**Penerapan Algoritma *Edit Distance* untuk Pengukuran Kemiripan Antar Dokumen Berbahasa Indonesia**

Iyan Mulyana1, Aries Maesya1, Andi Chairunnas1

1 Program Studi Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Pakuan, Bogor

{Iyan Mulyana, Aries Maesya, Andi Chairunnas, Iyandelon@yahoo.com

**Abstrak.** Salah satu pencegahan tindakan plagiarisme adalah pengukuran kemiripan dokumen dengan tingkat kecepatan dan nilai akurasi yang tinggi. Tujuan dari penelitian ini adalah membuat aplikasi untuk mengukur kemiripan dokumen berbahasa Indonesia menggunakan *edit distance*. Algoritma *edit distance* yaitu algoritrma yang berfungsi untuk mengukur nilai kemiripan atau kesamaan antar kata (*string*) berdasarkan total biaya terkecil dari transformasi salah satu kata menjadi kata yang lain dengan menggunakan *edit-rules,* yaitu penambahan karakter (*insertion*), penggantian karakter (*substitusion*) dan penghapusan karakter (*deletion*). Cara kerja algoritma *edit distance* dalam mencocokkan string yaitu mengukur kesamaan karakter pada urutan yang sama bukan berdasarkan kata yang sama. Hasil penelitian menujukan bahwa pengukuran kemiripan dengan tahap *preprocessing* manghasilkan nilai lebih baik dari pada tanpa *preprocessing.* Tahap *Preprosessing* dapat mengoptimalkan kerja algoritma *edit distance* sehingga meningkatkan nilai akurasi pengukuran kemiripan.

1. **Pendahuluan**

Karya ilmiah merupakan suatu tulisan yang memaparkan hasil [penelitian](http://belajarpsikologi.com/metode-penelitian-kualitatif/) yang telah dilakukan oleh seseorang atau sekelompok masyarakat ilmuwan. Karya ilmiah yang telah ditulis itu diharapkan menjadi wahana transformasi pengetahuan antara sekolah dengan masyarakat, atau orang-orang yang berminat membacanya. Bila karya ilmiah dimuat di media online akan semakin memudahkan banyak orang untuk mengakses karya ilmiah. Kewajiban publikasi karya ilmiah juga akan mendorong seseorang untuk bertindak jujur karena tulisannya dibaca oleh banyak orang dan dapat mengurangi tingkat plagiarism.

Salah satu pencegahan tindakan plagiarisme adalah pengukuran kemiripan dokumen dengan tingkat kecepatan dan nilai akurasi yang tinggi sangat dibutukan. Salah satu cara untuk mengetahui seberapa besar kemiripan suatu dokumen dengan dokumen lainnya dapat dengan menggunakan pendekatan *string metric* yaitu melakukan perbandingan string dengan memasukkannya ke dalam fungsi matematis tertentu. Pada penelitian yang telah dilakukan Goenawan, et al (2005), menyatakan bahwa salah satu algoritma yang dapat digunakan untuk perbandingan string atau pencocokkan string dalam pengukuran kemiripan antar dokumen adalah algoritma *edit distance*. Adapun, komplesitas waktu dari algoritma *edit distance* ini adalah kuadratik O(n²) yang jauh lebih baik daripada algoritma *brute force* dengan kompleksitas waktu eksponensial O().

Pada Penelitian ini digunakan algoritma *edit distance* dalam pengukuran kemiripan antar dokumennya. Algoritma *edit distance* berfungsi untuk mengukur nilai kemiripan atau kesamaan antar kata (*string*) berdasarkan total biaya terkecil dari transformasi salah satu kata menjadi kata yang lain dengan menggunakan *edit-rules,* yaitu penambahan karakter, penggantian karakter dan penghapusan karakter.

Penerapan algoritma *edit distance* diharapkan mampu mendapatkan nilai kemiripan dengan akurasi yang tinggi dan memenuhi kebutuhan efesiensi pengukuran kemiripan antar dokumen.

