

IMPLEMENTASI SPELLING CHECKER DENGAN ALGORITMA LEVENSHTAIN DISTANCE PADA ENSIKLOPEDIA IT (Information Technology) BERBASIS WEBSITE

Joko Jatminto

D3 Manajemen Informatika, Fakultas Teknik, Universitas Negeri Surabaya, minco_jokja@yahoo.com

I Kadek Dwi Nuryana

D3 Manajemen Informatika, Fakultas Teknik, Universitas Negeri Surabaya

Abstrak

Pengetahuan tentang teknologi informasi sangatlah penting di era globalisasi saat ini, banyak istilah tentang teknologi informasi yang belum di mengerti dan dipahami oleh seluruh masyarakat umum. Untuk kebutuhan pengetahuan tentang teknologi informasi perlu adanya pusat sumber pengetahuan yang bisa sebagai referensi atau sumber informasi.

Bahasa merupakan alat komunikasi lisan manusia baik secara lisan maupun tulisan. Peran bahasa Indonesia dan bahasa Inggris yang baik dan benar sebagai bahasa resmi nasional dan internasional memiliki arti yang sangat penting sebagai penjelasan tentang teknologi informasi.

Kesalahan pengetikan dan kontrol pengecekan bahasa yang baik dan benar jika dilakukan secara manual pastinya akan menghabiskan banyak waktu, apalagi di zaman serba modern seperti saat ini.

Kesalahan penulisan pada umumnya disebabkan kedekatan letak *keyboard*, slip jari, ataupun karena dua karakter yang letaknya tertukar. Untuk permasalahan itu muncullah ide untuk membuat sistem pengecekan ejaan atau *spelling checker* di ensiklopedia tentang teknologi informasi yang dikembangkan dengan berbasis *website* menggunakan bahasa pemrograman PHP dan menggunakan basis data kumpulan kosakata yang mengacu pada lisensi dokumen Copyright © 2003 IlmuKomputer.Com.

Kata Kunci : Teknologi Informasi, Kesalahan Penulisan, Berbasis Website, Spelling checker, Ensiklopedia

Abstract

Knowledge of information technology is crucial in the era globalisasi this time, many terms of information technology that has not been understood and understood by the general public. To the knowledge of the information technology needs of the need for the central source of knowledge that can be as references or resources.

Language is a tool of human lingual communication both orally and in writing. Role of Indonesian and English is good and right as the official language of national and international has a very important meaning as a description of information technology.

Typing error checking and control language is good and true if done manually certainly will spend a lot of time, let alone an age of modern convenience such as this.

Writing errors are mainly due to the proximity of the location of the *keyboard*, slip a finger, or because the two characters are located swapped. For the problems that came the idea to create a system of checking spelling or *spelling checker* in the encyclopedia of information technology based *website* developed using the programming language PHP and uses a database of vocabulary that refers to the collection of licensing documents Copyright © 2003 IlmuKomputer.Com.

Keywords: Information Technology, Writing Error, Based Website, Spelling checker, Encyclopedia

PENDAHULUAN

Kehidupan manusia pada masa sekarang ini tidak terlepas dari informasi, hal ini secara tidak langsung sudah menjadi kebutuhan di kehidupan sehari-hari. Informasi senantiasa membuka hal-hal yang baru dalam era globalisasi di masyarakat umum, perkembangan ilmu pengetahuan dan teknologi merupakan suatu hal yang sangat cepat perkembangannya dari tahun ke tahun. Teknologi komputer salah satunya yang digunakan untuk pengolahan dan pencarian data yang diproses untuk menghasilkan informasi terutama dalam jumlah yang sangat banyak, hal ini dimudahkan karena memiliki kecepatan, keakuratan serta efisiensi yang dibandingkan dengan sistem manual.

Kesalahan pengetikan dokumen memang sering sekali terjadi, tentu saja dalam penulisan dokumen tersebut adanya kesalahan pengetikan yang disebabkan oleh beberapa faktor yaitu : Letak huruf pada *keyboard* yang berdekatan, kesalahan karena kegagalan mekanis atau slip dari tangan atau jari, kesalahan yang disebabkan oleh ketidaksengajaan.

Proses *spelling checker* atau pengecekan kesalahan pengetikan dengan cara manual akan menghabiskan banyak waktu dan membutuhkan suatu sumber pasti sebagai acuan bahwa kata tersebut memang salah dalam proses penulisan. Efisiensi waktu yang dibutuhkan jika dilakukan dengan manual tentunya tidak akan optimal karena proses pengecekan kata membutuhkan waktu yang cukup lama sehingga memungkinkan adanya *human error*.

Bahasa sebagai salah satu alat komunikasi lisan manusia baik secara lisan maupun tulisan sangatlah penting untuk penyampaian informasi. Mengingat peran bahasa Indonesia dan bahasa Inggris sebagai bahasa resmi nasional dan internasional.

