

**ALGORITMA SPELLCHECKING TERNARY SEARCH TREE
DAN LEVENSHTEIN DISTANCE DALAM PENGOREKSIAN
KATA BERBAHASA INGGRIS**

Diajukan Sebagai Syarat untuk Menyelesaikan
Pendidikan Program Strata-1 pada
Jurusan Teknik Informatika Fakultas Ilmu Komputer UNSRI



Oleh

VALLERY MEDISTA RIZKY

NIM : 09021381320022

**Jurusan Teknik Informatika
FAKULTAS ILMU KOMPUTER UNIVERSITAS SRIWIJAYA
2018**

LEMBAR PENGESAHAN TUGAS AKHIR

**ALGORITMA SPELLCHECKING TERNARY SEARCH TREE DAN
LEVENSHTEIN DISTANCE DALAM PENGOREKSIAN KATA
BERBAHASA INGGRIS**

Oleh

Vallery Medista Rizky

NIM : 09021381320022

Palembang, 21 Desember 2018

Pembimbing I



Rifkie Primartha, M.T.
NIP. 197706012009121004

Pembimbing II



Kanda Januar Miraswan, M.T.
NIK. 1671080901900006

Mengetahui,

Ketua Jurusan Teknik Informatika



TANDA LULUS UJIAN TUGAS AKHIR

Pada hari Senin tanggal 17 Desember 2018 telah dilaksanakan ujian sidang tugas akhir oleh Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Sriwijaya.

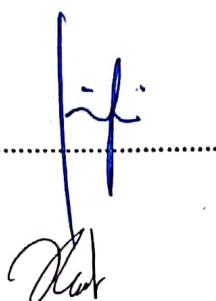
Nama : Vallery Medista Rizky

NIM : 09021381320022

Judul : Algoritma Spellchecking *Ternary Search Tree* dan *Levenshtein Distance* Dalam Pengoreksian Kata Berbahasa Inggris

1. Ketua Pengaji

Rifkie Primartha, S.T, M.T.
NIP. 197706012009121004



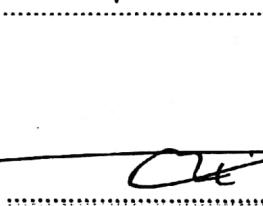
2. Sekretaris

Kanda Januar Miraswan, S.Kom., MT.
NIK. 1671080901900006



3. Pengaji I

Yunita, M.Sc
NIP. 198306062015042002



4. Pengaji II

Osvari Arsalan, M.T.
NIP. 198806282018031001

Mengetahui,

Ketua Jurusan Teknik Informatika



HALAMAN PERNYATAAN

Yang bertanda tangan di bawah ini :

Nama : Vallery Medista Rizky
NIM : 09021381320022
Program Studi : Teknik Informatika
Judul Skripsi : Algoritma *Spellchecking Ternary Search Tree* dan *Levenshtein Distance* dalam Pengoreksian Kata Berbahasa Inggris
Hasil Pengecekan Software *iThenticate/Turnitin* : 18 %

Menyatakan bahwa Laporan Projek saya merupakan hasil karya sendiri dan bukan hasil penjiplakan/plagiat. Apabila ditemukan unsur penjiplakan/plagiat dalam laporan projek ini, maka saya bersedia menerima sanksi akademik dari Universitas Sriwijaya sesuai dengan ketentuan yang berlaku.

Demikian, pernyataan ini saya buat dengan sebenarnya dan tidak ada paksaan oleh siapapun.

Palembang, 21 Desember2018



(Vallery Medista Rizky)
NIM. 09021381320022

Motto :

“Barang siapa keluar untuk mencari ilmu maka dia berada di jalan Allah “
(HR.Tirmidzi)

“You cannot escape your responsibility by evading it today”
(Abraham Lincoln)

It's fine to celebrate success but it is more important to learn the lessons of failure.
(Bill Gates)

Kupersembahkan karya tulis ini kepada :

- **Allah SWT**
- **Kedua orangtua ku Bapak Drs.H.M.Hatta Zainal dan Ibu Hj.Desty Eryani yang kusayangi.**
- **Adik ku Gyzella Sandra Clarissa**
- **Almamaterku**
- **IF Bilingual angkatan 2013**