1.1 Algoritma *Edit Distance*

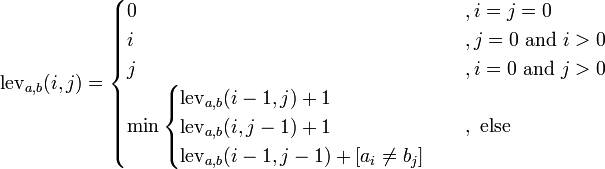
*Edit distance* atau *levenshtein-distance* adalah algoritma yangditemukan oleh Vladimir Levenshtein, seorang ilmuwan Rusia, pada tahun 1965. Algoritma ini berguna untuk memeriksa kemiripan dari dua buah *string* yang umumnya ditemukan pada aplikasi-aplikasi pengecekan suatu ejaan. Perhitungan *edit distance* didapatkan dari matriks yang digunakan untuk menghitung jumlah perbedaan *string* antara dua *string*. Perhitungan jarak antara dua *string* ini ditentukan dari jumlah minimum operasi perubahan untuk membuat *string* A menjadi *string* B Secara umum, operasi mengubah yang diperbolehkan untuk keperluan ini adalah: (Dani, 2006)

For citations in the text please use square brackets and consecutive numbers: [1], [2], [3], etc.

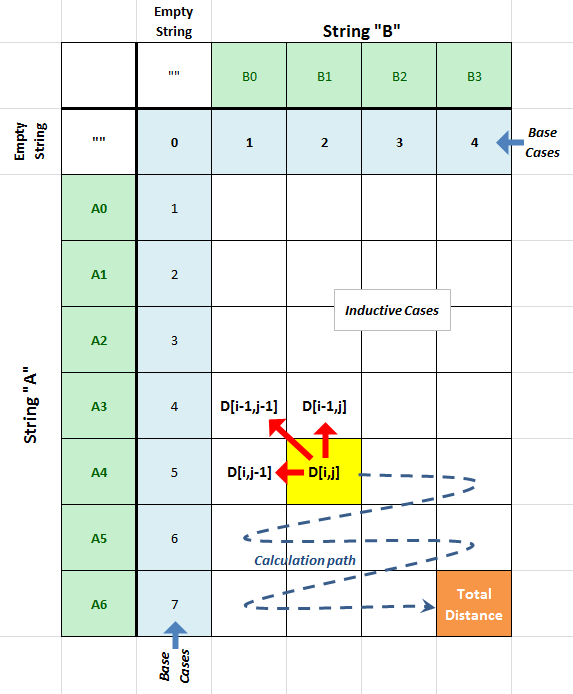
* Memasukkan karakter ke dalam *string*,
* Menghapus sebuah karakter dari suatu *string*,
* Mengganti karakter *string* dengan karakter lain.

Persamaan algoritma *edit distance* :

Silahan gunakan equation untuk menampilkan formula



**Tabel 1** Penentuan nilai matriks *edit distance*



Baris dan kolom pada tabel dua dimensi (matriks) di atas diisi dengan langkah-langkah sebagai berikut:

1. Elemen matriks [0,0] akan diisi dengan nilai 0

2. Elemen matriks [i,0] akan diisi dengan nilai matriks [i-1,0]+1

3. Elemen matriks [0,j] akan diisikan nilai matriks[0,j-1]+1

4. Elemen lainnya (matriks[i,j])diisi dengan urutan langkah di bawah ini:

1. Jika karakter ke-I pada *string* ke-1 memiliki kesamaan dengan karakter ke-J pada *string* ke-2 maka nilai matriks[i-1,j-1] akan dianggap ditambahkan 1 dari nilai sebelumnya.
2. Bandingkan 3 elemen matriks pada posisi matriks[i-1,j-1],matriks[i,j-1], dan matriks[i-1,j] untuk pencarian nilai minimum di antara ketiganya. Elemen dengan nilai terkecil akan dimasukkan nilainya ke dalam matriks[i,j]
3. ulangi langkah 1 dan 2 sampai semua elemen tabel terisi.
   1. **Menghitung Kemiripan (*Similarity)* dengan *Edit Distance***

Pengukuran ini dilakukan untuk mengetahui seberapa besar kemiripan teks antara dokumen asli dan dokumen pembanding. Langkah-langkah dalam menentukan *similarity* adalah: (Winoto, 2012)

Jika dimisalkan

S1 = *Source* *string*

S2 = Target *string*

Setelah dilakukan perhitugan dari kedua *string* tersebut menggunakan *edit distanc*e, maka algoritma ini akan memberikan angka sebagai perbedaaan dari kedua *string*.

Setelah didapatkan distance dari kedua *string* tersebut maka dapat ditentukan formula untuk menghitung derajat *similarity* kedua *string*. Formula tersebut adalah

(1)

Keterangan:

Max(S1,S2) merupakan nilai yang paling panjang yang diberikan dari perbandingan S1 dan S2.

