IMPLEMENTASI ALGORITMA LEVENSHTEIN DISTANCE DAN METODE EMPIRIS UNTUK MENAMPILKAN SARAN PERBAIKAN KESALAHAN PENGETIKAN DOKUMEN BERBAHASA INDONESIA

Ni Made Muni Adriyani, I Wayan Santiyasa, Agus Muliantara
Program Studi Teknik Informatika,
Jurusan Ilmu Komputer,
Fakultas Matematika Dan Ilmu Pengetahuan Alam,
Universitas Udayana

Email: muni.adriyani@cs.unud.ac.id

ABSTRAK

Bahasa merupakan alat komunikasi lingual manusia baik secara lisan maupun tulisan. Peran Bahasa Indonesia yang baik dan benar sebagai bahasa resmi nasional memiliki arti yang sangat penting. Apalagi dalam penyusunan karya ilmiah. Hal ini disebabkan adanya data dan informasi yang ada didalamnya digunakan sebagai acuan untuk penelitian atau pengkajian selanjutnya bagi ilmuan lainnya. Hal itu tentu saja menimbulkan beberapa masalah seperti kesalahan pengetikan. Kontrol pegecekan bahasa yang baik dan benar jika dilakukan secara manual pastinya akan menghabiskan banyak waktu, apalagi dijaman 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 Bahasa Indonesia yang dikembangkan dengan berbasis web menggunakan bahasa pemrograman *PHP* dan menggunakan basis data kumpulan kata berbahasa Indonesia yang mengacu pada KBBI. Algoritma yang digunakan untuk memberikan saran perbaikan adalah *Levenshtein Distance* yang dapat menghitung keterkaitan antar string dan menghitung jumlah keterbedaan antar dua string. Disamping itu kesalahan penulisan juga dapat disebabkan kurangnya spasi antar kata sehingga kata tersebut tidak mengandung makna maka dengan menerapkan metode empiris diharapkan dapat menjadi jalan keluarnya.

Kata Kunci : Kesalahan Penulisan, Berbasis Web, *Levenshtein Distance*, Metode Empiris

ABSTRACT

Language is a tool of human lingual communication both orally and in writing. Indonesian role is good and true as the official national language has very important meaning. Especially in the preparation of scientific papers. This is due to the absence of data and information contained there in is used as a reference for further studies or analyzes for other scientists. That certainly raises some issues such as typing errors. Checking control language is good and right if done manually but, will certainly spend a lot of time, let alone an age of modern versatile as it is today.

Writing errors are generally caused by the proximity of the location of the keyboard, slip a finger, or because the two characters are located swapped. To the problems that came the idea to create a System Checking Spelling Indonesian language that was developed by using a web-based programming language PHP and uses a basis data collection of Indonesian-language word that refers to KBBI. The algorithm used to provide suggestions for improvement is the Levenshtein Distance can calculate the relation between the string and count the number of keterbedaan between two strings. Besides writing errors can also be caused by a lack of spaces between words so that word does not contain the meaning of it by applying the empirical method is expected to be a way out.

Kata Kunci : Error Writing, Web-Based, Levenshtein Distance, Empirical Methods

1. PENDAHULUAN

Kesalahan pengetikan dokumen memang sering sekali terjadi. Apalagi belakangan ini kesadaran masyarakat untuk menuangkan idenya ke dalam artikel, jurnal ilmiah, tugas kuliah ataupun dokumen lainnya mengalami peningkatan. Tentu saja dalam penulisan dokumen tersebut adanya kesalahan pengetikan yang disebabkan oleh beberapa faktor seperti :

- Letak huruf pada *keyboard* yang berdekatan,
- Kesalahan karena kegagalan mekanis atau slip dari 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 memang salah dalam proses penulisannya. Efisiensi waktu yang dibutuhkan jika dilakukan dengan manual tentunya tidak akan optimal dan cukup membosankan sehingga kemungkinan adanya *human error* dapat mengakibatkan proses pengecekan kata menjadi tidak optimal(Wardiana, 2002).

Beberapa algoritma pun dapat diimplementasikan dalam memberikan kata saran yang paling mendekati dari kata yang salah pengetikannya. Salah satunya algoritma *Levenshtein Distance* yang dapat menghitung jarak keterbedaan antara dua string (Andhika,2010). Selain itu ada juga kesalahan pengetikan karena kurangnya spasi sehingga kata tersebut tidak memiliki arti. Untuk permasalahan kata yang berdempetan tersebut digunakan metode empiris yaitu dengan cara memecah kata menjadi dua bagian dan mencocokkan kata ke dalam basis data dan memberikan saran kemungkinan adanya kata yang diketik tanpa spasi.

