BAB 4
IMPLEMENTASI DAN PENGUJIAN

4.1. Implementasi Sistem

Pada tahap ini dilakukan penerjemahan perancangan berdasarkan analisis ke dalam bahasa pemrograman PHP. Adapun implementasi sistem terdiri dari implementasi perangkat keras, implementasi perangkat lunak, implementasi basis data, dan implementasi antarmuka.

4.1.1 Implementasi Perangkat Keras

Perangkat keras yang digunakan dalam pembangunan aplikasi deteksi dan koreksi kesalahan ejaan *real-word* menggunakan metode n-gram mempunyai spesifikasi sebagai berikut.

Tabel 4.1 Spesifikasi Perangkat Keras

No	Perangkat Keras	Spesifikasi
1	Processor	Intel atau AMD setara Pentium IV 3.0 Ghz
2	Monitor	Resolusi minimal 1024 x 768 pixel
3	Memori (RAM)	1024MB
4	Harddisk	320GB HDD

4.1.2 Implementasi Perangkat Lunak

Perangkat lunak yang digunakan dalam implementasi aplikasi ini dengan spesifikasi sebagai berikut.

Tabel 4.2 Spesifikasi Perangkat Lunak

No	Perangkat Keras	Spesifikasi
1	Sistem Operasi	Windows 7 Ultimate 64-bit
2	HTML *Editor*	Sublime text 3
3	*Web Server*	XAMPP
4	*Web Browser*	Google Chrome
5	DBMS	MySQL

4.1.3 Implementasi Basis Data

Implementasi basis data yang digunakan dalam sistem adalah sebagai berikut.



Gambar 4.1 Implementasi Basis Data

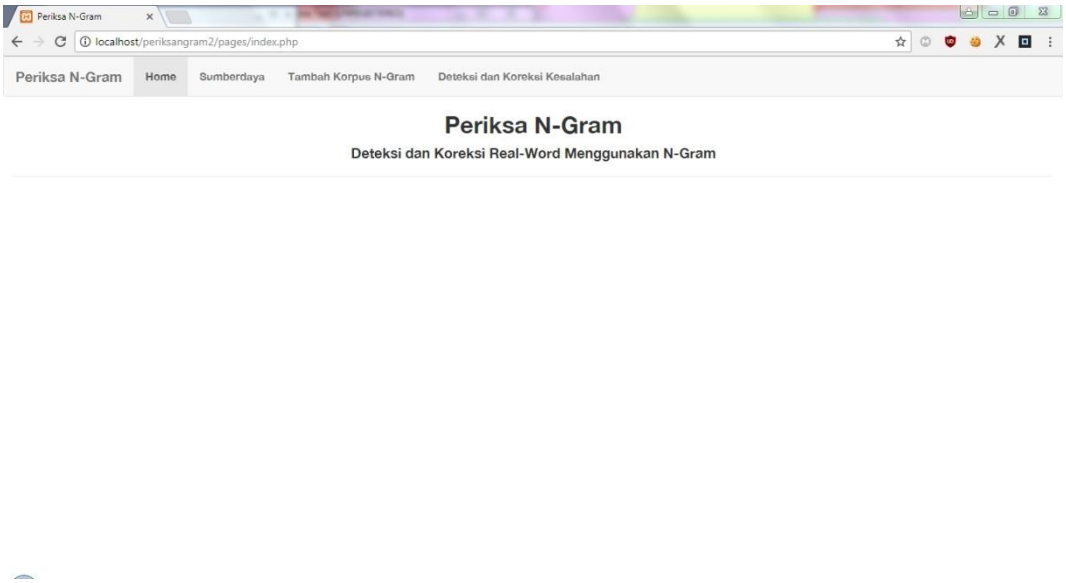
4.1.4 Implementasi Antarmuka

Implementasi antarmuka dilakukan untuk setiap tampilan aplikasi yang dibangun. Penjelasan implementasi antarmuka pada aplikasi deteksi dan koreksi kesalahan ejaan *real-word* menggunakan metode n-gram dapat dilihat pada tabel berikut.

