Jurnal Ilmiah
Teknologi Sistem
Informasi

Jitsi

Vol. 5 (2024) No. 4

https://jurnal-itsi.org/index.php/jitsi; E-ISSN: 2722-4600; ISSN: 2722-4619
DOI: 10.62527/jitsi.5.4.291

Similarity Judul Tugas Akhir Menggunakan Metoda Cosine, Jaccard, Rabin-Karp Pada Jurusan TI

Roni Putra [#], Sumema [#], Zalna Mustika [#]

[#] *Jurusan Teknologi Informasi, Politeknik Negeri Padang, Limau Manis, Padang, 25154, Indonesia*
E-mail: roni_putra[at]pnp.ac.id, sumema[at]pnp.ac.id, zalnamustika111[at]gmail.com

A B S T R A C T S

Plagiarism can occur anywhere and can be committed by anyone. For example, plagiarism by students when creating college assignments, where they unintentionally use someone else's writing or ideas and forget to cite the source. Based on the above explanation, it is necessary to develop an information system to assess the percentage of plagiarism for proposed titles. After conducting an implementation test to measure the similarity accuracy and processing speed, a comparison was made between the proposed titles and the existing data, totaling 940 titles. The results showed a similarity rate of 84.13% with the Rabin-Karp method, 61.24% with the Cosine method, and 42.86% with the Jaccard method. In terms of processing speed, the Rabin-Karp method ranked first with a time of 0.5023 seconds, followed by the Cosine method with 0.5095 seconds, and the Jaccard method with 0.5103 seconds. From the test results, Jaccard Similarity is more suitable for short texts with unique word sets, but it is less precise for longer texts with important word distributions. Meanwhile, the Rabin-Karp method is not designed to measure general text similarity; instead, it is intended for fast and precise substring searches in pattern matching contexts

Manuscript received Oct 23, 2024; revised Nov 21, 2024 accepted Nov 22 2024 Date of publication Dec 30, 2024

A B S T R A K

Tindakan plagiat dapat terjadi di mana saja dan dilakukan oleh siapa saja. Misalnya, tindakan plagiat yang dilakukan oleh mahasiswa saat membuat tugas perkuliahan, di mana secara tidak sengaja mereka mengambil tulisan atau gagasan milik orang lain dan lupa mencantumkan sumbernya. Berdasarkan uraian di atas, perlu dibangun suatu sistem informasi untuk melihat persentase plagiarisme terhadap judul yang akan diusulkan. Setelah melakukan uji implementasi untuk mengukur tingkat akurasi kemiripan dan kecepatan saat dijalankan, dilakukan perbandingan antara judul usulan dengan data yang sudah ada, yaitu sebanyak 940 judul. Hasilnya menunjukkan kemiripan sebesar 84,13% dengan metode Rabin-Karp, 61,24% dengan metode Cosine, dan 42,86% dengan metode Jaccard. Sedangkan untuk kecepatan proses, metode Rabin-Karp berada di urutan pertama dengan waktu 0,5023 detik, diikuti oleh metode Cosine dengan waktu 0,5095 detik, dan metode Jaccard dengan waktu 0,5103 detik. Dari hasil pengujian, Jaccard Similarity lebih cocok untuk teks pendek dengan kumpulan kata unik, tetapi kurang presisi jika diterapkan pada teks panjang dengan distribusi kata penting. Sementara itu, metode Rabin-Karp bukan untuk mengukur kesamaan teks secara umum, melainkan untuk pencarian substring yang cepat dan presisi dalam konteks pencocokan pola.

Keywords / Kata Kunci — ***Similarity; Sistem Informasi, Cosine; Jaccard; Rabin-Karp***

CORRESPONDING AUTHOR

Roni Putra
Jurusan Teknologi Informasi, Politeknik Negeri Padang, Limau Manis, Padang, 25154, Indonesia Selatan 12450, Indonesia
Email: roni_putra[at]pnp.ac.id

1. PENDAHULUAN

Semakin pesatnya perkembangan teknologi informasi dan komunikasi memberikan kemudahan dalam mengakses dan mencari informasi melalui internet. perkembangan teknologi informasi tidak hanya memberikan dampak positif, tetapi juga memberikan dampak negatif. Seperti mudahnya seseorang dalam melakukan penjiplakan terhadap karya orang lain.

