**JIKTEKS : Jurnal Ilmu Komputer dan Teknologi Informasi**
Vol. 03, No. 01, Desember 2024, hlm. 22 - 34
ISSN : 2986-5417
DOI : doi.org/10.70404/jikteks.v3i01.125
Laman Jurnal: jurnal.faatuatua.com/index.php/JIKTEKS

# Algoritma Smith-Waterman Untuk Mengidentifikasi Kemiripan Judul Proyek Mahasiswa

**Andre Pratama[1*], Junerdi Nababan[2], Nadiyah Shofa Salsabila[3]**
[1*]Informatika, Universitas Satya Terra Bhinneka, Indonesia
[2]Bisnis Digital, Universitas Satya Terra Bhinneka, Indonesia
[3]Teknik Informatika, Universitas Mikroskil, Indonesia
e-mail: [1*]*andrepratama@satyaterrabhinneka.ac.id*
e-mail: [2]*junerdin@satyaterrabhinneka.ac.id*
e-mail: [3]*nadiyah.sho12@gmail.com*

***Keywords:***

*Plagiarism,*
*Title Similarity,*
*Smith-Waterman,*
*Local Alignment,*
*Text Detection.*

**ABSTRACT**

Plagiarism in the academic environment is a significant problem, especially in the submission of project titles by students. To identify the similarity of project titles in order to prevent plagiarism, this research uses the *Smith-Waterman* algorithm. This algorithm, which was originally used in bioinformatics, is applied to compare texts with a local alignment approach. The research process includes several stages such as data collection, text *preprocessing* with *case folding*, *stemming*, and *tokenizing* before the application of the *Smith-Waterman* algorithm. The test results show that this algorithm is effective in detecting similarities between project titles, with varying percentages of similarity. This research is expected to be the first step in developing a more accurate plagiarism detection system in the academic environment.

***Kata Kunci***

*Plagiarisme,*
*Kemiripan Judul,*
*Smith-Waterman,*
*Penyelarasan Lokal,*
*Deteksi Teks.*

**ABSTRAK**

Plagiarisme dalam lingkungan akademik menjadi masalah yang signifikan, terutama dalam pengajuan judul proyek oleh mahasiswa. Untuk mengidentifikasi kemiripan judul proyek guna mencegah plagiarisme, penelitian ini menggunakan algoritma *Smith-Waterman*. Algoritma ini, yang awalnya digunakan dalam bioinformatika, diaplikasikan untuk membandingkan teks dengan pendekatan penyelarasan lokal. Proses penelitian meliputi beberapa tahapan seperti pengumpulan data, *preprocessing* teks dengan *case folding*, *stemming*, dan *tokenizing* sebelum penerapan algoritma *Smith-Waterman*. Hasil pengujian menunjukkan bahwa algoritma ini efektif dalam mendeteksi kemiripan antara judul proyek, dengan persentase kemiripan yang bervariasi. Penelitian ini diharapkan dapat menjadi langkah awal dalam pengembangan sistem deteksi plagiarisme yang lebih akurat di lingkungan akademik.

**Korespondensi Penulis *):**

Andre Pratama
Universitas Satya Terra Bhinneka
Jl. Sunggal Gg. Bakul, Sunggal, Kec. Medan Sunggal, Kota Medan, Sumatera Utara

*Diajukan: 18-12-2024 | Diterima: 20-12-2024 | Diterbitkan: 30-12-2024*

## 1. PENDAHULUAN

Di lingkungan akademik dan dunia pendidikan, tugas merupakan hal yang lazim diberikan oleh tenaga pengajar [1]. Salah satu masalah yang sering dihadapi mahasiswa adalah rasa malas, yang menyebabkan sebagian dari mereka hanya meniru ide ataupun jawaban dari teman-temannya. Masalah plagiarisme seperti ini telah menjadi isu yang umum di dunia pendidikan [2]. Bukan hanya di tingkat nasional, tetapi juga telah menjadi perhatian di seluruh dunia. Oleh karena itu, masalah ini perlu ditangani dengan serius agar plagiarisme tidak menjadi kebiasaan yang dianggap normal.

Salah satu contoh adalah ketika mahasiswa ingin mengajukan judul proyek dalam mata kuliah tertentu. Biasanya mahasiswa sering kesulitan untuk mengidentifikasi apakah judul yang diajukan sudah ada atau belum [3]. Akibatnya, proyek yang sebenarnya sudah diajukan seringkali diajukan kembali. Hal ini juga menimbulkan kekhawatiran akan adanya proyek dengan judul yang sama, yang ternyata telah diajukan sebelumnya, sehingga dapat dianggap sebagai

tindakan plagiarisme. Untuk itu, sebagai langkah awal untuk menurunkan angka plagiarisme, khususnya di dunia pendidikan, perlu dikembangkan suatu sistem atau model yang mampu mendeteksi tindakan tersebut, karena berdasarkan aspek yang diplagiat, ide atau judul yang ditiru termasuk ke dalam kategori plagiarisme [4].

Beberapa penelitian telah dilakukan untuk mengidentifikasi kemiripan suatu kata, kalimat, ataupun dokumen. Dalam penelitian [5], untuk mendeteksi kemiripan judul skripsi, digunakan kombinasi algoritma TF-IDF dan *Fuzzy Matching*. Penelitian tersebut membuktikan bahwa algoritma *term frequency – inverse document frequency* (TF-IDF) dan *Fuzzy Matching* menunjukkan hasil yang signifikan dalam deteksi kemiripan teks. Pada penelitian lain, algoritma TF-IDF digunakan untuk melakukan pengelompokan topik skripsi pada aplikasi repository STMIK Budi Darma. Hasil penelitian tersebut juga membuktikan bahwa algoritma TF-IDF mampu melakukan klasifikasi topik skripsi [6]. Namun, algoritma TF-IDF yang digunakan pada penelitian [5][6] memiliki kelemahan, yaitu algoritma tersebut hanya menghitung frekuensi kata tanpa mempertimbangkan panjang dokumen. Akibatnya, dokumen panjang cenderung memiliki bobot lebih tinggi, meskipun kata-kata tersebut mungkin kurang relevan [7]. Pada penelitian lain yang dilakukan untuk mendeteksi plagiarisme, dilakukan implementasi algoritma *Levenshtein Distance*. Hasil penelitian tersebut menunjukkan bahwa *Levenshtein Distance* mampu mendeteksi plagiarisme dan memberikan hasil yang baik untuk kata/kalimat pendek, tetapi kurang efektif untuk dokumen yang panjang [8].

Seiring dengan perkembangan teknologi, semakin banyak algoritma ataupun metode yang dapat diterapkan untuk mengidentifikasi plagiarisme, salah satunya adalah algoritma *Smith-Waterman*. Algoritma ini awalnya digunakan untuk mengidentifikasi atau menghitung kemiripan urutan biologis, seperti DNA, dengan cara menyelaraskan dua urutan. Dengan menganalogikan kata dalam kalimat seperti gen atau protein dalam DNA, algoritma ini bisa diterapkan untuk membandingkan kalimat [9]. Algoritma *Smith-Waterman* adalah teknik yang digunakan untuk mengidentifikasi kesamaan lokal (penyelarasan urutan), yang melibatkan penyusunan dua urutan lokal (serangkaian atau rangkaian). Berdasarkan fungsi proses penyelarasan urutan, algoritma ini dapat diterapkan dalam pemrograman komputer untuk mendeteksi kesamaan atau mengukur tingkat kemiripan antara suatu teks dengan teks lainnya [1]. Seperti yang sudah dilakukan pada sebuah penelitian [10] untuk mendeteksi plagiasi dokumen menggunakan algoritma *Smith-Waterman*. Hasil penelitian yang sudah dilakukan menunjukkan bahwa algoritma *Smith-Waterman* mampu mendeteksi mendeteksi kesamaan dokumen dengan menemukan letak kesamaan dari dokumen yang dibandingkan.