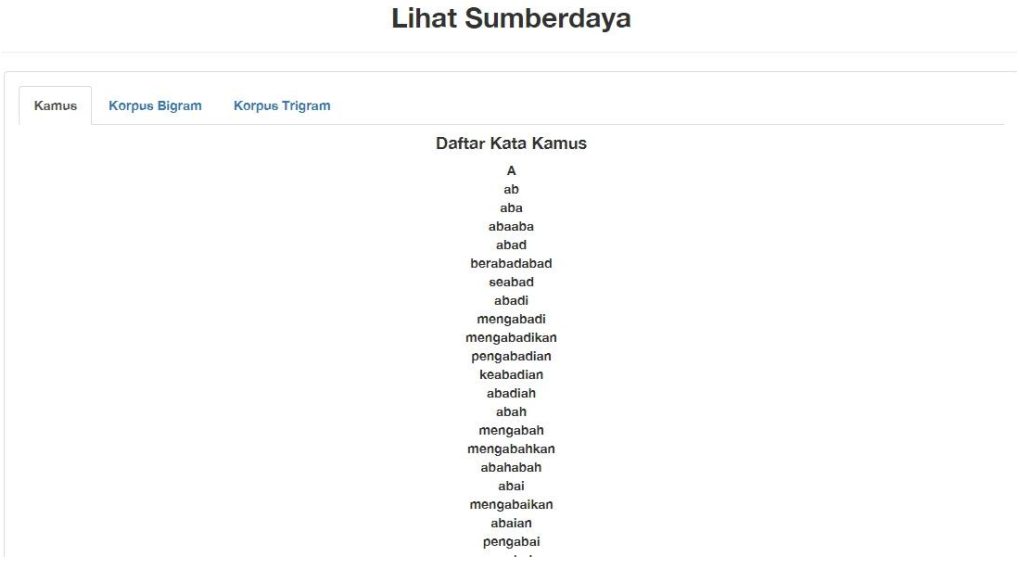
Tabel 4.3 Implementasi Antarmuka

No	Nama Antarmuka	Deskripsi	Nama File
1	Home	Menampilkan halaman utama aplikasi.	index.php
2	Sumber Daya	Menampilkan daftar kata kamus, korpus bigram, dan korpus trigram yang digunakan dalam aplikasi.	sumberdaya.php
3	Tambah Korpus N-Gram	Menampilkan halaman proses menambah korpus bigram dan trigram dengan data dari *file* masukan.	tambahkorpus.php
4	Deteksi dan Koreksi Kesalahan Kata	Menampilkan halaman proses deteksi dan koreksi kesalahan kata *real-word* pada *file* masukan.	periksa.php

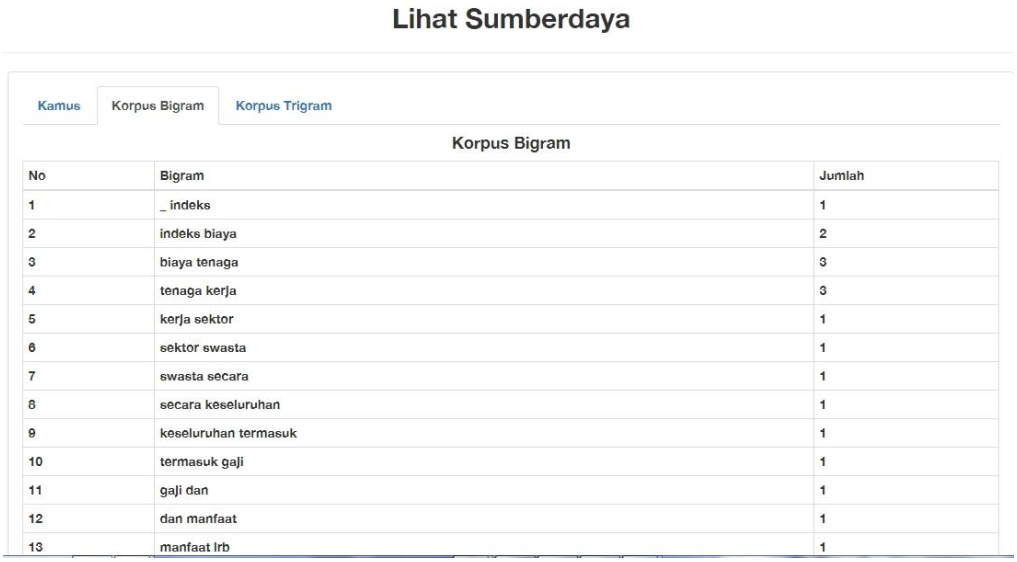
Implementasi antarmuka dapat dilihat pada gambar-gambar berikut.



Gambar 4.2 Antarmuka Halaman Home



Gambar 4.3 Antarmuka Daftar Kata Kamus



No	Bigram	Jumlah
1	_ indeks	1
2	indeks biaya	2
3	biaya tenaga	3
4	tenaga kerja	3
5	kerja sektor	1
6	sektor swasta	1
7	swasta secara	1
8	secara keseluruhan	1
9	keseluruhan termasuk	1
10	termasuk gaji	1
11	gaji dan	1
12	dan manfaat	1
13	manfaat lrb	1

Gambar 4.4 Antarmuka Korpus Bigram



No	Trigram	Jumlah
1	_ indeks biaya	1
2	indeks biaya tenaga	2
3	biaya tenaga kerja	3
4	tenaga kerja sektor	1
5	kerja sektor swasta	1
6	sektor swasta secara	1
7	swasta secara keseluruhan	1
8	secara keseluruhan termasuk	1
9	keseluruhan termasuk gaji	1
10	termasuk gaji dan	1
11	gaji dan manfaat	1
12	dan manfaat lrb	1
13	manfaat lrb benefit	1

Gambar 4.5 Antarmuka Korpus Trigram



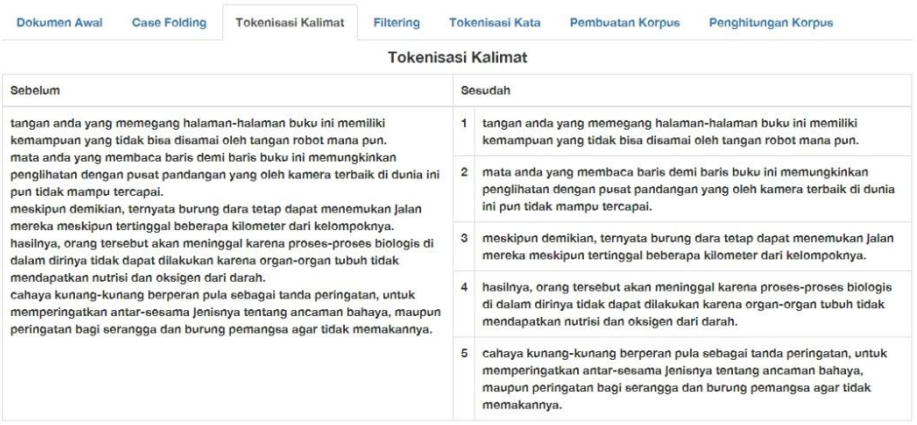
Gambar 4.6 Antarmuka Form Tambah Korpus N-Gram



