

IMPLEMENTASI *NATURAL LANGUAGE PROCESSING* DALAM PENGUKURAN KETEPATAN EJAAN YANG DISEMPURNAKAN (EYD) PADA ABSTRAK SKRIPSI MENGGUNAKAN ALGORITMA *FUZZY LOGIC*

Nisa Kurniasih Wangsanegara¹, Beki Subaeki²

^{1,2}Jurusan Teknik Informatika, Fakultas Sains dan Teknologi
UIN Sunan Gunung Djati Bandung
Email : nisawangsa@gmail.com, beki807@gmail.com

ABSTRAK

Ejaan yang Disempurnakan merupakan salah satu aspek penting dalam penulisan suatu dokumen. Penggunaan ejaan harus sesuai dengan yang tertera pada aturan baku yang dikeluarkan oleh Kementerian Pendidikan Nasional. Kesalahan yang banyak terjadi yaitu dalam penulisan kata, tanda baca, dan huruf kapital. Aplikasi ini akan mengidentifikasi dan menghitung jumlah kesalahan penulisan huruf kapital/ kata dan tanda baca. Pengukur penggunaan ketepatan EYD ini dibuat menggunakan metode *Fuzzy Logic Tsukamoto*. Proses yang dilakukan dalam metode Tsukamoto yaitu: fuzzifikasi, pembentukan *rule*, mesin inferensi, dan defuzzifikasi. Jumlah kata yang saat ini tersedia dalam aplikasi ini yaitu 31.759 kata yang sebagian besar diambil dari Kamus Besar Bahasa Indonesia. Aplikasi ini dibangun menggunakan MySQL sebagai *database* dan menggunakan *tools* PHP yang berbasis *website*. Berdasarkan pengujian yang dilakukan terhadap 20 abstrak skripsi, presentase kesesuaian hasil identifikasi yaitu 70% sesuai dengan hasil identifikasi secara manual.

Kata kunci: *EYD, penulisan, Fuzzy Logic Tsukamoto, PHP.*

ABSTRACT

Enhanced Spelling is one of the important aspects of writing a document. The use of spelling must match those on the standard rules issued by the Ministry of National Education. Errors that occur are in spelling, punctuation, and capitalization. This app will identify and count the number of writing errors capital letters / words and punctuation. Estimator use EYD accuracy is made using the method of Fuzzy Logic Tsukamoto. Process conducted in Tsukamoto methods are: fuzzification, the establishment of the rule, the inference engine and defuzzification. The number of words that are currently available in this application, namely 31 759 words, mostly taken from Indonesian dictionary. This application is built using MySQL as the database and using tools PHP-based website. Based on tests performed on 20 abstract thesis, namely the identification of conformity percentage of 70% in accordance with the results of the identification manually.

Kata kunci: *EYD, penulisan, Fuzzy Logic Tsukamoto, PHP.*

I. Pendahuluan

EYD merupakan kependekan dari Ejaan Yang Disempurnakan. EYD memuat tentang tata bahasa Indonesia yang baik dan benar. EYD diresmikan pada tanggal 16 Agustus 1972 oleh Presiden Republik Indonesia. Peresmian itu berdasarkan Putusan Presiden No. 57 Tahun 1972. EYD mengalami beberapa kali revisi, yaitu pada tahun 1987 melalui keputusan Menteri Pendidikan dan Kebudayaan Nomor 0543a/U/1987 tanggal 9 September 1987 tentang Penyempurnaan Pedoman Umum Ejaan Bahasa Indonesia yang Disempurnakan dan tahun 2009 melalui keputusan Menteri Pendidikan Nasional yang mengeluarkan Permen (Peraturan Menteri) Pendidikan Nasional Nomor 46 Tahun 2009 tanggal 31 Juli 2009 tentang Pedoman Umum Ejaan Bahasa Indonesia yang Disempurnakan.

EYD merupakan salah satu aspek penting dalam pembuatan suatu teks baik berupa *essay*, jurnal, karya ilmiah, dan lainnya. Kesalahan penulisan yang biasanya terjadi yaitu pada penulisan huruf kapital, penggunaan tanda baca, dan penulisan kata.