Ensiklopedia memang sudah banyak yang dipublikasikan, baik itu yang berkategori khusus bidang ilmu tertentu ataupun berkategori bebas. Hanya saja, sering kali ini terbentur masalah kapasitas media yang digunakan sehingga referensi pengetahuan yang disematkan terbatas dan sulit untuk diperbaharui. Kemajuan teknologi memberikan solusi atas masalah ini, yaitu dengan mengimplementasikan ensiklopedia dalam jaringan internet. Solusi ini juga mempunyai nilai tambah seperti lebih mudah diakses oleh siapapun dan dari manasaja, informasi dapat disajikan dengan format multimedia sehingga lebih mudah ditangkap dan dicerna, serta mempermudah pengolahan informasi sehingga dapat ditampilkan dengan lebih dinamis.

KAJIAN PUSTAKA

Ensiklopedia

Menurut Zibriel (2008). Ensiklopedia adalah kumpulan penjelasan hal-hal dari berbagai cabang ilmu pengetahuan yang tersusun menurut abjad atau menurut kategori tertentu. Ensiklopedia sendiri secara etimologi diambil dari bahasa Yunani yaitu *enkyklios paideia* yang artinya adalah sebuah lingkaran pengajaran yang lengkap dimana artinya ensiklopedia itu itu adalah pendidikan

yang mencakup semua ilmu pengetahuan. Perbedaan ensiklopedia dibandingkan kamus adalah ensiklopedia tidak sekedar memberikan arti atau definisi sebuah hal akan tetapi melengkapinya juga dengan informasi lain seputar hal yang sedang diterangkan.

Menurut Wikipedia (2015). Ensiklopedia adalah sejumlah tulisan yang berisi penjelasan yang menyimpan informasi secara komprehensif dan cepat dipahami serta dimengerti mengenai keseluruhan cabang ilmu pengetahuan atau khusus dalam satu cabang ilmu pengetahuan tertentu yang disusun berdasarkan abjad.

Kata "ensiklopedia" diambil dari bahasa Yunani ; *enkyklios paideia* yang berarti sebuah lingkaran atau pengajaran yang lengkap. Maksudnya ensiklopedia itu sebuah pendidikan paripurna yang mencakup semua lingkaran ilmu pengetahuan. Seringkali ensiklopedia dicampurbaurkan dengan kamus dan ensiklopedia-ensiklopedia awal memang berkembang dari kamus. Perbedaan utama antara kamus dan ensiklopedia ialah bahwa sebuah kamus hanya memberikan definisi setiap entri atau lemma dilihat dari sudut pandang linguistik atau hanya memberikan kata-kata sinonim saja, sedangkan sebuah ensiklopedia memberikan penjelasan secara lebih mendalam dari yang kita cari. Sebuah ensiklopedia mencoba menjelaskan setiap artikel sebagai sebuah fenomena. Atau lebih singkat: kamus adalah daftar kata-kata yang dijelaskan dengan kata-kata lainnya sedangkan sebuah ensiklopedia adalah sebuah daftar hal-hal yang kadang kala dilengkapi dengan gambar untuk lebih menjelaskan.

Dengan munculnya revolusi informasi digital, maka muncullah pula ensiklopedia dalam bentuk perangkat lunak di mana setiap entri bisa dicari dengan mudah. Contohnya seperti pada tahun 2001 muncul sebuah ensiklopedia populer di internet yaitu Wikipedia. Wikipedia berusaha menulis sebuah ensiklopedia yang terlengkap dalam semua bahasa didunia dan menyajikannya secara bebas di dunia maya. Ensiklopedia online ini tersaji dalam banyak bahasa, antara lain bahasa Indonesia.

Teknologi Informasi

Menurut Amijaya (2010). Teknologi informasi adalah studi atau penggunaan peralatan elektronika, terutama komputer untuk menyimpan, menganalisis, dan mendistribusikan informasi dalam bentuk apapun termasuk kata-kata, bilangan, dan gambar. Teknologi informasi mencakup perangkat keras dan perangkat lunak untuk melaksanakan satu atau sejumlah tugas pemrosesan data seperti menangkap, mentransmisikan, menyimpan, mengambil, memanipulasi, atau menampilkan data. Definisi teknologi informasi tidak hanya terbatas pada teknologi komputer (perangkat keras dan perangkat lunak) yang digunakan untuk memproses dan menyimpan informasi, melainkan juga mencakup teknologi komunikasi untuk mengirimkan informasi. Secara lebih umum, teknologi informasi adalah segala bentuk teknologi yang diterapkan untuk memproses dan mengirimkan informasi dalam bentuk elektronik.

Peranan teknologi informasi pada aktivitas manusia pada saat ini sangat besar. Teknologi informasi telah menjadi

fasilitator utama, memberikan andil besar terhadap perubahan mendasar pada struktur, operasi dan manajemen organisasi. Berkat teknologi ini, berbagai kemudahan dapat dirasakan manusia. Menurut Kadir (2003), peranan teknologi informasi meliputi :

- a. Teknologi informasi menggantikan peran manusia. Dalam tugas ini, teknologi informasi melakukan otomatisasi terhadap suatu tugas atau proses.
- b. Teknologi informasi memperkuat peran manusia, yakni dengan menyajikan informasi terhadap suatu tugas atau proses.
- c. Teknologi informasi berperan dalam restrukturisasi terhadap peran manusia. Dalam hal ini, teknologi berperan dalam melakukan perubahan-perubahan terhadap sekumpulan tugas atau proses.

