

**PERBANDINGAN ALGORITMA PENCARIAN *STRING*  
DALAM METODE *APPROXIMATE STRING MATCHING*  
UNTUK IDENTIFIKASI KESALAHAN PENGETIKAN  
TEKS BAHASA INDONESIA DAN SARAN PERBAIKAN**



**SKRIPSI**

**Disusun Sebagai Salah Satu Syarat  
Untuk Memperoleh Gelar Sarjana Komputer  
pada Jurusan Ilmu Komputer/ Informatika**

**Disusun oleh :**

**YENY ROCHMAWATI**

**24010311120017**

**JURUSAN ILMU KOMPUTER / INFORMATIKA  
FAKULTAS SAINS DAN MATEMATIKA  
UNIVERSITAS DIPONEGORO**

**2015**

## HALAMAN PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan di bawah ini :

Nama : Yeny Rochmawati

NIM : 24010311120017

Judul : Perbandingan Algoritma Pencarian *String* dalam Metode *Approximate String Matching* untuk Identifikasi Kesalahan Pengetikan Teks Bahasa Indonesia dan Saran Perbaikan

Dengan ini saya menyatakan bahwa dalam tugas akhir/ skripsi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan di dalam daftar pustaka.

Semarang, 17 September 2015



Yeny Rochmawati

24010311120017

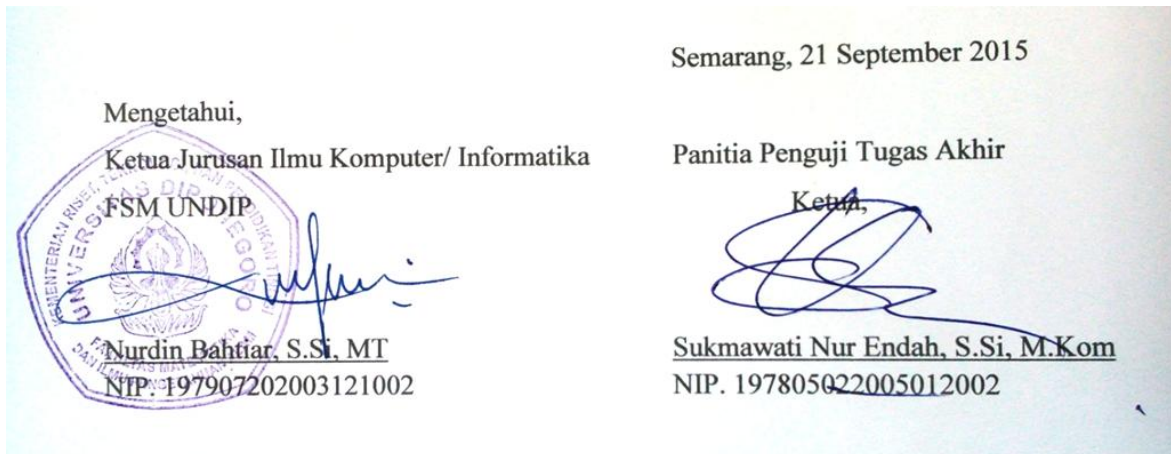
## HALAMAN PENGESAHAN

Judul : Perbandingan Algoritma Pencarian *String* dalam Metode *Approximate String Matching* untuk Identifikasi Kesalahan Pengetikan Teks Bahasa Indonesia dan Saran Perbaikan

Nama : Yeny Rochmawati

NIM : 24010311120017

Telah diujikan pada sidang tugas akhir pada tanggal 11 September 2015 dan dinyatakan lulus pada tanggal 17 September 2015.



## HALAMAN PENGESAHAN

Judul : Perbandingan Algoritma Pencarian *String* dalam Metode *Approximate String Matching* untuk Identifikasi Kesalahan Pengetikan Teks Bahasa Indonesia dan Saran Perbaikan

Nama : Yeny Rochmawati

NIM : 24010311120017

Telah diujikan pada sidang tugas akhir pada tanggal 11 September 2015.