Dari penjelasan di atas, penting bagi dunia pendidikan untuk mengembangkan dan menerapkan sistem atau model deteksi plagiarisme yang lebih efektif dan efisien, guna menjaga integritas akademik dan mendorong mahasiswa untuk menghasilkan sebuah karya proyek yang orisinil. Penelitian ini akan menggunakan algoritma *Smith-Waterman* untuk menciptakan solusi yang dapat meminimalisir terjadinya plagiarisme, yaitu mengidentifikasi kemiripan judul proyek atau skripsi sejak dini. Penelitian ini diharapkan dapat memberikan hasil yang optimal dalam menghadapi masalah plagiarisme di lingkungan pendidikan.

## 2. METODE PENELITIAN

Metode penelitian yang dilakukan pada dapat dilihat pada kerangka kerja berikut.


**Gambar 1.** Kerangka Kerja Penelitian

1. Identifikasi Masalah
   Pada tahap ini dilakukan pengenalan dan perumusan masalah yang ingin diselesaikan, yaitu masalah plagiarisme dan bagaimana cara mengidentifikasi kemiripan teks atau kata.
2. Studi Literatur
   Pada tahap ini dilakukan pengumpulan informasi dari artikel ilmiah, jurnal, buku dan sumber-sumber lain yang relevan untuk memahami latar belakang masalah, metode yang sudah ada, dan pendekatan terbaik untuk menyelesaikan masalah.
3. Pengumpulan Data
   Pada tahap ini dilakukan pengumpulan data yang diperlukan untuk analisis lebih lanjut. Data ini bisa berupa teks atau judul proyek yang akan dianalisis kemiripannya.
4. Analisis Data
   Setelah pengumpulan data, tahapan berikutnya yaitu melakukan analisis data untuk mengetahui apakah data tersebut bisa digunakan dengan cara melihat struktur atau ciri-ciri yang relevan untuk pendeteksian plagiarisme.
5. *Preprocessing* Data

Pada tahapan ini dilakukan pemrosesan awal terhadap data seperti *case folding* teks, *stemming* teks, dan *tokenizing* sebelum data tersebut diproses pada tahapan berikutnya.

6. Penerapan *Smith-Waterman*
   Pada tahapan ini, algoritma *Smith-Waterman* diterapkan untuk menyelaraskan urutan teks dan mendeteksi kesamaan lokal antara dokumen atau teks yang dianalisis.
7. Pengujian
   Tahapan berikutnya adalah melakukan pengujian terhadap sistem atau model yang telah dikembangkan untuk memastikan kinerjanya sesuai dengan harapan, serta untuk mengukur keefektifan algoritma dalam mengidentifikasi plagiarisme.
8. Analisis Hasil
   Pada tahapan ini, hasil dari pengujian sebelumnya akan dianalisis untuk memastikan apakah hasil tersebut sesuai dengan tujuan dari penelitian.
9. Penulisan Laporan
   Tahapan terakhir yaitu menulis laporan penelitan. Penulisan laporan penelitian dilakukan sebagai bentuk tanggung jawab dari penelitian yang telah dilakukan.

### 2.1 Plagiarisme

Plagiarisme umumnya didefinisikan sebagai tindakan mengambil karya, pemikiran, atau pendapat orang lain tanpa menyebutkan sumbernya, sehingga tampak seolah-olah itu adalah hasil karya, pemikiran, atau pendapat sendiri [4]. Tindakan ini tidak hanya merugikan penulis asli tetapi juga mengancam integritas akademik dan nilai kejujuran di lingkungan pendidikan.

Plagiarisme telah menjadi masalah yang telah lama ada di institusi akademik, dan seiring dengan meningkatnya akses terhadap informasi digital, praktik ini semakin meluas. Identifikasi plagiarisme merupakan tugas yang sulit serta memakan waktu, karena seringkali membutuhkan upaya besar dari para akademisi untuk meninjau dan menilai setiap tugas yang dikumpulkan secara menyeluruh [11]. Mereka harus menggunakan berbagai alat dan metode untuk memastikan bahwa karya yang diserahkan oleh mahasiswa adalah hasil orisinil, yang pada akhirnya menambah beban kerja mereka. Oleh karena itu, perlunya pendekatan yang lebih efisien dan teknologi yang mendukung menjadi semakin mendesak untuk mengatasi tantangan ini secara efektif.

### 2.2 *Case Folding*

*Case folding* adalah proses mengubah semua huruf dalam sebuah kata menjadi huruf kecil. Tujuan dari case folding adalah menyamakan bentuk semua kata dengan mengubahnya menjadi huruf kecil, sehingga teks yang diproses memiliki format yang seragam [12]. Teknik ini sangat penting dalam pemrosesan bahasa alami (NLP), di mana perbedaan kapitalisasi huruf tidak membawa informasi penting terkait makna kata itu sendiri. Sebagai contoh, dalam analisis teks, kata "Apple" dan "apple" seharusnya diperlakukan sebagai entitas yang sama, karena perbedaan kapitalisasi hanya berfungsi untuk menunjukkan aspek gaya penulisan, bukan perbedaan makna.

### 2.3 *Stemming*

*Stemming* adalah proses mengidentifikasi dan memisahkan kata dasar dari imbuhan dalam sebuah kalimat, dengan memisahkan bagian dasar kata dari awalan (*prefixes*), sisipan (*infixes*), dan akhiran (*suffixes*) yang melekat padanya [13]. Tujuan utama dari *stemming* adalah untuk mengurangi variasi kata yang dihasilkan oleh penggunaan imbuhan, sehingga kata-kata yang memiliki akar makna yang sama dapat dikelompokkan bersama. Misalnya, kata-kata seperti "berlari", "pelari", dan "dilakukan" akan diproses menjadi bentuk dasar yang serupa, seperti "lari" atau "laku", yang membantu meningkatkan efisiensi dalam analisis teks. Dengan menerapkan *stemming*, sistem dapat mengurangi jumlah bentuk kata yang perlu dianalisis, karena variasi kata yang muncul akibat imbuhan dapat disederhanakan menjadi bentuk dasar yang sama. Hal ini memudahkan model untuk memahami makna dasar dari teks yang diproses.

Namun, algoritma yang digunakan dalam stemming dapat bervariasi tergantung pada bahasa yang digunakan, karena setiap bahasa memiliki aturan morfologi yang berbeda. Aturan ini mempengaruhi cara imbuhan ditambahkan ke kata dasar dan bagaimana mereka dipisahkan. Sebagai contoh, kata umum yang disediakan oleh library *Sastrawi* dirancang untuk digunakan pada teks Bahasa Indonesia secara umum [14].

