

**KLASIFIKASI KATA BERDASARKAN KAMUS PADA DOKUMEN
BERTIPE WORD (.docx)**

SKRIPSI

Oleh

RAKA WIDHARMA KARTIKA



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2022**

ABSTRACT

WORD CLASSIFICATION BASED ON DICTIONARY ON DOCUMENTS OF WORD TYPE (.docx)

By

Raka Widharma Kartika

Word class is one of the central aspects of Indonesian grammar. To understand the division of word classes, it is less effective to do the division manually. So it takes a fast detection to classify word classes in the document. Therefore, we need 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: *Word Classification, Waterfall, Development*

ABSTRAK

KLASIFIKASI KATA BERDASARKAN KAMUS PADA DOKUMEN BERTIPE WORD (.docx)

Oleh

Raka Widharma Kartika

Kelas kata menjadi salah satu aspek tata bahasa Indonesia yang sentral. Untuk memahami pembagian kelas kata kurang efektif apabila melakukan pembagian secara manual. Sehingga dibutuhkan pendeteksian secara cepat untuk mengelompokkan kelas kata pada dokumen. Oleh karena itu, diperlukan sistem yang dapat membantu mempermudah user dalam meminimalkan waktu dan sumber daya manusia dalam menentukan kelas kata suatu dokumen teks itu sendiri. Pengembangan penelitian sistem klasifikasi kata berdasarkan kamus pada dokumen bertipe word (.docx) dengan menggunakan model *Waterfall* atau *Classic Life Cycle*, terbagi menjadi beberapa tahap yaitu spesifikasi dan analisis kebutuhan, desain sistem dan perangkat lunak, *coding* dan pengujian unit, integrasi dan pengujian sistem, serta instalasi dan pemeliharaan. Penelitian ini menghasilkan sistem klasifikasi kata berdasarkan kamus pada dokumen bertipe word (.docx). Keberhasilan yang didapat pada pengembangan sistem ini dibuktikan dengan adanya data statistik yang menunjukkan tingkat kepuasan pengguna dengan total rata-rata 88.35% yang mana nilai tersebut masuk dalam kategori “Sangat Baik”.

Kata Kunci: Klasifikasi Kata, *Waterfall*, Pengembangan

**KLASIFIKASI KATA BERDASARKAN KAMUS PADA DOKUMEN
BERTIPE WORD (.docx)**

**Oleh
RAKA WIDHARMA KARTIKA**

**Skripsi
Sebagai Salah Satu Syarat Untuk Mencapai Gelar
SARJANA KOMPUTER**

**Pada
Jurusan Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2022**

Judul Skripsi : **Klasifikasi Kata Berdasarkan Kamus pada Dokumen Bertipe Word (.docx)**

Nama Mahasiswa : *Raka Widharma Kartika*

Nomor Pokok Mahasiswa : 1517051029

Jurusan : Ilmu Komputer

Fakultas : Matematika dan Ilmu Pengetahuan Alam



[Signature]
Dwi Sakelmi, S.Si., M.Kom.
NIP. 19680611 199802 1 001

2. Ketua Jurusan Ilmu Komputer

[Signature]
Didik Kurniawan, M.T.
NIP. 19800419 200501 1 004

MENGESAHKAN

1. Tim Penguji

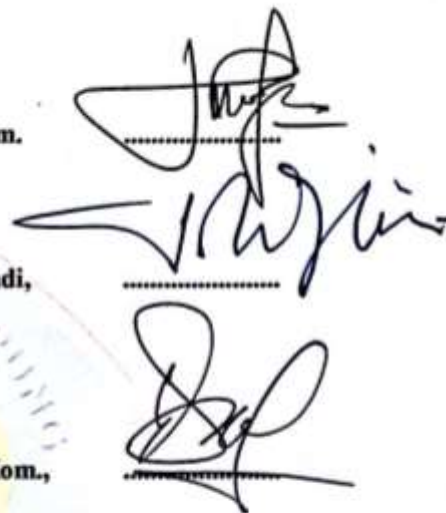
Ketua Penguji : Dwi Sakethi, S.Si., M.Kom.

Penguji

**Bukan Pembimbing : Drs. Rd. Irwan Adi Pribadi,
M.Kom.**

Penguji

**Bukan Pembimbing : Bambang Hermanto, S.Kom.,
M.Sc.**



2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam

Tanggal Lulus Ujian Skripsi: 22 Juni 2022

PERNYATAAN

Saya bertanda tangan di bawah ini, menyatakan bahwa skripsi saya yang berjudul **“Klasifikasi Kata Berdasarkan Kamus pada Dokumen Bertipe Word (.docx)”** merupakan karya saya sendiri dan bukan karya orang lain. Semua tulisan yang tertuang di skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti skripsi saya merupakan hasil penjiplakan atau dibuat orang lain, maka saya bersedia menerima sanksi berupa pencabutan gelar yang telah saya terima.

Bandar Lampung, 25 Juni 2022



Raka Widharma Kartika

NPM. 1517051029

RIWAYAT HIDUP

Penulis dilahirkan pada tanggal 29 Maret 1998 di Bumi Kencana, Seputih Agung, Lampung Tengah sebagai anak pertama dari dua bersaudara. Ayah bernama Ahmad

Yari dan Ibu Ratna Harwati.



Penulis menyelesaikan menyelesaikan Pendidikan Sekolah Dasar (SD) di SD 1 Bumi Kencana, Seputih Agung, Lampung Tengah pada tahun 2009, menyelesaikan Sekolah Menengah Pertama (SMP) di SMPN 1 Seputih Agung pada tahun 2012, kemudian menyelesaikan sekolah

di jenjang Sekolah Menengah Atas (SMA) SMAN 1 Seputih Agung dan lulus pada tahun 2015.

Pada tahun 2015, penulis terdaftar sebagai mahasiswa Jurusan Ilmu Komputer FMIPA Universitas Lampung melalui jalur SNMPTN. Selama menjadi mahasiswa, penulis melakukan Kerja Praktek pada bulan Januari 2018 di Kantor Dinas Pendidikan dan Kebudayaan Kota Metro. Pada Bulan Juli 2018, penulis melaksanakan Kuliah Kerja Nyata Kebangsaan di Desa Karang Anyar Kecamatan Labuhan Maringgai Kabupaten Lampung Timur.

MOTO

"Sesuatu yang belum dikerjakan, seringkali tampak mustahil; kita baru yakin kalau kita telah berhasil melakukannya dengan baik."

(Evelyn Underhill)

"Ketika Anda berkata "YA" kepada Allah tanpa syarat, Anda tidak akan tahu seberapa jauh "YA" tersebut akan membawa Anda."

(Hans ur Von Balthasar)

"Hal yang membuat sulit adalah ketakutanmu itu sendiri"

PERSEMBAHAN

Puji Syukur kepada Allah SWT sehingga skripsi ini dapat diselesaikan.

Skripsi yang disusun ini adalah suatu kebanggaan yang saya persembahkan

kepada:

Kedua Orang Tua ku yang Tercinta

Bapak dan Mamak yang senantiasa memberikan segala perhatian, kasih sayang, cinta, dukungan moril maupun materil serta doa terbaik bagi kesuksesan anak-anaknya yang tidak bisa diukur dan dibalas dengan apapun.

Rosi Rosalena dan keluarga besar yang senantiasa memberikan dukungan, kasih sayang, semangat, dan menanyakan kapan skripsi ini akan selesai.

Keluarga Besar Ilmu Komputer 2015

Serta Almamater Tercinta,

UNIVERSITAS LAMPUNG

SANWACANA

Puji Syukur penulis panjatkan terhadap Allah SWT sehingga penulis dapat menyelesaikan skripsi di Jurusan Ilmu Komputer Fakultas MIPA Universitas Lampung dengan judul “Klasifikasi Kata Berdasarkan Kamus pada Dokumen Bertipe Word (.docx)”.

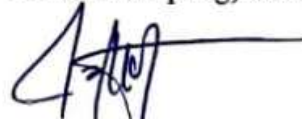
Penulis mengucapkan terima kasih yang paling tulus dan sebesar-besarnya kepada semua pihak yang telah membantu dalam pelaksanaan penelitian dan penyusunan laporan skripsi ini, antara lain:

1. Kedua orang tua yang sangat penulis sayangi, Bapak dan Mamak yang senantiasa mengajarkan kebaikan, memberikan perhatian, kasih sayang, do'a terbaik, dukungan, serta kepercayaan atas setiap keputusan yang diambil oleh penulis hingga detik ini.
2. Saudara penulis, Aditya Dwi Kartika yang selalu memberikan perhatian, pengertian, dukungan, dan semangat dalam setiap langkah penyusunan skripsi ini.
3. Bapak Dwi Sakethi, S.Si., M.Kom selaku pembimbing utama penulis dalam penelitian ini yang senantiasa memberikan arahan, bantuan ketika mengalami kesulitan selama penelitian, semangat, dan motivasi terbaik dalam menyelesaikan penelitian ini.
4. Bapak Drs. Rd. Irwan Adi Pribadi, M.Kom selaku pembahas 1 yang telah memberikan bantuan, masukan, dan saran yang bermanfaat untuk perbaikan penelitian dan penulisan laporan skripsi ini.
5. Bapak Bambang Hermanto S.Kom., M.Cs selaku pembahas 2 yang telah memberikan bantuan, masukan, dan saran yang bermanfaat untuk perbaikan penelitian dan penulisan laporan skripsi ini.
6. Bapak Dr. Eng. Suripto Dwi Yuwono, M.T selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.

7. Bapak Didik Kurniawan, S.Si., M.T selaku Ketua Jurusan dan pembimbing akademik yang telah memperhatikan serta memberikan motivasi kepada penulis dalam menempuh pendidikan di Ilmu Komputer Fakultas MIPA Universitas Lampung.
8. Bapak Dr. rer. nat. Akmal Junaidi, M.Sc selaku Sekretaris Jurusan Ilmu Komputer Fakultas MIPA Universitas Lampung.
9. Seluruh Dosen Jurusan Ilmu Komputer yang telah memberikan ilmu dan pelajaran terbaik selama penulis menempuh pendidikan di Jurusan Ilmu Komputer Universitas Lampung.
10. Seluruh Karyawan dan Staf Jurusan Ilmu Komputer Universitas Lampung yang telah memberikan bantuan dalam banyak hal selama ini.
11. Rosi Rosalena selaku calon pendamping hidupku yang sangat penulis sayangi dan senantiasa berada disisi penulis dalam suka dan duka, memberikan keceriaan, pengertian, motivasi, bantuan dan bersedia menjadi pendengar yang baik ketika penulis ingin berkeluh kesah selama ini.
12. Yohanes Riyanto yang telah memberikan arahan, dukungan dan semangat kepada penulis untuk menyelesaikan studi ini.
13. Keluarga besar Class A Ilmu Komputer 2015 yang telah memberikan kebersamaan, semangat, dan do'a sebagai teman sekelas.

Penulis menyadari bahwa penyusunan skripsi ini masih jauh dari kata sempurna. Namun penulis sangat mengharapkan skripsi ini dapat bermanfaat bagi para civitas akademik Universitas Lampung pada umumnya dan mahasiswa Ilmu Komputer pada khususnya.

