

**ANALISIS PERBANDINGAN ALGORITMA NAZIEF & ADRIANI
DENGAN ALGORITMA IDRIS UNTUK PENCARIAN KATA DASAR
PADA TEKS BAHASA INDONESIA BERBASIS *WEBSITE***

Oleh :

Adhi Prasadhatama

32140102



Program Studi Teknik Informatika

Fakultas Teknologi dan Desain

Universitas Bunda Mulia

Jakarta

2017

DAFTAR PUSTAKA

DAFTAR PUSTAKA	i
DAFTAR GAMBAR	iii
DAFTAR TABEL.....	v
BAB 1	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan dan Manfaat	2
1.3.1 Tujuan Penelitian	2
1.3.2 Manfaat Penelitian	2
1.4 Ruang Lingkup.....	2
BAB 2	4
2.1 Information Retrieval (IR)	4
2.2 Text Preprocessing.....	4
2.3 Algoritma Stemming Nazief-Adriani.....	4
2.4 Algoritma Stemming Idris	5
2.5 MySQL	6
2.6 Entity Relationship Diagram.....	7
2.7 Flowchart	7
2.8 Use Case Diagram.....	10
2.9 Sequence Diagram	12
2.10 Activity Diagram	14
2.11 Class Diagram.....	15
2.12 PHP	17
2.13 Waterfall	17
BAB 3	20
3.1 Analisis Kebutuhan Fungsional dan Kebutuhan Non Fungsional	20
3.1.1 Analisis Kebutuhan Fungsional	20
3.1.2 Analisis Kebutuhan Non Fungsional	20
3.2 Pemilihan Metode	22
3.2.1 Model Perancangan Sistem.....	22
3.2.2 Algoritma Nazief & Adriani	23
3.2.3 Algoritma Idris.....	24

3.3 Perancangan Basis Data	26
3.3.1 Entity Relationship Diagram (ERD)	26
3.4 Perancangan Proses.....	28
3.4.1 Flow Chart	28
3.4.1.1 Flow Chart Pengujian	29
3.4.2 Perancangan Proses Diagram UML	31
3.4.2.1 Use Case Diagram.....	31
3.4.2.2 Sequence Diagram	32
3.4.2.3 Activity Diagram	34
3.4.2.4 Class Diagram.....	36
3.5 Perancangan Tampilan.....	37
3.6 JADWAL PENELITIAN.....	43
BAB 4	44
4.1 Implementasi Antarmuka Pengguna	44
4.1.1 Tampilan <i>Login</i>	44
4.1.2 Tampilan <i>Dashboard</i>	45
4.1.3 Tampilan Pengujian	45
4.1.4 Tampilan Hasil Pengujian	46
4.1.5 Tampilan Rekap	47
4.1.6 Tampilan Detail Rekap	48
4.2 Implementasi Proses dan Algoritma	49
4.2.1 Implementasi Proses	49
4.2.2 Implementasi Algoritma	53
4.3 Pengujian Keberhasilan Metode	61
4.4 Kesimpulan	81
BAB 5	83
5.1 Kesimpulan	83
5.2 Saran	83
DAFTAR PUSTAKA	84

DAFTAR GAMBAR

Gambar 2.1 Gambar Tahapan Waterfall	18
Gambar 3.1 Gambar Entity Relationship Diagram (ERD) pengujian	26
Gambar 3.2 Gambar Flowchart pengujian	29
Gambar 3.3 Gambar Use case diagram pengujian	31
Gambar 3.4 Gambar sequence diagram pengujian.....	32
Gambar 3.5 Gambar Activity diagram pengujian.....	34
Gambar 3.6 Gambar Class diagram pengujian	36
Gambar 3.7 Gambar Form login.....	37
Gambar 3.8 Gambar Form dashboard.....	38
Gambar 3.9 Gambar Form pengujian	39
Gambar 3.10 Gambar Form hasil pengujian	40
Gambar 3.11 Gambar Form rekap	41
Gambar 3.12 Gambar Form detail rekap.....	42
Gambar 4.1 Gambar Tampilan login.....	44
Gambar 4.2 Gambar Tampilan dashboard.....	45
Gambar 4.3 Gambar Tampilan pengujian	45
Gambar 4.4 Gambar Tampilan hasil pengujian	46
Gambar 4.5 Gambar Tampilan rekap	47
Gambar 4.6 Gambar Tampilan detail rekap.....	48
Gambar 4.7 Gambar Code text preprocessing.....	50
Gambar 4.8 Gambar Code perhitungan waktu stemming.....	51
Gambar 4.9 Gambar Code menyimpan ke database.....	52
Gambar 4.10 Gambar Code Nazief & Adriani cek database.....	53
Gambar 4.11 Gambar Code Nazief & Adriani langkah-langkah stemming	53
Gambar 4.12 Gambar Code Nazief & Adriani infleksi akhiran	54
Gambar 4.13 Gambar Code Nazief & Adriani turunan akhiran	54
Gambar 4.14 Gambar Code Nazief & Adriani Imbuhan tidak diizinkan	55
Gambar 4.15 Gambar Code Nazief & Adriani turunan awalan	56
Gambar 4.16 Gambar Code Idris cek database.....	57
Gambar 4.17 Gambar Code Idris langkah-langkah stemming	58
Gambar 4.18 Gambar Code Idris imbuhan tidak diizinkan.....	58
Gambar 4.19 Gambar Code Idris turunan awalan.....	59
Gambar 4.20 Gambar Code Idris infleksi akhiran.....	60
Gambar 4.21 Gambar Code Idris turunan akhiran	60
Gambar 4.22 Gambar Pengujian 1 masukkan teks pengujian.....	62
Gambar 4.23 Gambar Pengujian 1 informasi umum pengujian.....	63
Gambar 4.24 Gambar Pengujian 1 informasi Nazief & Adriani.....	63
Gambar 4.25 Gambar Pengujian 1 informasi Idris	64
Gambar 4.26 Gambar Pengujian 1 rekap pengujian	64
Gambar 4.27 Gambar Pengujian 2 masukkan teks pengujian.....	66
Gambar 4.28 Gambar Pengujian 2 informasi umum pengujian.....	67
Gambar 4.29 Gambar Pengujian 2 informasi Nazief & Adriani.....	67
Gambar 4.30 Gambar Pengujian 2 informasi Idris.....	68

Gambar 4.31 Gambar Pengujian 2 rekap pengujian	68
Gambar 4.32 Gambar Pengujian 3 masukkan teks pengujian.....	70
Gambar 4.33 Gambar Pengujian 3 informasi umum pengujian.....	71
Gambar 4.34 Gambar Pengujian 3 informasi Nazief & Adriani.....	71
Gambar 4.35 Gambar Pengujian 3 informasi Idris	72
Gambar 4.36 Gambar Pengujian 3 rekap pengujian	72
Gambar 4.37 Gambar Pengujian 4 masukkan teks pengujian.....	74
Gambar 4.38 Gambar Pengujian 4 informasi umum pengujian.....	75
Gambar 4.39 Gambar Pengujian 4 informasi Nazief & Adriani.....	75
Gambar 4.40 Gambar Pengujian 4 informasi Idris	76
Gambar 4.41 Gambar Pengujian 4 rekap pengujian	76
Gambar 4.42 Gambar Pengujian 5 masukkan teks pengujian.....	78
Gambar 4.43 Gambar Pengujian 5 informasi umum pengujian.....	79
Gambar 4.44 Gambar Pengujian 5 informasi Nazief & Adriani.....	79
Gambar 4.45 Gambar Pengujian 5 informasi Idris	80
Gambar 4.46 Gambar Pengujian 5 rekap pengujian	80
Gambar 4.47 Gambar Kesimpulan pengujian	81

DAFTAR TABEL

Tabel 2.1 Tabel Kardinal crow's foot.....	7
Tabel 2.2 Tabel Simbol flowchart.....	8
Tabel 2.3 Tabel Simbol use case diagram	10
Tabel 2.4 Tabel Simbol sequence diagram	13
Tabel 2.5 Tabel Simbol activity diagram	14
Tabel 2.6 Tabel Simbol class diagram.....	16
Tabel 3.1 Tabel Jadwal penelitian	43

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Komunikasi merupakan suatu proses dimana seseorang atau beberapa orang menciptakan dan menggunakan informasi agar terhubung dengan lingkungan dan orang lain. Komunikasi pada umumnya dilakukan secara lisan atau verbal yang dapat dimengerti oleh kedua belah pihak. Dengan adanya teknologi kita dapat berkomunikasi dengan satu dan lainnya dengan mudah. Namun tanpa disadari bahwa berkomunikasi langsung dengan sebuah program masih dianggap sulit. Dikarenakan oleh perbedaan bahasa maupun tatanan bahasanya itu sendiri. Sehingga program tidak dapat dengan mudah mengerti dari apa yang dimaksudkan oleh manusia.

Untuk mengerti apa yang dimaksud oleh manusia sebuah program harus memiliki beberapa kemampuan seperti teks *preprocessing*, ekstraksi fitur, pencarian kata dasar, hingga pelatihan atau pembelajaran untuk mengembangkan pola pikirnya. *Stemming* merupakan suatu proses untuk menemukan kata dasar dari sebuah kata dengan menghilangkan semua imbuhan (*affixes*) baik yang terdiri dari awalan (*prefixes*), sisipan (*infixes*), akhiran (*suffixes*) dan kombinasi dari awalan dan akhiran (*confixes*) pada kata turunan. *Stemming* adalah inti dari teknik pemrosesan *natural language* untuk mendapatkan informasi kembali (*Information Retrieval*) yang efektif dan efisien dan secara luas dapat diterima oleh pengguna (*users*). *Stemming* digunakan untuk mengganti bentuk dari suatu kata menjadi kata dasar dari kata tersebut yang sesuai dengan struktur morfologi bahasa Indonesia yang baik dan benar. Pada pencarian kata dasar (*stemming*) khususnya bahasa Indonesia terdapat beberapa algoritma seperti Jelita Asian tahun 2005, Arifin & Setiono tahun 2002, Nazief & Adriani tahun 1996, Ahmad Yusoff Sembok tahun 1996, Vega tahun 2001, Idris tahun 2001 dan *Enhanced Confix Stripping Stemmer* tahun 2008.

Mengetahui ada berbagai macam algoritma untuk *stemming* bahasa Indonesia. Penulis memutuskan untuk mempersempit ruang penelitian ke algoritma Nazief & Adriani dan algoritma Idris. Dikarenakan algoritma Nazief & Adriani merupakan algoritma *stemming* yang populer untuk memproses teks berbahasa Indonesia, sedangkan algoritma Idris sendiri kurang populer untuk *stemming* bahasa Indonesia. Maka dari itu penelitian ini berjudul “Analisis Perbandingan Algoritma Nazief & Adriani dengan Algoritma Idris untuk Pencarian Kata Dasar pada Teks Bahasa Indonesia Berbasis *Website*”. Pada penelitian ini akan membahas mengenai cara kerja, kelebihan & kekurangan, hingga hasil dari tiap algoritma yang akan dibandingkan.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas maka dapat dirumuskan masalah sebagai berikut: Bagaimanakah membandingkan algoritma Nazief & Adriani dengan algoritma Idris dalam pencarian kata dasar pada teks bahasa Indonesia ?

1.3 Tujuan dan Manfaat

1.3.1 Tujuan Penelitian

1. Mengetahui algoritma mana yang dapat mencari kata dasar dengan proses lebih cepat antara algoritma Nazief & Adriani dengan algoritma Idris.
2. Mengetahui algoritma mana yang dapat mencari kata dasar dengan tingkat keakuratan tinggi antara algoritma Nazief & Adriani dengan algoritma Idris.

1.3.2 Manfaat Penelitian

1. Mengetahui kelebihan dan kekurangan dari masing-masing algoritma.
2. Dapat menjadi acuan dalam pemilihan algoritma untuk *menstemming* sebuah teks maupun dokumen yang berbahasa Indonesia.

1.4 Ruang Lingkup

Untuk membatasi penelitian ini agar tidak terlalu luas maka ruang lingkup penelitian ini adalah sebagai berikut :

1. File yang akan diproses hanya berupa teks, file gambar, audio, maupun video yang terlampir pada masukkan tidak akan diproses dan tidak dimunculkan.
2. Teks yang akan diolah berupa teks berbahasa Indonesia.
3. Parameter *output* berdasarkan kata dasar dari Kamus Besar Bahasa Indonesia (KBBI).
4. Sistem ini berbasis *website* berbahasa pemrograman PHP.
5. *Database* yang akan digunakan adalah MySQL

BAB 2

LANDASAN TEORI

2.1 Information Retrieval (IR)

Information retrievals System atau sistem temu balik informasi merupakan bagian dari *computer science* tentang pengambilan informasi dari dokumen-dokumen yang didasarkan pada isi dan konteks dari dokumen-dokumen itu sendiri (Hasanah, 2017). Informasi yang diambil dapat berupa teks, gambar, audio, dan video yang berguna untuk melakukan penyimpanan, pencarian, dan pemeliharaan informasi.

2.2 Text Preprocessing

Text processing berfungsi mengubah data tekstual yang tidak terstruktur ke dalam data terstruktur dan disimpan dalam basis data. Tahap *preprocessing* terdiri dari beberapa langkah yaitu : *case floding*, *tokenization*, *filtering* dan *stemming* (Rahmawati, 2016).

Pada proses *case floding* karakter selain huruf dihilangkan dan semua huruf diubah menjadi *lowercase*. *Tokenization* adalah proses dimana sebuah kalimat dipotong untuk menghasilkan kata-kata yang akan digunakan untuk proses selanjutnya. Pengambilan kata-kata penting dari hasil *tokenization* merupakan proses yang dilakukan dalam *filtering*. Proses *filtering* memiliki dua teknik yaitu *stop list* (membuang kata yang kurang penting) dan *word list* (menyimpan kata yang penting). Setelah mendapatkan kata-kata yang dirasa penting, selanjutnya dilakukan proses *stemming*. Proses pada tahap *stemming* ialah melakukan pengolahan dan pencarian kata dasar dari hasil *filtering* atau sekumpulan kata-kata yang sudah dianggap penting.

2.3 Algoritma Stemming Nazief-Adriani

Menurut Nugroho pada tahun 2017, Algoritma yang dibuat oleh Bobby Nazief dan Mirna Adriani memiliki tahapan sebagai berikut:

1. Cari kata yang akan *distem* dalam kamus. Jika ditemukan maka diasumsikan bahwa kata tersebut adalah kata dasar. Maka algoritma berhenti.
2. Infleksi akhiran (“-lah”, “-kah”, “-ku”, “-mu”, atau “-nya”) dibuang. Jika berupa partikel (“-lah”, “-kah”, “-tah” atau “-pun”) maka langkah ini diulangi lagi untuk menghapus kata ganti posesif (“-ku”, “-mu”, atau “-nya”), jika ada.
3. Hapus penurunan akhiran (“-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 penurunan awalan. Jika pada langkah 3 ada akhiran yang dihapus maka pergi ke langkah 4a, jika tidak pergi ke langkah 4b.
 - a. Periksa tabel kombinasi awalan akhiran yang tidak diizinkan. Jika ditemukan maka algoritma berhenti, jika tidak pergi ke langkah 4b.
 - b. Pada langkah ini dilakukan perulangan sebanyak tiga kali. Tentukan tipe awalan kemudian hapus awalan. Jika kata dasar 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 kata dasar lalu proses diakhiri.

2.4 Algoritma *Stemming* Idris

Menurut Permatasari pada tahun 2016, dijabarkan mengenai *stemming* menggunakan algoritma Idris adalah sebagai berikut :

1. a. Kata yang belum di-*stemming* dicari pada kamus umum atau Kamus Besar Bahasa Indonesia (KBBI). Jika kata itu langsung ditemukan, berarti

kata tersebut adalah kata dasar. Kata tersebut dikembalikan dan algoritma dihentikan.

b. Kata yang belum di-*stemming* dicari pada kamus lokal. Jika kata itu langsung ditemukan, berarti kata tersebut adalah kata dasar. Kata tersebut dikembalikan dan algoritma dihentikan. Kamus lokal digunakan karena algoritma Idris dibuat untuk *stemming* bahasa melayu, yang dimana kamus lokal berisi kata-kata dasar bahasa melayu. Sedangkan *stemming* bahasa Indonesia tidak memerlukan kamus lokal.

2. Menghilangkan turunan awalan. Langkah ini terus dilakukan sampai tidak ada lagi turunan awalan. Jika tidak ada lagi, maka lanjutkan ke langkah berikutnya. Untuk beberapa kombinasi imbuhan, dilakukan penghilangan awalan terlebih dahulu. Yaitu pada kombinasi imbuhan “ber-lah”, “ber-an”, “men-i”, “di-i”, “pe-i”, “ter-i”.
3. Hilangkan ifleksi akhiran terlebih dahulu. Jika hal ini berhasil dan akhiran adalah partikel (“-lah”, ”-pun” dan ”-kah”), langkah ini dilakukan lagi untuk menghilangkan kata ganti akhiran posesif (“ku”, “mu” atau ”nya”).
4. Hilangkan turunan akhiran. Langkah ini terus dilakukan sampai tidak ada lagi penurunan akhiran. Jika tidak ada lagi, maka lanjutkan ke langkah berikutnya.
5. Setelah setiap penghilangan imbuhan dilakukan, maka lakukan pengecekan menggunakan kamus. Jika kata ditemukan, maka algoritma berhenti dan tidak perlu dilakukan pengecekan terhadap imbuhan lainnya. Jika sampai selesai penghilangan imbuhan masih belum menemukan kata dasar, maka dilakukan *recoding*.
6. Jika semua langkah sudah dilakukan termasuk *recoding* dan tidak juga ditemukan dalam kamus, maka algoritma ini akan menganggap kata semula sebagai kata dasar.

2.5 MySQL

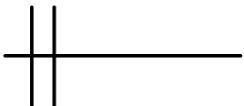
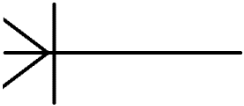
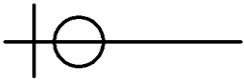
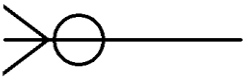
Menurut Setyawan pada 2012, MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL atau yang dikenal dengan DBMS (*Database Management System*), *database* ini *multithread*, multi pengguna. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU

General Public Licence (GPL), tetapi mereka juga menjual di bawah lisensi komersial untuk kasus-kasus yang bersifat khusus.

2.6 Entity Relationship Diagram

Menurut Sari pada tahun 2017, *Entity Relationship Diagram* adalah diagram yang berguna untuk memodelkan sistem yang nantinya akan dikembangkan dalam basis datanya. *Entity Relationship Diagram* tersusun atas tiga komponen, yaitu entitas, atribut dan relasi. Secara garis besar, entitas merupakan objek dasar yang terlihat dalam sistem. Atribut berperan sebagai penjelas entitas, dan relasi menunjukkan hubungan yang terjadi diantara dua entitas. *Entity Relationship Diagram* memiliki 2 model notasi yaitu, Chen dan *crow's foot*. Penjelasan Famsila pada 2016 mengenai kardinalitas dari model *crow's foot* adalah sebagai berikut:

Tabel 2.1 Tabel Kardinal *crow's foot*

No	Simbol	Keterangan
1.		Satu dan hanya satu
2.		Satu atau banyak
3.		Kosong atau satu
4.		Kosong atau banyak

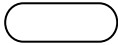
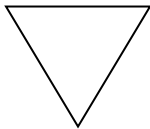
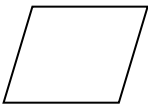
(Sumber: Famsila, 2016)



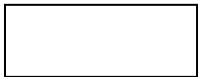

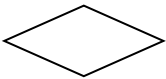

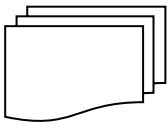
2.7 Flowchart

Flowchart adalah metode untuk menggambarkan tahap-tahap penyelesaian masalah beserta aliran data menggunakan simbol-simbol yang mudah untuk

dipahami (Sihite, 2017). *Flowchart* merupakan hal penting untuk menyederhanakan rangkaian proses yang berguna untuk memudahkan pengguna dalam memahami informasi tersebut.

Tabel 2.2 Tabel Simbol *flowchart*

No	Simbol	Pengertian	Keterangan
1.		Mulai / berakhir	Digunakan untuk memulai, mengakhiri, atau titik henti dalam sebuah proses atau program; juga digunakan untuk menunjukkan pihak eksternal.
2.		Dokumen	Sebuah dokumen atau laporan; dokumen dapat dibuat dengan tangan atau dicetak oleh komputer.
3.		Kegiatan Manual	Sebuah kegiatan pemrosesan yang dilaksanakan secara manual.
4.		Arsip	Arsip dokumen disimpan dan diambil secara manual. Huruf didalamnya menunjukkan cara pengurutan arsip: N = Urut Nomor; A = Urut Abjad; T = Urut Tanggal.
5.		<i>Input / Output</i> ; Jurnal / Buku Besar	Digunakan untuk menggambarkan berbagai media <i>input</i> dan <i>output</i> dalam sebuah bagan alir program.
6.		<i>Disk</i> Bermagnet	Data disimpan secara permanen pada disk bermagnet.

No	Simbol	Pengertian	Keterangan
7.		Penghubung Pada Halaman Berbeda	Menghubungkan bagan alir yang berada di halaman yang berbeda.
8.		Pemasukan Data <i>On Line</i>	Entri data alat oleh <i>on line</i> seperti terminal CRT dan komputer pribadi.
9.		Pemrosesan Komputer	Sebuah fungsi pemrosesan yang dilaksanakan oleh komputer biasanya menghasilkan perubahan terhadap data atau informasi
10.		Arus Dokumen atau Pemrosesan	Arus dokumen atau pemrosesan; arus normal adalah ke kanan atau ke bawah.
11.		Keputusan	Sebuah tahap pembuatan keputusan
12.		Penghubung Dalam Sebuah Halaman	Menghubungkan bagan alir yang berada pada halaman yang sama.
13.		Dokumen Rangkap	Digambarkan dengan menumpuk simbol dokumen dan pencetakan nomor dokumen dibagian depan dokumen pada bagian kiri atas.

(Sumber: Putri, 2013)

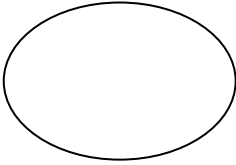
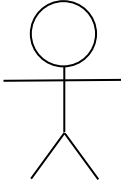
2.8 Use Case Diagram


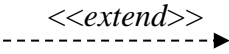
Menurut Wijaya pada tahun 2015 *Use case* diagram adalah gambaran dari beberapa atau seluruh aktor dan *use case* dengan tujuan mengenali interaksi mereka dalam suatu sistem. *Use case* diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem, yang ditentukan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem.

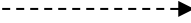
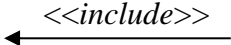
Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case* yaitu:

- a. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
- b. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Tabel 2.3 Tabel Simbol *use case* diagram

No	Nama	Simbol	Keterangan
1.	<i>Use Case</i>		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal di frasa nama <i>Use Case</i> .
2.	Aktor		Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat

No	Nama	Simbol	Keterangan
			itu sendiri. Aktor hanya memberikan informasi ke sistem, aktor hanya menerima informasi dari sistem, aktor memberikan dan menerima informasi ke sistem dan dari sistem.
3.	Asosiasi		Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor. Asosiasi merupakan hubungan statis antar elemen yang menggambarkan elemen yang memiliki atribut berupa elemen lain, atau elemen yang harus mengetahui eksistensi elemen lain.
4.	Ekstensi		Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walaupun tanpa <i>use case</i> tambahan itu, mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek.

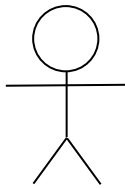

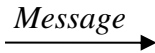

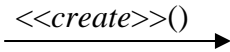
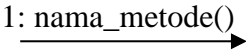
No	Nama	Simbol	Keterangan
5.	Generalisasi		Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari yang lainnya.
6.	<i>Include</i>		Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat.

(Sumber: Wijaya, 2015)

2.9 Sequence Diagram

Menurut Sugiarti pada 2013, *Sequence* diagram menggambarkan interaksi antar objek di dalam dan di sekitar termasuk pengguna, tampilan, dan sebagainya berupa *message* yang digambarkan terhadap waktu. *Sequence* diagram terdiri antara dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence* diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah *event* untuk menghasilkan *output* tertentu. *Sequence* diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirim dan diterima objek. Oleh karena itu, untuk menggambarkan *sequence* diagram harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta dengan metode-metode yang dimiliki kelas yang di instansiasi menjadi objek itu. Membuat *sequence* diagram juga dibutuhkan untuk melihat skenario yang ada.

Tabel 2.4 Tabel Simbol *sequence diagram*

No	Nama	Simbol	Keterangan
1.	Aktor		Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri. Aktor hanya memberikan informasi ke sistem, aktor hanya menerima informasi dari sistem, aktor memberikan dan menerima informasi ke sistem dan dari sistem.
2.	<i>Lifeline</i>		Menyatakan kehidupan suatu objek, untuk menggambarkan kelas dan objek.
3.	Objek		Menyatakan objek yang berinteraksi (pesan).
4.	Waktu aktif		Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif.
5.	Pesan tipe <i>create</i>		Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
6.	Pesan tipe <i>call</i>		Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau


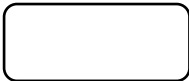
No	Nama	Simbol	Keterangan
			dirinya sendiri, sesuai dengan kelas objek yang berinteraksi.
7.	Pesan tipe <i>return</i>	1: keluaran ----->	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.

