584  0.021506  0.175190
1  0.059635  0.043889  0.092715  0.011313  0.007729
2  0.116602  0.000000  0.016899  0.149549  0.000000
3  0.084765  0.110440  0.116045  0.252838  0.063113
4  0.000000  0.366224  0.013861  0.012126  0.064991
```

Gambar 10. Output Dataframe Matrix Cosine Similarity Soal

```
Cosine Similarity Matrix untuk Kategori (sampel 5x5):
     0    1    2    3    4    5    6    7         8    9    ...  293       294  \
0  1.0  0.0  0.0  0.0  0.0  1.0  0.0  0.0  0.000000  0.0  ...  0.0  0.000000
1  0.0  1.0  0.0  0.0  0.0  0.0  0.0  0.0  0.000000  0.0  ...  1.0  0.000000
2  0.0  0.0  1.0  0.0  0.0  0.0  1.0  0.0  0.000000  0.0  ...  0.0  0.000000
3  0.0  0.0  0.0  1.0  0.0  0.0  0.0  0.0  0.000000  0.0  ...  0.0  0.000000
4  0.0  0.0  0.0  0.0  1.0  0.0  0.0  1.0  0.188082  0.0  ...  0.0  0.188082

     295  296  297  298  299  300  301  302
0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  1.0
1  1.0  0.0  1.0  0.0  0.0  1.0  0.0  0.0
2  0.0  0.0  0.0  1.0  0.0  0.0  0.0  0.0
3  0.0  0.0  0.0  0.0  0.0  0.0  1.0  0.0
4  0.0  0.0  0.0  0.0  1.0  0.0  0.0  0.0
```

Gambar 11. Output Dataframe Matrix Cosine Similarity Kategori

Penjelasan:

a. Soal ke-1 pada matrix *dataframe* (0,0) jika dibandingkan dengan soal ke-1 atau dirinya sendiri, nilai kesamaannya adalah 1,000, namun jika dibandingkan dengan soal ke-5 (0,4) memiliki kemiripan dengan nilai 0,118403.

b. Soal ke 2 pada matrix *dataframe* (1,1) jika dibandingkan dengan soal ke-2 atau dirinya sendiri, nilai kesamaannya adalah 1,000, namun jika dibandingkan dengan soal ke-4 (0,3) memiliki kemiripan dengan nilai 0,102696.

c. Begitu pula seterusnya hingga soal ke-303 pada matrix *dataframe* (302,302).

b. Menghitung rata-rata nilai *cosine similarity* antar soal dan antar kategori dan mengurutkannya

Proses ini mencakup mengambil kolom kategori dan teks_soal dari *database* dan menyimpannya untuk pemrosesan, menghitung rata-rata nilai *cosine similarity* antar soal dan antar kategori, menyimpannya ke dalam kolom baru untuk diproses setelahnya, mengurutkan soal berdasarkan nilai *cosine similarity* terendah hingga tertinggi. Hasil dari tahapan ini adalah sebagai berikut:

```
Soal 274:
  Kategori: pendidikan_remaja_sebaya
  Teks Soal: Berapa persen kisaran risiko penularan virus HIV Ibu Hamil
  Nilai rata-rata cosine similarity (Teks Soal): 0.0109
Soal 90:
  Kategori: pertolongan_pertama
  Teks Soal: Peragakan teknik menilai pernapasan ?
  Nilai rata-rata cosine similarity (Teks Soal): 0.0116
Soal 260:
  Kategori: remaja_sehat_peduli_sesama
  Teks Soal: Bagaimanakah cara Pencegahan Gizi Buruk ?
  Nilai rata-rata cosine similarity (Teks Soal): 0.0123
Soal 247:
  Kategori: remaja_sehat_peduli_sesama
  Teks Soal: Dimensi fisik (tubuh), Dimensi mental (otak), Dimensi emos
mensi manusia. Benar atau Salah ?
  Nilai rata-rata cosine similarity (Teks Soal): 0.0146
```

Gambar 12. Output Nilai Cosine Similarity Terendah

```
Soal 117:
  Kategori: donor_darah
  Teks Soal: Jelaskan apa yang dimaksud dengan donor darah sukarela ?
  Nilai rata-rata cosine similarity (Teks Soal): 0.0607
Soal 4:
  Kategori: donor_darah
  Teks Soal: Jelaskan apa yang dimaksud dengan donor darah sukarela ?
  Nilai rata-rata cosine similarity (Teks Soal): 0.0607
Soal 94:
  Kategori: donor_darah
  Teks Soal: Jelaskan yang dimaksud dengan transfusi darah ?
  Nilai rata-rata cosine similarity (Teks Soal): 0.0624
Soal 195:
  Kategori: kepemimpinan
  Teks Soal: Apa yang dimaksud dengan komunikasi?
  Nilai rata-rata cosine similarity (Teks Soal): 0.0656
```

Gambar 13. Output Nilai Cosine Similarity Tertinggi

Hasilnya, soal ke-274 adalah soal dengan rata-rata nilai *cosine similarity* terendah jika dibandingkan antar soal dengan nilai 0,0109, sedangkan soal ke-195 adalah soal dengan nilai rata-rata *cosine similarity* tertinggi jika dibandingkan antar soal dengan nilai 0,0656.

c. Mengelompokkan soal dengan nilai *cosine similarity* yang sama persis

Proses ini mencakup mengidentifikasi soal-soal dengan nilai *cosine similarity* yang identik maupun tidak identik, mengelompokkan, menyimpan dan juga menampilkanya.

Pertama-tama yaitu dengan menetapkan nilai ambang batas cosine similarity sebesar 1.0 untuk menentukan apakah ada soal yang identik satu sama lain. Serta membuatkan grup atau kelompok bagi soal-soal yang terdeteksi mempunyai nilai cosine similarity sama persis dan mengecek soal yang sudah diperiksa atau dikelompokkan. jika ada yang memiliki nilai kesamaan 1.0 atau identik maka akan dimasukkan ke kelompok serta menandai soal yang sudah diperiksa maupun dikelompokkan, sementara soal yang tidak memiliki nilai kesamaan 1.0 akan dibuatkan grup tersendiri. Untuk hasil outputnya adalah sebagai berikut:

```
Group 1:
  - Soal 1: Pada tanggal bulan dan tahun berapakah NERKAI terbentuk ?
  - Soal 114: Pada tanggal bulan dan tahun berapakah NERKAI terbentuk ?

