

# Penilaian Otomatis Cerdas Cermat Menggunakan Basis Data Sinonim Kata Dan *Cosine Similarity*

**Diterima:**  
10 Juni 2024  
**Revisi:**  
10 Juli 2024  
**Terbit:**  
1 Agustus 2024

**<sup>1\*</sup>Johan Rizky Triosaputra, <sup>2</sup>Ardi Sanjaya, <sup>3</sup>Julian Sahertian**  
<sup>1-3</sup>Universitas Nusantara PGRI Kediri  
<sup>1\*</sup>[johanrizkysa@gmail.com](mailto:johanrizkysa@gmail.com), <sup>2</sup>[dersky@gmail.com](mailto:dersky@gmail.com),  
<sup>3</sup>[juliansahertian@unpkediri.ac.id](mailto:juliansahertian@unpkediri.ac.id)

**Abstrak**— Penelitian ini bertujuan untuk mengembangkan aplikasi 'Cerdas Cermat' yang dapat mengevaluasi jawaban siswa dengan mengukur kemiripan antara jawaban siswa dan kunci jawaban menggunakan metode *cosine similarity* dan *database* sinonim. Metode yang digunakan meliputi preprocessing teks, pengubahan kata menjadi sinonim, serta perhitungan *cosine similarity* berdasarkan ID sinonim yang diambil dari *database*. Hasil akhir yang dihasilkan adalah presentase kemiripan antara jawaban siswa dengan kunci jawaban menggunakan metode *cosine similarity* yang dibantu menangani variasi kata yang berbeda namun satu makna menggunakan proses basis data sinonim kata.

**Kata Kunci**—Cerdas Cermat;Sinonim;Makna Kalimat;*Cosine Similarity*

**Abstract**— This research aims to develop a 'Cerdas Cermat' application that can evaluate student answers by measuring the similarity between student answers and answer keys using the cosine similarity method and a synonym database. The method used includes text preprocessing, converting words to synonyms, and calculating cosine similarity based on synonym IDs retrieved from the database. The final result is the percentage of similarity between student answers and answer keys using the cosine similarity method, assisted by handling different word variations with the same meaning using the synonym database process.

**Keywords**—Cerdas Cermat;Synonym;Sentence Meaning;Cosine Similarity

This is an open access article under the CC BY-SA License.



---

## Penulis Korespondensi:

Nama Penulis, Johan Rizky Triosaputra  
Teknik Informatika,  
Universitas Nusantara PGRI Kediri,  
Email: [johanrizkysa@gmail.com](mailto:johanrizkysa@gmail.com)  
ID Orcid: [<https://orcid.org/register>]  
Handphone: 082274660262

---

## I. PENDAHULUAN

Di era digital ini, teknologi informasi memainkan peran krusial dalam mendukung sektor pendidikan. Salah satu teknologi yang sangat berguna untuk pengembangan sistem pendidikan adalah Pemrosesan Bahasa Alami (*NLP*) [1], yang memungkinkan interaksi antara manusia dan komputer untuk memahami dan mengolah bahasa manusia. Dalam konteks pendidikan, teknologi *NLP* yang dapat memberikan manfaat besar adalah teknologi *Speech To Text* [2], yang mengonversi ucapan menjadi teks tertulis. Teknologi ini memudahkan siswa

dalam mengakses, merekam informasi, serta memahami dan memanfaatkan materi pembelajaran [3].

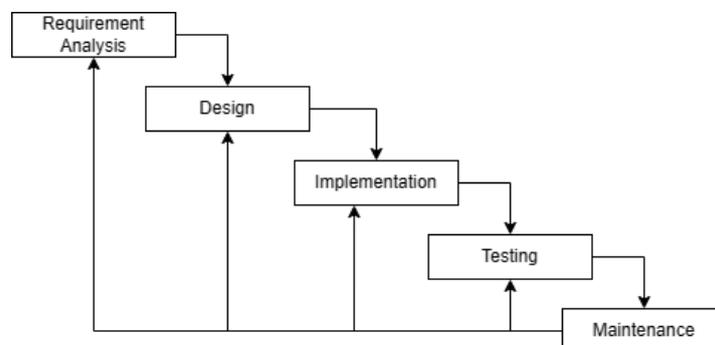
Di tingkat sekolah dasar, kompetisi cerdas cermat merupakan metode yang efektif untuk menguji pengetahuan siswa di berbagai mata pelajaran [4]. Namun, penilaian dalam kompetisi cerdas cermat sering kali dilakukan secara manual, yang dapat mengurangi pengalaman menarik bagi siswa dan menambah beban bagi guru dalam memberikan nilai secara cepat. Oleh karena itu, integrasi teknologi *Speech To Text* dalam aplikasi cerdas cermat dengan penilaian otomatis menjadi sangat penting.