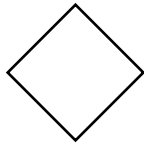


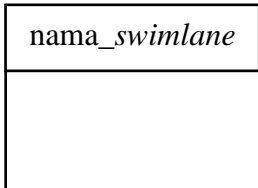
(Sumber: Sugiarti, 2013)

2.10 Activity Diagram

Activity diagram menggambarkan rangkaian alir aktivitas dalam sistem yang dirancang, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktivitas lainnya seperti *use case* atau interaksi (Wijaya, 2015). *Activity diagram* berupa *flow chart* yang digunakan untuk memperlihatkan alir kerja dari sistem.

Tabel 2.5 Tabel Simbol *activity diagram*

No	Nama	Simbol	Keterangan
1.	Status awal		Status awal aktivitas sistem, sebuah aktivitas memiliki sebuah status awal.
2.	Aktivitas		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.

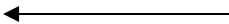
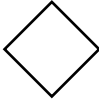
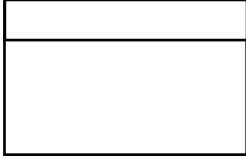
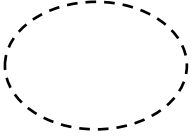
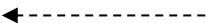
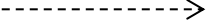
No	Nama	Simbol	Keterangan
3.	Percabangan atau <i>decision</i>		Menyatakan objek yang berinteraksi (pesan).
4.	Penggabungan atau <i>join</i>		Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.	Status akhir		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6.	<i>Swimlane</i>		Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

(Sumber: Wijaya, 2015)

2.11 Class Diagram

Diagram kelas atau *class* diagram menggambarkan struktur sistem yang didefinisikan dari segi kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Diagram kelas mendefinisikan jenis-jenis objek dalam sistem dan berbagai hubungan statis yang terdapat diantara objek (Sugiarti, 2013).

Tabel 2.6 Tabel Simbol *class diagram*

No	Nama	Simbol	Keterangan
1.	<i>Generalization</i>		Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2.	<i>Nary association</i>		Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3.	<i>Class</i>		Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4.	<i>Collaboration</i>		Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
5.	<i>Realization</i>		Operasi yang benar-benar dilakukan oleh suatu objek.
6.	<i>Dependency</i>		Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.

No	Nama	Simbol	Keterangan
7.	<i>Association</i>	_____	Menghubungkan antara objek satu dengan objek lainnya.

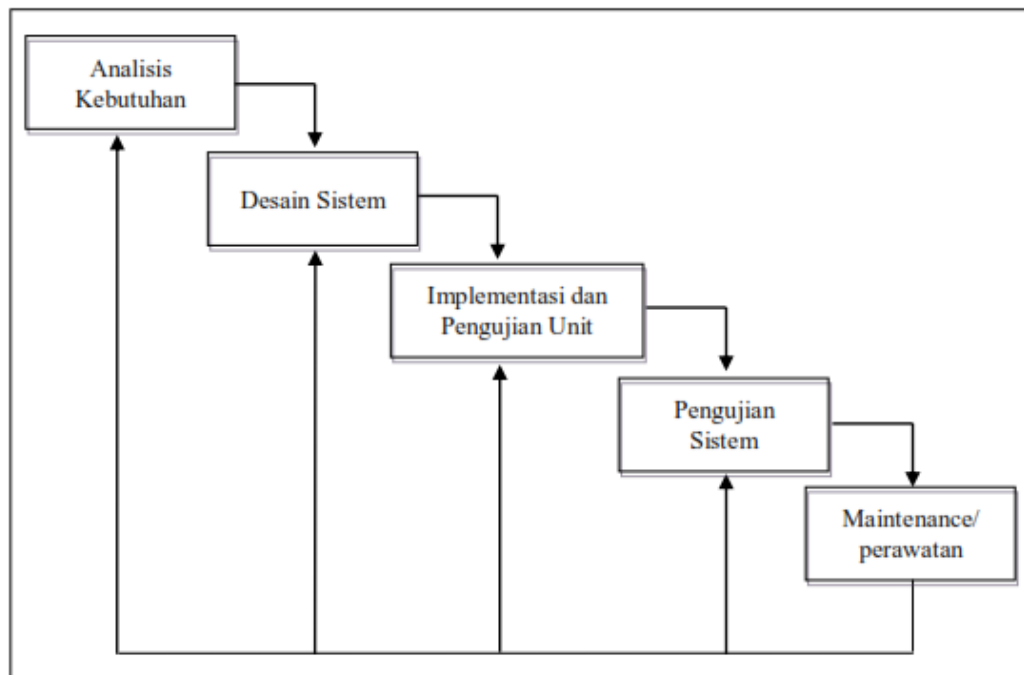
(Sumber: Sugiarti, 2013)

2.12 PHP

Menurut Kurniawan pada 2016, PHP adalah akronim rekursif untuk *Hypertext Preprocessor*. PHP merupakan bahasa yang didesain untuk pengembangan *web*. PHP ditemukan oleh Rasmus Lerdorf pada tahun 1995. Rereferensi implementasi dari PHP sekarang diproduksi oleh *The PHP Group*. PHP ini ditafsirkan oleh *web server* dengan modul *PHP processor* yang akan menghasilkan halaman *web*. Perintah PHP bisa langsung dimasukkan dalam dokumen HTML dari pada memanggil file eksternal untuk memproses data. PHP juga berevolusi dengan mempunyai kemampuan untuk memasukkan *command-line interface* dan bisa digunakan sebagai aplikasi grafik yang berdiri sendiri.

2.13 Waterfall

Menurut Oktarino pada 2015, *waterfall* adalah model pengembangan sistem yang setiap tahapnya harus diselesaikan terlebih dahulu secara penuh sebelum diteruskan ketahap berikutnya untuk menghindari terjadinya pengulangan tahapan.



Gambar 2.1 Gambar Tahapan Waterfall

Gambar diatas adalah tahapan umum dari model proses ini. Adapun penjelasan dari masing-masing tahapan adalah sebagai berikut.

1. Analisis Kebutuhan

Dalam tahapan ini penulis menentukan kebutuhan-kebutuhan pada sistem Analisa Perbandingan Algoritma Nazief & Adriani dengan Algoritma Idris untuk Pencarian Kata Dasar pada Teks Bahasa Indonesia Berbasis *Website* baik itu kebutuhan fungsional maupun kebutuhan non fungsional.

2. Desain Sistem

Dalam tahapan desain sistem ini, penulis membuat perancangan dari model atau desain sistem dengan menggunakan beberapa alat bantu untuk menggambarkan sistem berjalan ataupun sistem baru yang akan dikembangkan secara logika. Untuk rincian prosedur menggunakan *flowchart* sedangkan untuk menggambarkan susunan logis antar data dan hubungannya dengan sistem penulis menggunakan *Entity Relationship Diagram (ERD)*.

3. Implementasi dan Pengujian Unit

Pada tahap ini, penulis melakukan penerjemahan desain yang telah dibuat ke dalam bentuk *software* yang dirancang dengan bahasa pemrograman PHP dan basis data menggunakan MySQL. Selanjutnya melakukan pengujian terhadap

program yang dibangun per unit atau per modul kerja. Dimana semua fungsi-fungsi *software* tersebut diuji cobakan, agar *software* bebas dari *error* dan hasilnya harus benar-benar sesuai dengan kebutuhan yang sudah didefinisikan sebelumnya.

4. Pengujian Sistem

Pada tahap ini program yang telah dibuat dan diuji tiap unitnya kemudian disatukan menjadi suatu sistem dan diuji secara keseluruhan berguna untuk menguji tingkat integrasi antar unit yang dibuat sebelumnya.

5. *Maintenance* atau Perawatan

Pada tahap ini program akan diawasi kinerja dan hasil keluarannya. Jika terjadi sebuah kesalahan pada saat program beroperasi atau hasil keluaran tidak sesuai dengan yang diharapkan, maka akan dilakukan perawatan yang berguna untuk memperbaiki program.

BAB 3

ANALISIS DAN PERANCANGAN

3.1 Analisis Kebutuhan Fungsional dan Kebutuhan Non Fungsional

3.1.1 Analisis Kebutuhan Fungsional

1. *Usability*

Penggunaan aplikasi Analisis Perbandingan Algoritma Nazief & Adriani dengan Algoritma Idris untuk Pencarian Kata Dasar pada Teks Bahasa Indonesia Berbasis *Website* memiliki beberapa aspek kegunaan sebagai berikut:

- a. Aplikasi dapat membantu penelitian untuk menjabarkan kelebihan dan kekurangan dari masing-masing algoritma.
- b. Aplikasi ini dapat menjadi pembuktian untuk memudahkan pengambilan keputusan untuk pemilihan algoritma *stemming* mana yang memiliki proses hasil lebih cepat dan akurat dalam pengolahan teks bahasa Indonesia.

2. *Reliability*

Penggunaan aplikasi Analisis Perbandingan Algoritma Nazief & Adriani dengan Algoritma Idris untuk Pencarian Kata Dasar pada Teks Bahasa Indonesia Berbasis *Website* memiliki beberapa aspek kehandalan sebagai berikut:

- a. Aplikasi dapat menunjukkan perhitungan kecepatan proses dari masing-masing algoritma.
- b. Aplikasi dapat menunjukkan perhitungan keakuratan dari masing-masing algoritma.

3.1.2 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional dilakukan untuk menentukan spesifikasi yang dibutuhkan dalam menjalankan aplikasi Analisis Perbandingan Algoritma

Nazief & Adriani dengan Algoritma Idris untuk Pencarian Kata Dasar pada Teks Bahasa Indonesia dengan melibatkan spesifikasi dari sisi *Hardware*, *Software*, dan *Brainware*.

1. *Hardware*

Spesifikasi *Hardware* yang dibutuhkan dalam pengoperasian aplikasi Analisis Perbandingan Algoritma Nazief & Adriani dengan Algoritma Idris untuk Pencarian Kata Dasar pada Teks Bahasa Indonesia sebagai berikut :

- a. *Processor* Intel Pentium IV 2.4 Mhz PC.
- b. RAM minimal 128 MB.
- c. *Hard Disk* minimal 20 GB.
- d. Monitor 15”.
- e. *Keyboard* dan *mouse*.
- f. Internet minimal kecepatan 100 Kbps

2. *Software*

Prangkat lunak yang dibutuhkan untuk mengembangkan aplikasi adalah sebagai berikut:

- a. Sistem operasi *Windows* XP, Linux atau Unix yang dapat melakukan *text editor* terhadap file PHP.
- b. Bahasa pemrograman yang digunakan adalah PHP dengan menggunakan *text editor notepad*.
- c. Untuk simulator dapat menggunakan XAMPP sebagai *localhost server*.
- d. Untuk penyimpanan, *database* yang digunakan adalah MySQL.

Sedangkan untuk penggunaan aplikasi, perangkat lunak yang diperlukan adalah sebagai berikut:

- a. Sistem operasi *Windows* XP, Linux atau Unix dengan dukungan browser untuk menjalankan aplikasi berbasis *web* PHP.

3. *Brainware*

Aplikasi dirancang untuk dapat dengan mudah dipahami dan dijalankan oleh pengguna tanpa harus memiliki kemampuan khusus dalam bidang informatika atau komputer. Pengguna umum berusia minimal 17 tahun dapat menjalankan aplikasi ini tanpa banyak menemukan kesulitan.

3.2 Pemilihan Metode

3.2.1 Model Perancangan Sistem

Model perancangan sistem yang digunakan dalam membangun dan merancang aplikasi ini adalah model *waterfall*. Penggunaan model *waterfall* ini karena proses dilakukan secara bertahap, sehingga setiap proses tidak saling tumpang tindih dalam pelaksanaannya.

Tahapan-tahapan yang dilakukan pada perancangan sistem dengan model *waterfall* adalah sebagai berikut.

6. Analisis Kebutuhan

Untuk membangun sistem perbandingan algoritma Nazief & Adriani dengan algoritma Idris untuk pencarian kata dasar pada teks bahasa Indonesia, dilakukan penetapan fitur apa saja yang akan ada di dalam sistem sehingga sistem yang dibuat dapat sesuai dengan kebutuhan dan tujuan dapat tercapai. Spesifikasi sistem dan detail-detail sebelum sistem dibuat ditetapkan dengan rinci.

7. Desain Sistem

Setelah tujuan sistem dan fitur apa saja yang akan ada di dalamnya telah jelas, sistem perbandingan algoritma Nazief & Adriani dengan algoritma Idris untuk pencarian kata dasar pada teks bahasa Indonesia didesain sesuai dengan tetapan yang telah dibuat sehingga lebih mudah dalam pembuatan sistem.

8. Implementasi dan Pengujian Unit

Desain sistem perbandingan algoritma Nazief & Adriani dengan algoritma Idris untuk pencarian kata dasar pada teks bahasa Indonesia yang telah dibuat diimplementasikan ke dalam bentuk aplikasi dan dilakukan pengecekan fungsi yang dibuat secara berkala untuk menghindari kesalahan.

9. Pengujian Sistem

Setelah desain diimplementasi dan diuji fungsinya satu persatu, diintegrasikan sehingga menjadi sebuah sistem perbandingan algoritma Nazief & Adriani dengan algoritma Idris untuk pencarian kata dasar pada teks bahasa Indonesia yang mampu mengetahui kecepatan proses hingga akurasi hasil dari masukkan pengguna. Setelah menjadi sebuah aplikasi, dilakukan pengujian terhadap sistem perbandingan algoritma Nazief & Adriani dengan algoritma Idris untuk pencarian kata dasar pada teks bahasa Indonesia yang telah dibuat untuk memastikan bahwa sistem menerima masukan teks bahasa Indonesia, memproses teks tersebut untuk diolah, dan menghasilkan informasi dari hasil perbandingan algoritma Nazief & Adriani dengan algoritma Idris.

10. *Maintenance* atau Perawatan

Setelah sistem perbandingan algoritma Nazief & Adriani dengan algoritma Idris untuk pencarian kata dasar pada teks bahasa Indonesia diuji dan sesuai dengan kebutuhan pengguna, sistem ini dipasang dan mulai digunakan. Jika selama penggunaan terdapat kekurangan atau terjadi masalah, maka akan dilakukan pengembangan dan perbaikan.

3.2.2 Algoritma Nazief & Adriani

Algoritma Nazief & Adriani digunakan pada aplikasi ini untuk melakukan pencarian kata dasar berdasarkan teks bahasa Indonesia yang telah *diinputkan*.

Aplikasi ini menggunakan algoritma Nazief & Adriani alasannya yaitu:

1. Algoritma ini dirancang khusus untuk melakukan pencarian kata dasar bahasa Indonesia.
2. Panjang teks yang akan diolah tidak terbatas.

Berikut merupakan bagaimana algoritma Nazief & Adriani bekerja pada aplikasi pengujian :

Contoh kata: “menyalahkanku”.

1. Mengecek kata “menyalahkanku” pada kamus. Hasil pengecekan di kamus adalah tidak ditemukan. Maka kata “menyalahkanku” akan di proses ke langkah selanjutnya.
2. Membuang infleksi akhiran (“-lah”, “-kah”, “-ku”, “-mu”, atau “-nya”), maka kata “menyalahkanku” menjadi “menyalahkan”.
3. Membuang penurunan akhiran (“-i”, “-an” atau “-kan”), maka kata “menyalahkan” menjadi “menyalah” dan mengecek ke kamus. Hasil pengecekan tidak menemukan kata “menyalah”, maka dilanjutkan ke langkah selanjutnya.
4. Mengubah turunan awalan “meny-” menjadi huruf “s-”, maka kata “menyalah” menjadi kata “salah”, setelah di ubah kata ”salah” di cek pada kamus dan menghasilkan kata dasar yang benar. Maka kata “salah” menjadi kata dasar dari kata “menyalahkanku”.

3.2.3 Algoritma Idris

Algoritma Idris digunakan pada aplikasi ini untuk melakukan pencarian kata dasar berdasarkan teks bahasa Indonesia yang telah diinputkan.

Aplikasi ini menggunakan algoritma Idris alasannya yaitu:

1. Algoritma ini dirancang khusus untuk melakukan pencarian kata dasar bahasa melayu yang memiliki kemiripan dengan bahasa Indonesia.
2. Panjang teks yang akan diolah tidak terbatas.

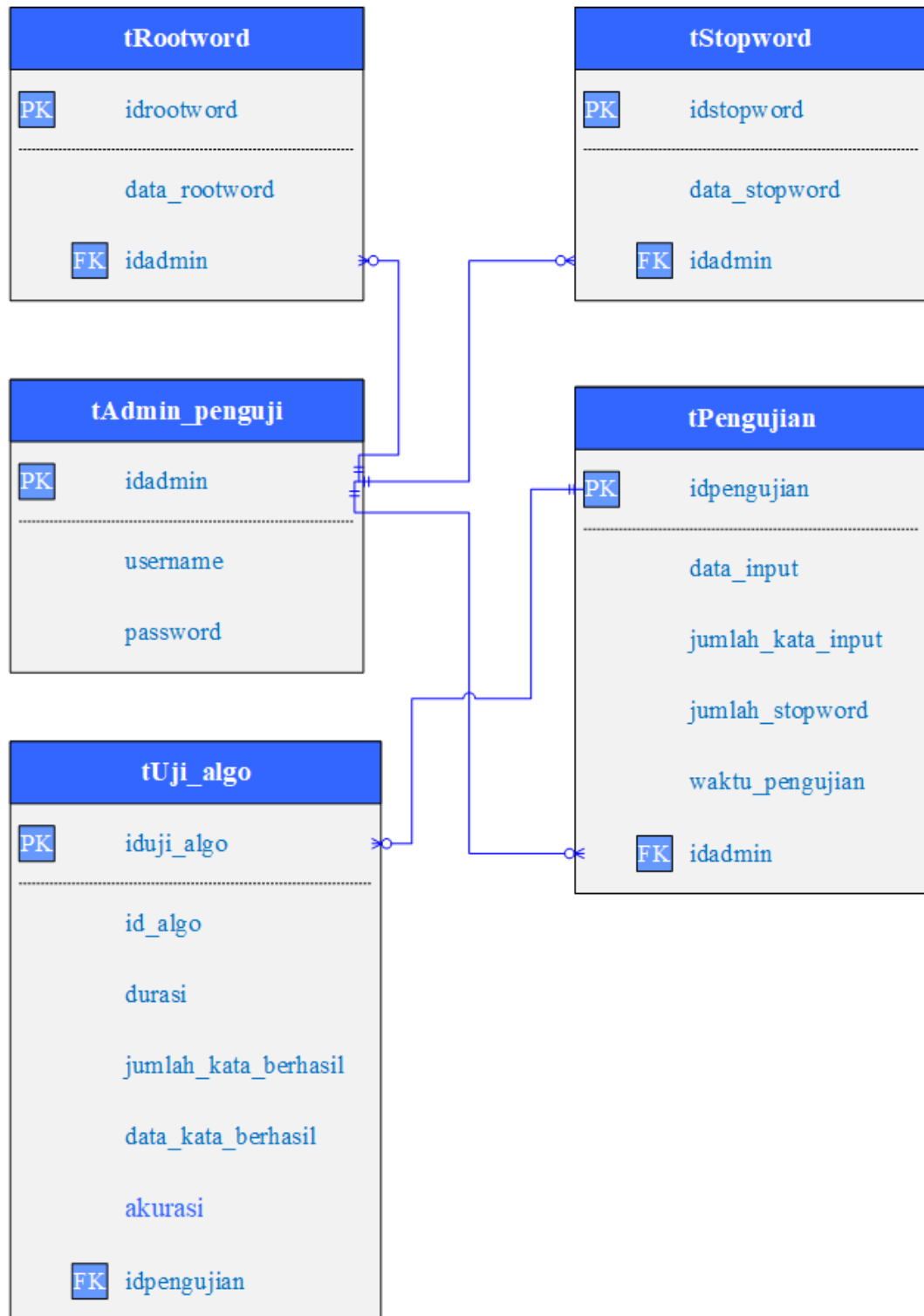
Berikut merupakan bagaimana algoritma Idris bekerja pada aplikasi pengujian :

Contoh kata: “menyalahkanku”.

1. Mengecek kata “menyalahkanku” pada kamus. Hasil pengecekan di kamus adalah tidak ditemukan. Maka kata “menyalahkanku” akan di proses ke langkah selanjutnya.
2. Mengubah turunan awalan “meny-” menjadi huruf “s-”, maka kata “menyalahkanku” menjadi kata “salahkanku”, setelah itu kata “salahkanku” di cek pada kamus, kata dasar belum ditemukan dan lanjut ke langkah selanjutnya.
3. Membuang infleksi akhiran (“-lah”, “-kah”, “-ku”, “-mu”, atau “-nya”), maka kata “salahkanku” menjadi “salahkan”.
4. Membuang penurunan akhiran (“-i”, “-an” atau “-kan”), maka kata “salahkan” menjadi “salah” dan mengecek ke kamus. Hasil pengecekan ditemukan, maka kata “salah” menjadi kata dasar dari kata “menyalahkanku”.

3.3 Perancangan Basis Data

3.3.1 Entity Relationship Diagram (ERD)



Gambar 3.1 Gambar Entity Relationship Diagram (ERD) pengujian

Keterangan *Entity Relationship Diagram* diatas:

1. Entitas yang terdapat pada diagram diatas:

- a. tRootword.
- b. tStopword.
- c. tAdmin_penguji.
- d. tPengujian.
- e. tUji_algo.

2. Atribut untuk setiap entitas:

- a. tRootword: idrootword, data_rootword, idadmin.
- b. tStopword: idstopword, data_stopword, idadmin.
- c. tAdmin_penguji: idadmin, username, password.
- d. tPengujian: idpengujian, data_input, jumlah_kata_input, jumlah_stopword, waktu_pengujian, idamin.
- e. tUji_algo: iduji_algo, id_algo, durasi, jumlah_kata_berhasil, data_kata_berhasil, akurasi, idpengujian.

3. Hubungan antar entitas dan penentuan kardinalitas :

- a. tAdmin_penguji digunakan untuk tRootword.
 - 1) Tabel utama: tAdmin_penguji.
 - 2) Tabel kedua: tRootword.
 - 3) *Relationship: one-to-many*(1:m).
 - 4) Atribut penghubung: idadmin (FK idadmin di tRootword).
- b. tAdmin_penguji digunakan untuk tStopword.
 - 1) Tabel utama: tAdmin_penguji.
 - 2) Tabel kedua: tStopword.

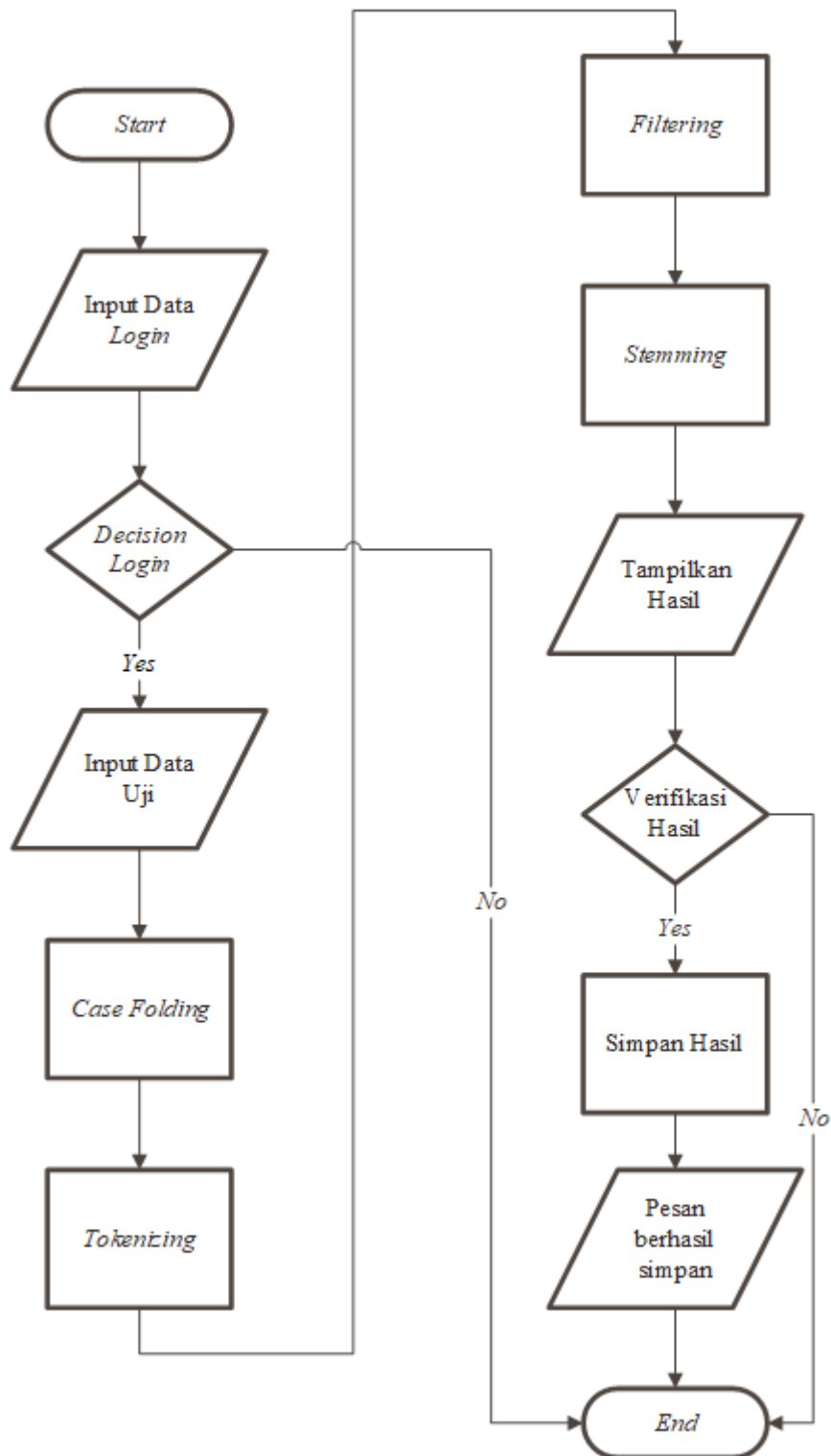
- 3) *Relationship: one-to-many(1:m)*.
 - 4) Atribut penghubung: idadmin (FK idadmin di tStopword).
- c. tAdmin_penguji digunakan untuk tPengujian.
- 1) Tabel utama: tAdmin_penguji.
 - 2) Tabel kedua: tPengujian.
 - 3) *Relationship: one-to-many(1:m)*.
 - 4) Atribut penghubung: idadmin (FK idadmin di tPengujian).
- d. tPengujian digunakan untuk tUji_algo.
- 1) Tabel utama: tPengujian.
 - 2) Tabel kedua: tUji_algo.
 - 3) *Relationship: one-to-many(1:m)*.
 - 4) Atribut penghubung: idpengujian (FK idpengujian di tUji_algo).