Bandar Lampung, 25 Juni 2022



Raka Widharma Kartika
NPM. 1517051029

DAFTAR ISI

	Halaman
ABSTRACT	ii
ABSTRAK	iii
RIWAYAT HIDUP	viii
MOTO	ix
PERSEMBAHAN	x
SANWACANA	xi
DAFTAR ISI	xiii
DAFTAR TABEL	xviii
DAFTAR GAMBAR	xviii
DAFTAR KODE PROGRAM	xx
I. PENDAHULUAN	1
1.1 Latar Belakang dan Masalah.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan Penelitian	2
1.4 Batasan Masalah	3
1.5 Manfaat	3
II. TINJAUAN PUSTAKA	4
2.1 Pembagian Kelas Kata	4
2.2 Sistem.....	5
2.3 <i>Database</i>	6

2.4	Hypertext Preprocessor (PHP)	6
2.5	MariaDB	7
2.6	Flowchart	8
2.7	Waterfall	11
2.8	Black-Box Testing	14
2.9	Preprocessing	15
2.9.1	Whitespace Insensitivity	15
2.9.2	Tokenisasi	16
2.9.3	Filtering	16
2.10	Dictionary Lookup	17
III.	METODE PENELITIAN	18
3.1	Waktu dan Tempat Penelitian	18
3.2	Alat dan Bahan Penelitian	18
3.3	Tahapan Penelitian	19
3.4	Perancangan Proses Sistem U-Correct	21
3.5	Rancangan Antarmuka Sistem	22
3.5.1	Rancangan Halaman Utama	22
3.5.2	Rancangan Halaman Klasifikasi	23
3.5.3	Rancangan Halaman Lihat Detail	24
3.5.4	Rancangan Halaman Kamus Kata Dasar	25
3.5.5	Rancangan Halaman Kamus Kata Asing	25
3.5.6	Rancangan Halaman Kamus Nama	26
3.5.7	Rancangan Halaman Cara Pakai	27
3.5.8	Rancangan Halaman Tentang	27
3.6	Rancangan Tabel Basis Data	28
3.7	Pengujian Sistem	31
IV.	HASIL DAN PEMBAHASAN	39
4.1	Analisis Kebutuhan Data	39
4.2	Implementasi Tampilan Antarmuka	39
4.3	Implementasi Kode Program	44
4.3.1	Fungsi <i>Upload</i> dan Ekstraksi	44
4.3.2	Whitespace Insensitivity	45
4.3.3	Tokenisasi	46
4.3.4	Filtering	46

4.3.5	<i>Dictionary Lookup</i>	47
4.4	Hasil Pengujian Sistem	48
4.4.1	Pengujian Fungsional	48
4.4.2	Pengujian Non Fungsional	56
V.	SIMPULAN DAN SARAN	58
5.1	Simpulan	58
5.2	Saran	58
	DAFTAR PUSTAKA	59

DAFTAR TABEL

	halaman
Tabel 1. Simbol-simbol Flowchart	9
Tabel 2. Test Scenario Menu	31
Tabel 3. Test Scenario Halaman Utama	33
Tabel 4. Test Scenario Halaman Klasifikasi	34
Tabel 5. Test Scenario Halaman Lihat Detail	35
Tabel 6. Test Scenario Halaman Kamus Kata Dasar	36
Tabel 7. Test Scenario Halaman Kamus Kata Asing	36
Tabel 8. Test Scenario Halaman Kamus Nama	37
Tabel 9. Test Scenario Halaman Cara Pakai	37
Tabel 10. Test Scenario Halaman Tentang	38
Tabel 11. Hasil Pengujian Fungsional Halaman Utama.....	48
Tabel 12. Hasil Pengujian Fungsional Halaman Beranda	50
Tabel 13. Hasil Pengujian Fungsional Halaman Hasil Klasifikasi Kata.....	51
Tabel 14. Hasil Pengujian Fungsional Halaman Lihat Detail	53
Tabel 15. Hasil Pengujian Fungsional Halaman Kamus Kata Dasar	54
Tabel 16. Hasil Pengujian Fungsional Halaman Kamus Kata Asing.....	54
Tabel 17. Hasil Pengujian Fungsional Halaman Kamus Nama	55
Tabel 18. Hasil Pengujian Fungsional Halaman Cara Pakai	55
Tabel 19. Hasil Pengujian Fungsional Halaman Tentang	56
Tabel 20. Hasil Pengujian Non Fungsional	57

DAFTAR GAMBAR

	halaman
Gambar 1. Model Waterfall (Sommerville, 2007)	12
Gambar 2. Contoh Whitespace Insensitivity	16
Gambar 3. Contoh Tokenisasi	16
Gambar 4. Contoh Filtering.....	17
Gambar 5. Tahapan Penelitian.....	19
Gambar 6. Flowchart Alur Sistem.....	21
Gambar 7. Rancangan Halaman Utama	23
Gambar 8. Rancangan Halaman Klasifikasi.....	24
Gambar 9. Rancangan Halaman Lihat Detail.....	24
Gambar 10. Rancangan Halaman Kamus Kata Dasar	25
Gambar 11. Rancangan Halaman Kamus Kata Asing.....	26
Gambar 12. Rancangan Halaman Kamus Nama	26
Gambar 13. Rancangan Halaman Cara Pakai.....	27
Gambar 14. Rancangan Halaman Tentang.....	28
Gambar 15. Rancangan Basis Data.....	28
Gambar 16. Rancangan Struktur Tabel jumlahkata.....	29
Gambar 17. Rancangan Struktur Tabel jumlahkata2.....	29
Gambar 18. Rancangan Struktur Tabel kata_dasar	29
Gambar 19. Rancangan Struktur Tabel kata_dasar_asing.....	30

Gambar 20. Rancangan Struktur Tabel nama.....	30
Gambar 21. Rancangan Struktur Tabel <i>tb_stoplist</i>	30
Gambar 22. Rancangan Struktur Tabel upload	30
Gambar 23. Tampilan Halaman Utama.....	40
Gambar 24. Tampilan Halaman Klasifikasi	40
Gambar 25. Halaman Lihat Detail.....	41
Gambar 26. Halaman Kamus Kata	41
Gambar 27. Halaman Kamus Kata Asing	42
Gambar 28. Halaman Kamus.....	42
Gambar 29. Halaman Cara Penggunaan.....	43
Gambar 30. Halaman Tentang.....	43

DAFTAR KODE PROGRAM

	halaman
Kode Program 1. Fungsi Pembacaan Dokumen Bertipe .docx	45
Kode Program 2. Fungsi Bersih Dari Tanda Baca Dokumen 1	45
Kode Program 3. Fungsi Bersih Dari Tanda Baca Dokumen 2	45
Kode Program 4. Fungsi Hapus Spasi Ganda	46
Kode Program 5. Fungsi Lower Case	46
Kode Program 6. Fungsi Tokenisasi	46
Kode Program 7. Fungsi Filtering.....	47
Kode Program 8. Fungsi Dictionary Lookup.....	47

I. PENDAHULUAN

1.1 Latar Belakang dan Masalah

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.

Dalam penggunaannya secara baku, 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 kelas nya 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 (Utami, 2017).

Peneliti memilih dokumen dengan ekstensi *word* (.docx) dikarenakan banyaknya penggunaan program pengolah kata *Microsoft Word* versi 2007 hingga yang terakhir yaitu 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. (Budiman & Izzatul, 2017).

Pada penelitian Ademariana dkk, (2020), menjelaskan topik penelitian Analisis Clustering Jenis Kata pada Text Laporan Kegiatan Kuliah Kerja Nyata (KKN) Universitas Lampung menggunakan *Word2vec* yang menjelaskan analisa dan klasterisasi pada kegiatan mahasiswa KKN Universitas Lampung dalam teks Bahasa Indonesia menggunakan *Word2Vec* berdasarkan kelas kata. Pada penelitian ini dilakukan pengembangan sistem klasifikasi kata, nama dan istilah asing pada dokumen bertipe *word* (.docx).

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang sebelumnya, maka rumusan masalah pada penelitian ini adalah bagaimana mengidentifikasi jenis kata berdasarkan kamus pada dokumen bertipe *word* (.docx).

1.3 Tujuan Penelitian

Adapun tujuan penelitian ini yaitu untuk mengidentifikasi jenis kata berdasarkan kamus pada dokumen bertipe *word* (.docx).

1.4 Batasan Masalah

Adapun batasan masalah dari penelitian ini adalah, sebagai berikut:

1. Sistem hanya memproses dokumen bertipe word (*.docx*).
2. Sistem ini diimplementasikan berbasis *web*, dengan bahasa pemrograman PHP (*Hypertext Preprocessor*) dan sistem basis data *MySQL*.
3. Bagian dokumen teks yang diproses adalah *file* teks digital yang bersifat plain teks, yaitu *file* teks yang terdiri dari huruf dan angka, tanpa mencakup gambar, tabel dan sejenisnya.
4. Penggunaan *utility* sistem ini mencakup kata kerja, kata benda, kata sifat, konjungsi, numeralia, preposisi, pronomina, adverbial, interjeksi, data nama dan istilah asing. Apabila kata yang terdeteksi yang tidak terdapat dalam kamus yang digunakan maka dinyatakan tidak ada kelas kata atau kosong.

1.5 Manfaat

Adapun manfaat dari sistem ini adalah sebagai berikut:

1. Mempermudah dalam identifikasi jenis kata berdasarkan kamus pada dokumen bertipe word (*.docx*).
2. Mengefisienkan waktu dalam mengidentifikasi jenis kata berdasarkan kamus pada dokumen bertipe word (*.docx*).
3. Sebagai bahan pembelajaran dan referensi untuk peneliti selanjutnya.

II. TINJAUAN PUSTAKA

2.1 Pembagian Kelas Kata

Penyelidikan mengenai kelas kata dalam bahasa tidak dapat mengabaikan uraian yang telah diberikan dalam buku atau karangan mengenai kelas kata dalam bahasa Melayu dan Indonesia. Kelas kata adalah golongan kata yang mempunyai kesamaan dalam perilaku formalnya; klasifikasi atas nomina, verba, adjektiva, dan sebagainya. Sedangkan menurut Kamus Besar Bahasa Indonesia V (Kbbi, 2016), kelas kata merupakan golongan (kategori) kata berdasarkan bentuk, fungsi atau makna. Kelas kata dibedakan menjadi beberapa sebagai berikut:

1. **Nomina (Kata Benda)**

Nomina atau kata benda adalah kelas kata yang menyatakan nama dari suatu tempat dan segala hal yang dibendakan.

2. **Verba (Kata Kerja)**

Ciri dari verba atau kata kerja biasanya dibatasi kata-kata yang menyatakan perbuatan atau tindakan. Verba atau kata kerja adalah kata-kata yang menyatakan perbuatan, tindakan, proses, gerak, dan keadaan.

3. **Adjektiva (Kata Sifat)**

Adjektiva atau kata sifat adalah kelas kata yang mengubah nomina atau pronominal, biasanya dengan menjelaskannya atau membuatnya menjadi lebih spesifik. Adjektiva dapat menerangkan kuantitas, kecukupan, urutan, kualitas, maupun penekanan suatu kata.