Aplikasi cerdas cermat membutuhkan sistem yang mampu mengukur kemiripan jawaban siswa dengan kunci jawaban. Berbagai metode untuk mengukur kemiripan kalimat telah banyak ditemukan dalam jurnal. Misalnya, Caterina dkk. mengukur kemiripan makna kalimat menggunakan kata kerja, kata benda, dan pendekatan path, dengan tingkat kemiripan yang dicapai sebesar 0,875 [5]. Abriana dan Yaqin mengukur *Sentence Similarity* dengan membandingkan kemiripan antar kata dalam kalimat dan mengakumulasikan matriks nilai *word similarity* menggunakan bobot kriteria, menghasilkan nilai *Sentence Similarity* [6]. Amalia dkk. mengembangkan sistem essay online menggunakan metode *Cosine Similarity* untuk mengurangi waktu koreksi jawaban siswa, dengan rata-rata akurasi sebesar 81% [7]. Alternatif lain adalah menggunakan *database* sinonim kata, seperti yang dilakukan oleh Sanjaya dkk., yang mengelompokkan kata berdasarkan sinonim dan memberikan label ID unik pada setiap kelompok kata, menghasilkan nilai kemiripan sebesar 94% dari 25 pengujian [8].

## II. METODE

### A. Metode Penelitian

Metode penelitian yang digunakan adalah metode *Waterfall*. Metode *Waterfall* mencakup lima tahap utama yaitu analisis kebutuhan, desain sistem, implementasi, Evaluasi, dan pemeliharaan [9].



Gambar 1. Tahapan Metode *Waterfall*

## 1. Analisa Kebutuhan

Pada tahap ini, dilakukan analisis kebutuhan untuk membangun sistem yang diinginkan dengan mempelajari jurnal dan artikel terkait, serta mengumpulkan data melalui wawancara dengan ahli. Tujuannya adalah untuk memahami kebutuhan pengguna dan spesifikasi sistem yang harus dipenuhi [10].

- a. Data yang dibutuhkan adalah kumpulan soal dan kunci jawaban, jawaban siswa, data pengguna (siswa, guru, admin), dan sinonim kata.
- b. Jenis data adalah teks (soal, jawaban, sinonim) dan numerik (nilai kemiripan).
- c. Variabel data adalah pertanyaan, kunci jawaban, jawaban siswa, ID pengguna, dan ID sinonim.
- d. Tujuan pengumpulan data: menyusun soal dan kunci jawaban yang valid, mengumpulkan jawaban siswa untuk penilaian, mengidentifikasi sinonim untuk perhitungan kemiripan, dan mengelola data pengguna untuk akses aplikasi.

## 2. Desain

Pada tahap desain, dibuat seluruh desain sistem yang mencakup perancangan *DFD*, *flowchart*, *use case diagram*, proses *input* dan *output*, serta desain *database*. Desain ini mencakup struktur data, alur proses, dan antarmuka pengguna yang akan digunakan dalam aplikasi.

Input Kalimat melakukan preprocessing sebagai berikut:

1. *Case folding* : Proses ini mengubah semua huruf kapital dalam teks menjadi huruf kecil.
2. *Tokenization* : Tahap ini memecah teks menjadi unit-unit kecil yang disebut token.
3. *Filtering* : *Filtering* adalah tahap penyaringan di mana kata-kata atau kalimat.
4. *Stemming* : Proses ini menghilangkan sufiks, prefiks, dan konfiks dari kata dalam kalimat.
5. Fungsi terbilang : Proses ini mengubah angka dalam kalimat menjadi bentuk teks, terutama berguna dalam konteks penjualan di mana angka sering muncul [11].

## 3. Implementasi

Tahap ini melibatkan pengkodean seluruh sistem menjadi sebuah aplikasi berbasis web menggunakan bahasa pemrograman *PHP*. Pada tahap ini, semua rancangan yang telah dibuat diimplementasikan dalam bentuk kode program yang dapat dijalankan.