3.4 Perancangan Proses

3.4.1 Flow Chart

Flow Chart berguna untuk mempresentasikan alir kendali dalam program pengujian.

3.4.1.1 Flow Chart Pengujian



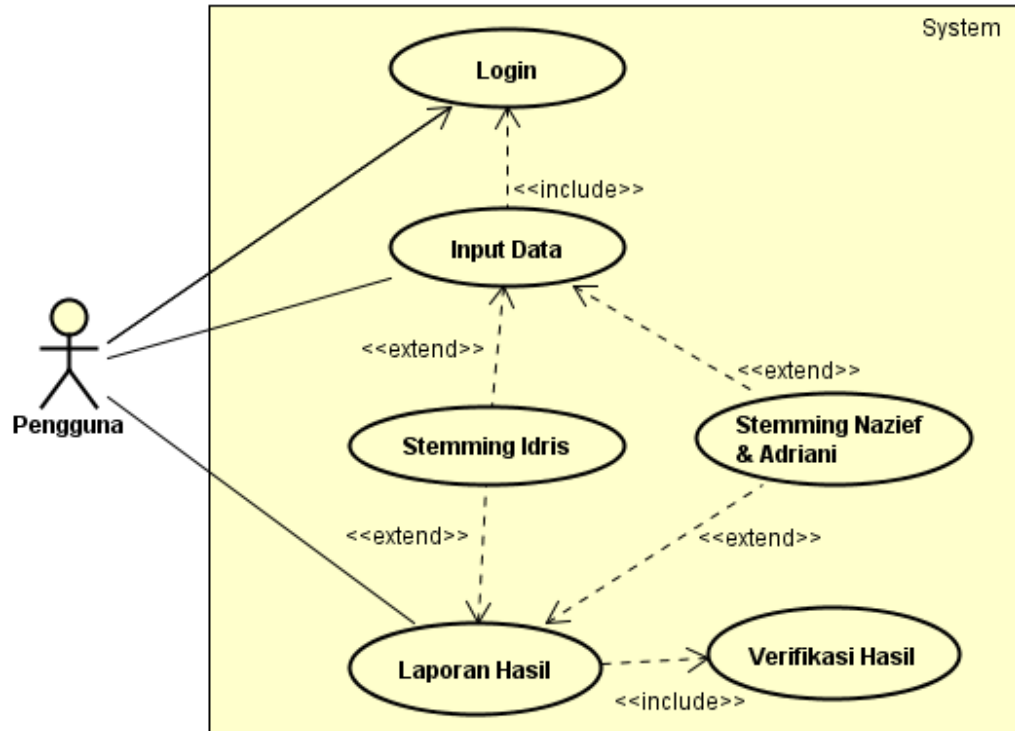
Gambar 3.2 Gambar *Flowchart* pengujian

Flowchart pada gambar 3.2 menggambarkan proses pengujian. Proses pertama penguji harus melakukan *login* dengan memasukkan *username* dan

password yang benar, lalu sistem akan mengecek apakah masukkan tadi benar atau salah. Jika masukkan salah maka proses akan di akhiri, namun apabila masukkan benar maka penguji dapat melakukan pengujian. Melakukan masukkan data uji untuk selanjutnya diproses. Proses pertama yang akan dilakukan adalah *case folding*, yaitu merubah tiap huruf yang telah di masukkan menjadi huruf kecil. Langkah selanjutnya *tokenizing* atau memotong kalimat menjadi kata-kata yang akan digunakan pada proses *filtering*. *Filtering* digunakan untuk menghilangkan kata-kata yang dianggap tidak penting. Setelah kata yang dirasa penting didapat dari *filtering*, maka proses berikutnya ialah melakukan *stemming*. Proses *stemming* melakukan pengolahan kata untuk menemukan kata dasar atau *root word*. Setelah proses *stemming* selesai maka akan menampilkan hasil pengolahan beserta rincian mengenai waktu pengujian, jumlah kata masukkan, jumlah kata yang dibuang, durasi pemrosesan, dan jumlah kata hasil. Setelah hasil ditampilkan maka penguji memutuskan untuk memverifikasi hasil yang telah diolah. Jika penguji memverifikasi hasil olahan maka informasi tadi akan disimpan dan menampilkan pesan bahwa informasi telah disimpan, namun jika penguji tidak memverifikasi hasil olahan maka proses berakhir.

3.4.2 Perancangan Proses Diagram UML

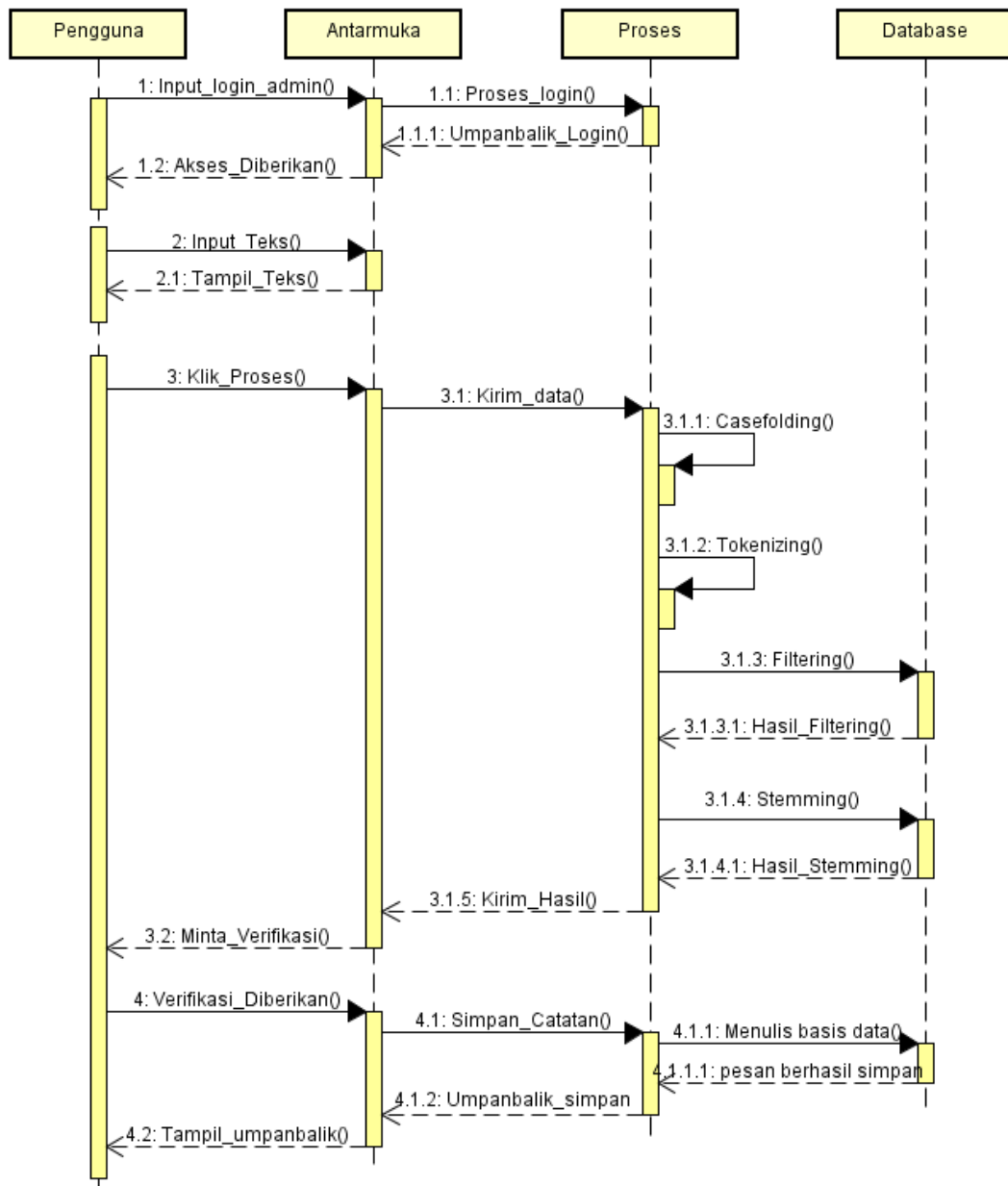
3.4.2.1 Use Case Diagram



Gambar 3.3 Gambar Use case diagram pengujian

Use Case diagram pada gambar 3.3 menggambarkan interaksi antara penguji dengan sistem pada saat melakukan pengujian. Pertama penguji harus melakukan *login* untuk masuk ke dalam sistem. Setelah *login* penguji memasukkan data yang akan diolah. Setelah mengisi masukkan, maka akan menjalankan operasi untuk *stemming* dengan algoritma Idris dan algoritma Nazief & Adriani. Setelah proses *stemming* selesai maka hasilnya akan di perlihatkan ke penguji untuk selanjutnya di verifikasi penguji sebagai konfirmasi bahwa pengujian telah selesai.

3.4.2.2 Sequence Diagram

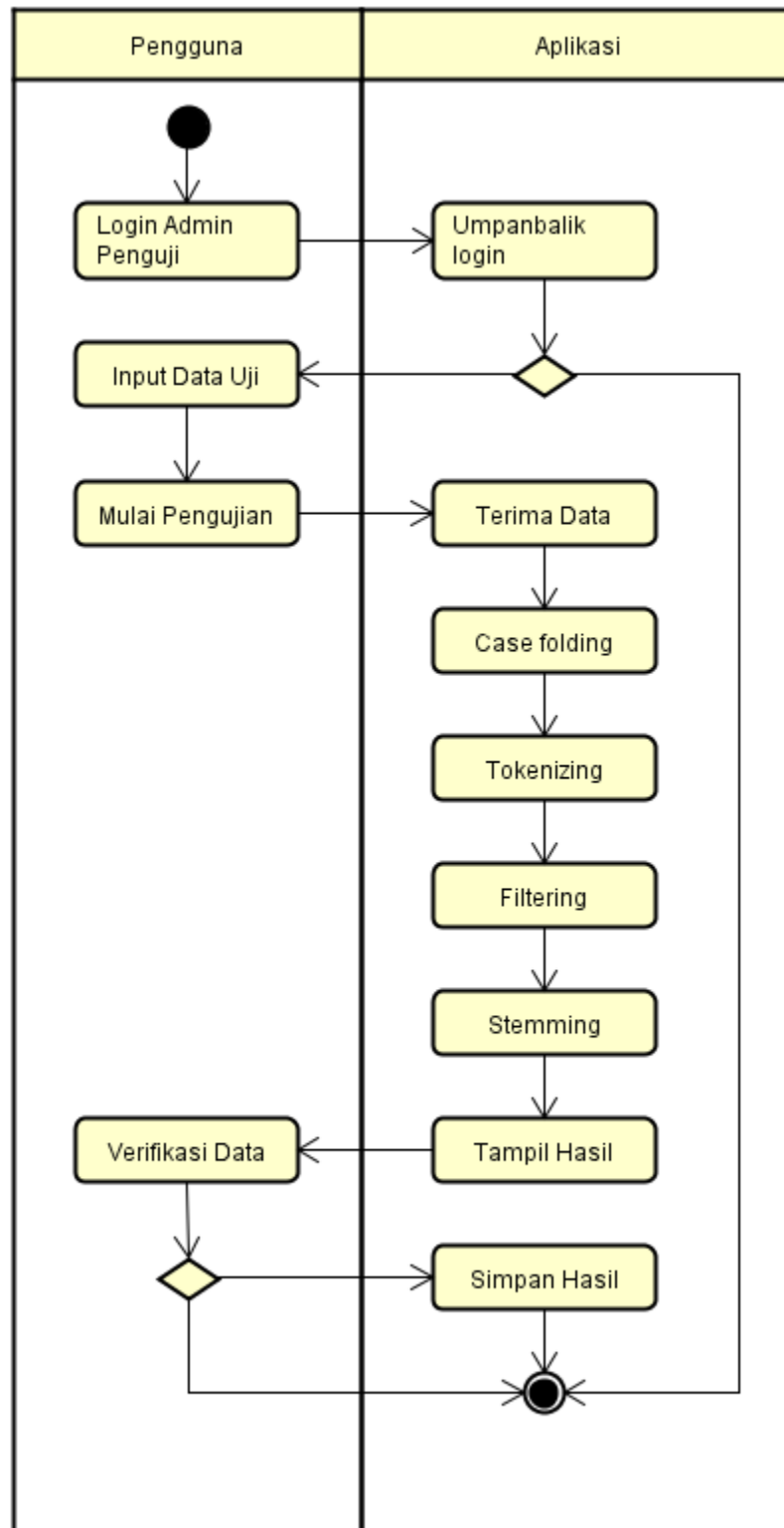


Gambar 3.4 Gambar *sequence diagram* pengujian

Sequence diagram pada gambar 3.4 yang menggambarkan proses yang terjadi pada saat pengujian. Pertama penguji memasukkan data *login* pada halaman antarmuka untuk masuk ke dalam sistem, antarmuka akan melanjutkan data masukkan ke bagian proses untuk di cek kebenaran data *login*. Hasil dari pengecekan akan dikirim kembali ke antarmuka setelah di proses agar penguji mengetahui hasil masukkan *login*. Ketika penguji berhasil masuk ke dalam sistem,

penguji dapat memasukkan data yang akan diuji dan melihat data masukkan yang akan diuji. Setelah penguji selesai melakukan masukkan data uji, penguji dapat memulai proses pengujian dengan menekan *button* pada antarmuka. Setelah *button* pada antarmuka ditekan maka antarmuka akan mengirim data uji masukkan ke bagian proses untuk diolah. Setelah data uji masukkan diterima bagian proses akan menjalankan operasi *case folding* dan *tokenizing*. Setelah proses *tokenizing* selesai proses akan memulai operasi *filtering* untuk menyaring kata yang dianggap tidak penting. Untuk penyaringan kata diperlukan koneksi basis data untuk mengetahui kumpulan kata-kata yang dianggap tidak penting sebagai dasar dari penyaringan. Setelah operasi *filtering* selesai, maka operasi *stemming* dapat dimulai. Hasil dari *filtering* menghasilkan kata-kata yang telah dianggap penting akan olah dan di cek hasil olahannya dengan kumpulan kata dasar yang telah disimpan di basis data. Setelah menghasilkan kata-kata dasar dari data uji masukkan, hasil kata-kata tersebut akan ditampilkan antarmuka untuk diperlihatkan ke penguji. Pada saat penguji mengetahui hasil pengolahan, penguji harus melakukan umpan balik pada antarmuka sebagai konfirmasi bahwa pengujian telah berhasil dan hasil pengujian akan di simpan pada basis data.

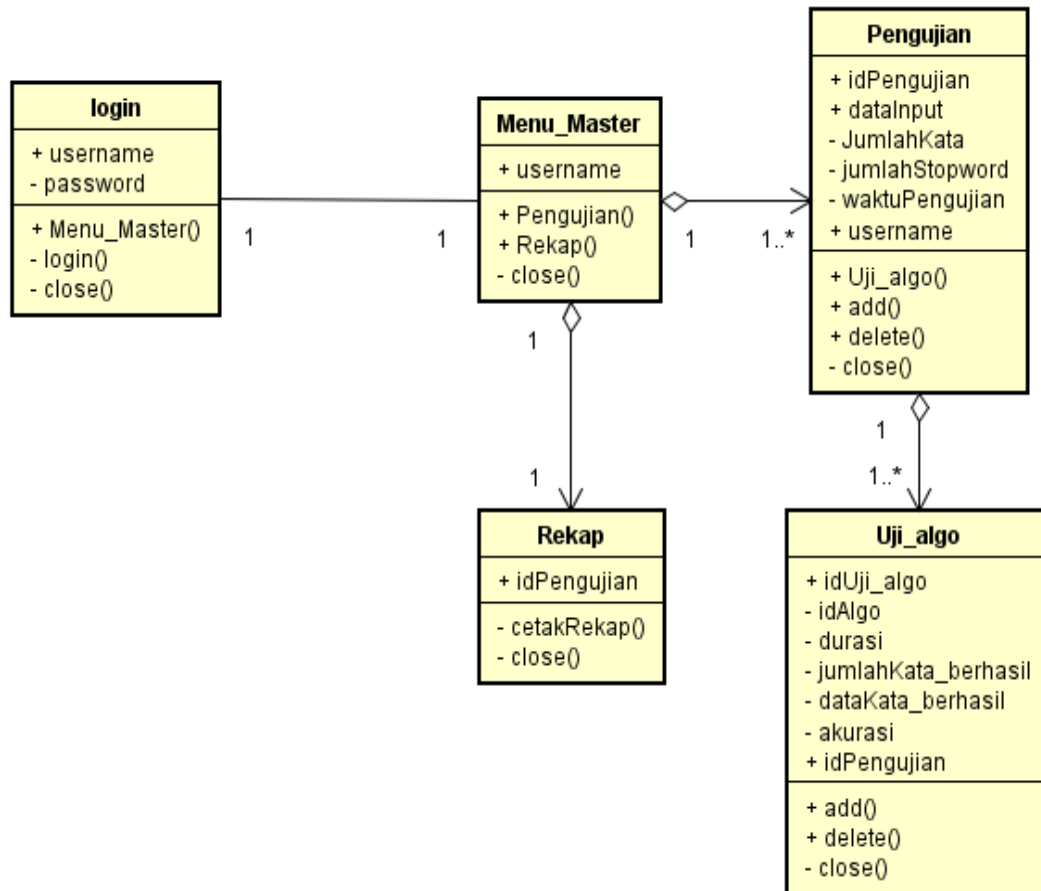
3.4.2.3 Activity Diagram



Gambar 3.5 Gambar Activity diagram pengujian

Activity diagram pada gambar 3.5 yang menggambarkan proses yang terjadi pada saat pengujian. Proses pertama penguji melakukan *login* untuk masuk ke dalam sistem. Jika masukkan untuk *login* benar maka penguji dapat masuk ke dalam sistem untuk melakukan pengujian. Ketika penguji berada di dalam sistem penguji dapat memasukkan data uji untuk pengujian. Ketika data uji selesai dimasukkan, proses dapat dimulai dengan menekan *button* pada antarmuka. Setelah *button* pada antarmuka ditekan maka sistem akan memproses data masukkan. Proses yang akan dilakukan sistem yaitu melakukan operasi *case folding* untuk menyamakan semua huruf ke huruf kecil, lalu tiap kalimat akan dipotong menjadi kata-kata pada proses *tokenizing*. Selanjutnya kata-kata yang telah terpotong akan di cek untuk membuang kata-kata yang dianggap yang tidak penting. Setelah mendapat kata-kata yang dianggap penting, kata-kata tersebut akan diolah dan dicek untuk mengetahui kata dasarnya. Pada saat mengetahui kata dasar dari masing-masing kata, hasilnya akan diperlihatkan ke penguji untuk di verifikasi dan disimpan hasil olahan tadi pada basis data.

3.4.2.4 Class Diagram

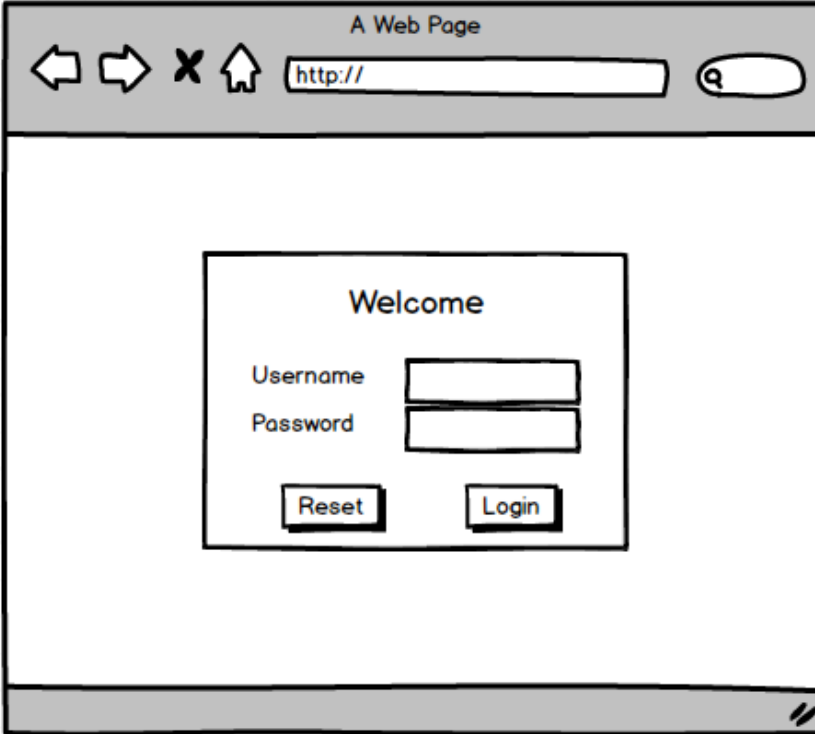


Gambar 3.6 Gambar Class diagram pengujian

Class diagram pada gambar 3.6 menggambarkan relasi antar objek yang terdiri dari 5 class. Class Menu_Master mempunyai relasi satu ke banyak dengan class Pengujian, sedangkan relasi satu ke satu dengan class login dan class Rekap. Untuk mengakses class Menu_Master harus melewati class login. Class Pengujian mempunyai relasi satu ke banyak dengan class Uji_algo untuk melakukan operasi stemming menggunakan algoritma Nazief & Adriani dan Idris.

3.5 Perancangan Tampilan

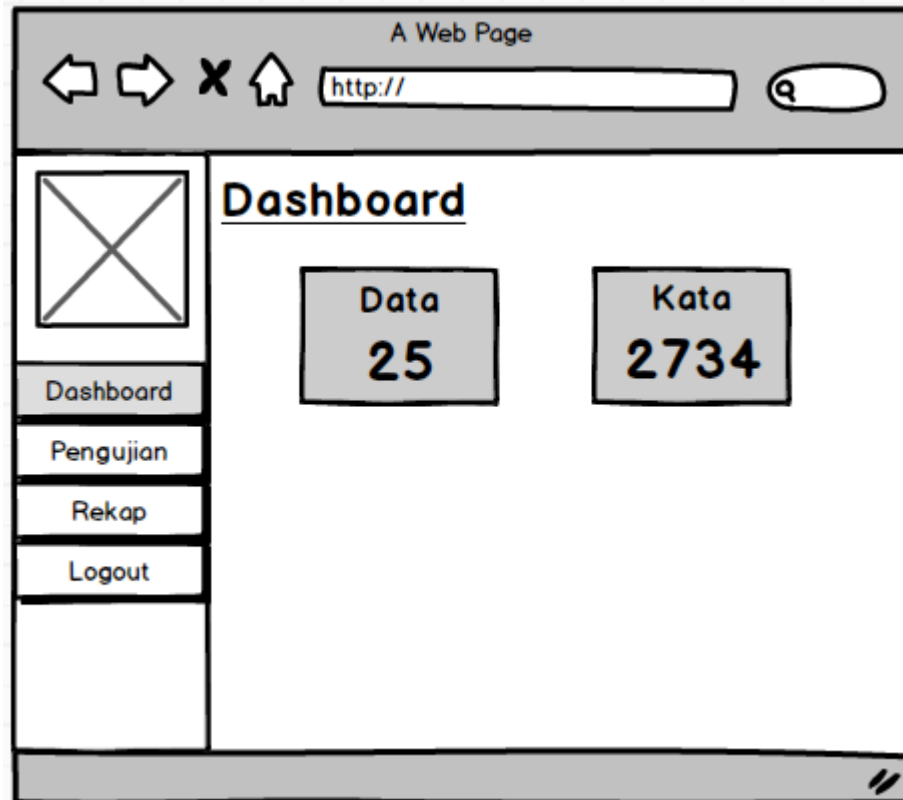
Berikut ini adalah rancangan tampilan antarmuka yang akan digunakan pada sistem pengujian:



The diagram illustrates a web browser window titled "A Web Page". The browser's address bar shows "http://". The main content area contains a login form with the heading "Welcome". The form includes two text input fields labeled "Username" and "Password", a "Reset" button, and a "Login" button.