4. **Adverbia (Kata Keterangan)**

Adverbia adalah kata yang dipakai untuk memberikan verba, adjektiva, preposisi, atau adverbia lain, misalnya sangat, lebih, tidak, dan sebagainya.

5. **Pronomina (Kata Ganti)**

Pronomina adalah salah satu jenis kata yang dipakai untuk dapat menggantikan posisi kata benda atau juga orang dalam suatu kalimat. Berdasarkan fungsinya

dapat dikatakan bahwa pronomina menduduki posisi yang umum diduduki oleh nomina, seperti subjek, objek dan juga predikat.

6. Preposisi (Kata Depan)

Menurut Rapar, dalam bukunya yang berjudul *Pengantar Logika, Asas-Asas Penalaran* mengatakan bahwa preposisi adalah istilah yang digunakan untuk kalimat pernyataan yang memiliki arti penuh dan utuh. Hal ini menyatakan bahwa suatu kalimat harus dapat dipercaya, diasingkan, disangkal, atau dibuktikan kebenarannya.

7. Konjungsi

Konjungsi atau kata hubung adalah kata yang digunakan untuk menghubungkan kata dengan kata, ungkapan dengan ungkapan, atau kalimat dengan kalimat.

8. Numeralia

Numeralia atau kata bilangan adalah kata yang menyatakan jumlah benda atau urutannya dalam suatu deretan. Sedangkan menurut Kamus Besar Bahasa Indonesia, numeralia berarti kata atau frasa yang menunjukkan bilangan atau kuantitas.

9. Interjeksi

Interjeksi atau kata seru adalah kata yang mengungkapkan perasaan dan maksud seseorang, misalnya ah dan aduh, atau melambangkan tiruan bunyi, misalnya meong. Bentuk ini biasanya tak dapat diberi afiks dan tidak memiliki dukungan sintaksis dengan bentuk lain.

2.2 Sistem

Sistem adalah rangkaian dari dua atau lebih komponen-komponen yang saling berhubungan, yang berinteraksi untuk mencapai suatu tujuan. Sebagian besar sistem terdiri dari subsistem yang lebih kecil yang mendukung sistem yang lebih besar. Adapun tujuan sistem yaitu menjelaskan tujuan sistem merupakan target atau sasaran akhir yang ingin dicapai oleh suatu sistem (Susanto, 2013). Terdapat beberapa definisi sistem menurut para ahli, di antaranya:

Menurut Maniah (2017), sistem dapat didefinisikan sebagai elemen-elemen berupa data, jaringan kerja dari prosedur-prosedur yang saling berhubungan, sumber daya manusia, teknologi baik *hardware* maupun *software* yang saling berinteraksi sebagai satu kesatuan untuk mencapai tujuan atau sasaran tertentu yang sama.

2.3 Database

Definisi *database* menurut Warsito *et al.*, (2015), adalah struktur penyimpanan data, untuk menambah, mengakses dan memproses data yang disimpan dalam sebuah *database* komputer, diperlukan sistem manajemen *database* seperti *MySQL* server. Sedangkan menurut Rusdiana & Ifran, (2014), *database* merupakan kumpulan informasi yang disimpan dalam komputer secara sistematis untuk memperoleh informasi dari basis data. Adapun manfaat dari *database* ini yaitu sebagai berikut:

1. Mengurangi duplikasi data.
2. Hubungan data dapat ditingkatkan.
3. Mengurangi pemborosan tempat simpanan luar.
4. Integritas, dan konsistensi data dapat selalu terjaga.

Pembuatan sistem informasi sangat berkaitan dengan perancangan *database*, dimana *database* merupakan bagian dari perancangan sebuah sistem informasi. Sistem informasi yang efektif harus mampu memberikan penggunaanya informasi yang cepat dan akurat. Berdasarkan beberapa pendapat tersebut, dapat disimpulkan bahwa *database* adalah kumpulan suatu data pada sistem informasi yang berfungsi sebagai media untuk penyimpanan data dan informasi yang berfungsi sebagai media untuk penyimpanan data dan informasi yang dapat diakses di sistem ataupun aplikasi dengan cepat.

2.4 Hypertext Preprocessor (PHP)

PHP diciptakan oleh Rasmus Lerdorf seseorang pemrogram C pada tahun 1995 merupakan bahasa berbentuk *script* *server-side* yang ditempatkan dalam server dan

diproses di *server*. Hasil berupa kode *Hyper Text Markup Language* (HTML) akan dikirim ke *client*, tempat pemakaian menggunakan *browser* (Kadir, 2008). Penggunaan PHP memungkinkan web dapat dibuat secara dinamis sehingga *maintenance* situs web menjadi lebih baik mudah dan efisien. Sistem kerja dari PHP diawali dengan permintaan yang berasal dari halaman *website* oleh *browser*. Berdasarkan URL atau alamat website dalam jaringan internet, *browser* akan menemukan sebuah alamat dari *web server*, mengidentifikasi halaman yang dikehendaki dan menyampaikan segala informasi yang dibutuhkan oleh *web server*. Selanjutnya *web server* akan mencari berkas yang diminta dan menampilkan isinya di *browser*. *Browser* yang mendapat-kan isinya segera menerjemahkan kode HTML dan menampilkannya (Firman, 2016).

PHP memiliki banyak kelebihan yang tidak dimiliki oleh bahasa *script* sejenis, PHP tidak terbatas pada hasil keluaran HTML dan PHP memiliki kemampuan untuk mengolah gambar, file PDF, dan *movie flash*. PHP juga dapat menghasilkan teks seperti XHTML dan file XML lainnya. Salah satu fitur yang dapat diandalkan oleh PHP adalah dukungannya terhadap banyak *database*, salah satunya adalah *MySQL*. Kelebihan-kelebihan PHP adalah sebagai berikut:

1. Kode Program terintegrasi dengan file HTML, sehingga pengembangan bisa berkonsentrasi langsung pada tampilan webnya.
2. Integrasi yang sangat luas ke berbagai *server database*.
3. Tidak ada proses *compiling* dan *linking*.

2.5 *MariaDB*

Definisi *MariaDB* menurut Mahendra dkk, (2017), *MariaDB* bersifat *open source* dan dikembangkan oleh pengembang yang sama dari *My Structured Query language* (*MySQL*). *MariaDB* merupakan salah satu *database server* yang digunakan untuk menyimpan data dan manajemen data. *MariaDB* dikembangkan oleh komunitas pengembang yang sebelumnya berkontribusi untuk *database MySQL* (Warman & Ramdaniansyah, 2018). Secara sintaks, semua perintah yang dapat dijalankan pada *MySQL* juga bisa berjalan pada *MariaDB*.

Selain itu, semua kemampuan *MySQL* juga dimiliki oleh *MariaDB*, adapun perbedaannya yaitu *MariaDB* menyertakan semua *engine open source* yang populer dan memiliki peningkatan kinerja dan kecepatan dibandingkan dengan *MySQL* (Afdhal dkk., 2017). Kelebihan dari *MariaDB* yaitu:

1. Portabilitas, *MariaDB* dapat berjalan stabil pada berbagai sistem operasi seperti Microsoft Windows, Linux, dan lain-lain.
2. Perangkat lunak sumber terbuka.
3. *Multi-user*, *MariaDB* dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. Ragam tipe data, *MariaDB* memiliki operator dan fungsi secara penuh yang mendukung perintah *select* dan *where* dalam perintah (*query*).
5. Perintah dan fungsi, *MariaDB* memiliki ragam tipe data yang sangat kaya seperti: *double*, *char*, *text*, *date*, *timestamp* dan lain-lain.
6. Keamanan, *MariaDB* memiliki beberapa lapisan keamanan seperti: nama *host* dan izin akses *user*.
7. Skalabilitas dan pembatasan, *MariaDB* mampu menangani basis data dalam skala besar.

2.6 Flowchart


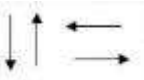


Flowchart merupakan diagram simbol yang menunjukkan arus data dan tahapan operasi dalam sebuah sistem yang digunakan baik oleh *editor* maupun oleh personal sistem. Adapun jenis-jenis *flowchart* yaitu sebagai berikut:






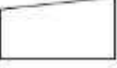

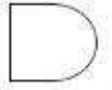
1. *Flowchart* dokumen adalah bagan alir yang hanya terdiri dari simbol-simbol dokumen yang digunakan dalam *flowchart*, tujuannya untuk mengetahui setiap dokumen yang digunakan dalam setiap sistem aplikasi dan mengidentifikasi titik awal dokumen, distribusi dokumen serta titik akhir setiap dokumen.
2. *Flowchart* sistem merupakan bagan yang menunjukkan alur kerja atau yang sedang dikerjakan di dalam sistem secara keseluruhan dan menjelaskan urutan dari prosedur-prosedur yang ada di dalam sistem.

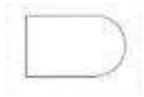
3. *Flowchart* semantik digunakan sebagai alat komunikasi antara analisis sistem dengan seseorang yang tidak familiar dengan simbol-simbol *flowchart* yang konvensional.
4. *Flowchart* program merupakan keterangan yang lebih rinci tentang bagaimana setiap langkah program atau prosedur sesungguhnya dilaksanakan, *flowchart* ini menunjukkan setiap langkah program atau prosedur dalam urutan yang tepat saat terjadi.
5. *Flowchart* proses merupakan teknik penggambaran rekayasa industrial yang memecah dan menganalisis langkah-langkah selanjutnya dalam suatu prosedur atau sistem (Nurpianti dkk., 2013).

Simbol-simbol *flowchart* dapat dilihat pada tabel 1.

Tabel 1. Simbol-simbol *Flowchart*

Nama Simbol	Keterangan
	<i>Terminal Point Symbol</i> / Simbol Titik Terminal adalah simbol yang digunakan sebagai permulaan (<i>start</i>) atau akhir (<i>stop</i>) dari suatu proses.
	<i>Flow Direction Symbol</i> / Simbol Arus adalah simbol ini digunakan guna menghubungkan simbol satu dengan simbol yang lain (<i>connecting line</i>).
	<i>Processing Symbol</i> / Simbol Proses adalah simbol yang digunakan untuk menunjukkan kegiatan yang dilakukan oleh komputer. Pada bidang industri (proses produksi barang), simbol ini menggambarkan kegiatan inspeksi atau yang biasa dikenal dengan simbol inspeksi.
	<i>Decision Symbol</i> / Simbol Keputusan adalah simbol yang digunakan untuk memilih proses atau keputusan berdasarkan kondisi yang ada. Simbol ini biasanya ditemui pada <i>flowchart</i> program.

