Vol. 10, No. 2, Oktober 2025, Page. 298-310
https://journal.uin-alauddin.ac.id/index.php/instek
p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

Penerapan Algoritma *Cosine Similarity* dalam Efektifitas Pengacakan Soal Ujian *Online*

Lukman[*1], Aidhil Prima Abdiguna[2], Rizki Yusliana Bakti[3]

[1,2,3]Informatika, Fakultas Teknik, Universitas Muhammadiyah Makassar, Indonesia
Email: [1]lukman@unismuh.ac.id, [2]105841100920@student.unismuh.ac.id,
[3]rizkiyusliana@unismuh.ac.id

Abstrak

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 penggunaan algoritma ini dinilai efektif dikarenakan selisih rata-rata antara hasil sistem dan ideal berada dikisaran 0-1, dimana berdasarkan validasi *Mean Absolute Error* (MAE) sebesar 0,2514 serta *Root Mean Squared Error* (RMSE) sebesar 0,4704, yang menunjukkan tingkat efektivitas tinggi dalam proses pengacakan.

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

Abstract

Effective randomization 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 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 this algorithm is considered effective because the average difference between the system and ideal results in the range of 0-1, where based on the validation from Mean Absolute Error (MAE) of 0.2514 and Root Mean Squared Error (RMSE) of 0.4704, indicating a high level of effectiveness in the randomization process.

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

1. PENDAHULUAN

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*[1] [2]. Ujian yang menggunakan metode berbasis komputer ataupun perangkat lainnya biasanya dilakukan secara daring

p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

atau *online* menawarkan berbagai keuntungan, termasuk di antaranya 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[3], [4]. Namun, 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[5]. Pengacakan soal yang efektif, tidak hanya harus memastikan pendistribusian soal yang diberikan tetapi juga harus mempertimbangkan kesamarataan antara soal-soal yang diberikan[6].

Penggunaan algoritma pengacakan tradisional seperti *fisher yates shuffle* dan *linear congruential generator(LCG)* memang menghasilkan distribusi acak secara numerik, tetapi tidak mempertimbangkan isi soal. Akibatnya, dua peserta ujian bisa saja menerima kumpulan soal dengan tingkat kesulitan atau topik yang sangat terkonsentrasi pada satu kategori tertentu. Kondisi ini berpotensi menimbulkan ketidakadilan dalam evaluasi karena sebagian peserta mungkin mendapatkan soal yang lebih mudah atau lebih sulit dibandingkan peserta lain[7], [8]. Oleh karena itu, diperlukan metode pengacakan yang tidak hanya acak secara statistik tetapi juga mempertimbangkan kesetaraan konten antar soal, seperti pendekatan berbasis Cosine Similarity yang mengukur kesamaan teks secara kontekstual[9], [10].

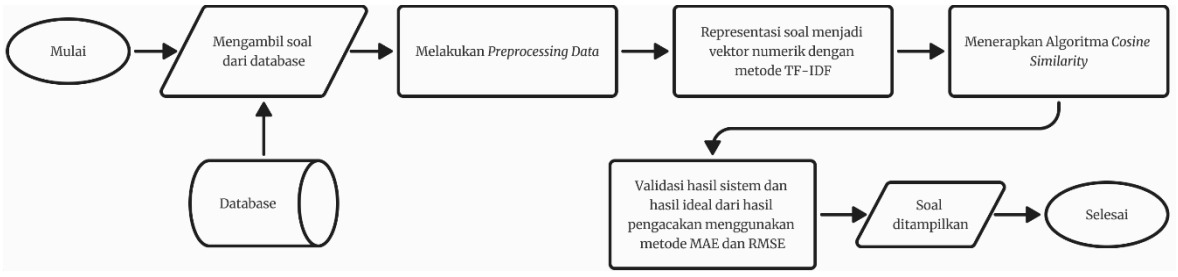
Beberapa penelitian sebelumnya yang menggunakan algoritma pengacakan seperti *fisher yates shuffle* dan *linear congruential generator(LCG)* memang mampu menghasilkan urutan soal yang acak, tetapi belum mempertimbangkan tingkat kesamaan konten antar soal[7], [11], [12]. Hal ini menyebabkan kemungkinan adanya pertanyaan yang terlalu mirip atau bahkan identik dalam satu set ujian. Selain itu, metode-metode tersebut hanya mengandalkan distribusi numerik tanpa mempertimbangkan isi soal, sehingga tidak menjamin variasi topik yang seimbang di setiap kategori soal[13], [14]. Oleh karena itu, diperlukan pendekatan berbasis kesamaan teks seperti Cosine Similarity untuk memastikan setiap soal yang diacak tidak memiliki kemiripan tinggi satu sama lain[15].