SPELLCHECKING TERNARY SEARCH TREE AND LEVENSHTEIN DISTANCE ALGORITHM SPELLING CORRECTION IN ENGLISH

VALLERY MEDISTA RIZKY

09021381320022

ABSTRACT

Errors in spelling or writing words in a text processing software are things that often occur, usually caused by user negligence. So from that it requires a software that can help correct the spelling of words, in this study combines two algorithmic methods, software will match input words with the closest words in the word dictionary using the Ternary Search Tree algorithm method, then perform the string search process and calculate the closest Edit distance using the Levenshtein Distance algorithm. Testing is done with 5 test configurations. With 5 different edit distance values. The highest accuracy test results are 98% in the 5th test configuration, and the fastest processing time is recorded by the configuration of the 1st experiment with an average processing time of 2 milliseconds. From the test results it can be seen, the greater the edit distance value used, will increase value of accuracy, but requires a longer processing time too, and vice versa, the smaller the edit distance value used, will increase the processing speed, but has an impact on the reduced level of accuracy.

Key Word : *Spellchecking, Ternary search tree, Spelling correction, Levenshtein distance*

SPELLCHECKING TERNARY SEARCH TREE AND LEVENSHTEIN DISTANCE ALGORITHM SPELLING CORRECTION IN ENGLISH

VALLERY MEDISTA RIZKY

09021381320022

ABSTRAK

Kesalahan dalam ejaan atau penulisan kata dalam sebuah perangkat lunak pengolahan teks adalah hal yang seringkali terjadi, biasanya disebabkan oleh kelalaian pengguna. Maka dari itu diperlukan sebuah perangkat lunak yang dapat membantu untuk mengoreksi ejaan kata, di dalam penelitian ini menggabungkan dua metode algoritma, perangkat lunak akan melakukan pencocokan kata inputan dengan kata terdekat di dalam kamus kata menggunakan metode algoritma *Ternary Search Tree*, kemudian melakukan proses pencarian string dan menghitung *Edit distance* yang terdekat menggunakan algoritma *Levenshtein Distance*. Pengujian dilakukan dengan 5 konfigurasi pengujian. Dengan 5 nilai *edit distance* yang berbeda. Hasil pengujian akurasi tertinggi sebesar 98% pada konfigurasi pengujian ke 5, dan waktu pemrosesan tercepat dicatatkan oleh konfigurasi percobaan ke 1 dengan rata-rata waktu pemrosesan 2 milidetik. Dari hasil pengujian dapat terlihat, Semakin besar *edit distance value* yang digunakan, akan berdampak pada meningkatnya nilai akurasi, namun membutuhkan waktu pemrosesan yang lebih lama pula, dan begitu juga sebaliknya, semakin kecil nilai *edit distance* yang digunakan, akan meningkatkan kecepatan pemrosesan, namun berdampak pada berkurangnya tingkat akurasi.

Kata Kunci: *Spellchecking, Ternary search tree, Spelling correction, Levenshtein distance*

KATA PENGANTAR

Puji dan syukur kehadirat Allah SWT karena atas rahmat-Nya penulis dapat menyelesaikan tugas akhir ini. Tugas akhir yang berjudul **“ALGORITMA SPELLCHECKING TERNARY SEARCH TREE DAN LEVENSHTEIN DISTANCE DALAM PENGOREKSIAN KATA BERBAHASA INGGRIS”** ini disusun untuk memenuhi salah satu persyaratan kelulusan tingkat S1 pada Jurusan Teknik Informatika Universitas Sriwijaya.