	<p><i>Input- Output/</i> Simbol Keluar-Masuk adalah simbol yang menunjukkan proses input-output yang terjadi tanpa bergantung dari jenis peralatannya.</p>
	<p><i>Predefined Process/</i> Simbol Proses Terdefinisi adalah simbol yang digunakan untuk menunjukkan pelaksanaan suatu bagian prosedur (sub-proses). Dengan kata lain, prosedur yang terinformasi di sini belum detail dan akan dirinci di tempat lain.</p>
	<p><i>Connector (On-page)</i> adalah simbol yang fungsinya untuk menyederhanakan hubungan antar simbol yang letaknya berjauhan atau rumit bila dihubungkan dengan garis dalam satu halaman.</p>
	<p><i>Connector (Off-page)</i> adalah simbol yang digunakan untuk menghubungkan simbol dalam halaman berbeda. label dari simbol ini dapat menggunakan huruf atau angka.</p>
	<p><i>Preparation Symbol/</i> Simbol Persiapan adalah simbol yang digunakan untuk mempersiapkan penyimpanan di dalam <i>storage</i>.</p>
	<p><i>Manual Input Symbol</i> adalah simbol digunakan untuk menunjukkan input data secara manual menggunakan <i>online</i> keyboard.</p>
	<p><i>Manual Operation Symbol/</i> Simbol Kegiatan Manual simbol yang digunakan untuk menunjukkan kegiatan/proses yang tidak dilakukan oleh komputer.</p>
	<p><i>Display Symbol</i> adalah simbol yang menyatakan penggunaan peralatan <i>output</i>, seperti layar monitor, printer, plotter dan lain sebagainya.</p>

	<p><i>Delay Symbol</i> adalah simbol yang digunakan untuk menunjukkan proses <i>delay</i> (menunggu) yang perlu dilakukan. Seperti menunggu surat untuk diarsipkan dll.</p>
---	---

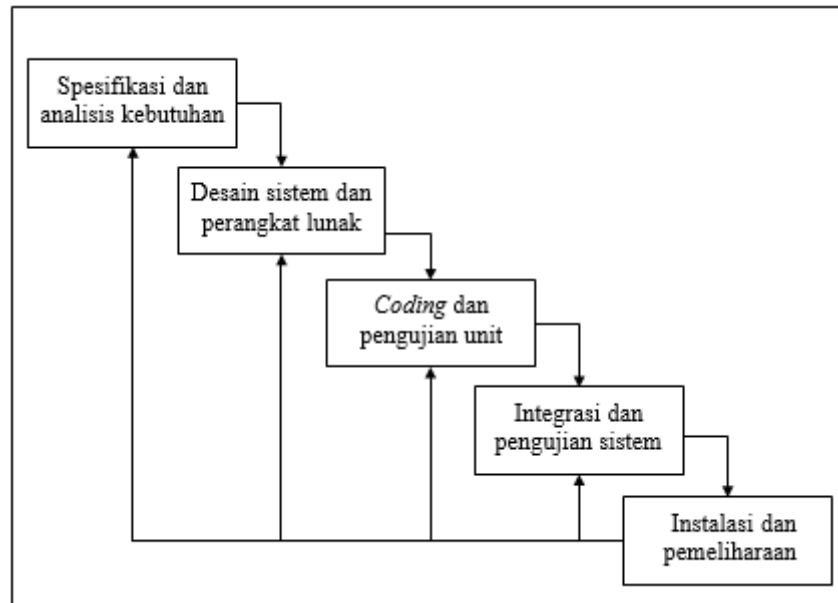
2.7 Waterfall

Dalam pengembangan suatu perangkat lunak, digunakan beberapa model, salah satunya 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.

Model *waterfall* mempunyai keunggulan dalam membangun dalam mengembangkan sistem, antara lain (Sommerville, 2007):

1. Kualitas dari sistem yang dihasilkan akan baik. Ini dikarenakan oleh pelaksanaannya secara bertahap. Sehingga tidak terfokus pada tahapan tertentu.
2. Dokumen pengembangan sistem sangat terorganisir, karena setiap fase harus terselesaikan dengan lengkap sebelum melangkah ke fase berikutnya. Jadi setiap fase atau tahapan akan mempunyai dokumen tertentu.

Sesuai dengan namanya *waterfall* (air terjun) maka tahapan dalam model ini disusun bertingkat dan setiap tahap dilakukan berurutan. Model *waterfall* dapat dilihat pada Gambar 1 (Sommerville, 2007).



Gambar 1. Model *Waterfall* (Sommerville, 2007)

Tahapan model waterfall dapat dijelaskan sebagai berikut:

1. Spesifikasi dan Analisis kebutuhan

Mengumpulkan kebutuhan secara lengkap kemudian dianalisis dan didefinisikan kebutuhan yang harus dipenuhi oleh program yang akan dibangun. Meningkatkan usaha proses pengumpulan persyaratan dan berfokus pada perangkat lunak. Untuk mengetahui program yang akan dibuat, maka harus diketahui cakupan informasi perangkat lunak, seperti fungsi, tingkah laku, kinerja dan antarmuka.

Setelah menganalisis data dan mengelompokkannya berdasarkan jenis datanya maka tahap selanjutnya adalah melakukan analisis kebutuhan sistem. Analisis kebutuhan dilakukan dengan menentukan kebutuhan sistem yang diperlukan untuk mempermudah dalam tahap desain sistem.

2. Desain Sistem dan Perangkat Lunak

Desain dikerjakan setelah kebutuhan selesai dikumpulkan secara lengkap. Tahap ini bertujuan untuk memberikan gambaran apa yang seharusnya

dikerjakan dan bagaimana tampilannya. Desain menspesifikasikan abstraksi sistem dan perangkat lunak serta mendefinisikan arsitektur sistem secara keseluruhan. Proses desain menerjemahkan persyaratan ke dalam representasi perangkat lunak yang bisa diperkirakan demi kualitasnya sebelum memulai pembuatan kode. Tahapan perancangan sistemnya adalah:

- a. Flowchart
- b. Perancangan Antarmuka

3. Coding dan pengujian unit

Peneliti melakukan implementasi dengan menggunakan PHP MySQL. Pada tahap ini, semua algoritma dan proses pada perancangan sistem akan diimplementasikan dalam sebuah aplikasi sebagai wujud dari sistem. Sistem ini akan dikembangkan dalam bahasa pemrograman PHP.

4. Integrasi dan pengujian sistem

Tahap pengujian adalah proses eksekusi suatu program dengan maksud menemukan kesalahan. *Test case* yang baik adalah *test case* yang memiliki probabilitas tinggi untuk menemukan kesalahan yang belum pernah ditemukan sebelumnya. Pengujian yang baik adalah pengujian yang mengungkap semua kesalahan yang belum pernah ditemukan sebelumnya. Ada dua macam metode pengujian, yaitu *black-box* dan *white-box*. Metode *black-box* yaitu menguji fungsionalitas dari perangkat lunak saja tanpa harus mengetahui struktur internal program (*source code*), sedangkan metode *white-box* yaitu menguji fungsionalitas dari perangkat lunak dengan menguji struktur internal program (*source code*). Tahap pengujian ini, Penulis melakukan pengujian sistem dengan metode *black- box*.

5. Instalasi dan pemeliharaan

Umumnya tahap ini merupakan tahap yang paling lama. Saat mengoperasikan program di lingkungannya dan melakukan pemeliharaan, akan melakukan penyesuaian atau perubahan karena adaptasi dengan situasi sebenarnya.

Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya. Perbaiki implementasi unit sistem dan peningkatan jasa sistem sebagai kebutuhan baru. Namun, pada tahap ini penulis tidak menjelaskan mengenai instalasi dan pemeliharaan sistem tersebut.

Dalam pelaksanaan metode waterfall tersebut, dapat terjadi tumpang tindih antar tahap dan pertukaran informasi antara tahap yang satu dengan tahap yang lainnya. Misalnya pada saat desain ditemukan masalah di tahap spesifikasi kebutuhan, pada saat coding ditemukan masalah di tahap desain, dan sebagainya. Proses pengembangan perangkat lunak ini tidak sepenuhnya linear, tetapi memperbolehkan serangkaian iterasi dalam pelaksanaannya (Sommerville, 2007).

2.8 Black-Box Testing

Pengujian adalah suatu proses pelaksanaan suatu program dengan tujuan menemukan suatu kesalahan. Suatu *test* yang sukses adalah bila *test* tersebut membongkar suatu kesalahan yang awalnya tidak ditemukan, salah satu jenis pengujian adalah *black-box testing* (Rahmadi dkk., 2015). *Black-box testing* merupakan pengujian suatu aplikasi yang membahas sisi luar suatu aplikasi perangkat lunak, di mana mulai dari tampilan hingga akses inputan, dalam strategi *testing*. *Black-box* memiliki beberapa metode antara lain:

1. *Equivalence partitioning*, membahas tentang *testing black-box* dalam aspek validasi inputan dilihat dari *valid class*, pengamatan isi inputan dan akurasi inputan.
2. *Boundary value analysis*, membahas kepada *testing black-box* dalam aspek keseluruhan menu dan modul, sehingga dapat diketahui sisi kesalahannya (Hidayat & Mutaqqin, 2020).

Black-box testing cenderung untuk menemukan hal-hal berikut:

1. Fungsi yang tidak benar atau tidak ada.
2. Kesalahan antarmuka.
3. Kesalahan pada struktur data dan akses basis data.

4. Kesalahan *performansi*.
5. Kesalahan inisialisasi dan terminasi (Rahmadi dkk., 2015).

Keunggulan dan kelemahan *black-box testing* yaitu:

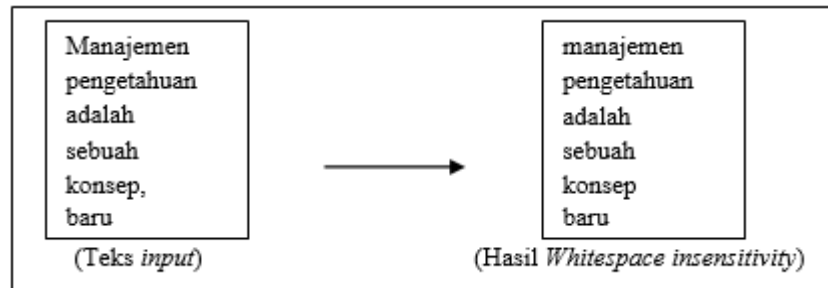
1. Keunggulan *black-box testing* yaitu, meskipun dalam pelaksanaan *testing* dapat menguji keseluruhan fungsional perangkat namun *formal black-box testing* yang sebenarnya dapat memilih subset tes yang secara efektif dan efisien dapat menemukan cacat, dengan cara *black-box testing* dapat membantu memaksimalkan program.
2. Kelemahan *black-box testing* yaitu, ketika tester melakukan *black-box testing*, tester tidak akan yakin apakah perangkat lunak yang diuji telah benar-benar lolos pengujian, hal ini karena kemungkinan masih ada beberapa jalur eksekusi yang belum pernah diuji untuk menemukan cacat perangkat lunak menggunakan *black-box testing* (Indriyani & Sihite, 2015).