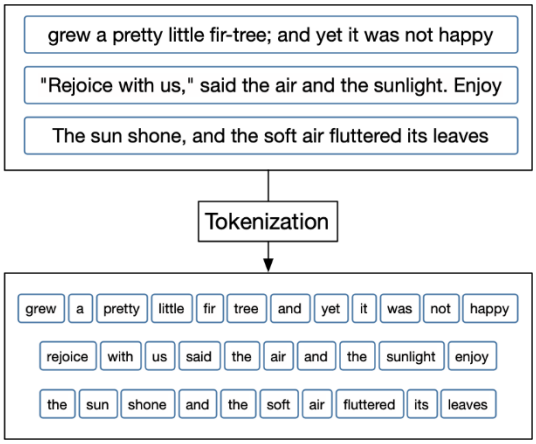
### 2.4 *Tokenizing*

*Tokenizing* atau tokenisasi adalah metode yang membagi rangkaian karakter dalam teks menjadi kata-kata, dengan tujuan membedakan karakter tertentu yang mungkin dianggap atau tidak dianggap sebagai pemisah kata [12]. Proses tokenisasi bertujuan untuk memecah teks mentah menjadi unit-unit yang lebih kecil yang disebut token. Token ini bisa berupa kata, frasa, angka, atau simbol yang memiliki arti tersendiri dalam konteks pemrosesan bahasa alami.

Contoh pemisah kata adalah karakter spasi seperti Enter, Tab, dan spasi biasa. Namun, karakter seperti tanda kutip tunggal (`), titik (.), titik koma (;), dan titik dua (:) juga dapat memiliki peran sebagai pemisah kata. Pemisah-

pemisah ini mengatur bagaimana kata atau simbol diidentifikasi dan dipisahkan, dan keputusan mengenai cara menangani karakter-karakter ini sangat bergantung pada konteks dan tujuan analisis.

Ilustrasi proses *tokenizing* dapat dilihat pada Gambar 2 berikut.




**Gambar 2.** Ilustrasi *Tokenizing*

### 2.5 *Smith-Waterman*

Penyelarasan urutan (*sequence alignment*) adalah salah satu teknik yang paling signifikan dalam bioinformatika. Hasil dari penyelarasan urutan menjadi dasar bagi banyak langkah selanjutnya. Teknik ini dapat digunakan untuk menemukan perbedaan dan kesamaan antara urutan-urutan yang diselaraskan [15], yang merupakan dasar bagi pengenalan urutan biologis, prediksi struktur, dan analisis fungsi.

Algoritma *Smith-Waterman* pada awalnya digunakan untuk mengidentifikasi atau menghitung kemiripan urutan biologis, seperti DNA, dengan cara menyelaraskan dua urutan. Dengan menganalogikan kata dalam kalimat seperti gen atau protein dalam DNA, algoritma ini bisa diterapkan untuk membandingkan kalimat [9]. Dalam konteks ini, *Smith-Waterman* mencari kesamaan antara dua urutan dengan cara menyelaraskan bagian-bagian tertentu dari urutan tersebut, memaksimalkan kemiripan lokal dan mengabaikan bagian-bagian yang tidak relevan.

Algoritma *Smith-Waterman* adalah teknik yang digunakan untuk mengidentifikasi kesamaan lokal (penyelarasan urutan), yang melibatkan penyusunan dua urutan lokal (serangkaian atau rangkaian). Proses ini melibatkan penilaian kecocokan setiap elemen dari urutan satu dengan elemen urutan lainnya, dengan mempertimbangkan berbagai kemungkinan pergeseran dan penyisipan untuk menemukan kesamaan maksimal. Berdasarkan fungsi proses penyelarasan urutan ini, algoritma ini dapat diterapkan dalam berbagai bidang, termasuk pemrograman komputer, untuk mendeteksi kesamaan atau mengukur tingkat kemiripan antara suatu teks dengan teks lainnya [1]. Misalnya, dalam aplikasi pemrosesan bahasa alami (NLP), algoritma *Smith-Waterman* digunakan untuk membandingkan teks atau kalimat untuk mencari kesamaan semantik, yang berguna dalam berbagai tugas seperti pencocokan teks, pemeriksaan plagiarisme, atau analisis sentimen.

Algoritma *Smith-Waterman* yang digunakan untuk perhitungan *local alignment* memperhatikan beberapa langkah penting sebagai berikut [1].

1. Menambahkan nilai pada setiap perbandingan, dengan ketentuan :
   a. Menggunakan nilai positif jika terdapat kemiripan.
   b. Menggunakan nilai negatif jika tidak ada kemiripan.
2. Memulai dengan menginisialisasi matriks dengan nilai 0 (nol).
3. Mengisi matriks penilaian menggunakan persamaan berikut.

$$H_{ij} = max\begin{Bmatrix} H[i-1,j-1] + S(i,j), \\ H[i,j-1] + W, \\ H[i-1,j] + W \end{Bmatrix} \quad (1)$$

Dimana :

H : Skor dalam Matriks
i : Baris
j : Kolom
S : Skor Perbandingan Karakter
W : Gap pada Perbandingan Karakter

4. Melakukan *traceback* mulai dari nilai tertinggi yang ditemukan pada matriks, tanpa memandang posisi.
5. Menyusun urutan berdasarkan hasil *traceback* yang diperoleh. Jika arah *traceback* ke atas, beri simbol garis datar (-) pada urutan atas. Jika arah traceback ke samping, beri simbol garis datar (-) pada urutan bawah. Jika arah traceback diagonal, tulis kedua urutan tersebut.
6. Menghitung *similarity score* dengan menjumlahkan nilai dari skor yang dimiliki oleh setiap urutan yang dibandingkan.

Untuk menghitung persentase kemiripan teks yang dibandingkan dapat menggunakan persamaan berikut**.**

$$similarity\ percentage = \frac{similarity\ score}{maximum\ possible\ score} \times 100\% \quad \text{.....(2)}$$

$$maximum\ possible\ score = min\{first\ token\ lenght, second\ token\ lenght\} \times match \quad \text{.....(3)}$$

Dimana :

| | |
|---|---|
| *similarity percentage* | : Persentase kemiripan data |
| *similarity score* | : Skor kemiripan hasil *traceback* |
| *maximum possible score* | : Skor maksimum kemiripan yang bisa diperoleh |
| *first token length* | : Panjang/jumlah token pertama |
| *second token leght* | : Panjang/jumlah token kedua |
| *match* | : Nilai skor ketika pasangan token adalah sama |

## 3. HASIL DAN ANALISIS *(Spacing: Before 20 pt; After 6 pt)*

### 3.1 Pengumpulan Data

Penelitian ini menggunakan 13 judul proyek mahasiswa dari mata kuliah Pengembangan Web di Program Studi Informatika, Universitas Satya Terra Bhinneka sebagai sumber data. Dalam perhitungan menggunakan algoritma *Smith-Waterman*, data tersebut berfungsi sebagai data uji yang dianalisis. Adapun 13 judul proyek tersebut dapat dilihat pada tabel berikut.