## 4. Evaluasi

Pada tahap evaluasi, dilakukan pengujian untuk memastikan apakah hasil keluaran dari sistem sudah sesuai dengan perhitungan atau terdapat kesalahan yang perlu diperbaiki. Pengujian dilakukan dengan berbagai skenario untuk memastikan aplikasi berfungsi sesuai yang diharapkan.

B. *Cosine Similarity*

Pengukuran tingkat kalimat dengan *Cosine Similarity* dilakukan dengan mengukur sudut antara dua vektor kalimat. Jika antar dua kalimat tersebut identik, maka sudutnya adalah 0 derajat dengan *Cosine Similarity* sama dengan 1. Sedangkan jika kalimat tersebut berbeda sepenuhnya maka sudutnya adalah 90 derajat dengan *Cosine Similarity* sama dengan 0 [12].

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

**III. HASIL DAN PEMBAHASAN**

A. Pengujian dan Hasil

Pada tabel 1 adalah penilaian guru untuk setiap sesinya dengan soal berjumlah 15, jika jawaban salah guru akan memberikan S sedangkan jika jawaban benar guru akan memberikan B.

Tabel 1. Hasil Penilaian Guru

Soal Ke-	Sesi									
	Ke-1	Ke-2	Ke-3	Ke-4	Ke-5	Ke-6	Ke-7	Ke-8	Ke-9	Ke-10
1	B	B	B	B	S	B	B	B	S	B
2	S	B	S	B	B	B	B	B	B	B
3	B	S	B	S	B	B	S	B	S	B
4	S	S	S	S	B	B	B	B	B	S
5	S	B	B	S	B	B	B	B	B	B
6	B	B	B	B	B	B	B	B	B	B
7	B	B	S	S	S	S	S	S	S	S
8	B	B	S	B	B	B	B	B	B	B
9	B	B	B	B	B	B	B	B	B	B
10	B	B	B	B	B	B	B	S	B	B
11	B	B	S	B	S	B	B	S	B	B
12	B	B	B	B	B	B	B	B	B	B
13	S	B	B	B	B	B	B	B	B	B
14	B	B	S	B	B	B	B	B	B	B
15	B	B	B	B	B	B	B	B	B	B

Pada tabel 2 merupakan hasil dari penilaian sistem. Sistem menilai berdasarkan sinonim yang ada pada basis data lalu menghitung dengan *cosine similarity*.

Tabel 2. Hasil Penilaian Sistem

Soal Ke-	Sesi									
	Ke-1	Ke-2	Ke-3	Ke-4	Ke-5	Ke-6	Ke-7	Ke-8	Ke-9	Ke-10
1	100	100	100	100	70,7	100	100	100	70,7	100
2	100	100	0	100	100	100	100	100	100	100
3	70,7	0	0	70,7	70,7	100	70,7	100	100	100

4	0	0	0	0	0	100	70,7	70,7	100	70,7
5	0	0	0	0	0	100	0	0	0	100
6	100	0	0	0	100	100	100	100	100	100
7	100	100	0	0	0	0	100	0	0	100
8	70,7	70,7	0	70,7	70,7	70,7	70,7	70,7	70,7	70,7
9	100	100	100	100	0	100	100	100	100	100
10	0	0	100	0	0	100	100	0	0	0
11	70,7	100	0	70,7	0	100	0	0	70,7	0
12	0	0	0	0	0	100	100	0	0	100
13	0	0	0	0	0	0	0	100	0	0
14	0	0	0	0	0	0	0	100	0	0
15	100	100	100	100	100	100	86,6	86,6	86,6	100
1	100	100	100	100	70,7	100	100	100	70,7	100

Karena perbedaan penilaian guru dengan sistem maka diperlukan menentukan *threshold* pada sistem. Seperti pada tabel 3, menggunakan *threshold* 70 sehingga nilai sistem yang diatas 70 akan masuk ke jawaban benar (B) sedangkan dibawah nilai sistem 70 akan masuk ke jawaban salah (S). Penentuan *threshold* pada tabel 3.

$$f(x) = \begin{cases} \text{jika } x < \theta, \text{ maka } 0 \\ \text{jika } x \geq \theta, \text{ maka } 1 \end{cases} \quad (2)$$