Semarang, 21 September 2015

Pembimbing



Dr. Retno Kusumaningrum, S.Si, M.Kom  
NIP. 198104202005012001

## ABSTRAK

Penggunaan Bahasa Indonesia yang baik dan benar memiliki peran penting khususnya untuk sebuah dokumen ilmiah (laporan, makalah, karya ilmiah, skripsi) dan dokumen resmi yang lain. Kesalahan pengetikan yang sering terjadi menyebabkan kata baku berubah menjadi kata tidak baku karena ejaan yang digunakan tidak sesuai. Proses pengecekan kesalahan pengetikan dengan cara manual akan menghabiskan banyak waktu dan tidak optimal karena memungkinkan adanya *human error*. Hal tersebut dapat ditangani dengan mengembangkan suatu sistem yang mampu mengidentifikasi kesalahan pengetikan agar sesuai dengan kaidah penulisan. Beberapa penelitian yang sudah dikembangkan terkait dengan pengecekan kata berbahasa Indonesia di Indonesia, untuk memberikan saran perbaikan menggunakan algoritma *Levenshtein Distance*. Pada metode *approximate string matching* terdapat empat jenis algoritma yaitu *Levenshtein Distance*, *Hamming Distance*, *Damerau Levenshtein Distance* dan *Jaro Winkler Distance*. Perbandingan kinerja keempat algoritma tersebut untuk pencarian *string* dalam pengecekan kata belum pernah dilakukan. Oleh karena itu, pada penelitian ini akan dilakukan perbandingan antara empat algoritma tersebut sehingga dapat diketahui algoritma mana yang lebih akurat dan tepat dalam pencarian *string* berdasarkan kesalahan penulisan yang bervariasi. Evaluasi yang dilakukan menggunakan *user relevance judgement* yang menghasilkan nilai *mean average precision* (MAP) untuk menentukan algoritma yang terbaik. Berdasarkan hasil penelitian terhadap 50 kata salah menunjukkan bahwa algoritma *Jaro Winkler Distance* terbaik dalam melakukan pengecekan kata dengan nilai MAP sebesar 0.87 yang terbagi dalam empat jenis kesalahan yaitu penghapusan huruf 0.92, penambahan huruf 0.90, penggantian huruf 0.70 dan penukaran huruf 0.95.

**Kata kunci:** Kesalahan pengetikan, *approximate string matching*, *Levenshtein*, *Hamming*, *Damerau Levenshtein*, *Jaro Winkler*

## ABSTRACT

The use of Indonesian language in well written has important roles for scientific writing (reports, papers, scientific papers, theses) and other official documents. Error typing are frequently occurred that cause standard words turn into non standard words default because the spelling used is not appropriate. Manually checking process for error typing spend a lot of time and is not optimal because increasing the risk of human error. This can be addressed by developing a system which capable to identify errors typing and conform with the writing rules. Several studies associated with word-checking in Indonesian language to provide suggestions for improvements used Levenshtein Distance algorithm had been developed. There are four types of algorithm on approximate string matching method, i.e. Levenshtein Distance, Hamming Distance, Damerau Levenshtein Distance and Jaro Winkler Distance. Performance comparison between fourth algorithm for search string in word-checking has not been done. This research will be carried out a comparison between the four algorithms in order to identify which algorithm is more accurate and precise in the search string based on various errors typing. Evaluations were performed using user relevance judgments which produces the mean average precision (MAP) to determine the best algorithm. Based on the results of study for 50 incorrect words showed that Jaro Winkler Distance algorithm is the best in word-checking with 0.87 of MAP value, which is divided into four types of errors 0.92 for deletion letter, 0.90 for adding letter, 0.70 for replacement letter and 0.95 for exchange letters.