Website

Menurut Murya Arief Basuki (2009). Merupakan suatu koleksi dokumen HTML pribadi atau perusahaan yang memuat informasi dalam *Web Server* (sistem komputer di suatu organisasi, yang berfungsi sebagai server (suatu unit komputer yang berfungsi untuk menyimpan informasi dan untuk mengelola jaringan komputer) untuk fasilitas *World Wide Web* atau *Web* , dan dapat diakses oleh seluruh pemakai Internet).

Menurut Lydia Iskandar (2012). *Website* disebut juga site, situs, situs web atau portal. Merupakan kumpulan halaman web yang berhubungan antara satu dengan lainnya, halaman pertama sebuah *website* adalah home page, sedangkan halaman demi halamannya secara mandiri disebut web page, dengan kata lain *website* adalah situs yang dapat diakses dan dilihat oleh para pengguna internet diseluruh dunia. *Website* adalah situs yang dapat diakses dan dilihat oleh para pengguna internet. Pengguna internet semakin hari semakin bertambah banyak, sehingga hal ini adalah potensi pasar yang berkembang terus.

Website merupakan salah satu sumber daya internet yang berkembang pesat. Pendistribusian informasi *website* dilakukan melalui pendekatan *hyperlink*, yang memungkinkan suatu teks, gambar, ataupun objek yang lain menjadi acuan untuk membuka halaman-halaman yang lain. Melalui pendekatan ini, seseorang dapat memperoleh informasi dengan beranjak dari satu halaman ke halaman lain.

Website berasal dari kata dalam bahasa Inggris yang bila diterjemahkan dalam bahasa Indonesia berarti "jaring laba-laba". Hampir sama dengan arti dari kata *website* itu sendiri, *website* telah membentang ke seluruh penjuru dunia. Tidak hanya terbatas pada lembaga-lembaga penelitian yang ingin memublikasi hasil riset, tetapi juga telah banyak digunakan oleh perusahaan bisnis yang ingin mengiklankan produk atau untuk melakukan transaksi bisnisnya. Sejarah *website* dimulai pada bulan Maret 1989 ketika Tim Berner-lee yang bekerja di laboratorium fisika partikel Eropa atau yang dikenal dengan nama CERN (Conseil Européen pour la Recherche Nucléaire) yang berada di Genewa, Swiss, mengajukan protocol (suatu tata cara untuk berkomunikasi) sistem distribusi informasi Internet yang

digunakan untuk berbagi informasi di antara para fisikawan.

Protocol inilah yang selanjutnya dikenal sebagai *protocol World Wide Web* dan dikembangkan oleh *World Wide Web Consortium* (W3C). sebagaimana diketahui, W3C adalah konsorsium dari sejumlah organisasi yang berkepentingan dalam pengembangan berbagai standar yang berkaitan dengan web.

Pada awalnya aplikasi web dibangun hanya dengan menggunakan bahasa yang disebut HTML (*HyperText Markup Language*) dan protokol yang digunakan dinamakan HTTP (*HyperText Transfer Protocol*). Pada perkembangan berikutnya, sejumlah script dan objek yang dikembangkan untuk memperluas kemampuan HTML. Pada saat ini, banyak script seperti itu; antara lain PHP dan ASP, sedangkan contoh yang berupa objek antarlain adalah *applet* (*Java*).

Aplikasi *website* sendiri dapat dibagi menjadi *website* statis dan *website* dinamis. *Website* statis dibentuk dengan menggunakan HTML saja. Kekurangan aplikasi ini terletak pada keharusan untuk memelihara program secara terus-menerus untuk mengikuti setiap perubahan yang terjadi. Kelamahan ini dapat diatasi dengan model aplikasi *website* dinamis.

Website dinamis adalah web yang menampilkan informasi yang bersifat dinamis (berubah-ubah) dan dapat saling berinteraksi dengan user. Biasanya untuk *website* statis yang 24 ditonjolkan adalah sisi tampilan yang banyak mengandung grafis sehingga untuk merancang *website* statis tidak diperlukan kemampuan pemrograman yang handal. Dengan memperluas kemampuan HTML, yakni dengan menggunakan perangkat lunak tambahan. Perubahan informasi dalam halaman-halaman *website* dapat ditangani melalui perubahan data, bukan melalui perubahan program. Sebagai implementasinya, aplikasi *website* dapat dikoneksikan ke basis data. Dengan demikian perubahan informasi dapat dilakukan oleh operator atau yang bertanggung jawab terhadap kemutakhiran data, dan tidak menjadi tanggung jawab pemrogram atau *webmaster*.