Keterangan :

$\theta = 70 =$   
*threshold*

$x =$  hasil  
 penilaian sistem

Tabel 3. Penggunaan *threshold* pada sistem

Soal	Sesi									
	Ke-1	Ke-2	Ke-3	Ke-4	Ke-5	Ke-6	Ke-7	Ke-8	Ke-9	Ke-10
1	B	B	B	B	B	B	B	B	B	B
2	S	B	S	B	B	B	B	B	B	B
3	B	S	S	B	B	B	B	B	B	B
4	S	S	S	S	B	B	S	B	B	B
5	S	S	S	S	B	S	S	S	S	B
6	B	S	S	S	B	B	B	B	B	B
7	B	B	S	S	S	B	S	B	B	B
8	B	B	B	B	B	B	S	B	B	B
9	B	B	B	B	S	B	B	B	B	B
10	S	S	B	S	S	B	B	S	S	B
11	B	B	S	B	S	B	S	B	B	B
12	B	S	S	S	S	B	B	S	B	B

13	S	S	S	S	S	S	S	B	B	B
14	S	S	S	S	S	S	S	B	B	B
15	B	B	B	B	B	B	B	B	B	B

Menentukan threshold menggunakan rumus akurasi yang membagi jumlah jawaban benar keseluruhan dengan semua jawaban keseluruhan (benar dan salah).

$$Akurasi = \frac{\text{jumlah klasifikasi benar}}{\text{jumlah semua data}} \dots\dots\dots(3)$$

Tabel 4 merupakan hasil pengujian nilai *threshold* dengan menguji nilai *threshold* 10 hingga 100 mendapatkan hasil yaitu sebagai berikut:

Tabel 4. Pengujian Nilai *Threshold*

Threshold	Sama	Tidak Sama
10	67,33	32,66
20	67,33	32,66
30	67,33	32,66
40	67,33	32,66
50	67,33	32,66
60	67,33	32,66
70	67,33	32,66
80	60	40
90	58,00	42,00
100	58,00	42,00

#### B. Analisis Hasil

Berdasarkan hasil pengujian di atas, dapat dilihat bahwa penggunaan *database* sinonim kata dengan metode *Cosine Similarity* terlihat jika siswa tersebut memasukkan satu kata jawaban sesuai dengan jawaban yang ada ID sinonimnya maka akan mendapatkan nilai 100, sedangkan jika memasukkan satu kata yang tidak ada ID sinonimnya maka jawaban siswa akan mendapatkan 0.

### IV. KESIMPULAN

#### A. Kesimpulan

Penggunaan *database* sinonim kata dengan metode *Cosine Similarity* secara efektif dapat mengidentifikasi kemiripan jawaban siswa dengan kunci jawaban. Siswa yang menjawab dan sinonimnya tersedia di *database* akan mendapatkan nilai kemiripan yang tinggi (100%). Ketika jawaban siswa tidak memiliki ID sinonim yang sesuai dalam *database*, nilai kemiripan yang dihasilkan adalah 0. Hal ini menunjukkan bahwa kesesuaian metode ini sangat bergantung pada kelengkapan dan akurasi *database* sinonim yang digunakan. Jawaban yang tidak tepat atau

penggunaan kata-kata yang berbeda namun tidak memiliki sinonim yang terdaftar dalam *database* menyebabkan skor kemiripan yang rendah. Ini menekankan pentingnya memperluas *database* sinonim untuk mencakup lebih banyak variasi kata yang mungkin digunakan oleh siswa.

#### B. Saran

Saran yang dapat disampaikan adalah menambahkan daftar sinonim yang lebih banyak agar jawaban siswa banyak yang sesuai dengan kunci jawaban. Pengujian lebih lanjut juga dapat membantu mengidentifikasi area yang memerlukan perbaikan seperti memasukkan kata yang sama tetapi tidak dianggap benar karena tidak masuk sinonim. Menggabungkan metode *Cosine Similarity* dengan algoritma lain yang mampu menangani variasi bahasa yang lebih kompleks dapat meningkatkan akurasi penilaian.