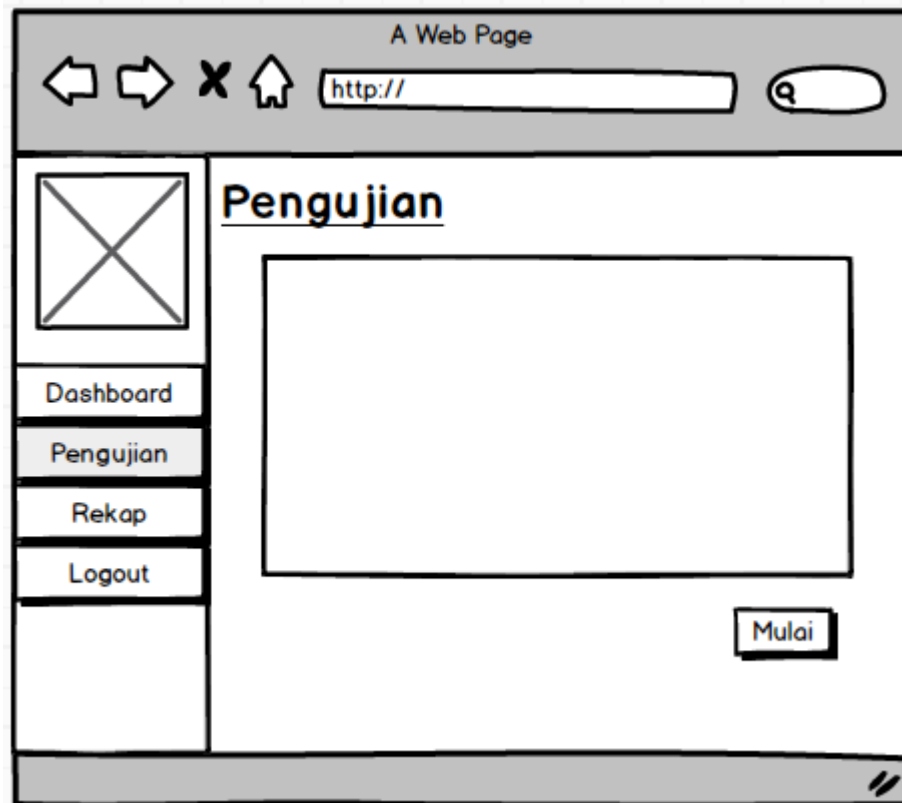
Gambar 3.7 Gambar Form login

Fungsi *from login* ialah untuk masuk ke dalam aplikasi pengujian. Di dalam *form* terdapat *text field* untuk mengisi *username* dan *password*, serta terdapat *button login* untuk masuk dan *button reset* mengkosongkan *text field*. Jika *login* berhasil maka akan penguji akan di arahkan ke laman *dashboard*.



Gambar 3.8 Gambar *Form dashboard*

Fungsi *form dashboard* ialah sebagai laman utama dari *web* pengujian. Pada bagian kiri terdapat gambar yang akan menjadi *icon* dari aplikasi *web* pengujian, serta menu utama yang di susun menurun. Pada bagian tengah laman, terdapat label yang bertuliskan *dashboard* sebagai identitas dari laman. Serta terdapat label dari jumlah data yang telah di uji dan jumlah kata yang proses.



Gambar 3.9 Gambar *Form* penguajian

Fungsi *form* Penguajian adalah sebagai antarmuka untuk memulai penguajian. Pada bagian kiri terdapat gambar dari *icon* penguajian dan menu utama yang di susun menurun. Pada bagian utama terdapat label Penguajian sebagai identitas laman, *text* area sebagai tempat masukkan data yang akan di uji. *Button* Mulai berfungsi sebagai *trigger* untuk memulai proses penguajian.

A Web Page

http://

Hasil Pengujian

Waktu Pengujian : 27 September 2017, 20:11:55
 Jumlah Kata Inputan : 75
 Jumlah stopword : 45

Nazief & Adriani

Durasi : 3 detik
 Jumlah Kata Hasil : 30

Lorem ipsum dolor sit amet consectetur adipiscing elit Proin viverra turpis quis liber dignissim at gravida massa scelerisque

Idris

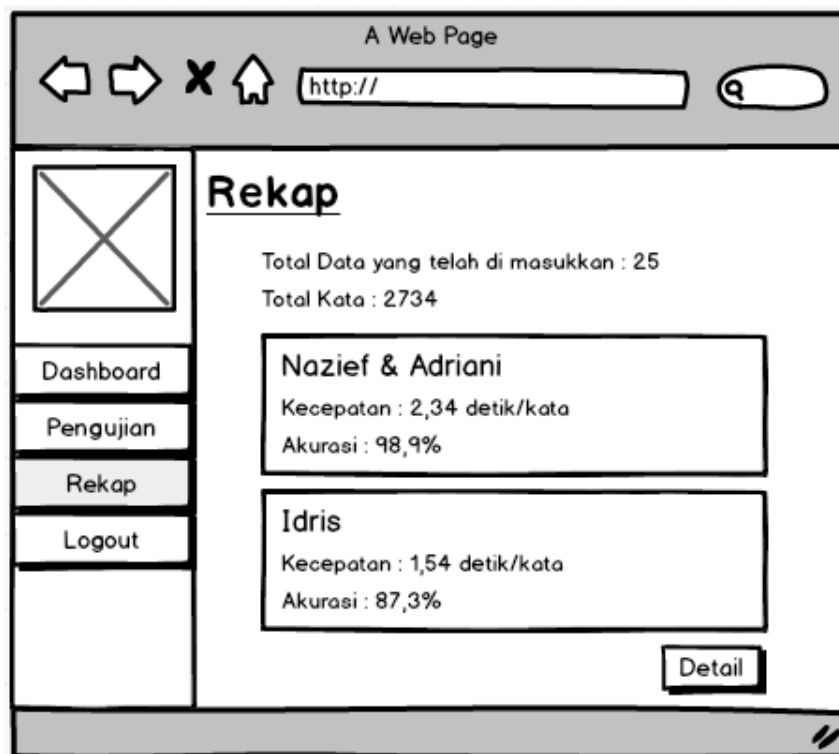
Durasi : 3 detik
 Jumlah Kata Hasil : 30

Lorem ipsum dolor sit amet consectetur adipiscing elit Proin viverra turpis quis liber dignissim at gravida massa scelerisque

Verifikasi

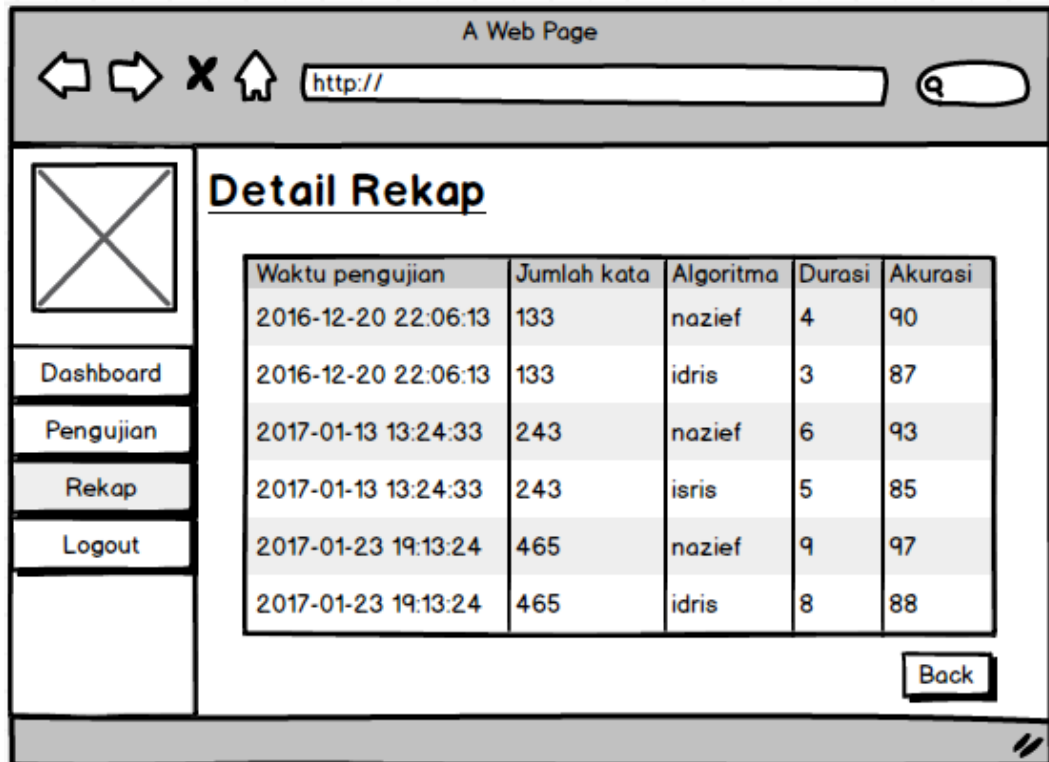
Gambar 3.10 Gambar Form hasil pengujian

Fungsi *form* hasil pengujian ialah menampilkan hasil proses pengujian terhadap data uji masukkan. Pada bagian kiri terdapat gambar dari *icon* pengujian dan menu utama yang di susun menurun. Pada bagian utama terdapat label Hasil Pengujian sebagai identitas laman, label waktu pengujian berfungsi sebagai penunjuk waktu pengujian di lakukan, label jumlah kata *inputan* berfungsi sebagai petunjuk jumlah kata yang terdapat pada data uji masukkan, label jumlah *stopword* berfungsi sebagai penunjuk jumlah kata yang di buang karena dianggap tidak penting, label Nazief & Adriani dan Idris digunakan sebagai penunjuk algoritma yang diterapkan, label durasi berfungsi sebagai penunjuk lama waktu pemrosesan yang dilakukan algoritma tersebut, label jumlah kata hasil berfungsi sebagai penunjuk jumlah kata yang di hasilkan dari pemrosesan dari masing-masing algoritma dan *text* area berfungsi sebagai penampung data kata yang berhasil di olah.



Gambar 3.11 Gambar *Form* rekap

Fungsi *form* Rekap ialah untuk menampilkan kesimpulan dari hasil pengujian. Pada bagian kiri terdapat gambar dari *icon* pengujian dan menu utama yang di susun menurun. Pada bagian utama terdapat label Rekap sebagai identitas laman, label total data yang telah di masukkan berfungsi sebagai penunjuk jumlah data yang telah di olah, label total kata berfungsi sebagai penunjuk jumlah kata yang telah di olah, terdapat 2 kotak sebagai wadah untuk menampilkan kedua hasil pengujian algoritma, label Nazief & Adriani dan Idris berfungsi sebagai penunjuk algoritma yang digunakan, label kecepatan digunakan sebagai penunjuk kecepatan algoritma tersebut dapat mengolah kata dalam satuan detik per kata, label akurasi digunakan sebagai penunjuk akurasi yang ditentukan oleh perhitungan jumlah kata yang dianggap penting dengan jumlah kata yang dihasilkan setelah diolah algoritma tersebut, serta *button* Detail sebagai penjelasan mengenai hasil yang ditampilkan pada laman ini.



Gambar 3.12 Gambar *Form* detail rekap

Fungsi *form* detail rekap ialah sebagai penjelasan mengenai rekap yang telah ditampilkan pada laman Rekap. Pada bagian kiri terdapat gambar dari *icon* pengujian dan menu utama yang di susun menurun. Pada bagian utama terdapat label Detail Rekap yang berfungsi sebagai identitas laman, tabel pada laman ini digunakan sebagai penyaji data pengujian yang telah disimpan, dan *button* Back berfungsi untuk kembali ke laman Rekap.

3.6 JADWAL PENELITIAN

Penjadwalan merupakan faktor yang penting dalam penelitian karena perencanaan yang jelas akan membuat penelitian berjalan dengan baik. Jadwal penelitian ini dapat dilihat pada tabel 3.2.

Tabel 3.1 Tabel Jadwal penelitian

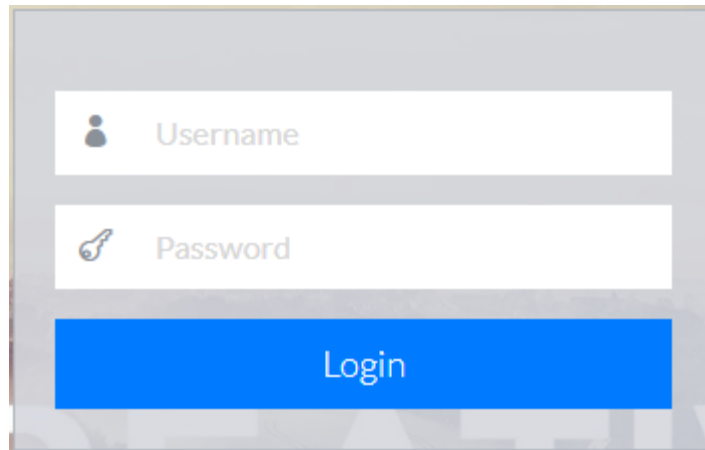
Kegiatan dan Waktu Pelaksanaan	Agustus				September				Oktober			
	1	2	3	4	1	2	3	4	1	2	3	4
Analisis Kebutuhan												
Pemilihan Metode												
Perancangan UML												
Perancangan Basis Data												
Pengkodean												
Pengujian												
Implementasi & Pemeliharaan												

BAB 4

IMPLEMENTASI APLIKASI

4.1 Implementasi Antarmuka Pengguna

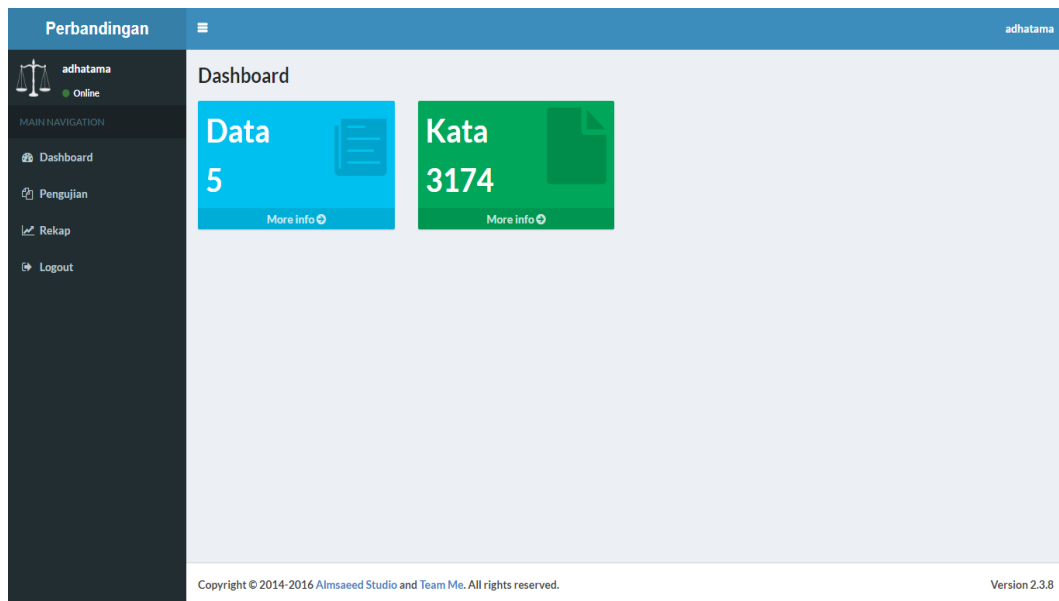
4.1.1 Tampilan *Login*

The image shows a login form with a light gray background. It contains two text input fields: the top one is labeled 'Username' with a person icon, and the bottom one is labeled 'Password' with a key icon. Below these fields is a prominent blue button with the text 'Login' in white.

Gambar 4.1 Gambar Tampilan *login*

Pada gambar diatas merupakan tampilan *form login* yang berfungsi untuk membatasi hak akses agar tidak sembarang orang dapat melakukan pengujian. Dalam *form login* terdapat 2 *text field* yang berfungsi sebagai tempat masukkan *username* dan *password*. Lalu terdapat *button login* untuk memulai autentikasi dari data yang dimasukkan. Jika *login* berhasil maka pengguna akan di arahkan ke laman *dashboard*, namun jika gagal maka pengguna kembali ke laman *login*.

4.1.2 Tampilan *Dashboard*



Gambar 4.2 Gambar Tampilan *dashboard*

Pada gambar diatas merupakan tampilan dari *form dashboard* yang berfungsi sebagai laman utama. Pada laman ini akan menampilkan sekilas data dari jumlah data yang telah diuji dan jumlah kata yang telah diproses pada sistem ini. Jika kedua persegi tersebut di klik maka pengguna akan diarahkan menuju laman rekap.

4.1.3 Tampilan Pengujian



Gambar 4.3 Gambar Tampilan pengujian

Pada gambar diatas merupakan tampilan dari laman pengujian yang berfungsi untuk memulai pengujian. Pada laman ini terdapat *text area* yang berfungsi sebagai wadah untuk teks masukkan yang akan di uji. Lalu tombol *submit* untuk memulai pengujian.

4.1.4 Tampilan Hasil Pengujian

Perbandingan Algoritma

Verifikasi Hasil

Waktu pengujian : 06-11-17 10:29:46
 Jumlah Kata Inputan : 391
 Jumlah Stopword : 155

Pernah memperhatikan antrean di ruang tunggu praktik dokter kandungan atau dokter anak? Atau di sekolah Playgroup/TK? Ada agen yang rajin mencari nasabah di 2 tempat tersebut. Tak jarang agen ini juga salah satu dari para orang tua ini.

Produk asuransi pendidikan ini rajin ditawarkan oleh agen-agen asuransi pada pasien dokter kandungan atau orang tua pasien dokter anak, juga di sekolah-sekolah Play Group atau TK. Mengapa? Karena di sana terdapat orang tua yang sudah membutuhkan biaya pendidikan, sehingga menjadi target market yang potensial.

Asuransi pendidikan banyak diambil oleh sebagian masyarakat Indonesia dengan alasan

Hasil Stemming Nazief & Adriani

Durasi : 11.424260139465 detik
 Jumlah Kata Hasil 236

hati antre ruang tunggu praktik dokter kandung dokter anak sekolah playgroup tk agen rajin cari nasabah 2 jarang agen salah orang tua produk asuransi didik rajin ditawarkan agen agen asuransi pasien dokter kandung orang tua pasien dokter anak sekolah sekolah play group tk orang tua butuh biaya didik target market potensial asuransi didik ambil masyarakat indonesia alas tambah manfaat proteksi risiko paham produk beli agen asuransi didik nama asuransi didik asuransi didik asuransi investasi dana didik kategori produk asuransi jiwa whole life endowment unit link jenis dekade asuransi didik kategori unit link unit link produk canggih beli salah contoh tax planning asuransi didik dana tahap cair jenjang masuk sekolah 5 kepesertaan jenjang dekat anak usia 4 dana tahap cair jenjang masuk smp beli asuransi didik mari lihat fakta didik asuransi 1 naik gaji indonesia

Jumlah Kata Salah : 0

Hasil Stemming Idris

Durasi : 7.1040930747986 detik
 Jumlah Kata Hasil 236

memperhatikan antrre ruang tunggu praktik dokter kandung dokter anak sekolah playgroup tk agen rajin cari nasabah 2 jarang agen salah orang tua produk asuransi pendidikan rajin ditawarkan agen agen asuransi pasien dokter kandung orang tua pasien dokter anak sekolah sekolah play group tk orang tua membutuhkan biaya pendidikan target market potensial asuransi pendidikan ambil masyarakat indonesia alas tambah manfaat proteksi risiko memahami produk beli agen asuransi pendidikan nama asuransi pendidikan asuransi pendidikan asuransi investasi dana pendidikan kategori produk asuransi jiwa whole life endowment unit link jenis dekade asuransi pendidikan kategori unit link unit link produk canggih beli salah contoh tax planning asuransi pendidikan dana tahap cair jenjang masuk sekolah 5 kepesertaan jenjang dekat anak usia 4 dana tahap cair

Jumlah Kata Salah : 0

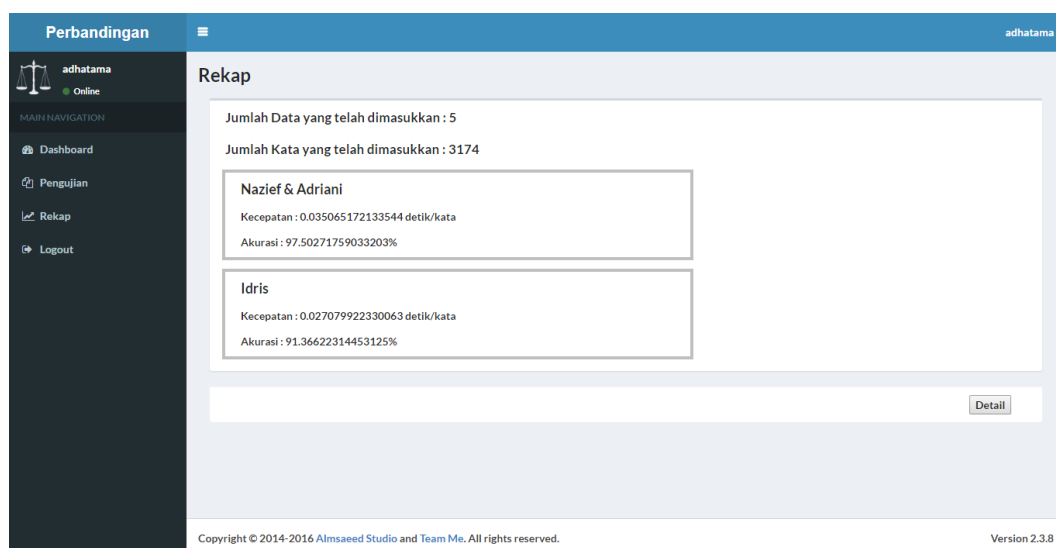
Submit

Copyright © 2014-2016 Almsaeed Studio and Team Me. All rights reserved.

Gambar 4.4 Gambar Tampilan hasil pengujian

Gambar diatas merupakan tampilan laman verifikasi hasil yang berfungsi untuk menampilkan hasil *stemming* dari algoritma Nazief & Adriani maupun algoritma Idris. Dibagian atas menunjukkan informasi umum mengenai pengujian seperti waktu pengujian, jumlah kata yang dimasukkan untuk diuji, hingga jumlah *stopword* yang telah dihilangkan dari teks yang akan di proses. Selanjutnya adalah tampilan informasi dari masing-masing algoritma penguji, seperti durasi pemrosesan, jumlah kata yang di olah, data kata telah diolah, serta disediakan *text field* untuk menampung masukkan jumlah kata yang tidak diharapkan dari hasil yang telah di proses.

4.1.5 Tampilan Rekap



Gambar 4.5 Gambar Tampilan rekap

Gambar diatas merupakan tampilan dari laman rekap yang berfungsi sebagai penyedia informasi mengenai pengujian yang telah dilakukan. Pada laman rekap data yang ditampilkan adalah sebagai berikut, jumlah data yang telah dimasukkan, jumlah kata yang telah dimasukkan, hingga kecepatan dan akurasi dari masing-masing algoritma penguji. Tersedia tombol *detail* jika ingin mengetahui informasi mengenai pengujian yang telah dilakukan.

4.1.6 Tampilan Detail Rekap

Perbandingan adhatama

adhatama
Online

MAIN NAVIGATION

- Dashboard
- Pengujian
- Rekap
- Logout

Detail Rekap

Show entries Search: Tambah

No.	Waktu Pengujian	Jumlah Kata	Algoritma	Durasi (detik)	Akurasi (%)
1	2017-11-03 14:43:26	310	Nazief & Adriani	13.8621	98.0645
2	2017-11-03 14:43:26	310	Idris	10.1075	89.3548
3	2017-11-03 15:14:28	137	Nazief & Adriani	5.07603	99.2701
4	2017-11-03 15:14:28	137	Idris	4.45825	97.8102
5	2017-11-03 15:26:31	463	Nazief & Adriani	26.247	99.3521
6	2017-11-03 15:26:31	463	Idris	19.229	90.9287
7	2017-11-03 15:40:38	609	Nazief & Adriani	39.4276	98.0296
8	2017-11-03 15:40:38	609	Idris	30.6918	90.1478
9	2017-11-03 16:03:06	341	Nazief & Adriani	14.0547	97.654
10	2017-11-03 16:03:06	341	Idris	10.6303	88.2698

Showing 1 to 10 of 60 entries Previous **1** 2 3 4 5 6 Next

Gambar 4.6 Gambar Tampilan detail rekap

Pada gambar diatas merupakan tampilan dari laman detail rekap yang berfungsi sebagai penampil informasi dari tiap pengujian yang telah dilakukan. Pada bagian atas kanan terdapat tombol tambah untuk menambah pengujian, kotak pencarian, hingga *combo box* untuk menentukan jumlah informasi yang akan ditampilkan, pada tabel yang ditampilkan terdapat atribut no untuk menunjukkan nomor, waktu pengujian, jumlah kata, algoritma, durasi, dan akurasi. Pada bagian bawah terdapat pilihan halaman untuk menuju halaman berikutnya mengenai detail rekap selanjutnya.