Natural Language Processing (Pemrograman Bahasa Alami) adalah pembuatan program yang memiliki kemampuan untuk memahami bahasa manusia. Tujuan dari pemrograman bahasa alami adalah melakukan proses pembuatan model komputasi dari bahasa sehingga dapat terjadi suatu interaksi antara manusia dengan komputer dengan perantara bahasa alami.

Dalam bukunya [4] menuliskan bahwa konsep tentang logika fuzzy diperkenalkan oleh Prof. Lotfi Astor Zadeh pada tahun 1962. Logika fuzzy adalah metodologi sistem kontrol pemecahan masalah, yang cocok untuk diimplementasikan pada sistem, mulai dari sistem yang sederhana, sistem kecil, *embedded system*, jaringan PC, *multi-channel* atau *work station* berbasis akuisisi data, dan sistem kontrol.

II. Landasan Teori

a. *Natural Language Processing*

Natural Language Processing (Pemrograman Bahasa Alami) adalah pembuatan program yang memiliki kemampuan untuk memahami bahasa manusia. Pada prinsipnya bahasa alami adalah suatu bentuk representasi dari suatu pesan yang ingin dikomunikasikan antar manusia.

(NLP) adalah upaya untuk mengekstrak lebih jauh representasi dari suatu teks bebas. Hal ini dapat dimasukkan secara kasar seperti mencari siapa melakukan apa kepada siapa, kapan, di mana, bagaimana dan mengapa. NLP biasanya membuat penggunaan konsep-konsep linguistic seperti kata benda, kata kerja, kata sifat, dan lainnya dan struktur gramatikal (baik direpresentasikan sebagai ungkapan-ungkapan seperti frase nomina atau frase preposisional, atau hubungan ketergantungan seperti subjek dari- atau objek-dari) [1].

b. Ejaan Yang Disempurnakan (EYD)

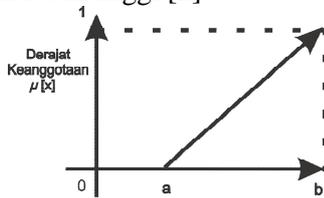
Ejaan yang Disempurnakan (EYD) merupakan kumpulan aturan-aturan yang mengatur penulisan kata beserta tanda baca dalam bahasa Indonesia. EYD diresmikan pada tanggal 16 Agustus 1972 oleh Presiden Republik Indonesia. Peresmian itu berdasarkan Putusan Presiden No. 57 Tahun 1972.

EYD mengalami beberapa kali revisi, yaitu pada tahun 1987 melalui keputusan Menteri Pendidikan dan Kebudayaan Nomor 0543a/U/1987 tanggal 9 September 1987 tentang Penyempurnaan Pedoman Umum Ejaan Bahasa Indonesia yang Disempurnakan dan tahun 2009 melalui keputusan Menteri Pendidikan Nasional yang mengeluarkan Permen (Peraturan Menteri) Pendidikan Nasional Nomor 46 Tahun 2009 tanggal 31 Juli 2009 tentang Pedoman Umum Ejaan Bahasa Indonesia yang Disempurnakan. Dalam Peraturan Menteri Pendidikan Nasional Nomor 46 Tahun 2009 tersebut dijelaskan mengenai aturan-aturan penulisan dalam Bahasa Indonesia diantaranya penulisan huruf kapital, penggunaan tanda baca [3].

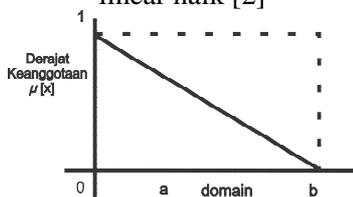
c. *Fuzzy Logic Tsukamoto*

Logika fuzzy adalah metodologi sistem kontrol pemecahan masalah, yang cocok untuk diimplementasikan pada sistem, mulai dari sistem yang sederhana, sistem kecil, *embedded system*, jaringan PC, *multi-channel* atau *work station* berbasis akuisisi data, dan sistem kontrol [4]. Logika fuzzy dikenalkan oleh Prof. Lotfi Astor Zadeh pada tahun 1962. Logika fuzzy memungkinkan memiliki derajat keanggotaan suatu variable berada pada interval 0-1. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi. Beberapa fungsi yang sering digunakan yaitu grafik keanggotaan

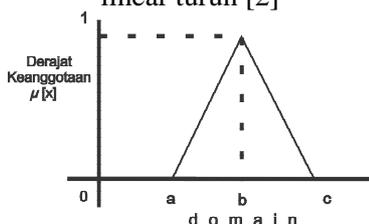
kurva linear dan grafik keanggotaan kurva segitiga. Pada grafik keanggotaan kurva linear naik (Gambar 1), kenaikan himpunan dimulai pada nilai domain yang memiliki derajat keanggotaan nol (0) bergerak kekanan menuju nilai domain yang memiliki derajat keanggotaan lebih tinggi [2].