Group 2:
  - Soal 2: Tanda apa saja yang perlu kita temukan saat melakukan pemeriksaan fisik ?
  - Soal 115: Tanda apa saja yang perlu kita temukan saat melakukan pemeriksaan fisik ?

Group 3:
  - Soal 3: Sebutkan jenis-jenis komunikasi ?
  - Soal 116: Sebutkan jenis -jenis komunikasi ?

Group 4:
  - Soal 4: Jelaskan apa yang dimaksud dengan donor darah sukarela ?
  - Soal 117: Jelaskan apa yang dimaksud dengan donor darah sukarela ?
```

Gambar 14. Output Soal-soal Identik

```
Group 64:
  - Soal 65: Sebutkan 5 jenis rongga yang terdapat dalam tubuh manusia ?

Group 65:
  - Soal 67: Sebutkan 3 fungsi dari darah ?

Group 66:
  - Soal 69: Sebutkan 4 komponen dimensi manusia ?

Group 67:
  - Soal 70: Jelaskan apa yang dimaksud dengan peran gender ?

Group 68:
  - Soal 71: Indonesia terletak di antara 3 lempeng utama dunia (the ring of fire), sebutkan!
```

Gambar 15. Output Soal-soal Tidak Identik

Hasilnya adalah terdapat banyak soal dengan nilai *cosine similarity* 1.0 atau sama persis, dan juga banyak soal yang tidak memiliki kesamaan atau sama persis satu sama lain.

d. Melakukan pengacakan soal berdasarkan nilai *cosine similarity*

Untuk menampilkan hasil pendistribusian soal secara merata per kategori, yang pertama dilakukan yaitu dengan menentukan jumlah soal yang diinginkan, jumlah pengacakan, ambang batas nilai cosine similarity agar tidak mengambil soal yang sama persis, mengelompokkan soal berbasarkan nilai cosine similarity kedalam group_id dan tempat untuk menyimpan hasil dari tiap pengacakan.

Kemudian melakukan pengacakan dengan menambahkan opsi untuk memvariasikan soal setiap pengacakan, mengambil 1 per 1 soal secara

acak dari group_id serta mereset sampel data untuk setiap iterasi atau pengacakan.

Selanjutnya yaitu menghitung seberapa banyak total kategori berdasarkan nilai cosine similarity, menentukan kuota soal per kategori dan memberi ekstra soal ke kategori dengan jumlah terendah.

Selanjutnya membuat kandidat soal berdasarkan kategori similarity nya, melakukan perulangan untuk memilih soal per kategori berdasarkan kemiripan nilai similarity nya, memeriksa apakah soal yang diambil terlalu mirip serta memasukkannya jika tidak mirip dan menghapus soal dari kandidat.

Selanjutnya menambahkan soal jika masih kurang dengan melihat distribusi per kategori, menghitung kategori yang telah terisi serta menemukan kategori mana saja yang masih kurang dan menyimpan hasil pengacakan.

Terakhir yaitu meminta untuk menginput salah satu angka berdasarkan total pengacakan yang dilakukan dengan tujuan untuk menampilkan salah satu pengacakan yang dipilih, misal memasukkan angka 1 dan 10 maka akan menampilkan pengacakan pertama dan ke sepuluh. Untuk hasil output tahapan ini adalah sebagai berikut:

```
Masukkan nomor iterasi yang ingin ditampilkan (1-50): 1

Hasil Pengacakan Iterasi Ke-1:
                    kategori  \
259  remaja_sehat_peduli_sesama
160                donor_darah
194              kepemimpinan
72             sejarah_gerakan
178            sejarah_gerakan
8      pendidikan_remaja_sebaya
134         pertolongan_pertama
151            sejarah_gerakan
100         pertolongan_pertama
81             sejarah_gerakan
```

Gambar 16. Output Hasil Pengacakan Cosine Similarity Ke-1 (Kategori)

```
                                                teks_soal  \
259          Bagaimanakah cara Pencegahan Gizi Buruk ?
160           Jelaskan yang dimaksud transfusi darah ?
194              Apa yang dimaksud dengan komunikasi?
72   Pada tahun berapakah konvensi Jenewa yang pert...
178  Pada masa penjajahan Belanda tanggal 21 oktobe...
8    Sebutkan perubahan yang terjadi selama tumbuh ...
134  Peragakan teknik penggunaan pen light dalam pe...
151  Sebutkan 2 gagasan yang di tulis Jean Henry Du...
100  Sebutkan pemeriksaan yang dilakukan pada tahap...
```

Gambar 17. Output Hasil Pengacakan Cosine Similarity Ke-1 (Soal dan Rata-rata Nilai)

```
Masukkan nomor iterasi yang ingin ditampilkan (1-50): 10

Hasil Pengacakan Iterasi Ke-10:
                      kategori  \
202              donor_darah
100        pertolongan_pertama
204               kepemimpinan
293        pertolongan_pertama
170               kepemimpinan
235             sejarah_gerakan
258   pendidikan_remaja_sebaya
70           ayo_siaga_bencana
21         pertolongan_pertama
300        pertolongan_pertama
```

Gambar 18. Output Hasil Pengacakan Cosine Similarity Ke-10 (Kategori)

```
                                                teks_soal  \
202                Genotip golongan darah O adalah ?
100  Sebutkan pemeriksaan yang dilakukan pada tahap...
204              Apa yang dimaksud dengan kelompok ?
293       Sebutkan 4 Gejala dan Tanda Kejang Panas !
170  Komunikasi yang dilakukan melalui gerak bahasa...
235  Pada masa penjajahan Belanda tanggal 21 oktobe...
258  Penyakit ini disebabkan virus yang menimbulkan...
70   Indonesia terletak di antara 3 lempeng utama d...
21   Peragakan teknik penggunaan pen light dalam pe...
300  Pingsan dapat terjadi karena peredaran darah d...
```

Gambar 19. Output Hasil Pengacakan Cosine Similarity Ke-10 (Soal dan Rata-rata Nilai)

Hasilnya disini dinampilkan 10 soal pertama dari pengacakan 1 dan 10 dengan memperlihatkan kategori, soal, dan rata-rata nilai *cosine similarity* antar soal.

4. Penerapan Metode Validasi *Mean Absolute Error* (MAE) dan *Root Mean Squared Error* (RMSE) untuk Pengecekan Efektifitas

Dalam tahapan ini, penerapan metode validasi MAE dan RMSE dilakukan untuk memastikan bahwa penggunaan algoritma dan metode yang dirancang mampu mencapai tujuan dari penelitian yang telah dijelaskan.

Untuk proses dari tahapan ini pertam yaitu dengan Mengimpor metode validasi *mean_absolute_error* serta *mean_squared_error* dari modul *sklearn* dan modul *numpy* yang digunakan untuk operasi matematika dan manipulasi data numerik.

Selanjutnya Menentukan total soal per pengacakan dan menentukan jumlah kategori berdasarkan jenis kategori database untuk validasi. Kemudian menghitung target distribusi ideal berdasarkan jumlah soal yang ditentukan sebelumnya per kategori lalu menyimpannya dan menghitung soal aktual per kategori.

Selanjutnya yaitu Menghitung MAE dan RMSE dari tiap pengacakan serta menghitung nilai rata-rata MAE dan RMSE dari setiap pengacakan dan menampilkannya. Untuk hasil output dari tahapan ini adalah sebagai berikut:

```
Iterasi 1:
  Distribusi Aktual  : [10, 7, 5, 3, 6, 9, 10]
  Distribusi Ideal   : [8, 7, 7, 7, 7, 7, 7]
  MAE                : 2.0000
  RMSE               : 2.3299

Iterasi 2:
  Distribusi Aktual  : [7, 14, 3, 5, 4, 12, 5]
  Distribusi Ideal   : [8, 7, 7, 7, 7, 7, 7]
  MAE                : 3.4286
  RMSE               : 3.9279

Iterasi 3:
  Distribusi Aktual  : [9, 13, 3, 4, 8, 11, 2]
  Distribusi Ideal   : [8, 7, 7, 7, 7, 7, 7]
  MAE                : 3.4286
  RMSE               : 3.8545

Iterasi 4:
  Distribusi Aktual  : [6, 11, 4, 6, 5, 14, 4]
  Distribusi Ideal   : [8, 7, 7, 7, 7, 7, 7]
  MAE                : 3.1429
  RMSE               : 3.6253

Iterasi 5:
  Distribusi Aktual  : [8, 12, 4, 3, 8, 12, 3]
  Distribusi Ideal   : [8, 7, 7, 7, 7, 7, 7]
  MAE                : 3.1429
  RMSE               : 3.6253
```

Gambar 20. Output Distribusi Data Aktual dan Ideal serta MAE dan RMSE per Soal

Hasilnya, disini dapat dilihat nilai aktual serta nilai prediksi berdasarkan kategori yang diharapkan dan rata-rata nilai MAE dan RMSE per pengacakan.

5. Evaluasi Hasil

Hasilnya, disini dapat dilihat nilai aktual serta nilai prediksi berdasarkan kategori yang diharapkan dan rata-rata nilai MAE dan RMSE per pengacakan.

Setelah semua tahapan diatas telah dilakukan, terakhir dilakukan evaluasi untuk menentukan apakah penerapan algoritma *cosine similarity* dalam membatu sistem pengacakan efektif atau tidak, akan diambil hasil evaluasi dari hasil sebanyak 10x, 50x dan 100x pengacakan dengan tiap soal per pengacakan berjumlah 50 dan 100 soal.

a. Untuk 10x pengacakan dengan 50 soal

```
Iterasi 8:
  Distribusi Aktual  : [7, 7, 7, 7, 8, 7, 7]
  Distribusi Ideal   : [8, 7, 7, 7, 7, 7, 7]
  MAE               : 0.2857
  RMSE              : 0.5345

Iterasi 9:
  Distribusi Aktual  : [7, 7, 8, 7, 7, 7, 7]
  Distribusi Ideal   : [8, 7, 7, 7, 7, 7, 7]
  MAE               : 0.2857
  RMSE              : 0.5345

Iterasi 10:
  Distribusi Aktual  : [7, 7, 7, 8, 7, 7, 7]
  Distribusi Ideal   : [8, 7, 7, 7, 7, 7, 7]
  MAE               : 0.2857
  RMSE              : 0.5345

Rata-rata MAE untuk 10 iterasi: 0.2571
Rata-rata RMSE untuk 10 iterasi: 0.4811
```

Gambar 21. Data Aktual dan Ideal serta MAE dan RMSE untuk 10x pengacakan dan 50 soal

Untuk hasil dari 10x pengacakan dengan 50 soal, nilai rata-rata MAE sebesar 0,2571 yang berarti efektif dikarenakan selisihnya antara 0-1, sedangkan untuk nilai rata-rata RMSE sebesar 0,4811 yang efektif dikarenakan selisihnya antara 0-1.

b. Untuk 10x pengacakan dengan 100 soal

```
Iterasi 8:
  Distribusi Aktual  : [14, 14, 15, 14, 15, 14, 14]
  Distribusi Ideal   : [15, 15, 14, 14, 14, 14, 14]
  MAE               : 0.5714
  RMSE              : 0.7559

Iterasi 9:
  Distribusi Aktual  : [14, 14, 15, 15, 14, 14, 14]
  Distribusi Ideal   : [15, 15, 14, 14, 14, 14, 14]
  MAE               : 0.5714
  RMSE              : 0.7559

Iterasi 10:
  Distribusi Aktual  : [14, 15, 14, 15, 14, 14, 14]
  Distribusi Ideal   : [15, 15, 14, 14, 14, 14, 14]
  MAE               : 0.2857
  RMSE              : 0.5345

Rata-rata MAE untuk 10 iterasi: 0.5143
Rata-rata RMSE untuk 10 iterasi: 0.7116
```

Gambar 22. Data Aktual dan Ideal serta MAE dan RMSE untuk 10x pengacakan dan 100 soal

Untuk hasil dari 10x pengacakan dengan 100 soal, nilai rata-rata MAE sebesar 0,5143 yang berarti efektif dikarenakan selisihnya antara 0-1, sedangkan untuk nilai rata-rata RMSE sebesar 0,7116 yang efektif dikarenakan selisihnya antara 0-1.

c. Untuk 50x pengacakan dengan 50 soal

```
Iterasi 48:
  Distribusi Aktual  : [7, 7, 7, 8, 7, 7, 7]
  Distribusi Ideal   : [8, 7, 7, 7, 7, 7, 7]
  MAE                : 0.2857
  RMSE               : 0.5345

Iterasi 49:
  Distribusi Aktual  : [7, 7, 8, 7, 7, 7, 7]
  Distribusi Ideal   : [8, 7, 7, 7, 7, 7, 7]
  MAE                : 0.2857
  RMSE               : 0.5345

