

Utility Klasifikasi Kata Berdasarkan Kamus pada Dokumen Bertipe Word (.docx) Berbasis Web

¹Dwi Sakethi dan ²Raka Widharma Kartika

^{1,2} Jurusan Ilmu Komputer FMIPA Universitas Lampung
Jalan Prof. Sumantri Brojonegoro No. 1 Bandar Lampung, Indonesia
e-mail: ¹dwijim@fmipa.unila.ac.id, ²raka17kw@gmail.com

Abstract — Word class is one of the central aspects in Indonesian grammar. To understand the classification of word classes, it is less effective to do it manually. So, developing a tool that can help fasten the process of classifying word classes in the document is need. Therefore, this research aims to build a system that can help make it easier for users to minimize time and human resources in determining the word class of a text document itself. The research development of a word classification system based on a dictionary on a word document (.docx) using the Waterfall or Classic Life Cycle, is divided into several stages, namely requirements specification and analysis, system and software design, coding and unit testing, system integration and testing, and installation and maintenance. This study resulted in a word classification system based on a dictionary on a word document (.docx). The success obtained in the development of this system is evidenced by the statistical data that shows the level of user satisfaction with a total average of 88.35% which is in the "Very Good" category.

Keywords: Development; Waterfall; Word Classification.

1. PENDAHULUAN

Bahasa Indonesia menjadi alat komunikasi yang digunakan oleh masyarakat Indonesia untuk keperluan sehari-hari misalnya belajar, bekerja sama dan berinteraksi. Bahasa Indonesia merupakan bahasa nasional dan bahasa resmi di Indonesia. Sebagai bahasa nasional, bahasa Indonesia tidak mengikat pemakainya untuk sesuai dengan kaidah dasar. Dalam pergaulan sehari-hari antar warga yang diutamakan adalah makna yang disampaikan. Pemakai bahasa Indonesia dalam konteks bahasa nasional dapat menggunakan dengan bebas menggunakan ujarannya baik lisan maupun tulis. Adapun bahasa formal adalah bahasa yang digunakan dalam komunikasi resmi seperti dalam penulisan jurnal dan laporan penelitian. Dalam hal ini, bahasa Indonesia harus digunakan sesuai dengan kaidah, tertib, cermat dan masuk akal. Bahasa Indonesia yang dipakai harus lengkap dan baku. Tingkat kebakuaannya diukur oleh aturan kebahasaan dan logika pemakaian.

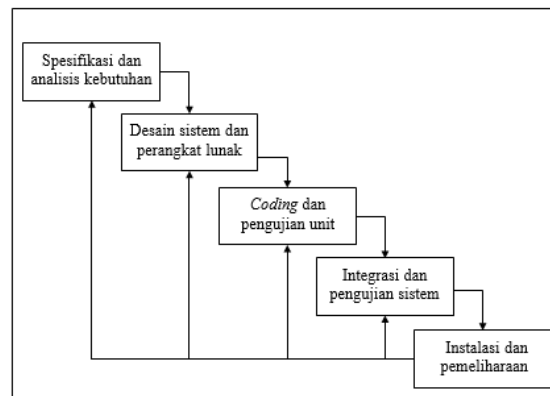
Kelas kata menjadi salah satu aspek tata bahasa Indonesia yang sentral. Oleh karena itu bila ingin memahami bahasa Indonesia, mau tidak mau orang harus memahami terlebih dahulu kelas kata dalam bahasa Indonesia. Dalam sebuah kalimat terdiri dari kumpulan kata yang kelasnya berbeda-beda dan untuk memahami pembagian kelas kata kurang efektif apabila melakukan pembagian kelas kata secara manual. Dengan adanya sistem yang dapat mengolah dokumen dengan proses klasifikasi, diharapkan dapat membantu mempermudah user dalam meminimalkan waktu dan sumber daya manusia dalam menentukan kelas kata suatu dokumen teks itu sendiri [1].

