APLIKASI PERBAIKAN EJAAN PADA KARYA TULIS ILMIAH DI PROGRAM STUDI TEKNIK INFORMATIKA DENGAN MENERAPKAN ALGORITMA *LEVENSHTEIN DISTANCE*

SKRIPSI

Diajukan Untuk Penulisan Skripsi Guna Memenuhi Salah Satu Syarat

Memperoleh Gelar Sarjana Komputer (S.Kom)

Pada Program Studi Teknik Informatika

UN PGRI Kediri

Disusun Oleh :

ROBY NUR HAMZAH

NPM: 12.1.03.02.0196

FAKULTAS TEKNIK (FT)

UNIVERSITAS NUSANTARA PERSATUAN GURU REPUBLIK INDONESIA

UN PGRI KEDIRI

2016

Skripsi oleh:

ROBY NUR HAMZAH

NPM: 12.1.03.02.0196

Judul:

APLIKASI PERBAIKAN EJAAN PADA KARYA TULIS ILMIAH DI PROGRAM STUDI TEKNIK INFORMATIKA DENGAN MENERAPKAN ALGORITMA *LEVENSHTEIN DISTANCE*

Telah disetujui untuk diajukan kepada

Panitia Ujian/ Sidang Program Studi Teknik Informatika

FT UN PGRI Kediri

Tanggal: 25 Juli 2016

Pembimbing I	Pembimbing II
Resty Wulanningrum, M.Kom	Patmi Kasih, M.Kom
NIDN. 0719068702	NIDN. 0701107802

Skripsi oleh:

ROBY NUR HAMZAH

NPM: 12.1.03.02.0196

Judul:

APLIKASI PERBAIKAN EJAAN PADA KARYA TULIS ILMIAH DI PROGRAM STUDI TEKNIK INFORMATIKA DENGAN MENERAPKAN ALGORITMA *LEVENSHTEIN DISTANCE*

Telah dipertahankan di depan Panitia Ujian/ Sidang Skripsi
Program Studi Teknik Informatika UN PGRI Kediri
Pada Tanggal: 12 Agustus 2016

Dan dinyatakan telah memenuhi syarat

Panitia Penguji:

1. Ketua : Resty Wulanningrum, M.Kom. _____
2. Penguji I : Irwan Setyowidodo, S.Pd., M.Si. _____
3. Penguji II : Patmi Kasih, M.Kom. _____

Mengetahui,
Dekan FT

Dr. SURYO WIDODO, M.Pd
NIP. 19640202 199103 1 002

APLIKASI PERBAIKAN EJAAN PADA KARYA TULIS ILMIAH DI PROGRAM STUDI TEKNIK INFORMATIKA DENGAN MENERAPKAN ALGORITMA *LEVENSHTEIN DISTANCE*

Roby Nur Hamzah
NPM: 12.1.03.02.0196
Fakultas Teknik – Teknik Informatika
Robyhamzah@gmail.com
Resty Wulanningrum, M.Kom. dan Patmi Kasih, M.Kom.
UNIVERSITAS NUSANTARA PGRI KEDIRI

ABSTRAK

Roby Nur Hamzah: Aplikasi Perbaikan Ejaan Pada Karya Tulis Ilmiah Di Program Studi Teknik Informatika Dengan Menerapkan Algoritma *Levenshtein Distance*, Skripsi, TI, FT UN PGRI Kediri, 2016.

Dalam membuat karya tulis ilmiah, mahasiswa Prodi Teknik Informatika sering sekali melakukan kesalahan-kesalahan penulisan dalam hal pengetikan. Kesalahan yang sering terjadi adalah kurangnya pengetahuan mahasiswa akan penulisan ejaan yang baku, kelalaian mahasiswa yang tidak disengaja, kesalahan pengaturan aplikasi yang digunakan untuk media pengetikan (*Microsoft Word, Notepad, Open Office Word*) dan beberapa hal lainnya. Untuk mengurangi kesalahan tersebut, perlu adanya sebuah aplikasi yang dapat melakukan koreksi kesalahan penulisan ejaan dengan menggunakan kamus kata baku (KBBI) sebagai acuan untuk saran perbaikan kesalahan penulisan.