**Keywords:** Errors typing, approximate string matching, Levenshtein, Hamming, Damerau Levenshtein, Jaro Winkler

## KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT karena berkat Rahmat dan Hidayah-Nya penulis dapat menyelesaikan laporan tugas akhir yang berjudul “Perbandingan Algoritma Pencarian *String* dalam Metode *Approximate String Matching* untuk Identifikasi Kesalahan Pengetikan Teks Bahasa Indonesia dan Saran Perbaikan” dengan baik dan lancar. Laporan tugas akhir ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata Satu (S1) pada Jurusan Ilmu Komputer/ Informatika Fakultas Sains dan Matematika Universitas Diponegoro Semarang.

Dalam penyusunan laporan tugas akhir ini, penulis banyak mendapat bimbingan, arahan dan bantuan dari berbagai pihak. Oleh karena itu dengan segala kerendahan hati, penulis ingin mengucapkan terima kasih dengan tulus kepada :

1. Prof. Dr. Widowati, M.Si, selaku Dekan FSM UNDIP
2. Nurdin Bahtiar, S.Si, M.T, selaku Ketua Jurusan Ilmu Komputer / Informatika.
3. Indra Waspada, ST, MTI, selaku Koordinator tugas akhir.
4. Dr. Retno Kusumaningrum, S.Si, M.Kom, selaku dosen pembimbing

Penulis menyadari bahwa dalam laporan ini masih banyak terdapat kekurangan baik dari penyampaian materi maupun isi dari materi itu sendiri. Hal ini dikarenakan keterbatasan kemampuan dan pengetahuan dari penulis. Oleh karena itu, kritik dan saran yang bersifat membangun sangat penulis harapkan.

Semoga laporan tugas akhir ini dapat bermanfaat bagi penulis dan juga pembaca pada umumnya.

Semarang, September 2015

Yeny Rochmawati

## DAFTAR ISI

	Hal
HALAMAN PERNYATAAN KEASLIAN SKRIPSI.....	ii
HALAMAN PENGESAHAN .....	iii
HALAMAN PENGESAHAN .....	iv
ABSTRAK .....	v
ABSTRACT .....	vi
KATA PENGANTAR.....	vii
DAFTAR ISI .....	viii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL .....	xiii
DAFTAR KODE .....	xiv
DAFTAR LAMPIRAN .....	xv
DAFTAR SIMBOL .....	xvi
BAB I PENDAHULUAN .....	1
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah.....	3
1.3. Tujuan dan Manfaat .....	3
1.4. Ruang Lingkup .....	3
1.5. Sistematika Penulisan .....	4
BAB II TINJAUAN PUSTAKA .....	6
2.1. Kata Baku dalam Bahasa Indonesia.....	6
2.2. Perkembangan Penelitian Identifikasi Kesalahan Kata .....	6
2.3. Operasi <i>String</i> pada <i>Approximate String Matching</i> .....	9
2.3.1 Algoritma <i>Hamming Distance</i> .....	10
2.3.2 Algoritma <i>Levenshtein Distance</i> .....	12



2.3.3	Algoritma <i>Damerau Levenshtein Distance</i> .....	15
2.3.4	Algoritma <i>Jaro Winkler Distance</i> .....	18
2.4.	Ekstraksi Kata .....	20
2.5.	Evaluasi.....	20
2.5.1	<i>Precision Recall</i> .....	20
2.5.2	<i>Average Precision</i> .....	22
2.5.3	<i>Mean Average Precision</i> .....	22
BAB III METODE PENELITIAN .....		24
3.1.	Pembentukan Kamus .....	24
3.2.	Pengumpulan Abstrak.....	25
3.3.	<i>Preprocessing</i> .....	25
3.4.	Pencocokan Kata Berdasarkan <i>Distance</i> .....	26
3.3.1.	<i>Hamming Distance</i> .....	27
3.3.2.	<i>Levenshtein Distance</i> .....	32
3.3.3.	<i>Damerau Levenshtein Distance</i> .....	38
3.3.4.	<i>Jaro Winkler Distance</i> .....	42
3.3.5.	Pemberian Rekomendasi Saran Kata.....	46
3.5.	Evaluasi.....	48
3.6.	Model Pengembangan Perangkat Lunak .....	50
3.6.1.	Analisis .....	50
3.6.2.	Desain .....	51
3.6.3.	Implementasi .....	56
3.6.4.	Pengujian .....	61
BAB IV HASIL EKSPERIMEN DAN ANALISA.....		63
4.1.	Data.....	63
4.1.1.	Kesalahan karena Penghapusan Huruf .....	63
4.1.2.	Kesalahan karena Penambahan Huruf.....	64