Konsep yang mendasari aplikasi *website* sebenarnya sederhana. Operasi yang melatar belakanginya melibatkan pertukaran informasi antara komputer yang meminta informasi, yang disebut klien, dan komputer yang memasok informasi (disebut *server*). Lebih detailnya, server yang melayani permintaan dari klien sesungguhnya berupa suatu perangkat lunak yang dinamakan *web server*. Secara internal, *web server* inilah yang berkomunikasi dengan perangkat lunak lain yang disebut *middleware* dan *middleware* inilah yang berhubungan dengan basis data (*database*). Model seperti inilah yang mendukung *website* dinamis. Dengan menggunakan pendekatan *website* dinamis, dimungkinkan untuk membentuk aplikasi berbasis *website* yang berinteraksi dengan database.

Mesin Pencarian (Search Engine)

Menurut Wikipedia (2015). *Search engine* merupakan perangkat pencari informasi dari dokumen-dokumen yang tersedia. Hasil pencarian umumnya ditampilkan dalam bentuk daftar yang seringkali diurutkan menurut

tingkat akurasi ataupun rasio pengunjung atas suatu berkas yang disebut sebagai hits. Informasi yang menjadi target pencarian bisa terdapat dalam berbagai macam jenis berkas seperti halaman situs web, gambar, ataupun jenis-jenis berkas lainnya. Beberapa mesin pencari juga diketahui melakukan pengumpulan informasi atas data yang tersimpan dalam suatu basisdata ataupun direktori web.

Kebanyakan *search engine* ini dirancang untuk melakukan pencarian atas berkas-berkas yang tersimpan dalam layanan www, ftp, publikasi milis, ataupun *news group* dalam suatu jaringan komputer.

Manfaat *search engine* bergantung pada relevansi hasil-hasil yang diberikannya. Meskipun mungkin ada jutaan halaman web yang mengandung suatu kata atau frasa, sebagian halaman mungkin lebih relevan, populer, atau otoritatif daripada yang lain. Kebanyakan mesin pencari menggunakan berbagai metode untuk menentukan peringkat hasil pencarian agar mampu memberikan hasil "terbaik" lebih dahulu. Cara mesin menentukan halaman mana yang paling sesuai, dan urutan halaman-halaman itu diperlihatkan, sangat bervariasi. Metode-metode nya juga berubah seiring waktu dengan berubahnya penggunaan internet dan berevolusinya teknik-teknik baru.

Spelling Checker

Menurut Atmajaya (2009). *Spelling checker* adalah sebuah fasilitas yang berfungsi untuk mengecek kesalahan penulisan ejaan suatu kata berdasarkan bahasa tertentu. Umumnya berkerja dengan menandai kata yang berejaan salah dengan kurva berwarna merah.

Fitur tambahan yang dimiliki oleh sebuah *spelling checker* adalah words suggestion yang berfungsi membantu user dengan memberikan daftar kata-kata yang memiliki ejaan yang mendekati *keyword*. Fitur lain yang biasa dimiliki oleh sebuah *spelling checker* adalah *auto correction*, yang berfungsi mengubah secara otomatis kata yang salah menjadi sebuah kata yang memiliki ejaan yang dekat. Semua tingkat kedekatan ejaan antara kata yang salah dengan kata yang muncul di daftar *suggestion* ataupun kata yang muncul karena fitur *auto correction* ditentukan dengan banyak metode salah satunya metode algoritma *Levenshtein Distance*.

Spelling checker atau yang disebut pemeriksa ejaan adalah aplikasi yang memeriksa semua kata dalam sebuah dokumen untuk menghindari kesalahan pengejaan. Pemeriksa ejaan dapat berupa aplikasi mandiri atau sebagai bagian dari aplikasi yang lebih besar, seperti aplikasi pengolah kata, kamus elektronik, atau mesin pencari web.

Cara kerja *spelling checker* memindai kata-kata pada suatu naskah dan mengekstraknya, membandingkan kata yang salah dengan memberikan pilihan kepada pengguna terhadap kata-kata yang diketahui ejaannya oleh pemeriksa ejaan tersebut. Ini mungkin hanya akan menampilkan beberapa daftar kata, atau juga mengandung beberapa informasi tambahan, seperti kata hubung serta atribut leksikal dan gramatikal.

Dalam beberapa kasus, pemeriksa ejaan memberikan saran kata yang salah; ini dikarenakan ketidak akuratan kata yang terdapat dalam program tersebut.

Levenshtein Distance

Menurut Iwan Saputera (2013). Dalam jurnal penelitian yang berjudul "Aplikasi SMS Filtering Pada Smartphone Android Dengan Menggunakan Metode *Levenshtein Distance*". *Levenshtein Distance* dibuat oleh Vladimir Levenshtein pada tahun 1965, 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. Ada 3 macam operasi utama yang dapat dilakukan oleh algoritma ini yaitu :

a) Operasi Pengubahan Karakter

Operasi pengubahan karakter merupakan operasi menukar sebuah karakter dengan karakter lain contohnya penulis menuliskan *string* "yang" menjadi "yamg". Dalam kasus ini karakter "m" diganti dengan huruf "n".

b) Operasi Penambahan Karakter

Operasi penambahan karakter berarti enambahkan karakter ke dalam suatu *string*. Contohnya *string* "kepad" menjadi *string* "kepada", dilakukan penambahan karakter "a" di akhir *string*. Penambahan karakter tidak hanya dilakukan di akhir kata, namun bias ditambahkan diawal maupun disisipkan di tengah *string*.

c) Operasi Penghapusan Karakter

Operasi penghapusan karakter dilakukan untuk menghilangkan karakter dari suatu *string*. Contohnya *string* "baru" karakter terakhir dihilangkan sehingga menjadi *string* "baru". Pada operasi ini dilakukan penghapusan karakter "r".