Pada aplikasi yang dibuat dalam penelitian ini, dimanfaatkan algoritma *Levenshtein Distance* untuk melakukan koreksi kata dengan menghitung jarak terdekat dari *string* sumber (s) dengan *string* target (t). Jika selisih *string* sumber (s) dengan *string* target (t) memiliki jarak terendah, maka akan dijadian saran perbaikan berdasarkan urutan jarak *string* terendah hingga terbesar. Acuan perhitungan dengan algoritma dalam penelitian ini dibatasi dengan jarak toleransi = 2 dan banyaknya saran perbaikan adalah 20 kata. Aplikasi dibuat dengan konsep sederhana dimana hanya mencocokkan kata yang ada pada dokumen dengan kata baku yang ada pada kamus dalam basisdata. Jika terdapat kesalahan penulisan, maka aplikasi memberi saran perbaikan dan menganjurkan untuk mengganti kesalahan tersebut berdasarkan saran yang sudah diberikan, dimana saran yang diberikan hanya sebatas kesalahan per-kata.

Hasil dari sistem yang telah dibuat dalam penelitan ini adalah sistem yang dibuat dapat melakukan koreksi dan pemberian saran perbaikan kesalahan ejaan pada karya tulis ilmiah dengan menggunakan algoritma *Levenshtein Distance*. Sementara untuk acuan saran perbaikan kata menggunakan data dari KBBI yang disimpan pada *database* dan di koneksikan dengan aplikasi untuk menjadi acuan saran perbaikan kesalahan penulisan ejaan. Implementasi algoritma *Levenshtein distance* berjalan baik, dimana algoritma menghitung kesalahan penulisan ejaan dengan menghitung jarak terendah sesuai dengan rumus perhitungan yang ada. Namun jika terdapat kesalahan penulisan lebih dari dua karakter, dan tidak ada pada kamus, maka kesalahan tersebut diabaikan oleh aplikasi.

Kata kunci: Perbaikan Ejaan, Karya Tulis Ilmiah, Program Studi Teknik Informatika, Algoritma *Levenshtein Distance*.

I. LATAR BELAKANG

Karya tulis merupakan uraian atau laporan tentang kegiatan, temuan atau informasi yang disajikan untuk tujuan dan sasaran tertentu. Kita menyusun suatu karya tulis dengan maksud agar dapat dibaca oleh orang tertentu, golongan masyarakat tertentu, atau masyarakat luas. Oleh karena itu, kualitas suatu karya tulis menjadi sangat penting (Lenni, 2012).

Kualitas suatu karya tulis dapat dilihat dari beberapa faktor. Salah satunya ditentukan dari proses penulisan yang baik, sesuai dengan ejaan yang benar berdasarkan kamus besar Bahasa Indonesia. Menurut Maxine Hairston (1986: 6) menyebutkan bahwa "tulisan yang baik itu harus bersifat signifikan, jernih, ekonomis, bersifat membangun, dan gramatik (*good writing is significant, clear, unifiel, economical, developed, and grammatical*)".

Dalam dunia pendidikan, karya tulis digunakan sebagai acuan untuk menuangkan hasil pembelajaran ke dalam bentuk karya tulis ilmiah dan dapat dipertanggung jawabkan secara ilmiah. Program Studi Teknik Informatika yang bergerak di bidang pendidikan, juga mewajibkan mahasiswanya untuk membuat karya tulis ilmiah sebagai salah satu syarat dalam memperoleh gelar sarjana.

