

# IMPLEMENTASI ALGORITMA NAZIEF-ADRIANI PADA FITUR TEBAK KATA DI WEB EDUKASI BAHASA INDONESIA

Handrie Noprisson<sup>1</sup>, Vina Ayumi<sup>2</sup>

Fakultas Ilmu Komputer, Universitas Mercu Buana  
Jl. Raya Meruya Selatan, Kembangan, Jakarta 11650

E-mail: <sup>1</sup>handrie.noprisson@mercubuana.ac.id, <sup>2</sup> vina.ayumi@mercubuana.ac.id

**Abstract—** *One of the rules or grammar of Bahasa Indonesia that must be known is about the introduction of basic words. In the field of computing, one of the algorithms developed and related to the word aspect in Indonesian is the Nazief-Adriani algorithm. The research methodology consists of 5 stages: data collection, analysis, design, implementation and report writing. Implementation of the Nazief-Adriani algorithm is performed on a guessword game feature that is used to guess the underlying word of the existing affix. As a result of the research, the Nazief-Adriani algorithm has been successfully implemented and tested for several words of attribute (pelatihan, pendidikan dan menghasilkan) on the basic word-guess games on the Web Indonesian Education well but the lack of the Nazief-Adriani algorithm is incapable of detecting and removing affix-infix or middle affix / inserts.*

**Abstrak—** Salah satu aturan atau tata bahasa Bahasa Indonesia yang harus diketahui adalah mengenai pengenalan kata dasar dan kata berimbuhan. Dalam bidang komputasi, salah satu algoritma yang dikembangkan dan berhubungan dengan aspek kata dasar dan kata berimbuhan Bahasa Indonesia adalah algoritma Nazief-Adriani. Metodologi penelitian terdiri dari 5 tahap yaitu pengumpulan data, analisis, desain, implementasi dan penulisan laporan. Implementasi algoritma Nazief-Adriani dilakukan pada fitur permainan tebak kata yang digunakan untuk menebak kata dasar dari imbuhan yang ada. Sebagai hasil penelitian, algoritma Nazief-Adriani berhasil diimplementasikan dan diuji coba untuk beberapa kata imbuhan (pelatihan, pendidikan dan menghasilkan) pada permainan tebak kata dasar di Web Edukasi Bahasa Indonesia dengan baik namun kekurangan dari algoritma Nazief-Adriani adalah belum mampunya untuk mendeteksi dan menghapus afiks-infiks atau imbuhan tengah/sisipan.

**Keywords—** Pemrosesan Teks, Bahasa Indonesia, Nazief Adriani

## I. Pendahuluan

Perkembangan teknologi informasi dan komunikasi telah menciptakan sejumlah variasi aplikasi yang dapat digunakan oleh masyarakat umum. Beberapa aplikasi dalam bentuk *offline* maupun *online* telah menjadi kebutuhan untuk menunjang aktifitas sehari-hari. Aplikasi saat ini yang cukup menarik untuk dikembangkan adalah aplikasi yang dapat digunakan sebagai media pembelajaran untuk menunjang peningkatan intelektual pengguna.

Salah satu jenis intelektual yang harus dimiliki dan dikembangkan adalah kemampuan intelektual berbahasa. Di Indonesia, bahasa resmi yang digunakan

sebagai alat komunikasi adalah bahasa Indonesia. Meskipun Bahasa Indonesia sering digunakan dalam kehidupan sehari-hari, namun banyak masyarakat terutama pelajar yang kurang mengetahui dengan baik aturan dalam menggunakan Bahasa Indonesia baik dalam lisan maupun tulisan [1].

Seperti yang dilansir *Kompas* dalam pemberitaannya di media internet, hasil pengumuman ujian nasional (UN) yang serentak sudah dilakukan selama sepekan di sejumlah daerah, termasuk Kota Malang, Jawa Timur, nilai Bahasa Inggris sejumlah siswa lebih tinggi dibanding nilai Bahasa Indonesia [1]. Padahal sebagai generasi penerus bangsa terpelajar, seharusnya lebih mengenal dan mengetahui tata Bahasa

Indonesia dibandingkan tata Bahasa Asing. Salah satu aturan atau tata bahasa Bahasa Indonesia yang harus diketahui adalah mengenai pengenalan kata dasar dan kata berimbuhan.