Gambar 1 Grafik keanggotaan kurva linear naik [2]



Gambar 2 Grafik keanggotaan kurva linear turun [2]

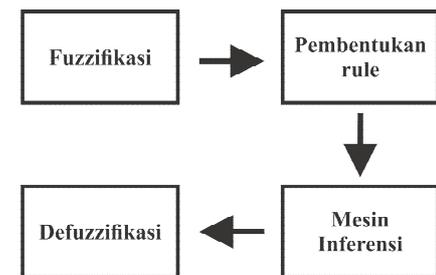


Gambar 3 Grafik Keanggotaan Segitiga [2]

Fungsi keanggotaan:

$$\mu[x] = \begin{cases} 0; & x \leq a \text{ atau } x \geq c \\ \frac{x-a}{b-a}; & a \leq x \leq b \\ \frac{c-x}{c-b}; & b \leq x \leq c \end{cases}$$

Alur kerja logika fuzzy dimulai dengan fuzzifikasi, pembentukan rule, mesin inferensi, dan defuzzifikasi. Cara kerja logika fuzzy dapat dilihat pada Gambar 4.



Gambar 4 Cara Kerja Logika Fuzzy

Algoritma ini diterapkan pada saat penghitungan nilai penulisan abstrak skripsi.

III. Analisis dan Perancangan

Proses pada logika fuzzy yaitu fuzzifikasi, pembentukan aturan, mesin inferensi, dan defuzzifikasi.

a. Fuzzifikasi

1. Fungsi keanggotaan SP

$$\mu_{\text{sedikit}} [x] = \begin{cases} 1, & x \leq 76 \\ \frac{114-x}{38}; & 76 < x < 114 \\ 0, & x \geq 114 \end{cases}$$

$$\mu_{\text{cukup}} [x] = \begin{cases} 0; & x \leq 76 \text{ atau } x \geq 228 \\ \frac{x-76}{38}; & 76 < x < 114 \\ \frac{114-x}{114}; & 114 \leq x \leq 228 \end{cases}$$

$$\mu_{\text{banyak}} [x] = \begin{cases} \frac{x-76}{38}; & 114 \leq x \leq 228 \\ 0; & x < 114 \end{cases}$$

Nilai keanggotaan untuk kesalahan penulisan input 5 adalah:

$$\begin{aligned} \mu_{\text{sp}}_{\text{banyak}} [5] &= 0 \\ \mu_{\text{sp}}_{\text{cukup}} [5] &= 0 \\ \mu_{\text{sp}}_{\text{sedikit}} [5] &= 1 \end{aligned}$$

2. Fungsi keanggotaan ST

$$\mu_{\text{sedikit}} [x] = \begin{cases} 1, & x \leq 7 \\ \frac{11-x}{4}; & 7 < x \leq 11 \\ 0, & x > 11 \end{cases}$$

$$\mu_{\text{cukup}} [x] = \begin{cases} 0; & x \leq 7 \text{ atau } x \geq 22 \\ \frac{x-7}{4}; & 7 < x < 11 \\ \frac{11-x}{11}; & 11 \leq x \leq 22 \end{cases}$$

$$\mu_{\text{banyak}} [x] = \begin{cases} \frac{x-7}{4}; & 11 \leq x \leq 22 \\ 0; & x < 11 \end{cases}$$

Nilai Keanggotaan untuk kesalahan tanda baca input 0 adalah:

$$\begin{aligned} \mu_{\text{st}}_{\text{banyak}} [0] &= 0 \\ \mu_{\text{st}}_{\text{cukup}} [0] &= 0 \\ \mu_{\text{st}}_{\text{sedikit}} [0] &= 1 \end{aligned}$$

- b. Pembentukan Aturan
Aturan untuk menghitung nilai abstrak dapat dilihat pada Tabel 1.