4.1.3.	Kesalahan karena Penggantian Huruf.....	64
4.1.4.	Kesalahan karena Penukaran Huruf .....	65
4.2.	Skenario Eksperimen .....	65
4.2.1.	Eksperimen 1 .....	66
4.2.2.	Eksperimen 2 .....	67
4.3.	Hasil dan Analisa .....	68
4.3.1.	Hasil Skenario Eksperimen 1 dan Analisa .....	68
4.3.2.	Hasil Skenario Eksperimen 2 dan Analisa .....	69
BAB V PENUTUP .....		74
5.1	Kesimpulan .....	74
5.2	Saran .....	74
DAFTAR PUSTAKA.....		75
LAMPIRAN .....		77

## DAFTAR GAMBAR

	Hal
Gambar 2.1. Ilustrasi Operasi Penghapusan.....	9
Gambar 2.2. Ilustrasi Operasi Penyisipan .....	10
Gambar 2.3. Ilustrasi Operasi Penggantian .....	10
Gambar 2.4. Ilustrasi Algoritma <i>Hamming Distance</i> .....	11
Gambar 2.5. Ilustrasi Algoritma <i>Levenshtein Distance</i> .....	14
Gambar 2.6. Ilustrasi Algoritma <i>Damerau Levenshtein Distance</i> .....	16
Gambar 3.1. Blok Proses Garis Besar Penyelesaian Masalah.....	24
Gambar 3.2. Diagram Alir <i>Preprocessing</i> .....	26
Gambar 3.3. Diagram Alir Pencocokan Kata.....	27
Gambar 3.4. Diagram Alir Pencocokan Kata Algoritma <i>Hamming Distance</i> .....	29
Gambar 3.5. Diagram Alir Pencocokan Kata Algoritma <i>Levenshtein Distance</i> .....	34
Gambar 3.6. Diagram Alir Pencocokan Kata Algoritma <i>Damerau Levenshtein Distance</i> .	40
Gambar 3.7. Diagram Alir Pencocokan Kata Algoritma <i>Jaro Winkler Distance</i> .....	45
Gambar 3.8. Diagram Alir Rekomendasi Saran Kata .....	47
Gambar 3.9. DFD Level 0 Aplikasi Perbandingan Algoritma Pencarian <i>String</i> .....	51
Gambar 3.10. DFD Level 1 Aplikasi Perbandingan Algoritma Pencarian <i>String</i> .....	52
Gambar 3.11. Diagram Alir Perancangan Fungsional.....	53
Gambar 3.12. Rancangan Antarmuka Halaman Utama .....	54
Gambar 3.13. Rancangan Antarmuka Halaman Cek Kata .....	55
Gambar 3.14. Rancangan Antarmuka Halaman Saran Kata .....	55
Gambar 3.15. Rancangan Antarmuka Halaman Deskripsi.....	56
Gambar 3.16. Tampilan Halaman Utama.....	57
Gambar 3.17. Tampilan Halaman Cek Kata.....	57
Gambar 3.18. Tampilan Halaman Saran Kata Benar .....	58
Gambar 3.19. Tampilan Halaman Saran Kata dengan Algoritma <i>Hamming Distance</i> .....	58
Gambar 3.20. Tampilan Halaman Saran Kata dengan Algoritma <i>Levenshtein Distance</i> ....	59
Gambar 3.21. Tampilan Halaman Saran Kata dengan Algoritma <i>Damerau Levenshtein Distance</i> .....	59
Gambar 3.22. Tampilan Halaman Saran Kata dengan Algoritma <i>Jaro Winkler Distance</i> ..	60
Gambar 3.23. Tampilan Halaman Deskripsi .....	60