1. **Perancangan Sistem**

Penerapan algoritma *edit distance* pada pengukuran kemiripan dokumen ini mempunyai cara kerja yaitu pertama user memasukkan dokumen teks, kemudian dokumen teks akan diproses pada tahap *preprocessing*. Tahapan ini menghasilkan kumpulan kata-kata penting (kata dasar) yang selanjutnya diukur kemiripannya dengan menerapkan algoritma *edit distance*, yaitu dengan melakukan pencocokan string dari antar dokumen terhadap kumpulan kata-kata. Hasil pengukurannya berupa presentase kemiripan antar dokumen. Cara kerja sistem ditunjukan pada Gambar 1.



Gambar 1. Alur kerja system

Hindari gambar dengan warna latar

Secara garis besar sistem pengukuran kemiripan dokumen teks ini dibangun oleh dua tahapan utama yaitu tahap *preprocessing* (*text preprocessing*) dan tahap pencocokan string (algoritma *edit distance*). Sepeti ditunjukan pada Gambar 2.



Gambar 2. Tahapan Proses Pengukuran Kemiripan Dokumen

* 1. **Tahap *Preprocessing***

Tahap *preprocessing* merupakan tahapan dimana dilakukan proses untuk mengurangi noise yang terdapat dalam dokumen. Tahapan *preprocessing* yang dilakukan dalam sistem ini adalah *case folding* (mengubah ukuran huruf dan penghilangan simbol-simbol), *tokenizing* (pemotongan kata), *filtering* (penghilangan kata-kata yang tidak penting), *stemming* (penghilangan imbuhan) dan *sorting*(pengurutan), seperti yang ditunjukkan pada Gambar 3. Hasil dari tahapan *preprocessing* yaitu kumpulan kata dasar yang akan digunakan sebagai pencocokan string atau pembandingan dokumen.

\



Gambar 3. Tahapan *Preprocessing*

* 1. **Tahap pencocokan string (Algoritma *Edit Distance*)**

Setelah proses *preprocessing* langkah selanjutnya adalah pencocokan string dari dokumen uji dokumen pembanding menggunakan algoritma *edit distance*. Berikut ini *flowchart* algoritma *edit distance* pada Gambar 4.



Gambar 4. *Flowchart* algoritma *edit distance*

1. **Hasil dan Pembahasan**

Hasil dari implementasi penerapan algoritma *edit distance*, terdiri dari dua macam pengukuran yaitu pengukuran kemiripan antar dua dokumen ditunjukkan pada Gambar 5, dan pengukuran kemiripan satu dokumen dengan dokumen yang berada dalam *database* ditunjukkan pada Gambar 6. File yang dapat di ukur kemiripannya hanya file dengan ekstensi .txt.

Pada aplikasi pengukuran kemiripan Dokumenini File .txt sebagai dokumen inputan diproses melalui tahap *preprocessing* antara lain : Pertama *case folding* yang berfungsi untuk mengubah semua huruf menjadi huruf kecil dan selain karakter angka dan huruf dihilangkan (dianggap *delimiter*/pembatas). Kedua adalah proses tokenisasi yang berfungsi untuk memecah sekumpulan karakter dalam suatu teks ke dalam satuan kata. Tahap ketiga adalah proses *filtering* yang berfungsi untuk mengurangi ukuran dimensi dengan menghapus kata-kata yang tidak penting atau disebut sebagai *stopword/stoplist* dan menyimpan kata penting*(wordlist)*, *stopword* ini disimpan pada *database*. Setelah melalui proses *filtering* kemudian ke proses *stemming*, *stemming* ini berfungsi untuk menghapus kata imbuhan menjadi kata dasar,

Proses terakhir pada *preprocessing* adalah *sorting* yang berfungsi untuk mengurutkan string. Hasil *text preprocessing* kemudian dihitung jumlah karakter dan nilai *edit distance* untuk mendapatkan nilai *persentase* kemiripan antar dokumen.



Gambar 5. Halaman Pengukuran (*One to One*)



Gambar 6. Halaman Pengukuran dengan *Database (One To Many)*

1. **Evaluasi sistem**

Evaluasi sistem dilakukan untuk mengetahui tingkat akurasi dari Metode yang digunakan. Evaluasi dilakukan dengan cara membandingkan hasil perhitungan manual dengan hasil sistem.