### DAFTAR PUSTAKA

- [1] F. Rumaisa, Y. Puspitarani, A. Rosita, A. Zakiah, and S. Violina, "Penerapan Natural Language Processing (NLP) Di Bidang Pendidikan," *Jurnal Inovasi Masyarakat*, vol. I, no. 03, pp. 232–235, Sep. 2021, doi: 10.33197/jim.vol1.iss3.2021.799.
- [2] K. Nugroho, "IMPLEMENTASI SISTEM SPEECH TO TEXT BERBASIS ANDROID MENGGUNAKAN APP INVENTOR SPEECH RECOGNIZER," *jurnal ilmiah INFOKAM*, Semarang, Mar. 2019. doi: <https://doi.org/10.53845/infokam.v15i1.167>.
- [3] T. Hidayat and A. Mulyoto, "Perancangan Aplikasi Pembelajaran Siswa Berbasis Web Menggunakan Speech To Text Pada Sdn 2 Pabuaran," *OKTAL : Jurnal Ilmu Komputer dan Science*, vol. 1, no. 10, pp. 1579–1585, Jan. 2022, [Online]. Available: <https://journal.mediapublikasi.id/index.php/oktal>
- [4] A. S. Nasution and Nurdailah, "MEMBANGKITKAN MINAT BELAJAR SISWA MELALUI LOMBA CERDAS CERMAT," *Prosiding Seminar Nasional Hasil Pengabdian*, pp. 10–13, 2018.
- [5] Y. Caterina, M. A. Yaqin, and S. Zaman, "Pengukuran Kemiripan Makna Kalimat dalam Bahasa Indonesia Menggunakan Metode Path," *Fountain of Informatics Journal*, vol. 6, no. 2, p. 45, Mar. 2021, doi: 10.21111/fij.v6i2.4844.
- [6] G. U. Abriana and M. A. Yaqin, "Analisis Implementasi Metode Semantic Similarity untuk Pengukuran Kemiripan Makna antar Kalimat," *ILKOMNIKA: Journal of Computer Science and Applied Informatics*, vol. 1, no. 2, pp. 47–57, Dec. 2019, doi: 10.28926/ilkomnika.v1i2.15.
- [7] E. L. Amalia, A. J. Jumadi, I. A. Mashudi, and D. W. Wibowo, "ANALISIS METODE COSINE SIMILARITY PADA APLIKASI UJIAN ONLINE ESAI OTOMATIS ( STUDI

- KASUS JTI POLINEMA ),” *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, vol. 8, no. 2, pp. 343–348, Apr. 2019, doi: 10.25126/jtiik.202184356.
- [8] A. Sanjaya, A. Bagus Setiawan, U. Mahdiyah, I. Nur Farida, A. Risky Prasetyo, and U. Nusantara PGRI Kediri, “PENGUKURAN KEMIRIPAN MAKNA MENGGUNAKAN COSINE SIMILARITY DAN BASIS DATA SINONIM KATA MEASUREMENT OF MEANING SIMILARITY USING COSINE SIMILARITY AND WORD SYNONYMS DATABASE,” vol. 10, no. 4, 2023, doi: 10.25126/jtiik.2023106864.
- [9] A.-A. Solehudin, N. Wahyu, N. Fariz, R. F. Permana, and A. Saifudin, “Rancang Bangun Digitalisasi Persediaan Barang Berbasis Web Menggunakan Metode Waterfall,” 2023. [Online]. Available: <https://journal.mediapublikasi.id/index.php/logic>
- [10] Shylesh S, “A Study of Software Development Life Cycle Process Models,” *International Journal of Advanced Research in Computer Science*, Pakistan, Jan. 2017. doi: <https://doi.org/10.26483/ijarcs.v8i1.2844>.
- [11] A. Sanjaya and S. D. Sasongko, “UJI KEMIRIPAN KALIMAT MENGGUNAKAN FUNGSI TERBILANG PADA PRE-PROCESSING DAN COSINE SIMILARITY DALAM BAHASA INDONESIA,” *jurnal ilmiah NERO*, Kediri, 2022. doi: 10.21107/nero.v7i2.321.
- [12] A. Bastian, H. Sujadi, and P. A. Sukmana, “RANCANG BANGUN APLIKASI PENILAIAN UJIAN ESSAY DENGAN MENGGUNAKAN ALGORITMA NAZIEF & ANDRIANI DAN METODE COSINE SIMILARITY,” *INFOTECH journal*, vol. 4, no. 2, pp. 62–68, 2018, doi: DOI: 10.31949/inf.v4i2.1168.