Gambar 4.1. Skenario Eksperimen Penelitian .....	66
Gambar 4.2. Grafik Eksperimen 1 .....	68
Gambar 4.3. Grafik Eksperimen 2.....	72

## DAFTAR TABEL

	Hal
Tabel 2.1. Penelitian Lain Terkait Identifikasi Kesalahan Pengetikan Kata.....	8
Tabel 2.2. Perhitungan <i>Recall</i> dan <i>Precision</i> .....	21
Tabel 2.3. Rekomendasi Kata.....	22
Tabel 2.4. <i>Precision Recall</i> .....	22
Tabel 3.1. Tabel Algoritma <i>Hamming Distance</i> .....	30
Tabel 3.2. Perhitungan <i>Hamming Distance</i> Menggunakan Tabel.....	32
Tabel 3.3. Perhitungan <i>Levenshtein Distance</i> Menggunakan Tabel .....	35
Tabel 3.4. Perhitungan <i>Damerau Levenshtein Distance</i> Menggunakan Tabel .....	41
Tabel 3.5. Sampel Kata yang Salah.....	48
Tabel 3.6. Kebutuhan Fungsional Sistem.....	51
Tabel 3.7. Kebutuhan Non Fungsional Sistem .....	51
Tabel 3.8. Rencana Pengujian Fungsional Sistem.....	61
Tabel 4.1. Kesalahan Penulisan karena Penghapusan Huruf.....	64
Tabel 4.2. Kesalahan Penulisan karena Penambahan Huruf .....	64
Tabel 4.3. Kesalahan Penulisan karena Penggantian Huruf .....	65
Tabel 4.4. Kesalahan Penulisan karena Penukaran Huruf.....	65
Tabel 4.5. Hasil Eksperimen 1.....	68
Tabel 4.6. Hasil MAP Penghapusan Huruf .....	70
Tabel 4.7. Hasil MAP Penambahan Huruf .....	70
Tabel 4.8. Hasil MAP Penggantian Huruf.....	71
Tabel 4.9. Hasil MAP Penukaran Huruf .....	71
Tabel 4.10. Hasil Eksperimen 2.....	72

## DAFTAR KODE

	Hal
Kode 2.1. Algoritma <i>Levenshtein Distance</i> .....	15
Kode 2.2. Algoritma <i>Damerau Levenshtein Distance</i> .....	17

## DAFTAR LAMPIRAN

	Hal
Lampiran 1. Hasil <i>Precision Recall</i> Algoritma <i>Hamming Distance</i> .....	78
Lampiran 2. Hasil <i>Precision Recall</i> Algoritma <i>Levenshtein Distance</i> .....	88
Lampiran 3. Hasil <i>Precision Recall</i> Algoritma <i>Damerau Levenshtein Distance</i> .....	98
Lampiran 4. Hasil <i>Precision Recall</i> Algoritma <i>Jaro Winkler Distance</i> .....	108
Lampiran 5. Hasil <i>Mean Average Precision</i> Eksperimen 1 .....	118
Lampiran 6. Hasil <i>Mean Average Precision</i> Eksperimen 2 .....	122
Lampiran 7. <i>Source code</i> fungsi.....	128
Lampiran 8. Tabel Hasil dan Evaluasi Pengujian .....	134