Pada kesempatan ini, penulis ingin menyampaikan ucapan terima kasih yang tak terhingga kepada pihak-pihak telah memberikan dukungan, bimbingan, motivasi dan kemauan kepada penulis untuk menyelesaikan tugas akhir ini, yaitu kepada:

1. Orang tua ku Bapak Drs.H.M.Hatta Zainal dan Ibu Hj.Desty Eryani dan adik ku Gyzella Sandra Clarissa dan seluruh keluarga atas semua bantuan, dukungan serta doanya.
2. Bapak Jaidan Jauhari, S. Pd. M.T.selaku Dekan Fakultas Ilmu Komputer Universitas Sriwijaya;
3. Bapak Rifkie Primartha, S,T. M.T. selaku Ketua Jurusan Teknik Informatika Fakultas ilmu Komputer Universitas Sriwijaya;
4. Bapak Rifkie Primartha, S.T. M.T. selaku pembimbing 1 yang telah membimbing dan membantu penulis;
5. Bapak Kanda Januar Miraswan,S.Kom.,MT, selaku pembimbing 2 yang telah membimbing dan membantu penulis;
6. Ibu Yunita, M.Sc dan Bapak Osvari Arsalan, M.T selaku penguji;
7. Bapak dan Ibu Dosen yang selama ini telah melimpahkan ilmunya kepada penulis selama proses belajar mengajar di Fakultas Ilmu Komputer Universitas Sriwijaya.

8. Teman-teman seperjuangan, Adrian Hwang Winata, Fuad Wahid, Rizqi Herlangga, M.Rizki Aiman, Syardinal, Eva sapayona, serta Roberto Derman Daeli yang merantau ke Jakarta, Terima kasih untuk persahabatan dan masa-masa perkuliahan yang kita jalani bersama baik dalam suka maupun duka, semoga menjadi pengalaman hidup yang tidak terlupakan
9. Semua teman-teman Teknik Informatika Bilingual angkatan 2013,;
10. Untuk semua pihak yang telah membantu penyelesaian tugas akhir ini dan tidak dapat disebutkan satu persatu.

Akhir kata, penulis menyadari bahwa tugas akhir ini jauh dari kata sempurna. Untuk itu penulis mengharapkan kritik dan saran yang membangun dari semua pihak untuk penyempurnaan tugas akhir ini dan semoga tugas akhir ini dapat bermanfaat.

Palembang, 21 Desember 2018

Penulis,

DAFTAR ISI

	Halaman
ABSTRAK.....	i
HALAMAN PENGESAHAN.....	ii
DAFTAR ISI	iii
DAFTAR GAMBAR	vi
DAFTAR TABEL	viii
BAB I PENDAHULUAN.....	I-1
1.1 Pendahuluan	I-1
1.2 Latar Belakang	I-2
1.3 Rumusan Masalah	I-3
1.4 Tujuan Penelitian	I-3
1.5 Manfaat Penelitian	I-3
1.6 Batasan Masalah.....	I-4
1.7 Sistematika penulisan.....	I-4
1.8 Kesimpulan	I-5
BAB II TINJAUAN PUSTAKA.....	II-1
2.1 Pendahuluan	II-1
2.2 Landasan Teori.....	II-1
2.2.1 <i>Spelling Correction</i>	II-2
2.2.2 <i>Ternary Search Tree</i>	II-2
2.2.3 <i>Approximate String Matching</i>	II-4
2.2.4 <i>Levenshtein Distance</i>	II-4
2.2.5 <i>Rational Unified Process</i>	II-6
2.3 Penelitian Lain yang Relevan	II-8
2.4 Kesimpulan.....	II-10

BAB III METODOLOGI PENELITIAN.....	III-1
3.1 Pendahuluan.....	III-1
3.2 Unit Penelitian.....	III-1
3.3 Metode Pengumpulan Data.....	III-1
3.3.1 Jenis Data.....	III-1
3.3.1 Sumber Data.....	III-2
3.3.3 Teknik Pengumpulan Data.....	III-2
3.4 Tahapan Penelitian.....	III-3
3.5 Rencana Pengujian Perangkat Lunak.....	III-4
3.6 Metode Pengembangan Perangkat Lunak.....	III-5
3.7 Penjadwalan Penelitian.....	III-10
BAB IV PENGEMBANGAN PERANGKAT LUNAK.....	IV-1
4.1 Pendahuluan.....	IV-1
4.2 Fase Insepsi.....	IV-1
4.2.1 Analisis Masalah.....	IV-1
4.2.2 Analisis Data.....	IV-2
4.2.3 Analisis <i>Levenshtein distance</i>	IV-3
4.2.4 Analisis <i>Ternary Search Tree</i>	IV-5
4.2.5 Deskripsi Umum Sistem.....	IV-7
4.2.6 Analisis Kebutuhan Perangkat Lunak.....	IV-9
4.3 Perancangan Perangkat Lunak.....	IV-10
4.3.1 Model <i>Use Case</i>	IV-10
4.3.1.1 Diagram <i>Use Case</i>	IV-11
4.3.1.2 Tabel Definisi Aktor.....	IV-11
4.3.1.3 Tabel Definisi <i>Use Case</i>	IV-11