Tabel 1 Aturan Fuzzy

No	Kode	Aturan	Konsekuensi
1	[R1]	If SP sedikit and ST sedikit then	Nilai A
2	[R2]	If SP sedikit and ST cukup then	Nilai B
3	[R3]	If SP sedikit and ST banyak then	Nilai C
4	[R4]	If SP cukup and ST sedikit then	Nilai B
5	[R5]	If SP cukup and ST cukup then	Nilai C
6	[R6]	If SP cukup and ST banyak then	Nilai D
7	[R7]	If SP banyak and ST sedikit then	Nilai C
8	[R8]	If SP banyak and ST cukup then	Nilai D
9	[R9]	If SP banyak and ST banyak then	Nilai E

- c. Mesin Inferensi
Pada metode Tsukamoto, digunakan fungsi implikasi MIN yaitu dengan cara mengambil nilai minimum dari variabel *input* sebagai *output*-nya untuk mendapatkan α -predikat tiap-tiap *rule*. Masing-masing nilai α -predikat akan digunakan untuk menghitung keluaran hasil inferensi secara tegas (*crisp*) masing-masing *rule*.

$$\alpha = \mu_{SP \text{ sedikit}} \wedge \mu_{ST \text{ sedikit}}$$

$$\text{Predikat}_1 = \text{Min}(\mu_{SP \text{ sedikit}}, \mu_{ST \text{ sedikit}})$$

- d. Defuzzifikasi
Proses defuzzifikasi ini menggunakan metode rata-rata (*average*).

$$z^* = \frac{\alpha \cdot \text{Pred}_1 + \dots + \alpha \cdot \text{Pred}_9}{\alpha + \dots + \alpha}$$

$$z^* = \frac{(1 \cdot 100)}{1}$$

$$z^* = 100 \text{ (A)}$$

IV. Implementasi dan Pengujian

a. Halaman Utama

Pada halaman utama terdapat *text area* untuk teks yang akan diidentifikasi. *User* dapat *copy-paste* teks pada *text area* atau dengan memanfaatkan fitur *upload file*.

Jenis file yang dapat dibaca yaitu doc, docx, txt, dan pdf. Gambar 5. merupakan gambar dari halaman utama aplikasi. Pada halaman ini terdapat dua kotak, yang pertama untuk memasukkan teks yang akan diidentifikasi kesalahan penulisannya, dan yang kedua untuk melihat hasil identifikasi. Berikut merupakan potongan *source code* dari halaman utama aplikasi.

```

<div class='container-fluid'>
  <a href="index"><div class='page-header'>
    <h1 class='pull-left'>
      <i class='icon-book'></i>
      <span>Cek EYD</span>
      <small>(Ejaan Yang
Disempurnakan)</small>
    </h1>
  </div></a>
  <div class="row-fluid" >
    <div class="span6 box box-nomargin">
      <div id="sub3">
        <div class="box-header green-
background">
          <div class="title"><i class="icon-
edit"></i> Masukkan Teks:</div>
        </div>
        <div class="box-content">
          <div>
            <textarea id="sumber"
class="span12" rows="16" ></textarea>
          </div>
          <button class="btn btn-primary btn-
block btn-large" name="button" valign="right"
id="cek" style="margin-bottom:5px"
type="submit">
            <i class="icon-chevron-sign-
right"></i> Proses</button>
          <a href="file" type='button'
name='type' class="btn btn-link"
style="color:blue;">Upload File</a>
        </div>
      </div>
      <div class="span6 box">
        <div class="box-header red-background">
          <div class="title"><i class="icon-
level-down"></i> Hasil</div>
        </div>
        <div class="box-content" id="hasil">
          </div>
        </div>
      </div></div>

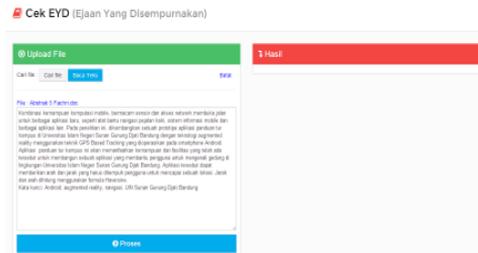
```