## DAFTAR SIMBOL

<i>ind_kamus</i>	= indeks kata kamus
<i>num_kamus</i>	= jumlah semua kata dalam kamus
<i>m</i>	= $\text{length}(\text{kata})$
<i>n</i>	= $\text{length}(\text{kata\_kamus})$
<i>length(kata)</i>	= panjang kata
<i>length(kata_kamus)</i>	= panjang kata kamus
<i>i</i>	= indeks huruf kata masukan ( $i = 1, 2, \dots, \text{length}(\text{kata})$ )
<i>j</i>	= indeks huruf kata kamus ( $j = 1, 2, \dots, \text{length}(\text{kata\_kamus})$ )
<i>ham_dist</i>	= <i>hamming distance</i>
<i>distance</i>	= jarak yang diperoleh
<i>suggest</i>	= huruf yang diganti
<i>all_distance</i>	= nilai akhir perhitungan jarak yang diperoleh
<i>all_suggest</i>	= kata saran yang diperoleh
<i>d</i>	= variabel untuk <i>distance</i>
<i>d<sub>1</sub></i>	= simbol untuk operasi <i>substitusi</i>
<i>d<sub>2</sub></i>	= simbol untuk operasi <i>delete</i>
<i>d<sub>3</sub></i>	= simbol untuk operasi <i>insert</i>
<i>s<sub>1</sub></i>	= $\text{length}(\text{kata}) \rightarrow$ panjang kata
<i>s<sub>2</sub></i>	= $\text{length}(\text{kata\_kamus}) \rightarrow$ panjang kata kamus
<i>t</i>	= transposisi (jumlah karakter yang posisinya tertukar)
<i>ℓ</i>	= panjang karakter yang sama sebelum ditemukan ketidaksamaan maksimal 4 ( <i>prefix length</i> )
<i>p</i>	= nilai konstanta <i>scaling factor</i> bernilai 0,1
<i>dj</i>	= nilai <i>jaro distance</i>
<i>dw</i>	= nilai <i>jaro winkler distance</i>
<i>a</i>	= indeks dari x
<i>b</i>	= indeks dari z
<i>S</i>	= <i>string</i> sumber
<i>T</i>	= <i>string</i> target



# BAB I

## PENDAHULUAN

Bab ini menyajikan mengenai latar belakang masalah, rumusan masalah, tujuan dan manfaat, ruang lingkup, dan sistematika penulisan tugas akhir mengenai Perbandingan Algoritma Pencarian *String* dalam Metode *Approximate String Matching* untuk Identifikasi Kesalahan Pengetikan Teks Bahasa Indonesia dan Saran Perbaikan.

### 1.1. Latar Belakang

Bahasa Indonesia sangat penting peranannya bagi kehidupan bermasyarakat, berbangsa, dan bernegara. Selain itu bahasa Indonesia juga memiliki peranan yang penting dalam dunia pendidikan dan pengajaran. Penggunaan bahasa Indonesia yang baik dan benar bertujuan untuk mengembangkan kemampuan menggunakan bahasa Indonesia dalam segala fungsinya, yaitu sarana berkomunikasi, sarana berpikir, sarana persatuan dan sarana kebudayaan. Penggunaan bahasa Indonesia yang baik dan benar nantinya akan digunakan untuk beberapa aspek keterampilan meliputi menyimak, berbicara, membaca dan menulis. Keempat aspek keterampilan tersebut saling berkaitan satu sama lain dengan cara yang beraneka ragam.

Keterampilan menulis merupakan keterampilan yang penting dalam kehidupan, baik dalam kehidupan pendidikan maupun masyarakat. Menulis adalah kegiatan menyampaikan gagasan yang tidak dapat secara langsung diterima dan direaksi oleh pihak yang dituju. Aktivitas menulis merupakan salah satu manifestasi kemampuan dan keterampilan berbahasa paling akhir yang dikuasai pengguna bahasa setelah mendengarkan, membaca, dan berbicara (Nurgiyantoro, 2010).

Terdapat kaidah penulisan yang perlu diperhatikan seperti penggunaan kata baku dan tidak baku. Khususnya untuk sebuah dokumen ilmiah (laporan, makalah, karya ilmiah, skripsi) dan dokumen resmi yang lain harus sesuai dengan EYD yang telah ditetapkan. Kesalahan pengetikan yang sering terjadi dapat menyebabkan kata baku berubah menjadi kata tidak baku karena ejaan yang digunakan tidak sesuai. Beberapa faktor yang menyebabkan terjadinya kesalahan pengetikan adalah letak huruf pada *keyboard* yang berdekatan, kesalahan karena *slip* pada tangan atau jari, kesalahan yang disebabkan oleh ketidaksengajaan.