2.9 Preprocessing

Preprocessing adalah mempersiapkan dokumen teks yang tidak terstruktur menjadi data terstruktur yang siap digunakan untuk proses selanjutnya (Ipmawati dkk, 2017). Tahapan *text preprocessing* yang dilakukan diantaranya adalah:

2.9.1 Whitespace Insensitivity

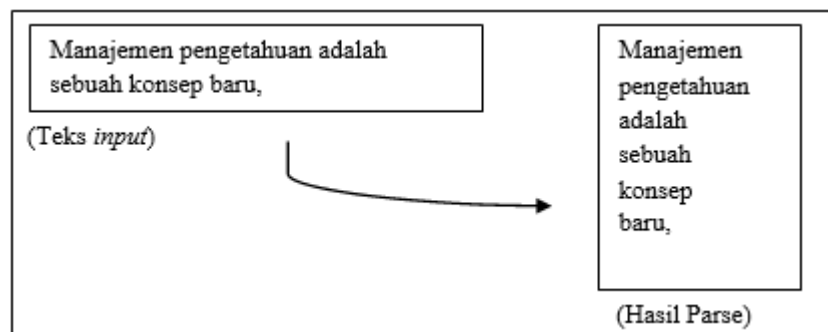
Whitespace insensitivity merupakan tahapan yang menghilangkan karakter-karakter tertentu seperti tanda baca dan mengubah semua *token* ke bentuk huruf kecil (*lower case*). Contoh proses *Whitespace insensitivity* dapat dilihat pada Gambar 2.



Gambar 2. Contoh *Whitespace Insensitivity*

2.9.2 Tokenisasi

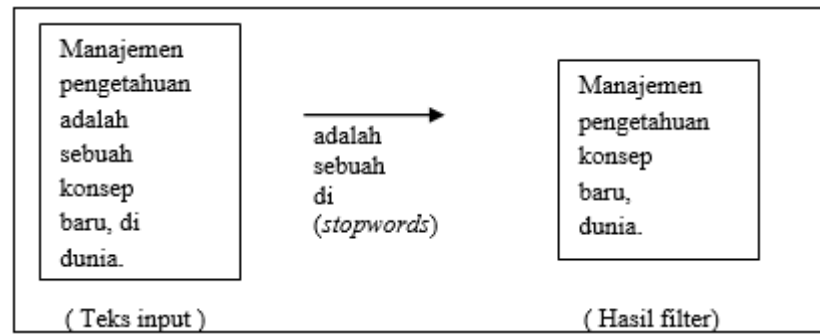
Tokenisasi secara garis besar dapat diartikan sebagai proses pemecahan sekumpulan karakter dalam suatu teks ke dalam suatu kata. Proses untuk membagi teks yang dapat berupa kalimat, paragraf atau dokumen menjadi token-token atau bagian-bagian tertentu. *Tokenisasi* merupakan proses pemisahan suatu rangkaian karakter berdasarkan karakter spasi (Fatkhul, 2012). Contoh proses *tokenisasi* terdapat dalam Gambar 3.



Gambar 3. Contoh *Tokenisasi*

2.9.3 Filtering

Filtering merupakan tahap 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*). *Stoplist/stopwords* adalah kata-kata yang tidak deskriptif yang dapat dibuang dalam pendekatan *bags-of-words*. Contoh *stopwords* adalah “yang”, “dan”, “di”, “dari” dan sebagainya. Contoh proses *filtering* terdapat dalam Gambar 4.



Gambar 4. Contoh *Filtering*

2.10 Dictionary Lookup

Dictionary lookup merupakan metode yang melakukan pencarian secara sederhana untuk melihat apakah input kata yang dimasukkan berada dalam kamus atau daftar kata yang ada. Jika tidak ada maka kata tersebut dianggap tidak mempunyai kelas kata. (Maghfira dkk, 2017).

III. METODE PENELITIAN

3.1 Waktu dan Tempat Penelitian

Penelitian ini dilaksanakan pada bulan April 2020. Tempat penelitian dilaksanakan di laboratorium Jurusan Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung.

3.2 Alat dan Bahan Penelitian

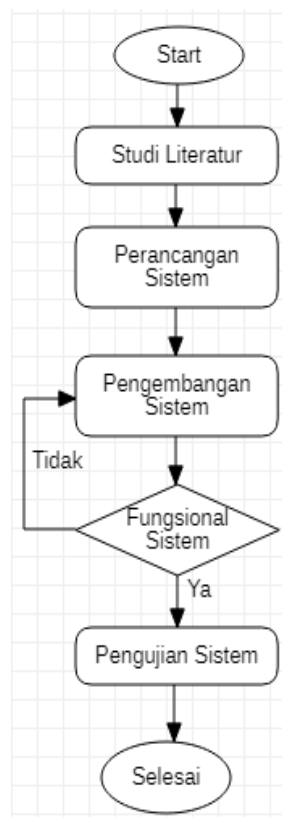
Alat yang digunakan dalam penelitian ini yaitu berupa perangkat keras (*hardware*) dan perangkat lunak (*software*) yang digunakan sebagai alat pendukung dalam melaksanakan penelitian. Adapun perangkat tersebut adalah sebagai berikut:

1. *Hardware* berfungsi sebagai perangkat keras untuk mengembangkan dan mengumpulkan data pada aplikasi ini adapun *hardware* yang digunakan adalah sebagai berikut:
 - a. *Processor*: INTEL CORE i5-7200U 2.5GHz with Turbo Boost up to 3.1GHz.
 - b. RAM: 4.00 GB DDR4 Memory.
 - c. *Storage*: 931.51 GB.
 - d. *System type*: Windows 10 Enterprise, 64-bit Operating System, x64-based processor.
2. *Software* berfungsi sebagai perangkat lunak dalam menerapkan rancangan yang telah dibuat, dibutuhkan beberapa *software* untuk membuat program aplikasi web ini adalah sebagai berikut:
 - a. Program editor: Visual Studio Code
 - b. Web Browser: Google Chrome
 - c. Database server: MariaDB
 - d. Web server: Apache

- e. *Microsoft Word*
- f. *Star UML*

3.3 Tahapan Penelitian

Prosedur penelitian yang dilakukan pada penelitian ini terdiri dari beberapa tahap dapat dilihat pada Gambar 3.3.



Gambar 5. Tahapan Penelitian

Berdasarkan Gambar 5, penjelasan dari beberapa tahapan yang akan dilakukan pada saat penelitian yaitu:

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 didapatkan dari berbagai sumber seperti jurnal, buku, internet dan pustaka.

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*).

3. Pengembangan Sistem

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

4. Implementasi

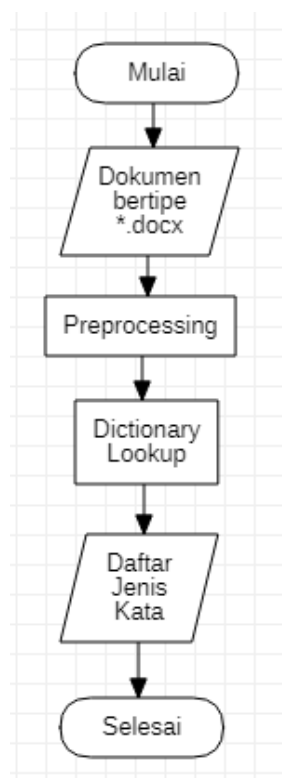
Pada tahap ini data atau informasi kebutuhan penelitian yang telah di peroleh diterapkan pada pengembangan *utility* sistem.

5. Pengujian Sistem

Pada tahap pengujian sistem ini dilakukan dengan melakukan pengujian secara fungsional dari fitur yang disediakan 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. Pada penelitian ini pengujian sistem menggunakan metode *Black-box Testing* dengan teknik *Equivalence Partitioning*.

3.4 Perancangan Proses Sistem U-Correct

Perancangan proses sistem klasifikasi kata berdasarkan kamus pada dokumen bertipe word (*.docx*) merupakan gambaran dari kebutuhan dalam perangkat lunak sebelum memulai pembuatan kode.



Gambar 6. *Flowchart* Alur Sistem

Berdasarkan Gambar 7 dapat dijelaskan proses dari *utility* ini yaitu:

3.4.1 Proses *Upload* Dokumen

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

3.4.2 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. Daftar *stopwords* ada pada Lampiran 1.

3.4.3 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.

3.5 Rancangan Antarmuka Sistem

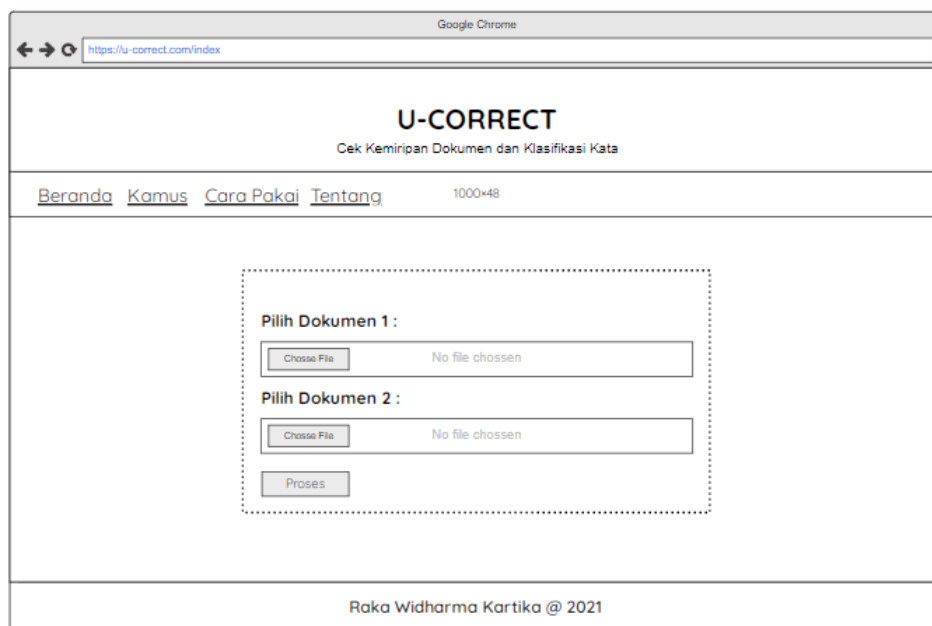
Antarmuka dibuat sederhana untuk memudahkan pengguna dalam penggunaannya. Adapun rancangan antarmuka sistem klasifikasi kata berdasarkan kamus pada dokumen bertipe word (*.docx*) yaitu sebagai berikut:

1. Antarmuka Halaman Utama
2. Antarmuka Halaman Klasifikasi
3. Antarmuka Halaman Lihat Detail
4. Antarmuka Halaman Kamus Kata Dasar
5. Antarmuka Halaman Kamus Kata Asing
6. Antarmuka Halaman Kamus Nama
7. Antarmuka Halaman Cara Pakai
8. Antarmuka Halaman Tentang

3.5.1 Rancangan Halaman Utama

Terdapat 3 menu pada halaman utama yaitu Menu Beranda, Kamus, Cara Pakai dan Tentang. Halaman Utama Sistem terdapat pada Menu Beranda. Menu Beranda

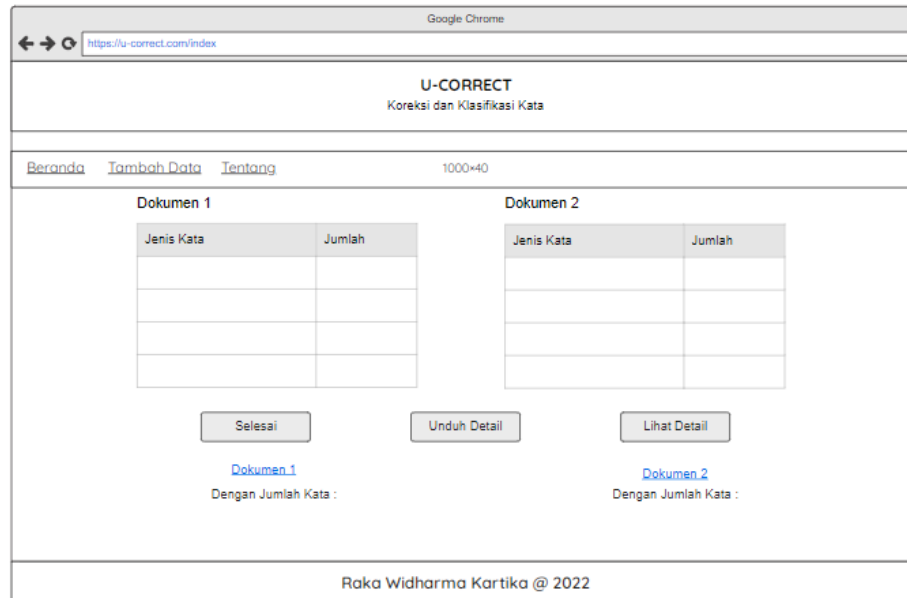
menampilkan deskripsi singkat sistem sekaligus menampilkan *form upload* dokumen. Ada 2 tombol *choose file* masing-masing pada dokumen 1 dan 2 berfungsi untuk memasukkan dokumen yang bertipe word (.docx). dan *button* Proses untuk perancangan halaman ini dapat dilihat pada Gambar 9.