Penelitian ini berfokus pada penerapan algoritma cosine similarity dalam sistem pengacakan soal ujian *online*. Permasalahan yang diangkat mencakup bagaimana cara algoritma cosine similarity bisa diterapkan dalam sistem pengacakan soal serta apakah penerapan algoritma ini efektif digunakan dalam sistem pengacakan soal ujian *online*.

2. METODE PENELITIAN

2.1. Perancangan Sistem

2.1.1. *Flowchart* Sistem



Gambar 1. *Flowchart* Sistem

Untuk lebih memudahkan pembaca dalam memahami maksud dari gambar di atas berikut adalah penjelasannya:

a. Mulai, ini adalah tahap di mana proses dari sistem yang telah dibuat dilakukan,
b. *Database*, tempat kumpulan soal-soal serta masing-masing kategorinya tersimpan,
c. Mengambil soal dari *database*, pada tahap ini sistem mengambil soal yang tersedia pada *database*,
d. Melakukan *Preprocessing Data*, pada tahap ini soal yang telah diambil dari database diproses atau dibersihkan agar lebih mudah dilakukan pemrosesan oleh sistem,
e. Representasi soal menjadi vektor numerik dengan metode TF-IDF, di tahap ini dilakukan representasi soal menjadi vektor numerik dari *database*, hal ini dilakukan agar algoritma *cosine similarity* dapat melakukan perhitungan nilai kesamaan,

p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

f. Menerapkan Algoritma *Cosine Similarity*, pada tahap ini soal yang telah direpresentasikan kemudian diproses oleh *Cosine Similarity* untuk menghitung, mengelompokkan serta melakukan pengacakan berdasarkan nilai kesamaan antar soal,
g. Validasi hasil sistem dan hasil ideal dari hasil pengacakan menggunakan metode *Mean Absolute Error* (MAE) dan *Root Mean Squared Error* (RMSE), pada tahap ini, hasil sistem dan hasil ideal soal kemudian divalidasi menggunakan metode MAE dan RMSE, untuk memastikan bahwa soal-soal yang dibuat memenuhi standar pengacakan yang ideal,
h. Soal ditampilkan, pada tahap ini soal ditampilkan dari hasil pengacakan,
i. Selesai, Proses sistem ini selesai.

2.2. Tahapan Analisis Data

Berikut adalah langkah-langkah yang digunakan dalam penelitian untuk menganalisis data:

2.2.1. Menyiapkan Dataset

Tabel 1. Jumlah Dataset 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**

Langkah pertama ialah menyiapkan *dataset*, data yang dipakai berupa soal-soal lomba dari Markas PMI Kota Makassar, yang dibagi menjadi 7 kategori yang diambil dari 7 materi pokok Kepalangmerahan PMI. Dapat dilihat pada tabel 1, kategori apa saja yang digunakan, total soal per kategori serta total dari keseluruhan soal dalam *database*.

2.2.2. *Preprocecssing 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. menghapus tanda baca, angka maupun karakter khusus,
c. menghapus spasi berlebih di awal dan akhir soal.

2.2.3. Penggunaan Metode TF-IDF

TF-IDF (*Term Frequency-Inverse Document Frequency*) adalah metode yang digunakan dalam teks *mining* untuk menilai seberapa penting suatu kata dalam sebuah dokumen, relatif terhadap sekumpulan dokumen[15]. TF-IDF digunakan untuk merepresentasikan kata-kata dalam bentuk vektor numerik, dan merupakan dasar dalam algoritma pengukuran kesamaan, seperti *cosine similarity*[9] [10].