Gambar 4.7 Antarmuka Dokumen Awal Tambah Korpus



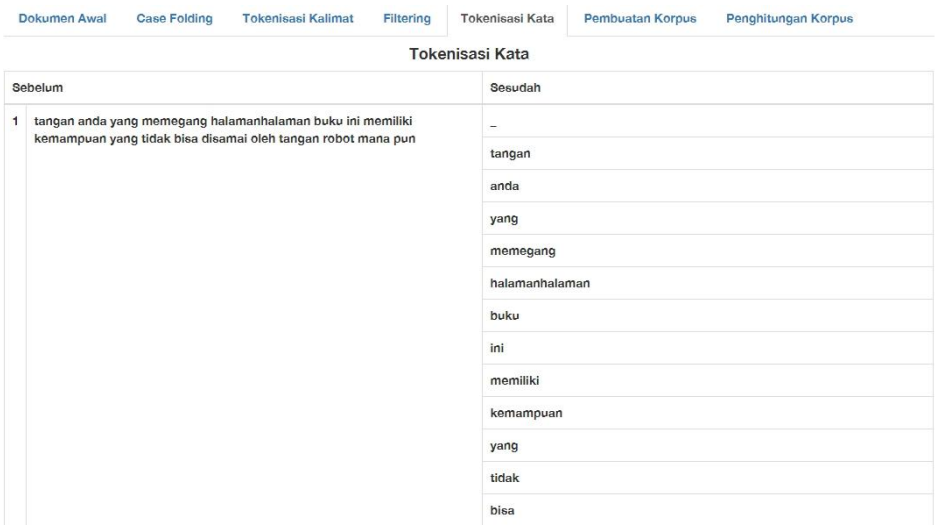
Gambar 4.8 Antarmuka *Case Folding*



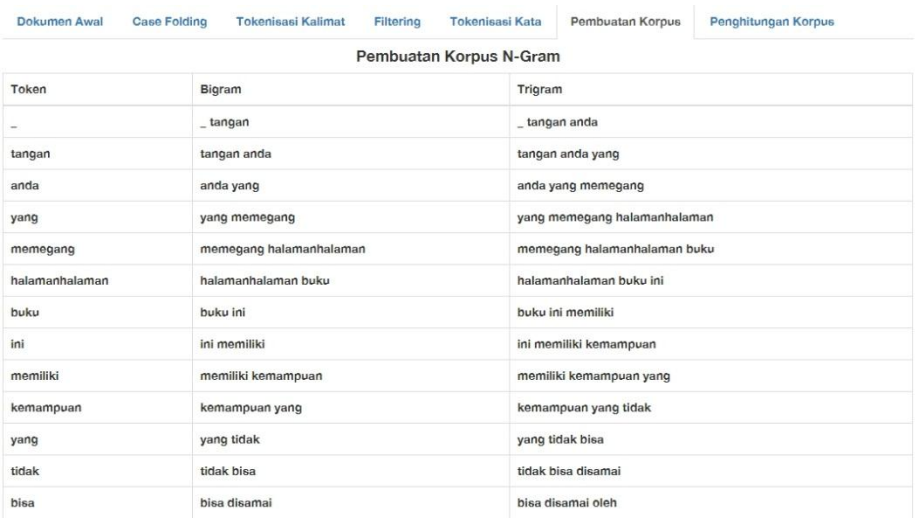
Gambar 4.9 Antarmuka Tokenisasi Kalimat



Gambar 4.10 Antarmuka *Filtering* Tanda Baca



Gambar 4.11 Antarmuka Tokenisasi Kata



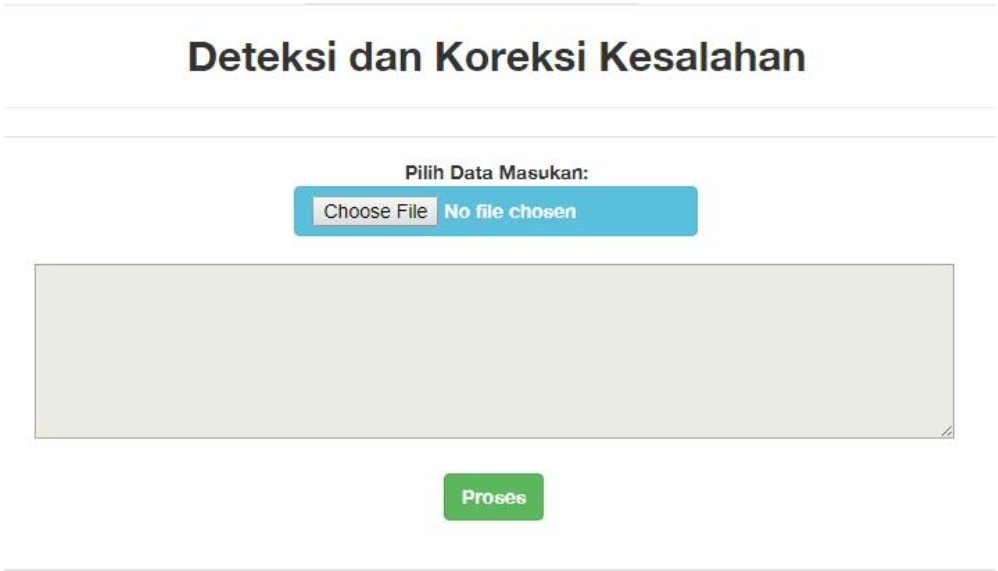
Token	Bigram	Trigram
_	_ tangan	_ tangan anda
tangan	tangan anda	tangan anda yang
anda	anda yang	anda yang memegang
yang	yang memegang	yang memegang halamanhalaman
memegang	memegang halamanhalaman	memegang halamanhalaman buku
halamanhalaman	halamanhalaman buku	halamanhalaman buku ini
buku	buku ini	buku ini memiliki
ini	ini memiliki	ini memiliki kemampuan
memiliki	memiliki kemampuan	memiliki kemampuan yang
kemampuan	kemampuan yang	kemampuan yang tidak
yang	yang tidak	yang tidak bisa
tidak	tidak bisa	tidak bisa disamai
bisa	bisa disamai	bisa disamai oleh