Program Studi Teknik Informatika yang berada dibawah Fakultas Teknik Universitas Nusantara PGRI Kediri adalah salah satu Program Studi yang paling banyak mempunyai jumlah mahasiswa yang aktif di Fakultas Teknik. Didirikannya Program Sudi Teknik Informatika ini adalah sebagai salah satu wujud dari peran aktif Universitas Nusantara PGRI Kediri dalam pendidikan tinggi nasional. Universitas Nusantara PGRI Kediri melihat bidang komputer dan teknologi informasi adalah bidang yang perlu dibangun fondasi keilmuannya di lingkungan Universitas Nusantara PGRI Kediri. Oleh karena itu, untuk meningkatkan mutu pendidikan, salah satunya yaitu memperbaiki kualitas karya tulis ilmiah yang dibuat oleh mahasiswa.

Ketika membuat karya tulis ilmiah, sering sekali terjadi kesalahan dalam hal pengetikan. Kesalahan tersebut dapat berupa kurangnya pengetahuan mahasiswa akan ejaan yang benar sesuai dengan kamus besar Bahasa Indonesia, kelalaian mahasiswa yang tidak disengaja, kesalahan pengaturan aplikasi yang digunakan untuk media pengetikan (*Microsoft Word, Notepad, Open Office Word*) dan beberapa hal lain yang menyebabkan terjadinya kesalahan ejaan kata.

Berdasarkan uraian di atas, maka dibutuhkan suatu aplikasi yang dapat membantu Program Studi Teknik Informatika dalam memperbaiki kesalahan pengetikan ejaan dalam karya tulis ilmiah yang dibuat oleh mahasiswa agar kualitas karya tulis menjadi lebih baik. Aplikasi ini

nantinya akan melakukan deteksi dan pemeriksaan terhadap kata yang pengejaannya salah, dan kemudian menampilkan kandidat perbaikan untuk memperbaiki kata yang sesuai dengan kamus bahasa Indonesia.

Untuk mempermudah mencari kemiripan kata yang sesuai dengan pengejaan yang benar berdasarkan kamus besar bahasa Indonesia, maka digunakan sebuah algoritma yang bernama *Levenshtein distance*. Algoritma ini merupakan sebuah algoritma untuk mencari nilai minimum dari *string* kata yang salah setelah membandingkan dengan kata dalam kamus data dengan perubahan yang diijinkan yaitu mengubah huruf, menyisipkan huruf dan menghapus sebuah huruf.

II. METODE

A. Algoritma Levenshtein Distance

Algoritma *Levenshtein Distance* ditemukan oleh Vladimir Levenshtein, seorang ilmuan asal Rusia pada tahun 1965, dimana algoritma ini sering juga disebut dengan Edit Distance. Yang dimaksud dengan *distance* adalah jumlah modifikasi yang dibutuhkan untuk mengubah suatu bentuk *string* ke bentuk *string* yang lain, sebagai contoh hasil penggunaan algoritma ini, *string* “komputer” dan “computer” memiliki *distance* 1 karena hanya perlu dilakukan satu operasi saja untuk mengubah satu *string* ke *string* yang lain. Dalam kasus dua *string* di atas, *string* “computer” dapat menjadi “komputer” hanya dengan melakukan satu penukaran karakter “c” menjadi “k” (Andhika, 2010).

Algoritma *Levenshtein Distance* digunakan secara luas dalam berbagai bidang, misalnya mesin pencari, pengecek ejaan (*spelling checking*), pengenal pembicaraan (*speech recognition*), pengucapan dialek, analisis DNA, pendeteksi pemalsuan, dan lain-lain. Algoritma ini menghitung jumlah operasi *string* paling sedikit yang diperlukan untuk mentransformasikan suatu *string* menjadi *string* yang lain (Adiwidya, 2009). Algoritma *Levenshtein Distance* bekerja dengan menghitung jumlah minimum pentranformasian suatu *string* menjadi *string* lain yang meliputi penghapusan, penyisipan, dan penukaran (Husain, 2013).