Di sini (t) diibaratkan kata sedangkan (d) adalah soal

a. Rumus TF-IDF:

$$TF - IDF(t,d) = TF(t,d) \times IDF(t,D) \quad (1)$$

p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

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 (*Term Frequency*), mengukur frekuensi kemunculan suatu kata dalam soal tertentu. Jika sebuah kata sering muncul dalam soal, nilainya akan lebih tinggi.

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

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

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

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 ini digunakan untuk proses mengubah vektor tiap kata dalam *database* soal ke dalam bentuk numerik dan menganalisis soal ujian berdasarkan frekuensi kata yang muncul di dalamnya sebelum ke proses selanjutnya.

Pemilihan metode TF-IDF didasarkan pada kesederhanaan dan kemampuannya dalam menangkap bobot penting setiap kata tanpa memerlukan dataset besar untuk pelatihan. TF-IDF efektif dalam mengurangi pengaruh kata-kata umum (seperti “yang”, “dan”, “atau”) serta memberikan penekanan lebih pada kata-kata yang khas dari setiap soal. Hal ini menjadikan TF-IDF ideal untuk sistem pengacakan soal, karena representasi vektornya tetap ringan secara komputasi namun cukup informatif untuk digunakan dalam perhitungan kesamaan dengan algoritma *cosine similarity*.

2.2.4. Penerapan *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[12]. 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:

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

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.

Secara garis besar, alur metode pengacakan yang digunakan adalah sebagai berikut:

p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

a. Menghitung nilai kesamaan antar soal menggunakan *cosine similarity*,
b. Mengurutkan soal berdasarkan nilai kesamaan dari yang terendah ke tertinggi,
c. Mengeelompokkan soal-soal dengan nilai kesamaan yang identik,
d. Melakukan proses pengacakan dengan mengambil satu soal secara acak dari setiap kelompok untuk memastikan keragaman,
e. Membatasi pengambilan soal dengan ambang kesamaan di bawah nilai 0.9 agar tidak muncul soal yang terlalu mirip dalam satu set ujian,
f. Menyimpan hasil pengacakan dan lakukan validasi efektivitas menggunakan MAE dan RMSE.

Nilai ambang batas 0.9 dipilih berdasarkan hasil eksperimen awal dan mengacu pada penelitian sebelumnya yang menunjukkan bahwa nilai kesamaan di atas 0.9 umumnya merepresentasikan teks dengan kemiripan sangat tinggi atau identik[8], [9]. Dengan batas ini, sistem dapat menyeimbangkan antara menjaga variasi soal dan menghindari kemiripan berlebihan tanpa mengurangi pengacakan.

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

Mean Absolute Error (MAE) dan *Root Mean Squared Error* (RMSE) adalah dua metode untuk mengevaluasi evaluasi tingkat kesalahan dalam hasil ideal suatu model terhadap hasil sistem[14]. Hasil sistem merupakan nilai yang sebenarnya terjadi dari proses yang telah dilakukan, sedangkan hasil ideal adalah hasil yang diharapkan dari proses yang dilakukan[16]. Dalam penelitian ini, "hasil ideal" merujuk pada distribusi soal yang merata untuk setiap kategori, yakni jumlah soal total dibagi dengan jumlah kategori. Pendekatan ini dianggap ideal karena dalam ujian yang adil, setiap peserta harus mendapatkan komposisi soal yang seimbang dari seluruh kategori materi. Ketidakseimbangan jumlah soal antar kategori dapat menyebabkan ketidakadilan antar peserta dalam mendapatkan variasi kategori soal, sehingga pengujian efektivitas algoritma dilakukan dengan membandingkan hasil pengacakan sistem terhadap distribusi ideal tersebut.

a. Mean Absolute Error (MAE)

MAE adalah rata-rata dari nilai absolut selisih antara hasil sistem dan hasil ideal, rumusnya:

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

Di mana yi adalah hasil sistem, $\widehat{yi}$ adalah hasil ideal, dan n adalah jumlah data. MAE memberikan gambaran langsung tentang tingkat kesalahan hasil ideal rata-rata tanpa tanda.

b. Root Mean Squared Error (RMSE)

RMSE adalah akar kuadrat dari rata-rata kuadrat selisih antara hasil sistem dan hasil ideal, 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 hasil ideal yang sangat meleset.