Gambar 4.12 Antarmuka Pembuatan Korpus N-Gram

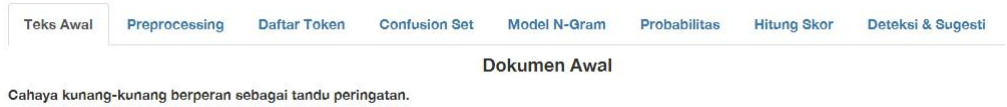


Bigram	Jumlah	Trigram	Jumlah
_ tangan	1	_ tangan anda	1
tangan anda	1	tangan anda yang	1
anda yang	2	anda yang memegang	1
yang memegang	1	yang memegang halamanhalaman	1
memegang halamanhalaman	1	memegang halamanhalaman buku	1
halamanhalaman buku	1	halamanhalaman buku ini	1
buku ini	2	buku ini memiliki	1
ini memiliki	1	ini memiliki kemampuan	1
memiliki kemampuan	1	memiliki kemampuan yang	1
kemampuan yang	1	kemampuan yang tidak	1
yang tidak	1	yang tidak bisa	1
tidak bisa	1	tidak bisa disamai	1
bisa disamai	1	bisa disamai oleh	1

Gambar 4.13 Antarmuka Penghitungan Korpus N-Gram



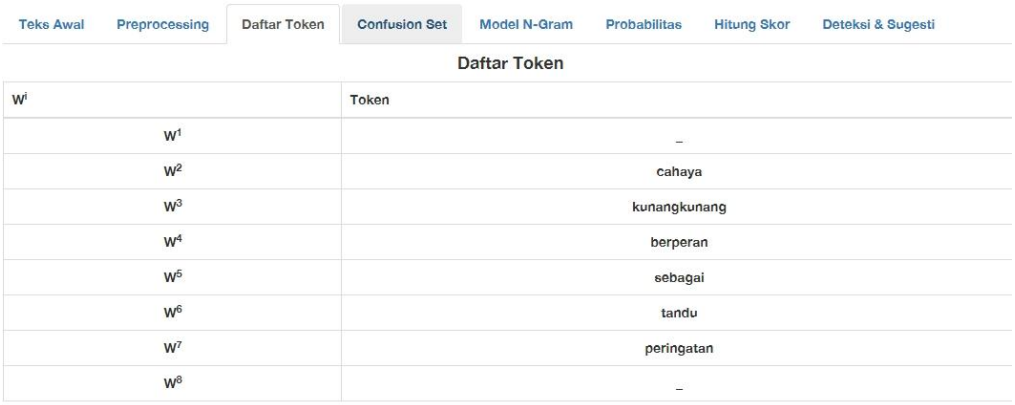
Gambar 4.14 Antarmuka Form Deteksi dan Koreksi Kesalahan



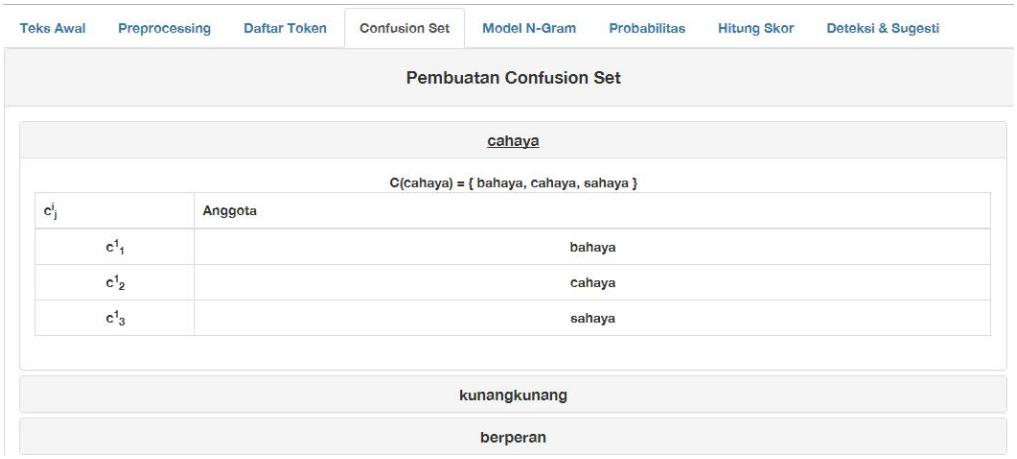
Gambar 4.15 Antarmuka Dokumen awal Deteksi dan Koreksi



