BAB 4

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi Sistem

Implementasi merupakan tahap menerjemahkan perancangan berdasarkan hasil analisis. Tujuan implementasi adalah untuk mengkonfirmasikan modul program perancangan pada para pelaku sistem sehingga pengguna dapat memberikan masukkan kepada pengembang sistem.

4.1.1 Perangkat Implementasi Data

Pembuatan *database* dan tabel dilakukan dengan menggunakan aplikasi DBMS MySQL. Implementasi *database* dalam SQL adalah sebagai berikut:

1. Pembuatan *database*

Berikut ini adalah Sintaks SQL untuk pembuatan *database* untuk aplikasi *stemming engine*.

```
1  CREATE DATABASE stemming_dbstemming;
```

2. Pembuatan tabel

Berikut ini adalah daftar tabel untuk stemming engine yang ditulis dalam format .sql:

Tabel 4.1 Pembuatan Tabel

No	Nama Tabel	Nama file
1	penulis	penulis.sql
2	pemeriksa	pemeriksa.sql
3	dokumen	dokumen.sql
4	kata_dasar	kata_dasar.sql
5	stopword	stopword.sql
6	token	token.sql
7	filter	filter.sql
8	stem	stem.sql
9	tidakbaku	tidakbaku.sql

Dari Tabel 4.1 dapat diimplementasikan ke sebuah sintaks SQL. Berikut ini adalah sintakx-sintaks SQL untuk pembuatan tabel:

a. Sintaks penulis.sql

```
1  CREATE TABLE `penulis` (
2    `id_penulis` INT(11) NOT NULL AUTO_INCREMENT,
3    `username` VARCHAR(15) NOT NULL,
4    `password` VARCHAR(41) NOT NULL,
5    `nama` VARCHAR(50) NOT NULL,
6    `instansi` VARCHAR(50) DEFAULT NULL,
7    `alamat` VARCHAR(100) DEFAULT NULL,
8    `no_telepon` VARCHAR(15) NOT NULL,
9    `email` VARCHAR(20) NOT NULL,
10   PRIMARY KEY (`id_penulis`),
11   UNIQUE KEY `username` (`username`),
12   UNIQUE KEY `email` (`email`)
13 ) ENGINE=INNODB AUTO_INCREMENT=19 DEFAULT CHARSET=latin1
```

b. Sintaks pemeriksa.sql

```
1  CREATE TABLE `pemeriksa` (
2    `id_pemeriksa` INT(11) NOT NULL AUTO_INCREMENT,
3    `username` VARCHAR(15) NOT NULL,
4    `password` VARCHAR(15) NOT NULL,
5    `nama` VARCHAR(50) NOT NULL,
6    PRIMARY KEY (`id_pemeriksa`),
7    UNIQUE KEY `username` (`username`)
8  ) ENGINE=INNODB AUTO_INCREMENT=2 DEFAULT CHARSET=latin1
```

c. Sintaks dokumen.sql

```
1  CREATE TABLE `dokumen` (
2    `id_dokumen` INT(11) NOT NULL AUTO_INCREMENT,
3    `judul` VARCHAR(50) NOT NULL,
4    `nama_file` VARCHAR(50) NOT NULL,
5    `tipe` VARCHAR(15) NOT NULL,
6    `ukuran` INT(11) NOT NULL,
7    `tanggal_unggah` DATETIME NOT NULL,
8    `status` CHAR(1) NOT NULL,
9    `id_penulis` INT(11) NOT NULL,
10   `id_pemeriksa` INT(11) DEFAULT NULL,
11   PRIMARY KEY (`id_dokumen`),
12   KEY `FK_dokumen` (`id_penulis`),
13   KEY `FK_dokumen2` (`id_pemeriksa`),
14   CONSTRAINT `FK_dokumen` FOREIGN KEY (`id_penulis`)
15 REFERENCES `penulis` (`id_penulis`) ON DELETE CASCADE ON
16 UPDATE CASCADE,
17   CONSTRAINT `FK_dokumen2` FOREIGN KEY (`id_pemeriksa`)
18 REFERENCES `pemeriksa` (`id_pemeriksa`) ON DELETE CASCADE
19 ON UPDATE CASCADE
20 ) ENGINE=INNODB AUTO_INCREMENT=9 DEFAULT CHARSET=latin1
```

d. Sintaks kata_dasar.sql

```
1  CREATE TABLE `kata_dasar` (
2    `id_katdas` INT(11) NOT NULL AUTO_INCREMENT,
3    `kata` VARCHAR(50) NOT NULL,
4    `id_pemeriksa` INT(11) NOT NULL,
5    PRIMARY KEY (`id_katdas`),
6    KEY `FK_kata_dasar` (`id_pemeriksa`)
7  ) ENGINE=INNODB AUTO_INCREMENT=28526 DEFAULT
8  CHARSET=latin1
```