Untuk menentukan Levenshtein Distance anatara dua kata kita membutuhkan persamaan matriks sebagai berikut :

$$\text{Max}(i,j) \quad \text{if } \min(i,j) = 0 \\ \text{otherwise}$$

Lev a, b (i,j) =

$$\begin{aligned} & \text{Lev a, b (i-1, j) + 1} \\ \text{Min} & \text{Leva, b (i, j-1) + 1} \\ & \text{Lev a, b (i-1, j-1) + 1 (a \neq b)} \\ & \text{If + 0 = a(i) = b(j)} \end{aligned}$$

lev a,b adalah matriks lev a,b

i adalah baris matriks

j adalah kolom matriks

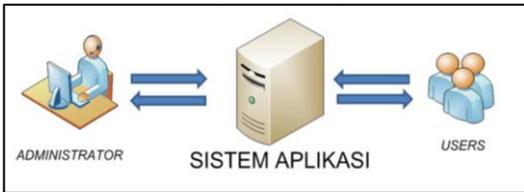
Berikut ini merupakan tabel matriks untuk mencari nilai levenshtein distance antara dua *string*.

		S	L	A	M	A	T
0	0	1	2	3	4	5	6
S	1	0	1	2	3	4	5
E	2	1	1	2	3	4	5
L	3	2	1	2	3	4	5
A	4	3	2	1	2	3	4
M	5	4	3	2	1	2	3
A	6	5	4	3	2	1	2
T	7	6	5	4	3	2	1

Pada kasus diatas merupakan tabel matriks untuk mencari nilai Levenshtein distance dari kata selamat dan smlat, dari tabel matriks diatas di dapat nilai Levenshtein adalah satu, yaitu operasi 1 operasi penambahan. Setelah mencari nilai Levenshtein distance.

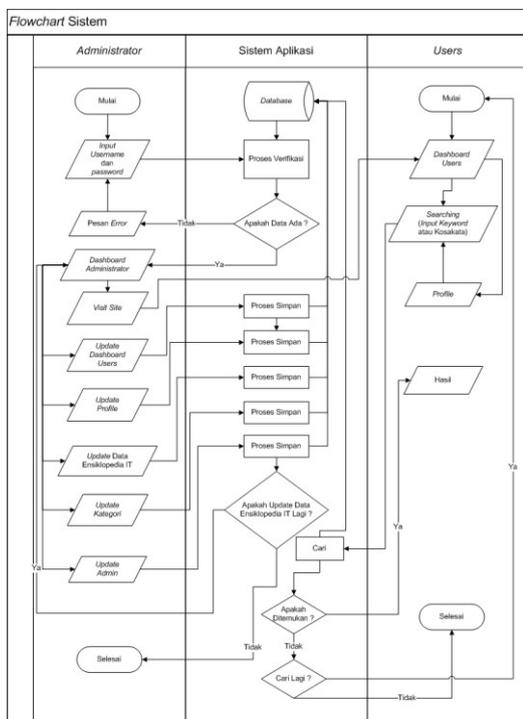
METODE REKAYASA

Gambaran Umum Aplikasi yang Diusulkan



Gambar 1. Gambaran Umum Aplikasi

Gambar 1 di atas merupakan gambaran umum dari aplikasi.



Gambar 2. Flowchart Sistem

Gambar 2 di atas merupakan gambaran umum dari sistem.

Levenshtein Distance

Teks atau input-an keyword/kosakata ini mempunyai cara kerja yaitu dengan user memasukkan keyword/kosakata yang ingin dicari dan dibandingkan dengan keyword/kosakata yang tersimpan di database sistem aplikasi ensiklopedia it (information technology).

Hasil proses antara string yang di masukkan yaitu keyword/kosakata dengan string (keyword/kosakata) yang tersimpan di database sistem aplikasi ensiklopedia it (information technology) maka akan dapat diketahui hasil terdekat yang memiliki kemiripan.

Penulis akan melakukan analisis bagaimana penerapan algoritma levenshtein distance dalam pencocokan string. Misalnya terdapat dua string yaitu string yang di masukkan dan string yang tersimpan di database.