Dalam bidang komputasi, untuk pengenalan sebuah pola dikenal dengan *pattern recognition* [2], [3]. Salah satu algoritma yang dikembangkan dan berhubungan dengan pengenalan pola aspek kata dasar dan kata berimbuhan Bahasa Indonesia adalah algoritma Nazief-Adriani. Algoritma Nazief-Adriani sendiri merupakan algoritma *stemming* yang mentransformasikan kata-kata yang terdapat dalam suatu dokumen ke kata-kata akarnya (*root word*) dengan menggunakan aturan-aturan tertentu. Sebagai contoh, kata bersama, kebersamaan, menyamai, akan ditransformasi ke *root word*-nya yaitu "sama". Algoritma *stemming* untuk bahasa yang satu berbeda dengan algoritma *stemming* untuk bahasa lainnya, sebab algoritma *stemming* dibuat dengan menyesuaikan dengan struktur bahasa yang akan digunakan [4].

Berdasarkan latar belakang diatas dengan memanfaatkan Algoritma Nazief-Adriani, ilmu pemrograman *website* serta ilmu sastra mengenai kata dasar dan berimbuhan, maka penulis tertarik untuk melakukan penelitian dan memilih judul "Implementasi Algoritma Nazief-Adriani Untuk Fitur Tebak Kata di Web Edukasi Bahasa". Aplikasi lainnya juga telah dikembangkan untuk mempromosikan kembali warisan [3], [5] dan identitas bangsa Indonesia [6]. Diharapkan aplikasi berbasis *website* ini juga dapat digunakan sebagai media edukasi bagi masyarakat maupun pelajar untuk lebih mengenal Bahasa Indonesia sebagai identitas bangsa terutama mengenai kata-kata dasar dan kata berimbuhan.

## II. Studi Literatur

### A. Penelitian Terkait

Penelitian oleh [7] yang melakukan penelitian mengenai kemiripan teks menggunakan algoritma Nazief-Adriani untuk mengubah kata imbuhan menjadi kata dasar. Kemiripan teks dilihat dari hasil perhitungan *cosine similarity index* dari kata-kata dasar yang ada dalam teks tersebut.

Selain itu, penelitian oleh [8] melakukan penelitian mengenai pengklasifikasian dokumen teks. Penelitian menggabungkan beberapa algoritma antara lain Nazief Adriani, KNN dan Fungsi *Similarity Cosin*. Pada penelitian ini pengklasifikasian dokumen teks dilihat dari kata dasar yang dalam dokumen teks. Pengubahan kata imbuhan yang ada dalam dokumen teks menjadi kata dasar menggunakan algoritma Nazief Adriani.

### B. Algoritma Nazief Adriani

Algoritma Nazief dan Mirna Adriani ini memiliki tahap-tahap sebagai berikut [4]:

1. Cari kata yang akan distem dalam kamus. Jika ditemukan maka diasumsikan bahwa kata tersebut adalah *root word*. Maka algoritma berhenti.
2. *Inflection Suffixes* ("-lah", "-kah", "-ku", "-mu", atau "-nya") dibuang. Jika berupa *particles* ("-lah", "-kah", "-tah" atau "-pun") maka langkah ini diulangi lagi untuk menghapus *Possesive Pronouns* ("-ku", "-mu", atau "-nya"), jika ada.
3. Hapus *Derivation Suffixes* ("-i", "-an" atau "-kan"). Jika kata ditemukan di kamus, maka algoritma berhenti. Jika tidak maka ke langkah 3a.
  - a. Jika "-an" telah dihapus dan huruf terakhir dari kata tersebut adalah "-k", maka "-k" juga ikut dihapus. Jika kata tersebut ditemukan dalam kamus maka algoritma berhenti. Jika tidak ditemukan maka lakukan langkah 3b.
  - b. Akhiran yang dihapus ("-i", "-an" atau "-kan") dikembalikan, lanjut ke langkah 4.
4. Hapus *Derivation Prefix*. Jika pada langkah 3 ada sufiks yang dihapus maka pergi ke langkah 4a, jika tidak pergi ke langkah 4b.
  - a. Periksa tabel kombinasi awalan-akhiran yang tidak diijinkan. Jika ditemukan maka algoritma berhenti, jika tidak pergi ke langkah 4b.
  - b. For  $i = 1$  to 3, tentukan tipe awalan kemudian hapus awalan. Jika *root word* belum juga ditemukan lakukan langkah 5, jika sudah