Penjiplakan atau disebut juga dengan plagiarisme adalah tindakan mengambil karya orang lain, baik berupa gagasan, karangan pendapat, dan ide milik orang lain kemudian menjadikanya seolah-olah milik sendiri [1]. Tindakan plagiat bisa terjadi dimana saja dan dilakukan oleh siapa saja. Seperti tindakan plagiat yang dilakukan oleh mahasiswa saat membuat sebuah tugas perkuliahan yang tidak sengaja mengambil tulisan atau gagasan milik orang lain dan lupa untuk memberikan atau mencantumkan sumbernya. Perbuatan ini tidak hanya bisa terjadi saat pembuatan tugas kuliah, tetapi juga bisa terjadi saat pembuatan proyek akhir [2]

Untuk mengatasi tindakan Penjiplakan terhadap judul tugas akhir sudah ada maka di buatlah sebuah sistem informasi yang dapat menentukan persentase similarity dengan membandingkan judul usulan dengan judul yang sudah ada, metoda yang digunakan untuk menentukan similarity terdiri dari Cosine, Jaccard dan Rabin-Karp

2. METODOLOGI PENELITIAN

Dalam membuat sistem untuk menentukan kesamaan dari judul tugas akhir yang di usulkan dengan data yang telah ada menggunakan 3 metoda:

a Cosine Similarity

Cosine adalah salah satu metode paling populer untuk mengukur kesamaan antara dua vektor dalam ruang multidimensi. Metode ini banyak digunakan dalam bidang Text Mining, Natural Language Processing (NLP), dan Information Retrieval [2] [10]

Rumus Cosine Similarity antara dua vektor AAA dan BBB adalah sebagai berikut:

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

Dimana:

- A – B (hasil kali saklar) antara dua vektor A dan B.
- ‖A‖ adalah magnitude (panjang) dari vektor A
- ‖B‖ adalah magnitude (panjang) dari vektor B

b Jaccard Similarity

Jaccard Similarity antara dua himpunan A dan B didefinisikan sebagai rasio ukuran **intersection** (irisan) dari dua himpunan terhadap ukuran **union** (gabungan) dari kedua himpunan tersebut. [4][6]

Secara matematis, rumus Jaccard Similarity adalah:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \quad (2)$$

Dimana:

- A∩B adalah jumlah elemen yang berada di **irisan** antara himpunan A dan B.
- A∪B adalah jumlah elemen yang berada di **gabungan** himpunan A dan B
- Nilai **Jaccard Similarity** berkisar antara 0 dan 1:
  0 artinya tidak ada elemen yang sama antara dua himpunan (tidak mirip sama sekali) dan **1** artinya kedua himpunan memiliki elemen yang identik (sangat mirip atau sama)

c Rabin-Karp Similarity

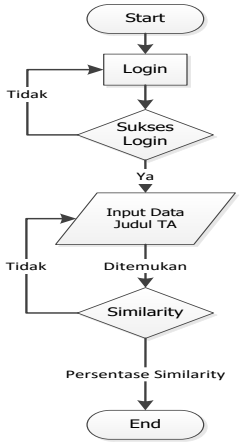
Rabin-Karp Similarity adalah pengembangan dari algoritma **Rabin-Karp**, yang pada dasarnya adalah algoritma pencarian string berbasis **hashing**. Namun, dalam konteks kemiripan (similarity), Rabin-Karp bisa digunakan untuk mengukur **similarity** atau kemiripan antara dua teks dengan cara memanfaatkan hashing untuk membandingkan substring atau potongan-potongan dari kedua teks [11][12]

Gambar 1 merupakan *flowchart* penggunaan sistem informasi similarity di jurusan Teknologi Informasi. *flowchart* ini menggambarkan alur proses kerja sebuah sistem untuk memeriksa *similarity* (kesamaan) pada judul tugas akhir (TA). Berikut adalah penjelasan rinci tiap langkahnya:

1. **Start**
2. **Login**
   Pengguna harus melakukan login ke sistem.
   - Jika login **tidak berhasil** (decision: "Tidak"), pengguna akan kembali ke langkah awal untuk mencoba login ulang.
   - Jika login berhasil (decision: "Ya"), proses akan dilanjutkan ke langkah berikutnya.
3. Input **Data Judul TA**
   Pengguna memasukkan data berupa judul tugas akhir (TA) yang ingin diperiksa.

- Jika data **tidak ditemukan** dalam sistem (decision: "Tidak"), pengguna diarahkan untuk menginput ulang data.
- Jika data **ditemukan** (decision: "Ya"), sistem melanjutkan ke proses berikutnya.