Proses pengecekan kesalahan pengetikan dengan cara manual akan menghabiskan banyak waktu dan membutuhkan suatu sumber pasti sebagai acuan bahwa kata tersebut salah dalam proses penulisannya. Efisiensi waktu yang dibutuhkan untuk pengecekan manual tidak optimal karena memungkinkan adanya *human error*. Oleh karena itu diperlukan suatu sistem yang mampu mengidentifikasi kesalahan pengetikan agar sesuai dengan kaidah penulisan EYD yang benar. Sehingga kesalahan-kesalahan penulisan dapat dihindari.

Salah satu pendekatan yang dapat dilakukan untuk mengidentifikasi kesalahan pengetikan adalah metode pencocokan string. Pencocokan *string* (*string matching*) menurut *Dictionary of Algorithms and Data Structures, National Institute of Standards and Technology (NIST)*, diartikan sebagai sebuah permasalahan untuk menemukan pola susunan karakter *string* di dalam *string* lain atau bagian dari isi teks. Pencocokan *string* merupakan bagian penting dari sebuah proses pencarian *string* (*string searching*) dalam sebuah dokumen. Hasil dari pencarian sebuah *string* dalam dokumen tergantung dari teknik atau cara pencocokan *string* yang digunakan.

Pencocokan *string* secara garis besar dibedakan menjadi dua yaitu *exact string matching* (pencocokan *string* secara tepat dengan susunan karakter dalam *string* yang dicocokkan) dan *inexact string matching* (pencocokan *string* secara samar, yaitu pencocokan *string* dimana *string* yang dicocokkan memiliki kemiripan namun keduanya memiliki susunan karakter yang berbeda) (Syaroni & Munir, 2005). Dua pendekatan *inexact string matching* meliputi *approximate string matching* yang mencocokkan *string* berdasarkan kemiripan penulisan dan *phonetic string matching* yang mencocokkan *string* berdasarkan kemiripan ucapan. *Approximate string matching* dapat digunakan untuk pencarian *string* berdasarkan *string* yang sama dan *string* yang memiliki kemiripan penulisan dengan *string* yang terdapat pada kamus. Metode ini dapat digunakan untuk pencarian kata tidak baku karena dapat mengidentifikasi *string* yang sama dan yang memiliki kemiripan penulisan.

Beberapa algoritma yang dapat digunakan untuk pencarian *string* dalam *approximate string matching* diantaranya adalah *Hamming Distance*, *Levenshtein Distance*, *Damerau Levenshtein Distance* (pengembangan dari *Levenshtein Distance*) dan *Jaro Winkler Distance*. Algoritma tersebut memiliki rumus masing-masing yang digunakan untuk memproses fungsi pencarian *string* dengan tingkat keakuratan yang berbeda serta waktu yang berbeda.

Oleh karena itu pada penelitian ini digunakan metode *approximate string matching* dengan melakukan perbandingan antara empat algoritma pencarian *string* yang ada di dalamnya dimana teks atau kalimat yang akan dicek diproses sehingga mengeluarkan hasil berupa kata mana saja yang salah dan saran perbaikan kata baku yang sesuai sehingga dapat diketahui algoritma mana yang lebih akurat dan tepat dalam pencarian *string* berdasarkan kesalahan penulisan yang bervariasi.

## **1.2. Rumusan Masalah**

Berdasarkan uraian latar belakang tersebut maka dapat dibuat rumusan masalah yaitu bagaimana perbandingan algoritma *Hamming Distance*, *Levenshtein Distance*, *Damerau Levenshtein Distance* dan *Jaro Winkler Distance* sebagai algoritma pencarian *string* pada metode *approximate string matching* untuk identifikasi kesalahan pengetikan teks bahasa Indonesia dan saran perbaikan serta menentukan algoritma yang terbaik dari keempat algoritma yang digunakan?