* 1. **Pengukuran kemiripan melalui perhitungan manual**

Proses penentuan nilai *edit distance* diperoleh dengan mencocokan antar dua string dalam bentuk matriks yang digunakan untuk menghitung jumlah perbedaan antar string. Perhitungan manual untuk memperoleh nilai persentase kemiripan antar dokumen adalah sebagai berikut :

1. Input teks dokumen: S1 = algortma edit distance

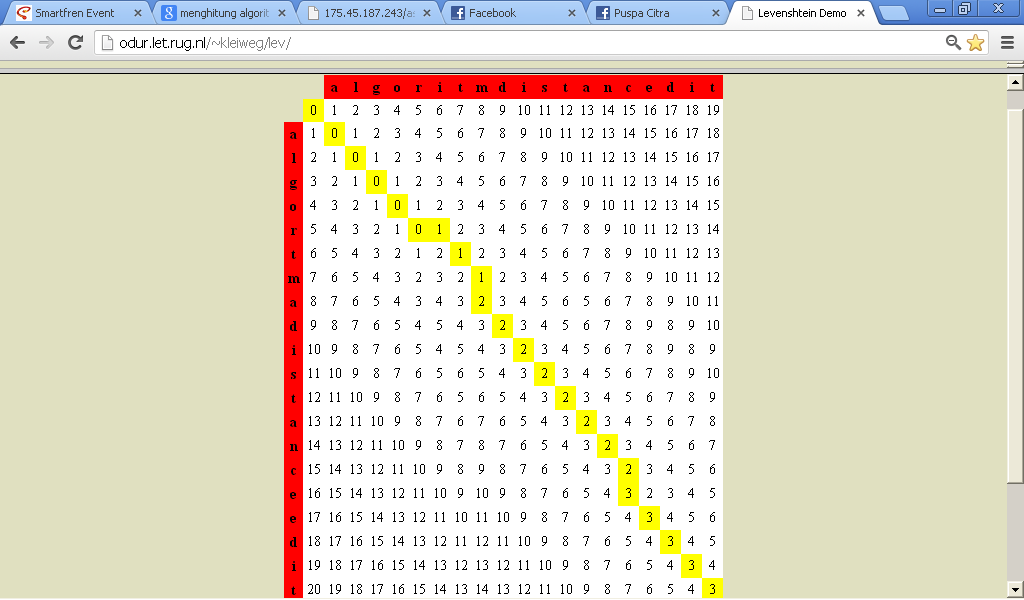
S2 = algoritm edit distanc

1. Setelah isi dokumen melalui *text preprocessing (case folding, tokenizing, filtering, stemming, dan sorting)*

S1 = algortmadistanceedit

S2 = algoritmdistancedit

1. Menentukkan nilai *edit distance* dengan menggunakan matriks dalam tabel:
2. Isi nilai matriks m[i,0] = i
3. Isi nilai matriks m[0,j] = j
4. Jika karakter S1[i] = S2[j] maka isi nilai m[i,j] = m[i-1,j-1]
5. Jika karakter S1[i] ≠ S2[j] maka isi nilai m[i,j] = max(m[i-1,j]+1, m[i,j-1]+1, m[i-1,j-1]+1)



Gambar 7. matriks menentukan nilai *edit distance* pada percobaan 1A

Algoritma *edit distance* digambarkan dalam tabel matriks n+1 × m+1, dimana n dan m adalah panjang karakter dari dua string tersebut. Nilai *edit distance* diperoleh dari tabel matriks di atas, terletak pada pojok kanan bawah yaitu bernilai 3.

*Distance* = 3

*Max*(S1,S2)= 20(panjang dari S1)

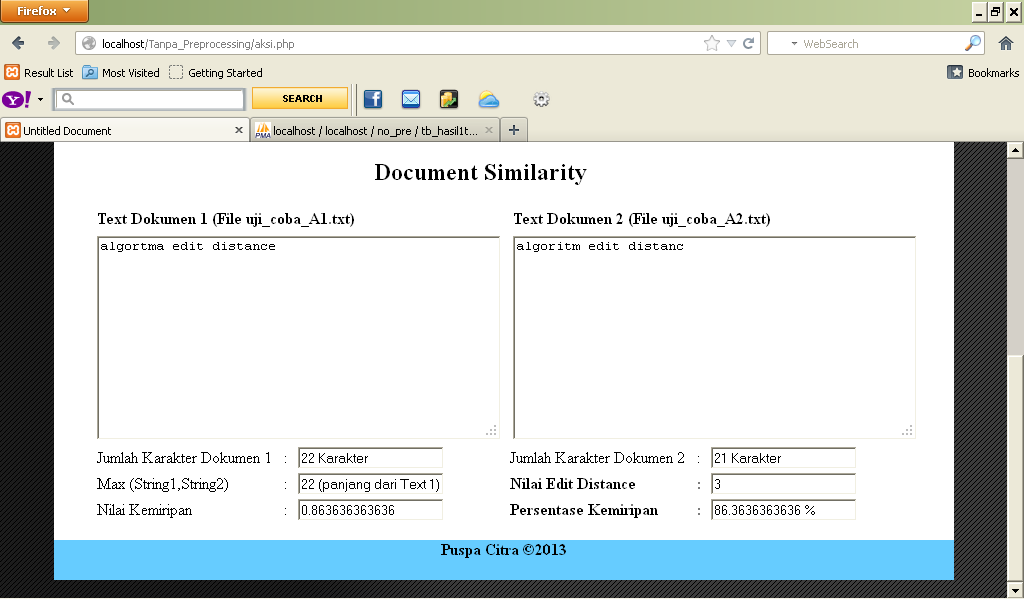
Persentase nilai kemiripan yang diperoleh dari perhituangan manual di atas dari perbandingan string “algortmadistanceedit” dan “algoritmdistancedit” adalah 85%. Memiliki nilai edit distance sama dengan 3, yang berarti mempunyai perbedaaan 3 karakter.