maka algoritma berhenti. Catatan: jika awalan kedua sama dengan awalan pertama algoritma berhenti.

5. Melakukan *Recoding*.
6. Jika semua langkah telah selesai tetapi tidak juga berhasil maka kata awal diasumsikan sebagai *root word*. Proses selesai.

Tipe awalan ditentukan melalui langkah-langkah berikut:

1. Jika awalnya adalah: "di-", "ke-", atau "se-" maka tipe awalnya secara berturut-turut adalah "di-", "ke-", atau "se-".
2. Jika awalnya adalah "te-", "me-", "be-", atau "pe-" maka dibutuhkan sebuah proses tambahan untuk menentukan tipe awalnya.
3. Jika dua karakter pertama bukan "di-", "ke-", "se-", "te-", "be-", "me-", atau "pe-" maka berhenti.

TABEL I  
KOMBINASI AWALAN AKHIRAN YANG TIDAK DIJINKAN

Awalan	Akhiran yang tidak diijinkan
be-	-i
di-	-an
ke-	-l, -kan
me-	-an
se-	-l, kan

Sumber: [4]

TABEL II  
JENIS AWALAN BERDASARKAN TIPE AWALANNYA

Tipe Awalan	Akhiran yang harus dihapus
di-	di-
Ke-	Ke-
Se-	Se-
Te-	Te-
Ter-	Ter-
Ter-luluh	Ter-

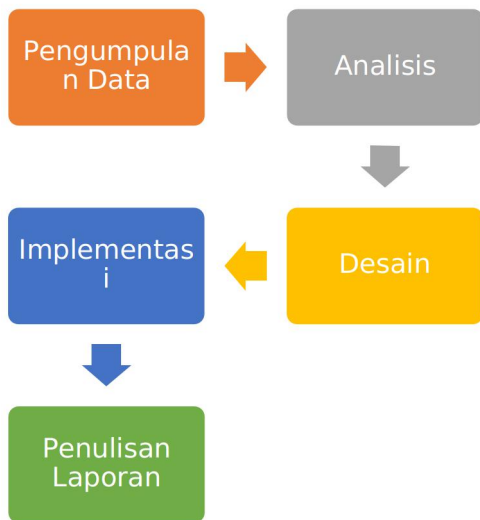
Sumber: [4]

Untuk mengatasi keterbatasan pada algoritma di atas, maka ditambahkan aturan-aturan dibawah ini:

1. Aturan untuk reduplikasi.
  - a. Jika kedua kata yang dihubungkan oleh kata penghubung adalah kata yang sama maka *root word* adalah bentuk tunggalnya, sebagai contoh: "buku-buku" *root word*-nya adalah "buku".
  - b. Kata lain, misalnya "bolak-balik", "berbalas-balasan, dan "seolah-olah". Untuk mendapatkan *root word*-nya, kedua kata diartikan secara terpisah. Jika keduanya memiliki *root word* yang sama maka diubah menjadi bentuk tunggal, contoh: kata "berbalas-balasan", "berbalas" dan "balasan" memiliki *root word* yang sama yaitu "balas", maka *root word* "berbalas-balasan" adalah "balas". Sebaliknya, pada kata "bolak-balik", "bolak" dan "balik" memiliki *root word* yang berbeda, maka *root word*-nya adalah "bolak-balik".
2. Tambahan bentuk awalan dan akhiran serta aturannya.
  - a. Untuk tipe awalan "mem-", kata yang diawali dengan awalan "memp-" memiliki tipe awalan "mem-".
  - b. Tipe awalan "meng-", kata yang diawali dengan awalan "mengk-" memiliki tipe awalan "meng-".