Metode MAE dan RMSE digunakan untuk memvalidasi nilai antara hasil sistem dan hasil ideal berdasarkan kategori yang ditampilkan dari hasil pengacakan. Tujuannya adalah untuk memastikan apakah penerapan algoritma *cosine similarity* sesuai dalam membantu efektifitas sistem pengacakan.

Untuk menentukan apakah hasil dari pengacakan dengan algoritma *cosine similarity* efektif atau tidak kita perlu memvalidasinya, misalnya dalam 50 soal yang diambil dengan 7 kategori yang berbeda maka hasil idealnya adalah 7 soal per kategori serta terdapat penambahan 1 soal untuk salah satu kategori secara acak, dan jika selisih hasil sistem dan hasil idealnya berada di antara angka 0-1 maka bisa dikatakan pengacakan dengan algoritma tersebut efektif, jika berada di antara 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[16].

2.2.6. 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 50x serta terdapat sebanyak 50 soal dalam proses 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.

3. HASIL DAN PEMBAHASAN

3.1. Pengujian dan Hasil Model

Untuk proses pengujiannya dibagi menjadi beberapa tahapan, berikut tahapan-tahapan pengujiannya:

3.1.1. Preprocessing Data

Sebelum memasuki tahapan *preprocessing* data dilakukan, dilakukan koneksi *jupyter notebook* ke *database MySQL*. Setelah koneksi berhasil disambungkan selanjutnya yaitu mengimpor modul untuk pengolahan dan menampilkan data serta membuat koneksi antara *sqlalchemy* dan *database* dan kemudian menampilkannya. Masuk ke tahapan *preprocessing* data atau tahap pembersihan, maksud dari pembersihan ini yaitu agar dapat dengan mudah diproses oleh mesin dengan cara mengubahnya ke huruf kecil, mengubah atau menghapus tanda baca, serta menghapus spasi berlebih dari tiap-tiap soal.

3.1.2. Penerapan Metode Term Frequency-Inverse Document Frequency

Pada tahap ini data yang telah dibersihkan kemudian diproses dengan mengubahnya menjadi representasi vektor numerik, kemudian digunakan untuk mengubah data masing-masing soal dan kategori yang telah melewati *preprocessing* menjadi vektor numerik dan menampilkannya dalam bentuk sparse. Untuk hasil *output* dapat dilihat pada Gambar 2 dan 3 berikut ini.

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

Gambar 2. *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

Gambar 3. *Output* Hasil TF-IDF Kategori

Vol. 10, No. 2, Oktober 2025, Page. 298-310 p-ISSN: 2541-1179
https://journal.uin-alauddin.ac.id/index.php/instek e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

Gambar 2 dan 3 menampilkan sebagian sampel hasil perubahan vektor 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

3.1.3. 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:

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

Menggunakan algoritma *cosine similarity* untuk menghitung nilai kesamaan antar soal dan antar kategori yang telah melewati proses TF-IDF, kemudian hasilnya disimpan yang kemudian ditampilkan dalam matrix dengan sampel tampilan 5x5. Kemudian hasilnya disimpan dalam *dataframe* untuk diproses setelahnya. Untuk hasil *output* dapat dilihat pada gambar 4 dan 5 berikut ini.

```
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 4. *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 5. *Output Dataframe Matrix Cosine Similarity* Kategori

Penjelasan:

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

Vol. 10, No. 2, Oktober 2025, Page. 298-310
https://journal.uin-alauddin.ac.id/index.php/instek
p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

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

Proses ini mencakup mengambil kolom kategori dan 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 *output* dari tahapan ini dapat dilihat pada gambar 6 dan 7 berikut ini.

```
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
```

Gambar 6. *Output* Nilai *Cosine Similarity* Terendah

```
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 7. *Output* Nilai *Cosine Similarity* Tertinggi

Dapat dilihat dari sebagian sampel yang ditampilkan pada gambar 6 dan 7, 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.