Penelitian ini menggunakan dokumen dengan ekstensi word (.docx) karena banyaknya penggunaan program pengolah kata *Microsoft Word* versi 2007 hingga yang versi 2019 sebagai alat edit dokumen. Proses pembuatan alat bantu ini dilakukan secara bertahap, dimulai dengan memasukkan dua file dokumen bertipe word (.docx) lalu mengolah isi dokumen pada tahap pra-proses dan ditemukan string kata. Alat bantu klasifikasi kata ini dibuat dengan Pemrograman *Hypertext Preprocessor* (PHP) dengan bantuan database dan kamus yang disimpan dalam database. Hasil yang diperoleh dalam alat bantu adalah klasifikasi kata dari dua dokumen [2].

2. METODOLOGI PENELITIAN

2.1. Metode Pengembangan Sistem

Metode yang digunakan penelitian ini adalah model *Waterfall* atau *Classic Life Cycle*. Model ini merupakan model klasik yang bersifat sistematis, berurutan dalam membangun perangkat lunak. Ada lima tahap dalam model *waterfall*, yaitu spesifikasi dan analisis kebutuhan, desain sistem dan perangkat lunak, *coding* dan pengujian unit, integrasi dan pengujian sistem, serta instalasi dan pemeliharaan [3].



Gambar 1. Metode *waterfall* [3].

2.2. Tahapan Penelitian

Penelitian ini dilakukan dengan beberapa tahapan, yaitu studi literatur, perancangan sistem, pengembangan sistem dan pengujian sistem. Alur tahapan penelitian tersebut dapat dilihat pada Gambar 2.



Gambar 2. Tahapan penelitian.

2.2.1. Studi Literatur

Studi literatur dilakukan untuk menghimpun data-data atau sumber yang berhubungan dengan topik yang diangkat dalam suatu penelitian yang akan dilakukan. Studi literatur ini bisa disapatkan dari berbagai sumber seperti jurnal, buku, internet dan pustaka.

2.2.2. Perancangan Sistem

Perancangan sistem merupakan tahap pengembangan sistem yang dilakukan dengan membuat desain rancangan sistem. Hal ini bertujuan untuk mempermudah pengguna dalam memahami rancangan sistem yang dibuat. Pembuatan alur proses sistem menggunakan *flowchart* diagram dengan pemodelan UML (*Unified Modeling Language*).

2.2.3. Pengembangan Sistem

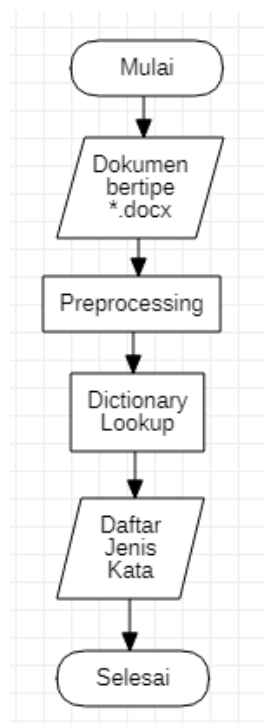
Pada tahap pengembangan sistem dilakukan pengkodean dengan mengubah rancangan yang telah dibuat menjadi kumpulan kode atau instruksi yang akan dijalankan oleh komputer. Proses pengkodean ini menggunakan bahasa pemrograman *Hypertext Preprocessor* (PHP) [4] dan basis data *MariaDB* [5]. Hasil dari proses pengembangan sistem ini adalah sebuah aplikasi web.

2.2.4. Pengujian Sistem

Pada penelitian ini pengujian sistem menggunakan metode *Black-Box Testing* dengan teknik *Equivalence Partitioning* [6]. Pengujian dilakukan secara fungsional berdasarkan fitur yang ada pada sistem tanpa harus mengetahui kode program di dalamnya (*Black-Box Testing*). Untuk melakukan pengujian ini, dibutuhkan pengguna atau *user* untuk menguji semua fitur yang ada dan memastikan bahwa semua fitur dapat merespon dan memberikan hasil yang sesuai dengan harapan pengguna.