String yang di masukkan (lev a) : case
String yang tersimpan di database (lev b) : cse
Algoritma Levenshtein Distance :

$$\text{Lev a, b (i,j)} = \begin{cases} \text{Max (i,j)} & \text{if min (i,j) = 0} \\ & \text{otherwise} \\ \text{Min Lev a, b (i-1, j) + 1} \\ \text{Min Lev a, b (i, j-1) + 1} \\ \text{Lev a, b (i-1, j-1) + 1} & \text{if } a(i) \neq b(j) \\ & \text{if } a(i) = b(j) \end{cases}$$

lev a,b adalah matriks lev a,b
i adalah baris matriks
j adalah kolom matriks

Tabel 1. Tabel Hasil Dari Proses levenshtein Distance :

		C	A	S	E
	0	1	2	3	4
C	1	0	1	2	3
S	2	1	1	1	2
E	3	2	2	2	1

Contoh dari perhitungan levenshtein distance menggunakan 2 string yang berbeda kemudian dihitung edit distance-nya pada tabel 3.1. Dapat dilihat hasil perhitungan edit distance antara 2 string "case" dan "cse" adalah 1. Pengecekan dimulai dari iterasi awal dari kedua string kemudian dilakukan operasi penambahan, penyisipan dan penghapusan. Nilai edit distance-nya yaitu pada ujung kanan bawah matriks. Hanya ada satu proses penyisipan yang dilakukan yaitu penyisipan karakter "a" pada string "case" sehingga menjadi "cse". Pada kasus pengecekan ejaan proses perhitungan ini dilakukan sejumlah kata yang ada pada basis data. Tentu saja untuk saran yang terbaik dibutuhkan daftar keyword/kosakata yang lengkap. Sehingga kata yang disarankan bisa mendekati yang diharapkan oleh pengguna.

Evaluasi Uji Perhitungan Sistem

Berikut penjelasannya, Proses algoritma levenshtein distance adalah sebagai berikut:

- Membuat matrik dengan ukuran (lev a + 1) x (lev b + 1)
- Melakukan pencocokan dengan melakukan perbandingan dari setiap karakter lev a dengan karakter lev b. Iterasi (1,1)
- Jika karakter 1 lev a dengan karakter 1 lev b sama maka nilai pada cell (1,1) diisi dengan 0 yang

merupakan nilai matrik $(x-1, y-1) + 0$ karena merupakan $a(i) = b(j)$

		C	A	S	E
	0	1	2	3	4
C	1	0			
S	2				
E	3				

Iterasi (1,2)

Jika karakter 1 lev a dengan karakter 2 lev b berbeda maka nilai cell didapatkan dari nilai terkecil dari:

- 1) Nilai cell $(x-1, y-1) + 1$ karena $a(i) \neq b(j)$, pada contoh diatas nilainya adalah 2.
- 2) Nilai cell $(x-1, y) + 1$, pada contoh diatas nilainya adalah 3.
- 3) Nilai cell $(x, y-1) + 1$, pada contoh diatas nilainya adalah 1.

Maka jarak antara karakter 1 lev a dengan karakter 2 lev b bernilai 1.

		C	A	S	E
	0	1	2	3	4
C	1	0			
S	2	1			
E	3				

Iterasi (3,1)

Jika karakter 3 lev a dengan karakter 1 lev b berbeda maka nilai cell didapatkan dari nilai terkecil dari:

- 1) Nilai cell $(x-1, y-1) + 1$ karena $a(i) \neq b(j)$, pada contoh diatas nilainya adalah 3.
- 2) Nilai cell $(x-1, y) + 1$, pada contoh diatas nilainya adalah 2.
- 3) Nilai cell $(x, y-1) + 1$, pada contoh diatas nilainya adalah 4.

Maka jarak antara karakter 3 lev a dengan karakter 1 lev b bernilai 2.

		C	A	S	E
	0	1	2	3	4
C	1	0	1	2	
S	2	1	1		
E	3	2	2		

Iterasi (3,2)

Jika karakter 1 lev a dengan karakter 2 lev b sama maka nilai cell didapatkan dari nilai terkecil dari:

- 1) Nilai cell $(x-1, y-1) + 0$ karena $a(i) = b(j)$, pada contoh diatas nilainya adalah 1 .
- 2) Nilai cell $(x-1, y) + 0$, pada contoh diatas nilainya adalah 1.
- 3) Nilai cell $(x, y-1) + 0$, pada contoh diatas nilainya adalah 2.

Maka jarak antara karakter 3 lev a dengan karakter 2 lev b bernilai 1.

		C	A	S	E
	0	1	2	3	4
C	1	0	1	2	
S	2	1	1	1	
E	3	2	2		

Iterasi (3,3)

Jika karakter 3 lev a dengan karakter 3 lev b berbeda maka nilai cell didapatkan dari nilai terkecil dari:

- 1) Nilai cell $(x-1, y-1) + 1$ karena $a(i) \neq b(j)$, pada contoh diatas nilainya adalah 2.
- 2) Nilai cell $(x-1, y) + 1$, pada contoh diatas nilainya adalah 3.
- 3) Nilai cell $(x, y-1) + 1$, pada contoh diatas nilainya adalah 2.

Maka jarak antara karakter 3 lev a dengan karakter 3 lev b bernilai 2.

		C	A	S	E
	0	1	2	3	4
C	1	0	1	2	
S	2	1	1	1	
E	3	2	2	2	

Iterasi (4,1)

Jika karakter 4 lev a dengan karakter 1 lev b berbeda maka nilai cell didapatkan dari nilai terkecil dari:

1) Nilai cell $(x-1, y-1) + 1$ karena $a(i) \neq b(j)$, pada contoh diatas nilainya adalah 4.