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

Pada tahap ini, soal yang memiliki nilai kesamaan yang sama persis atau identik akan dimasukkan ke kelompok serta menandai soal yang sudah diperiksa maupun dikelompokkan, sementara soal yang tidak memiliki nilai kesamaan atau tidak mirip akan dibuatkan grup tersendiri. Untuk hasil *output* dapat dilihat pada gambar 8 dan 9 berikut ini.

```
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 ?
```

Gambar 8. *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 ?
```

Gambar 9. *Output* Soal-soal Tidak Identik

Pada gambar 8 dan 9, dapat dilihat dari sebagian sampel dari soal-soal yang dikelompokkan ke dalam grup soal yang identik satu memiliki kemiripan dengan soal yang lain serta soal yang tidak identik dengan soal lainnya yang ditampilkan, Hasilnya adalah terdapat banyak soal yang sama persis, dan juga banyak soal yang tidak memiliki kesamaan satu sama lain.

3.1.3.4. 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 berdasarkan nilai *cosine similarity* dan menambahkan 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 tiap-tiap grup serta memuat ulang 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. Kemudian membuat kandidat soal berdasarkan kategori similaritasnya, melakukan perulangan untuk memilih soal per kategori berdasarkan kemiripan nilai similaritasnya, 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 memasukkan 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 dapat dilihat pada gambar 10 hingga 13 berikut ini.

```
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 10. *Output* Hasil Pengacakan *Cosine Similarity* Ke-1 (Kategori)

p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

```
                                          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 11. *Output* Hasil Pengacakan *Cosine Similarity* Ke-1 (Soal)

```
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 12. *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 13. *Output* Hasil Pengacakan *Cosine Similarity* Ke-10 (Soal)

Dapat dilihat dari sebagian sampel yang ditampilkan pada gambar 10 hingga 13, yang di mana dapat dilihat perbedaan hasil dari proses pengacakan yang dilakukan.

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

Untuk proses dari tahapan ini yaitu pertama dengan 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 dari hasil sistem 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 dapat dilihat pada gambar 14 berikut ini.

```
Iterasi 1:
  Distribusi Hasil Sistem  : [7, 7, 7, 7, 8, 7, 7]
  Distribusi Hasil Ideal   : [8, 7, 7, 7, 7, 7, 7]
  MAE              : 0.2857
  RMSE             : 0.5345

Iterasi 2:
  Distribusi Hasil Sistem  : [7, 7, 7, 7, 8, 7, 7]
  Distribusi Hasil Ideal   : [8, 7, 7, 7, 7, 7, 7]
  MAE              : 0.2857
  RMSE             : 0.5345

Iterasi 3:
  Distribusi Hasil Sistem  : [7, 7, 7, 7, 7, 7, 8]
  Distribusi Hasil Ideal   : [8, 7, 7, 7, 7, 7, 7]
  MAE              : 0.2857
  RMSE             : 0.5345

Iterasi 4:
  Distribusi Hasil Sistem  : [7, 7, 8, 7, 7, 7, 7]
  Distribusi Hasil Ideal   : [8, 7, 7, 7, 7, 7, 7]
  MAE              : 0.2857
  RMSE             : 0.5345
```

Gambar 14. *Output* Distribusi Hasil Sistem dan Hasil Ideal serta MAE dan RMSE per Soal

Hasilnya, di sini dapat dilihat hasil sistem dan hasil ideal berdasarkan kategori yang diharapkan dan rata-rata nilai MAE dan RMSE per pengacakan.

3.2. 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 50x serta terdapat sebanyak 50 soal dalam proses pengacakan dengan 7 kategori soal berbeda.

Setelah semua tahapan di atas 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 proses pengacakan yang telah dilakukan.