2.4 Alur Proses Sistem

Tahapan proses utama dari sistem klasifikasi kata berdasarkan kamus pada dokumen bertipe *word* (.docx) sebagai berikut:



Gambar 3. Alur proses sistem.

2.4.1 Proses Upload Dokumen

Tahap awal pada *utility* ini yaitu melakukan unggah dokumen bertipe *word* (.docx) sistem tidak dapat memproses dokumen selain berekstensi tersebut.

2.4.2 Tahap *Pre-processing*

Pada tahap *pre-processing* meliputi beberapa proses yaitu sebagai berikut:

- a. *Whitespace insensitivity*
Proses ini menghapus tanda baca dan mengubah menjadi huruf kecil (*lowercase*) pada isi dokumen.
- b. Tokenisasi
Proses ini membagi teks dokumen menjadi token atau bagian-bagian tertentu. Tokenisasi merupakan proses pemisahan suatu rangkaian karakter berdasarkan karakter spasi.
- c. *Filtering*
Proses ini menghilangkan kata-kata yang tidak penting (*stopwords*) pada isi dokumen..

2.4.3 Tahap *Dictionary Lookup*

Setelah mendapatkan data *array* pada tahap *preprocessing* maka dilakukan pencocokan pada kamus berdasarkan jenis kata. Hasil akhir pada proses ini berupa daftar jenis kata dan jumlahnya berdasarkan jenis kata.

3. HASIL DAN PEMBAHASAN

Aplikasi klasifikasi kelas kata juga dikembangkan oleh [7] terkait *clustering* jenis kata pada *text* laporan kegiatan Kuliah Kerja Nyata (KKN) Universitas Lampung menggunakan Word2vec untuk data berbahasa Indonesia. Pada penelitian ini digunakan data kamus dengan jumlah data sebanyak 403.342 yang terdiri dari 28.526 jenis kata, 370.103 suku kata asing, 3.622 nama, dan 1.091 *stopword*. Data latih yang digunakan dalam penelitian ini yaitu dengan format file *.docx*.

3.1 Implementasi Tampilan Antarmuka

Berikut ini contoh hasil implementasi antarmuka *utility* klasifikasi kata berdasarkan kamus pada dokumen bertipe *word (.docx)* berdasarkan susunan program yang telah dibangun.

3.1.1. Halaman Utama

Pada halaman ini terdapat tiga menu yaitu beranda, kamus, cara pakai dan tentang. Halaman utama merupakan tampilan awal *utility*, terdapat sebuah *form upload* yang berfungsi untuk mengunggah dokumen bertipe *word (.docx)* yang akan diproses untuk melihat daftar jenis-jenis kata yang ada pada dokumen. Tampilan halaman tersebut dapat dilihat pada Gambar 4.



Gambar 4. Halaman utama.

3.1.2. Halaman Klasifikasi Kata

Halaman ini menampilkan hasil klasifikasi kata berdasarkan dokumen. Pada halaman lihat detail, terdapat tombol ‘selesai’ untuk kembali ke halaman Beranda, tombol ‘unduh detail’ untuk mengunduh daftar klasifikasi kata yang telah diproses dan dapat diunduh berupa *file* berekstensi *.docx, serta tombol ‘lihat detail’ untuk melihat detail klasifikasi dokumen yang disajikan dalam sebuah tabel, yang dapat dilihat pada Gambar 5.

Jenis	Jumlah	Jenis	Jumlah
Kata Benda	11	Kata Benda	8
Kata Kerja	3	Kata Sifat	2
Kata Sifat	2	kosong/tidak ada jenis kata	14
kosong/tidak ada jenis kata	11	kata asing	8
kata asing	14	nama	10
nama	0	Adverbia	0
Adverbia	0	Interjeksi	0
Interjeksi	0	Kata Kerja	0
Konjungsi	0	Konjungsi	0
Lain-lain	0	Lain-lain	0
Numeralia	0	Numeralia	0
Preposisi	0	Preposisi	0
Pronomina	0	Pronomina	0

Dokumen 1: Dengan Jumlah Kata 30
 Dokumen 2: Dengan Jumlah Kata 28
 Dokumen ke-1 dan ke-2 memiliki kemiripan = 68.965517241379% dan perbedaan 31.034482758621%

Gambar 5. Halaman utama.