Gambar 7. Rancangan Halaman Utama

3.5.2 Rancangan Halaman Klasifikasi

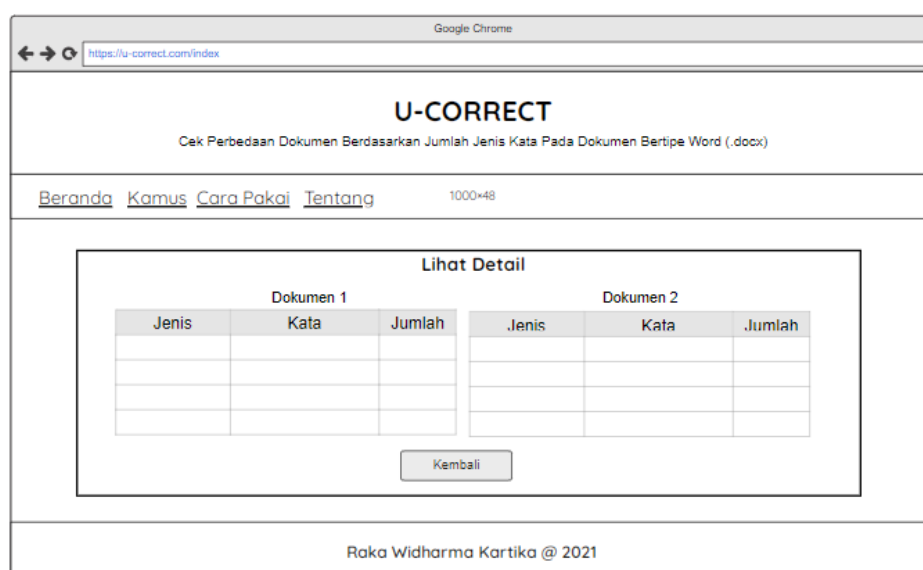
Halaman Klasifikasi berfungsi untuk menampilkan jenis kata pada dokumen 1 dan 2 setelah melalui tahap *pre-processing*. Hasil data yang sudah diproses ditampilkan dalam bentuk tabel yang terdiri dari Jenis Kata dan Jumlah. Terdapat tombol Selesai untuk kembali ke Halaman Utama, rancangan halaman ini dapat dilihat pada Gambar 10.



Gambar 8. Rancangan Halaman Klasifikasi

3.5.3 Rancangan Halaman Lihat Detail

Pada halaman lihat detail menampilkan daftar tabel dokumen pertama dan kedua. Kolom yang pertama yaitu jenis kata, kolom kedua berisi kata, dan kolom ketiga berisi frekuensi jumlah kata. Rancangan Halaman Lihat Detail dapat dilihat pada Gambar 11.



Gambar 9. Rancangan Halaman Lihat Detail

3.5.4 Rancangan Halaman Kamus Kata Dasar

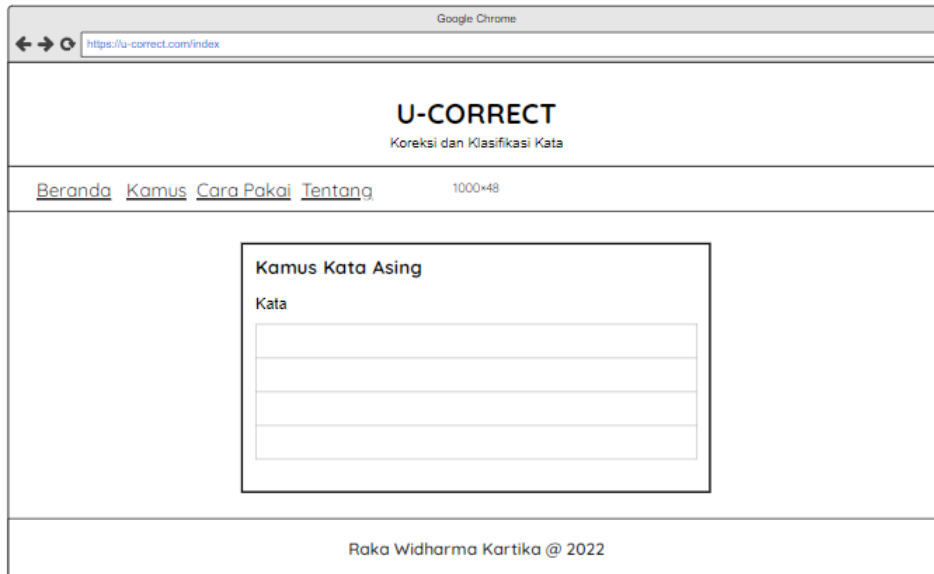
Halaman Kamus Kata Dasar berfungsi untuk menampilkan kamus yang digunakan dalam proses pengecekan kata dalam *utility* ini. Rancangan halaman Kamus Kata Dasar dapat dilihat pada Gambar 12.



Gambar 10. Rancangan Halaman Kamus Kata Dasar

3.5.5 Rancangan Halaman Kamus Kata Asing

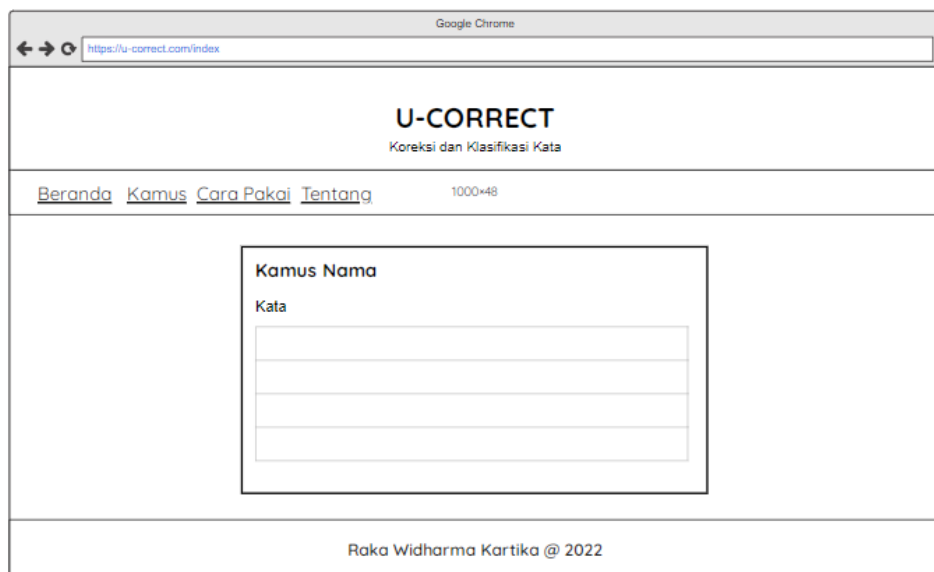
Halaman Kamus Kata Asing berfungsi untuk menampilkan daftar istilah asing yang digunakan dalam proses pengecekan kata dalam *utility* ini. Rancangan halaman Kamus Kata Asing dapat dilihat pada Gambar 13.



Gambar 11. Rancangan Halaman Kamus Kata Asing

3.5.6 Rancangan Halaman Kamus Nama

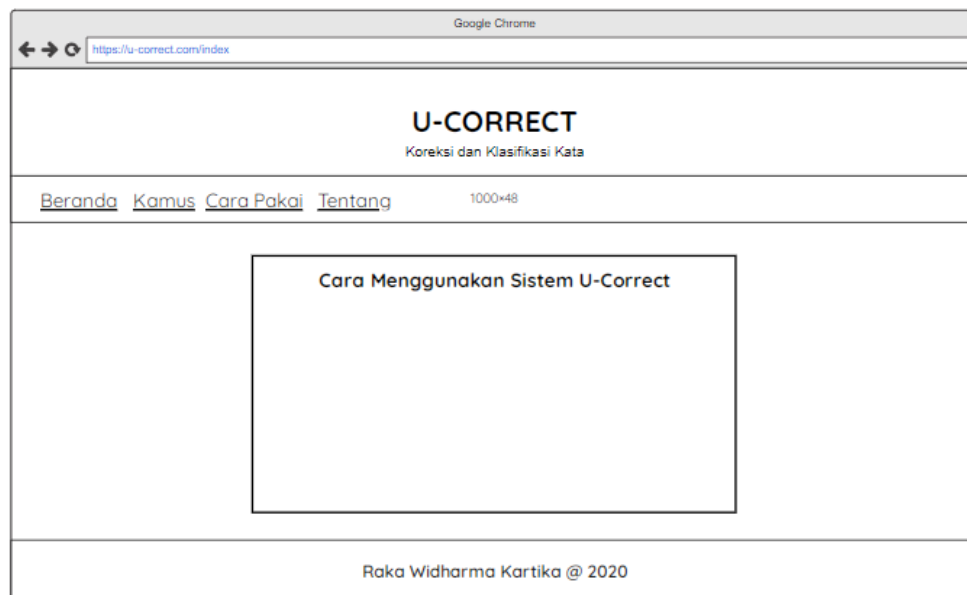
Halaman Kamus Nama berfungsi untuk menampilkan daftar nama yang sudah dimasukkan ke dalam *database*. Digunakan dalam proses pengecekan dokumen dalam sistem. Rancangan Antarmuka Halaman Kamus Nama dapat dilihat pada Gambar 14.