4.2 Implementasi Proses dan Algoritma

4.2.1 Implementasi Proses

Proses yang di kerjakan oleh sistem perbandingan algoritma Nazief & Adriani dengan algoritma Idris memiliki 3 tahapan secara garis besar, yaitu proses *text preprocessing*, perhitungan waktu *stemming*, dan menyimpan ke *database*.

4.2.1.1 Proses *Text Preprocessing*

Proses pertama yang dilakukan setelah tombol *submit* ditekan pada laman pengujian ialah *text preprocessing*. Proses ini bertujuan untuk menyiapkan teks yang akan diuji, dimulai dari *case folding* atau mengubah semua karakter menjadi huruf kecil, *tokenizing* atau melakukan pemotongan paragraf yang akan menjadi kata-kata, dan *filtering* atau menghilangkan *stopword*. Berikut merupakan implementasi *text preprocessing* didalam sistem.

```

<?php
require_once ('koneksi/connect.php');
//tampung data inputan
$kata_awal = $_POST['content'];
//tampung data inputan untuk di proses
$str = $kata_awal;
//casefolding atau mengubah ke huruf kecil
$str = mb_convert_case($str, MB_CASE_LOWER, "UTF-8");

//hilangkan special char
$str = str_replace(' ', '-', $str);
$str = preg_replace('/[^A-Za-z0-9\ - ? " " ! ]/', ' ', $str);
$str = str_replace('-', ' ', $str);
$str = str_replace(' ', ' ', $str);
$str = str_replace(' ', ' ', $str);
//memanggil data stopwords dari database
$sql = mysql_query("SELECT data_stopword from tabel_stopword");
$stopword = array();
//menyimpan stopwords yang dipanggil ke variabel array
while($result = mysql_fetch_array($sql)){
    $stopword[] = $result['data_stopword'];
}
//tokenizing
$exploded1=explode(" ", $str);
$str = str_replace(' ', '-', $exploded1);
$str = preg_replace('/[^A-Za-z0-9\ - ]/', ' ', $str);
$str = str_replace('-', ' ', $str);
//menyimpan kata ke variabel
//dengan melihat kata yang berbeda dengan stopwords
$result_stop = array_diff($str, $stopword);
//menata susunan array
$simplified1=implode(" ", $result_stop);
$exploded2=explode(" ", $simplified1);
?>

```

Gambar 4.7 Gambar Code text preprocessing

Pada gambar diatas merupakan gambar *code* PHP hasil implementasi proses *text preprocessing*. Pada bagian pertama terdapat *code* untuk melakukan koneksi ke *database*, lalu inisiasi teks yang telah di masukkan ke dalam variabel, melakukan perubahan tiap karakter ke huruf kecil, serta menghilangkan karakter spesial, memanggil data stopwords, pemotongan paragraf menjadi kata-kata, lalu menghilangkan *stopword*, hingga penataan ulang *array*.

4.2.1.2 Proses Perhitungan Waktu *Stemming*

Proses perhitungan waktu stemming digunakan untuk menentukan berapa lama dari masing-masing algoritma pengujian dapat menyelesaikan tahapannya.

Perhitungan waktu dilakukan secara terpisah untuk masing-masing algoritma. Berikut merupakan implementasi perhitungan waktu di dalam sistem.

```

$rs_nazief = Array();//inisialisasi variabel
$stampung_nazief = Array();//inisialisasi variabel
$time_start_nazief = microtime(true);//mulai waktu
foreach ($exploded2 as $rs_nazief){
    //menggunakan fungsi stemming Nazief & Adriani
    $stampung_nazief[] = stemming_nazief($rs_nazief);
}
$time_end_nazief = microtime(true);//menghentikan waktu
$time_nazief = $time_end_nazief - $time_start_nazief;//hitung waktu

$rs_idris = Array();//inisialisasi variabel
$stampung_idris = Array();//inisialisasi variabel
$time_start_idris = microtime(true);//mulai waktu
foreach ($exploded2 as $rs_idris){
    //menggunakan fungsi stemming Idris
    $stampung_idris[] = stemming_idris($rs_idris);
}
$time_end_idris = microtime(true);//menghentikan waktu
$time_idris = $time_end_idris - $time_start_idris;//hitung waktu

```

Gambar 4.8 Gambar *Code* perhitungan waktu *stemming*

Pada gambar diatas menunjukkan pada bagian awal ialah melakukan inisiasi variabel yang akan digunakan. Setelah inisiasi variabel perhitungan waktu dimulai, dan perulangan untuk memanggil fungsi *stemming* dilakukan ke tiap kata yang telah dihasilkan *text preprocessing*. Setelah perulangan selesai perhitungan waktu di hentikan dan dihitung waktunya berdasarkan dari waktu mulai hingga waktu berhenti.

4.2.1.3 Proses Menyimpan ke *Database*

Proses menyimpan ke *database* merupakan langkah yang penting, dikarenakan hasil data yang diolah akan disimpan melalui proses ini. Berikut adalah implementasi proses menyimpan ke *database*.

```
//menyimpan informasi umum pengujian ke database
function isi_table_uji($content_asli,$jumlah_kata_inputan,$jumlah_stopword,$time_test,$idadmin,$db){
    $sql = "insert into table_uji(data_input,jumlah_kata_input,jumlah_stopword,waktu_pengujian,admin_idadmin) values
    (:teks_asli,:jumlah_kata_awal,:jumlah_hasil_stopword,:waktu_test,:id_admin)";
    $stmt = $db->prepare($sql);
    $stmt->execute(['teks_asli' => $content_asli,
    'jumlah_kata_awal' => $jumlah_kata_inputan,
    'jumlah_hasil_stopword' => $jumlah_stopword,
    'waktu_test' => $time_test,
    'id_admin' => $idadmin]);
}

//memanggil id table_uji yang baru disimpan untuk uji_algo
function cek_idtableuji($time_test,$db){
    $sql = "select * from table_uji where waktu_pengujian=:time_test order by idtable_uji DESC";
    $stmt = $db->prepare($sql);
    $stmt->execute(['time_test'=> $time_test]);
    $row = $stmt->fetch();
    $id= $row['idtable_uji'];
    return $id;
}

//menyimpan informasi detail pengujian berdasarkan algoritmanya
function isi_uji_algo($id_algo,$durasi,$akurasi,$jumlah_kata_hasil,$data_kata_berhasil,$idtable_uji,$idadmin,$db){
    $sql = "insert into
    uji_algo(id_algo,durasi,akurasi,jumlah_kata_berhasil,data_kata_berhasil,table_uji_idtable_uji,table_uji_admin_idadmin)
    values (:jenis_algo,:waktu,:akurasi,:jumlah_kata,:data_kata,:table_uji,:admin)";
    $stmt = $db->prepare($sql);
    $stmt->execute(['jenis_algo' => $id_algo,
    'waktu' => $durasi,
    'akurasi' => $akurasi,
    'jumlah_kata' => $jumlah_kata_hasil,
    'data_kata' => $data_kata_berhasil,
    'table_uji' => $idtable_uji,
    'admin' => $idadmin]);
}
```

Gambar 4.9 Gambar Code menyimpan ke *database*

Pada tahanan pertama ialah menyimpan informasi umum pengujian ke tabel *database* 'table_uji', lalu periksa id 'table_uji' yang baru di simpan agar penyimpanan informasi khusus pengujian tetap terhubung, lalu menyimpan informasi khusus pengujian pada tabel 'uji_algo'.

4.2.2 Implementasi Algoritma

4.2.2.1 Algoritma Nazief & Adriani

```
//fungsi pencarian akar kata
function stemming_nazief($katal){
    $cekKata = cekKamus($katal);
    if($cekKata == true){ // Cek Kamus
        return $katal; // Jika Ada maka kata tersebut adalah kata dasar
    }else{ //jika tidak ada dalam kamus maka dilakukan stemming
        $stepl = step2($katal);
        return $stepl;
    }
}
}
```

Gambar 4.10 Gambar Code Nazief & Adriani cek database

Gambar diatas merupakan implementasi dari langkah pertama dari algoritma Nazief & Adriani, yaitu melakukan pengecekan ke *database* dari kata yang telah dimasukkan.

```
function step2($kata){
    $first=$kata;
    for ($i = 1; $i <= 3; $i++) { //batas recoding
        $kata = Del_Inflection_Suffixes($kata);
        /* echo $kata . "<br>"; */
        if(cekKamus($kata)){
            $i+3; //menghentikan perulangan
            return $kata;
        }else{
            $kata = Del_Derivation_Suffixes($kata);
            /* echo $kata . "<br>"; */
            if(cekKamus($kata)){
                $i+3; //menghentikan perulangan
                return $kata;
            }else{
                $cekkata = Cek_Prefix_Disallowed_Sufixes($kata);
                /* echo $kata . "<br>"; */
                if($cekkata==true){
                    $i+3; //menghentikan perulangan
                    return $kata;
                }else{
                    $kata = Del_Derivation_Prefix($kata);
                    /* echo $kata . "<br>"; */
                    if(cekKamus($kata)){
                        $i+3; //menghentikan perulangan
                        return $kata;
                    }
                }
            }
        }
    }
    return $first;
}
```

Gambar 4.11 Gambar Code Nazief & Adriani langkah-langkah stemming

Selanjutnya menyusun tahapan-tahapan yang akan dilakukan jika kata yang dimasukkan tidak terdapat di *database*. Dimulai dari menghilangkan infleksi akhiran, turunan akhiran, mengecek kombinasi imbuhan yang tidak diizinkan, lalu menghilangkan turunan awalan.

```
//fungsi untuk menghapus suffix seperti -ku, -mu, -kah, dsb
function Del_Inflection_Suffixes($kata){
    $kataAsal = $kata;

    if(preg_match('/([km]u|nya|[kl]ah|pun)\z/i',$kata)){ // Cek Inflection Suffixes
        $__kata = preg_replace('/([km]u|nya|[kl]ah|pun)\z/i','', $kata);

        return $__kata;
    }
    return $kataAsal;
}
```

Gambar 4.12 Gambar *Code Nazief & Adriani* infleksi akhiran

Gambar diatas merupakan implementasi untuk menghilangkan infleksi akhiran, seperti -ku, -mu, -nya, -kah, -lah, -pun. Jika infleksi akhiran telah dihilangkan kata dikembalikan untuk dilakukan langkah selanjutnya.

```
// Hapus Derivation Suffixes ("-i", "-an" atau "-kan")
function Del_Derivation_Suffixes($kata){
    $kataAsal = $kata;
    if(preg_match('/(i|an)\z/i',$kata)){ // Cek Suffixes
        $__kata = preg_replace('/(i|an)\z/i','', $kata);
        if(cekKamus($__kata)){ // Cek Kamus
            return $__kata;
        }else if(preg_match('/(kan)\z/i',$kata)){
            $__kata = preg_replace('/(kan)\z/i','', $kata);
            if(cekKamus($__kata)){
                return $__kata;
            }
        }
    }
    /*- Jika Tidak ditemukan di kamus -*/
    }
    return $kataAsal;
}
```

Gambar 4.13 Gambar *Code Nazief & Adriani* turunan akhiran

Pada gambar diatas menampilkan implementasi langkah untuk menghilangkan turunan akhiran, seperti -i, -an, -kan. Jika sudah dihilangkan kata akan di cek pada *database* jika tidak ditemukan kata dikembalikan seperti sebelum

terjadinya penghapusan terhadap turunan akhiran dan di kembalikan untuk di proses ke langkah selanjutnya.

```
// Cek Prefix Disallowed Sufixes (Kombinasi Awalan dan Akhiran yang tidak diizinkan)
function Cek_Prefix_Disallowed_Sufixes($kata){

    if(preg_match('/^(be)[[:alpha:]](i)\z/i',$kata)){ // be- dan -i
        return true;
    }

    if(preg_match('/^(se)[[:alpha:]](i|kan)\z/i',$kata)){ // se- dan -i,-kan
        return true;
    }

    if(preg_match('/^(di)[[:alpha:]](an)\z/i',$kata)){ // di- dan -an
        return true;
    }

    if(preg_match('/^(me)[[:alpha:]](an)\z/i',$kata)){ // me- dan -an
        return true;
    }

    if(preg_match('/^(ke)[[:alpha:]](i|kan)\z/i',$kata)){ // ke- dan -i,-kan
        return true;
    }
    return false;
}
```

Gambar 4.14 Gambar Code Nazief & Adriani Imbuhan tidak diizinkan

Pada gambar diatas menunjukkan hasil implementasi pengecekan terhadap imbuhan yang tidak diizinkan. Jika kata yang diproses termasuk dalam imbuhan yang tidak diizinkan, maka proses *stemming* akan berakhir untuk kata tersebut. Namun jika tidak terdapat pada daftar tersebut maka kata dikembalikan untuk diproses ke tahapan berikutnya.

```

// Hapus Derivation Prefix ("di-", "ke-", "se-", "te-", "be-", "me-", atau "pe-")
function Del_Derivation_Prefix($kata) {
    $kataAsal = $kata;
    //for($j=1; $j<=3;$j++){
    /* — Tentukan Tipe Awalan — */
    if(preg_match('/^(di|[ks]e)/', $kata)) { // Jika di-, ke-, se-
        $__kata = preg_replace('/^(di|[ks]e)/', '', $kata);
        if(cekKamus($__kata)) {return $__kata;}
        $__kata = Del_Derivation_Suffixes($__kata); //recoding
        if(cekKamus($__kata)) {
            return $__kata;
        } else {
            if(preg_match('/^(diper)/', $kata)) { //diper-
                $__kata = preg_replace('/^(diper)/', '', $kata);
                return $__kata;
            }
        }
    }

    if(preg_match('/^(ke[bt]er)/', $kata)) { //keber- dan keter-
        $__kata = preg_replace('/^(ke[bt]er)/', '', $kata);
        return $__kata;
    }

    if(preg_match('/^(bt[e])/', $kata)) { //Jika awalnya adalah "te-", "ter-", "be-", "ber-"

        $__kata = preg_replace('/^(bt[e]lr)/', '', $kata);
        if(cekKamus($__kata)) {return $__kata;}
        $__kata = Del_Derivation_Suffixes($__kata); //recoding
        if(cekKamus($__kata)) {
            return $__kata;
        } else {
            $__kata = preg_replace('/^(bt[e])/', '', $kata);
            return $__kata;
        }
    }

    if(preg_match('/^([mp]e)/', $kata)) {
        $__kata = preg_replace('/^([mp]e)/', '', $kata);
        if(cekKamus($__kata)) {return $__kata;}
        $__kata = Del_Derivation_Suffixes($__kata); //recoding
        if(cekKamus($__kata)) {
            return $__kata;
        } else {
            if(preg_match('/^(memper)/', $kata)) {
                $__kata = preg_replace('/^(memper)/', '', $kata);
                return $__kata;
            }
            if(preg_match('/^([mp]eng)/', $kata)) {
                $__kata = preg_replace('/^([mp]eng)/', '', $kata);
                if(cekKamus($__kata)) {return $__kata;}
                $__kata = Del_Derivation_Suffixes($__kata); //recoding
                if(cekKamus($__kata)) {
                    return $__kata;
                } else {
                    $__kata = preg_replace('/^([mp]eng)/', 'k', $kata);
                    return $__kata;
                }
            }
            if(preg_match('/^([mp]eny)/', $kata)) {
                $__kata = preg_replace('/^([mp]eny)/', 's', $kata);
                return $__kata;
            }
            if(preg_match('/^([mp]e[lr])/', $kata)) {
                $__kata = preg_replace('/^([mp]e[lr])/', '', $kata);
                return $__kata;
            }
            if(preg_match('/^([mp]en)/', $kata)) {
                $__kata = preg_replace('/^([mp]en)/', 't', $kata);
                if(cekKamus($__kata)) {return $__kata;}
                $__kata = Del_Derivation_Suffixes($__kata); //recoding
                if(cekKamus($__kata)) {
                    return $__kata;
                } else {
                    $__kata = preg_replace('/^([mp]en)/', '', $kata);
                    return $__kata;
                }
            }
            if(preg_match('/^([mp]em)/', $kata)) {
                $__kata = preg_replace('/^([mp]em)/', 'p', $kata);
                if(cekKamus($__kata)) {return $__kata;}
                $__kata = Del_Derivation_Suffixes($__kata);
                if(cekKamus($__kata)) {
                    return $__kata;
                } else {
                    $__kata = preg_replace('/^([mp]em)/', '', $kata);
                    return $__kata;
                }
            }
        }
    }
    return $kataAsal;
}

```

Gambar 4.15 Gambar Code Nazief & Adriani turunan awalan

Berikut merupakan hasil implementasi untuk menghilangkan turunan awalan. Pada bagian ini akan dilakukan pencocokan awalan dengan kata yang dimasukkan, jika terdapat kecocokan awalan maka akan dihapus dan dilakukan cek *database*. Setelah semua proses dilakukan dan tidak ditemukan, maka kata awal sebelum di proses *stemming* dianggap sebagai kata dasar.

4.2.2.2 Algoritma Idris

```
//fungsi pencarian akar kata
function stemming_idris($katal){
    $kataAsal = $katal;
    $cekKata = cekKamus_idris($katal);
    if($cekKata == true){ // Cek Kamus
        return $katal; // Jika Ada maka kata tersebut adalah kata dasar
    }else{ //jika tidak ada dalam kamus maka dilakukan stemming
        $step1_idris = step2_idris($katal);
        return $step1_idris;
    }
}
?>
```

Gambar 4.16 Gambar Code Idris cek *database*

Gambar diatas merupakan implementasi dari langkah pertama dari algoritma Idris, yaitu melakukan pengecekan ke *database* dari kata yang telah dimasukkan.

```

function step2_idris($kata){
    $first=$kata;
    $kata = Cek_Prefix_Disallowed_Sufixes_idris($kata);
    if(cekKamus_idris($kata)){
        return $kata;
    }else{
        $kata = Del_Derivation_Prefix_idris($kata);
        if(cekKamus_idris($kata)){
            return $kata;
        }else{
            $kata = Del_Inflection_Suffixes_idris($kata);
            if(cekKamus_idris($kata)){
                return $kata;
            }else{
                $kata = Del_Derivation_Suffixes_idris($kata);
                if(cekKamus_idris($kata)){
                    return $kata;
                }else{return $first;}
            }
        }
    }
}

```

Gambar 4.17 Gambar *Code Idris* langkah-langkah *stemming*

Selanjutnya menyusun tahapan-tahapan yang akan dilakukan jika kata yang dimasukkan tidak terdapat di *database*. Dimulai dari mengecek kombinasi imbuhan yang tidak diizinkan, menghilangkan turunan awalan, menghilangkan infleksi akhiran, dan menghilangkan turunan akhiran.

```

// Cek Prefix Disallowed Sufixes (Kombinasi Awalan dan Akhiran yang tidak diizinkan)
function Cek_Prefix_Disallowed_Sufixes_idris($kata){
    $_katal = $kata;
    if(preg_match('/^(ber)[[:alpha:]](lah|an)\z/i',$kata)){ // ber- dan -lah,-an
        $_kata2 = preg_replace('/^(ber)/','',$kata);
        return $_kata2;
    }

    if(preg_match('/^(men|di|pe|ter)[[:alpha:]](i)\z/i',$kata)){ // men-,di-,pe-,ter- dan -i
        $_kata2 = preg_replace('/^(men|di|pe|ter)/','',$kata);
        return $_kata2;
    }
    return $_katal;
}

```

Gambar 4.18 Gambar *Code Idris* imbuhan tidak diizinkan

Pada gambar diatas menunjukkan hasil implementasi pengecekan terhadap imbuhan yang tidak diizinkan. Jika kata yang diproses termasuk dalam imbuhan yang tidak diizinkan, maka awalan akan di hilangkan dan dikembalikan untuk

dilanjutkan ke tahap selanjutnya. Namun jika tidak terdapat pada daftar tersebut maka kata dikembalikan untuk diproses ke tahapan berikutnya.

```
function Del_Derivation_Prefix_idris($kata){
    $kataAsal = $kata;

    /* — Tentukan Tipe Awalan — */

    if(preg_match('/^(diper)/',$kata){ //diper-
        $__kata__ = preg_replace('/^(diper)/','',$kata);
        return $__kata__;
    }

    if(preg_match('/^(di|ks)e)/',$kata){ // Jika di-,ke-,se-
        $__kata__ = preg_replace('/^(di|ks)e)/','',$kata);
        return $__kata__;
    }

    if(preg_match('/^(ke|bt|er)/',$kata){ //keber- dan keter-
        $__kata__ = preg_replace('/^(ke|bt|er)/','',$kata);
        return $__kata__;
    }

    if(preg_match('/^([bt]e)/',$kata){ //Jika awalnya adalah "te-","ter-","be-","ber-"
        $__kata__ = preg_replace('/^([bt]e[lr])/','',$kata);
        if(cekKamus_idris($__kata__)){
            return $__kata__;
        }

        $__kata__ = preg_replace('/^([bt]e)/','',$kata);
        return $__kata__;
    }

    if(preg_match('/^([mp]e)/',$kata){

        $__kata__ = preg_replace('/^([mp]e)/','',$kata);
        if(cekKamus_idris($__kata__)){
            return $__kata__;
        }else{
            if(preg_match('/^(memper)/',$kata){
                $__kata__ = preg_replace('/^(memper)/','',$kata);
                if(cekKamus_idris($__kata__)){
                    return $__kata__;
                }
            }
            if(preg_match('/^([mp]eng)/',$kata){
                $__kata__ = preg_replace('/^([mp]eng)/','',$kata);
                if(cekKamus_idris($__kata__)){
                    return $__kata__;
                }else{
                    $__kata__ = preg_replace('/^([mp]eng)/','k',$kata);
                }
            }
            if(preg_match('/^([mp]eny)/',$kata){
                $__kata__ = preg_replace('/^([mp]eny)/','s',$kata);
            }
            if(preg_match('/^([mp]e[lr])/',$kata){
                $__kata__ = preg_replace('/^([mp]e[lr])/','',$kata);
            }
            if(preg_match('/^([mp]en)/',$kata){
                $__kata__ = preg_replace('/^([mp]en)/','t',$kata);
                if(cekKamus_idris($__kata__)){
                    return $__kata__;
                }else{
                    $__kata__ = preg_replace('/^([mp]en)/','',$kata);
                }
            }
            if(preg_match('/^([mp]em)/',$kata){
                $__kata__ = preg_replace('/^([mp]em)/','',$kata);
                if(cekKamus_idris($__kata__)){
                    return $__kata__;
                }else{
                    $__kata__ = preg_replace('/^([mp]em)/','p',$kata);
                }//elsempem
            }//end mpem if
            return $__kata__; }//else mpe
        }//end mpe if
    }
    return $kataAsal;
}
```

Gambar 4.19 Gambar Code Idris turunan awalan

Berikut merupakan hasil implementasi untuk menghilangkan turunan awalan. Pada bagian ini akan dilakukan pencocokan awalan dengan kata yang dimasukkan, jika terdapat kecocokan awalan maka akan dihapus lalu dikembalikan untuk dilanjutkan ke tahapan selanjutnya.

```
//fungsi untuk menghapus suffix seperti -ku, -mu, -kah, dsb
function Del_Inflection_Suffixes_idris($kata){
    $kataAsal = $kata;
    //if(preg_match('/([km]u|nya|[kl]ah|pun)\z/i',$kata)){ // Cek Inflection Suffixes
    if(preg_match('/([kl]ah|pun)\z/i',$kata)){ // Cek Inflection Suffixes
        $__kata = preg_replace('/([kl]ah|pun)\z/i','', $kata);
        if(cekKamus_idris($__kata)){
            return $__kata;
        }else{
            $kata = $__kata;
        }
    }

    if(preg_match('/([km]u|nya)\z/i',$kata)){ // Cek Inflection Suffixes
        $__kata = preg_replace('/([km]u|nya)\z/i','', $kata);
        if(cekKamus_idris($__kata)){
            return $__kata;
        }
    }
    return $kataAsal;
}
```

Gambar 4.20 Gambar Code Idris infleksi akhiran

Gambar diatas merupakan implementasi untuk menghilangkan infleksi akhiran, seperti -ku, -mu, -nya, -kah, -lah, -pun. Jika infleksi akhiran telah dihilangkan kata dikembalikan untuk dilakukan langkah selanjutnya.

```
// Hapus Derivation Suffixes ("-i", "-an" atau "-kan")
function Del_Derivation_Suffixes_idris($kata){
    $kataAsal = $kata;
    if(preg_match('/(i|an)\z/i',$kata)){ // Cek Suffixes
        $__kata = preg_replace('/(i|an)\z/i','', $kata);
        if(cekKamus_idris($__kata)){ // Cek Kamus
            return $__kata;
        }
    } else if(preg_match('/(kan)\z/i',$kata)){
        $__kata = preg_replace('/(kan)\z/i','', $kata);
        if(cekKamus_idris($__kata)){
            return $__kata;
        } else{
            $__kata = Del_Derivation_Prefix_idris($__kata); //recoding
            return $__kata;
        }
    }
    /*- Jika Tidak ditemukan di kamus -*/
}
}
```

Gambar 4.21 Gambar Code Idris turunan akhiran

Pada gambar diatas menampilkan implementasi langkah untuk menghilangkan turunan akhiran, seperti -i, -an, -kan. Jika sudah dihilangkan kata akan di cek pada *database* jika tidak ditemukan kata semula akan ditetapkan menjadi kata dasar.