**Tabel 1.** Data Judul Proyek Mahasiswa

| No. | Kode | Judul Proyek |
|---|---|---|
| 1 | D1 | Digitalisasi Manajemen Laundry dengan Aplikasi Berbasis Web |
| 2 | D2 | Optimalisasi Manajemen Laundry dengan Aplikasi Berbasis Web |
| 3 | D3 | Pengembangan Aplikasi Web Manajemen Absensi dan Keselamatan Karyawan |
| 4 | D4 | Pengembangan Aplikasi Point of Sales (POS) Berbasis Web |
| 5 | D5 | Pengembangan Aplikasi Point of Sales dan Pemesanan Makanan Berbasis Web |
| 6 | D6 | Optimalisasi Proses Reservasi Restoran Melalui Web Application E-Booking |
| 7 | D7 | Rancang Bangun Aplikasi Pre-Order Ikan dan Seafood Berbasis Web |
| 8 | D8 | Pengembangan Aplikasi Reservasi Salon Kencatikan Berbasis Web |
| 9 | D9 | Digitalisasi Pemesanan Makanan dan Pencatatan Transaksi dengan Aplikasi Web |
| 10 | D10 | Pengembangan Aplikasi Web Untuk Penugasan Siswa SD 104214 |
| 11 | D11 | Pengembangan Aplikasi Penjualan Bakso Berbasis Web |
| 12 | D12 | Pemanfaatan Aplikasi Web Sebagai Sarana Promosi dan Manajemen Bisnis |
| 13 | D13 | Pengembangan Bimbel Digital Sebagai Solusi Modern Untuk Pembelajaran Yang Lebih Personal Dan Efektif |

### 3.2 *Case Folding* Data

Berdasarkan Tabel 1, setelah seluruh data uji terkumpul, langkah berikutnya adalah tahap praproses *case folding*, yaitu mengubah semua huruf kapital menjadi huruf kecil. Adapun data hasil *case folding* ditunjukkan pada Tabel 2 berikut.

**Tabel 2.** Data Judul Proyek Mahasiswa Hasil *Case Folding*

| No. | Kode | Judul Proyek |
|---|---|---|
| 1 | D1 | digitalisasi manajemen laundry dengan aplikasi berbasis web |
| 2 | D2 | pengembangan aplikasi reservasi salon kencatikan berbasis web |
| 3 | D3 | pengembangan aplikasi web manajemen absensi dan keselamatan karyawan |
| 4 | D4 | pengembangan aplikasi point of sales (pos) berbasis web |
| 5 | D5 | pengembangan aplikasi point of sales dan pemesanan makanan berbasis web |
| 6 | D6 | optimalisasi proses reservasi restoran melalui web application e-booking |
| 7 | D7 | rancang bangun aplikasi pre-order ikan dan seafood berbasis web |

| | | |
|---|---|---|
| 8 | D8 | optimalisasi manajemen laundry dengan aplikasi berbasis web |
| 9 | D9 | digitalisasi pemesanan makanan dan pencatatan transaksi dengan aplikasi web |
| 10 | D10 | pengembangan aplikasi web untuk penugasan siswa sd 104214 |
| 11 | D11 | pengembangan aplikasi penjualan bakso berbasis web |
| 12 | D12 | pemanfaatan aplikasi web sebagai sarana promosi dan manajemen bisnis |
| 13 | D13 | pengembangan bimbel digital sebagai solusi modern untuk pembelajaran yang lebih personal dan efektif |

### 3.3 *Stemming* Data

Setelah melalui proses *case folding*, data judul proyek seperti yang ada pada Tabel 2 akan dilanjutkan ke proses *stemming* untuk mengurangi variasi kata yang dihasilkan oleh penggunaan imbuhan, sehingga kata-kata tersebut akan kembali ke kata dasarnya. Proses *stemming* akan menggunakan *Sastrawi stemmer* karena *Sastrawi* dirancang untuk digunakan pada teks Bahasa Indonesia dan ini cocok untuk data yang digunakan. Data hasil proses *stemming* dapat dilihat pada Tabel 3 berikut.

**Tabel 3.** Data Judul Proyek Mahasiswa Hasil *Stemming*

| No. | Kode | Judul Proyek |
|---|---|---|
| 1 | D1 | digitalisasi manajemen laundry dengan aplikasi bas web |
| 2 | D2 | optimalisasi manajemen laundry dengan aplikasi bas web |
| 3 | D3 | kembang aplikasi web manajemen absensi dan selamat karyawan |
| 4 | D4 | kembang aplikasi point of sales pos bas web |
| 5 | D5 | kembang aplikasi point of sales dan mesan makan bas web |
| 6 | D6 | optimalisasi proses reservasi restoran lalu web application e-booking |
| 7 | D7 | rancang bangun aplikasi pre-order ikan dan seafood bas web |
| 8 | D8 | kembang aplikasi reservasi salon cantik bas web |
| 9 | D9 | digitalisasi mesan makan dan catat transaksi dengan aplikasi web |
| 10 | D10 | kembang aplikasi web untuk tugas siswa sd 104214 |
| 11 | D11 | kembang aplikasi jual bakso bas web |
| 12 | D12 | manfaat aplikasi web bagai sarana promosi dan manajemen bisnis |
| 13 | D13 | kembang bimbel digital bagai solusi modern untuk ajar yang lebih personal dan efektif |

### 3.4 *Tokenizing* Data

Pada tahapan ini, data judul proyek yang telah melalui proses *stemming* seperti yang terlihat pada Tabel 3 akan dilanjutkan ke proses *tokenizing* untuk untuk memecah teks mentah menjadi unit-unit yang lebih kecil yang disebut token. Data hasil proses *tokenizing* dapat dilihat pada Tabel 4 berikut.

**Tabel 4.** Data Judul Proyek Mahasiswa Hasil *Tokenizing*

| No. | Kode | Judul Proyek |
|---|---|---|
| 1 | D1 | 'digitalisasi', 'manajemen', 'laundry', 'dengan', 'aplikasi', 'bas', 'web' |
| 2 | D2 | 'optimalisasi', 'manajemen', 'laundry', 'dengan', 'aplikasi', 'bas', 'web' |
| 3 | D3 | 'kembang', 'aplikasi', 'web', 'manajemen', 'absensi', 'dan', 'selamat', 'karyawan' |
| 4 | D4 | 'kembang', 'aplikasi', 'point', 'of', 'sales', 'pos', 'bas', 'web' |
| 5 | D5 | 'kembang', 'aplikasi', 'point', 'of', 'sales', 'dan', 'mesan', 'makan', 'bas', 'web' |
| 6 | D6 | 'optimalisasi', 'proses', 'reservasi', 'restoran', 'lalu', 'web', 'application', 'e-booking' |
| 7 | D7 | 'rancang', 'bangun', 'aplikasi', 'pre-order', 'ikan', 'dan', 'seafood', 'bas', 'web' |
| 8 | D8 | 'kembang', 'aplikasi', 'reservasi', 'salon', 'cantik', 'bas', 'web' |
| 9 | D9 | 'digitalisasi', 'mesan', 'makan', 'dan', 'catat', 'transaksi', 'dengan', 'aplikasi', 'web' |
| 10 | D10 | 'kembang', 'aplikasi', 'web', 'untuk', 'tugas', 'siswa', 'sd', '104214' |
| 11 | D11 | 'kembang', 'aplikasi', 'jual', 'bakso', 'bas', 'web' |
| 12 | D12 | 'manfaat', 'aplikasi', 'web', 'bagai', 'sarana', 'promosi', 'dan', 'manajemen', 'bisnis' |
| 13 | D13 | 'kembang', 'bimbel', 'digital', 'bagai', 'solusi', 'modern', 'untuk', 'ajar', 'yang', 'lebih', 'personal', 'dan', 'efektif' |