4.3.1.4 Skenario <i>Use Case</i>	IV-12
4.4 Fase Elaborasi.....	IV-13
4.4.1 Perancangan Antar Muka.....	IV-13
. 4.4.2 Kebutuhan Sistem.....	IV-17
4.4.3 Sequence Diagram.....	IV-18
4.5 Fase Konstruksi.....	IV-19
. 4.5.1 Kelas Analisis.....	IV-19
4.5.2 <i>Class Diagram</i>	IV-20
4.5.3 implementasi Kelas.....	IV-21
4.6 Fase Transisi.....	IV-22
4.6.1 Kebutuhan sistem.....	IV-22
4.6.2 Lingkungan Pengujian.....	IV-23
4.6.4 Rencana Pengujian.....	IV-23
4.6.5 Kasus Uji,.....	IV-23
BAB V HASIL DAN ANALISIS PENELITIAN.....	V-1
5.1 Pendahuluan.....	V-1
5.2 Data Hasil Pengujian.....	V-1
5.2.1 Konfigurasi Percobaan 1.....	V-2
5.2.2 Konfigurasi Percobaan 2.....	V-3
5.2.3 Konfigurasi Percobaan 3.....	V-4
5.2.4 Konfigurasi Percobaan 4.....	V-5
5.2.5 Konfigurasi Percobaan 5.....	V-6
5.2.6 Perbandingan Hasil Pengujian.....	V-7
5.3 Analisis Hasil Pengujian.....	V-8

BAB VI KESIMPULAN DAN SARAN.....	VI-1
6.1Kesimpulan.....	VI-1
6.2 Saran.....	VI-2

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

Halaman

Gambar 2.1 <i>Ternary Search tree</i>	II-3
Gambar 2.2 Arsitektur <i>RUP (Rational Unified Process)</i>	II-7
Gambar 4.1 Langkah awal metode <i>Levenshtein Distance</i>	IV-4
Gambar 4.2 Proses pencocokan huruf ‘B’ dari kamus dan masukan.....	IV-4
Gambar 4.3 Proses pencocokan huruf ‘B’ dari kamus dan huruf.....	IV-4
Gambar 4.4 Proses pencocokan huruf ‘B’ dari kamus dan huruf ‘X’	IV-5
Gambar 4.5 Hasil pencocokan kata.....	IV-5
Gambar 4.5. Ternary Search Tree.....	IV-6
Gambar 4.7 Diagram alur perangkat lunak.....	IV-8
Gambar 4.8 Diagram <i>Use Case</i>	IV-11
Gambar 4.9 Antarmuka Perangkat Lunak Sebelum melakukan proses.....	IV-14
Gambar 4.10 Antarmuka Perangkat Lunak Saat memilih <i>edit distance</i>	IV-15
Gambar 4.11 Antarmuka Perangkat Lunak menampilkan Suggestion list... ..	IV-16
Gambar 4.12 Sequence Diagram.....	IV-18
Gambar 4.13 <i>Sequence Diagram button edit distance</i>	IV-19
Gambar 4.14 Kelas Analisis Pengoreksian kata.....	IV-20
Gambar 4.15 Diagram Kelas.....	IV-21
Gambar 5.1 Grafik Perbandingan Hasil Percobaan I, II,III,IV dan V	V-8