Iterasi 50:
  Distribusi Aktual  : [7, 7, 7, 8, 7, 7, 7]
  Distribusi Ideal   : [8, 7, 7, 7, 7, 7, 7]
  MAE                : 0.2857
  RMSE               : 0.5345

Rata-rata MAE untuk 50 iterasi: 0.2514
Rata-rata RMSE untuk 50 iterasi: 0.4704
```

Gambar 23. Data Aktual dan Ideal serta MAE dan RMSE untuk 50x pengacakan dan 50 soal

Untuk hasil dari 50x pengacakan dengan 50 soal, nilai rata-rata MAE sebesar 0,2514 yang berarti efektif dikarenakan selisihnya antara 0-1, sedangkan untuk nilai rata-rata RMSE sebesar 0,4704 yang efektif dikarenakan selisihnya antara 0-1.

d. Untuk 50x pengacakan dengan 100 soal

```
Iterasi 48:
  Distribusi Aktual  : [14, 15, 14, 15, 14, 14, 14]
  Distribusi Ideal   : [15, 15, 14, 14, 14, 14, 14]
  MAE                : 0.2857
  RMSE               : 0.5345

Iterasi 49:
  Distribusi Aktual  : [14, 14, 15, 14, 14, 14, 15]
  Distribusi Ideal   : [15, 15, 14, 14, 14, 14, 14]
  MAE                : 0.5714
  RMSE               : 0.7559

Iterasi 50:
  Distribusi Aktual  : [14, 14, 14, 15, 15, 14, 14]
  Distribusi Ideal   : [15, 15, 14, 14, 14, 14, 14]
  MAE                : 0.5714
  RMSE               : 0.7559

Rata-rata MAE untuk 50 iterasi: 0.4000
Rata-rata RMSE untuk 50 iterasi: 0.6106
```

Gambar 24. Data Aktual dan Ideal serta MAE dan RMSE untuk 50x pengacakan dan 100 soal

Untuk hasil dari 50x pengacakan dengan 100 soal, nilai rata-rata MAE sebesar 0,4000 yang berarti efektif dikarenakan selisihnya antara 0-1, sedangkan untuk nilai rata-rata RMSE sebesar 0,6106 yang efektif dikarenakan selisihnya antara 0-1.

e. Untuk 100x pengacakan dengan 50 soal

```
Iterasi 98:
  Distribusi Aktual  : [7, 8, 7, 7, 7, 7, 7]
  Distribusi Ideal   : [8, 7, 7, 7, 7, 7, 7]
  MAE               : 0.2857
  RMSE              : 0.5345

Iterasi 99:
  Distribusi Aktual  : [7, 7, 7, 7, 7, 7, 8]
  Distribusi Ideal   : [8, 7, 7, 7, 7, 7, 7]
  MAE               : 0.2857
  RMSE              : 0.5345

Iterasi 100:
  Distribusi Aktual  : [7, 7, 7, 7, 7, 8, 7]
  Distribusi Ideal   : [8, 7, 7, 7, 7, 7, 7]
  MAE               : 0.2857
  RMSE              : 0.5345

Rata-rata MAE untuk 100 iterasi: 0.2486
Rata-rata RMSE untuk 100 iterasi: 0.4650
```

Gambar 25. Data Aktual dan Ideal serta MAE dan RMSE untuk 100x pengacakan dan 50 soal

Untuk hasil dari 100x pengacakan dengan 50 soal, nilai rata-rata MAE sebesar 0,2486 yang berarti efektif dikarenakan selisihnya antara 0-1, sedangkan untuk nilai rata-rata RMSE sebesar 0,4650 yang efektif dikarenakan selisihnya antara 0-1.

f. Untuk 100x pengacakan dengan 100 soal

```
Iterasi 98:
  Distribusi Aktual  : [14, 15, 14, 14, 15, 14, 14]
  Distribusi Ideal   : [15, 15, 14, 14, 14, 14, 14]
  MAE               : 0.2857
  RMSE              : 0.5345

Iterasi 99:
  Distribusi Aktual  : [14, 14, 14, 14, 14, 15, 15]
  Distribusi Ideal   : [15, 15, 14, 14, 14, 14, 14]
  MAE               : 0.5714
  RMSE              : 0.7559

Iterasi 100:
  Distribusi Aktual  : [14, 14, 15, 14, 14, 15, 14]
  Distribusi Ideal   : [15, 15, 14, 14, 14, 14, 14]
  MAE               : 0.5714
  RMSE              : 0.7559