2) Nilai cell $(x-1, y) + 1$, pada contoh diatas nilainya adalah 3.

3) Nilai cell $(x, y-1) + 1$, pada contoh diatas nilainya adalah 5.

Maka jarak antara karakter 4 lev a dengan karakter 1 lev b bernilai 3.

		C	A	S	E
	0	1	2	3	4
C	1	0	1	2	3
S	2	1	1	1	
E	3	2	2	2	

Iterasi (4,2)

Jika karakter 4 lev a dengan karakter 2 lev b berbeda maka nilai cell didapatkan dari nilai terkecil dari:

1) Nilai cell $(x-1, y-1) + 1$ karena $a(i) \neq b(j)$, pada contoh diatas nilainya adalah 3.

2) Nilai cell $(x-1, y) + 1$, pada contoh diatas nilainya adalah 2.

3) Nilai cell $(x, y-1) + 1$, pada contoh diatas nilainya adalah 4.

Maka jarak antara karakter 4 lev a dengan karakter 2 lev b bernilai 2.

		C	A	S	E
	0	1	2	3	4
C	1	0	1	2	3
S	2	1	1	1	2
E	3	2	2	2	

Iterasi (4,3)

Jika karakter 4 lev a dengan karakter 3 lev b sama maka nilai cell didapatkan dari nilai terkecil dari:

1) Nilai cell $(x-1, y-1) + 0$ karena $a(i) = b(j)$, pada contoh diatas nilainya adalah 1.

2) Nilai cell $(x-1, y) + 0$, pada contoh diatas nilainya adalah 2.

3) Nilai cell $(x, y-1) + 0$, pada contoh diatas nilainya adalah 2.

Maka jarak antara karakter 4 lev a dengan karakter 3 lev b bernilai 1.

		C	A	S	E
	0	1	2	3	4
C	1	0	1	2	3
S	2	1	1	1	2
E	3	2	2	2	1

Setelah semua proses iterasi selesai maka didapatkan jarak *string* yang di masukkan (*lev a*) dengan *string* yang tersimpan di *database* (*lev b*) pada cell (3,4) adalah sebesar 1 atau *edit distance* = 1, jadi jarak levenshtein antara *string* dari *lev a* yaitu cse dan *string lev b* yaitu case adalah 1.

HASIL UJI COBA DAN PEMBAHASAN

Pembahasan Program

Form Searching Kategori

Form ini menjelaskan sebelum proses pencarian *users* diberi pilihan tentang klasifikasi sesuai kategori yang bertujuan untuk memudahkan *users* dalam proses pencarian. Tampilan *form searching* kategori dapat dilihat pada gambar 3. di bawah ini.



Gambar 3. Form Searching Kategori

Form Searching

Setelah memilih kategori muncul halaman ini yang menampilkan tempat pencarian istilah teknologi informasi yang memasukkan *keyword*/kosakata berupa *text* dan tombol *search* yang berfungsi untuk memproses dan menampilkan hasil dari *searching* di halaman atau *form searched*. Tampilan *form searching* dapat dilihat pada gambar 4. di bawah ini.



Gambar 4. Form Searching

Form Searched

Pada *form searched* merupakan proses pencarian dan hasil dari *searching* yang menampilkan penjelasan dari istilah atau *keyword*/kosakata yang dicari dalam bahasa Indonesia dan bahasa Inggris, dan kategori sebagai klasifikasi sesuai jenis dan juga gambar. Tampilan *form searched* dapat dilihat pada gambar 5. di bawah ini.



Gambar 5. Form Searched

SIMPULAN DAN SARAN

A. Simpulan

Perancangan sistem aplikasi ensiklopedia it (*information technology*) ini menggunakan *spelling checker* dengan algoritma *levenshtein distance* dan menggunakan metode pengumpulan data dari studi literatur. Sedangkan metode pengembangan perangkat lunak menggunakan paradigma dengan metode *waterfall* yang meliputi beberapa proses.

Simpulan dari hasil perancangan sistem aplikasi ensiklopedia it (*information technology*) ini dengan menerapkan algoritma *levenshtein distance* adalah :

1. Metode *levenshtein distance* dapat diterapkan untuk melakukan proses pengkoreksian kata kunci pada aplikasi pencarian berbasis web.

2. Sistem dapat memproses inputan kata kunci dalam jumlah yang banyak dengan durasi waktu eksekusi yang tergantung dari kompleksitas kata kunci yang di *input*-kan.
3. Dapat membantu mengatasi permasalahan pada kesalahan pengetikan dengan mekanisme penambahan, penyisipan dan penghapusan karakter.
4. Algoritma *levenshtein distance* akan bekerja lebih baik jika kedua data yang dibandingkan mempunyai urutan kata yang sama.
5. Algoritma *levenshtein distance* akan lama dalam proses jika kedua data yang dibandingkan mempunyai urutan kata yang tidak sama dan banyak karakter yang berbeda.