### 3.5 Idenfikasi Kemiripan Data dengan *Smith-Waterman*

Pada tahapan ini, *Smith-Waterman* akan digunakan untuk mengidentifikasi kemiripan yang paling signifikan dari dua buah rangkaian teks yang diambil dari data judul proyek sebelumnya. Langkah pertama adalah melakukan inisiasi matriks pada dua buah urutan teks. Selanjutnya matriks tersebut akan diisi dengan skor yang sesuai. Adapun matriks yang terbentuk adalah matriks berukuran M x N, dimana M adalah banyaknya token pada urutan pertama dan N adalah

banyaknya token pada urutan kedua. Misalkan untuk perbandingan D1 dan D2 pada Tabel 4 sebelumnya, maka matriks skor yang terbentuk dapat dilihat pada Tabel 5 berikut.

**Tabel 5.** Matriks perhitungan skor untuk sample D1 dan D2

| | **-** | **digitalisasi** | **manajemen** | **laundry** | **dengan** | **aplikasi** | **bas** | **web** |
|---|---|---|---|---|---|---|---|---|
| **-** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **optimalisasi** | 0 | | | | | | | |
| **manajemen** | 0 | | | | | | | |
| **laundry** | 0 | | | | | | | |
| **dengan** | 0 | | | | | | | |
| **aplikasi** | 0 | | | | | | | |
| **bas** | 0 | | | | | | | |
| **web** | 0 | | | | | | | |

Untuk melakukan pengisian matriks akan dilakukan perbandingan, dimana jika token *i* sama dengan token *j* maka akan diberikan nilai *match*, yaitu 3. Sebaliknya, apabila token dari kedua urutan berbeda maka akan diberikan nilai *mismatch*, yaitu -1 serta *gap*, yaitu -2. Adapun untuk menentukan skor digunakan formula dari persamaan 1 di atas.

Adapun perhitungan untuk sample matriks pada Tabel 5 di atas adalah sebagai berikut.

Untuk $H_{11}$ :

$$H_{11} = max\{ H[1-1,1-1] + S(1,1), H[1,1-1] + W, H[1-1,1] + W \}$$
$$H_{11} = max\{ H[0,0] + S(1,1), H[1,0] + W, H[0,1] + W \}$$
$$H_{11} = max\{ 0 + \text{-}1, 0 + \text{-}2, 0 + \text{-}2 \}$$
$$H_{11} = max\{ \text{-}1, \text{-}2, \text{-}2 \}$$
$$H_{11} = \text{-}1$$

Untuk $H_{12}$ :

$$H_{12} = max\{ H[1-1,2-1] + S(1,2), H[1,2-1] + W, H[1-1,2] + W \}$$
$$H_{12} = max\{ H[0,1] + S(1,2), H[1,1] + W, H[0,2] + W \}$$
$$H_{12} = max\{ 0 + \text{-}1, \text{-}1 + \text{-}2, 0 + \text{-}2 \}$$
$$H_{12} = max\{ \text{-}1, \text{-}3, \text{-}2 \}$$
$$H_{12} = \text{-}1$$

Untuk $H_{13}$ :

$$H_{13} = max\{ H[1-1,3-1] + S(1,3), H[1,3-1] + W, H[1-1,3] + W \}$$
$$H_{13} = max\{ H[0,2] + S(1,3), H[1,2] + W, H[0,3] + W \}$$
$$H_{13} = max\{ 0 + \text{-}1, \text{-}1 + \text{-}2, 0 + \text{-}2 \}$$
$$H_{13} = max\{ \text{-}1, \text{-}3, \text{-}2 \}$$
$$H_{13} = \text{-}1$$

Untuk $H_{14}$ :

$$H_{14} = max\{ H[1-1,4-1] + S(1,4), H[1,4-1] + W, H[1-1,4] + W \}$$
$$H_{14} = max\{ H[0,3] + S(1,4), H[1,3] + W, H[0,4] + W \}$$
$$H_{14} = max\{ 0 + \text{-}1, \text{-}1 + \text{-}2, 0 + \text{-}2 \}$$
$$H_{14} = max\{ \text{-}1, \text{-}3, \text{-}2 \}$$
$$H_{14} = \text{-}1$$

Untuk $H_{15}$ :

$$H_{15} = max\{ H[1-1,5-1] + S(1,5), H[1,5-1] + W, H[1-1,5] + W \}$$
$$H_{15} = max\{ H[0,4] + S(1,5), H[1,4] + W, H[0,5] + W \}$$
$$H_{15} = max\{ 0 + \text{-}1, \text{-}1 + \text{-}2, 0 + \text{-}2 \}$$
$$H_{15} = max\{ \text{-}1, \text{-}3, \text{-}2 \}$$
$$H_{15} = \text{-}1$$

Untuk $H_{16}$ :

$$H_{16} = max\{ H[1-1,6-1] + S(1,6), H[1,6-1] + W, H[1-1,6] + W \}$$
$$H_{16} = max\{ H[0,5] + S(1,6), H[1,5] + W, H[0,6] + W \}$$
$$H_{16} = max\{ 0 + \text{-}1, \text{-}1 + \text{-}2, 0 + \text{-}2 \}$$
$$H_{16} = max\{ \text{-}1, \text{-}3, \text{-}2 \}$$
$$H_{16} = \text{-}1$$

Untuk $H_{17}$ :

$$H_{17} = max\{ H[1-1,7-1] + S(1,7), H[1,7-1] + W, H[1-1,7] + W \}$$
$$H_{17} = max\{ H[0,6] + S(1,6), H[1,6] + W, H[0,6] + W \}$$
$$H_{17} = max\{ 0 + \text{-}1, \text{-}1 + \text{-}2, 0 + \text{-}2 \}$$

$$H_{17} = max\{\text{-}1, \text{-}3, \text{-}2\}$$
$$H_{17} = \text{-}1$$

Perhitungan akan terus dilakukan hingga baris dan kolom terakhir pada matriks. Sehingga matriks skor yang terbentuk dapat dilihat pada Tabel 6 berikut.

**Tabel 6.** Matriks perhitungan skor untuk sample D1 dan D2

| | - | digitalisasi | manajemen | laundry | dengan | aplikasi | bas | web |
|---|---|---|---|---|---|---|---|---|
| **-** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **optimalisasi** | 0 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| **manajemen** | 0 | -1 | 2 | 0 | -2 | -2 | -2 | -2 |
| **laundry** | 0 | -1 | 0 | 5 | 3 | 1 | -1 | -3 |
| **dengan** | 0 | -1 | -2 | 3 | 8 | 6 | 4 | 2 |
| **aplikasi** | 0 | -1 | -2 | 1 | 6 | 11 | 9 | 7 |
| **bas** | 0 | -1 | -2 | -1 | 4 | 9 | 14 | 12 |
| **web** | 0 | -1 | -2 | -3 | 2 | 7 | 12 | 17 |

Setelah semua skor untuk setiap pasangan token diperoleh, langkah selanjutnya adalah melakukan *traceback*. Proses ini dimulai dari nilai tertinggi dalam matriks, dengan mencari jalur melalui perbandingan nilai dari tiga arah: horizontal, vertikal, dan diagonal. Adapun hasil *traceback* yang diperoleh dapat dilihat pada Tabel 7 dan Tabel 8 berikut.