Gambar 5 Halaman Utama

b. Baca Teks

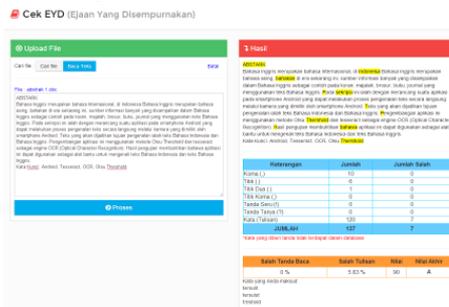
Halaman selanjutnya yaitu baca teks. Setelah *user* meng-upload file, *user* menekan tombol baca teks dan teks akan terlihat pada kolom 1. Dapat dilihat pada Gambar 6.



Gambar 6. Baca Teks

c. Hasil Identifikasi

Setelah *user* menekan tombol proses, maka akan muncul teks hasil identifikasi pada kolom 2 beserta keterangan kesalahan dan nilai. Dapat dilihat pada Gambar 7.



Gambar 7. Hasil Identifikasi

Berikut potongan *source code* saat proses identifikasi teks:

```

1 <?php
2 $str = $_POST['sumber'];
3 include 'periksa_kata.php';
4 include 'tanda_baca.php';
5 include 'hitung.php';
6
7 $sali = nl2br($hasil);
8 $p = explode(" ", $sali);
9 $j = sizeof($p);
10 $hitung = strlen($str);
11
12 if($hitung > 0){
13     $kalimat = "";
14     $sawalan = "";
15     foreach($p as $sancur){
16         $sk = trim($sancur);
17         if(strlen($sk) break; // brarti abis titik di akhir
18         $sk = substr($sk, 0, 1);
19         $sk1 = ucfirst($sk);
20         $sk2 = substr($sk1, 0, 1);
21         $str = substr($sk1, 1);
22         if(strlen($str) > 0){
23             $kalimat .= "<font color=red><b> $sk2</b></font>$str.";
24             $sawalan .= " $sk";
25         }
26         else{
27             $kalimat .= " $sk.";
28         }
29     }
30 }
31 $hit = strlen($sawalan);
32 $skoma = preg_match_all("/[a-zA-Z]+/i", $sali, $smatch);
33 $stitik = preg_match_all("/[.!?;:]+/i", $sali, $smatch);
34 $stikoma = preg_match_all("/[<\/>]+/i", $sali, $smatch);
35 $stitikoma = preg_match_all("/[<\/>]+/i", $sali, $smatch);
36 $standasenu = preg_match_all("/[a-zA-Z]+/i", $sali, $smatch);
37 $standatanya = preg_match_all("/[a-zA-Z]+/i", $sali, $smatch);
38
39 $angka = preg_match_all("/[0-9]+/i", $sali, $smatch);
40 $tbaca = preg_match_all("/[a-zA-Z]+/i", $sali, $smatch);
41
42 $sk1 = substr_count($kalimat, ',');
43 $sk2 = substr_count($kalimat, '.');
44 $sk3 = substr_count($kalimat, '!');
45 $sk4 = substr_count($kalimat, '?');
46 $sk5 = substr_count($kalimat, ':');
47 $sk6 = substr_count($kalimat, ';');
48
49 $jumlahTanda = $sk1 + $sk2 + $sk3 + $sk4 + $sk5 + $sk6;

```

Baris ke-2 merupakan *code* untuk mengambil teks dari teks area. Teks yang diambil tadi, akan dipecah-pecah menjadi

per kata (baris ke-8) dan akan dihitung jumlah katanya (baris ke-9). Kemudian teks yang telah dipecah akan diidentifikasi (baris ke-12 sampai baris ke-40). Lalu, akan dihitung jumlah tanda baca yang ada dalam teks tersebut (baris ke-42 sampai baris ke-48).

Algoritma *Fuzzy Logic Tsukamoto* digunakan ketika proses pemberian nilai kesalahan tulisan dan saat penentuan nilai akhir dari teks yang diidentifikasi. Nilai berupa angka dari 0-100, dan nilai akhir yaitu A, B, C, D, dan E.

d. Pengujian

Pengujian dilakukan dengan dua cara, pertama dilakukan secara manual oleh pembuat aplikasi, dan kedua oleh aplikasi yang telah dibuat. Dalam pengujian dihitung berapa banyak kesalahan tanda baca dan penulisan yang terdapat dalam setiap abstrak skripsi. Abstrak skripsi yang diuji sebanyak 20 abstrak skripsi. Tabel 2 merupakan hasil pengujian pada abstrak skripsi yang dilakukan.