## **1.3. Tujuan dan Manfaat**

Tujuan yang ingin dicapai dari penelitian ini adalah membandingkan algoritma pencarian *string* yaitu algoritma *Hamming Distance*, *Levenshtein Distance*, *Damerau Levenshtein Distance* dan *Jaro Winkler Distance* sebagai algoritma pencarian *string* pada metode *approximate string matching* untuk identifikasi kesalahan pengetikan teks bahasa Indonesia dan saran perbaikan sehingga menghasilkan algoritma mana yang terbaik.

Adapun manfaat yang diharapkan dari penelitian ini adalah memberikan hasil algoritma yang terbaik untuk identifikasi kesalahan penulisan sehingga dapat digunakan sebagai acuan tentang algoritma apa yang akan dipakai kedepannya untuk mengidentifikasi kesalahan pengetikan dalam teks bahasa Indonesia untuk dokumen ilmiah yang sesuai dengan EYD.

## **1.4. Ruang Lingkup**

Batasan-batasan dalam Tugas Akhir ini diperlukan agar dalam pengerjaannya tidak melebihi target yang akan diteliti. Adapun ruang lingkup dalam pembuatan aplikasi ini adalah sebagai berikut:

1. *Input* data berupa teks dalam format xml dengan sampel data berupa abstrak artikel ilmiah dengan maksimal jumlah kata sebanyak 200 kata.
2. *Output* berupa penemuan kesalahan pengetikan dalam teks bahasa Indonesia dan saran perbaikan yang sesuai dengan kamus sebanyak sepuluh kata yang mirip pada masing-masing algoritma tanpa mengganti secara otomatis kata yang salah dengan saran perbaikannya.
3. Berbasis web menggunakan bahasa pemrograman PHP.
4. Aplikasi ini dievaluasi oleh dua *user relevance judgement* untuk mengetahui saran perbaikan yang relevan.

### 1.5. Sistematika Penulisan

Sistematika penulisan yang digunakan dalam tugas akhir ini terbagi dalam beberapa pokok bahasan, yaitu:

#### BAB I PENDAHULUAN

Bab ini memberikan gambaran mengenai latar belakang masalah, rumusan masalah, tujuan dan manfaat, ruang lingkup serta sistematika penulisan tugas akhir mengenai Perbandingan Algoritma Pencarian *String* dalam Metode *Approximate String Matching* untuk Identifikasi Kesalahan Pengetikan Teks Bahasa Indonesia dan Saran Perbaikan.

#### BAB II TINJAUAN PUSTAKA

Bab ini memberikan kajian pustaka yang berhubungan dengan tema tugas akhir sebagai landasan untuk perumusan dan analisis permasalahan pada tugas akhir. Bab ini menyajikan kata baku dalam Bahasa Indonesia, perkembangan penelitian mengenai identifikasi kesalahan pengetikan kata, operasi *string* pada *approximate string matching* yang digunakan dalam masing-masing algoritma, penjelasan algoritma untuk penentuan *distance*, penjelasan ekstraksi kata serta penjelasan evaluasi yang digunakan.

#### BAB III METODE PENELITIAN

Bab ini menyajikan garis besar penyelesaian masalah tugas akhir yang diawali dengan penyajian garis besar penyelesaian masalah dalam bentuk blok proses. Garis besar penyelesaian masalah diawali dengan

pembentukan kamus, memasukkan teks dalam format xml sebagai sampel data abstrak artikel ilmiah, *preprocessing* berupa penghilangan tanda baca atau simbol dan tokenisasi, pencocokan kata dengan metode *approximate string matching* beserta pemberian rekomendasi saran kata, pengujian dan proses evaluasi masing-masing algoritma, model pengembangan perangkat lunak.

#### BAB IV HASIL EKSPERIMEN DAN ANALISA

Bab ini menguraikan hasil eksperimen dan analisa pada penelitian dimulai dari definisi data dan koleksi kata yang salah, penjelasan mengenai semua skenario eksperimen dan analisa dari setiap eksperimen yang telah dilakukan.

#### BAB V PENUTUP

Bab ini menjabarkan kesimpulan dari uraian yang telah diulas pada bab-bab sebelumnya dan saran untuk pengembangan penelitian lebih lanjut.