Gambar 12. Rancangan Halaman Kamus Nama

3.5.7 Rancangan Halaman Cara Pakai

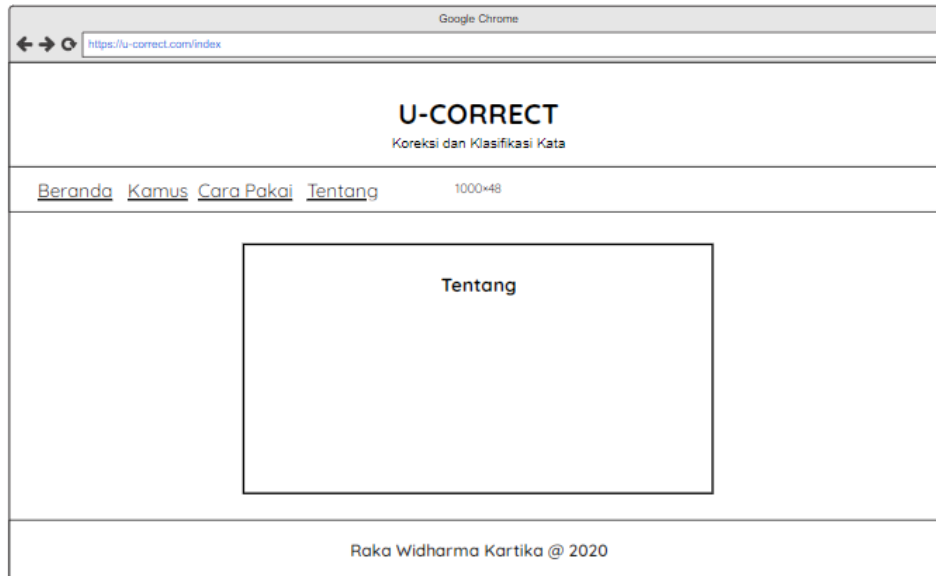
Halaman Cara Pakai berfungsi untuk memberikan informasi mengenai cara penggunaan sistem klasifikasi kata berdasarkan kamus pada dokumen bertipe word (.docx). Rancangan halaman ini dapat dilihat pada Gambar 15.



Gambar 13. Rancangan Halaman Cara Pakai

3.5.8 Rancangan Halaman Tentang

Halaman tentang berfungsi untuk memberikan informasi mengenai *utility* dan berisi informasi mengenai sistem klasifikasi kata berdasarkan kamus pada dokumen bertipe word (.docx). Rancangan halaman ini dapat dilihat pada Gambar 16.



Gambar 14. Rancangan Halaman Tentang

3.6 Rancangan Tabel Basis Data

Pengembangan sistem klasifikasi kata berdasarkan kamus pada dokumen bertipe word (.docx) menggunakan basis data MariaDB, dalam basis data ini terdapat tujuh tabel, yaitu *jumlahkata*, *jumlahkata2*, *kata_dasar*, *kata_dasar_asing*, *nama*, *tb_stoplist* dan *upload*. Rancangan basis data dapat dilihat pada Gambar 3.10.

```

+-----+
| Tables_in_kamus |
+-----+
| jumlahkata      |
| jumlahkata2    |
| kata_dasar      |
| kata_dasar_asing |
| nama            |
| tb_stoplist     |
| upload          |
+-----+

```

Gambar 15. Rancangan Basis Data

Tabel *jumlahkata* berfungsi untuk menyimpan data jumlah kata pada dokumen pertama yang dimasukkan. Terdapat empat *field*, yaitu *field* kata untuk menyimpan kata, *field* keterangan untuk menyimpan keterangan benar untuk konfirmasi kata berdasarkan jenis kata, *field* jumlah untuk menyimpan jumlah kata yang muncul,

field jenis_kata untuk menyimpan jenis kata. Rancangan struktur tabel data nama dapat dilihat pada Gambar 3.11.

Field	Type	Null	Key	Default	Extra
kata	varchar(25)	NO		NULL	
keterangan	varchar(100)	NO		NULL	
jumlah	int(11)	NO		NULL	
jenis_kata	varchar(100)	NO		NULL	

Gambar 16. Rancangan Struktur Tabel *jumlahkata*

Tabel *jumlahkata2* berfungsi untuk menyimpan hasil pemrosesan dokumen kedua. Terdapat empat *field*, yaitu *field* kata untuk menyimpan kata, *field* keterangan untuk menyimpan keterangan benar untuk konfirmasi kata berdasarkan jenis kata, *field* jumlah untuk menyimpan jumlah kata yang muncul, *field* jenis_kata untuk menyimpan jenis kata. Rancangan struktur tabel data nama dapat dilihat pada Gambar 3.12.

Field	Type	Null	Key	Default	Extra
kata	varchar(25)	NO		NULL	
keterangan	varchar(100)	NO		NULL	
jumlah	int(11)	NO		NULL	
jenis_kata	varchar(100)	NO		NULL	

Gambar 17. Rancangan Struktur Tabel *jumlahkata2*

Tabel *kata_dasar* berfungsi untuk menyimpan data kamus kata dasar, rancangan struktur tabel kata dasar dapat dilihat pada Gambar 3.13.

Field	Type	Null	Key	Default	Extra
id_katadasar	int(10)	NO	PRI	NULL	auto_increment
katadasar	varchar(15)	NO		NULL	
tipe_katadasar	varchar(10)	NO		NULL	

Gambar 18. Rancangan Struktur Tabel *kata_dasar*

Tabel kata dasar asing berfungsi untuk menyimpan data kamus kata asing, rancangan struktur tabel keterangan tempat dapat dilihat pada Gambar 3.13.


```

+-----+-----+-----+-----+-----+-----+
| Field      | Type           | Null | Key | Default | Extra           |
+-----+-----+-----+-----+-----+-----+
| no         | int(11)        | NO   | PRI | NULL    | auto_increment |
| kata_dasar | varchar(255)   | NO   |     | NULL    |                 |
+-----+-----+-----+-----+-----+-----+

```

Gambar 19. Rancangan Struktur Tabel *kata_dasar_asing*

Tabel data nama berfungsi untuk menyimpan data kamus nama, rancangan struktur tabel data nama dapat dilihat pada Gambar 3.14.

```

+-----+-----+-----+-----+-----+-----+
| Field      | Type           | Null | Key | Default | Extra           |
+-----+-----+-----+-----+-----+-----+
| id         | int(11)        | NO   | PRI | NULL    | auto_increment |
| ket_nama  | varchar(255)   | NO   |     | NULL    |                 |
+-----+-----+-----+-----+-----+-----+

```

Gambar 20. Rancangan Struktur Tabel *nama*

Tabel *tb_stoplist* berfungsi untuk menyimpan data *stoplist*, rancangan struktur tabel *tb_stoplist* dapat dilihat pada Gambar 3.15.

```

+-----+-----+-----+-----+-----+-----+
| Field      | Type           | Null | Key | Default | Extra           |
+-----+-----+-----+-----+-----+-----+
| id_stoplist | int(10)        | NO   | PRI | NULL    | auto_increment |
| stoplist   | varchar(50)    | NO   |     | NULL    |                 |
+-----+-----+-----+-----+-----+-----+

```

Gambar 21. Rancangan Struktur Tabel *tb_stoplist*

Tabel *upload* berfungsi untuk menyimpan sementara nama dokumen yang akan di proses, rancangan struktur tabel *upload* dapat dilihat pada Gambar 3.14.

```

+-----+-----+-----+-----+-----+-----+
| Field | Type           | Null | Key | Default | Extra           |
+-----+-----+-----+-----+-----+-----+
| id    | int(10)        | NO   | PRI | NULL    | auto_increment |
| nama  | varchar(100)   | NO   |     | NULL    |                 |
+-----+-----+-----+-----+-----+-----+

```

Gambar 22. Rancangan Struktur Tabel *upload*

3.7 Pengujian Sistem

Pada tahap pengujian sistem ini dilakukan dengan melakukan pengujian secara fungsional dari fitur yang disediakan 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. Pada penelitian ini rancangan pengujian sistem akan dilakukan dengan teknik pengujian *Black-box Testing* dengan teknik *Equivalence Partitioning* dan skala yang digunakan yaitu skala *likert*. Rencana pengujian *Black-box Testing (Equivalence Partitioning)* dapat dilihat pada Tabel 2.

Tabel 2. *Test Scenario Menu*

1.	<i>Module/Function</i>	Menu Beranda
	<i>Assumption</i>	Sistem sudah diakses
	<i>Pre-condition</i>	Sistem <i>running</i> dan <i>accessible</i>
	<i>Test Case</i>	Klik Menu Beranda
	<i>Expected Result</i>	Sistem dapat menampilkan halaman Beranda
	<i>Actual Result</i>	
	<i>Status</i>	
	<i>Test Result</i>	
2.	<i>Module/Function</i>	Menu Kamus Kata Dasar dan Berimbuhan
	<i>Assumption</i>	Sistem sudah diakses
	<i>Pre-condition</i>	Sistem <i>running</i> dan <i>accessible</i>
	<i>Test Case</i>	Klik pada Menu Kamus Kata Dasar dan Berimbuhan

	<i>Expected Result</i>	Sistem dapat menampilkan halaman Kamus Kata Dasar dan Berimbuhan
	<i>Actual Result</i>	
	<i>Status</i>	
	<i>Test Result</i>	
3.	<i>Module/Function</i>	Menu Kamus Kata Asing
	<i>Assumption</i>	Sistem sudah diakses
	<i>Pre-condition</i>	Sistem <i>running</i> dan <i>accessible</i>
	<i>Test Case</i>	Klik pada Menu Kamus Kata Asing
	<i>Expected Result</i>	Sistem dapat menampilkan halaman Kamus Kata Asing
	<i>Actual Result</i>	
	<i>Status</i>	
	<i>Test Result</i>	
4.	<i>Module/Function</i>	Menu Kamus Nama
	<i>Assumption</i>	Sistem sudah diakses
	<i>Pre-condition</i>	Sistem <i>running</i> dan <i>accessible</i>
	<i>Test Case</i>	Klik pada Menu Kamus Nama
	<i>Expected Result</i>	Sistem dapat menampilkan halaman Kamus Nama
	<i>Actual Result</i>	
	<i>Status</i>	
	<i>Test Result</i>	
5.	<i>Module/Function</i>	Menu Cara Pakai
	<i>Assumption</i>	Sistem sudah diakses

	<i>Pre-condition</i>	Sistem <i>running</i> dan <i>accessible</i>
	<i>Test Case</i>	Klik pada Menu Cara Pakai
	<i>Expected Result</i>	Sistem dapat menampilkan halaman Cara Pakai
	<i>Actual Result</i>	
	<i>Status</i>	
	<i>Test Result</i>	
6.	<i>Module/Function</i>	Menu Tentang
	<i>Assumption</i>	Sistem sudah diakses
	<i>Pre-condition</i>	Sistem <i>running</i> dan <i>accessible</i>
	<i>Test Case</i>	Klik pada Menu Tentang
	<i>Expected Result</i>	Sistem dapat menampilkan halaman Tentang
	<i>Actual Result</i>	
	<i>Status</i>	
	<i>Test Result</i>	

Tabel 3. *Test Scenario* Halaman Utama

1.	<i>Module/Function</i>	Tombol <i>Choose File</i>
	<i>Assumption</i>	Pengguna menekan tombol <i>Choose File</i>
	<i>Pre-condition</i>	Sistem <i>running</i> dan <i>accessible</i>
	<i>Test Case</i>	Memasukkan file dokumen berekstensi *.docx.
	<i>Expected Result</i>	Sistem dapat memasukkan dokumen berekstensi *.docx
	<i>Actual Result</i>	