DAFTAR TABEL

	Halaman
Tabel 2.1 Kata Yang Salah dan Kemungkinan Kata	II-1
Tabel 2.2 Matriks Pencarian nilai jarak.....	II-6
Tabel 3.1 Langkah-langkah RUP.....	III-6
Tabel 3.2 Penjadwalan penelitian	III-11
Tabel 4.1 Kebutuhan fungsional	IV-9
Tabel 4.2 Kebutuhan non fungsional perangkatlunak.....	IV-9
Tabel 4.3.1.2 Definisi aktor.....	IV-9
Tabel 4.3.1.3 Definisi <i>use case</i>	IV-11
Tabel 4.3.1.4 Skenario <i>use case</i>	IV-12
Tabel 4.4 Implementasi Kelas.....	IV-22
Tabel 4.5 Rencana Pengujian <i>Use Case</i> Melakukan Pengoreksian kata	IV-24
Tabel 4.6 Rencana Pengujian <i>Use Case</i> Melakukan Pengoreksian kata ,.....	IV-25

BAB 1

PENDAHULUAN

1.1 Pendahuluan

Kesalahan dalam ejaan atau penulisan kata dalam pengolahan teks adalah hal yang seringkali terjadi dikarenakan kelalaian pengguna. Maka dari itu diperlukan sebuah perangkat lunak yang dapat menjadi sebuah solusi dan membantu dalam mengoreksi ejaan kata.

Dalam bab ini akan menjelaskan mengenai latar belakang, perumusan masalah, tujuan dan manfaat penelitian, batasan masalah/ruang lingkup, metodologi penelitian, serta sistematika penulisan.

1.2 Latar Belakang

Sebuah *error* atau kesalahan dalam ejaan atau penulisan kata berbahasa Inggris di dalam pengolahan teks adalah hal yang tidak jarang terjadi, seringkali disebabkan oleh kecerobohan pengguna, serta kurangnya penggunaan bahasa Inggris dalam kehidupan sehari-hari di Indonesia, oleh karena itu diperlukan sebuah perangkat lunak yang dapat membantu untuk mengoreksi ejaan kata berbahasa Inggris, saat perangkat lunak mendeteksi kata berbahasa Inggris yang salah eja, maka perangkat lunak akan melakukan pencocokan kata inputan dengan kata terdekat di dalam kamus kata.

Penelitian sebelumnya yang dilakukan oleh (Martins & Silva, 2004) menggunakan metode struktur data *Ternary search tree* untuk menyimpan memori dari kamus kata beserta frekuensi kemunculannya, yang digunakan dalam memilih kata yang paling mendekati dengan kata salah eja yang di inputkan, menghasilkan akurasi sebesar 48% kata yang tepat terhadap data uji.

Dalam penelitian yang dilakukan (Norvig,2016), dengan menggunakan *Bayes probability theorem*, mendapatkan hasil penelitian yang menghasilkan akurasi 68% dengan kecepatan pemrosesan 35 kata/detik terhadap variasi kata yang diujikan.

Menurut penelitian yang dilakukan oleh (Rokhmah & Suprapto, 2012) mengatakan tingkat kemiripan kata menggunakan algoritma *Levenshtein Distance* memiliki akurasi 77,35% terhadap banyak variasi kata yang salah. Berdasarkan pertimbangan diatas maka dipilihlah algoritma *Levensthein Distance* untuk digunakan dalam penelitian ini karena algoritma ini memiliki tingkat akurasi pencarian kemiripan kata yang cukup tinggi.

Dalam penelitian ini, penulis berharap dapat meningkatkan performa kecepatan pemrosesan ejaan kata dengan menggabungkan metode *TST* dan *Levenshtein*, yang diharapkan dapat menjadi kontribusi positif dalam mengembangkan metode pengoreksian kata.

1.3 Rumusan Masalah

Rumusan masalah pada penelitian ini adalah dapatkah kombinasi algoritma *Ternary Search Tree* dan *Levensthein Distance* meningkatkan akurasi dan kecepatan proses pengoreksian ejaan kata pada aplikasi pengoreksian ejaan kata berbahasa Inggris.

1.4 Tujuan Penelitian

1. Mengetahui seberapa besar peningkatan performa kecepatan pemrosesan koreksi ejaan kata beserta tingkat akurasi nya dengan Algoritma *Ternary Search Tree* dan *Levensthein Distance* pada aplikasi pengoreksian ejaan kata berbahasa Inggris.