Tabel 2 Hasil Pengujian

No	Abstrak ke-	Jenis File	Jumlah kesalahan		Ket.
			Berdasarkan periksa manual	Berdasarkan hasil aplikasi	
1	1	doc	7	7	√
2	2	doc	8	8	√
3	3	docx	0	0	√
4	4	docx	2	2	√
5	5	doc	1	1	√
6	6	docx	2	2	√
7	7	doc	1	2	√
8	8	docx	2	2	√
9	9	docx	6	6	√
10	10	pdf	3	15	√
11	11	pdf	4	29	√
12	12	pdf	1	11	√
13	13	pdf	2	6	√
14	14	doc	5	5	√
15	15	pdf	8	21	√
16	16	txt	0	0	√
17	17	txt	2	2	√
18	18	txt	7	7	√
19	19	txt	2	2	√
20	20	txt	5	5	√

Presentase kesesuaian hasil hitung aplikasi dan manual:

$$\text{Presentase} = (14/20) \times 100\% \\ = 70 \%$$

Berdasarkan hasil persentase diatas maka penulis dapat menyimpulkan bahwa hasil 70% sudah reliable. Ini dikarenakan pengujian yang dilakukan secara berulang dihasilkan nilai yang konsiten dengan kondisi yang berbeda. Hasil pesentase diatas dapat dijadikan acuan untuk pengkuran abstrak pada penulisan skripsi dikarenakan derajat ketepatannya sangat besar.

Perbedaan antara hasil hitung aplikasi dan manual disebabkan oleh kurangnya kosakata yang terdapat dalam *database* aplikasi dan kesalahan identifikasi tanda baca.

e. Kesimpulan

Berdasarkan uraian implementasi dan pengujian aplikasi pengukur ketepatan penggunaan EYD dapat disimpulkan bahwa metode *Fuzzy Logic Tsukamoto* dapat diterapkan dalam perhitungan nilai penggunaan tanda baca dan penulisan kata yang terdapat pada sebuah abstrak skripsi. Hasil pengujian terhadap 20 abstrak skripsi menunjukkan 70% sesuai dengan pemeriksaan manual.

f. Saran

Setelah mengevaluasi proses dan hasil dari aplikasi pengukur ketepatan penggunaan EYD ini, maka penulis memiliki saran untuk pengembangan aplikasi selanjutnya yaitu penambahan identifikasi kata asing untuk teks yang diidentifikasi.

g. Daftar Pustaka

- [1] Daeli, Irlani. *Natural Language Processing Analysis Of Sentences With Turbo Prolog*. Universitas Gunadarma.
- [2] Kadir, Abdul dan Terra CH. Triwahyuni. 2003. *Pengenalan Teknologi Informasi*. CV. Yogyakarta: Andi Offset.

- [3] Kao, Anne and Stephen R. Poteet (Eds). 2007. *Natural Language Processing and Text Mining*. London: Springer.
- [4] Kusumadewi, Sri, Hari Purnomo. 2010. *Aplikasi Logika Fuzzy Untuk Pendukung Keputusan*. Yogyakarta: Graha Ilmu.
- [5] Manik, Ngarap Im. 2010. *Perancangan Program Aplikasi Pengenalan Teks Menggunakan Fuzzy Logic*. Jakarta: Universitas BINUS.
- [6] Menteri Pendidikan Nasional Republik Indonesia. 2009. *Pedoman Umum Ejaan Bahasa Indonesia Yang Disempurnakan*. Jakarta.
- [7] Pressman, Roger S.. 2002. *Rekayasa Perangkat Lunak (Buku Satu)*. CV. Andi Offset. Yogyakarta.
- [8] Sutojo, T, Edy Mulyanto dan Vincent Suhartono. 2011. *Kecerdasan Buatan*. Yogyakarta: Andi.
- [9] Yakub. 2012. *Pengantar Sistem Informasi*. Yogyakarta: Graha Ilmu.