3.2. Implementasi Kode Program

Adapun inti dari pengembangan *utility* klasifikasi kata berdasarkan kamus pada dokumen bertipe .docx ini adalah mengembangkan alat yang dapat melakukan proses klasifikasi jenis kata yang terjadi di dalam dokumen. Berikut merupakan implementasi koding dari pengembangan yang dilakukan.

3.2.1 Fungsi Upload dan Ekstraksi

```
private function read_docx(){
    $striped_content = '';
    $content = '';
    $zip = zip_open($this->filename);
    if (!$zip || is_numeric($zip)) return false;
    while ($zip_entry = zip_read($zip)) {
        if (zip_entry_open($zip, $zip_entry) == FALSE) continue;
        if (zip_entry_name($zip_entry) != "word/document.xml") continue;

        $content .= zip_entry_read($zip_entry, zip_entry_filesize($zip_entry));
        zip_entry_close($zip_entry);
    } // end while

    zip_close($zip);
    $content = str_replace('</w:r></w:p></w:tc><w:tc>', " ", $content);
    $content = str_replace('</w:r></w:p>', "\r\n", $content);
    $striped_content = strip_tags($content);

    return $striped_content;}
public function convertToText() {

    if(isset($this->filename) && !file_exists($this->filename)) {
        return "File Not exists";
    }
}
```

```
}

$fileArray = pathinfo($this->filename);
$file_ext = $fileArray['extension'];
if($file_ext == "doc" || $file_ext == "docx" )
{
    if($file_ext == "doc") {
        return $this->read_doc();
    } elseif($file_ext == "docx") {
        return $this->read_docx();
    }
} else {
    return "Invalid File Type";
}
}
```

Kode Program 1. Fungsi pembacaan dokumen bertipe .docx.

3.2.2 Whitespace insensitivity

Setelah antarmuka halaman utama menerima *input* berupa dokumen bertipe .docx lalu variabel yang telah terisi string masuk ke dalam tahap *whitespace insensitivity* yaitu isi yang ada di dalam dokumen akan dibersihkan dari karakter selain huruf kecil “a-z”, huruf besar “A-Z” menggunakan fungsi seperti Kode Program 2 untuk dokumen 1, dan Kode Program 3 untuk dokumen 2.

```
$selain_document2=preg_replace("/[^a-zA-Z]/","",$kecil_document1);
```

Kode Program 2. Fungsi bersih dari tanda baca dokumen 1.

```
$selain_document2=preg_replace("/[^a-zA-Z]/","",$kecil_document2);
```

Kode Program 3. Fungsi bersih dari tanda baca dokumen 2.

Lalu dibersihkan dari spasi ganda menggunakan Kode Program 4.

```
function hapus_spasi_spam($cus){
    $cus = trim($cus);
    while(strpos($cus," ")){
        $cus = str_replace(" ","",$cus);
    }
    return $cus;
}
```

Kode Program 4. Fungsi hapus spasi ganda.

Tipe karakter yang diseleksi yaitu karakter berupa digit (numerik) dan *letter* (teks string), kemudian pada tahap akhir dijadikan *lower case* yaitu mengubah seluruh karakter yang mengandung huruf menjadi huruf kecil semua dengan menggunakan Kode Program 5.

```
$kecil_document2=strtolower($isi_document2);
```

Kode Program 5. Fungsi *lower case*.

3.3.1 Tokenisasi

Selanjutnya proses tokenisasi dilakukan pencacahan kalimat menjadi perkata berdasarkan karakter spasi. Berikut Kode Program 6 untuk proses tokenisasi.

```
$pecah_document2=explode(" ", $dacok_document2);
```

Kode Program 1. Fungsi tokenisasi *filtering*.