* 1. **Pengukuran kemiripan Menggunakan Aplikasi**

1. Input teks dokumen: S1 = algortma edit distance

S2 = algoritm edit distanc

1. Hasil uji coba pada Gambar 8 diperoleh sebesar 86,36%, karena pada beberapa kata memiliki karakter yang berbeda.



Gambar 8. Pengukuran kemiripan melalui Aplikasi

Setelah beberapa kali dilakukan percobaan terdapat perbedaan hasil persentase pengukuran kemiripan secara manual dengan yang dihasilkan melalui aplikasi sekitar 1% - 3% . Hal ini disebabkan Algoritma *edit distance* dalam pencocokkan string mengukur kesamaan karakter pada urutan yang sama bukan berdasarkan kata yang sama, Kondisi ini yang sering membedakan antar perhitungan manual dan aplikasi.

1. **Simpulan**

Dari hasil penelitian dapat disimpulkan sebagai berikut :

1. Aplikasi pengukuran kemiripan dokumen dengan algoritma *edit distance* dapat digunakan untuk identifikasi plagiarise dengan nilai akurasi sistem sebesar 95 %.
2. Algoritma *edit distance* melakukan pencocokkan string berdasarkan kesamaan karakter pada urutan yang sama bukan berdasarkan kata yang sama. Sehingga Hal ini mengurangi tingkat akurasi proses pengukuran kemiripan antar dua dokumen.
3. Tingkat akurasi Aplikasi pengukuran kemiripan dokumen menggunakan Algoritma *edit distance* sangat dipengaruhi oleh tahap *Preprosessing .* Halini disebabkan melalui *Preprocesing* menghasilkan string yang urut.

**Daftar Pustaka**

1. Dani, T.G. Limandra & L.R.E Adiseputra. 2006. *Deteksi Kemiripan Kode Program dengan Metode Preprocessing dan Perhitungan Leveinshtein* Distance, ISSN : 1411-6286*.* Prosiding Seminar Ilmiah Nasional Komputer dan Sistem Intelijen (KOMMIT 2006). Universitas Gunadarma, Depok.
2. Firdaus, H.D. 2008. *Deteksi Plagiat Dokumen Menggunakan Algoritma Rabin-Karp*. Makalah IF2251 Strategi Algoritmik. Institut Teknologi Bandung, Bandung.
3. Goenawan, W., R. Augustinus & K. Sembiring. 2005.*Penerapan Algoritma Edit Distance Pada Pendeteksian Praktik Plgiat*. Makalah STMIK. Institut Teknologi Bandung, Bandung.
4. Khusnaini, W. 2012. *Penentuan Kemiripan Data Karya Ilmiah IPB Menggunakan Algoritma Levenshtein.*Skripsi. Institut Pertanian Bogor, Bogor.
5. Kurniawati, A., K. A. Sekarwati & I.W.S. Wicaksana. 2012*. Arsitektur untuk aplikasi Deteksi Kesamaan Dokumen Bahasa Indonesia*. Makalah dalam Konferensi Nasional Sistem Informasi 2012. Universitas Gunadarma, Depok.
6. Thalib, K & R. Kusumawati. 2004. *Pembuatan Program Aplikasi untuk Pendeteksi Kemiripan Dokumen Teks dengan Algoritma Smith-Waterman*.Makalah. Universitas Gunadarma, Depok.
7. Winoto, H. 2012. *Deteksi Kemiripan Isi Dokumen Teks Menggunakan Algotitma Levenshtein Distance*. Skripsi. Universitas Islam Negeri Maulana Malik Ibrahim, Malang.