4. **Similarity**
   Sistem akan memproses data yang telah dimasukkan untuk memeriksa tingkat kesamaan (*similarity*) dengan data yang sudah ada di dalam database.
5. **Persentase Similarity**
   Sistem menghasilkan hasil berupa persentase tingkat kesamaan antara judul TA yang diinputkan dengan data di dalam sistem.
6. **End**
   Proses selesai, dan pengguna mendapatkan hasil persentase similarity



GAMBAR 1. Flowchart Sistem

Coding Program

```
<?php
function getTerms($string)
{
    // Ubah string ke huruf kecil dan hilangkan karakter non-alfabet
    $string = strtolower(preg_replace('/[^a-zA-Z0-9\s]/', '', $string));

    // Pisahkan kata-kata dan buat frekuensi kata
    $words = explode(' ', $string);
    $terms = array();

    foreach ($words as $word) {
        if (isset($terms[$word])) {
            $terms[$word]++;
        } else {
            $terms[$word] = 1;
        }
    }

    return $terms;
}

function cosineSimilarity($vectorA, $vectorB)
{
    $dotProduct = 0;
    $magnitudeA = 0;
    $magnitudeB = 0;

    // Hitung dot product dan magnitudo dari vektor A dan B
    foreach ($vectorA as $key => $value) {
        if (isset($vectorB[$key])) {
            $dotProduct += $value * $vectorB[$key];
        }
        $magnitudeA += pow($value, 2);
    }

    foreach ($vectorB as $key => $value) {
        $magnitudeB += pow($value, 2);
    }

    // Cosine similarity formula
    $magnitudeA = sqrt($magnitudeA);
    $magnitudeB = sqrt($magnitudeB);

    if ($magnitudeA * $magnitudeB == 0) {
        return 0; // Jika salah satu magnitudo adalah 0, return 0
    } else {
        return $dotProduct / ($magnitudeA * $magnitudeB);
    }
}

function similarityPercentage($similarityScore)
```

```
{
    // Ubah nilai cosine similarity (0 - 1) menjadi persentase
    return $similarityScore * 100;
}
function getUniqueTerms($string)
{
    // Ubah string ke huruf kecil dan hilangkan karakter non-alfabet
    $string = strtolower(preg_replace('/[^a-zA-Z0-9\s]/', '', $string));

    // Pisahkan kata-kata dan buat array dari kata unik
    $words = explode(' ', $string);
    return array_unique($words);
}

function jaccardSimilarity($setA, $setB)
{
    // Irisan (Intersection) antara dua himpunan
    $intersection = array_intersect($setA, $setB);

    // Gabungan (Union) antara dua himpunan
    $union = array_unique(array_merge($setA, $setB));

    // Jaccard similarity formula
    $similarity = count($intersection) / count($union);

    return $similarity;
}
function rabinKarp($text, $pattern, $prime = 101) {
    $m = strlen($pattern); // Panjang pola
    $n = strlen($text); // Panjang teks
    $d = 256; // Jumlah karakter dalam alfabet (ASCII)
    $p = 0; // Nilai hash untuk pola
    $t = 0; // Nilai hash untuk substring teks
    $h = 1; // Nilai h
    $occurrences = 0; // Menghitung kemunculan pola

    // h akan bernilai "pow(d, m-1) % prime"
    for ($i = 0; $i < $m - 1; $i++) {
        $h = ($h * $d) % $prime;
    }

    // Hitung nilai hash awal untuk pola dan substring pertama dari teks
    for ($i = 0; $i < $m; $i++) {
        $p = ($d * $p + ord($pattern[$i])) % $prime;
        $t = ($d * $t + ord($text[$i])) % $prime;
    }

    // Geser pola di atas teks satu per satu
    for ($i = 0; $i <= $n - $m; $i++) {
        // Cek nilai hash pola dan substring teks
        if ($p == $t) {
            // Jika nilai hash sama, periksa karakter satu per satu
            for ($j = 0; $j < $m; $j++) {
                if ($text[$i + $j] != $pattern[$j]) {
                    break;
                }
            }

            // Jika pola ditemukan
            if ($j == $m) {
                $occurrences++; // Tambah jumlah kemunculan pola
            }
        }

        // Hitung nilai hash untuk substring teks berikutnya
        if ($i < $n - $m) {
            $t = ($d * ($t - ord($text[$i]) * $h) + ord($text[$i + $m])) % $prime;

            // Jika nilai hash negatif, buat menjadi positif
            if ($t < 0) {
                $t = ($t + $prime);
            }
        }
    }
```

```
    return $occurrences;
}

function calculateSimilarity($text1, $text2) {
    $totalSubstrings = 0;
    $totalMatches = 0;
    $substringLength = 3; // Menggunakan panjang substring 3 untuk membandingkan

    // Ambil semua substring dari text1
    for ($i = 0; $i <= strlen($text1) - $substringLength; $i++) {
        $pattern = substr($text1, $i, $substringLength);
        $matches = rabinKarp($text2, $pattern);

        $totalMatches += min($matches, 1); // Hitung hanya 1 kemiripan per substring
        $totalSubstrings++;
    }

    // Hitung persentase kemiripan
    $similarityPercentage = ($totalMatches / $totalSubstrings) * 100;

    return round($similarityPercentage, 2);
}
```

3. HASIL DAN PEMBAHASAN

3.1. Implementasi

Pengujian persentase similarity serta kecepatan dalam menampilkan data Pengujian ini membandingkan judul tugas akhir sebanyak 940 dengan judul akan di usulkan.