e. Sintaks stopword.sql

```
1  CREATE TABLE `stopword` (
2    `id_stopword` INT(11) NOT NULL AUTO_INCREMENT,
3    `kata_stopword` VARCHAR(30) NOT NULL,
4    `id_pemeriksa` INT(11) NOT NULL,
5    PRIMARY KEY (`id_stopword`),
6    KEY `FK_stopword` (`id_pemeriksa`),
7    CONSTRAINT `FK_stopword` FOREIGN KEY (`id_pemeriksa`)
8  REFERENCES `pemeriksa` (`id_pemeriksa`)
9  ) ENGINE=INNODB AUTO_INCREMENT=21 DEFAULT CHARSET=latin1
```

f. Sintaks token.sql

```
CREATE TABLE `token` (
  `id_token` INT(11) NOT NULL AUTO_INCREMENT,
  `kata_token` VARCHAR(30) NOT NULL,
  `id_dokumen` INT(11) NOT NULL,
  PRIMARY KEY (`id_token`),
  KEY `FK_token` (`id_dokumen`),
  CONSTRAINT `FK_token` FOREIGN KEY (`id_dokumen`)
REFERENCES `dokumen` (`id_dokumen`) ON DELETE CASCADE ON
UPDATE CASCADE
) ENGINE=INNODB AUTO_INCREMENT=9 DEFAULT CHARSET=latin1
```

g. Sintaks filter.sql

```
CREATE TABLE `filter` (
  `id_filter` INT(11) NOT NULL AUTO_INCREMENT,
  `kata_filter` VARCHAR(30) NOT NULL,
  `id_token` INT(11) NOT NULL,
  PRIMARY KEY (`id_filter`),
  KEY `FK_filter` (`id_token`),
  CONSTRAINT `FK_filter` FOREIGN KEY (`id_token`)
REFERENCES `token` (`id_token`) ON DELETE CASCADE ON
UPDATE CASCADE
) ENGINE=INNODB DEFAULT CHARSET=latin1
```

h. Sintaks stem.sql

```
CREATE TABLE `stem` (
  `id_stem` INT(11) NOT NULL AUTO_INCREMENT,
  `kata_stem` VARCHAR(30) NOT NULL,
  `id_filter` INT(11) NOT NULL,
  PRIMARY KEY (`id_stem`),
  KEY `FK_stem` (`id_filter`),
  CONSTRAINT `FK_stem` FOREIGN KEY (`id_filter`)
REFERENCES `filter` (`id_filter`) ON DELETE CASCADE ON
UPDATE CASCADE
) ENGINE=INNODB DEFAULT CHARSET=latin1
```

i. Sintaks tidak_baku.sql

```
CREATE TABLE `tidakbaku` (
  `id_tidakbaku` INT(11) NOT NULL AUTO_INCREMENT,
  `kata_tidakbaku` VARCHAR(50) DEFAULT NULL,
  `id_filter` INT(11) DEFAULT NULL,
  PRIMARY KEY (`id_tidakbaku`),
  KEY `FK_tidakbaku` (`id_filter`),
  CONSTRAINT `FK_tidakbaku` FOREIGN KEY (`id_filter`)
REFERENCES `filter` (`id_filter`) ON DELETE CASCADE ON
UPDATE CASCADE
) ENGINE=INNODB DEFAULT CHARSET=latin1
```

4.1.2 Implementasi Antarmuka

Implementasi antarmuka dilakukan dengan menampilkan setiap tampilan sistem yang dibangun dan pengkodeannya dalam bentuk *file* program. Adapun untuk tampilan implementasi aplikasi *stemming engine* akan dilampirkan pada halaman lampiran. Deskripsi implementasi antarmuka untuk pengguna akan dijelaskan pada

1. Implementasi antarmuka penulis

Tabel 4.2 Implementasi antarmuka penulis

Menu	Proses ke-	Nama File
Halaman registrasi	Proses 1	registrasi.php
		library.php
		session_start.php
		koneksi.php
		style.css
Halaman login penulis	Proses 1	index.php
		session_start.php
		style.css
Halaman lupa kata sandi	Proses 1	lupa_password.php
		library.php
		session_start.php
		koneksi.php
		style.css
Halaman utama penulis	Proses 2	home.php
		session_start.php
		koneksi.php
		library.php
		style.css
Halaman daftar dokumen	Proses 2	dokumen_daftar.php
		session_start.php
		koneksi.php
		library.php
		style.css
Halaman profil	Proses 1	profil.php
		session_start.php
		koneksi.php
		library.php
		style.css

2. Implementasi antarmuka pemeriksa

Tabel 4.3 Implementasi antarmuka pemeriksa

Menu	Proses ke-	Nama File
Halaman login pemeriksa	Proses 1	index.php
		session_start.php
		koneksi.php
		library.php
		style.css
Halaman utama pemeriksa	Proses 2	pemeriksa_home.php
		session_start.php
		koneksi.php
		library.php
		style.css
Halaman pengolahan data kata dasar	Proses 3	daftar_katdas.php
		session_start.php
		koneksi.php
		library.php
		style.css
Halaman pengolahan data stopword	Proses 3	daftar_stopword.php
		session_start.php
		koneksi.php
		library.php
		style.css
Halaman hasil *tokenizing*	Proses 4	tokenizing_hasil.php
		session_start.php
		koneksi.php
		library.php
		style.css
		lib_stemming.php
Halaman hasil *filtering*	Proses 4	filtering_hasil.php
		session_start.php
		koneksi.php
		library.php
		style.css
		lib_stemming.php
Halaman hasil *stemming*	Proses 4	stemming_hasil.php
		session_start.php
		koneksi.php
		library.php
		style.css
		lib_stemming.php