**Tabel 7.** Jalur *traceback* pada matriks perhitungan skor untuk sample D1 dan D2

| | - | optimalisasi | manajemen | laundry | dengan | aplikasi | bas | web |
|---|---|---|---|---|---|---|---|---|
| **-** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **digitalisasi** | 0 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| **manajemen** | 0 | -1 | 2 | 0 | -2 | -2 | -2 | -2 |
| **laundry** | 0 | -1 | 0 | 5 | 3 | 1 | -1 | -3 |
| **dengan** | 0 | -1 | -2 | 3 | 8 | 6 | 4 | 2 |
| **aplikasi** | 0 | -1 | -2 | 1 | 6 | 11 | 9 | 7 |
| **bas** | 0 | -1 | -2 | -1 | 4 | 9 | 14 | 12 |
| **web** | 0 | -1 | -2 | -3 | 2 | 7 | 12 | 17 |

**Tabel 8.** Hasil *traceback* untuk D1 dan D2

| No. | Sample | Hasil *traceback* |
|---|---|---|
| 1 | D1 dan D2 | web : web = 17; bas : bas = 14; aplikasi : aplikasi = 11; dengan : dengan = 8; laundry : laundry = 5; manajemen : manajemen = 2; digitalisasi : optimalisasi = -1 |

Setelah hasil *traceback* diperoleh, langkah selanjutnya adalah memberikan skor berdasarkan pasangan token yang dilacak. Jika kedua token dalam pasangan tersebut sama, maka diberikan skor 3. Jika tidak sama, diberikan skor -1. Skor untuk setiap pasangan token ini kemudian akan dihitung seperti yang ditunjukkan pada Tabel 9.

**Tabel 9.** Hasil *traceback* untuk D1 dan D2

| No. | Sample | Hasil *traceback* |
|---|---|---|
| 1 | D1 dan D2 | 3 + 3 + 3 + 3 + 3 + 3 + (-1) |

Adapun skor akhir atau *similarity score* yang diperoleh untuk sample D1 dan D2 adalah 17.

Langkah terakhir adalah menghitung persentase kemiripan kedua judul yang dibandingkan sebelumnya. Untuk menghitung persentase kemiripan (*similarity percentage)* judul dapat menggunakan persamaan 2 dan 3 yang sebelumnya sudah dijelaskan.

$$maximum\ possible\ score = min\{7,7\} \times 3$$
$$maximum\ possible\ score = 7 \times 3$$
$$maximum\ possible\ score = 21$$

$$similarity\ percentage = \frac{17}{21} \times 100\%$$
$$similarity\ percentage = 0.8095 \times 100\%$$
$$similarity\ percentage = 80.95\%$$

*Similarity percentage* (persentase kemiripan) dari pasangan D1 dan D2 adalah 80.95%. Skor ini menunjukkan bahwa pada pasangan D1 dan D2, jumlah token yang sama (match) jauh lebih banyak dibandingkan dengan token yang berbeda (mismatch).

Adapun hasil perhitungan untuk semua data dapat dilihat pada Tabel di bawah ini.

**Tabel 10.** Persentase Kemiripan Judul D1 Terhadap D2-13

| No. | Judul Proyek | | Similarity Score | Similarity Percentage (%) |
|---|---|---|---|---|
| 1 | D1 | D2 | 17 | 80.95 |
| 2 | D1 | D3 | 3 | 14.29 |
| 3 | D1 | D4 | 2 | 9.52 |
| 4 | D1 | D5 | 2 | 9.52 |
| 5 | D1 | D6 | 0 | 0 |
| 6 | D1 | D7 | 1 | 4.76 |
| 7 | D1 | D8 | 2 | 9.52 |
| 8 | D1 | D9 | 4 | 19.05 |
| 9 | D1 | D10 | 3 | 14.29 |
| 10 | D1 | D11 | 4 | 22.22 |
| 11 | D1 | D12 | 3 | 14.29 |
| 12 | D1 | D13 | 0 | 0 |

**Tabel 11.** Persentase Kemiripan Judul D2 Terhadap D1, D3-13

| No. | Judul Proyek | | Similarity Score | Similarity Percentage (%) |
|---|---|---|---|---|
| 1 | D2 | D1 | 17 | 80.95 |
| 2 | D2 | D3 | 3 | 14.29 |
| 3 | D2 | D4 | 2 | 9.52 |
| 4 | D2 | D5 | 2 | 9.52 |
| 5 | D2 | D6 | 3 | 14.29 |
| 6 | D2 | D7 | 1 | 4.76 |
| 7 | D2 | D8 | 2 | 9.52 |
| 8 | D2 | D9 | 4 | 19.05 |
| 9 | D2 | D10 | 3 | 14.29 |
| 10 | D2 | D11 | 4 | 22.22 |
| 11 | D2 | D12 | 3 | 14.29 |
| 12 | D2 | D13 | 0 | 0 |

**Tabel 12.** Persentase Kemiripan Judul D3 Terhadap D1-D2, D4-13

| No. | Judul Proyek | | Similarity Score | Similarity Percentage (%) |
|---|---|---|---|---|
| 1 | D3 | D1 | 3 | 14.29 |
| 2 | D3 | D2 | 3 | 14.29 |
| 3 | D3 | D4 | 6 | 25 |
| 4 | D3 | D5 | 6 | 25 |
| 5 | D3 | D6 | 1 | 4.17 |
| 6 | D3 | D7 | 2 | 8.33 |
| 7 | D3 | D8 | 6 | 28.57 |
| 8 | D3 | D9 | 5 | 20.83 |
| 9 | D3 | D10 | 9 | 37.5 |
| 10 | D3 | D11 | 6 | 33.33 |
| 11 | D3 | D12 | 5 | 20.83 |
| 12 | D3 | D13 | 3 | 12.5 |

**Tabel 13.** Persentase Kemiripan Judul D4 Terhadap D1-D3, D5-13

| No. | Judul Proyek | | Similarity Score | Similarity Percentage (%) |
|---|---|---|---|---|
| 1 | D4 | D1 | 2 | 9.52 |
| 2 | D4 | D2 | 2 | 9.52 |
| 3 | D4 | D3 | 6 | 25 |
| 4 | D4 | D5 | 16 | 66.67 |
| 5 | D4 | D6 | 0 | 0 |

| | | | | |
|---|---|---|---|---|
| 6 | D4 | D7 | 4 | 16.67 |
| 7 | D4 | D8 | 7 | 33.33 |
| 8 | D4 | D9 | 2 | 8.33 |
| 9 | D4 | D10 | 6 | 25 |
| 10 | D4 | D11 | 6 | 33.33 |
| 11 | D4 | D12 | 2 | 8.33 |
| 12 | D4 | D13 | 3 | 12.5 |