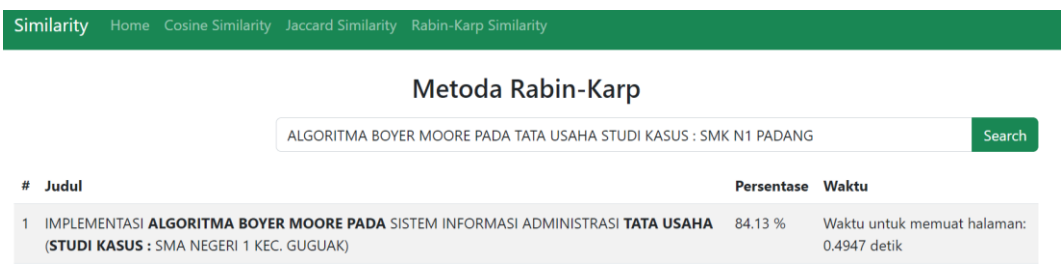
TABEL 1. Pengujian

Pengujian	Metoda		
	Cosine	Jaccard	Rabin-Karp
Persentase Kemiripan (%)	61.24 %	42.86 %	84.13 %
Kecepatan (Detik)	0.5095	0.5103	0.5023

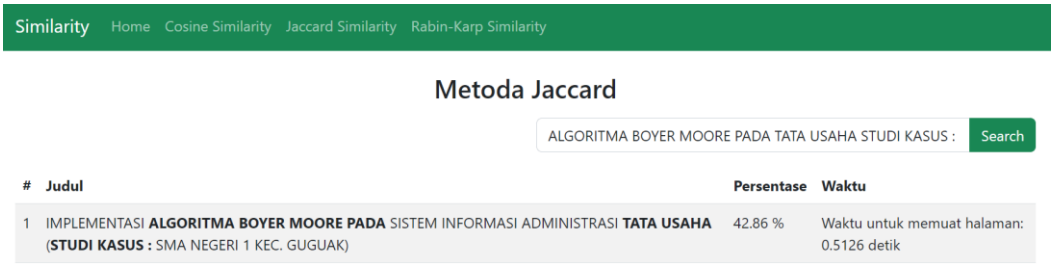
Hasilnya menunjukkan kemiripan sebesar 84,13% dengan metode Rabin-Karp, 61,24% dengan metode Cosine, dan 42,86% dengan metode Jaccard. Sedangkan untuk kecepatan proses, metode Rabin-Karp berada di urutan pertama dengan waktu 0,5023 detik, diikuti oleh metode Cosine dengan waktu 0,5095 detik, dan metode Jaccard dengan waktu 0,5103 detik.

Dari hasil pengujian, Jaccard Similarity lebih cocok untuk teks pendek dengan kumpulan kata unik, tetapi kurang presisi jika diterapkan pada teks panjang dengan distribusi kata penting. Sementara itu, metode Rabin-Karp bukan untuk mengukur kesamaan teks secara umum, melainkan untuk pencarian substring yang cepat dan presisi dalam konteks pencocokan pola.

GAMBAR 1. Pengujian Metoda Cosine



GAMBAR 2. Pengujian Metoda Rabin-Karp



GAMBAR 3. Pengujian Metoda Jaccard

4. KESIMPULAN

Hasil pengujian dengan membandingkan judul usulan dengan data yang sudah ada sebanyak 940 judul dapat di simpulkan persentase yang didapatkan dengan 3 metoda yang di gunakan paling tinggi 84.13 % pada metoda Rabin-Karp kemudian di urutan ke dua 61.24 % menggunakan metoda Cosine dan yang terakhir 42.86 % pada metoda Jaccard, sedangkan dari segi kecepatan proses similarity pada urutan pertama 0.5023 detik menggunakan Rabin-Karp di lanjutkan 0.5095 detik menggunakan metoda Cosine dan paling akhir 0.5103 detik pada metoda Jaccard. Cosine Similarity menawarkan presisi tertinggi untuk dokumen panjang karena mempertimbangkan frekuensi kata dan dapat menangkap nuansa dalam distribusi kata, Jaccard Similarity lebih cocok untuk teks pendek dengan set kata unik, tetapi kurang presisi jika teks panjang dan distribusi kata penting dan Rabin-Karp bukan untuk mengukur kesamaan teks secara umum, melainkan untuk pencarian substring yang cepat dan presisi dalam konteks pencocokan pola

REFERENSI

[1] KBBI, "Kamus Besar Bahasa Indonesia (KBBI)," 2020. https://kbbi.web.id/plagiat (accessed Dec. 21, 2020).