4.3 Pengujian Keberhasilan Metode

Pengujian dilakukan dengan menganalisis nilai yang dihasilkan berupa durasi pemrosesan dan akurasi yang didapat dari memasukkan jumlah kata dengan hasil *stemming* yang tidak sesuai. Skenario pengujian yang dilakukan adalah melakukan pengujian sebanyak 5 kali dengan teks yang berbeda-beda. Lalu menghasilkan keluaran yang berupa teks hasil *stemming*, dilanjutkan dengan menghitung jumlah kata hasil *stemming* yang tidak sesuai dari masing-masing algoritma.

Pengujian Pertama

Berikut merupakan informasi mengenai teks yang akan menjadi pengujian pertama.

Nama teks : Legenda Danau Toba.

Jumlah kata : 520 kata.

Dokumentasi pengujian teks pertama dapat dilihat pada gambar 4.22 hingga gambar 4.26.

Perbandingan Algoritma



Pengujian

Di wilayah Sumatera hiduplah seorang petani yang sangat rajin bekerja. Ia hidup sendiri sebatang kara. Setiap hari ia bekerja menggarap ladang dan mencari ikan dengan tidak mengenal lelah. Hal ini dilakukannya untuk memenuhi kebutuhannya sehari-hari.

Pada suatu hari petani tersebut pergi ke sungai di dekat tempat tinggalnya, ia bermaksud mencari ikan untuk lauknya hari ini. Dengan hanya bekal sebuah kail, umpan dan tempat ikan, ia pun langsung menuju ke sungai. Setelah sesampainya di sungai, petani tersebut langsung melemparkan kailnya. Sambil menunggu kailnya dimakan ikan, petani tersebut berdoa, "Ya Allah, semoga aku dapat ikan banyak hari ini". Beberapa saat setelah berdoa, kail yang dilemparkannya tadi nampak bergoyang-goyang. Ia segera menarik kailnya. Petani tersebut sangat senang sekali, karena ikan yang didapatkannya sangat besar dan cantik sekali.

Setelah beberapa saat memandangi ikan hasil tangkapannya, petani itu sangat terkejut. Ternyata ikan yang ditangkapnya itu bisa berbicara. "Tolong aku jangan dimakan Pak!! Biarkan aku hidup", teriak ikan itu. Tanpa banyak tanya, ikan tangkapannya itu langsung dikembalikan ke dalam air lagi. Setelah mengembalikan ikan ke dalam air, petani itu bertambah terkejut, karena tiba-tiba ikan tersebut berubah menjadi seorang wanita yang sangat cantik.

"Jangan takut Pak, aku tidak akan menyakitimu", kata si ikan. "Siapa kamu ini? Bukankah kamu seekor ikan?". Tanya petani itu. "Aku adalah seorang putri yang dikutuk, karena melanggar aturan kerajaan", jawab wanita itu. "Terimakasih engkau sudah membebaskan aku dari kutukan itu, dan sebagai imbalannya aku bersedia kau jadikan istri", kata wanita itu. Petani itupun setuju. Maka jadilah mereka sebagai suami istri. Namun, ada satu janji yang telah disepakati, yaitu mereka tidak boleh menceritakan bahwa asal-usul Putri dari seekor ikan. Jika janji itu dilanggar maka akan terjadi petaka dahsyat.

Setelah beberapa lama mereka menikah, akhirnya kebahagiaan Petani dan istrinya bertambah, karena istri Petani melahirkan seorang bayi laki-laki. Anak mereka tumbuh menjadi anak yang sangat tampan dan kuat, tetapi ada kebiasaan yang membuat heran semua orang. Anak tersebut selalu merasa lapar, dan tidak pernah merasa kenyang. Semua jatah makanan dilahapnya tanpa sisa.

Hingga suatu hari anak petani tersebut mendapat tugas dari ibunya untuk mengantarkan makanan dan minuman ke sawah di mana ayahnya sedang bekerja. Tetapi tugasnya tidak dipenuhinya. Semua makanan yang seharusnya untuk ayahnya dilahap habis, dan setelah itu dia tertidur di sebuah gubug. Pak tani menunggu kedatangan anaknya, sambil menahan haus dan lapar. Karena tidak tahan menahan lapar, maka ia langsung pulang ke rumah. Di tengah perjalanan pulang, pak tani melihat anaknya sedang tidur di gubug. Petani tersebut langsung membangunkannya. "Hey, bangun!, teriak petani itu.

Setelah anaknya terbangun, petani itu langsung menanyakan makanannya. "Mana makanan buat ayah?", Tanya petani. "Sudah habis kumakan", jawab si anak. Dengan nada tinggi petani itu langsung memarahi anaknya. "Anak tidak tau diuntung! Tak tahu diri! Dasar anak ikan!," umpat si Petani tanpa sadar telah mengucapkan kata pantangan dari istrinya.

Setelah petani mengucapkan kata-kata tersebut, seketika itu juga anak dan istrinya hilang lenyap tanpa bekas dan jejak. Dari bekas injakan kakinya, tiba-tiba menyemburlah air yang sangat deras. Air meluap sangat tinggi dan luas sehingga membentuk sebuah telaga. Dan akhirnya membentuk sebuah danau. Danau itu akhirnya dikenal dengan nama Danau Toba.

Gambar 4.22 Gambar Pengujian 1 masukkan teks pengujian

Verifikasi Hasil

Waktu pengujian : 06-11-17 16:43:58
 Jumlah Kata Inputan : 520
 Jumlah Stopword : 265

Di wilayah Sumatera hiduplah seorang petani yang sangat rajin bekerja. Ia hidup sendiri sebatang kara. Setiap hari ia bekerja menggarap lading dan mencari ikan dengan tidak mengenal lelah. Hal ini dilakukannya untuk memenuhi kebutuhannya sehari-hari.

Pada suatu hari petani tersebut pergi ke sungai di dekat tempat tinggalnya, ia bermaksud mencari ikan untuk lauknya hari ini. Dengan hanya berbekal sebuah kail, umpan dan tempat ikan, ia pun langsung menuju ke sungai. Setelah sesampainya di sungai, petani tersebut langsung melemparkan kailnya. Sambil menunggu kailnya dimakan ikan, petani tersebut berdoa, "Ya Allah, semoga aku dapat ikan banyak hari ini". Beberapa saat setelah berdoa, kail yang dilemparkannya tadi nampak bergoyang-goyang. Ia segera menarik

Gambar 4.23 Gambar Pengujian 1 informasi umum pengujian

Hasil Stemming Nazief & Adriani

Durasi : 7.8000290393829 detik
 Jumlah Kata Hasil 255

wilayah sumatera hidup petan rajin hidup batang kara garap lading cari ikan kenal lelah laku penuh butuh hari petan pergi sungai tinggal cari ikan lauk bekal kail umpan ikan langsung sungai sampai sungai petan langsung lempar kail tunggu kail makan ikan petan doa ya allah moga ikan doa kail lempar nampak goyang goyang tarik kail petan senang ikan dapat cantik pandang ikan hasil tangkap petan kejut ikan tangkap bicara tolong makan biar hidup teriak ikan ikan tangkap langsung kembali air kembali ikan air petan tambah kejut ikan ubah wanita cantik takut sakit si ikan ekor ikan petan putri kutuk langgar atur raja wanita terimakasih engkau bebas kutuk imbal sedia kau jadi istri wanita petan itu tuju suami istri janji sepakat cerita usul puter ekor ikan janji langgar petaka dahsyat meni bahagia petan istri tambah istri petan lahir bayi laki laki anak tumbuh anak tampan kuat biasa heran orang anak lapar kenyang jatah makan lahap sisa anak petan tugas ibu antar makan minum sawah ayah tugas penuh makan ayah lahap habis tidur gubug tani tunggu datang anak nah haus lapar tahan nah lapar langsung pulang rumah jalan pulang tani anak tidur gubug petan langsung bangun hey bangun teriak petan anak bangun petan langsung makan makan ayah petan habis kumakan si anak nada petan langsung marah anak anak tau untung dasar anak ikan umpat si petan sadar pantang istri petan anak istri hilang lenyap bekas jejak bekas injak kaki sembur air deras air luap luas bentuk telaga bentuk danau danau kenal nama danau toba

Jumlah Kata Salah : 19

Gambar 4.24 Gambar Pengujian 1 informasi Nazief & Adriani

Hasil Stemming Idris

Durasi : 7.9879069328308 detik

Jumlah Kata Hasil 255

wilayah sumatera hidup tani rajin hidup batang kara garap lading cari ikan mengenal lelah dilakukannya penuh kebutuhannya hari tani pergi sungai tinggal cari ikan lauk bekal kail umpan ikan langsung sungai sampai sungai tani langsung empur kail tunggu kail makan ikan tani doa ya allah moga ikan doa kail dilemparkannya nampak goyang goyang tarik kail tani senang ikan didapatkannya cantik pandang ikan hasil tangkapannya tani kejut ikan tangkap bicara tolong makan biar hidup teriak ikan ikan tangkapannya langsung kembali air mengembalikan ikan air tani tambah kejut ikan ubah wanita cantik takut menyakiti si ikan ekor ikan tani putri kutuk langgar atur raja wanita terimakasih engkau membebaskan kutuk imbalannya sedia kau jadi istri wanita tani itu tuju suami istri janji sepakat cerita usul puter ekor ikan janji langgar petaka dahsyat nikah bahagia tani istri tambah istri tani melahirkan bayi laki laki anak tumbuh anak tampan kuat biasa heran orang anak lapar kenyang jatah makan lahap sisa anak tani tugas ibu antar makan minum sawah ayah tugas dipenuhinya makan ayah lahap habis tidur gubug tani tunggu datang anak tahan haus lapar tahan tahan lapar langsung pulang rumah jalan pulang tani anak tidur gubug tani langsung membangunkannya hey bangun teriak tani anak bangun tani langsung makanannya makan ayah tani habis kumakan si anak nada tani langsung parah anak anak tau untung dasar anak ikan umpat si tani sadar pantang istri tani anak istri hilang lenyap bekas jejak bekas injak kaki menyemburlah air deras air luap luas bentuk telaga bentuk danau danau kenal nama danau toba

Jumlah Kata Salah : 19

Gambar 4.25 Gambar Pengujian 1 informasi Idris

No.	Waktu Pengujian	Jumlah Kata	Algoritma	Durasi (detik)	Akurasi (%)
1	2017-11-06 16:43:58	255	Nazief & Adirani	7.80003	92.549
2	2017-11-06 16:43:58	255	Idris	7.98791	92.549

Gambar 4.26 Gambar Pengujian 1 rekap pengujian

Pada pengujian pertama di gambar 4.22 melakukan masukkan teks pada *text area* yang disediakan, lalu pada 4.23 menampilkan informasi umum mengenai pengujian seperti waktu pengujian yang dilakukan pada 06 November 2017 pukul 16:43:58 WIB, jumlah kata masukkan 520 kata, jumlah *stopword* sebanyak 265 kata, hingga teks awal masukkan.

Pada gambar 4.24 merupakan tampilan informasi yang dihasilkan dari proses *stemming* Nazief & Adirani seperti durasi pemrosesan berlangsung selama 7,8 detik, jumlah kata yang dihasilkan 255 kata, data kata hasil *stemming*, serta hasil perhitungan jumlah kata yang dihasilkan salah sebanyak 19 kata.

Pada gambar 4.25 merupakan tampilan informasi yang dihasilkan dari proses *stemming* Idris seperti durasi pemrosesan berlangsung selama 7,99 detik,

jumlah kata yang dihasilkan 255 kata, data kata hasil *stemming*, serta hasil perhitungan jumlah kata yang dihasilkan salah sebanyak 19 kata.

Setelah dilakukan pengujian, pada gambar 4.26 menampilkan laman detail rekap mengenai pengujian yang telah dilakukan. Hasil yang di dapat berupa informasi jumlah kata, jenis algoritma yang digunakan, durasi, dan akurasi.

Pengujian Kedua

Berikut merupakan informasi mengenai teks yang akan menjadi pengujian kedua.

Nama teks : Legenda Sangkuriang.

Jumlah kata : 645 kata.

Dokumentasi pengujian teks kedua dapat dilihat pada gambar 4.27 hingga gambar 4.31.

Perbandingan Algoritma

☰

Pengujian

Pada suatu hari, seperti biasanya Sangkuriang pergi ke hutan untuk berburu. Setelah sesampainya di hutan, Sangkuriang mulai mencari buruan. Dia melihat ada seekor burung yang sedang bertengger di dahan, lalu tanpa berpikir panjang Sangkuriang langsung menembaknya, dan tepat mengenai sasaran. Sangkuriang lalu memerintah Tumang untuk mengejar buruannya tadi, tetapi si Tumang diam saja dan tidak mau mengikuti perintah Sangkuriang. Karena sangat jengkel pada Tumang, maka Sangkuriang lalu mengusir Tumang dan tidak diijinkan pulang ke rumah bersamanya lagi.

Sesampainya di rumah, Sangkuriang menceritakan kejadian tersebut kepada ibunya. Begitu mendengar cerita dari anaknya, Dayang Sumbi sangat marah. Diambilnya sendok nasi, dan dipukulkan ke kepala Sangkuriang. Karena merasa kecewa dengan perlakuan ibunya, maka Sangkuriang memutuskan untuk pergi mengembara, dan meninggalkan rumahnya.

Setelah kejadian itu, Dayang Sumbi sangat menyesali perbuatannya. Ia berdoa setiap hari, dan meminta agar suatu hari dapat bertemu dengan anaknya kembali. Karena kesungguhan dari doa Dayang Sumbi tersebut, maka Dewa memberinya sebuah hadiah berupa kecantikan abadi dan usia muda selamanya.

Setelah bertahun-tahun lamanya Sangkuriang mengembara, akhirnya ia berniat untuk pulang ke kampung halamannya. Sesampainya di sana, dia sangat terkejut sekali, karena kampung halamannya sudah berubah total. Rasa senang Sangkuriang tersebut bertambah ketika saat di tengah jalan bertemu dengan seorang wanita yang sangat cantik jelita, yang tidak lain adalah Dayang Sumbi. Karena terpesona dengan kecantikan wanita tersebut, maka Sangkuriang langsung melamarnya. Akhirnya lamaran Sangkuriang diterima oleh Dayang Sumbi, dan sepakat akan menikah di waktu dekat.

Pada suatu hari, Sangkuriang meminta izin calon istrinya untuk berburu di hutan. Sebelum berangkat, ia meminta Dayang Sumbi untuk mengcangkam dan merapikan ikat kapalnya. Alangkah terkejutnya Dayang Sumbi, karena pada saat dia merapikan ikat kepala Sangkuriang, ia melihat ada bekas luka. Bekas luka tersebut mirip dengan bekas luka anaknya. Setelah bertanya kepada Sangkuriang tentang penyebab lukanya itu, Dayang Sumbi bertambah tekejut, karena ternyata benar bahwa calon suaminya tersebut adalah anaknya sendiri.

Dayang Sumbi sangat bingung sekali, karena dia tidak mungkin menikah dengan anaknya sendiri. Setelah Sangkuriang pulang berburu, Dayang Sumbi mencoba berbicara kepada Sangkuriang, supaya Sangkuriang membatalkan rencana pernikahan mereka. Permintaan Dayang Sumbi tersebut tidak disetujui Sangkuriang, dan hanya dianggap angin lalu saja.

Setiap hari Dayang Sumbi berpikir bagaimana cara agar pernikahan mereka tidak pernah terjadi. Setelah berpikir keras, akhirnya Dayang Sumbi menemukan cara terbaik. Dia mengajukan dua buah syarat kepada Sangkuriang. Apabila Sangkuriang dapat memenuhi kedua syarat tersebut, maka Dayang Sumbi mau dijadikan istri, tetapi sebaliknya jika gagal maka pernikahan itu akan dibatalkan.

Syarat yang pertama Dayang Sumbi ingin supaya sungai Citarum dibendung. Dan yang kedua adalah, meminta Sangkuriang untuk membuat sampan yang sangat besar untuk menyeberang sungai. Kedua syarat itu harus diselesaikan sebelum fajar menyingsing.

Sangkuriang menanggapi kedua permintaan Dayang Sumbi tersebut, dan berjanji akan menyelesaikannya sebelum fajar menyingsing. Dengan kesaktian yang dimilikinya, Sangkuriang lalu mengerahkan teman-temannya dari bangsa jin untuk membantu menyelesaikan tugasnya tersebut. Diam-diam, Dayang Sumbi mengintip hasil kerja dari Sangkuriang. Betapa terkejutnya dia, karena Sangkuriang hampir menyelesaikan semua syarat yang diberikan Dayang Sumbi sebelum fajar.

Dayang Sumbi lalu meminta bantuan masyarakat sekitar untuk menggelar kain sutera berwarna merah di sebelah timur kota. Ketika melihat warna merah di timur kota, Sangkuriang mengira kalau hari sudah menjelang pagi. Sangkuriang langsung menghentikan pekerjaannya dan merasa tidak dapat memenuhi syarat yang telah diajukan oleh Dayang Sumbi.

Gambar 4.27 Gambar Pengujian 2 masukkan teks pengujian

Verifikasi Hasil

Waktu pengujian : 06-11-17 16:59:57

Jumlah Kata Inputan : 645

Jumlah Stopword : 304

Pada jaman dahulu, di Jawa Barat hiduplah seorang putri raja yang bernama Dayang Sumbi. Ia mempunyai seorang anak laki-laki yang bernama Sangkuriang. Anak tersebut sangat gemar berburu di dalam hutan. Setiap berburu, dia selalu ditemani oleh seekor anjing kesayangannya yang bernama Tumang. Tumang sebenarnya adalah titisan dewa, dan juga bapak kandung Sangkuriang, tetapi Sangkuriang tidak tahu hal itu dan ibunya memang sengaja merahasiakannya.

Pada suatu hari, seperti biasanya Sangkuriang pergi ke hutan untuk berburu. Setelah sesampainya di hutan, Sangkuriang mulai mencari buruan. Dia melihat ada seekor burung yang sedang bertengger di dahan, lalu tanpa berpikir panjang Sangkuriang langsung

Gambar 4.28 Gambar Pengujian 2 informasi umum pengujian

Hasil Stemming Nazief & Adriani

Durasi : 15.806177854538 detik

Jumlah Kata Hasil 341

jam jawa barat hidup putri raja nama dayang sumbi anak laki laki nama sangkuriang anak gemar buru hutan buru teman ekor anjing sayang nama tumang tumang titis dewa kandung sangkuriang sangkuriang ibu sengaja rahasia sangkuriang pergi hutan buru sampai hutan sangkuriang cari buru ekor burung tengger dahan pikir sangkuriang langsung tembak sasar sangkuriang perintah tumang kejar buru si tumang diam ikut perintah sangkuriang jengkel tumang sangkuriang usir tumang diijinkan pulang rumah sama sampai rumah sangkuriang cerita jadi ibu dengar cerita anak dayang sumbi marah ambil sendok nasi pukul kepala sangkuriang kecewa laku ibu sangkuriang putus pergi embara tinggal rumah jadi dayang sumbi sesal buat doa bertemu anak sungguh doa dayang sumbi dewa beri hadiah cantik abadi usia muda tahun sangkuriang embara niat pulang kampung halaman sampai kejut kampung halaman ubah total senang sangkuriang tambah jalan bertemu wanita cantik jelita dayang sumbi pesona cantik wanita sangkuriang langsung lamar lamar sangkuriang terima dayang sumbi sepakat meni sangkuriang ijin calon istri buru hatan berangkat dayang sumbi encang rapik ikat kapalanya alangkah kejut dayang sumbi rapik ikat kepala sangkuriang bekas luka bekas luka bekas luka anak sangkuriang sebab luka dayang sumbi tambah kejut calon suami anak dayang sumbi bingung meni anak sangkuriang pulang buru dayang sumbi coba bicara sangkuriang sangkuriang batal rencana nikah minta dayang sumbi disetujui sangkuriang anggap angin dayang sumbi pikir nikah pikir keras dayang sumbi temu baik ajuk buah syarat sangkuriang sangkuriang penuh syarat dayang sumbi jadi istri gagal nikah batal syarat dayang sumbi sungai citarum bendung sangkuriang sampan seberang sungai syarat selesai fajar singsing sangkuriang sanggup minta dayang sumbi janji selesai fajar singsing sakti milik sangkuriang kerah teman teman bangsa ijin bantu selesai tugas diam diam dayang sumbi intip hasil kerja sangkuriang betapa kejut sangkuriang menyelesaikan syarat dayang sumbi fajar dayang sumbi bantu masyarakat gelar kain sutera warna merah sebelah timur kota warna merah timur kota sangkuriang jelang pagi sangkuriang langsung henti kerja penuh syarat ajuk dayang sumbi jengkel kecewa sangkuriang jebol bendung jebol bendung banjir kota rendam air sangkuriang tendang sampan layang jatuh telungkup gunung nama tangkuban perahu

Jumlah Kata Salah : 10

Gambar 4.29 Gambar Pengujian 2 informasi Nazief & Adriani

Hasil Stemming Idris

Durasi : 11.097884178162 detik
Jumlah Kata Hasil 341

jam jawa barat hidup putri raja nama dayang sumbi anak laki laki nama sangkuriang anak gemar buru hutan buru teman ekor anjing kesayangannya nama tumang tumang titis dewa kandung sangkuriang sangkuriang ibu sengaja merahasiakannya sangkuriang pergi hutan buru sampai hutan sangkuriang cari buru ekor burung tengger dahan pikir sangkuriang langsung embak sasar sangkuriang perintah tumang mengejar buruannya si tumang diam mengikuti perintah sangkuriang jengkel tumang sangkuriang usir tumang diijinkan pulang rumah bersamanya sampai rumah sangkuriang cerita jadi ibu dengar cerita anak dayang sumbi marah ambil sendok nasi pukul kepala sangkuriang kecewa laku ibu sangkuriang putus pergi embara meninggalkan rumah jadi dayang sumbi menyesali perbuatannya doa temu anak sungguh doa dayang sumbi dewa memberinya hadiah cantik abadi usia muda tahun sangkuriang embara niat pulang kampung halaman sampai kejut kampung halaman ubah total senang sangkuriang tambah jalan temu wanita cantik jelita dayang sumbi pesona cantik wanita sangkuriang langsung amar lamar sangkuriang terima dayang sumbi sepakat nikah sangkuriang ijin calon istri buru hutan berangkat dayang sumbi mengengahkan apik ikat kapalnya alangkah terkejutnya dayang sumbi apik ikat kepala sangkuriang bekas luka bekas luka bekas luka anak sangkuriang penyebab luka dayang sumbi tambah kejut calon suami anak dayang sumbi bingung nikah anak sangkuriang pulang buru dayang sumbi coba bicara sangkuriang sangkuriang membatalkan rencana nikah minta dayang sumbi disetujui sangkuriang anggap angin dayang sumbi pikir nikah pikir keras dayang sumbi menemukan baik mengajukan buah syarat sangkuriang sangkuriang penuh syarat dayang sumbi jadi istri gagal nikah batal syarat dayang sumbi sungai citarum bendung sangkuriang sampan menyeberang sungai syarat selesai fajar menyinggung sangkuriang menyanggupi minta dayang sumbi janji menyelesaikannya fajar menyinggung sakti dimilikinya sangkuriang gerah teman man bangsa jin bantu menyelesaikan tugas diam diam dayang sumbi intip hasil kerja sangkuriang betapa terkejutnya sangkuriang menyelesaiklan syarat dayang sumbi fajar dayang sumbi bantu masyarakat gelar kain sutera warna merah belah timur kota warna merah timur kota sangkuriang jelang pagi sangkuriang langsung menghentikan pekerjaannya penuh syarat ajuk dayang sumbi jengkel kecewa sangkuriang jebol bendungan jebol bendungan banjir kota rendam air sangkuriang tendang sampan sampan layang jatuh telungkup gunung nama tangkuban perahu

Jumlah Kata Salah : 39

Gambar 4.30 Gambar Pengujian 2 informasi Idris

No.	Waktu Pengujian	Jumlah Kata	Algoritma	Durasi (detik)	Akurasi (%)
1	2017-11-06 16:59:57	341	Nazief & Adriani	15.8062	97.0675
2	2017-11-06 16:59:57	341	Idris	11.0979	88.563

Gambar 4.31 Gambar Pengujian 2 rekap pengujian

Pada pengujian kedua di gambar 4.27 melakukan masukkan teks pada *text area* yang disediakan, lalu pada 4.28 menampilkan informasi umum mengenai pengujian seperti waktu pengujian yang dilakukan pada 06 November 2017 pukul 16:59:57 WIB, jumlah kata masukkan 645 kata, jumlah *stopword* sebanyak 304 kata, hingga teks awal masukkan.