### III. Metodologi Penelitian

Metodologi penelitian terdiri dari 5 tahap yaitu pengumpulan data, analisis, desain, implementasi dan penulisan laporan seperti yang terlihat di Gambar 1.



Gbr. 1 Metodologi Penelitian

Tahap pengumpulan data dilakukan dengan cara studi pustaka dan mengumpulkan dataset teks bahasa Indonesia. Tahap analisis dilakukan dengan mempelajari cara kerja algoritma terhadap dataset yang sudah dikumpulkan. Tahap desain dilakukan dengan cara merancang

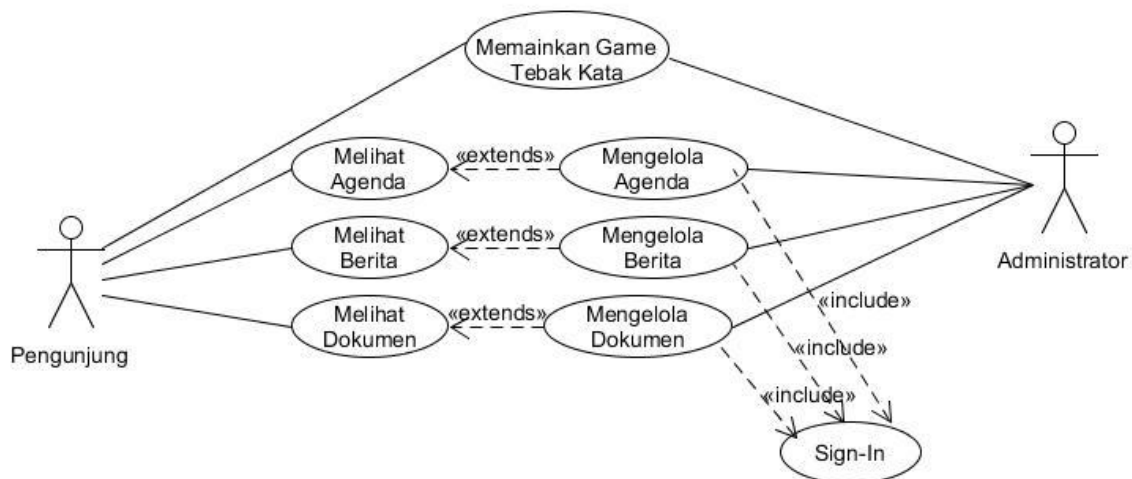
aplikasi sesuai dengan hasil analisis. Tahap implementasi dilakukan dengan cara menerjemahkan desain menjadi kode pemrograman PHP. Hasil implementasi dan analisis kinerja algoritma pada aplikasi kemudian dituliskan dalam bentuk laporan penelitian.

#### IV. Hasil dan Diskusi

Penelitian ini fokus pengimplementasian algoritma Nazief-Adriani pada salah satu fitur Web Edukasi Bahasa, sehingga web edukasi hanya digambarkan dalam bentuk *use case diagram* dan antar muka web edukasi.

##### A. Use Case Diagram

Pada Web Edukasi Bahasa terdiri dari fungsional antara lain, agenda, berita, dokumen, permainan tebak kata dan *sign-in*, seperti yang terlihat pada Gambar 2 berikut ini.