Menu	Proses ke-	Nama File
Halaman daftar kata tidak baku	Proses 4	tidakbaku_hasil.php
		session_start.php
		koneksi.php
		library.php
		style.css
		lib_stemming.php
Halaman hasil pemeriksaan	Proses 4	tidakbaku_hasil.php
		session_start.php
		koneksi.php
		library.php
		style.css
		lib_stemming.php
Halaman *history* hasil *tokenizing*	Proses 4	view_tokenizing_hasil.php
		session_start.php
		koneksi.php
		library.php
		style.css
		lib_stemming.php
Halaman *history* hasil *filtering*	Proses 4	view_filtering_hasil.php
		session_start.php
		koneksi.php
		library.php
		style.css
		lib_stemming.php
Halaman *history* hasil *stemming*	Proses 4	view_stemming_hasil.php
		session_start.php
		koneksi.php
		library.php
		style.css
		lib_stemming.php
Halaman *history* daftar kata tidak baku	Proses 4	view_tidakbaku_hasil.php
		session_start.php
		koneksi.php
		library.php
		style.css
		lib_stemming.php

4.2 Pengujian Sistem

Pengujian yang digunakan untuk menguji sistem *stemming engine* ini adalah pendekatan pengujian *black box* dan pengujian *beta*.

4.2.1 Rencana Pengujian

Rencana pengujian bertujuan untuk mendaftarkan proses-proses yang akan diuji dan pendekatan pengujiannya. Untuk lebih jelasnya dapat dilihat pada Tabel 4.4.

Tabel 4.4 Rencana Pengujian

Menu Uji	Butir Uji	Jenis Pengujian
Login	Verifikasi nama pengguna	*Black Box*
	Verifikasi kata sandi	*Black Box*
Registrasi penulis	Registrasi	*Black Box*
Pengolahan data dokumen	Tambah dokumen	*Black Box*
	Ubah dokumen	*Black Box*
	Cari dokumen	*Black Box*
Pengolahan data kata dasar	Tambah kata dasar	*Black Box*
	Ubah kata dasar	*Black Box*
	Cari kata dasar	*Black Box*
Pengolahan data stopword	Tambah stopword	*Black Box*
	Ubah stopword	*Black Box*
	Cari stopword	*Black Box*
Pemeriksaan dokumen	*Stemming*	*Black Box*

4.2.2 Pengujian Black Box

Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak. Disebut juga pengujian *behavioral* atau pengujian partisi. Pengujian *black box* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian *input* yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu

program. Untuk menguji aplikasi stemming engine ini adalah menggunakan *Equivalence Partitioning*, *Cause Effect Relationship* dan *Performance Testing*.

1. *Equivalence Partitioning*

Untuk melakukan pengujian *black box equivalence partitioning* harus terlebih dahulu menentukan *Valid Class* dan *Invalid Class*. Berikut ini adalah menu – menu yang akan diuji:

a. Registrasi

Terdiri dari beberapa masukan, yaitu:

1) Nama pengguna

Valid Class 1 : [A..Z | a..z | 0..9]

Valid Class 2 : [A..Z]

Valid Class 3 : [a..z]

Valid Class 4 : [0..9]

Invalid Class : bukan [A..Z | a..z | 0..9]

Tabel 4.5 Kasus Dan Hasil Uji Registrasi (Nama Pengguna)

Kasus Dan Hasil Uji Registrasi (Nama Pengguna)				
Valid Class	Data Masukan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
1	Apri123	Tidak ada pesan *error*	Tidak ada pesan *error*	[√] Diterima [ ] Ditolak
2	HAZEE	Tidak ada pesan *error*	Tidak ada pesan *error*	[√] Diterima [ ] Ditolak
3	hazee	Tidak ada pesan *error*	Tidak ada pesan *error*	[√] Diterima [ ] Ditolak
4	12345	Tidak ada pesan *error*	Tidak ada pesan *error*	[√] Diterima [ ] Ditolak
Invalid Class	!@#$	Menampilkan pesan : masukan username dengan benar! co : udin12	Menampilkan pesan : masukan username dengan benar! co : udin12	[√] Diterima [ ] Ditolak

2) Kata sandi

Valid Class 1 : [A..Z | a..z | 0..9]

Valid Class 2 : [A..Z]

Valid Class 3 : [a..z]

Valid Class 4 : [0..9]

Invalid Class : bukan [A..Z | a..z | 0..9]

Tabel 4.6 Kasus Dan Hasil Uji Registrasi (Kata Sandi)