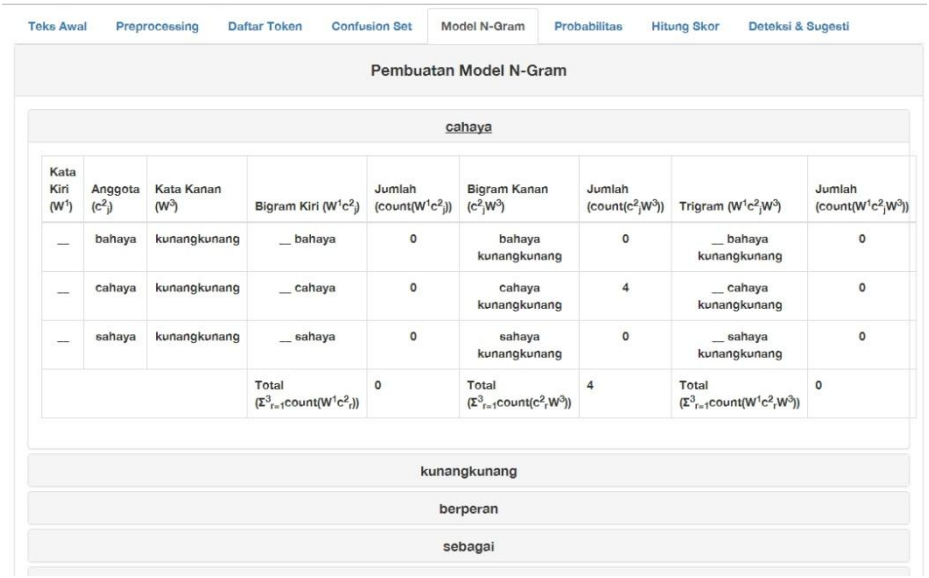
Gambar 4.16 Antarmuka Preprocessing



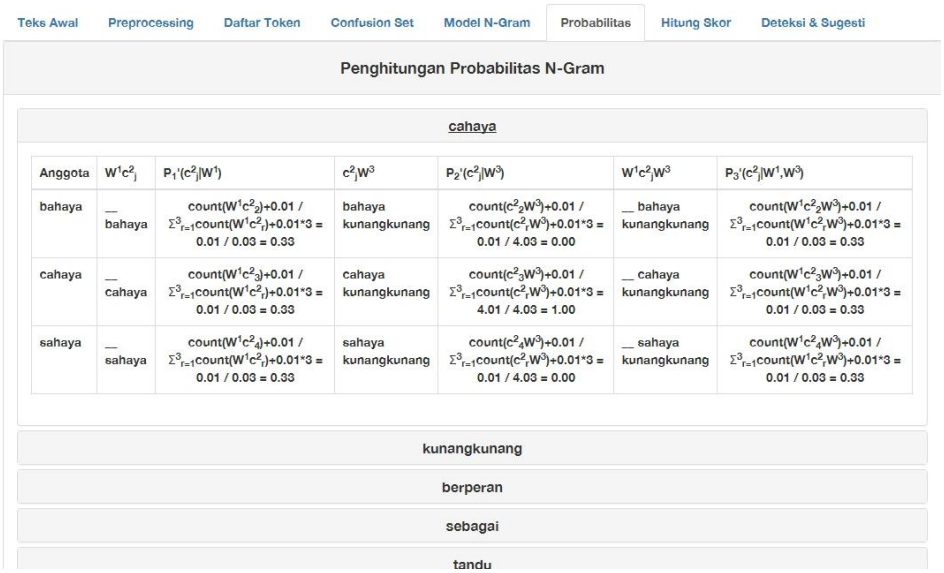
Gambar 4.17 Antarmuka Daftar Token



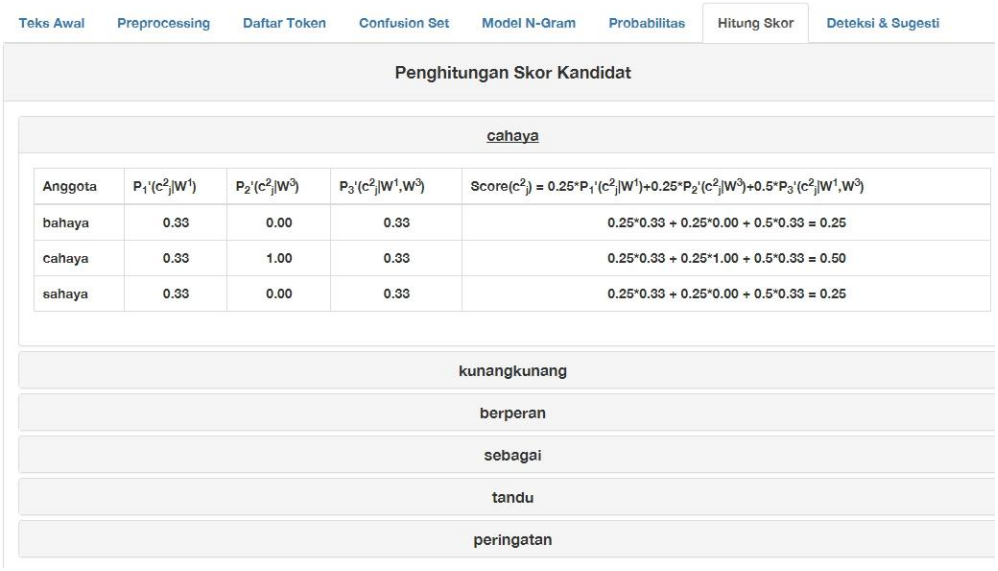
Gambar 4.18 Antarmuka Pembuatan *Confusion Set*