Sistem aplikasi ensiklopedia it (*information technology*) ini sebagai salah satu sumber referensi yang dapat mempercepat dalam pencarian tentang istilah teknologi informasi.

B. Saran

Untuk penelitian lebih lanjut tentang penelitian ini perlu ditambahkan beberapa pengembangan diantaranya dalam pembuatan sistem aplikasi ensiklopedia it (*information technology*) ini antara lain :

1. Tentu saja untuk saran yang terbaik dibutuhkan daftar *keyword*/kosakata yang lengkap, sehingga kata yang disarankan bisa mendekati yang diharapkan oleh pengguna.
2. Optimasi kata perbaikan yang diberikan sistem dapat ditingkatkan dengan mengimplementasikan metode-metode yang lebih efektif dan efisien.
3. Sebaiknya menambahkan atau menggunakan *preprocessing similarity* akan menjadi lebih baik dalam pendeteksian ke akuratan pencarian kata yang ingin di cari, akan tetapi juga memberikan efek terhadap lamanya proses pendeteksian.
4. Dalam sistem aplikasi ini tentu masih banyak kekurangan dan kelemahannya, untuk itu peneliti mohon kritik dan saran membangun dalam pengembangan dan perbaikan sistem selanjutnya.

DAFTAR PUSTAKA

- Adiwiidya, B.M.D. 2009. "Algoritma Levenshtein Dalam Pendekatan Approximate String Matching". Bandung : Institut Teknologi Bandung.
- Adriyani, Ni Made Muni. 2012. IMPLEMENTASI ALGORITMA LEVENSHTein DISTANCE DAN METODE EMPIRIS UNTUK MENAMPILKAN SARAN PERBAIKAN KESALAHAN PENGETIKAN DOKUMEN BERBAHASA INDONESIA. Bali: Program Studi Teknik Informatika, Jurusan Ilmu Komputer, Fakultas Matematika Dan Ilmu Pengetahuan Alam, Universitas Udayana.
- Amijaya, Gilang Rizky. 2010. "PENGARUH PERSEPSI TEKNOLOGI INFORMASI, KEMUDAHAN, RESIKO DAN FITUR LAYANAN TERHADAP MINAT ULANG NASABAH BANK DALAM MENGGUNAKAN INTERNET BANKING (Studi Pada Nasabah Bank BCA)". Bandung : Informatika.
- Anonim "Pengertian Mesin Pencarian" http://id.wikipedia.org/wiki/Mesin_pencari_web Diakses pada tahun 2015.
- Atmajaya, Gede Esa Deva. 2008. *PEMBUATAN SPELLINGCHECKER UNTUK BAHASA INDONESIA DENGAN JAVA 2 STANDARD EDITION*. Depok : Informatika.
- Basuki, Murya Arief. 2009. *ANALISA WEBSITE UNIVERSITAS MURIA KUDUS*. Kudus: Dosen Fakultas Teknik Universitas Muria Kudus.
- Iskandar, Lydia. 2012. *ANALISA KEPUASAN MAHASISWA TERHADAP WEBSITE UNIVERSITAS ATMAJAYA DAN HUBUNGANNYA DENGAN LOYALITAS MAHASISWA*. Universitas Gunadarma.
- Junedy.S, Richard. 2014. "PERANCANGAN APLIKASI DETEKSI KEMIRIPAN ISI DOKUMEN TEKS DENGAN MENGGUNAKAN METODE LEVENSHTein DISTANCE". Medan: STMik Budi Darma.
- Kadir, Abdul. 2006. *Dasar Aplikasi Database MySql Delphi*. Yogyakarta: Andi Offset.
- Kadir, A & Triwahyuni, T.C. (2003). *Pengenalan Teknologi Informasi*, Yogyakarta: Andi Offset.
- Kristanto, Harianto. 1994. *Konsep dan Perancangan Database*. Yogyakarta: Andi Offset.
- Kristanto, Harianto. 1994. *Konsep dan Perancangan Database*. Yogyakarta: Andi Offset.
- Maseleno, Andino. 2003. "Kamus Istilah Komputer dan Informatika". Yogyakarta : Informatika.
- Nugroho, Bunafit. 2004. *PHP & MySQL dengan Editor Dreamweaver MX*. Yogyakarta: Andi Offset.
- Prawiyanti, Adita Ayu. 2013. *PERANCANGAN SISTEM INFORMASI INVENTARIS PROGRAM STUDI TEKNIK INFORMATIKA UNIVERSITAS SURAKARTA*. Surakarta: Program Studi Teknik Informatika, Universitas Surakarta.
- Purwanto, Eddy. 2012. *PERBANDINGAN STRATEGI REPLIKASI PADA SISTEM BASIS DATA TERDISTRIBUSI*. Palembang: Jurusan Teknik Informatika Universitas Bina Darma.
- Saputra, Iwan. 2013. "Aplikasi SMS Filtering Pada Smartphone Android dengan Menggunakan Metode Levenshtein distance". Bandung : Informatika. FT UMRAH.
- Zibriel, Igel. 2008. "*Ensiklopedia Nusantara Menggunakan Orientasi Web 2.0*". Bandung : Informatika.