Kasus Dan Hasil Uji Registrasi (Kata Sandi)				
Valid Class	Data Masukan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
1	Apri123	Tidak ada pesan *error*	Tidak ada pesan *error*	[√] Diterima [ ] Ditolak
2	HAZEE	Tidak ada pesan *error*	Tidak ada pesan *error*	[√] Diterima [ ] Ditolak
3	hazee	Tidak ada pesan *error*	Tidak ada pesan *error*	[√] Diterima [ ] Ditolak
4	12345	Tidak ada pesan *error*	Tidak ada pesan *error*	[√] Diterima [ ] Ditolak
Invalid Class	!@#$	Menampilkan pesan : masukan kata sandi dengan benar! co : udin12	Menampilkan pesan : masukan kata sandi dengan benar! co : udin12	[√] Diterima [ ] Ditolak

3) Nama

Valid Class 1 : [A..Z | a..z]

Valid Class 2 : [A..Z]

Valid Class 3 : [a..z]

Invalid Class : bukan [A..Z | a..z]

Tabel 4.7 Kasus Dan Hasil Uji Registrasi (Nama)

Kasus Dan Hasil Uji Registrasi (Nama)				
Valid Class	Data Masukan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
1	Jajang	Tidak ada pesan *error*	Tidak ada pesan *error*	[√] Diterima [ ] Ditolak
2	JAJANG	Tidak ada pesan *error*	Tidak ada pesan *error*	[√] Diterima [ ] Ditolak
3	jajang	Tidak ada pesan *error*	Tidak ada pesan *error*	[√] Diterima [ ] Ditolak
Invalid Class	!@#$	Menampilkan pesan : masukan kata sandi dengan benar! co : udin12	Menampilkan pesan : masukan kata sandi dengan benar! co : udin12	[√] Diterima [ ] Ditolak

4) No Telepon

Valid Class : [0..9]

Invalid Class : bukan [0..9]

Tabel 4.8 Kasus Dan Hasil Uji Registrasi (No Telepon)

Kasus Dan Hasil Uji Registrasi (No Telepon)				
	Data Masukan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
Valid Class	085210727451	Tidak ada pesan *error*	Tidak ada pesan *error*	[√] Diterima [ ] Ditolak
Invalid Class	notelepon	Menampilkan pesan : masukan no telepon dengan benar! co : 085610722222	Menampilkan pesan : masukan no telepon dengan benar! co : 085610722222	[√] Diterima [ ] Ditolak

b. Tambah dokumen

Terdiri dari dua masukan, yaitu:

1) Judul :

Valid Class 1 : [A..Z | a..z | 0..9]

Valid Class 2 : [A..Z]

Valid Class 3 : [a..z]

Valid Class 4 : [0..9]

Invalid Class : bukan [A..Z | a..z | 0..9]

Tabel 4.9 Kasus Dan Hasil Uji Tambah Dokumen(Judul)

Kasus Dan Hasil Uji Tambah Dokumen (Judul)				
Valid Class	Data Masukan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
1	Sample1	Tidak ada pesan *error*	Tidak ada pesan *error*	[√] Diterima [ ] Ditolak
2	SAMPLE	Tidak ada pesan *error*	Tidak ada pesan *error*	[√] Diterima [ ] Ditolak
3	sample	Tidak ada pesan *error*	Tidak ada pesan *error*	[√] Diterima [ ] Ditolak
4	12345	Tidak ada pesan *error*	Tidak ada pesan *error*	[√] Diterima [ ] Ditolak
Invalid Class	!@#$	Menampilkan pesan : Hanya bias menggunakan karakter A-Z, a-z, 0-9.	Menampilkan pesan : Hanya bias menggunakan karakter A-Z, a-z, 0-9.	[√] Diterima [ ] Ditolak

2) Dokumen

Valid Class : format file .txt

Invalid Class : bukan format .txt

Tabel 4.10 Kasus dan Hasil Uji Tambah Dokumen (Dokumen)

Kasus Dan Hasil Uji Tambah Dokumen (Dokumen)				
	Data Masukan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
Valid Class	Sample1.txt	Menampilkan pesan: Dokumen berhasil diunggah.	Menampilkan pesan: Dokumen berhasil diunggah.	[√] Diterima [ ] Ditolak
Invalid Class	Sample1.pdf	Menampilkan pesan : dokumen harus berformat .txt	Menampilkan pesan : dokumen harus berformat .txt	[√] Diterima [ ] Ditolak

c. Ubah Dokumen

Terdiri dari satu masukan, yaitu:

Judul:

Valid Class 1 : [A..Z | a..z | 0..9]

Valid Class 2 : [A..Z]

Valid Class 3 : [a..z]

Valid Class 4 : [0..9]

Invalid Class : bukan [A..Z | a..z | 0..9]

Tabel 4.11 Kasus Dan Hasil Uji Ubah Dokumen (Judul)