Rata-rata MAE untuk 100 iterasi: 0.3914
Rata-rata RMSE untuk 100 iterasi: 0.6039
```

Gambar 26. Data Aktual dan Ideal serta MAE dan RMSE untuk 100x pengacakan dan 100 soal

Untuk hasil dari 100x pengacakan dengan 100 soal, nilai rata-rata MAE sebesar 0,3914 yang berarti efektif dikarenakan selisihnya antara 0-1, sedangkan untuk nilai rata-rata RMSE sebesar 0,6039 yang efektif dikarenakan selisihnya antara 0-1.

Dari hasil pengujian diatas dapat dievaluasi bahwa, penerapan algoritma *cosine similarity* dalam sistem pengacakan efektif jika dilihat rata-rata hasil distribusi per kategori dari evaluasi MAE dan RMSE nya, hal ini dapat divalidasi dikarenakan nilai selisih yang berada diantara angka 0-1.

BAB V
PENUTUP

A. Kesimpulan

Berdasarkan rumusan masalah, tujuan serta ruang lingkup penelitian yang telah ditetapkan yaitu, untuk mengetahui bagaimana cara algoritma cosine similarity bisa diterapkan dalam pengacakan soal ujian online serta untuk mengetahui apakah penerapan algoritma *cosine similarity* efektif digunakan dalam sistem pengacakan soal ujian online, berikut adalah kesimpulan yang dapat diambil.

1. Penerapan algoritma *cosine similarity* dalam pengacakan ujian online dapat dilakukan, dengan terlebih dahulu melalui tahapan *preprocessing* soal serta kategori dan penggunaan metode *Term Frequency-Inverse Document Frequency* (TF-IDF). Serta hanya digunakan sebelum melakukan sistem pengacakan ujian *online* dengan cara memvalidasi apakah ada soal yang telah diambil itu sama dengan membandingkan nilai kesamaan *cosine similarity* antar soal yang sebelumnya telah diambil dan juga digunakan untuk mengetahui ada berapa banyak kategori soal dari hasil tahapan perhitungan nilai *cosine similarity* nya.
2. Serta untuk efektifitasnya dalam sistem pengacakan soal dapat dikatakan efektif digunakan, hal ini berdasarkan hasil validasi menggunakan metode pengujian *Mean Absolute Error* (MAE) dan *Root Mean Squared Error* (RMSE) yang dibuktikan dengan distribusi kategori dari tiap pengacakan yang akurat dilihat dari nilai aktual dan nilai prediksinya.

B. Saran

Berdasarkan hasil kesimpulan serta analisis penelitian dari algoritma ini, berikut saran yang diberikan untuk pengembangan lebih lanjut:

1. Menambahkan algoritma pengacakan tambahan untuk membantu penyempurnaan penggunaan algoritma *cosine similarity* dalam mendistribusi soal secara merata per kategori,
2. Menggunakan metode validasi lain dengan tujuan untuk menguji keefektifitas algoritma *cosine similarity* dalam membantu sistem pengacakan dari sudut pandang lain,
3. Menggunakan metode validasi lain dengan tujuan untuk menguji apakah soal dari tiap pengacakan telah cukup bervariasi,
4. Menambahkan data soal lebih banyak terutama soal pilihan ganda dari setiap kategori untuk pengujian lebih lanjut,
5. Dalam penelitian ini tepatnya pada tahapan *preprocessing* tidak dilakukan proses untuk menghilangkan *stopwords* atau kata hubung karena dirasa penting untuk tetap digunakan, jadi untuk penelitian selanjutnya dapat dicoba dengan menghilangkannya pada tahapan *preprocessing*.

Dengan memperhatikan saran-saran tersebut, diharapkan dapat membantu penelitian dan pengembangan lebih lanjut serta dapat menyempurnakan sistem pengacakan menggunakan algoritma *cosine similarity*.

DAFTAR PUSTAKA

Andriani, N., & Wibowo, A. (2021). Implementasi Text Mining Klasifikasi Topik Tugas Akhir Mahasiswa Teknik Informatika Menggunakan Pembobotan TF-IDF dan Metode Cosine Similarity Berbasis Web. *Senamika*, *September*, 130–137.

Askar, A., Pasnur, P., Asrul, A., Amiruddin, A., Resha, M., & Wijaya T, A. (2023). Implementation of Random Shuffle Algorithm to Randomize Questions in Anti-Corruption Prevention Game. *Brilliance: Research of Artificial Intelligence*, *3*(2), 244–251. https://doi.org/10.47709/brilliance.v3i2.3126

Azmi, M. (2022). Analisis Tingkat Plagiasi Dokumen Skripsi Dengan Metode Cosine Similarity Dan Pembobotan Tf-Idf. *TEKNIMEDIA: Teknologi Informasi Dan Multimedia*, *2*(2), 90–95. https://doi.org/10.46764/teknimedia.v2i2.51

Candra Susanto, P., Ulfah Arini, D., Yuntina, L., Panatap Soehaditama, J., & Nuraeni, N. (2024). Konsep Penelitian Kuantitatif: Populasi, Sampel, dan Analisis Data (Sebuah Tinjauan Pustaka). *Jurnal Ilmu Multidisplin*, *3*(1), 1–12. https://doi.org/10.38035/jim.v3i1.504

Dafitri, H., Aulia, R., Usman, A., & Fathia, J. N. (2023). Aplikasi Ujian Tryout Berbasis Client Server Menggunakan Linier Congruent Method (LCM) Pada Sekolah Menengah Kejuruan. *Explorer*, *3*(2), 88–94.

Darwis, M., Pranoto, G. T., Wicaksana, Y. E., & Yaddarabullah, Y. (2020). Implementation of TF-IDF Algorithm and K-mean Clustering Method to Predict Words or Topics on Twitter. *JISA(Jurnal Informatika Dan Sains)*, *3*(2), 49–55. https://doi.org/10.31326/jisa.v3i2.831

Daulay, A. A., & Ekadiansyah, E. (2024). Metode LCM dan Dice Coefficient dalam Pengacakan Soal Ujian di SMK Swasta Teladan Medan. *Jurnal Info Digit (JID)*, *2*(2), 514–531.

Fanani, N. A., Dia, A., Sari, I., Guru, P., Dasar, S., Gresik, U. M., Guru, P., Dasar, S., Gresik, U. M., & Karakter, P. (2024). *ISSN 3030-8496 Jurnal Matematika dan Ilmu Pengetahuan Alam*. *1*(2), 21–32.

Hanif Ridwannulloh, M. (2021). Implementasi Algoritma Fisher Yates Shuffle

Dalam Pembuatan Ujian Online Berbasis Web. *Jurnal Informatika-COMPUTING*, *08*, 16–21.

Hodson, T. O. (2022). Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not. *Geoscientific Model Development*, *15*(14), 5481–5487. https://doi.org/10.5194/gmd-15-5481-2022

Mulyana, D. I., & Marjuki. (2022). Optimasi Prediksi Harga Udang Vaname Dengan Metode Rmse Dan Mae Dalam Algoritma Regresi Linier. *Jurnal Ilmiah Betrik*, *13*(1), 50–58. https://doi.org/10.36050/betrik.v13i1.439

Prakarsa, A., Sunarto, A. A., & Prajoko, P. (2020). Model Pengacakan Soal Ujian Online SMA Menggunakan Metode Linear Congruential Generator dan Fisher Yates. *Progresif: Jurnal Ilmiah Komputer*, *16*(2), 133. https://doi.org/10.35889/progresif.v16i2.519

Prasetyadi, R., & Budi Nugroho, N. (2020). Implementasi Metode Multiplicative Random Number Generator (MRNG) Pada Aplikasi Ujian Sekolah Berbasis Komputer. *Jurnal CyberTech*, *3*(2), 224–229. https://ojs.trigunadharma.ac.id/

Qhorifadillah, U., Lestari, S., & Chulkamdi, M. T. (2022). Perancangan Aplikasi Bank Soal Berbasis Website Dengan Algoritma Fisher Yates Shuffle Dan Cosine Similarity (Studi Kasus Di Smk Indraprasta Wlingi). *JATI (Jurnal Mahasiswa Teknik Informatika)*, *6*(1), 352–359. https://doi.org/10.36040/jati.v6i1.4232

Rusdiyanto, R., Hakim, L., & Martadinata, A. T. (2022). Aplikasi Ujian Online Dan Penerapan Algoritma Lcg Untuk Proses Pengacakan Soal Ujian Di Smk Negeri Tugumulyo. *JUTIM (Jurnal Teknik Informatika Musirawas)*, *7*(2), 99–108. https://doi.org/10.32767/jutim.v7i2.1764

Saputra, R. A., Awalda Tariza, I., & Pramono, B. (2022). Implementasi Algoritma Multiply With Carry Generator (MWCG) dalam Pengacakan Soal Ujian Semester Berbasis Web pada SMKN 1 Kendari. *Maret*, *7*(1), 60–67. http://openjournal.unpam.ac.id/index.php/informatika60

Setiawan, I., Suprihatin, H., & Pujiastuti, E. (2022). Pengembangan Sistem Computer Based Test Pada Smk Bintang Harapan Cibarusah Bekasi Untuk Pelaksanaan Ujian. *Jurnal AbdiMas Nusa Mandiri*, *4*(2), 38–42.

https://doi.org/10.33480/abdimas.v4i2.2878

Sukmaningtyas, Y. N., Akbar, R. M., & Rohma, G. (2024). *Penerapan Predictive Analytics untuk Analisis Faktor-faktor yang Mempengaruhi Performa Akademik Siswa menganalisis faktor-faktor yang mempengaruhi performa akademik siswa telah*. *4*(2), 127–145.

Suriani, N., Risnita, & Jailani, M. S. (2023). Konsep Populasi dan Sampling Serta Pemilihan Partisipan Ditinjau Dari Penelitian Ilmiah Pendidikan. *Jurnal IHSAN : Jurnal Pendidikan Islam*, *1*(2), 24–36. https://doi.org/10.61104/ihsan.v1i2.55

Wardani, S. U. K. (2021). Efektivitas Penggunaan Sistem Computer Based Test dan Paper Based Test dalam Pelaksanaan Ujian Tengah Semester Bahasa Indonesia di SMPN 6 Singaraja. *Jurnal Pendidikan Bahasa Dan Sastra Indonesia Undiksha*, *11*(4), 491. https://doi.org/10.23887/jjpbs.v11i4.39676

LAMPIRAN

Lampiran 1. Source Code

Sel 1, koneksi *jupyter notebook* ke *database MySQL*:

```
import mysql.connector
import pandas as pd

# Koneksi ke database MySQL
try:
    conn = mysql.connector.connect(
        host="localhost",
        user="root",
        password="1234",
        database="skripsi"
    )
    cursor = conn.cursor()
    print("Koneksi ke database berhasil!")
except mysql.connector.Error as err:
    print(f"Koneksi ke database gagal: {err}")
```

Sel 2, impor modul untuk pengolahan dan pemrosesan data:

```
from sqlalchemy import create_engine
import warnings

# Sembunyikan peringatan, karena tidak mengguanggu
jalannya kode
warnings.filterwarnings("ignore", category=UserWarning)

# Buat koneksi database ke SQLAlchemy
engine =
create_engine("mysql+pymysql://root:1234@localhost/skri
psi")

# Query untuk membaca data dari tabel
query = "SELECT kategori, teks_soal FROM soal_ujian;"
data = pd.read_sql(query, conn)

# Tampilkan data untuk memastikan
print("Sampel data :")
print(data.head(10))

# Query untuk menghitung jumlah soal per kategori
query = """
```

```
SELECT kategori, COUNT(*) AS total_soal
FROM soal_ujian
GROUP BY kategori;
"""

# Eksekusi query
cursor.execute(query)

# Ambil hasil dan masukkan ke dalam DataFrame
results = cursor.fetchall()
df = pd.DataFrame(results, columns=['Kategori', 'Total
Soal'])

# Tampilkan hasil
print("\n", df)
```

Sel 3, *preprocessing* data:

```
import string
import re
import nltk
from nltk.corpus import wordnet

# Mengimpor kolom yang akan digunakan untuk diproses
data = data[['kategori', 'teks_soal']]

# Fungsi membersihkan teks
def clean_text(text):
    text = text.lower()  # Mengubah ke huruf kecil
    text = text.replace("_", " ")  # Mengubah tanda "_"
menjadi spasi
    text = text.translate(str.maketrans('', '',
string.punctuation))  # Menghapus tanda baca
    text = text.strip()  # Menghapus spasi berlebih di
awal/akhir
    return text

# Preprocessing data teks pada kategori dan teks_soal
data['clean_kategori'] =
data['kategori'].apply(clean_text)
data['clean_teks_soal'] =
data['teks_soal'].apply(clean_text)

# Menghapus data yang kosong karena tidak digunakan
data = data.dropna()
```

```
# Menampilkan data yang telah diproses
print("\nSampel data sebelum dan setelah
preprocessing:\n")
print(data.head(10))
```

Sel 4, TF-IDF:

```
from sklearn.feature_extraction.text import
TfidfVectorizer

# Inisialisasi TF-IDF Vectorizer untuk teks soal dan
kategori
tfidf_vectorizer_teks = TfidfVectorizer()
tfidf_vectorizer_kategori = TfidfVectorizer()

# Transformasi teks soal dan kategori ke representasi
TF-IDF
tfidf_matrix_teks =
tfidf_vectorizer_teks.fit_transform(data['clean_teks_so
al'])
tfidf_matrix_kategori =
tfidf_vectorizer_kategori.fit_transform(data['clean_kat
egori'])

# Menampilkan hasil TF-IDF
print("\nTF-IDF Matrix untuk Teks Soal (dalam bentuk
sparse):")
print(tfidf_matrix_teks)
print("\nTF-IDF Matrix untuk Kategori (dalam bentuk
sparse):")
print(tfidf_matrix_kategori)
```

Sel 5, menghitung nilai kesamaan *cosine similarity* antar soal dan antar kategori:

```
from sklearn.metrics.pairwise import cosine_similarity

# Menghitung cosine similarity antar teks soal & antar
kategori
cosine_sim_matrix_teks =
cosine_similarity(tfidf_matrix_teks)
cosine_sim_matrix_kategori =
cosine_similarity(tfidf_matrix_kategori)
```

```
# Menyimpan hasil cosine similarity ke dalam DataFrame
untuk teks soal & kategori
cosine_sim_df_teks =
pd.DataFrame(cosine_sim_matrix_teks, index=data.index,
columns=data.index)
cosine_sim_df_kategori =
pd.DataFrame(cosine_sim_matrix_kategori,
index=data.index, columns=data.index)