	<i>Status</i>	
	<i>Test Result</i>	
2.	<i>Module/Function</i>	Tombol Proses
	<i>Assumption</i>	Pengguna telah menekan tombol Proses
	<i>Pre-condition</i>	Dokumen 1 dan dokumen 2 sudah dimasukkan
	<i>Test Case</i>	Klik pada <i>button</i> Proses
	<i>Expected Result</i>	Sistem dapat menampilkan Halaman Klasifikasi
	<i>Actual Result</i>	
	<i>Status</i>	
	<i>Test Result</i>	

Tabel 4. *Test Scenario* Halaman Klasifikasi

1.	<i>Module/Function</i>	Tombol Selesai
	<i>Assumption</i>	Pengguna menekan tombol Selesai
	<i>Pre-condition</i>	Sistem menampilkan halaman hasil klasifikasi kata
	<i>Test Case</i>	Klik pada <i>button</i> Selesai
	<i>Expected Result</i>	Sistem dapat kembali ke Halaman Utama
	<i>Actual Result</i>	
	<i>Status</i>	
	<i>Test Result</i>	
2.	<i>Module/Function</i>	Tombol Unduh Detail

	<i>Assumption</i>	Pengguna menekan tombol Unduh Detail
	<i>Pre-condition</i>	Sistem menampilkan halaman hasil klasifikasi kata
	<i>Test Case</i>	Klik pada <i>button</i> Unduh Detail
	<i>Expected Result</i>	Sistem dapat mengunduh dokumen hasil klasifikasi berekstensi <i>docx</i> ke perangkat pengguna
	<i>Actual Result</i>	
	<i>Status</i>	
	<i>Test Result</i>	
3.	<i>Module/Function</i>	Tombol Lihat Detail
	<i>Assumption</i>	Pengguna menekan tombol Lihat Detail
	<i>Pre-condition</i>	Sistem menampilkan Halaman Klasifikasi
	<i>Test Case</i>	Klik pada <i>button</i> Lihat Detail
	<i>Expected Result</i>	Sistem dapat menampilkan halaman Lihat Detail
	<i>Actual Result</i>	
	<i>Status</i>	
	<i>Test Result</i>	

Tabel 5. *Test Scenario* Halaman Lihat Detail

1.	<i>Module/Function</i>	Tombol Kembali
	<i>Assumption</i>	Pengguna menekan tombol Lihat Detail
	<i>Pre-condition</i>	Sistem menampilkan halaman Lihat Detail
	<i>Test Case</i>	Klik tombol Kembali

<i>Expected Result</i>	Sistem dapat kembali ke Halaman Klasifikasi
<i>Actual Result</i>	
<i>Status</i>	
<i>Test Result</i>	

Tabel 6. *Test Scenario* Halaman Kamus Kata Dasar

1	<i>Module/Function</i>	Tombol Kembali
	<i>Assumption</i>	Sistem sudah diakses
	<i>Pre-condition</i>	Sistem <i>running</i> dan <i>accessible</i>
	<i>Test Case</i>	Klik tombol Kembali
	<i>Expected Result</i>	Sistem dapat kembali ke Halaman Utama
	<i>Actual Result</i>	
	<i>Status</i>	
	<i>Test Result</i>	

Tabel 7. *Test Scenario* Halaman Kamus Kata Asing

1.	<i>Module/Function</i>	Tombol Kembali
	<i>Assumption</i>	Sistem sudah diakses
	<i>Pre-condition</i>	Sistem <i>running</i> dan <i>accessible</i>
	<i>Test Case</i>	Klik tombol Kembali
	<i>Expected Result</i>	Sistem dapat kembali ke Halaman Utama
	<i>Actual Result</i>	

<i>Status</i>
<i>Test Result</i>

Tabel 8. *Test Scenario* Halaman Kamus Nama

1.	<i>Module/Function</i>	Tombol Kembali
	<i>Assumption</i>	Sistem sudah diakses
	<i>Pre-condition</i>	Sistem <i>running</i> dan <i>accessible</i>
	<i>Test Case</i>	Klik tombol Kembali
	<i>Expected Result</i>	Sistem dapat kembali ke halaman Utama
	<i>Actual Result</i>	
	<i>Status</i>	
	<i>Test Result</i>	

Tabel 9. *Test Scenario* Halaman Cara Pakai

1.	<i>Module/Function</i>	Tombol Kembali
	<i>Assumption</i>	Sistem sudah diakses
	<i>Pre-condition</i>	Sistem <i>running</i> dan <i>accessible</i>
	<i>Test Case</i>	Klik tombol Kembali
	<i>Expected Result</i>	Sistem dapat kembali ke halaman Beranda
	<i>Actual Result</i>	
	<i>Status</i>	
	<i>Test Result</i>	

Tabel 10. *Test Scenario* Halaman Tentang

1.	<i>Module/Function</i>	Tombol Kembali
	<i>Assumption</i>	Sistem sudah diakses
	<i>Pre-condition</i>	Sistem <i>running</i> dan <i>accessible</i>
	<i>Test Case</i>	Klik tombol Kembali
	<i>Expected Result</i>	Sistem dapat kembali ke Halaman Utama
	<i>Actual Result</i>	
	<i>Status</i>	
	<i>Test Result</i>	

V. SIMPULAN DAN SARAN

5.1 Simpulan

Berdasarkan hasil penelitian yang dilakukan penulis pada pengembangan sistem klasifikasi kata pada dokumen bertipe word (.docx) diperoleh beberapa kesimpulan sebagai berikut:

1. Pembuatan sistem klasifikasi kata berdasarkan kamus pada dokumen bertipe *word (.docx)* berhasil dibangun.
2. Sistem dapat menghasilkan klasifikasi kata berdasarkan kamus pada dua dokumen *word* bertipe *.docx.
3. Berdasarkan pengujian 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 persentase rata-rata 88.35%.

5.2 Saran

Penelitian ini masih dapat dikembangkan lebih lanjut lagi. Adapun saran untuk mengembangkan penelitian ini adalah sebagai berikut:

1. Sistem dapat dikembangkan menjadi sistem yang dapat mendeteksi banyak dokumen.
2. Perlu adanya pengembangan sistem yang lebih kompleks misalnya dengan menambahkan fitur *similarity* yang dapat digunakan untuk mendeteksi tingkat plagiarisme pada dokumen.
3. Sistem dapat memproses dokumen dengan ekstensi lain seperti *Portable Document Format (*.pdf)*.

DAFTAR PUSTAKA

- Ademariana, K., Aristoteles, A., Lumbanraja, F. R., & Rico Andrian, R. A. (2021). Clustering K-Means Jenis Kata Pada Laporan Kegiatan Kuliah Kerja Nyata (KKN) Universitas Lampung Menggunakan Word2vec. *Jurnal Pepadun Vol 2 No. 2*, 221-228.
- Afdhal, Muchallil, & Ardeman, S. (2017). Kinerja Server Basis Data Pada Aplikasi Web Berbasis Raspberry Pi. *Jurnal Online Teknik Elektro, II(3)*, 58-63.
- Budiman, U., & Izzatul. (2017). Sistem Pengolahan Kalimat Bahasa Indonesia Menjadi Kata Dasar dengan Metode Stemmer Berbasis Pemrograman Hypertext Preprocessor. *Journal Trisula LP2M Undar Volume 1*, 501-507.
- Fatkhum, A. (2012). Implementasi Search Engine Mesin Pencarian Menggunakan Metode Vector Space Model. *Jurnal Dinamika Teknik, V(1)*, 1-8.
- Firman, d. (2016). Sistem Informasi Perpustakaan Online Berbasis Web. *Journal Teknik Elektro dan Komputer, V(2)*, 29-36.
- Hidayat, T., & Mutaqqin, M. (2020). Pengujian Sistem Informasi Pendaftaran dan Pembayaran Wisuda Online menggunakan Black-box Testing dengan Metode Equivalence Partitioning dan Boundary Value Analysis. *Jutis (Jurnal Teknik Informatika), VI(1)*, 25-29.
- Indriyani, F., & Sihite, K. R. (2015). Pengenalan Huruf, Angka dan Warna Pada Anak Usia Dini melalui Pembelajaran Berbasis Multimedia. *Jurnal Paradigma*.
- Ipmawati, J., Kusriani, & Luthfi, E. T. (2017). Komparasi Teknik Klasifikasi Teks Mining Pada Analisis Sentimen. *Indonesian Journal on Networking and Security, Vol 6*.
- Kadir, A. (2008). *Dasar Pemrograman WEB Dinamis Menggunakan PHP* (III ed.). Yogyakarta: Andi.
- Kbbi. (2016). *Kamus Besar Bahasa Indonesia (KBBI)*. Jakarta: Kementerian Pendidikan Dan Budaya.

- Maghfira, T. N., Cholissidin, I., & Widodo, A. W. (2017, Juni). Deteksi Kesalahan Ejaan dan Penentuan Rekomendasi Koreksi Kata yang Tepat Pada Dokumen Jurnal JTIK Menggunakan Dictionary Lookup dan Damerau-Levenshtein Distance. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, VI, 498-506.
- Mahendra, D. G., & Kasyaful, A. (2017). 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*, IV(1), 69-74.
- Maniah, D. H. (2017). *Analisis dan Perancangan Sistem Informasi*. Yogyakarta: Deepublish.
- Nurpianti, A. Irfan, M. K., & Taupik, O. (2013). Pembuatan Aplikasi Anbiya Pedia Ensiklopedi Muslim Anak Berbasis Web. *Jurnal Teknik Informatika Fakultas Sains dan Teknologi*, VII(1), 2-33.
- Rahmadi, H., Mustaqbal, M. S., & Firdaus, R. F. (2015). Pengujian Aplikasi Menggunakan Black-Box Testing Boundary Value Analysis (Studi Kasus: Aplikasi Prediksi Kelulusan SNMPTN). *Jurnal Ilmiah Teknologi Informasi Terapan*, I(3), 31-36.
- Rusdiana, M. M., & Ifran, M. (2014). *Sistem Informasi Manajemen*. Bandung: Pustaka Setia.
- Sommerville, I. (2007). Software Engineering 9th edition. *Addison Wesley: Publisher*, pp. 29-32.
- Susanto, A. (2013). Dalam *Sistem Informasi Akuntansi* (hal. 23). Bandung: Lingga Jaya.
- Utami, S. R. (2017). Pembelajaran Aspek Tata Bahasa dalam Buku Pelajaran Bahasa Indonesia. *AKSIS: Jurnal Pendidikan Bahasa dan Sastra Indonesia*, Vol 1 No.2.
- Warman, I., & Ramdaniansyah, R. (2018). Analisis Perbandingan Kinerja Query Database Management System (DBMS) Antara MySQL 5.7.16 dan MariaDB 10.1. *Jurnal TEKNOIF Teknik Informatika Institut Teknologi Padang*, VI(1), 32-41.

Warsito, A. B., Yusup, M., & Makaram, M. I. (2015). Perancangan SIS+ Menggunakan Metode YII Framework. *CCIT Journal*, 8(2), 24-33.