**Tabel 14.** Persentase Kemiripan Judul D5 Terhadap D1-D4, D6-13

| No. | Judul Proyek | | Similarity Score | Similarity Percentage (%) |
|---|---|---|---|---|
| 1 | D5 | D1 | 2 | 9.52 |
| 2 | D5 | D2 | 2 | 9.52 |
| 3 | D5 | D3 | 6 | 25 |
| 4 | D5 | D4 | 16 | 66.67 |
| 5 | D5 | D6 | 0 | 0 |
| 6 | D5 | D7 | 4 | 14.81 |
| 7 | D5 | D8 | 6 | 28.57 |
| 8 | D5 | D9 | 5 | 18.52 |
| 9 | D5 | D10 | 6 | 25 |
| 10 | D5 | D11 | 6 | 33.33 |
| 11 | D5 | D12 | 2 | 7.41 |
| 12 | D5 | D13 | 3 | 10 |

**Tabel 15.** Persentase Kemiripan Judul D6 Terhadap D1-D5, D7-13

| No. | Judul Proyek | | Similarity Score | Similarity Percentage (%) |
|---|---|---|---|---|
| 1 | D6 | D1 | 0 | 0 |
| 2 | D6 | D2 | 3 | 14.29 |
| 3 | D6 | D3 | 1 | 4.17 |
| 4 | D6 | D4 | 0 | 0 |
| 5 | D6 | D5 | 0 | 0 |
| 6 | D6 | D7 | 0 | 0 |
| 7 | D6 | D8 | 1 | 4.76 |
| 8 | D6 | D9 | 0 | 0 |
| 9 | D6 | D10 | 1 | 4.17 |
| 10 | D6 | D11 | 0 | 0 |
| 11 | D6 | D12 | 1 | 4.17 |
| 12 | D6 | D13 | 0 | 0 |

**Tabel 16.** Persentase Kemiripan Judul D7 Terhadap D1-D6, D8-13

| No. | Judul Proyek | | Similarity Score | Similarity Percentage (%) |
|---|---|---|---|---|
| 1 | D7 | D1 | 1 | 4.76 |
| 2 | D7 | D2 | 1 | 4.76 |
| 3 | D7 | D3 | 2 | 8.33 |
| 4 | D7 | D4 | 4 | 16.67 |
| 5 | D7 | D5 | 4 | 14.81 |
| 6 | D7 | D6 | 0 | 0 |
| 7 | D7 | D8 | 3 | 14.29 |
| 8 | D7 | D9 | 1 | 3.7 |
| 9 | D7 | D10 | 2 | 8.33 |
| 10 | D7 | D11 | 2 | 11.11 |
| 11 | D7 | D12 | 2 | 7.41 |
| 12 | D7 | D13 | 0 | 0 |

**Tabel 17.** Persentase Kemiripan Judul D8 Terhadap D1-D7, D9-13

| No. | Judul Proyek | | Similarity Score | Similarity Percentage (%) |
|---|---|---|---|---|
| 1 | D8 | D1 | 2 | 9.52 |
| 2 | D8 | D2 | 2 | 9.52 |

| | | | | |
|---|---|---|---|---|
| 3 | D8 | D3 | 6 | 28.57 |
| 4 | D8 | D4 | 7 | 33.33 |
| 5 | D8 | D5 | 6 | 28.57 |
| 6 | D8 | D6 | 1 | 4.76 |
| 7 | D8 | D7 | 3 | 14.29 |
| 8 | D8 | D9 | 2 | 9.52 |
| 9 | D8 | D10 | 6 | 28.57 |
| 10 | D8 | D11 | 8 | 44.44 |
| 11 | D8 | D12 | 2 | 9.52 |
| 12 | D8 | D13 | 3 | 14.29 |

**Tabel 18.** Persentase Kemiripan Judul D9 Terhadap D1-D8, D10-13

| No. | Judul Proyek | | Similarity Score | Similarity Percentage (%) |
|---|---|---|---|---|
| 1 | D9 | D1 | 4 | 19.05 |
| 2 | D9 | D2 | 4 | 19.05 |
| 3 | D9 | D3 | 5 | 20.83 |
| 4 | D9 | D4 | 2 | 8.33 |
| 5 | D9 | D5 | 5 | 18.52 |
| 6 | D9 | D6 | 0 | 0 |
| 7 | D9 | D7 | 1 | 3.7 |
| 8 | D9 | D8 | 2 | 9.52 |
| 9 | D9 | D10 | 5 | 20.83 |
| 10 | D9 | D11 | 2 | 11.11 |
| 11 | D9 | D12 | 5 | 18.52 |
| 12 | D9 | D13 | 0 | 0 |

**Tabel 19.** Persentase Kemiripan Judul D10 Terhadap D1-D9, D11-13

| No. | Judul Proyek | | Similarity Score | Similarity Percentage (%) |
|---|---|---|---|---|
| 1 | D10 | D1 | 3 | 14.29 |
| 2 | D10 | D2 | 3 | 14.29 |
| 3 | D10 | D3 | 9 | 37.5 |
| 4 | D10 | D4 | 6 | 25 |
| 5 | D10 | D5 | 6 | 25 |
| 6 | D10 | D6 | 1 | 4.17 |
| 7 | D10 | D7 | 2 | 8.33 |
| 8 | D10 | D8 | 6 | 28.57 |
| 9 | D10 | D9 | 5 | 20.83 |
| 10 | D10 | D11 | 6 | 33.33 |
| 11 | D10 | D12 | 5 | 20.83 |
| 12 | D10 | D13 | 3 | 12.5 |

**Tabel 20.** Persentase Kemiripan Judul D11 Terhadap D1-D10, D12-13

| No. | Judul Proyek | | Similarity Score | Similarity Percentage (%) |
|---|---|---|---|---|
| 1 | D11 | D1 | 4 | 22.22 |
| 2 | D11 | D2 | 4 | 22.22 |
| 3 | D11 | D3 | 6 | 33.33 |
| 4 | D11 | D4 | 6 | 33.33 |
| 5 | D11 | D5 | 6 | 33.33 |
| 6 | D11 | D6 | 0 | 0 |
| 7 | D11 | D7 | 2 | 11.11 |
| 8 | D11 | D8 | 8 | 44.44 |
| 9 | D11 | D9 | 2 | 11.11 |
| 10 | D11 | D10 | 6 | 33.33 |
| 11 | D11 | D12 | 2 | 11.11 |
| 12 | D11 | D13 | 3 | 16.67 |

**Tabel 21.** Persentase Kemiripan Judul D12 Terhadap D1-D11, D13

| No. | Judul Proyek | | Similarity Score | Similarity Percentage (%) |
|---|---|---|---|---|
| 1 | D12 | D1 | 3 | 14.29 |
| 2 | D12 | D2 | 3 | 14.29 |
| 3 | D12 | D3 | 5 | 20.83 |
| 4 | D12 | D4 | 2 | 8.33 |
| 5 | D12 | D5 | 2 | 7.41 |
| 6 | D12 | D6 | 1 | 4.17 |
| 7 | D12 | D7 | 2 | 7.41 |
| 8 | D12 | D8 | 2 | 9.52 |
| 9 | D12 | D9 | 5 | 18.52 |
| 10 | D12 | D10 | 5 | 20.83 |
| 11 | D12 | D11 | 2 | 11.11 |
| 12 | D12 | D13 | 0 | 0 |