Pada gambar 4.29 merupakan tampilan informasi yang dihasilkan dari proses *stemming* Nazief & Adirani seperti durasi pemrosesan berlangsung selama 15,8 detik, jumlah kata yang dihasilkan 341 kata, data kata hasil *stemming*, serta hasil perhitungan jumlah kata yang dihasilkan salah sebanyak 10 kata.

Pada gambar 4.30 merupakan tampilan informasi yang dihasilkan dari proses *stemming* Idris seperti durasi pemrosesan berlangsung selama 11,09 detik,

jumlah kata yang dihasilkan 341 kata, data kata hasil *stemming*, serta hasil perhitungan jumlah kata yang dihasilkan salah sebanyak 39 kata.

Setelah dilakukan pengujian, pada gambar 4.31 menampilkan laman detail rekap mengenai pengujian yang telah dilakukan. Hasil yang di dapat berupa informasi jumlah kata, jenis algoritma yang digunakan, durasi, dan akurasi.

Pengujian Ketiga

Berikut merupakan informasi mengenai teks yang akan menjadi pengujian ketiga.

Nama teks : Cerita Rakyat Timun Mas.

Jumlah kata : 637 kata.

Dokumentasi pengujian teks pertama dapat dilihat pada gambar 4.32 hingga gambar 4.36.

Perbandingan Algoritma

☰

Pengujian

Di suatu desa hiduplah seorang janda tua yang bernama mbok Sarni. Tiap hari dia menghabiskan waktunya sendirian, karena mbok Sarni tidak memiliki seorang anak. Sebenarnya dia ingin sekali mempunyai anak, agar bisa membantunya bekerja.

Pada suatu sore pergilah mbok Sarni ke hutan untuk mencari kayu, dan ditengah jalan mbok Sarni bertemu dengan raksasa yang sangat besar sekali. "Hei, mau kemana kamu?", tanya si Raksasa. "Aku hanya mau mengumpulkan kayu bakar, jadi ijinkanlah aku lewat", jawab mbok Sarni. "Hahahaha.... kamu boleh lewat setelah kamu memberiku seorang anak manusia untuk aku santap", kata si Raksasa. Lalu mbok Sarni menjawab, "Tetapi aku tidak mempunyai anak".

Setelah mbok Sarni mengatakan bahwa dia tidak punya anak dan ingin sekali punya anak, maka si Raksasa memberinya biji mentimun. Raksasa itu berkata, "Wahai wanita tua, ini aku berikan kamu biji mentimun. Tanamlah biji ini di halaman rumahmu, dan setelah dua minggu kamu akan mendapatkan seorang anak. Tetapi ingat, serahkan anak itu padaku setelah usianya enam tahun".

Setelah dua minggu, mentimun itu nampak berbuah sangat lebat dan ada salah satu mentimun yang cukup besar. Mbok Sarni kemudian mengambilnya, dan setelah dibelah ternyata isinya adalah seorang bayi yang sangat cantik jelita. Bayi itu kemudian diberi nama Timun Mas.

Semakin hari Timun Mas semakin tumbuh besar, dan mbok Sarni sangat gembira sekali karena rumahnya tidak sepi lagi. Semua pekerjaannya bisa selesai dengan cepat karena bantuan Timun Mas.

Akhirnya pada suatu hari datanglah si Raksasa untuk menagih janji. Mbok Sarni sangat ketakutan, dan tidak mau kehilangan Timun Mas. Kemudian mbok Sarni berkata, "Wahai raksasa, datanglah kesini dua tahun lagi. Semakin dewasa anak ini, maka semakin enak untuk di santap". Si Raksasa pun setuju dan meninggalkan rumah mbok Sarni.

Waktu dua tahun bukanlah waktu yang lama, karena itu tiap hari mbok Sarni mencari akal bagaimana caranya supaya anaknya tidak dibawa si Raksasa. Hati mbok Sarni sangat cemas sekali, dan akhirnya pada suatu malam mbok Sarni bermimpi. Dalam mimpinya itu, ia diberitahu agar Timun Mas menemui petapa di Gunung.

Pagi harinya mbok Sarni menyuruh Timun Mas untuk segera menemui petapa itu. Setelah bertemu dengan petapa, Timun Mas kemudian bercerita tentang maksud kedatangannya. Sang petapa kemudian memberinya empat buah bungkus kecil yang isinya biji mentimun, jarum, garam, dan terasi. "Lemparkan satu per satu bungkus ini, kalau kamu dikejar oleh raksasa itu", perintah petapa. Kemudian Timun Mas pulang ke rumah, dan langsung menyimpan bungkus dari sang petapa.

Paginya raksasa datang lagi untuk menagih janji. "Wahai wanita tua, mana anak itu? Aku sudah tidak tahan untuk menyantapnya", teriak si Raksasa. Kemudian mbok Sarni menjawab, "Janganlah kau ambil anakku ini wahai raksasa, karena aku sangat sayang padanya. Lebih baik aku saja yang kamu santap". Raksasa tidak mau menerima tawaran dari mbok Sarni itu, dan akhirnya marah besar. "Mana anak itu? Mana Timun Mas?", teriak si raksasa.

Karena tidak tega melihat mbok Sarni menangis terus, maka Timun Mas keluar dari tempat sembunyinya. "Aku di sini raksasa, tangkaplah aku jika kau bisa!!!", teriak Timun Mas.

Raksasa pun mengejarnya, dan Timun Mas mulai melemparkan kantong yang berisi mentimun. Sungguh sia-sia, butan menjadi ladang mentimun yang lebat buahnya. Raksasa pun menjadi terhambat karena batang

Gambar 4.32 Gambar Pengujian 3 masukkan teks pengujian

Verifikasi Hasil

Waktu pengujian : 06-11-17 17:25:00

Jumlah Kata Inputan : 637

Jumlah Stopword : 302

Di suatu desa hiduplah seorang janda tua yang bernama mbok Sarni. Tiap hari dia menghabiskan waktunya sendirian, karena mbok Sarni tidak memiliki seorang anak. Sebenarnya dia ingin sekali mempunyai anak, agar bisa membantunya bekerja.

Pada suatu sore pergilah mbok Sarni ke hutan untuk mencari kayu, dan ditengah jalan mbok Sarni bertemu dengan raksasa yang sangat besar sekali. "Hei, mau kemana kamu?", tanya si Raksasa. "Aku hanya mau mengumpulkan kayu bakar, jadi ijinlanlah aku lewat", jawab mbok Sarni. "Hahahaha.... kamu boleh lewat setelah kamu memberiku seorang anak manusia untuk aku santap", kata si Raksasa. Lalu mbok Sarni menjawab, "Tetapi aku tidak mempunyai anak".

Gambar 4.33 Gambar Pengujian 3 informasi umum pengujian

Hasil Stemming Nazief & Adriani

Durasi : 10.980304956436 detik

Jumlah Kata Hasil 335

desa hidup janda tua nama mbok sarni habis mbok sarni milik anak anak bantu sore pergi mbok sarni hutan cari kayu tengah jalan mbok sarni bertemu raksasa hei mana si raksasa kumpul kayu bakar ijinlanlah mbok sarni hahahaha beri anak manusia santap si raksasa mbok sarni anak mbok sarni anak anak si raksasa beri biji mentimun raksasa wanita tua biji mentimun tanam biji halaman rumah minggu anak serah anak pada usia enam minggu mentimun nampak buah lebat salah mentimun mbok sarni ambil dibelah isi bayi cantik jelita bayi nama timun mas timun mas tumbuh mbok sarni gembira rumah sepi kerja selesai cepat bantu timun mas datang si raksasa tagih janji mbok sarni takut hilang timun mas mbok sarni raksasa datang sini dewasa anak enak santap si raksasa tuju tinggal rumah mbok sarni mbok sarni cari akal anak bawa si raksasa hati mbok sarni cemas malam mbok sarni mimpi mimpi diberitahu timun mas temu tapa gunung pagi hari mbok sarni suruh timun mas temu tapa bertemu tapa timun mas cerita maksud datang sang tapa beri buah bungkus isi biji mentimun jarum garam terasi lempar bungkus kejar raksasa perintah tapa timun mas pulang rumah langsung simpan bungkus sang tapa pagi raksasa tagih janji wanita tua anak tahan santap teriak si raksasa mbok sarni kau ambil anak raksasa sayang santap raksasa terima tawaran mbok sarni marah anak timun mas teriak si raksasa tega mbok sarni tangis timun mas sembunyi raksasa tangkap kau teriak timun mas raksasa kejar timun mas lempar kantong isi mentimun sungguh ajaib hutan ladang mentimun lebat buah raksasa hambat batang timun lilit tubuh si raksasa hasil bebas mngejar timun mas timun mas tabur kantong isi jarum kecap tumbuh pohon pohon bambu tajam kaki darah darah tancap bambu si raksasa kejar timun mas buka kantong tiga isi garam hutan lautan luas lautan mudah si raksasa timun mas tabur terasi bentuk lautan lumpur didih si raksasa cebur dalam raksasa mati timun mas ucap syukur tuhan yme selamat raksasa kejam timun mas mbok sarni hidup bahagia damai

Jumlah Kata Salah : 3

Gambar 4.34 Gambar Pengujian 3 informasi Nazief & Adriani

Hasil Stemming Idris

Durasi : 8.149670124054 detik
Jumlah Kata Hasil 335

desa hidup janda tua nama mbok sarni menghabiskan mbok sarni memiliki anak anak membantunya sore pergilah mbok sarni hutan cari kayu tengah jalan mbok sarni temu raksasa hei mana si raksasa mengumpulkan kayu bakar ijinkanlah mbok sarni hahahaha memberiku anak manusia santap si raksasa mbok sarni anak mbok sarni anak anak si raksasa memberinya biji mentimun raksasa wanita tua biji mentimun tanam biji halaman rumah minggu anak serahkan anak pada usia enam minggu mentimun nampak buah lebat salah mentimun mbok sarni mengambilnya belah isi bayi cantik jelita bayi nama timun mas timun mas tumbuh mbok sarni gembira rumah sepi pekerjaannya selesai cepat bantu timun mas datang si raksasa tagih janji mbok sarni takut hilang timun mas mbok sarni raksasa datang sini dewasa anak anak santap si raksasa tuju meninggalkan rumah mbok sarni mbok sarni cari akal anak bawa si raksasa hati mbok sarni cemas malam mbok sarni mimpi mimpi diberitahu timun mas menemui tapa gunung pagi hari mbok sarni menyuruh timun mas menemui tapa temu tapa timun mas cerita maksud kedatangannya sang tapa memberinya buah bungkus isi biji mentimun jarum garam terasi lempar bungkus kejar raksasa perintah tapa timun mas pulang rumah langsung menyimpan bungkus sang tapa pagi raksasa tagih janji wanita tua anak tahan menyantapnya teriak si raksasa mbok sarni kau ambil anak raksasa sayang santap raksasa terima tawaran mbok sarni marah anak timun mas teriak si raksasa tega mbok sarni tangis timun mas sembunyinya raksasa tangkap kau teriak timun mas raksasa mengejanya timun mas empur kantong isi mentimun sungguh ajaib hutan ladang mentimun lebat buah raksasa hambat batang timun lilit tubuh si raksasa hasil bebas mngejar timun mas timun mas abur kantong isi jarum kejar tumbuh pohon pohon bambu tajam kaki darah darah tancap bambu si raksasa mengejar timun mas buka kantong tiga isi garam hutan lautan luas lautan mudah si raksasa timun mas abur terasi terbentuklah lautan lumpur didih si raksasa cebur dalam raksasa mati timun mas ucap syukur tuhan yme selamat raksasa kejam timun mas mbok sarni hidup bahagia damai

Jumlah Kata Salah : 24

Gambar 4.35 Gambar Pengujian 3 informasi Idris

No.	Waktu Pengujian	Jumlah Kata	Algoritma	Durasi (detik)	Akurasi (%)
1	2017-11-06 17:25:00	335	Nazief & Adriani	10.9803	99.1045
2	2017-11-06 17:25:00	335	Idris	8.14967	92.8358

Gambar 4.36 Gambar Pengujian 3 rekap pengujian

Pada pengujian pertama di gambar 4.32 melakukan masukkan teks pada *text area* yang disediakan, lalu pada 4.33 menampilkan informasi umum mengenai pengujian seperti waktu pengujian yang dilakukan pada 06 November 2017 pukul 17:25:00 WIB, jumlah kata masukkan 637 kata, jumlah *stopword* sebanyak 302 kata, hingga teks awal masukkan.

Pada gambar 4.34 merupakan tampilan informasi yang dihasilkan dari proses *stemming* Nazief & Adirani seperti durasi pemrosesan berlangsung selama 10,98 detik, jumlah kata yang dihasilkan 335 kata, data kata hasil *stemming*, serta hasil perhitungan jumlah kata yang dihasilkan salah sebanyak 3 kata.

Pada gambar 4.35 merupakan tampilan informasi yang dihasilkan dari proses *stemming* Idris seperti durasi pemrosesan berlangsung selama 8,15 detik, jumlah kata yang dihasilkan 335 kata, data kata hasil *stemming*, serta hasil perhitungan jumlah kata yang dihasilkan salah sebanyak 24 kata.

Setelah dilakukan pengujian, pada gambar 4.36 menampilkan laman detail rekap mengenai pengujian yang telah dilakukan. Hasil yang di dapat berupa informasi jumlah kata, jenis algoritma yang digunakan, durasi, dan akurasinya.

Pengujian Keempat

Berikut merupakan informasi mengenai teks yang akan menjadi pengujian keempat.

Nama teks : Legenda Harimau Makan Durian.

Jumlah kata : 639 kata.

Dokumentasi pengujian teks kedua dapat dilihat pada gambar 4.37 hingga gambar 4.41.

Perbandingan Algoritma

☰

Pengujian

Desa Kemingking Dalam merupakan termasuk wilayah kecamatan Taman Rajo, kabupaten Muaro Jambi, Provinsi Jambi. Daerah ini terkenal dengan berbagai macam hasil bumi salah satunya adalah durian. Di desa Kemingking Dalam, musim durian biasanya tiba satu atau dua tahun sekali dengan hasil yang berlimpah. Durian dari daerah ini terkenal karena bentuknya yang tidak terlalu besar namun memiliki rasa khas yang manis dan legit. Setiap musim panen tiba, masyarakat desa Kemingking Dalam akan berbondong-bondong menunggui durian yang runtuh di kebun mereka masing-masing. Mereka menjaga kebun ini bersama keluarga mereka baik di waktu siang maupun malam. Tetapi, ketika musim panen hampir usai dan buah yang ada di pohon tinggal sedikit, masyarakat desa Kemingking Dalam tidak akan lagi menunggui kebun mereka di malam hari. Berkenaan dengan kebiasaan ini, terdapat sebuah cerita di dalamnya.

Pada suatu masa ketika desa Kemingking Dalam masih merupakan desa dengan pemerintahan tersendiri dan raja-rajanya masih berkuasa. Rakyat hidup berdampingan dalam kedamaian dan kesejahteraan berkat pemimpin yang bijaksana. Namun, tiba-tiba segala kemakmuran itu terganggu dengan hadirnya seekor harimau besar dari negeri seberang. Harimau ini buas, bengis, dan lapar. Ia tidak hanya menghabiskan ternak warga masyarakat, tetapi lambat laun harimau ini mulai menyerang manusia. Membuat belasan orang meninggal sedangkan puluhan lainnya luka-luka dengan cacat pada tubuhnya.

Melihat hal ini, Raja yang berkuasa di saat itu tidak dapat tinggal diam. Ia kemudian memerintahkan salah seorang prajuritnya yang paling sakti untuk mengatasi krisis yang terjadi di kerajaannya. Prajurit ini dengan patuh pergi mencari harimau untuk mengusir atau membunuhnya. Ketika berhadapan dengan sang harimau prajurit ini langsung menyerang dengan segala daya upaya yang dimilikinya. Namun sang harimau yang sangat besar dan kuat dapat dengan mudah mematahkan pedang dan tombak senjata sang prajurit serta melukai prajurit hingga terluka parah.

Mengetahui kondisinya yang tidak lagi memungkinkannya untuk bertarung secara maksimal, sang prajurit kemudian melarikan diri dari sang harimau dengan segenap kesaktiannya yang tersisa ia dapat menghindari pengejaran si harimau selama beberapa musim. Hingga akhir tahun itu tiba, cedera yang diderita sang prajurit masih belum pulih sepenuhnya. Ia masih belum sanggup untuk melawan sang harimau yang terus menengejanya seorang diri. Hingga ketika itu sampailah sang prajurit di sebuah daerah yang masih merupakan bagian dari wilayah Desa Kemingking Dalam sekarang ini yang dipenuhi aroma manis dan tanahnya dipenuhi buah yang penuh duri.

Di tempat ini sang prajurit tidak dapat lagi melarikan diri dan ia telah bertekad untuk melawan sang harimau apapun taruhannya. Ketika sang harimau mendapati sang prajurit tidak lagi melarikan diri ia pun menyerang sang prajurit tanpa ampun. Mereka kemudian bertarung dengan seluruh kemampuan mereka. Hingga kemudian sang prajurit menyadari kehadiran buah yang permukaannya dipenuhi duri itu. Ia kemudian menggunakan buah yang di masa kini dikenal dengan nama Durian sebagai senjatanya. Sang prajurit melempar harimau jahat itu dengan durian terus menerus hingga harimau itu terluka parah dan menyadari bahwa ia telah kalah.

Saat hendak menghabiskan sang harimau, harimau pun meminta ampun atas semua kesalahan yang telah ia lakukan di masa lalu. Ia pun berjanji kepada sang prajurit untuk tidak lagi menyerang warga asalkan ia diperbolehkan untuk melahap sebagian dari buah yang penuh duri yang tumbuh di tanah mereka itu. Karena rasa kasihan dan iba serta karena melihat kesungguhan dari sang harimau, maka sang prajurit pun membiarkan harimau untuk terus hidup dengan syarat ia tidak akan mendapat ampun lagi apabila ia melanggar janjinya pada sang prajurit.

Maka setelah sekian lama dalam pelarian kembalilah sang prajurit dengan kemenangan di pihaknya. Ia pun melaporkan segala yang terjadi kepada Rajanya dan meneruskan sumpah sang harimau kepada seluruh masyarakat untuk dihormati dan dipatuhi. Hingga sekarang, sumpah sang harimau terus dijaga oleh masyarakat desa Kemingking Dalam. Sehingga meskipun hutan desa Kemingking Dalam termasuk dalam wilayah kekuasaan harimau, harimau-harimau ini tidak pernah menampakkan diri ataupun menyerang warga. Mereka hanya muncul di waktu malam ketika musim durian hampir usai untuk melahap buah-buah terakhir yang telah diperjanjikan untuknya.

Gambar 4.37 Gambar Pengujian 4 masukkan teks pengujian

Verifikasi Hasil

Waktu pengujian : 06-11-17 18:56:15

Jumlah Kata Inputan : 639

Jumlah Stopword : 325

Desa Kemingking Dalam merupakan termasuk wilayah kecamatan Taman Rajo, kabupaten Muaro Jambi, Provinsi Jambi. Daerah ini terkenal dengan berbagai macam hasil bumi salah satunya adalah durian. Di desa Kemingking Dalam, musim durian biasanya tiba satu atau dua tahun sekali dengan hasil yang berlimpah. Durian dari daerah ini terkenal karena bentuknya yang tidak terlalu besar namun memiliki rasa khas yang manis dan legit. Setiap musim panen tiba, masyarakat desa Kemingking Dalam akan berbondong-bondong menunggui durian yang runtuh di kebun mereka masing-masing. Mereka menjaga kebun ini bersama keluarga mereka baik di waktu siang maupun malam. Tetapi, ketika musim panen hampir usai dan buah yang ada di pohon tinggal sedikit, masyarakat desa Kemingking Dalam tidak akan lagi menunggui kebun mereka di malam hari. Berkenaan

Gambar 4.38 Gambar Pengujian 4 informasi umum pengujian

Hasil Stemming Nazief & Adriani

Durasi : 10.270946025848 detik

Jumlah Kata Hasil 314

desa kemingking wilayah camat taman rajo kabupaten muaro jambi provinsi jambi daerah kenal hasil bumi salah satu durian desa kemingking musim durian hasil limpah durian daerah kenal bentuk milik khas manis legit musim panen masyarakat desa kemingking bondong bondong tunggu durian runtuh kebun jaga kebun keluarga siang malam musim panen buah pohon tinggal masyarakat desa kemingking tunggu kebun malam biasa cerita dalam desa kemingking desa perintah sendiri raja raja kuasa rakyat hidup damping damai sejahtera berkat pimpin bijaksana makmur ganggu hadir ekor harimau neger seberang harimau buas bengis lapar habis ternak warga masyarakat lambat laun harimau serang manusia belas orang tinggal puluh luka luka cacat tubuh raja kuasa tinggal diam perintah salah prajurit sakti atas krisis raja prajurit patuh pergi cari harimau usir bunuh hadap sang harimau prajurit langsung serang daya upaya milik sang harimau kuat mudah patah pedang tombak senjata sang prajurit luka prajurit luka parah kondisi mungkin tarung maksimal sang prajurit larik sang harimau segenap sakti sisa hindar kejar si harimau musim cidera derita sang prajurit pulih penuh sanggup lawan sang harimau kejar sampai sang prajurit daerah wilayah desa kemingking penuh aroma manis tanah penuh buah penuh duri sang prajurit larik tekad lawan sang harimau apa taruh sang harimau dapat sang prajurit larik serang sang prajurit ampun tarung mampu sang prajurit sadar hadir buah muka penuh duri buah kenal nama durian senjata sang prajurit lempar harimau jahat durian terus harimau luka parah sadar kalah habis sang harimau harimau ampun salah laku janji sang prajurit serang warga boleh lahap buah penuh duri tumbuh tanah kasihan iba sungguh sang harimau sang prajurit biar harimau hidup syarat ampun langgar janji sang prajurit kian lari kembali sang prajurit menang lapor raja terus sumpah sang harimau masyarakat hormat patuh sumpah sang harimau jaga masyarakat desa kemingking hutan desa kemingking wilayah kuasa harimau harimau harimau tampak serang warga muncul malam musim durian lahap buah buah janji untuk

Jumlah Kata Salah : 1

Gambar 4.39 Gambar Pengujian 4 informasi Nazief & Adriani

Hasil Stemming Idris

Durasi : 8.2385890483856 detik
 Jumlah Kata Hasil 314

desa kemingking wilayah camat taman rajo kabupaten muaro jambi provinsi jambi daerah kenal hasil bumi salah satu durian desa kemingking musim durian hasil limpah durian daerah kenal bentuknya memiliki khas manis legit musim panen masyarakat desa kemingking bondong bondong menunggu durian runtuh kebun jaga kebun keluarga siang malam musim panen buah pohon tinggal masyarakat desa kemingking menunggu kebun malam biasa cerita dalam desa kemingking desa perintah sendiri raja raja kuasa rakyat hidup berdampingan damai sejahtera berkat pimpin bijaksana makmur ganggu hadir ekor harimau neger seberang harimau buas bengis lapar menghabisi ternak warga masyarakat lambat laun harimau menyerang manusia asan orang tinggal puluh luka luka cacat tubuh raja kuasa tinggal diam perintah salah prajurit sakti mengatasi krisis kerajaannya prajurit patuh pergi cari harimau usir membunuhnya berhadapan sang harimau prajurit langsung menyerang daya upaya dimilikinya sang harimau kuat mudah patah pedang tombak senjata sang prajurit melukai prajurit luka parah kondisi memungkinkannya tarung maksimal sang prajurit arik sang harimau segenap kesaktiannya sisa menghindari pengejaran si harimau musim cidera derita sang prajurit penuh sanggup lawan sang harimau mengejanya sampai sang prajurit daerah wilayah desa kemingking penuh aroma manis tanah penuh buah penuh duri sang prajurit arik tekad lawan sang harimau apa taruhannya sang harimau dapat sang prajurit arik menyerang sang prajurit ampun tarung mampu sang prajurit menyadari hadir buah permukaannya penuh duri buah kenal nama durian senjatanya sang prajurit lempar harimau jahat durian terus harimau luka parah menyadari kalah menghabisi sang harimau harimau ampun salah laku janji sang prajurit menyerang warga boleh lahap buah penuh duri tumbuh tanah kasihan iba sungguh sang harimau sang prajurit membiarkan harimau hidup syarat ampun langgar janji sang prajurit kian ari kembalilah sang prajurit menang melaporkan raja meneruskan sumpah sang harimau masyarakat hormat patuh sumpah sang harimau jaga masyarakat desa kemingking hutan desa kemingking wilayah kuasa harimau harimau harimau menampakkan menyerang warga muncul malam musim durian lahap buah buah janji untuk

Jumlah Kata Salah : 37

Gambar 4.40 Gambar Pengujian 4 informasi Idris

No.	Waktu Pengujian	Jumlah Kata	Algoritma	Durasi (detik)	Akurasi (%)
1	2017-11-06 18:56:15	314	Nazief & Adriani	10.2709	99.6815
2	2017-11-06 18:56:15	314	Idris	8.23859	88.2166

Gambar 4.41 Gambar Pengujian 4 rekap pengujian

Pada pengujian kedua di gambar 4.37 melakukan masukkan teks pada *text area* yang disediakan, lalu pada 4.38 menampilkan informasi umum mengenai pengujian seperti waktu pengujian yang dilakukan pada 06 November 2017 pukul 18:56:15 WIB, jumlah kata masukkan 639 kata, jumlah *stopword* sebanyak 325 kata, hingga teks awal masukkan.