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

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

Iterasi 50:
  Distribusi Hasil Sistem  : [7, 7, 7, 8, 7, 7, 7]
  Distribusi Hasil 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 15. Contoh Hasil Sistem dan Hasil Ideal serta MAE dan RMSE untuk 50x pengacakan dengan 50 soal

Dapat dilihat dari sebagian sampel yang ditampilkan pada gambar 15, hasil dari 50x pengacakan dengan 50 soal, dapat dilihat pada sistem yang telah dibuat distribusi idealnya adalah 7 soal per kategori serta terdapat penambahan 1 soal untuk salah satu kategori secara acak, dan dari hasil distribusi yang

p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

dilakukan oleh sistem didapatkan adalah soal terdistribusi secara merata dengan 7 soal per kategori serta terdapat penambahan 1 soal untuk salah satu kategori secara acak. Untuk nilai rata-rata validasi menggunakan MAE sebesar 0,2514 dan untuk nilai rata-rata validasi menggunakan RMSE sebesar 0,4704 yang berarti efektif dikarenakan selisihnya antara 0-1.

Untuk memperjelas keunggulan hasil pengacakan menggunakan algoritma *cosine similarity*, perbandingan dilakukan dengan metode pengacakan yang umum digunakan, yaitu *fisher yates shuffle* dan *linear congruential generator*(LCG). Kedua metode tersebut mampu menghasilkan urutan soal secara acak secara numerik, tetapi tidak mempertimbangkan kesamaan antar soal. Dalam pengujian awal, *fisher-yates shuffle* dan LCG menunjukkan sebaran soal yang tidak seimbang antar kategori, dengan potensi kemunculan soal yang serupa pada satu paket ujian. Sebaliknya, metode *cosine similarity* dapat mengurangi tingkat kemiripan tersebut dengan memperhatikan nilai kesamaan antar soal berdasarkan representasi vektor TF-IDF. Hal ini terbukti dari nilai rata-rata MAE dan RMSE. Dengan demikian, pendekatan berbasis *cosine similarity* menawarkan pengacakan yang lebih adil antara kesamaan soal dan kategori.

Meskipun hasil penelitian ini menunjukkan efektivitas penerapan algoritma *cosine similarity*, terdapat beberapa keterbatasan yaitu dataset yang digunakan masih terbatas pada satu sumber, pengukuran kesamaan dilakukan hanya berbasis pada kemiripan teks tanpa mempertimbangkan tingkat kesulitan soal. Keterbatasan ini dapat menjadi dasar pengembangan penelitian selanjutnya agar mencakup variasi dan parameter yang lebih luas. Secara praktis, hasil penelitian ini dapat diterapkan pada sistem ujian daring ataupun platform *e-learning* untuk memastikan distribusi soal yang lebih adil antar peserta. Dengan mempertimbangkan kesamaan antar soal, sistem ujian dapat mengurangi risiko ketimpangan kemiripan soal.

4. 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 hasil sistem dan hasil idealnya.
3. Penggunaan *Bloom Taxonomy* untuk melihat level soal agar hasil pengacakan yang dilakukan tidak timpang dalam hal tingkat kesulitan soal.

DAFTAR PUSTAKA

[1] I. Setiawan, H. Suprihatin, and E. Pujiastuti, “Pengembangan Sistem Computer Based Test Pada Smk Bintang Harapan Cibarusah Bekasi Untuk Pelaksanaan Ujian,” *J. AbdiMas Nusa Mandiri*, vol. 4, no. 2, pp. 38–42, 2022, doi: 10.33480/abdimas.v4i2.2878.

[2] R. A. Saputra, I. Awalda Tariza, and B. Pramono, “Implementasi Algoritma Multiply With Carry Generator (MWCG) dalam Pengacakan Soal Ujian Semester Berbasis Web pada SMKN 1 Kendari,” *Maret*, vol. 7, no. 1, pp. 60–67, 2022, [Online]. Available: http://openjournal.unpam.ac.id/index.php/informatika60

[3] S. U. K. Wardani, “Efektivitas Penggunaan Sistem Computer Based Test dan Paper Based Test dalam Pelaksanaan Ujian Tengah Semester Bahasa Indonesia di SMPN 6 Singaraja,” *J. Pendidik. Bhs. dan Sastra Indones. Undiksha*, vol. 11, no. 4, p. 491, 2021, doi: 10.23887/jjpbs.v11i4.39676.

p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

[4] N. A. Fanani *et al.*, "ISSN 3030-8496 Jurnal Matematika dan Ilmu Pengetahuan Alam," vol. 1, no. 2, pp. 21–32, 2024.