2. ALGORITMA LEVENSHTEIN DAN METODE EMPIRIS

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 :

1. Operasi Pengubahan Karakter

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

2. Operasi Penambahan Karakter

Operasi penambahan karakter berarti menambahkan 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 bisa ditambahkan diawal maupun disisipkan di tengah *string* .

3. Operasi Penghapusan Karakter

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

Algoritma ini berjalan mulai dari pojok kiri atas sebuah array dua dimensi yang telah diisi sejumlah karakter sring awal dan string target dan diberikan nilai *cost.* Nilai *cost* pada ujung kanan bawah menjadi nilai *edit distance* yang menggambarkan jumlah perbedaan dua string.

		s	a	y	a
	0	1	2	3	4
s	1	0	1	2	3
y	2	1	2	1	2
a	3	2	1	2	1

Gambar 2.1. Tabel matriks perhitungan *Edit distance*

Contoh dari perhitungan levenshtein menggunakan 2 string yang berbeda kemudian dihitung *edit distance*-nya pada tabel 1. Dapat dilihat hasil perhitungan *edit distance* antara 2 string ‘sya’ dan ‘saya’ 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* ‘sya’ sehingga menjadi ‘saya.’ Pada kasus pengecekan ejaan proses perhitungan ini dilakukan sejumlah kata yang ada pada basis data. Tentu saja untuk saran yang terbaik dibutuhkan daftar kata berhasa Indonesia yang lengkap. Sehingga kata yang disarankan bisa mendekati yang diharapkan oleh pengguna.

Selain menggunakan algoritma levenshtein ada satu metode lain yang digunakan dalam koreksi ejaan ini. Metode ini bekerja dengan cara memisahkan string menjadi beberapa kemungkinan kata kemudian dicocokkan ke dalam basis data. Misalnya saja penulisan kata ‘burungnuri’. Jika dicari dalam kamus Bahasa Indonesia tentu saja tidak memiliki arti. Cara kerja metode ini adalah mencoba semua kemungkinan.

Dari Tabel 2.2. Dapat dilihat bahwa kata ‘burung’ dan ‘nuri’ memiliki arti dalam Bahasa Indonesia dan terdaftar dalam basis data. Sehingga saran perbaikannya menjadi ‘burung nuri’. Pertama string dipecah dengan menyisipkan spasi. String ‘burung nuri’ akan menjadi :

String 1	String 2
b	urungnuri
bu	rungnuri
bur	ungnuri
buru	ngnuri
burun	gnuri
burung	nuri
burungn	uri
burungnu	ri
burungnur	i

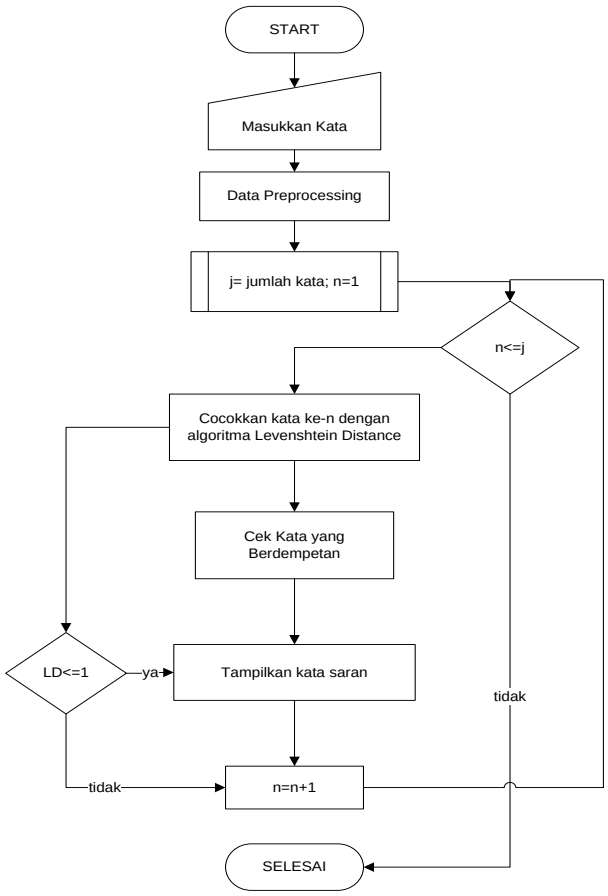
Gambar 2.2. Pencocokan kata yang telah dipecah

3. IMPLEMENTASI SISTEM

3.1 Perancangan sistem

Dalam tahap implementasi dilakukan pencarian dan pengumpulan informasi yang dibutuhkan selama perancangan sistem. Metode yang digunakan untuk mengumpulkan informasi dan data adalah metode studi literatur, yaitu dengan mempelajari literatur yang terkait dengan penelitian, antara lain mengenai Algoritma *Levenshtein Distanse* dan penerapannya pada sistem pengecekan ejaan berbahasa Indonesia, bahasa pemrograman PHP, HTML dan Javascript.