Pada gambar 4.39 merupakan tampilan informasi yang dihasilkan dari proses *stemming* Nazief & Adirani seperti durasi pemrosesan berlangsung selama 10,27 detik, jumlah kata yang dihasilkan 314 kata, data kata hasil *stemming*, serta hasil perhitungan jumlah kata yang dihasilkan salah sebanyak 1 kata.

Pada gambar 4.40 merupakan tampilan informasi yang dihasilkan dari proses *stemming* Idris seperti durasi pemrosesan berlangsung selama 8,24 detik,

jumlah kata yang dihasilkan 314 kata, data kata hasil *stemming*, serta hasil perhitungan jumlah kata yang dihasilkan salah sebanyak 37 kata.

Setelah dilakukan pengujian, pada gambar 4.41 menampilkan laman detail rekap mengenai pengujian yang telah dilakukan. Hasil yang di dapat berupa informasi jumlah kata, jenis algoritma yang digunakan, durasi, dan akurasi.

Pengujian Kelima

Berikut merupakan informasi mengenai teks yang akan menjadi pengujian kelima.

Nama teks : Cinderelas.

Jumlah kata : 733 kata.

Dokumentasi pengujian teks pertama dapat dilihat pada gambar 4.42 hingga gambar 4.46.

Perbandingan Algoritma

☰

Pengujian

Kerajaan Jenggala dipimpin oleh seorang raja yang bernama Raden Putra. Ia didampingi oleh seorang permaisuri yang baik hati dan seorang selir yang memiliki sifat iri dan dengki. Raja Putra dan kedua istrinya tadi hidup di dalam istana yang sangat megah dan damai. Hingga suatu hari selir raja merencanakan sesuatu yang buruk pada permaisuri raja. Hal tersebut dilakukan karena selir Raden Putra ingin menjadi permaisuri.

Selir baginda lalu berkomplot dengan seorang tabib istana untuk melaksanakan rencana tersebut. Selir baginda berpura-pura sakit parah. Tabib istana lalu segera dipanggil sang Raja. Setelah memeriksa selir tersebut, sang tabib mengatakan bahwa ada seseorang yang telah menaruh racun dalam minuman tuan putri. "Orang itu tak lain adalah permaisuri Baginda sendiri," kata sang tabib. Baginda menjadi murka mendengar penjelasan tabib istana. Ia segera memerintahkan patih untuk membuang permaisuri ke hutan dan membunuhnya.

Sang Patih segera membawa permaisuri yang sedang mengandung itu ke tengah hutan belantara. Tapi, patih yang bijak itu tidak mau membunuh sang permaisuri. Rupanya sang patih sudah mengetahui niat jahat selir baginda. "Tuan putri tidak perlu khawatir, hamba akan melaporkan kepada Baginda bahwa tuan putri sudah hamba bunuh," kata patih. Untuk mengelabui raja, sang patih melumuri pedangnya dengan darah kelinci yang ditangkapnya. Raja merasa puas ketika sang patih melapor kalau ia sudah membunuh permaisuri.

Setelah beberapa bulan berada di hutan, sang permaisuri melahirkan seorang anak laki-laki. Anak itu diberinya nama Cindelas. Cindelas tumbuh menjadi seorang anak yang cerdas dan tampan. Sejak kecil ia sudah berteman dengan binatang penghuni hutan. Suatu hari, ketika sedang asyik bermain, seekor rajawali menjatuhkan sebutir telur ayam. Cindelas kemudian mengambil telur itu dan bermaksud menetaskannya. Setelah 3 minggu, telur itu menetas menjadi seekor anak ayam yang sangat lucu. Cindelas memelihara anak ayamnya dengan rajin. Kian hari anak ayam itu tumbuh menjadi seekor ayam jantan yang gagah dan kuat. Tetapi ada satu yang aneh dari ayam tersebut. Bunyi kokok ayam itu berbeda dengan ayam lainnya. "Kukuruyuk... Tuanku Cindelas, rumahnya di tengah rimba, atapnya daun kelapa, ayahnya Raden Putra...", kokok ayam itu

Cindelas sangat takjub mendengar kokok ayamnya itu dan segera memperlihatkan pada ibunya. Lalu, ibu Cindelas menceritakan asal usul mengapa mereka sampai berada di hutan. Mendengar cerita ibunya, Cindelas bertekad untuk ke istana dan membeberkan kejahatan selir baginda. Setelah di ijinakan ibunya, Cindelas pergi ke istana ditemani oleh ayam jantannya. Ketika dalam perjalanan ada beberapa orang yang sedang menyabung ayam. Cindelas kemudian dipanggil oleh para penyabung ayam. "Ayo, kalau berani, adulah ayam jantanmu dengan ayamku," tantangnya. "Baiklah," jawab Cindelas. Ketika diadu, ternyata ayam jantan Cindelas bertarung dengan perkasa dan dalam waktu singkat, ia dapat mengalahkan lawannya. Setelah beberapa kali diadu, ayam Cindelas tidak terkalahkan.

Berita tentang kehebatan ayam Cindelas tersebar dengan cepat hingga sampai ke Istana. Raden Putra akhirnya pun mendengar berita itu. Kemudian, Raden Putra menyuruh hulubalangnya untuk mengundang Cindelas ke istana. "Hamba menghadap paduka," kata Cindelas dengan santun. "Anak ini tampan dan cerdas, sepertinya ia bukan keturunan rakyat jelata," pikir baginda. Ayam Cindelas diadu dengan ayam Raden Putra dengan satu syarat, jika ayam Cindelas kalah maka ia bersedia kepalanya dipancung, tetapi jika ayamnya menang maka setengah kekayaan Raden Putra menjadi milik Cindelas.

Dua ekor ayam itu bertarung dengan gagah berani. Tetapi dalam waktu singkat, ayam Cindelas berhasil menaklukkan ayam sang Raja. Para penonton bersorak sorai mengelu-elukan Cindelas dan ayamnya. "Baiklah aku mengaku kalah. Aku akan menepati janjiku. Tapi, siapakah kau sebenarnya, anak muda?" Tanva Baginda Raden Putra. Cindelas segera membungkuk seperti membisikkan sesuatu pada ayamnya.

Submit

Gambar 4.42 Gambar Pengujian 5 masukkan teks pengujian

Verifikasi Hasil

Waktu pengujian : 06-11-17 19:21:09

Jumlah Kata Inputan : 733

Jumlah Stopword : 283

Kerajaan Jenggala dipimpin oleh seorang raja yang bernama Raden Putra. Ia didampingi oleh seorang permaisuri yang baik hati dan seorang selir yang memiliki sifat iri dan dengki. Raja Putra dan kedua istrinya tadi hidup di dalam istana yang sangat megah dan damai. Hingga suatu hari selir raja merencanakan sesuatu yang buruk pada permaisuri raja. Hal tersebut dilakukan karena selir Raden Putra ingin menjadi permaisuri.

Selir baginda lalu berkomplot dengan seorang tabib istana untuk melaksanakan rencana tersebut. Selir baginda berpura-pura sakit parah. Tabib istana lalu segera dipanggil sang Raja. Setelah memeriksa selir tersebut, sang tabib mengatakan bahwa ada seseorang yang telah menaruh racun dalam minuman tuan putri. "Orang itu tak lain adalah permaisuri

Gambar 4.43 Gambar Pengujian 5 informasi umum pengujian

Hasil Stemming Nazief & Adriani

Durasi : 14.578009128571 detik

Jumlah Kata Hasil 450

raja jenggala pimpin raja nama raden putra damping permaisuri hati selir milik sifat iri dengki raja putra istri hidup istana megah damai selir raja rencana buruk permaisuri raja selir raden putra permaisuri selir baginda komplot tabib istana laksana rencana selir baginda pura pura sakit parah tabib istana panggil sang raja periksa selir sang tabib taruh racun minum tuan putri orang permaisuri baginda sang tabib baginda murka dengar jelas tabib istana perintah patih buang permaisuri hutan bunuh sang patih bawa permaisuri kandung hutan belantara patih bijak bunuh sang permaisuri sang patih niat jahat selir baginda tuan putri khawatir hamba lapor baginda tuan putri hamba bunuh patih kelabu raja sang patih lumur pedang darah kelinci tangkap raja puas sang patih lapor bunuh permaisuri hutan sang permaisuri lahir anak laki laki anak beri nama cindelas cindelas tumbuh anak cerdas teman binatang huni hutan asyik main ekor rajawali jatuh butir telur ayam cindelas ambil telur tetas 3 minggu telur tetas ekor anak ayam lucu cindelas pelihara anak ayam rajin kian anak ayam tumbuh ekor ayam jantan gagah kuat aneh ayam bunyi kokok ayam beda ayam kukuruyuk tuanku cindelas rumah rimba atap daun kelapa ayah raden putra kokok ayam cindelas takjub dengar kokok ayam ibu cindelas cerita usul hutan dengar cerita ibunda cindelas tekad istana beber jahat selir baginda ijinkan ibunda cindelas pergi istana teman ayam jantan jalan orang sabung ayam cindelas panggil sabung ayam ayo berani adu ayam jantan ayam tantang baik cindelas adu ayam jantan cindelas tarung perkasa singkat alah lawan kali adu ayam cindelas kalah berita hebat ayam cindelas sebar cepat istana raden putra dengar berita raden putra suruh hulubalang undang cindelas istana hamba hadap paduka cindelas santun anak tampan cerdas turun rakyat jelata pikir baginda ayam cindelas adu ayam raden putra syarat ayam cindelas kalah sedia kepala pancung ayam menang kaya raden putra milik cindelas ekor ayam tarung gagah berani singkat ayam cindelas hasil takluk ayam sang raja tonton sorak sorai elu elu cindelas ayam baik a kalah tepat janji kau anak muda baginda raden putra cindelas bungkuk bisik ayam ayam bunyi kukuruyuk tuanku cindelas rumah rimba atap daun kelapa ayah raden putra ayam jantan kokok ulang ulang raden putra peranjat dengar kokok ayam cindelas baginda heran baginda nama hamba cindelas hamba permaisuri baginda sama sang patih hadap cerita peristiwa permaisuri salah baginda raden putra hukum timpal selir baginda murka selir raden putra buang hutan raden putra peluk anak maaf salah raden putra hulubalang jemput permaisuri hutan raden putra permaisuri cindelas kumpul raden putra tinggal dunia cindelas ganti kedudukan ayah perintah neger adil bijaksana

Jumlah Kata Salah : 4

Gambar 4.44 Gambar Pengujian 5 informasi Nazief & Adriani

Hasil Stemming Idris

Durasi : 10.426416873932 detik
Jumlah Kata Hasil 450

raja jengala pimpin raja nama raden putra damping permaisuri hati selir memiliki sifat iri dengki raja putra istri hidup istana megah damai selir raja merencanakan buruk permaisuri raja selir raden putra permaisuri selir baginda komplot tabib istana melaksanakan rencana selir baginda pura pura sakit parah tabib istana panggil sang raja periksa selir sang tabib taruh racun minum tuan putri orang permaisuri baginda sang tabib baginda murka dengar jelas tabib istana perintah patih buang permaisuri hutan membunuhnya sang patih bawa permaisuri gandung hutan belantara patih bijak bunuh sang permaisuri sang patih niat jahat selir baginda tuan putri khawatir hamba melaporkan baginda tuan putri hamba bunuh patih mengelabui raja sang patih umur dang darah kelinci tangkap raja puas sang patih lapor bunuh permaisuri hutan sang permaisuri melahirkan anak laki laki anak beri nama cindelar cindelar tumbuh anak cerdas tampan teman binatang huni hutan asyik main ekor rajawali jatuh butir telur ayam cindelar ambil telur menetaskannya 3 minggu telur tetas ekor anak ayam lucu cindelar pelihara anak ayam rajin kian anak ayam tumbuh ekor ayam jantan gagah kuat aneh ayam bunyi kokok ayam beda ayam kukuruyuk tuanku cindelar rumah rimba atap daun kelapa ayah raden putra kokok ayam cindelar takjub dengar kokok ayam ibu cindelar cerita usul hutan dengar cerita ibunda cindelar tekad istana membeberkan jahat selir baginda ijinkan ibunda cindelar pergi istana teman ayam jantan jalan orang menyabung ayam cindelar panggil penyabung ayam ayo berani adu ayam jantan ayam tantang baik cindelar adu ayam jantan cindelar tarung perkasa singkat galah lawan kali adu ayam cindelar terkalahkan berita hebat ayam cindelar sebar cepat istana raden putra dengar berita raden putra menyuruh hulubalang undang cindelar istana hamba hadap paduka cindelar santun anak tampan cerdas turun rakyat jelata pikir baginda ayam cindelar adu ayam raden putra syarat ayam cindelar kalah sedia pala pancung ayam menang kaya raden putra milik cindelar ekor ayam tarung gagah berani singkat ayam cindelar hasil menaklukkan ayam sang raja tonton sorak sorai elu elu cindelar ayam baik aku kalah menepati janji kau anak muda baginda raden putra cindelar bungkok membisikkan ayam ayam bunyi kukuruyuk tuanku cindelar rumah rimba atap daun kelapa ayah raden putra ayam jantan kokok ulang ulang raden putra peranjat dengar kokok ayam cindelar baginda heran baginda nama hamba cindelar hamba permaisuri baginda bersamaan sang patih hadap cerita peristiwa permaisuri salah baginda raden putra hukum timpal lir baginda murka selir raden putra buang hutan raden putra peluk anak maaf kesalahannya raden putra hulubalang jemput permaisuri hutan raden putra permaisuri cindelar kumpul raden putra tinggal dunia cindelar menggantikan duduk ayah perintah negerinya adil bijaksana

Jumlah Kata Salah : 24

Gambar 4.45 Gambar Pengujian 5 informasi Idris

No.	Waktu Pengujian	Jumlah Kata	Algoritma	Durasi (detik)	Akurasi (%)
1	2017-11-06 19:21:09	450	Nazief & Adriani	14.578	99.1111
2	2017-11-06 19:21:09	450	Idris	10.4264	94.6667

Gambar 4.46 Gambar Pengujian 5 rekap pengujian

Pada pengujian pertama di gambar 4.42 melakukan masukkan teks pada *text area* yang disediakan, lalu pada 4.43 menampilkan informasi umum mengenai pengujian seperti waktu pengujian yang dilakukan pada 06 November 2017 pukul 19:21:09 WIB, jumlah kata masukkan 733 kata, jumlah *stopword* sebanyak 283 kata, hingga teks awal masukkan.

Pada gambar 4.44 merupakan tampilan informasi yang dihasilkan dari proses *stemming* Nazief & Adirani seperti durasi pemrosesan berlangsung selama 14,58 detik, jumlah kata yang dihasilkan 450 kata, data kata hasil *stemming*, serta hasil perhitungan jumlah kata yang dihasilkan salah sebanyak 4 kata.

Pada gambar 4.45 merupakan tampilan informasi yang dihasilkan dari proses *stemming* Idris seperti durasi pemrosesan berlangsung selama 10,43 detik,

jumlah kata yang dihasilkan 450 kata, data kata hasil *stemming*, serta hasil perhitungan jumlah kata yang dihasilkan salah sebanyak 24 kata.

Setelah dilakukan pengujian, pada gambar 4.46 menampilkan laman detail rekap mengenai pengujian yang telah dilakukan. Hasil yang di dapat berupa informasi jumlah kata, jenis algoritma yang digunakan, durasi, dan akurasi.

4.4 Kesimpulan



Gambar 4.47 Gambar Kesimpulan pengujian

Gambar 4.47 merupakan perhitungan dari seluruh pengujian yang telah dilakukan. Seperti yang telah dilakukan pengujian sebelumnya jumlah data yang telah dimasukkan adalah 5 dan dari 5 kali pengujian memiliki jumlah kata masukkan 3174 kata. Pada bagian perhitungan kecepatan dan akurasi dari masing-masing algoritma terlihat perbedaan. Algoritma Nazief & Adriani memiliki tingkat akurasi yang lebih tinggi dibandingkan dengan algoritma Idris, namun pada bagian kecepatan algoritma Idris memiliki keunggulan dari pada algoritma Nazief & Adriani. Hal ini disebabkan oleh perbedaan susunan tahapan yang dilalui oleh kedua algoritma tersebut. Tahapan penghilangan turunan awalan pada bagian algoritma Nazief & Adriani terletak di bagian akhir sehingga kata yg diolah dapat memilih awalan yang akan dihilangkan agar menghasilkan hasil yang lebih

maksimal. Sedangkan pada algoritma Idris peletakan tahapan penghilangan turunan awalan terletak di bagian awal, sehingga jika awalan yang telah dihapus akan langsung diteruskan ke tahap selanjutnya, tanpa mengetahui kebenaran awalan yang telah dihilangkan sudah benar, contohnya “membuang” setelah melalui tahapan penghilangan turunan awalan akan menjadi “mbuang” dan akan membuat hasil akhir menjadi rancu. Pada bagian waktu memiliki bahasan yang sama karena algoritma Nazief & Adriani perlu mencocokkan akhiran hingga tepat, namun pada bagian algoritma Idris hanya perlu menemukan kasus awalan yang pas maka akan langsung dihilangkan tanpa perlu mengetahui penghilangan yang dilakukan sudah benar atau belum.

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah dilakukan pengujian pada sistem yang dibangun, dapat disimpulkan bahwa:

1. Algoritma *stemming* Idris memiliki waktu pemrosesan yang lebih cepat dibandingkan dengan algoritma *stemming* Nazief & Adriani.
2. Algoritma *stemming* Nazief & Adriani memiliki akurasi hasil yang lebih tinggi dibandingkan dengan algoritma *stemming* Idris.

5.2 Saran

Saran yang diberikan untuk pengembangan lebih lanjut dari perbandingan algoritma *stemming* Nazief & Adriani dengan algoritma Idris pada teks bahasa Indonesia adalah sebagai berikut:

1. Untuk pengembangan lebih lanjut, disarankan memilih algoritma *stemming* yang belum pernah diteliti, contohnya algoritma *stemming* Lovins yang dibandingkan dengan algoritma *stemming* Nazief & Adriani.
2. Untuk pengembangan lebih lanjut, disarankan mengimplementasikan algoritma *stemming* Nazief & Adriani atau algoritma *stemming* Idris untuk pengolahan *text mining* atau program yang memerlukan pengolahan teks.

DAFTAR PUSTAKA

- Ariprasetya, Y. 2016. *Sistem Informasi Geografis PT Prudential Life Assurance Menggunakan Metode Haversine Formula Berbasis Android*. Skripsi. Tidak diterbitkan. Fakultas Teknik dan Ilmu Komputer. Universitas Potensi Utama: Sumatera Utara.
<http://repository.potensi-utama.ac.id/jspui/handle/123456789/1433>
- Darwis, SA, Abdullah, R, dan Idris, N. 2012. *Exhaustive Affix Stripping and a Malay Word Register to Solve Stemming Errors and Ambiguity Problem in Malay Stemmers*. Kuala Lumpur: Malaysian Journal of Computer Science. Vol. 24, No. 4:196-209.
https://umexpert.um.edu.my/file/publication/00001783_88454.pdf
- Famsila, K. 2016. *Pengembangan Sistem Informasi Manajemen Usaha Dagang X*. Skripsi. Tidak diterbitkan. Fakultas Teknologi Industri. Universitas Atma Jaya: Yogyakarta.
<http://e-journal.uajy.ac.id/11046/>
- Hasanah, Nur. 2017. *Sistem Pencarian Skripsi Berbasis Information Retrieval di FASTIKOM UNSIQ*. Jawa Tengah: Jurnal PPKM. Vol. 1, No. 1:105-113.
<http://jurnalppkm.unsiq.ac.id/index.php/ppkm/article/view/115>
- Kurniawan, B, Effendi, S, dan Sitompul, OS. 2012. *Klasifikasi Konten Berita Dengan Metode Text Mining*. Sumatera Utara: Jurnal Dunia Teknologi. Vol. 1, No. 1:14-19.
<https://jurnal.usu.ac.id/index.php/duniait/article/view/409>
- Kurniawan, YB, Wibowo, A, dan Dewi, LP. 2016. *Website Publikasi dan Evaluasi Kegiatan Lembaga Kemahasiswaan Universitas Kristen Petra*. Surabaya: Jurnal Infra. Vol 4, No. 2: 98-104.
<https://media.neliti.com/media/publications/104864-ID-website-publikasi-dan-evaluasi-kegiatan.pdf>
- Novitasari, Dian. 2016. *Perbandingan Algoritma Stemming Porter dengan Arifin Setiono untuk Menentukan Tingkat Ketepatan Kata Dasar*. Jakarta: Jurnal String. Vol. 1, No. 2:120-129.

<http://journal.lppmunindra.ac.id/index.php/STRING/article/view/1031>

Nugroho, HT. 2017. *Pengaruh Algoritma Stemming Nazief-Adriani Terhadap Kinerja Algoritma Winnowing Untuk Mendeteksi Plagiarisme Bahasa Indonesia*. Tangerang: ULTIMA Computing. Vol. IX, No. 1:36-40.

<http://ejournals.umn.ac.id/index.php/SK/article/view/572>

Oktarino, A. 2015. *Perancangan Sistem Informasi Rekam Medis Pasien pada Klinik Bersalin Kasih Ibu Menggunakan Metode Waterfall*. Jambi: Scientia Journal. Vol. 4, No. 3:239-247.

<http://www.stikesprima-jambi.ac.id/ojs/index.php/sc/article/view/123>

Permatasari, N. 2016. *Analisa Perbandingan Algoritma Idris dan Algoritma Enhanced Confix (ECS) Stemmer Pada Dokumen Teks Bahasa Indonesia*. Skripsi. Tidak diterbitkan. Fakultas Teknik dan Ilmu Komputer. Universitas Komputer Indonesia: Bandung.

<https://repository.unikom.ac.id/130/>

Putri, NB. 2013. *Penerapan Sistem Informasi Akuntansi Penerimaan Pendapatan Pada Pt. Kereta Api Indonesia (Persero) Daop Viii Surabaya*. Skripsi. Tidak diterbitkan. Sekolah Tinggi Ilmu Ekonomi Indonesia: Surabaya.

<http://repository.stiesia.ac.id/623/>

Rahmawati, L, Sihwi, SW, dan Suryani, E. 2016. *Analisa Clustering Menggunakan Metode K-Means dan Hierarchical Clustering (Studi Kasus: Dokumen Skripsi Jurusan Kimia, Fmipa, Universitas Sebelas Maret)*. Surakarta: ITSmart: Jurnal Teknologi dan Informasi. Vol. 3, No. 2: 66-73.

<https://jurnal.uns.ac.id/itsmart/article/viewFile/654/1616>

Sari, HP, dan Muhartini, R. 2017. *Sistem Aplikasi Pengolahan Nilai Raport Sdn Tanjunganom 2 Kecamatan Tanjunganom Nganjuk*. Blitar: Jurnal Antivirus. Vol. 11, No. 1: 65-80.

<http://unisbablitar.ejournal.web.id/index.php/antivirus/article/view/199>

Setyawan, E, dan Purnama, BE. 2012. *Pembangunan Sistem Informasi Perpustakaan Sekolah Dasar Negeri (SDN) Pacitan*. Pacitan: Journal Speed. Vol. 4, No. 3: 20-24.

<http://www.ijns.org/journal/index.php/speed/article/view/942>

Sihite, JA, dan Christianti, M. 2017. *Sistem Informasi Pengelolaan Daftar Kehadiran Dan Jadwal Pembicara Berbasis Website Dengan Mail Gateway Pada Komunitas Jc Bandung*. Bandung: Jurnal Teknik Informatika dan Sistem Informasi. Vol. 3, No. 2: 281-300.

<http://jutisi.maranatha.edu/index.php/jutisi/article/view/618>

Sugiarti, Y. 2013. *Analisis & Perancangan UML (Unified Modeling Language) Generated VB.6*. Graha Ilmu: Yogyakarta.

Utomo, MS. 2013. *Implementasi Stemmer Tala pada Aplikasi Berbasis Web*. Semarang: Jurnal Terknologi Informasi DINAMIK. Vol. 18, No. 1:41-45.

<http://www.unisbank.ac.id/ojs/index.php/fti1/article/view/1673>

Widayanati, R, dan Hidayat, F. 2013. *Desain Dan Implementasi Sistem Informasi Administrasi Pada Cv. Dian Abadi Malang Berbasis Desktop*. Malang: Jurnal Dinamika Dotcom. Vol. 4, No. 1: 1-14.

<http://ejurnal.stimata.ac.id/index.php/DINAMIKA/article/viewFile/54/94>

Wijaya, MF. 2015. *Pengembangan Sistem Informasi Rumah Tangga Berbasis Website Pada Divisi Data Pelanggan Dan Sistem Informasi Pt. Pelabuhan Indonesia Ii (Persero) Tanjung Priok*. Skripsi. Tidak diterbitkan. Fakultas Matematika dan Ilmu Pengetahuan Alam. Universitas Lampung: Lampung.

<http://digilib.unila.ac.id/14982/>