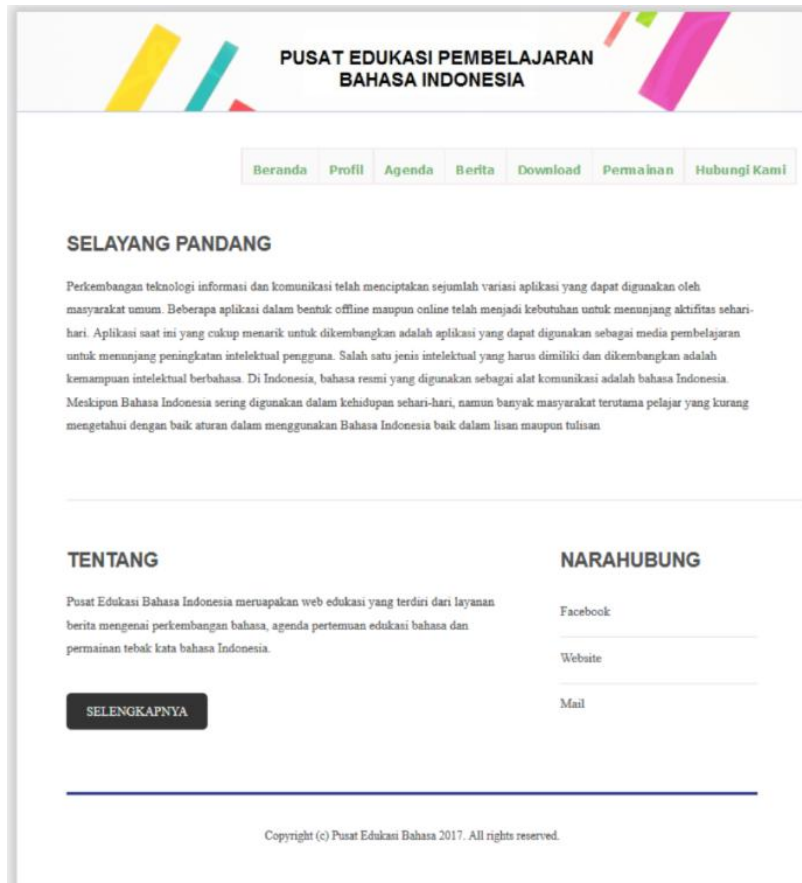


Gbr. 2 Use Case Diagram Web Edukasi

##### B. Antar Muka

Pada penelitian ini telah berhasil membangun Web Edukasi Bahasa

dengan antar muka yang sederhana yang dibagi menjadi tiga *section* yaitu, *header*, *body* dan *footer* seperti yang ditampilkan pada Gambar 3 berikut ini.



Gbr. 3 Antar Muka Web Edukasi

Implementasi algoritma Nazief-Adriani dilakukan pada fitur permainan tebak kata yang digunakan untuk menebak kata dasar dari imbuhan yang ada. Algoritma Nazief-Adriani ini sering digunakan untuk proses *stemming* atau proses pengubahan berimbuhan menjadi kata dasar dengan mengelompokkan bentuk imbuhan menjadi *Inflection Suffixes* (“-lah”, “-kah”, “-ku”, “-mu”, atau “-nya”), *Derivation Suffixes* (“-i”, “-an” atau “-kan”), dan *Derivation Prefix*. Namun, keakuratan dari algoritma ini tergantung pada banyaknya kata dasar yang ada dalam *database*. Potongan *source code* pemrograman dari proses *stemming* dapat dilihat pada Gambar 4.

```

1 $kataAsal = $kata;
2 /* 1. Cek Kata di Kamus jika Ada SELESAI */
3   if(cekKamus($kata)){ // Cek Kamus
4     return $kata; // Jika Ada kembalikan
5   }
6 /* 2. Buang Infection suffixes (\-lah", \-kah", \-
7   ku", \-mu", atau \-nya") */
8   $kata = Del_Inflection_Suffixes($kata);
9 /* 3. Buang Derivation suffix (\-i" or \-an") */
10  $kata = Del_Derivation_Suffixes($kata);
11 /* 4. Buang Derivation prefix */
12  $kata = Del_Derivation_Prefix($kata);