Gambar 4.19 Antarmuka Pembuatan Model N-Gram



Gambar 4.20 Antarmuka Penghitungan Probabilitas N-Gram



Gambar 4.21 Antarmuka Penghitungan Skor Kandidat



Gambar 4.22 Antarmuka Deteksi dan Koreksi Kesalahan

4.2. Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan bahwa sistem yang dibuat sesuai dengan rancangan sistem yang telah dibangun sebelumnya. Pada tahap pengujian sistem terdiri dari beberapa bagian, yaitu skenario pengujian, hasil pengujian, dan kesimpulan pengujian.

4.2.1 Skenario Pengujian

Skenario pengujian dilakukan berdasarkan pengujian fungsionalitas dan pengujian nilai akurasi dari hasil deteksi dan koreksi kesalahan kata yang dihasilkan.

4.2.1.1 Skenario Pengujian Fungsionalitas

Pengujian fungsionalitas pada aplikasi deteksi dan koreksi kesalahan kata *real-word* ini menggunakan metode *black box*, dimana pengujian berfokus pada *output* yang dihasilkan dalam menggapai *input* dan kondisi eksekusi yang dipilih untuk menampilkan *output* atau proses setiap tahapnya. Adapun skenario pengujian fungsionalitas dapat dilihat sebagai berikut.

Tabel 4.4 Skenario Pengujian Fungsionalitas

No	Nama Proses	Poin Pengujian	Jenis Pengujian
1	*Preprocessing* (Korpus N-Gram)	*Case Folding*	*Black Box*
		Tokenisasi Kalimat	*Black Box*
		Filtering Tanda Baca	*Black Box*
		Tokenisasi Kata	*Black Box*
		Pembuatan Korpus N-Gram	*Black Box*
		Penghitungan Korpus N-Gram	*Black Box*

No	Nama Proses	Poin Pengujian	Jenis Pengujian
2	*Preprocessing* (Deteksi dan Koreksi Kesalahan)	*Case Folding*	*Black Box*
		Tokenisasi Kalimat	*Black Box*
		Filtering Tanda Baca	*Black Box*
		Tokenisasi Kata	*Black Box*
3	Deteksi dan Koreksi Kesalahan Kata *Real-Word*	Pembuatan *Confusion Set*	*Black Box*
		Pembuatan Model N-Gram	*Black Box*
		Penghitungan Probabilitas N-Gram	*Black Box*
		Penghitungan Skor	*Black Box*
		Deteksi Kesalahan dan Sugesti Koreksi	*Black Box*

4.2.1.2 Skenario Pengujian Nilai Akurasi

Pengujian nilai akurasi pada aplikasi deteksi dan koreksi kesalahan kata *real-word* ini dilakukan untuk untuk mengetahui sejauh mana hasil deteksi dan koreksi kesalahan kata *real-word* yang dihasilkan benar-benar tepat. Korpus n-gram yang digunakan bersumber dari korpus PAN-BPPT [23] sebanyak total 500 ribu kata dan UI-Tagged-Corpus [22] sebanyak 1 juta kata, yang menghasilkan korpus bigram sebanyak 755 ribu string dan korpus trigram sebanyak 1 juta string. Teks yang digunakan sebagai data uji adalah 30 artikel berita yang berasal dari www.detik.com dan www.kompas.co.id. Nilai akurasi didapat dengan membagi jumlah deteksi sistem dengan jumlah deteksi yang dilakukan secara manual.

4.2.2 Pengujian

Pengujian-pengujian pada aplikasi deteksi dan koreksi kesalahan kata *real-word* ini terdiri dari pengujian fungsionalitas dan pengujian nilai performansi dari hasil deteksi dan koreksi kesalahan kata yang dihasilkan.

4.2.2.1 Pengujian Fungsionalitas

Pengujian *black box* berfokus pada persyaratan fungsional dari sebuah sistem yang dibangun dan menemukan kesalahan program. Hasil pengujian untuk antarmuka dapat dilihat sebagai berikut.

Tabel 4.5 Pengujian Fungsionalitas *Preprocessing* (Korpus N-Gram)

Aktivitas yang Dilakukan	Data Masukan	Keluaran yang Diharapkan	Pengamatan	Kesimpulan
Case Folding	Teks korpus	Teks dengan semua karakter adalah *lowercase*	Menampilkan teks yang semua karakternya *lowercase*	[√] Diterima [ ] Ditolak
Tokenisasi Kalimat	Teks korpus	Kumpulan token kalimat	Menampilkan kumpulan token kalimat	[√] Diterima [ ] Ditolak

Aktivitas yang Dilakukan	Data Masukan	Keluaran yang Diharapkan	Pengamatan	Kesimpulan
Filtering Tanda Baca	Teks korpus	Kumpulan kalimat tanpa tanda baca	Menampilkan kumpulan kalimat tanpa tanda baca	[√] Diterima [ ] Ditolak
Tokenisasi Kata	Teks korpus	Kumpulan token kata	Menampilkan kumpulan token kata	[√] Diterima [ ] Ditolak
Pembuatan Korpus N-Gram	Teks korpus	Korpus bigram dan trigram	Menampilkan korpus bigram dan trigram	[√] Diterima [ ] Ditolak
Penghitungan Korpus N-Gram	Teks korpus	Korpus bigram dan trigram yang telah dihitung	Menampilkan korpus bigram dan trigram yang telah dihitung	[√] Diterima [ ] Ditolak