1.5 Manfaat Penelitian

1. Memudahkan proses mengolah sebuah naskah teks, dan menghasilkan penulisan yang sesuai dengan penulisan Bahasa Inggris yang baik dan benar.
2. Sebagai alternatif perangkat lunak yang membantu mengurangi kesalahan penulisan yang terjadi akibat *human error*.
3. Dapat menjadi acuan dan menghadirkan kontribusi positif dalam penelitian selanjutnya.

1.6 Batasan Masalah

1. Data yang *diinputkan* berupa teks berbahasa Inggris.
2. Teks yang *diinputkan* berupa karakter *alphabet* a-z.
3. Masukkan berupa *input* kata dari keyboard.
4. Perbaikan kata dilakukan secara manual dengan cara pengguna memilih kata yang benar pada *Suggestion List*.

1.7 Sistematika Penulisan

Sistematika penulisan skripsi ini adalah sebagai berikut :

BAB I. PENDAHULUAN

Pada bab ini diuraikan mengenai latar belakang, perumusan masalah, tujuan dan manfaat penelitian, batasan masalah/ruang lingkup, metodologi penelitian, dan sistematika penulisan.

BAB II. KAJIAN LITERATUR

Pada bab ini akan dibahas dasar-dasar teori yang digunakan dalam penelitian, seperti definisi-definisi sistem, metode *Ternary Search Tree*, *Levenshtein distance*, software yang akan digunakan, tahap pengembangan perangkat lunak, desain model, serta bahasa pemrograman *Java*

BAB III. METODOLOGI PENELITIAN

Pada bab ini akan dibahas mengenai tahapan yang akan dilaksanakan pada penelitian ini. Masing-masing rencana tahapan penelitian dideskripsikan dengan rinci dengan mengacu pada suatu kerangka kerja. Di akhir bab ini berisi perancangan manajemen proyek pada pelaksanaan penelitian.

1.8 Kesimpulan

Dalam penelitian ini, penulis berharap dapat meningkatkan performa kecepatan pemrosesan ejaan kata beserta akurasinya dengan menggabungkan metode *Ternary Search Tree* dan *Levenshtein*, yang diharapkan dapat memberikan kontribusi positif dalam mengembangkan metode pengoreksian kata.

DAFTAR PUSTAKA

- Bentley, J., & Sedgewick, R. (1998). *Ternary Search Trees*. *Dr.Dobb's Journal*, 23(4):20-25, April 1998.
- Fachrurrozi, M., & Manik, A. A. (n.d.). Perbaikan Ejaan Kata pada Dokumen Bahasa Indonesia dengan Metode Cosine Similarity.
- Jadhav, S. A., Somayajulu, D. V. L. N., Bhattu, S. N., Subramanyam, R. B. V, & Suresh, P. (2013). Topic dependent cross-word Spelling Corrections for Web Sentiment Analysis. *Proceedings of the 2013 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2013*, 1093–1096. <http://doi.org/10.1109/ICACCI.2013.6637329>.
- Kruchten, P. (2003). *The Rational Unified Process An Introduction*. <http://doi.org/10.1109/ICSE.2002.146346>
- Kilinc, D. (2016). An accurate toponym-matching measure based on approximate string matching. *Journal of Information Science*, 42(2), 138–149. <http://doi.org/10.1177/0165551515590097>
- Lhoussain, A. S., Hicham, G., & Abdellah, Y. (2015). Adaptating the levenshtein distance to contextual spelling correction. *International Journal of Computer Science and Applications*, 12(1), 127–133.

Madaharsa, B., & Adiwidya, D. (2009). Algoritma Levenshtein Dalam Pendekatan Approximate.

Martin, B., & Silva,J. (2004). *Spelling Correction for Search Engine Queries*, 2004, Departamento de Informática Faculdade de Ciências da Universidade de Lisboa 1749-016 Lisboa, Portugal.

bmartins@xldb.di.fc.ul.pt, mjs@di.fc.ul.pt

Norvig,P.(2016).*How to write a Spelling Corrector.*. August 2016
<https://norvig.com/spell-correct.html>

P. Dewi Rokhmah, Suprapto, A. S. (2012). PROGRAM APLIKASI EDITOR KATA BAHASA INDONESIA MENGGUNAKAN METODE APPROXIMATE STRING MATCHING DENGAN ALGORITMA LEVENSHTEIN DISTANCE BERBASIS JAVA.