[2] M. Z. Naf'an, A. Burhanuddin, and A. Riyani, "Penerapan Cosine Similarity dan Pembobotan TF-IDF untuk Mendeteksi Kemiripan Dokumen," J. Linguist. Komputasional, vol. 2, no. 1, pp. 23–27, 2019, doi: 10.26418/jlk.v2i1.17

[3] T. W. Yit et al., "INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION journal homepage : www.joiv.org/index.php/joiv INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION Transformer in mRNA Degradation Prediction." [Online]. Available: www.joiv.org/index.php/joiv

[4] R. A. Wibowo, D. Nugroho, and B. Widada, "Penggunaan Metode Cosine Similarity Pada Sistem Pengelompokan Kerja Praktek, Tugas Akhir Dan Skripsi," J. TIKomSiN, vol. 5, no. 1, pp. 32–38, 2017.

[5] O. Nurdiana, J. Jumadi, and D. Nursantika, "Perbandingan Metode Cosine Similarity Dengan Metode Jaccard Similarity Pada Aplikasi Pencarian Terjemah Al-Qur'an Dalam Bahasa Indonesia," J. Online Inform., vol. 1, no. 1, p. 59, 2016, doi: 10.15575/join.v1i1.12.

[6] I. S. Pratama, Nawassyarif, and J. Aliyah, "Pengembangan sistem informasi sarana dan prasarana Di Universitas Tetknologi Sumbawa (UTS) berbasis web," J. Inform. Teknol. Sains, vol. 1, no. 1, pp. 39–49, 2019.

[7] S. S. S. Purwandari, Rancang Bangun Search Engine Tafsir Al-Quran Yang Mampu Memproses Teks Bahasa Indonesia Menggunakan Metode Jaccard Similarity, Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang, 2012, pp. 9-27

[8] M. Z. Naf'an, A. Burhanuddin, and A. Riyani, "Penerapan Cosine Similarity dan Pembobotan TF-IDF untuk Mendeteksi Kemiripan Dokumen," J. Linguist. Komputasional, vol. 2, no. 1, pp. 23–27, 2019, doi: 10.26418/jlk.v2i1.17

[9] M. Shafira, H. Amnur, and R. Afyenni, "Load Balancing Menggunakan Algoritma Round Robin Dengan Stickness Pada AWS", jitsi, vol. 2, no. 4, pp. 116 - 123, Dec. 2021.

[10] R. T. Wahyuni, D. Prastiyanto, and E. Supraptono, "Penerapan Algoritma Cosine Similarity dan Pembobotan TF-IDF pada Sistem Klasifikasi Dokumen Skripsi," J. Tek. Elektro, vol. 9, no. 1, pp. 18–23, 2017, doi: 10.15294/jte.v9i1.10955.

[11] I. Abdullah and E. Aribowo, "Rancang Bangun Aplikasi Pengecekan Kemiripan Judul Skripsi Dengan Metode Cosine Similarity (Studi Kasus : Program Studi Teknik Informatika Uad)," JSTIE (Jurnal Sarj. Tek. Inform., vol. 6, no. 2, pp. 43–52, 2018, doi: 10.12928/jstie.v6i2.15241

[12] R. A. Wibowo, D. Nugroho, and B. Widada, "Penggunaan Metode Cosine Similarity Pada Sistem Pengelompokan Kerja Praktek, Tugas Akhir Dan Skripsi," J. TIKomSiN, vol. 5, no. 1, pp. 32–38, 2017.

[13] A. Putera Utama Siahaan and Sugianto, "Analisis k-gram, basis dan modulo rabin-karp sebagai penentu akurasi persentase kemiripan dokumen," SENASPRO 2017 | Seminar Nasional dan Gelar Produk, pp. 198–206, 2017

[14] M. A. Yulianto and N. Nurhasanah, "The Hybrid of Jaro-Winkler and Rabin-Karp Algorithm in Detecting Indonesian Text Similarity," Jurnal Online Informatika, vol. 6, no. 1, p. 88, Jun. 2021, doi: 10.15575/join.v6i1.640