Proses pada ini yaitu melakukan pengambilan kata-kata penting dari hasil token. Pengambilan kata-kata ini dapat dilakukan dengan membuang kata yang kurang penting (*stopwords*) atau menyimpan kata-kata penting (*wordlist*) dengan menggunakan Kode Program 7.

```
$simpan=0;
foreach ($tampungan as $stop_list){
    $kueri = "SELECT * FROM tb_stoplist where stoplist = '$stop_list'";
    $hasil = mysqli_query($conn, $kueri);
    $cek = mysqli_fetch_assoc($hasil);

    $cekcek = $cek ['stoplist'] ?? '';
    if ($stop_list!=$cekcek){
        $simpan_arr[$simpan] = $stop_list;
        $simpan++;
    }
}
```

Kode Program 7. Fungsi *filtering*.

3.3.2 Dictionary Lookup

Data kamus yang digunakan disimpan dalam *database* dan variabel *array* yang telah melalui proses *whitespace insensitivity*, *tokenisasi*, dan *filtering* kemudian dilakukan pengecekan dengan data kamus melalui proses *dictionary lookup*. Fungsi untuk proses ini dapat dilihat pada Kode Program 8.

```
foreach ($arr_document21 as $stop2){
    $kueri_stop1 =mysqli_query($conn, "SELECT katadasar FROM kata_dasar where katadasar = '$stop2'");
    $kueri_stop2 =mysqli_query($conn, "SELECT ket_nama FROM nama where ket_nama = '$stop2'");
    $kueri_stop3 =mysqli_query($conn, "SELECT kata_dasar FROM kata_dasar_asing where kata_dasar = '$stop2'");
    $cek_stop1 = mysqli_fetch_assoc($kueri_stop1);
    $cek_stop2 = mysqli_fetch_assoc($kueri_stop2);
    $cek_stop3 = mysqli_fetch_assoc($kueri_stop3);

    $cek1_stop1 = $cek_stop1 ['katadasar'] ?? '';
    $cek2_stop2 = $cek_stop2 ['ket_nama'] ?? '';
    $cek3_stop3 = $cek_stop3 ['kata_dasar'] ?? '';

    if (($stop2=$cek1_stop1) || ($stop2=$cek2_stop2) || ($stop2=$cek3_stop3))
    {
        $simpan_stop2[$simpan_2] = $stop2;
        $simpan_2++;
    }
}
```

Kode Program 8. Fungsi *dictionary lookup*.

3.3 Pengujian Sistem

3.3.1 Pengujian Fungsional

Pengujian fungsional yang dilakukan pada sistem klasifikasi kata berdasarkan kamus pada dokumen bertipe word (.docx) ini untuk mengetahui hasil uji coba pada setiap menu atau fungsi yang ada pada sistem. Tabel 1 menunjukkan hasil pengujian fungsional dengan menggunakan metode *black-box testing*.

Tabel 1. Hasil pengujian fungsional halaman utama

Halaman	Hasil Uji
Halaman Utama	Terpenuhi
Halaman Klasifikasi	Terpenuhi
Halaman Lihat Detail	Terpenuhi
Halaman Kamus Kata Dasar	Terpenuhi
Halaman Kamus Kata Asing	Terpenuhi
Halaman Kamus Nama	Terpenuhi
Halaman Cara Pakai	Terpenuhi
Halaman Tentang	Terpenuhi

3.3.2 Pengujian Nonfungsional

Pengujian nonfungsional merupakan pengujian berdasarkan tingkat kepuasan pengguna terhadap sistem klasifikasi kata berdasarkan kamus pada dokumen bertipe word (.docx). Pengujian ini dilakukan dengan mengisi kuesioner di Google Form yang dimulai pada tanggal 25 April sampai 27 April 2021 dengan jumlah total responden sebanyak 34 orang. Hasil pengujian non fungsional dapat dilihat pada Tabel 2.