Pengecekan dilakukan per kata, kalimat masukan akan masuk ke dalam tahap *preprocessing* terlebih dahulu, sebelum diproses lebih lanjut. Proses *preprocessing*-nya meliputi penghilangan tanda baca, tokenisasi terhadap masing-masing kata. Kemudian setiap token akan dicocokkan ke basis data menggunakan Algoritma *Levenshtein Distance* dan Metode Empiris. Setelah dilakukan perhitungan maka sistem akan menampilkan kata saran yang mendekati kesalahan penulisan. Diagram alir Sistem pengecekan ejaan kata adalah sebagai berikut.



Gambar 3.1. *Flowchart* Sistem Pengecekan Ejaan

3.2 Perancangan Basis Data

Perancangan basis data meliputi perancangan tabel yang akan digunakan untuk menyimpan data kata Berbahasa Indonesia yang dapat diunduh dari http://indodic.com/ IndoWordList. zip. Referensi kata yang tersedia

cukup banyak yaitu 41,057 kata Berbahasa Indonesia.

3.3 Implementasi Algoritma Levenshtein Distance Dan Metode Empiris

a. Data *Preprocessing*

Data preprocessing meliputi pengecekan karakter masukan apakah termasuk masukan berupa huruf, angka atau simbol. Berikut ini adalah potongan kode proses tokenisasi.

b. Proses Tokenisasi

```
$panjang=strlen($a);
$c=0;
$d=0;
$kata="";
for ($i=0;$i<$panjang;$i++)
{
   if(strcmp($a[$i]," ")==0)
   {
     $kata="";
     for($j=$d; $j<$i; $j++)
   {
     $kata=$kata."".$a[$j];
   }
     $b[$c] = $kata;
     $c=$c+1;
     $b[$c] = $a[$i];
     $c=$c+1;
     $d = $i+1;
   }
   else
   {
     $kata="";
for($j=$d; $j<$panjang; $j++)
   {
   $kata=$kata."".$a[$j];
   $b[$c] = $kata;
   }
}
}
return $b;
```

b. Algoritma Levenshtein Distance

```
$m[$i][$j]=0;
for($i=1;$i<=$x;$i++){
      $m[$i][0]= $i;
}
for($j=1;$j<=$y;$j++){
      $m[0][$j]=$j;
}
for($i=1;$i<=$x;$i++)
{
for($j=1;$j<=$y;$j++)
{
      if($sb1[$i]==$sb2[$j]){
      $cost=0;
            }
      else{
      $cost=1;
            }
      $m1=$m[$i-1][$j-1]+ $cost;
      $m2=$m[$i-1][$j]+1;
      $m3=$m[$i][$j-1]+ 1;
      $m[$i][$j]=min($m3,$m2,$m1);
        if ($i==$x and $j==$y)
          {
           $minimum = $m[$i][$j];
           if ($m[$i][$j]<=1){
            $katasaran[$z]=$kata2;
            $z=$z+1;
            }
          }
 }
}
return $katasaran;
```

d. Metode Empiris

```
function pecah ()
{
Inisialisasi $teks,$panjang_teks;
for($i=0;$i<1;$i++)
      {
      $b=($jumlah+1)-$i;
for($j=1;$j<$b;$j++)
      {
$text=substr($data['test'],$i,$j);
$text2=substr($data['test'], $j, $b);
```

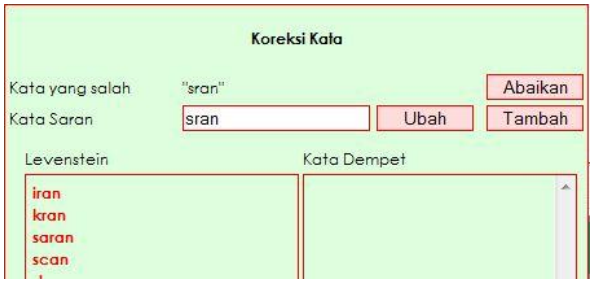
```
if(cocok($text)==1)
	{
$datatext = array($text,$text2);
	}
}
echo "<br>";
}
	return $datatext;
}
function cocok($teks)
{
$hasil=0;
$query= mysql_query("SELECT *
FROM daftar_kata where
daftar_kata='$teks'");
$data=mysql_fetch_assoc($query);
if($data['daftar_kata'] != NULL)
	{
		$hasil=1;
	}
	return $hasil;
}
```

4. HASIL DAN PEMBAHASAN

Sistem ini telah diimplementasikan dengan pemrograman berbasis web sehingga memudahkan pengguna jika ingin diakses dari mana saja. Pada halaman utama sistem, pengguna dapat memasukkan inputan kalimat, menekan tombol *submit* dan kalimat masukan akan diproses oleh sistem.