Selisih perbedaan antar *string* dapat diperoleh dengan memeriksa apakah suatu *string* **sumber (s)** sesuai dengan *string* **target (t)**. Nilai selisih perbedaan ini disebut juga *Edit Distance*/jarak *Levenshtein*. Jarak *Levenshtein* antar *string* **(s)** dan *string* **(t)** tersebut adalah fungsi **D** yang memetakan **(s,t)** ke suatu bilangan ***real non-negatif***, sebagai contoh diberikan

dua buah *string* **s = $s_1, s_2, s_3, \ldots\ldots s_m$** dan **t = $t_1, t_2, t_3, \ldots\ldots t_n$** dengan **|s| = m** dan **|t| = n** sepanjang **alfabet *V* berukuran r** sehingga **s** dan **t** anggota dari *v**. **s_j** adalah karakter pada posisi **ke-j** pada *string* **s** dan **t_i** adalah karakter pada posisi **ke-i** pada *string* **t**. Sehingga jarak *Levenshtein* dapat didefinisikan sebagai berikut: (Harahap, 2013)

$$D(s,t) = d(s_1,t_1) + d(s_2,t_2) + d(s_3,t_3) \ldots\ldots d(s_m,t_n) \quad (1)$$

$$D(s,t) = \sum_{i=1}^{r} d(sj, ti) \quad (2)$$

Dimana : $s_j, t_i \in V$ untuk $i = 1,2,3\ldots\ldots, r$

$d(s_j,t_i) = 0$ jika $s_j = t_i$ dan

$d(s_j,t_i) = 1$ jika $s_j \neq t_i$

B. Simulasi Algoritma

Untuk mengetahui proses perhitungan algoritma *Levenshtein Distance* dalam memperbaiki kesalahan ejaan, maka akan dilakukan sebuah simulasi algoritma dengan contoh sebagai berikut:

1. Diketahui sebuah *string* sumber (s) = "teknuk" dan string target (t) = "teknik". Untuk menyamakan string sumber dengan string target, maka dilakukan perhitungan dengan algoritma *Levenshtein Distance*.

   Inisialisasi perhitungan

   a. *String* sumber (s) = "teknuk"

   b. *String* target (t) = "teknik"

   c. Hitung jumlah string kata, dan buat harga mutlaknya (jumlah string tidak boleh negatif): |s| = 6 dan |t| = 6.

   d. Masukkan kedalam rumus perhitungan:

   $D(s,t) = \sum_{i=1}^{r} d(sj, ti)$

   $= d(s_1,t_1) + d(s_1,t_1) + d(s_1,t_1) + d(s_1,t_1) + d(s_1,t_1) + d(s_1,t_1)$

   = d(t, t) + d(e, e) + d(k, k) + d(n, n) + d(u, i) + d(k, k)

   = 0 + 0 + 0 + 0 + 1 + 0

   = 1

   Sehingga jarak *levenshtein* antara string s = "teknuk" dan t = "teknik" adalah $D(s,t) = 1$.

   e. Buat tabel matrik berdasarkan tabel penjelasan algoritma

Tabel 1.1 Matrik langkah pertama, ketika *i=1*

	sumber	t	e	k	n	u	k
target	0	1	2	3	4	5	6
t	1	0					
e	2	1					
k	3	2					
n	4	3					
i	5	4					
k	6	5					

Tabel 1.2 Matrik langkah kedua ketika *i=2*

	sumber	t	e	k	n	u	k
target	0	1	2	3	4	5	6
t	1	0	1				
e	2	1	0				
k	3	2	1				
n	4	3	2				
i	5	4	3				
k	6	5	4				

Tabel 1.3 Matrik langkah ketiga ketika *i=3*

	Sumber	t	e	k	n	u	k
target	0	1	2	3	4	5	6
t	1	0	1	2			
e	2	1	0	1			
k	3	2	1	0			
n	4	3	2	1			
i	5	4	3	2			
k	6	5	4	3			

Tabel 1.4 Matrik langkah keempat ketika *i=4*

	sumber	t	e	k	n	u	k
target	0	1	2	3	4	5	6
t	1	0	1	2	3		
e	2	1	0	1	2		
k	3	2	1	0	1		
n	4	3	2	1	0		
i	5	4	3	2	1		
k	6	5	4	3	2		