Tabel 2. Hasil pengujian non fungsional

No.	Pernyataan	Jumlah Responden: 38					Total Skor	Rata-rata (%)
		Skor						
		SS	S	CS	KS	TS		
1.	Sistem telah memberikan informasi yang dibutuhkan.	16	17	1	0	0	151	88,82
2.	Tampilan sistem menarik.	15	17	2	0	0	149	87,65
3.	Semua fungsi yang ada di sistem mudah dipahami atau dimengerti	16	15	3	0	0	149	87,65
4.	Sistem dapat berinteraksi dengan pengguna.	12	20	2	0	0	146	85,88
5.	Label/tombol yang digunakan dalam sistem mudah dimengerti	19	14	1	0	0	154	90,58
6.	Sistem memiliki proses input yang mudah.	16	16	2	0	0	150	88,23
7.	Sistem informasi mudah digunakan.	18	15	1	0	0	153	90,00

		Jumlah Responden: 38					Total Skor	Rata-rata (%)
No.	Pernyataan	Skor						
		SS	S	CS	KS	TS		
8.	Tata letaknya (<i>display</i>) mudah dibaca.	15	17	2	0	0	149	87,65
9.	Setiap tampilan halaman memberikan informasi yang sesuai.	19	14	1	0	0	154	90,58
10.	Saya merasa senang menggunakan sistem informasi ini.	14	17	3	0	0	147	86,47

Berdasarkan Tabel 2. hasil pengujian nonfungsional, pengisian kuesioner oleh 34 responden diperoleh data dengan nilai rata-rata 88.35%.

4. KESIMPULAN

Berdasarkan hasil pengembangan yang dilakukan penulis pada sistem klasifikasi kata pada dokumen bertipe *word (.docx)*, dapat disimpulkan bahwa sistem klasifikasi kata berdasarkan kamus pada dokumen bertipe *word (.docx)* berhasil dikembangkan. Sistem dapat menghasilkan klasifikasi kata berdasarkan kamus pada dua dokumen *word* bertipe **.docx*. Tingkat kepuasan pengguna pada sistem klasifikasi kata berdasarkan kamus pada dokumen *word (.docx)* sebanyak 34 responden memperoleh hasil yang masuk ke dalam kategori sangat baik dengan presentase rata-rata 88.35%.

DAFTAR PUSTAKA

- [1] S. R. Utami, "Pembelajaran Aspek Tata Bahasa dalam Buku Pelajaran Bahasa Indonesia," *AKSIS: Jurnal Pendidikan Bahasa dan Sastra Indonesia*, 1(2), 2017.
- [2] U. Budiman dan Izzatul, "Sistem Pengolahan Kalimat Bahasa Indonesia Menjadi Kata Dasar dengan Metode Stemmer Berbasis Pemrograman Hypertext Preprocessor," *Journal Trisula LP2M Undar*, 5(1), 501-507, 2017.
- [3] I. Sommerville, *Software Engineering 9th edition*. Addison Wesley: Publisher, pp. 29-32, 2007
- [4] A. Firman, H.F. Wowor, dan X. Najoan, "Sistem Informasi Pepustakaan Online Berbasis Web," *Journal Teknik Elektro dan Komputer*, 5(2), 29-36, 2016.
- [5] D. G. Mahendra dan A. Kasyaful, "Analisis Availabilitas dan Reliabilitas Multi-master Database Server dengan State Snapshot Transfers (SST) Jenis Rsync pada MariaDB Galera Cluster," *Jurnal Teknologi Informasi dan Ilmu Komputer*, 4(1), 69-74, 2017.
- [6] T. Hidayat dan M. Mutaqqin, "Pengujian Sistem Informasi Pendaftaran dan Pembayaran Wisuda Online menggunakan Black-Box Testing dengan Metode Equivalence Partitioning dan Boundary Value Analysis," *Jutis (Jurnal Teknik Informatika)*, 6(1), 25-29, 2020.
- [7] Ademariana, K., Aristoteles, A., Lumbanraja, F. R., dan Rico Andrian, R. A., "Clustering K-Means Jenis Kata Pada Laporan Kegiatan Kuliah Kerja Nyata (KKN) Universitas Lampung Menggunakan Word2vec", *Jurnal Pepadun*, 2(2), 221-228, 2021.