# Menampilkan hasil cosine similarity untuk teks soal &
kategori (5x5 sampel)
print("\nCosine Similarity Matrix untuk Teks Soal
(sampel 5x5):")
print(cosine_sim_df_teks.head(5))
print("\nCosine Similarity Matrix untuk Kategori
(sampel 5x5):")
print(cosine_sim_df_kategori.head(5))
```

Sel 6, hitung rata-rata nilai kesamaan setiap soal dan setiap kategori serta mengurutkan dari nilai kesamaan terendah:

```
query = "SELECT kategori, teks_soal FROM soal_ujian;"
data = pd.read_sql(query, conn)

# Menghitung rata-rata nilai cosine similarity untuk
setiap soal & setiap kategori
cosine_mean_similarity_teks =
cosine_sim_df_teks.mean(axis=1)
cosine_mean_similarity_kategori =
cosine_sim_df_kategori.mean(axis=1)

# Menambahkan kolom baru pada DataFrame untuk nilai
rata-rata cosine similarity
data['cosine_mean_similarity_teks'] =
cosine_mean_similarity_teks
data['cosine_mean_similarity_kategori'] =
cosine_mean_similarity_kategori

# Mengurutkan soal berdasarkan nilai cosine similarity
untuk teks soal (dari terendah ke tertinggi)
sorted_data_teks =
data.sort_values(by='cosine_mean_similarity_teks',
ascending=True)
```

```
# Tampilkan urutan soal berdasarkan cosine similarity
terkecil untuk teks soal
print("Soal berdasarkan nilai cosine similarity TEKS
SOAL (terendah hingga tertinggi):")
for i, row in sorted_data_teks.iterrows():
    print(f"Soal {i + 1}:")
    print(f"  Kategori: {row['kategori']}")
    print(f"  Teks Soal: {row['teks_soal']}")
    print(f"  Nilai rata-rata cosine similarity (Teks
Soal): {row['cosine_mean_similarity_teks']:.4f}")
```

Sel 7, mengelompokkan soal dengan nilai kesamaan yang identik dan tidak identik:

```
# Mengelompokkan soal yang sama persis
# Ambang batas untuk menentukan soal identik
similarity_threshold = 1.0  # Nilai 1.0 untuk kesamaan
sempurna

# List untuk menyimpan grup soal
groups = []
visited = set()  # Menyimpan indeks soal yang sudah
diperiksa

# Iterasi melalui matriks cosine similarity
for i in range(len(cosine_sim_matrix_teks)):
    if i in visited:
        continue  # Lewati soal yang sudah
dikelompokkan

    # Cari soal yang memiliki nilai similarity >=
threshold
    similar_indices = [j for j in
range(len(cosine_sim_matrix_teks)) if
cosine_sim_matrix_teks[i, j] >= similarity_threshold
and j != i]

    if similar_indices:
        # Tambahkan soal yang mirip ke dalam grup
        group = [i] + similar_indices
        groups.append(group)
        visited.update(group)  # Tandai soal dalam grup
sebagai sudah diperiksa
    else:
        groups.append([i])  # Soal yang tidak punya
pasangan dimasukkan sendiri
```

```
# Tampilkan hasil grup soal identik
print("Kelompok soal dengan cosine similarity =
{:.2f}:".format(similarity_threshold))
for idx, group in enumerate(groups, 1):
    print(f"\nGroup {idx}:")
    for soal_idx in group:
        print(f"  - Soal {soal_idx + 1}:
{data.iloc[soal_idx]['teks_soal']}")
```

Sel 8, pengacakan soal dengan menentukan batas kemiripan soal:

```
import numpy as np
import pandas as pd
import warnings

# Mengacak soal secara random berdasarkan kategori
# Menyembunyikan peringatan
warnings.filterwarnings("ignore",
category=DeprecationWarning)

# Jumlah soal yang ingin diambil dalam setiap
pengacakan
total_questions = 50
num_iterations = 50

# Batas kemiripan soal yang diambil
similarity_threshold = 0.9

# Mengelompokkan Soal Berdasarkan Cosine Similarity
dalam group_id
groups_dict = {}
visited = set()

for i in range(len(cosine_sim_matrix_teks)):
    if i in visited:
        continue

    similar_indices =
np.where(cosine_sim_matrix_teks[i] >= 1.0)[0]
    group = set(similar_indices)
    groups_dict[i] = group
    visited.update(group)
```

```
groups = [list(group) for group in
groups_dict.values()]

# Membuat DataFrame grup soal yang identik
grouped_data = pd.DataFrame({'group_id':
range(len(groups)), 'indices': groups})

data['group_id'] = -1
for group_id, indices in enumerate(groups):
    data.loc[indices, 'group_id'] = group_id

# Melakukan Proses Pengacakan
random_samples = []

for iteration in range(num_iterations):
    np.random.seed(iteration)  # Variasi pengacakan
setiap iterasi

    # Ambil satu soal secara acak dari setiap group_id
    unique_groups =
data.groupby('group_id').sample(n=1,
random_state=iteration).copy()

    # Reset sampled_data untuk setiap iterasi
    sampled_data = []
    selected_indices = []

    # Hitung total kategori berdasarkan nilai cosine
similarity
    category_similarity_scores =
cosine_mean_similarity_kategori.to_dict()
    sorted_categories =
sorted(category_similarity_scores,
key=category_similarity_scores.get)

    # menentukan quota soal per kategori dengan
mempertimbangkan similarity
    category_counts =
data['kategori'].value_counts().to_dict()
    min_per_category = total_questions //
len(category_counts)
    extra_questions = total_questions %
len(category_counts)
```

```
    category_quota = {cat: min_per_category for cat in
category_counts}

    # Memberi ekstra soal ke kategori dengan nilai
cosine similarity kategori terendah
    for cat in sorted_categories:
        if extra_questions > 0 and cat in
category_quota:
            category_quota[cat] += 1
            extra_questions -= 1

    # Daftar kandidat soal berdasarkan kategori
similarity
    candidate_indices = list(unique_groups.index)

    # Loop untuk memilih soal berdasarkan quota
kategori dengan memperhatikan similarity
    while len(sampled_data) < total_questions and
candidate_indices:
        candidate_index =
np.random.choice(candidate_indices)
        candidate_category = data.loc[candidate_index,
'kategori']

        # Periksa apakah soal terlalu mirip dengan soal
yang sudah dipilih
        if category_quota[candidate_category] > 0:
            is_similar = any(
                cosine_sim_matrix_teks[selected_index,
candidate_index] >= similarity_threshold
                for selected_index in selected_indices
            )