Tabel 1.4 Matrik langkah kelima ketika *i=5*

	sumber	t	e	k	n	u	k
target	0	1	2	3	4	5	6
t	1	0	1	2	3	4	
e	2	1	0	1	2	3	
k	3	2	1	0	1	2	
n	4	3	2	1	0	1	
i	5	4	3	2	1	1	
k	6	5	4	3	2	2	

Tabel 1.5 Matrik langkah keenam ketika *i=6*

	sumber	t	e	k	n	u	k
target	0	1	2	3	4	5	6
t	1	0	1	2	3	4	5
e	2	1	0	1	2	3	4
k	3	2	1	0	1	2	3
n	4	3	2	1	0	1	2
i	5	4	3	2	1	1	2
k	6	5	4	3	2	2	1

Dalam implementasi pada koreksi kesalahan ejaan, algoritma *Levenshtein Distance* akan mencari seluruh kata pada kamus yang dijadikan sebagai pembanding dan koreksi kesalahan ejaan. Setiap kata yang memiliki perbedaan jarak (*distance*) minimum, akan mendapat pertimbangan solusi untuk saran perbaikan. Jadi setiap kata yang memiliki tingkat kemiripan yang tinggi (*distance* terendah) terhadap kata yang tidak sesuai dengan kamus bahasa Indonesia, maka akan menjadi saran utama dalam perbaikan.

III. HASIL DAN KESIMPULAN

A. Implementasi Hasil

Implementasi merupakan hasil dari sistem berupa wujud jadi dari aplikasi yang sudah dibuat yang meliputi tampilan program, proses program hingga evaluasi program. Berikut implementasi sistem yang sudah dibuat:

1. Tampilan Program

a. Tampilan Awal

Tampilan awal program hanya berisi menu awal, yaitu menu buka untuk membuka dan meng-*upload* file dokumen di media penyimpanan *online* google *drive* dan menu kamus pengguna untuk melihat kata apa saja yang sudah admin tambahkan ke dalam kamus.

Gambar 2.1 Tampilan awal program

b. Tampilan *upload* dokumen

Tampilan ini adalah tampilan untuk meng-*upload* dokumen ke dalam media penyimpanan *online* google *drive*. Syarat untuk meng-*upload* dokumen yang akan dikoreksi adalah admin harus

mengisi nama dan NPM mahasiswa, dimana ini bertujuan untuk mengetahui siapa pemilik karya tulis yang akan diperbaiki kesalahan penulisan ejaan.

Gambar 2.2 Tampilan *upload* dokumen

Setelah dokumen sudah di upload, maka dokumen dapat di buka untuk dilakukan proses perbaikan kesalahan ejaan pada dokumen yang sudah dipilih.

Gambar 2.3 Tampilan buka dokumen

c. Tampilan tambah kamus pengguna

Menu tampilan tambah kamus pengguna digunakan untuk menambahkan kata yang dianggap benar, namun disalahkan oleh aplikasi karena tidak adanya kata tersebut pada kamus saran perbaikan. Akan tetapi kata tersebut merupakan kata baku, dimana hal ini dikarenakan kamus yang dijadikan acuan saran perbaikan kurang lengkap.

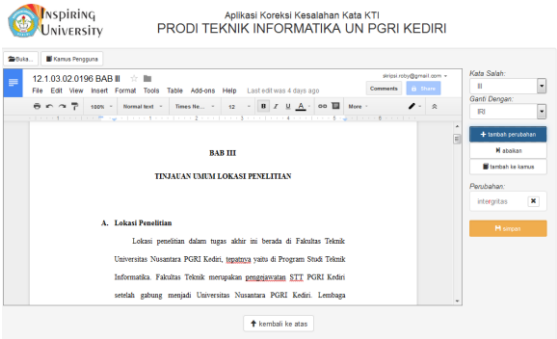
Gambar 2.4 Tampilan tambah kamus pengguna

d. Tampilan utama program

Ketika dokumen dibuka, maka dokumen secara otomatis akan dicari kesalahan penulisan ejaan dan akan dikumpulkan pada *form kata salah*. Admin kemudian harus memilih dan mencari kata mana yang akan diperbaiki kesalahannya. Ketika kata salah sudah di pilih, maka admin harus memilih saran perbaikan akan muncul pada *form ganti dengan*. Jika sudah, maka admin harus menekan *buttom tambah perubahan*.