[5] A. A. Daulay and E. Ekadiansyah, "Metode LCM dan Dice Coefficient dalam Pengacakan Soal Ujian di SMK Swasta Teladan Medan," *J. Info Digit*, vol. 2, no. 2, pp. 514–531, 2024.

[6] M. Hanif Ridwannulloh, "Implementasi Algoritma Fisher Yates Shuffle Dalam Pembuatan Ujian Online Berbasis Web," *J. Inform.*, vol. 08, pp. 16–21, 2021.

[7] A. Prakarsa, A. A. Sunarto, and P. Prajoko, "Model Pengacakan Soal Ujian Online SMA Menggunakan Metode Linear Congruential Generator dan Fisher Yates," *Progresif J. Ilm. Komput.*, vol. 16, no. 2, p. 133, 2020, doi: 10.35889/progresif.v16i2.519.

[8] A. Askar, P. Pasnur, A. Asrul, A. Amiruddin, M. Resha, and A. Wijaya T, "Implementation of Random Shuffle Algorithm to Randomize Questions in Anti-Corruption Prevention Game," *Brill. Res. Artif. Intell.*, vol. 3, no. 2, pp. 244–251, 2023, doi: 10.47709/brilliance.v3i2.3126.

[9] N. Andriani and A. Wibowo, "Implementasi Text Mining Klasifikasi Topik Tugas Akhir Mahasiswa Teknik Informatika Menggunakan Pembobotan TF-IDF dan Metode Cosine Similarity Berbasis Web," *Senamika*, no. September, pp. 130–137, 2021.

[10] M. Darwis, G. T. Pranoto, Y. E. Wicaksana, and Y. Yaddarabullah, "Implementation of TF-IDF Algorithm and K-mean Clustering Method to Predict Words or Topics on Twitter," *JISA(Jurnal Inform. dan Sains)*, vol. 3, no. 2, pp. 49–55, 2020, doi: 10.31326/jisa.v3i2.831.

[11] R. Rusdiyanto, L. Hakim, and A. T. Martadinata, "Aplikasi Ujian Online Dan Penerapan Algoritma Lcg Untuk Proses Pengacakan Soal Ujian Di Smk Negeri Tugumulyo," *JUTIM (Jurnal Tek. Inform. Musirawas)*, vol. 7, no. 2, pp. 99–108, 2022, doi: 10.32767/jutim.v7i2.1764.

[12] U. Qhorifadillah, S. Lestari, and M. T. Chulkamdi, "Perancangan Aplikasi Bank Soal Berbasis Website Dengan Algoritma Fisher Yates Shuffle Dan Cosine Similarity (Studi Kasus Di Smk Indraprasta Wlingi)," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 6, no. 1, pp. 352–359, 2022, doi: 10.36040/jati.v6i1.4232.

[13] R. Prasetyadi and N. Budi Nugroho, "Implementasi Metode Multiplicative Random Number Generator (MRNG) Pada Aplikasi Ujian Sekolah Berbasis Komputer," *J. CyberTech*, vol. 3, no. 2, pp. 224–229, 2020, [Online]. Available: https://ojs.trigunadharma.ac.id/

[14] D. I. Mulyana and Marjuki, "Optimasi Prediksi Harga Udang Vaname Dengan Metode Rmse Dan Mae Dalam Algoritma Regresi Linier," *J. Ilm. Betrik*, vol. 13, no. 1, pp. 50–58, 2022, doi: 10.36050/betrik.v13i1.439.

[15] M. Azmi, "Analisis Tingkat Plagiasi Dokumen Skripsi Dengan Metode Cosine Similarity Dan Pembobotan Tf-Idf," *Tek. Teknol. Inf. dan Multimed.*, vol. 2, no. 2, pp. 90–95, 2022, doi: 10.46764/teknimedia.v2i2.51.

[16] T. O. Hodson, "Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not," *Geosci. Model Dev.*, vol. 15, no. 14, pp. 5481–5487, 2022, doi: 10.5194/gmd-15-5481-2022.