            # Tambahkan ke hasil pengacakan jika tidak
terlalu mirip
            if not is_similar:
                sampled_data.append(unique_groups.loc[c
andidate_index])
                selected_indices.append(candidate_index
)
                category_quota[candidate_category] -= 1

        # Hapus soal tersebut dari kandidat
        candidate_indices.remove(candidate_index)
```

```
    # Tambahkan soal jika masih kurang, dengan
mempertimbangkan distribusi kategori
    while len(sampled_data) < total_questions:
        remaining_candidates =
data.loc[~data.index.isin(selected_indices)]

        if remaining_candidates.empty:
            break  # Tidak ada lagi soal yang bisa
diambil

        # Hitung kategori yang sudah terisi
        current_category_counts = pd.Series([q.kategori
for q in sampled_data]).value_counts().to_dict()

        # Temukan kategori yang paling sedikit terisi
        least_filled_category =
min(current_category_counts,
key=current_category_counts.get, default=None)

        if least_filled_category:
            category_candidates =
remaining_candidates[remaining_candidates['kategori']
== least_filled_category]
            if not category_candidates.empty:
                extra_sample =
category_candidates.sample(n=1)
            else:
                extra_sample =
remaining_candidates.sample(n=1)  # Jika tidak ada soal
di kategori tersebut
        else:
            extra_sample =
remaining_candidates.sample(n=1)  # Ambil sembarang
jika semua kategori sudah seimbang

        sampled_data.append(extra_sample.iloc[0])
        selected_indices.append(extra_sample.index[0])


    # Simpan hasil setiap iterasi ke dalam
random_samples
    random_samples.append(pd.DataFrame(sampled_data))

# Fungsi untuk Menampilkan Hasil
def display_random_sample(iteration):
```

```
    if iteration < 1 or iteration > num_iterations:
        print(f"Iterasi yang dipilih tidak valid. Harus
antara 1 dan {num_iterations}.")
        return

    print(f"\nHasil Pengacakan Iterasi Ke-
{iteration}:")
    print(random_samples[iteration - 1][['kategori',
'teks_soal', 'cosine_mean_similarity_teks']])

    print(f"\nDistribusi per kategori (Iterasi Ke-
{iteration}):")
    print(random_samples[iteration -
1]['kategori'].value_counts())

# Input Pengguna untuk Memilih Iterasi
while True:
    try:
        selected_iteration = int(input(f"Masukkan nomor
iterasi yang ingin ditampilkan (1-{num_iterations}):
"))
        display_random_sample(selected_iteration)
        break
    except ValueError:
        print("Input harus berupa angka.")
```

Sel 9, validasi distribusi soal dengan MAE dan RMSE:

```
from sklearn.metrics import mean_absolute_error,
mean_squared_error
import numpy as np

# Target pengacakan soal per kategori
total_questions = 50
num_categories = len(data['kategori'].unique())
base_questions_per_category = total_questions //
num_categories
extra_questions = total_questions % num_categories

# Hitung target distribusi ideal
ideal_distribution = [base_questions_per_category] *
num_categories
for i in range(extra_questions):
    ideal_distribution[i] += 1
```

```
# Simpan hasil validasi
validation_results = []

# Validasi setiap iterasi pengacakan
for iteration, sample in enumerate(random_samples):
    # Hitung distribusi soal aktual per kategori
    actual_distribution =
sample['kategori'].value_counts().reindex(data['kategor
i'].unique(), fill_value=0).tolist()

    # Hitung MAE dan RMSE
    mae = mean_absolute_error(ideal_distribution,
actual_distribution)
    rmse =
np.sqrt(mean_squared_error(ideal_distribution,
actual_distribution))

    # Simpan hasil validasi
    validation_results.append({
        'iteration': iteration + 1,
        'MAE': mae,
        'RMSE': rmse,
        'actual_distribution': actual_distribution
    })

# Tampilkan hasil validasi untuk semua iterasi
for result in validation_results:
    print(f"Iterasi {result['iteration']}:")
    print(f"  Distribusi Aktual  :
{result['actual_distribution']}")
    print(f"  Distribusi Ideal   :
{ideal_distribution}")
    print(f"  MAE                : {result['MAE']:.4f}")
print(f"  RMSE               : {result['RMSE']:.4f}\n")

# Rata-rata MAE dan RMSE untuk semua iterasi
average_mae = np.mean([result['MAE'] for result in
validation_results])
average_rmse = np.mean([result['RMSE'] for result in
validation_results])

print(f"Rata-rata MAE untuk {num_iterations} iterasi:
{average_mae:.4f}")
print(f"Rata-rata RMSE untuk {num_iterations} iterasi:
{average_rmse:.4f}")
```

MAJELIS PENDIDIKAN TINGGI PIMPINAN PUSAT MUHAMMADIYAH
UNIVERSITAS MUHAMMADIYAH MAKASSAR
UPT PERPUSTAKAAN DAN PENERBITAN
Alamat kantor: Jl.Sultan Alauddin NO.259 Makassar 90221 Tlp.(0411) 866972,881593, Fax.(0411) 865588

بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ

SURAT KETERANGAN BEBAS PLAGIAT

UPT Perpustakaan dan Penerbitan Universitas Muhammadiyah Makassar,
Menerangkan bahwa mahasiswa yang tersebut namanya di bawah ini:

Nama : Aidhil Prima Abdiguna

Nim : 105841100920

Program Studi : Teknik Informatika

Dengan nilai:

No	Bab	Nilai	Ambang Batas
1	Bab 1	8 %	10 %
2	Bab 2	11 %	25 %
3	Bab 3	8 %	10 %
4	Bab 4	8 %	10 %
5	Bab 5	3 %	5 %

Dinyatakan telah lulus cek plagiat yang diadakan oleh UPT- Perpustakaan dan Penerbitan Universitas Muhammadiyah Makassar Menggunakan Aplikasi Turnitin.

Demikian surat keterangan ini diberikan kepada yang bersangkutan untuk dipergunakan seperlunya.

Makassar, 04 Februari 2025
Mengetahui,

Kepala UPT- Perpustakaan dan Pernerbitan,

Nursinah, S.Hum.,M.I.P
NBM. 964 591

Jl. Sultan Alauddin no 259 makassar 90222
Telepon (0411)866972,881 593,fax (0411)865 588
Website: www.library.unismuh.ac.id
E-mail : perpustakaan@unismuh.ac.id