Kasus Dan Hasil Uji Ubah Dokumen (Judul)				
Valid Class	Data Masukan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
1	Sample1	Tidak ada pesan *error*	Tidak ada pesan *error*	[√] Diterima [ ] Ditolak
2	SAMPLE	Tidak ada pesan *error*	Tidak ada pesan *error*	[√] Diterima [ ] Ditolak
3	sample	Tidak ada pesan *error*	Tidak ada pesan *error*	[√] Diterima [ ] Ditolak
4	12345	Tidak ada pesan *error*	Tidak ada pesan *error*	[√] Diterima [ ] Ditolak
Invalid Class	!@#$	Menampilkan pesan : Hanya bias menggunakan karakter A-Z, a-z, 0-9.	Menampilkan pesan : Hanya bias menggunakan karakter A-Z, a-z, 0-9.	[√] Diterima [ ] Ditolak

d. Cari Dokumen

Terdiri dari satu masukan, yaitu:

Kata kunci:

Valid Class 1 : [A..Z | a..z | 0..9]

Valid Class 2 : [A..Z]

Valid Class 3 : [a..z]

Valid Class 4 : [0..9]

Invalid Class : bukan [A..Z | a..z | 0..9]

Tabel 4.12 Kasus Dan Hasil Uji Cari Dokumen

Kasus Dan Hasil Uji Cari Dokumen				
Valid Class	Data Masukan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
1	Sample1	Menampilkan Data yang dicari	Menampilkan Data yang dicari	[√] Diterima [ ] Ditolak
2	SAMPLE	Menampilkan Data yang dicari	Menampilkan Data yang dicari	[√] Diterima [ ] Ditolak
3	sample	Menampilkan Data yang dicari	Menampilkan Data yang dicari	[√] Diterima [ ] Ditolak
4	12345	Menampilkan Data yang dicari	Menampilkan Data yang dicari	[√] Diterima [ ] Ditolak
Invalid Class	!@#$	Menampilkan pesan : Hanya bias menggunakan karakter A-Z, a-z, 0-9.	Menampilkan pesan : Hanya bias menggunakan karakter A-Z, a-z, 0-9.	[√] Diterima [ ] Ditolak

e. Tambah Kata Dasar

Terdiri dari satu masukan, yaitu:

Kata Dasar:

Valid Class 1 : [A..Z | a..z]

Valid Class 2 : [A..Z]

Valid Class 3 : [a..z]

Invalid Class : bukan [A..Z | a..z]

Tabel 4.13 Kasus Dan Hasil Uji Tambah Kata Dasar

Kasus Dan Hasil Uji Tambah Kata Dasar				
Valid Class	Data Masukan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
1	jalan	Tidak ada pesan *error*	Tidak ada pesan *error*	[√] Diterima [ ] Ditolak
2	JALAN	Tidak ada pesan *error*	Tidak ada pesan *error*	[√] Diterima [ ] Ditolak
Invalid Class	!@#$	Menampilkan pesan : Hanya bias menggunakan karakter A-Z, a-z.	Menampilkan pesan : Hanya bias menggunakan karakter A-Z, a-z.	[√] Diterima [ ] Ditolak

f. Ubah Kata Dasar

Terdiri dari satu masukan, yaitu:

Kata Dasar:

Valid Class 1 : [A..Z | a..z]

Valid Class 2 : [A..Z]

Valid Class 3 : [a..z]

Invalid Class : bukan [A..Z | a..z]

Tabel 4.14 Kasus Dan Hasil Uji Tambah Kata Dasar

Kasus Dan Hasil Uji Tambah Kata Dasar				
Valid Class	Data Masukan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
1	baru	Tidak ada pesan *error*	Tidak ada pesan *error*	[√] Diterima [ ] Ditolak
2	BARU	Tidak ada pesan *error*	Tidak ada pesan *error*	[√] Diterima [ ] Ditolak
Invalid Class	!@#$	Menampilkan pesan : Hanya bias menggunakan karakter A-Z, a-z.	Menampilkan pesan : Hanya bias menggunakan karakter A-Z, a-z.	[√] Diterima [ ] Ditolak

g. Cari Kata Dasar:

Terdiri dari satu masukan, yaitu:

Kata kunci:

Valid Class 1 : [A..Z | a..z | 0..9]

Valid Class 2 : [A..Z]

Valid Class 3 : [a..z]

Valid Class 4 : [0..9]

Invalid Class : bukan [A..Z | a..z | 0..9]

Tabel 4.15 Kasus Dan Hasil Uji Cari Dokumen

Kasus Dan Hasil Uji Cari Dokumen				
Valid Class	Data Masukan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
1	Sample1	Menampilkan Data yang dicari	Menampilkan Data yang dicari	[√] Diterima [ ] Ditolak
2	SAMPLE	Menampilkan Data yang dicari	Menampilkan Data yang dicari	[√] Diterima [ ] Ditolak
3	sample	Menampilkan Data yang dicari	Menampilkan Data yang dicari	[√] Diterima [ ] Ditolak
4	12345	Menampilkan Data yang dicari	Menampilkan Data yang dicari	[√] Diterima [ ] Ditolak
Invalid Class	!@#$	Menampilkan pesan : Hanya bias menggunakan karakter A-Z, a-z, 0-9.	Menampilkan pesan : Hanya bias menggunakan karakter A-Z, a-z, 0-9.	[√] Diterima [ ] Ditolak

2. *Cause Effect Relationship Testing*

Teknik ini merupakan suplemen dari pengujian sebelumnya yaitu *equivalence partitioning* dengan menyediakan cara untuk memilih kombinasi data input. Melibatkan kondisi input(*cause*) dan kondisi output(*effect*) untuk mencegah pendefinisian kasus uji yang terlalu banyak.