Pada tampilan utama ini, terdapat *buttom* untuk mengabaikan kesalahan kata. Hal ini bertujuan jika terdapat kata benar namun dianggap salah karena dalam kamus tidak terdapat kata tersebut. Untuk menghindari hal tersebut, maka ditambahkan *buttom tambah ke kamus* supaya kata baku yang

belum ada pada kamus dapat ditambahkan oleh admin.



Gambar 2.5 Tampilan utama program

B. Kesimpulan

Berdasarkan hasil penelitian dan implementasi sistem, maka dapat diambil kesimpulan sebagai berikut:

1. Aplikasi ini dikhususkan pada file karya tulis ilmiah yang berformat *docx yang ada pada Prodi Teknik Informatika. Pencarian kesalahan penulisan ejaan dengan cara membandingkan setiap kata yang ada pada dokumen dengan kamus kata baku yang ada pada aplikasi. Apabila terdapat kata yang salah dan tidak sama dengan kata baku pada kamus, maka aplikasi akan menyatakan bahwa kata tersebut salah atau tidak baku dan akan memberikan saran perbaikan.
2. Acuan saran perbaikan menggunakan data dari KBBI Kemendikbud versi 3 yang disimpan pada *database* dan di koneksikan dengan aplikasi untuk menjadi acuan saran perbaikan kesalahan penulisan ejaan.
3. Implementasi algoritma *Levenshtein distance* berjalan baik, dimana algoritma menghitung kesalahan penulisan ejaan dengan menghitung jarak terendah dan mencari saran perbaikan dengan batas toleransi kesalahan = 2 dan jumlah saran perbaikan dibatasi dengan 20 saran perbaikan. Hal ini karena dalam penelitian yang dilakukan, rata-rata kesalahan penulisan ejaan tidak lebih dari 2 karakter huruf. Jika terdapat kesalahan kata lebih dari 2 karakter dan tidak terdapat pada kamus kata baku yang dijadikan acuan perbaikan, maka aplikasi akan mengabaikan kesalahan tersebut. Sehingga kesalahan kata pada dokumen tidak dapat dikoreksi.

IV. DAFTAR PUSTAKA

Andhika, F. R. 2010. *Penerapan String Suggestion Dengan Algoritma Levenshtein Distance Dan Alternatif Algoritma Lain Dalam Aplikasi*. Skripsi. Institut Teknologi Bandung.

Andriani, M. Ariana, S. & Andri. 2014. Aplikasi Koreksi Kesalahan Berbasis Pada Tulisan Berbahasa Indonesia Untuk Meningkatkan Kualitas Penulisan Karya Ilmiah. Makalah disajikan pada Prosiding Seminar Nasional Aplikasi Sains & Teknologi (SNAST), Yogyakarta, 15 November 2014.

(Online), tersedia: http://repository.akprind.ac.id, diunduh 26 Desember 2015.

Arifin, Zaenal. 2008. Dasar-dasar Penulisan Karya Ilmiah. Jakarta: Grasindo.

Hairston, Maxine. 1986. Contemporary Composition. Boston: Hougton Mifflin Company.

Harahap, B. S. 2013. Implementasi Algoritma Cocke-Younger-Kasami (Cyk) Dan Levenshtein Untuk Mengoreksi Kesalahan Pengejaan Kalimat Bahasa Inggris. Skripsi. Universitas Sumatera Utara.