**Tabel 22.** Persentase Kemiripan Judul D13 Terhadap D1-D12

| No. | Judul Proyek | | Similarity Score | Similarity Percentage (%) |
|---|---|---|---|---|
| 1 | D13 | D1 | 0 | 0 |
| 2 | D13 | D2 | 0 | 0 |
| 3 | D13 | D3 | 3 | 12.5 |
| 4 | D13 | D4 | 3 | 12.5 |
| 5 | D13 | D5 | 3 | 10 |
| 6 | D13 | D6 | 0 | 0 |
| 7 | D13 | D7 | 0 | 0 |
| 8 | D13 | D8 | 3 | 14.29 |
| 9 | D13 | D9 | 0 | 0 |
| 10 | D13 | D10 | 3 | 12.5 |
| 11 | D13 | D11 | 3 | 16.67 |
| 12 | D13 | D12 | 0 | 0 |

Berdasarkan Tabel 10 sampai Tabel 22, skor tertinggi dicapai oleh pasangan D1 dan D2 dengan *similarity score* 17 dan *similarity percentage* 80.95%. Ini menunjukkan bahwa pasangan token yang sama lebih dominan dibandingkan dengan pasangan token yang berbeda untuk D1 dan D2. Sebaliknya, skor terendah ditemukan pada pasangan D6 dan D13 yang memiliki banyak nilai *similarity score* dan *similarity percentage* sama dengan 0.

## 4. KESIMPULAN

Penelitian ini memanfaatkan algoritma *Smith-Waterman* untuk mengidentifikasi kesamaan antara judul proyek mahasiswa. Algoritma tersebut berhasil mendeteksi kesamaan lokal dengan memberikan nilai kesamaan berdasarkan perbandingan token dalam teks. Hasil pengujian menunjukkan bahwa algoritma *Smith-Waterman* efektif dalam menemukan kesamaan di antara judul-judul proyek, dengan berbagai tingkat persentase kesamaan yang terdeteksi. Penelitian lanjutan dapat difokuskan pada optimisasi algoritma *Smith-Waterman* untuk mempercepat proses deteksi, terutama saat diterapkan pada dataset yang sangat besar. Selain itu, algoritma ini dapat diuji dalam sistem nyata dengan menentukan ambang batas (*threshold*) tertentu untuk memutuskan apakah suatu teks dianggap plagiasi atau tidak, guna meningkatkan akurasi dan efisiensi dalam aplikasi dunia nyata.

## REFERENSI

[1] L. Amsir, H. Harlinda, S. Anraeni, and P. Lestari Lokapitasari Belluano, "Penerapan Algortima Smith Waterman Untuk Mengukur Kemiripan Tugas Kuliah Mahasiswa," *Pros. Semin. Nas. Komun. dan Inform. #3 Tahun*, no. 2009, pp. 63–68, 2019.

[2] E. P. Nimasari, "Persepsi Mahasiswa Terhadap Plagiat," *Prem. Educ. J. Pendidik. Dasar dan Pembelajaran*, vol. 7, no. 02, p. 115, 2017, doi: 10.25273/pe.v7i2.1623.

[3] B. H. Irawan, M. S. H. Simarangkir, and E. Erlinna, "Deteksi Kemiripan Judul Skripsi Menggunakan Algoritma Levenshtein Distance Pada Kampus Stmik Mic Cikarang," *Edutic - Sci. J. Informatics Educ.*, vol. 7, no. 2, pp. 143–149, 2021, doi: 10.21107/edutic.v7i2.10051.

[4] L. Magdalena, R. Lie, D. Chandra, and N. J. Perdana, "Kesadaran Akan Tindakan Plagiarisme Di Kalangan Mahasiswa," *J. Serina Sains, Tek. dan Kedokt.*, vol. 1, no. 1, pp. 123–132, 2023, doi: 10.24912/jsstk.v1i1.27137.

[5] M. F. Azima, A. N. Listanto, P. Studi, T. Informatika, F. Matching, and D. Plagiarisme, "Kombinasi Algoritma TF-IDF dan Fuzzy Matching untuk Deteksi Kemiripan Judul Skripsi," vol. 19, no. x, pp. 1–11, 2024.

[6] H. Sari, G. Leonarde Ginting, and T. Zebua, "Penerapan Algoritma Text Mining Dan Tf-Idf Untuk

Pengelompokan Topik Skripsi," *Terap. Inform. Nusant.*, vol. 2, no. 7, pp. 414–432, 2021, [Online]. Available: https://ejurnal.seminar-id.com/index.php/tin

[7] D. Septiani and I. Isabela, "Analisis Term Frequency Inverse Document Frequency (TF-IDF) Dalam Temu Kembali Informasi Pada Dokumen Teks," *SINTESIA J. Sist. dan Teknol. Inf. Indones.*, vol. 1, no. 2, pp. 81–88, 2023.

[8] R. Adawiyah and N. E. Saragih, "Implementasi Algoritma Levenshtein Distance Dalam Mendeteksi Plagiarisme," *J. Comput. Sci. Inf. Technol.*, vol. 5, no. 1, pp. 54–63, 2022, [Online]. Available: https://jurnal.ulb.ac.id/index.php/JCoInT/article/view/3086

[9] F. Thalib and R. Kusumawati, "Pembuatan Program Aplikasi untuk Pendeteksian Kemiripan Dokumen Teks dengan Algoritma Smith-Waterman," 2014.

[10] A. Meitaningsih, A. S. Aribowo, and N. H. Cahyana, "Text Mining Untuk Mendeteksi Plagiasi Dokumen Dengan Penerapan Stemming Nazief-Adriani Dan Algoritma Smith-Waterman," *Telematika*, vol. 17, no. 2, p. 99, 2020, doi: 10.31315/telematika.v1i1.3377.

[11] H. Cheers, Y. Lin, and S. P. Smith, "Academic source code plagiarism detection by measuring program behavioral similarity," *IEEE Access*, vol. 9, pp. 50391–50412, 2021, doi: 10.1109/ACCESS.2021.3069367.

[12] M. U. Albab, Y. Karuniawati P, and M. N. Fawaiq, "Optimization of the Stemming Technique on Text preprocessing President 3 Periods Topic," *J. Transform.*, vol. 20, no. 2, pp. 1–10, 2023, [Online]. Available: https://journals.usm.ac.id/index.php/transformatika/page1

[13] V. P. Carolina, E. Utami, and A. Yaqin, "Riset Jurnal Literatur : Morfologi Stemming Enhance Confix Stripping, Sastrawi Dan Tala Pada Bahasa Daerah Melayu-Ambon," *J. homepage J. Electr. Eng. Comput.*, vol. xx, No. xx, no. xx, pp. 186–196, 2024, doi: 10.33650/jeecom.v4i2.

[14] A. E. Budiman and A. Widjaja, "Analisis Pengaruh Teks Preprocessing Terhadap Deteksi Plagiarisme Pada Dokumen Tugas Akhir," *J. Tek. Inform. dan Sist. Inf.*, vol. 6, no. 3, pp. 475–488, 2020, doi: 10.28932/jutisi.v6i3.2892.

[15] Z. Xia *et al.*, "A Review of Parallel Implementations for the Smith–Waterman Algorithm," *Interdiscip. Sci. – Comput. Life Sci.*, vol. 14, no. 1, pp. 1–14, 2022, doi: 10.1007/s12539-021-00473-0.