Tabel 4.6 Pengujian Fungsionalitas *Preprocessing* (Deteksi dan Koreksi Kesalahan)

Aktivitas yang Dilakukan	Data Masukan	Keluaran yang Diharapkan	Pengamatan	Kesimpulan
Case Folding	Teks uji	Teks dengan semua karakter adalah *lowercase*	Menampilkan teks yang semua karakternya *lowercase*	[√] Diterima [ ] Ditolak
Tokenisasi Kalimat	Teks uji	Kumpulan token kalimat	Menampilkan kumpulan token kalimat	[√] Diterima [ ] Ditolak
Filtering Tanda Baca	Teks uji	Kumpulan kalimat tanpa tanda baca	Menampilkan kumpulan kalimat tanpa tanda baca	[√] Diterima [ ] Ditolak
Tokenisasi Kata	Teks uji	Kumpulan token kata	Menampilkan kumpulan token kata	[√] Diterima [ ] Ditolak

Tabel 4.7 Pengujian Fungsionalitas Deteksi dan Koreksi Kesalahan *Real-Word*

Aktivitas yang Dilakukan	Data Masukan	Keluaran yang Diharapkan	Pengamatan	Kesimpulan
Pembuatan *Confusion Set*	Teks uji	*Confusion set* untuk setiap kata	Menampilkan *confusion set* untuk setiap kata	[√] Diterima [ ] Ditolak
Pembuatan Model N-Gram	Teks uji	Model n-gram untuk setiap anggota *confusion set*	Menampilkan model n-gram untuk setiap anggota *confusion set*	[√] Diterima [ ] Ditolak
Penghitungan Probabilitas N-Gram	Teks uji	Probabilitas n-gram untuk setiap anggota *confusion set*	Menampilkan probabilitas n-gram untuk setiap anggota *confusion set*	[√] Diterima [ ] Ditolak
Penghitungan Skor	Teks uji	Skor untuk setiap anggota *confusion set*	Menampilkan skor untuk setiap anggota *confusion set*	[√] Diterima [ ] Ditolak
Deteksi Kesalahan dan Sugesti Koreksi	Teks uji	Kata yang salah dan sugesti koreksi katanya	Menampilkan kata yang salah dan sugesti koreksi katanya	[√] Diterima [ ] Ditolak

4.2.2.2 Pengujian Nilai Akurasi

Pengujian nilai akurasi metode deteksi dan koreksi kesalahan kata *real-word* menggunakan metode n-gram dapat dilihat sebagai berikut.

Tabel 4.8 Pengujian Akurasi Deteksi

Teks (jumlah kalimat)	Jumlah Deteksi		Akurasi	Analisis Kesalahan Deteksi
	Sistem	Manual		
28 (9 kalimat)	0	6	0%	N-gram tidak terdapat pada korpus (1); Skor peringkat tertinggi terlalu kecil (5);
29 (9 kalimat)	1	4	25%	N-gram tidak terdapat pada korpus (1); Skor peringkat tertinggi terlalu kecil (3)
30 (4 kalimat)	2	3	67%	Skor peringkat tertinggi terlalu kecil (3)
31 (8 kalimat)	0	3	0%	Skor peringkat tertinggi terlalu kecil (3)
33 (10 kalimat)	0	5	0%	N-gram tidak terdapat pada korpus (1); Skor peringkat tertinggi terlalu kecil (4)
34 (6 kalimat)	0	1	0%	Skor peringkat tertinggi terlalu kecil (1)
35 (5 kalimat)	0	3	0%	Skor peringkat tertinggi terlalu kecil (3)
36 (5 kalimat)	0	3	0%	Skor peringkat tertinggi terlalu kecil (3)
37 (9 kalimat)	2	4	50%	Skor peringkat tertinggi terlalu kecil (2)
38 (9 kalimat)	0	2	0%	N-gram tidak terdapat pada korpus (1); Skor peringkat tertinggi terlalu kecil (1)
39 (10 kalimat)	0	6	0%	N-gram tidak terdapat pada korpus (2); Skor peringkat tertinggi terlalu kecil (4)
40 (6 kalimat)	0	4	0%	Skor peringkat tertinggi terlalu kecil (4)
41 (5 kalimat)	1	4	25%	N-gram tidak terdapat pada korpus (1); Skor peringkat tertinggi terlalu kecil (3)
42 (6 kalimat)	0	5	0%	Skor peringkat tertinggi terlalu kecil (5)
43 (5 kalimat)	1	4	25%	N-gram tidak terdapat pada korpus (1); Skor peringkat tertinggi terlalu kecil (2)
44 (7 kalimat)	0	6	0%	N-gram tidak terdapat pada korpus (2); Skor peringkat tertinggi terlalu kecil (4)
45 (5 kalimat)	0	3	0%	N-gram tidak terdapat pada korpus (1); Skor peringkat tertinggi terlalu kecil (2);
47 (6 kalimat)	0	4	0%	N-gram tidak terdapat pada korpus (1); Skor peringkat tertinggi terlalu kecil (3)
48 (7 kalimat)	1	3	33%	Skor peringkat tertinggi terlalu kecil (2)
49 (5 kalimat)	0	2	0%	N-gram tidak terdapat pada korpus (1); Skor peringkat tertinggi terlalu kecil (1)
50 (6 kalimat)	2	5	40%	Skor peringkat tertinggi terlalu kecil (3)
51 (9 kalimat)	0	2	0%	N-gram tidak terdapat pada korpus (1); Skor peringkat tertinggi terlalu kecil (1)