Husain, S. W. 2013. Penerapan Algoritma Binary Search Dan Metode Approximate String Matching Pada Aplikasi Kamus Bahasa Indonesia-Mongondow Berbasis Mobile. Skripsi. Universitas Negeri Gorontalo.

Iskandar, Denny. 2012. Materi Karya Tulis Ilmiah. (online). tersedia: http://file.upi.edu/Direktori/FPBS/JUR._PEND._BHS._DAN_SASTRA_INDONESIA/196606291991031_DENNY_ISKANDAR/Materi_Karya_Tulis Ilmiah.pdf, diunduh 11 Desember 2015.

Jogiyanto, H.M. 2005. Analisis Dan Desain Sistem Informasi. Jogjakarta: Andi Edisi. 3. tersedia http://www.scholar.google.com diunduh 22 Desember 2015.

Lenni, ST. 2012. Bab 1 Pendahuluan Tata Tulis Karya Tulis Ilmiah. (online). tersedia: http://dosen.narotama.ac.id/wp-content/uploads/2012/12/BAB-1-PENDAHULUAN-TATA-TULIS-KARYA-ILMIAH.doc, diunduh 11 Desember 2015.

McCrimmon. 1984. Writing with a Purpose. Boston: Hougton Mifflin Company.

Meta Sumarni. 2015. Analisis Kesalahan Penggunaan Ejaan Pada Peserta Didik SDN Kelas V SDN 1 Pahandut Seberang. Skripsi. Dipublikasikan. Palangka Raya: FKIP Universitas Muhammadiyah Palangka Raya.

Muliantara, A. Santiyasa, W.I. & Adriyani M.M.N. 2012. Implementasi Algoritma Levenshtein Distance Dan Metode Empiris Dan Menampilkan Saran Perbaikan Kesalahan Pengetikan Dokumen Berbahasa Indonesia. Jurnal Elektronik Ilmu Komputer, (online), tersedia: http://ojs.unud.ac.id, diunduh 25 November 2015.

Pedoman umum ejaan bahasa indonesia yang disempurnakan (EYD). 1997. Surabaya: Apollo.

Raharjo, Budi. 2011. Belajar otodidak membuat database menggunakan MySQL Bandung : INFORMATIKA Bandung.

Rio Devilito. 2014. Analisis Kesalahan Berbahasa Pada Text Bacaan Dalam Buku Ajar Bahasa Indonesia Kelas VII

Wahana Pengetahuan. Skripsi. Dipublikasikan. Surakarta: FKIP UNS.

Rosa, A.S. & Salahudin, M. 2011. Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek). Bandung: Modula.

Ryza, C.U.P. 2014. Sistem Pendukung Keputusan Seleksi Penerimaan Siswa Baru Dengan Metode Fuzzy Multi Attribute Decision Making Model Yager(Studi Kasus : Smp It Iqra Bengkulu). Skripsi Universitas Bengkulu. tersedia http://repository.unib.ac.id diunduh 21 Desember 2015.

Safri, M.F., Dayawati, R.N. & Wibowo, A.T. 2011. Implementasi Dan Analisis Algoritma Edit Distance Untuk Spelling Checker Pada Document Teks Bahasa Indonesia. Jurnal Elektronik, (online), tersedia:*https://repository.telkomuniversity.ac.id, diunduh 24 November 2015.*

Sidik, Betha.2014. Pemrograman Web PHP. Bandung: Informatika.

Suryani, A.A. Wirayuda, B.A.T. & Ferdinanda B.E. 2010. Penerapan Algoritma Levenshtein Distance Pada Dektop Search Berbasis Information Retrieval Implementation of Levenshtein Distance Algorithm on Dektop Search Which Based on Information Retrieval. Jurnal Elektronik, (online), tersedia: *https://repository.telkomuniversity.ac.id, diunduh 24 November 2015.*

Syihabuddin. 2006. "Ihwal Menulis Akademik dalam MPK Bahasa Indonesia di Perguruan Tinggi". Makalah disajikan dalam Pelatihan Nasional Dosen Bahasa Indonesia di Yogyakarta.