Berikut ini adalah menu-menu yang akan dilakukan pengujian dengan menggunakan pengujian *cause effect relationship testing*:

a. Menu Login

Tabel 4.16 Pengujian *Cause Effect* Menu Login

Menu Login	
Input States	*Output States*
1. Nama Pengguna diisi benar	A. Masuk halaman utama
2. Kata Sandi diisi benar	B. Nama pengguna masih kosong
	C. Kata sandi masih kosong

Tabel 4.17 Hasil Pengujian *Cause Effect* Menu Login

Cause/Effect	1	2	3
1	1	0	1
2	1	1	0
A	1	0	0
B	0	1	0
C	0	0	1

b. Menu Registrasi

Tabel 4.18 Pengujian *Cause Effect* Menu Registrasi

Menu Registrasi	
Input States	*Output States*
1. Nama Pengguna diisi benar	A. Simpan ke tabel penulis
2. Kata Sandi diisi benar	B. Nama Pengguna masih kosong
3. Nama diisi dengan benar	C. Kata sandi masih kosong
4. No Telepon diisi dengan benar	D. Nama masih kosong
5. Email diisi dengan benar	E. No Telepon masih kosong
	F. Email masih kosong

Tabel 4.19 Hasil Pengujian *Cause Effect* Menu Registrasi

Cause/Effect	1	2	3	4	5
1	1	1	1	1	1
2	1	0	1	1	1
3	1	0	0	1	1
4	1	0	0	0	1
5	1	0	0	0	0
A	1	0	0	0	0
B	0	1	0	0	0
C	0	1	1	0	0
D	0	1	1	0	0
E	0	1	1	1	0
F	0	1	1	1	1

c. Menu Tambah Dokumen

Tabel 4.20 Pengujian *Cause Effect* Menu Tambah Dokumen

Menu Tambah Dokumen	
Input States	*Output States*
1. Judul diisi benar	A. Simpan ke tabel dokumen
2. Dokumen diisi benar	B. Judul salah
	C. Dokumen salah

Tabel 4.21 Hasil Pengujian *Cause Effect* Menu Tambah Dokumen

Cause/Effect	1	2	3
1	1	0	1
2	1	1	0
A	1	0	0
B	0	1	0
C	0	0	1

d. Menu Tambah Stopword

Tabel 4.22 Pengujian *Cause Effect* Menu Tambah Stopword

Menu Tambah Stopword	
Input States	*Output States*
1. Kata Stopword diisi benar	A. Simpan ke tabel stopword
	B. Stopword salah

Tabel 4.23 Hasil Pengujian *Cause Effect* Menu Tambah Stopword

Cause/Effect	1	2
1	1	0
A	1	0
B	0	1

e. Menu Tambah Kata Dasar

Tabel 4.24 Pengujian *Cause Effect* Menu Tambah Kata Dasar

Menu Tambah Kata Dasar	
Input States	*Output States*
1. Kata kata dasar diisi benar	A. Simpan ke tabel kata dasar
	B. Kata dasar salah

Tabel 4.25 Hasil Pengujian *Cause Effect* Menu Tambah Kata Dasar

Cause/Effect	1	2
1	1	0
A	1	0
B	0	1

3. *Performance Testing*

Pengujian ini dilakukan untuk mengetahui sejauh mana kecepatan dari aplikasi yang telah dibangun. Pengujian ini dilakukan dengan menggunakan perangkat keras sebagai berikut:

Tabel 4.26 Spesifikasi Perangkat Keras Untuk *Performance Testing*

No.	Perangkat Keras	Spesifikasi
1	Processor	Core i5 Processor 460M : 2.53 GHz
2	Display	14" 16:9 HD (1366x768) LED backlight
3	VGA	1 GB
4	Harddisk	640 GB
5	Memory	Memori DDR3 2GB

Pengujian kecepatan proses *stemming* dilakukan pada lima jenis dokumen yang berbeda-beda. Jenis dokumen yang dilakukan pengujian dibedakan dengan jumlah kata yang terdapat dalam dokumen tersebut.