Gambar4.1 adalah tampilan awal Sistem Pengecekan Ejaan Berbahasa Indonesia. Pengguna dapat mengetikkan kata atau kalimat yang akan dicek kebenarannya, kemudian klik tombol submit. Maka hasil pengecekan akan muncul di kolom bawahnya. Tulisan yang salah ketik akan ditandai dengan warna merah. Jika terdapat kesalahan penulisan akan tampil pop up beperti gambar 4.2 yang menampilkan kesalahan dan saran perbaikan penulisan.

Gambar 4.1 Halaman Utama Sistem



Gambar 4.2 Halaman yang Menampilkan Kata Saran

Pengguna hanya perlu memilih kata saran yang dianggap paling benar pada sistem tekan tombol ubah, kemudian sistem akan mengganti kata yang salah ketik dengan kata pilihan pengguna. Jika kata dianggap salah dan sistem tidak dapat menampilkan saran, pengguna bisa menambahkan kata tersebut ke dalam basis data dengan menekan tombol tambah. Jika pengguna merasa tidak ada kesalahan tetapi sistem menandai kata itu salah, pengguna bisa memilih tombol abaikan.

5. KESIMPULAN

Dari pembahasan yang sudah dilakukan untuk mengimplementasikan algoritma *Levenshtein Distance* dan Metode Empiris dalam membangun Sistem Pengecekan Ejaan dapat ditarik kesimpulan sebagai berikut :

1. Dengan menerapkan Algoritma Levenshtein Distance dapat membantu mengatasi permasalahan pada kesalahan pengetikan dengan mekanisme penambahan, penyisipan dan penghapusan karakter.
2. Optimasi kata perbaikan yang diberikan sistem dapat ditingkatkan dengan mengimplementasikan Metode Empiris untuk mengetahui adanya kata yang ditulis tanpa spasi, sehingga saran yang diberikan bisa mencapai harapan dari pengguna.
3. Pengecekan ini masih sebatas pengecekan kesalahan pengetikan, bukan pengecekan pola kalimat Berbahasa Indonesia.

6. UCAPAN TERIMA KASIH

Sehubungan dengan telah terselesaikannya penelitian ini, maka penullis mengucapkan terima kasih kepada berbagai pihak yang telah membantu, antara lain:

1. Bapak Drs. I Wayan Santiyasa, M.Si., selaku Ketua Jurusan Ilmu Komputer Fakultas MIPA Universitas Udayana dan sebagai pembimbing dalam penyelesaian jurnal ini.
2. Bapak Agus Muliantara S.Kom, M.Kom., selaku Pembimbing yang telah banyak memberikan bimbingan, saran dan motivasi dalam penulisan.
3. Seluruh dosen, staf pegawai, dan rekan rekan mahasiswa di Jurusan Ilmu Komputer Fakultas MIPA Universitas Udayana yang telah meluangkan waktu untuk memberikan saran dan masukan untuk menyempurnakan penelitian ini.

Dalam Penulisan makalah ini penulis merasa masih banyak kekurangan baik pada teknis penulisan maupun materi. Untuk itu kritik dan saran dari semua pihak sangat penulis harapkan demi penyempurnaan pembuatan jurnal ini.

7. DAFTAR PUSTAKA

1. Ilmy, M.B., Rahmi, N., Bu'ulölö R.L. 2006. "*Penerapan Algoritma Levenshtein Distance untuk Mengoreksi Kesalahan Pengejaan pada Editor Teks".* Bandung : Institut Teknologi Bandung.

2. Andhika, Fatardhi Rizky. 2010. "*Penerapan String Suggestion dengan Algoritma Levenshtein Distance dan Alternatif Algoritma Lain dalam Aplikasi*". Bandung : Institut Teknologi Bandung.

3. Adiwidya, B.M.D. 2009. "*Algoritma Levenshtein Dalam Pendekatan Approximate String Matching*". Bandung : Institut Teknologi Bandung.