Teks (jumlah kalimat)	Jumlah Deteksi		Akurasi	Analisis Kesalahan Deteksi
	Sistem	Manual		
52 (5 kalimat)	0	2	0%	Skor peringkat tertinggi terlalu kecil (2)
53 (5 kalimat)	0	2	0%	N-gram tidak terdapat pada korpus (1); Skor peringkat tertinggi terlalu kecil (1)
54 (6 kalimat)	1	2	50%	Skor peringkat tertinggi terlalu kecil (1)
55 (7 kalimat)	0	4	0%	N-gram tidak terdapat pada korpus (2); Skor peringkat tertinggi terlalu kecil (2)
56 (10 kalimat)	0	4	0%	N-gram tidak terdapat pada korpus (1); Skor peringkat tertinggi terlalu kecil (3)
57 (8 kalimat)	1	5	20%	Skor peringkat tertinggi terlalu kecil (4)
58 (5 kalimat)	0	3	0%	Skor peringkat tertinggi terlalu kecil (3)
59 (5 kalimat)	0	3	0%	N-gram tidak terdapat pada korpus (1); Skor peringkat tertinggi terlalu kecil (2)
	Rata-Rata		11%	

Tabel 4.9 Pengujian Akurasi Koreksi

Kata Salah	Sugesti Koreksi		Status
	Sistem	Manual	
ge	ke	ke	Benar
pertapa	pertama	pertama	Benar
kantar	kantor	kantor	Benar
basal	asal	asal	Benar
dengar	dengan	dengan	Benar
ki	ke	ke	Benar
rukyat	rakyat	rakyat	Benar
berikat	serikat	serikat	Benar
segara	negara	negara	Benar
maab	maaf	maaf	Benar
pepak	sepak	sepak	Benar
dik	di	di	Benar
	Total Benar		100%

4.2.3 Hasil Pengujian

Pengujian sistem yang dilakukan menunjukkan bahwa aplikasi yang yang dibuat untuk mendeteksi dan mengkoreksi kesalahan kata *real-word* sudah memenuhi persyaratan fungsional. Dari Tabel 4.8 analisis hasil pengujian deteksi dan koreksi kesalahan kata *real-word* menggunakan metode n-gram menghasilkan nilai rata-rata akurasi sebesar 11%. Ada dua penyebab mengapa akurasi deteksi yang dihasilkan sangat kecil, yaitu n-gram tidak terdapat pada korpus dan skor peringkat tertinggi terlalu kecil. Berikut adalah penjelasannya.

1. N-gram tidak terdapat pada korpus, ditemukan pada 17 dari 30 teks yang diuji. Kesalahan ini terjadi saat semua bigram dan trigram kata yang dihasilkan tidak ditemukan kemunculannya dalam korpus n-gram, yang menghasilkan peringkat skor yang sama untuk setiap anggota *confusion set*, sehingga kata salah dianggap sebagai benar. Kesalahan ini disebabkan oleh kecilnya ukuran korpus n-gram yang dipakai.
2. Skor peringkat tertinggi terlalu kecil, ditemukan pada 30 dari 30 teks yang diuji. Kesalahan ini terjadi saat skor peringkat tertinggi anggota *confusion set* tidak cukup untuk menyatakan bahwa kata salah adalah salah. Batas suatu kata dinyatakan salah adalah jika skornya 0 atau kurang dari 1% skor peringkat tertinggi. Pada kasus ini skor kata salah lebih besar dari 1% skor peringkat tertinggi sehingga kata yang salah dianggap sebagai benar.

Ada dua penyebab kesalahan ini. Pertama adalah karena kecilnya ukuran korpus yang dipakai. Kedua adalah karena metode *smoothing* yang digunakan mengubah semua skor 0 menjadi bukan 0. Sehingga kata-kata yang harusnya dinyatakan salah karena bernilai 0 menjadi harus dibandingkan dengan batas 1% skor tertinggi, dan karena tidak melewati batas, menjadi dianggap benar.

Untuk mengatasi masalah ini dapat dengan menambah ukuran korpus n-gram yang digunakan. Korpus n-gram bahasa Inggris sudah mencapai 155 juta string berbeda berdasarkan korpus sebesar 520 juta kata [24]. Sedangkan korpus n-gram yang digunakan dalam penelitian ini hanya berukuran total 1 juta 700 ribu string berbeda, sangat kecil jika dibandingkan dengan korpus n-gram bahasa Inggris. Korpus yang berukuran lebih besar akan memuat lebih banyak kemunculan bigram dan trigram, sehingga akan menghasilkan skor maksimum yang lebih tinggi. Selain itu dapat juga dengan mengganti metode *smoothing* yang digunakan. Metode *additive smoothing* yang digunakan ternyata menambah terlalu banyak nilai probabilitas kepada n-gram yang memiliki jumlah kemunculan 0, sehingga skor kata salah tidak dianggap sebagai kesalahan oleh sistem. Dengan mengubah metode *smoothing* diharapkan nilai probabilitas n-gram berjumlah kemunculan 0 tidak berubah terlalu besar sehingga tidak mengganggu perhitungan skor.