Tabel 4.27 Hasil Pengujian *Performance Testing*

Pengujian ke-	Nama Dokumen	Jumlah Kata	Waktu *Stemming* (detik)
1	Sample 1.txt	10	2,26
2			2,32
3			2,31
4			2,31
5			2,39
1	Sample 2.txt	15	3,38
2			3,19
3			3,20
4			3,21
5			3,24
1	Sample 3.txt	20	3,98
2			3,66
3			3,90
4			3,50
5			3,89
1	Sample 4.txt	25	4,99
2			4,70
3			4,84
4			4,62
5			4,70
1	Sample 5.txt	30	5,97
2			5,81
3			5,73
4			5,86
5			5,66

Dari Tabel 4.27 dapat disimpulkan bahwa untuk melakukan proses stemming pada aplikasi yang telah dibuat jika jumlah kata pada sebuah dokumen semakin banyak maka semakin banyak pula waktu yang diperlukan untuk melakukan proses stemming.

4.2.3 Pengujian Beta

Pengujian Beta merupakan pengujian yang dilakukan dengan ada interaksi langsung antara pengguna sebenarnya dan perangkat lunak. Cara yang dilakukan dalam penelitian ini adalah dengan dua cara, yaitu melakukan wawancara pada pemeriksa dan menyebarkan kuesioner pada penulis. Pengujian dalam bentuk kuesioner ini yang terdiri dari 5 pertanyaan dan saran untuk setiap kuesioner (kuesioner dilampirkan).

4.2.3.1 Wawancara untuk pemeriksa

Wawancara kepada pihak pemeriksa, untuk saat ini di Balai Bahasa Bandung hanya terdapat satu pemeriksa. Berikut ini adalah hasil dari wawancara:

Tabel 4.28 Hasil Wawancara Dengan Pemeriksa

NO	PERTANYAAN	JAWABAN				
		Sangat Setuju	Setuju	Cukup Setuju	Kurang Setuju	Tidak Setuju
1	Apakah tampilan aplikasi ini terlihat menarik?		√			
2	Apakah aplikasi ini mudah digunakan?		√			
3.	Apakah aplikasi ini dapat membantu proses pemeriksaan dokumen yang mengandung kata tidak baku?	√				
4.	Apakah aplikasi ini dapat mempercepat proses pemeriksaan dokumen?	√				
5.	Apakah Aplikasi ini dapat membantu pendistribusian dokumen dari penulis?	√				

Berdasarkan hasil wawancara dengan pemeriksa dapat disimpulkan bahwa aplikasi ini memiliki tampilan yang menarik, mudah digunakan, sangat membantu proses pemeriksaan, mempercepat proses pemeriksaan dan membantu pendistribusian dokumen dari penulis ke Balai Bahasa Bandung.

4.2.3.2 Kuesioner untuk penulis

Berikut adalah pertanyaan – pertanyaan yang diajukan kepada penulis:

1. Apakah aplikasi ini mudah dipelajari dan digunakan?
    a. Sangat mudah
    b. Mudah
    c. Cukup mudah
    d. Tidak mudah
    e. Sangat tidak mudah
2. Apakah tampilan aplikasi terlihat menarik?
    a. Sangat menarik
    b. Menarik
    c. Cukup menarik
    d. Tidak menarik
    e. Sangat tidak menarik
3. Apakah aplikasi ini interaktif?
    a. Sangat interaktif
    b. Interaktif
    c. Cukup interaktif

d. Tidak interaktif

e. Sangat tidak interaktif

4. Apakah aplikasi ini membantu menampilkan informasi yang dibutuhkan?

   a. Sangat membantu

   b. Membantu

   c. Cukup membantu

   d. Tidak membantu

   e. Sangat tidak membantu

5. Aplikasi ini membantu pendistribusian dokumen karya penulis ?

   a. Sangat membantu

   b. Membantu

   c. Cukup membantu

   d. Tidak membantu

   e. Sangat tidak membantu

4.2.4 Kesimpulan Pengujian

Kesimpulan pengujian merupakan kesimpulan dari masing-masing pendekatan pengujian, dalam penelitian ini menggunakan dua pendekatan pengujian yaitu:

1. Kesimpulan Pengujian *Black Box*
2. Kesimpulan Pengujian *Beta*

4.2.4.1 Kesimpulan Pengujian *Black Box*

1. *Equivalence partitioning*

Berdasarkan hasil pengujian *black box* dengan pendekatan *equivalence partitioning* dengan kasus uji diatas dapat ditarik kesimpulan bahwa pembangunan perangkat lunak *stemming engine* bahasa Indonesia ini bebas dari kesalahan penulisan dan secara fungsional mengeluarkan hasil yang sesuai dengan yang diharapkan.

2. *Cause Effect Relationship*

Berdasarkan hasil pengujian *cause effect eelationship* aplikasi yang telah dibangun fungsionalitasnya sudah sesuai dengan yang diharapkan.

3. *Performance Testing*

Berdasarkan hasil pengujian *performance testing* dapat disimpulkan bahwa untuk melakukan proses stemming pada aplikasi yang telah dibuat jika jumlah kata pada sebuah dokumen semakin banyak maka semakin banyak pula waktu yang diperlukan untuk melakukan proses stemming.

4.2.4.2 Kesimpulan Pengujian *Beta*

Pengujian beta ini dilakukan dengan cara melakukan pengujian langsung pada pengguna yang terlibat dengan perangkat lunak ini. Penelitian ini dilakukan di dua lingkungan:

1. Lingkungan penulis

Penelitian di lingkungan penulis bertujuan untuk mengetahui sejauh mana sistem yang telah dibangun dapat memberikan kemudahan untuk mendistribusikan dokumen hasil karya penulis ke Balai Bahasa Bandung.