```

```

1   return $kata;
2
3   function Del_Inflection_Suffixes($kata){
4     $kataAsal = $kata;
5     if(ereg('([km]u|nya|[kl]ah|pun)$',$kata)
6   ){ // Cek Inflection Suffixes
7     $_kata =
8     eregi_replace('([km]u|nya|[kl]ah|pun)$','',$kata
9   );
10    if(ereg('([klt]ah|pun)$',$kata)){ // Jika berupa
11    particles ("-lah", "-kah", "-tah" atau "-pun")
12    if(ereg('([km]u|nya)$',$_kata)){
13      // Hapus Possesive Pronouns ("-ku", "-mu",
14      atau "-nya")
15      $_kata__ =
16      eregi_replace('([km]u|nya)$','',$_kata);
17      return $_kata__;
18    }
19    }
20    return $_kata;
21  }
22
23  }
24  return $kataAsal;
25
26  ....
27  }
28
29  function Del_Derivation_Suffixes($kata){
30    $kataAsal = $kata;
31    if(preg_match('/(kan)$',$kata)){ // Cek
32    Suffixes
33      $_kata =
34      preg_replace('/(kan)$','',$kata);
35    if(cekKamus($_kata)){ // Cek Kamus
36      return $_kata;
37    }
38  }

```



Berdasarkan jawaban pengguna dan hasil proses stemming, kata pendidikan memiliki kata dasar 'didik', sehingga antar muka akan menampilkan informasi 'Jawaban Anda Benar' seperti yang terlihat pada Gambar 7.



Gbr. 7 Antar Muka 'Jawaban Benar'

## V. Kesimpulan

Kesimpulan yang dapat diambil dari penelitian ini adalah sebagai berikut:

1. Penelitian ini menghasilkan Web Edukasi Bahasa Indonesia yang telah dilengkapi permainan tebak kata dasar yang dibangun dengan menggunakan bahasa pemrograman *Hypertext Markup Language* (HTML), *Hypertext Preprocessor* (PHP), *Cascading Style Sheet* (CSS), *Javascript* dan *Structured Query Language* (SQL) dengan berbantuan editor Notepad++ serta MySQL sebagai *database server*.
2. Algoritma Nazief-Adriani berhasil diimplementasikan dan diuji coba untuk beberapa kata imbuhan (pelatihan, pendidikan dan menghasilkan) pada permainan tebak kata dasar di Web Edukasi Bahasa Indonesia dengan baik.

## Referensi

- [1] J. Yudono, "Bahasa Indonesia Kian Kurang Diminati," *Kompas.com*, 2011. [Online]. Available: <https://edukasi.kompas.com/read/2011/05/21/18042068/Bahasa.Indonesia.Kian.Kurang.Diminati>. [Accessed: 24-Mar-2018].
- [2] I. Nurhaida, R. Manurung, and A. M.

Arymurthy, "Extraction Methods for Batik Recognition," in *2012 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, 2012, pp. 978-979.

- [3] I. Nurhaida, A. Noviyanto, M. Manurung, and A. M. Arymurthy, "Automatic Indonesian's Batik Pattern Recognition using SIFT Approach," in *ICCSCI - 1st International Conference on Computer Science and Computational Intelligence, Jakarta*, 2015.
- [4] L. Agusta, "Perbandingan Algoritma Stemming Porter Dengan Algoritma Nazief & Adriani Untuk Stemming Dokumen Teks Bahasa Indonesia," in *Konferensi Nasional Sistem dan Informatika*, 2009, pp. 196-201.
- [5] I. Nurhaida, R. Manurung, and A. M. Arymurthy, "Performance comparison analysis features extraction methods for batik recognition," in *International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, 2012.
- [6] D. Fitriana, A. N. Hidayanto, R. A. Zen, and A. M. Arymurthy, "APDATI: E-Fishing Logbook for Integrated Tuna Fishing Data Management," *J. Theor. Appl. Inf. Technol.*, vol. 75, no. 2, 2015.
- [7] R. V. Imbar, Adelia, Mewati Ayub, and A. Rehatta, "Implementasi Cosine Similarity dan Algoritma Smith-Waterman untuk Mendeteksi Kemiripan Teks," *J. Inform.*, vol. 10, no. 1, 2014.
- [8] D. Susandi and U. Sholahudin, "Pemanfaatan Vector Space Model pada Penerapan Algoritma Nazief Adriani, KNN dan Fungsi Similarity Cosine untuk Pembobotan IDF dan WIDF pada Prototipe Sistem Klasifikasi Teks Bahasa Indonesia," *J. ProTekInfo*, vol. 3, no. 1, 2016.