2. Lingkungan pemeriksa

Penelitian di lingkungan pemeriksa bertujuan untuk mengetahui sejauh mana aplikasi yang telah dibangun dapat membantu proses pemeriksaan dokumen yang mengandung kata tidak baku dari hasil karya para penulis.

Berdasarkan data hasil kusioner tersebut, dapat dicari presentase masing-masing jawaban dengan menggunakan rumus : Y =(P/Q)* 100%

Keterangan :

P = Total Skor

Q = Skor Tertinggi

Y = Nilai persentase

Untuk mengukur sikap dari pengguna aplikasi yang telah dibangun bisa menggunakan skala likert. Skala Likert merupakan skala yang digunakan untuk mengukur persepsi, sikap atau pendapat seseorang atau kelompok mengenai sebuah peristiwa atau fenomena sosial, berdasarkan definisi operasional yang telah ditetapkan oleh peneliti.

Berikut ini adalah skor yang diberikan untuk jawaban kuesioner yang telah diberikan kepada penelitian para penulis:

1. Jawaban a diberi nilai 5
2. Jawaban b diberi nilai 4
3. Jawaban c diberi nilai 3
4. Jawaban d diberi nilai 2
5. Jawaban e diberi nilai 1

Untuk mengetahui interpretasi skor hasil perhitungan dapat dilihat pada tabel sebagai berikut:

Tabel 4.29 Interpretasi Skor Perhitungan

Nilai	Keterangan
0% – 20%	Sangat Tidak
21% – 40%	Tidak
41% – 60%	Cukup
61% – 80%	Standar
81% – 100%	Sangat

Berikut ini adalah hasil persentase masing-masing jawaban yang sudah dihitung nilainya dengan menggunakan rumus diatas. Kuisioner ini diujikan kepada 15 orang. Berikut ini adalah pertanyaan untuk kesioner:

1. Apakah aplikasi ini mudah dipelajari dan digunakan?

Tabel 4.30 Persentase Kuesioner Pertanyaan 1

Pertanyaan	No	Keterangan	Responden	Skor	Persentase (%)
1	1.	Sangat Mudah	4	20	(61/75)*100=81,3
	2.	Mudah	8	32	
	3.	Cukup Mudah	3	9	
	4.	Tidak Mudah	0	0	
	5.	Sangat tidak mudah	0	0	
Jumlah			15	61	

Hasil presentase dari Tabel 4.30 adalah sebanyak 81,3%. Nilai perhitungan tersebut berarti aplikasi yang telah dibangun masuk ke kriteria interpretasi sangat mudah dipelajari dan digunakan.

2. Apakah tampilan aplikasi terlihat menarik?

Tabel 4.31 Persentase Kuesioner Pertanyaan 2

Pertanyaan	No	Keterangan	Responden	Skor	Persentase (%)
2	1.	Sangat menarik	0	0	(52/75)*100=69,3
	2.	Menarik	7	28	
	3.	Cukup menarik	8	24	
	4.	Tidak mudah	0	0	
	5.	Sangat tidak mudah	0	0	
Jumlah			15	52	

Hasil presentase dari Tabel 4.31 adalah sebanyak 69,3%. Nilai perhitungan tersebut berarti tampilan aplikasi yang telah dibangun masuk ke kriteria interpretasi standar atau cukup menarik.

3. Apakah aplikasi ini interaktif?

Tabel 4.32 Persentase Kuesioner Pertanyaan 3

Pertanyaan	No	Keterangan	Responden	Skor	Persentase (%)
3	1.	Sangat interaktif	0	0	(49/75)*100=65,3
	2.	interaktif	4	16	
	3.	Cukup interaktif	11	33	
	4.	Tidak interaktif	0	0	
	5.	Sangat tidak interaktif	0	0	
Jumlah			15	49	

Hasil presentase dari Tabel 4.32 adalah sebanyak 65,3%. Nilai perhitungan tersebut berarti aplikasi yang telah dibangun masuk ke kriteria interpretasi standar atau cukup interaktif.

4. Apakah aplikasi ini membantu menampilkan informasi yang dibutuhkan?

Tabel 4.33 Persentase Kuesioner Pertanyaan 4

Pertanyaan	No	Keterangan	Responden	Skor	Persentase (%)
4	1.	Sangat membantu	3	15	(61/75)*100=81,3
	2.	membantu	10	40	
	3.	Cukup membantu	2	6	
	4.	Tidak membantu	0	0	
	5.	Sangat tidak membantu	0	0	
Jumlah			15	61	

Hasil presentase dari Tabel 4.33 adalah sebanyak 81,3%. Nilai perhitungan tersebut berarti aplikasi yang telah dibangun masuk ke kriteria interpretasi sangat membantu untuk menampilkan informasi yang dibutuhkan.

5. Aplikasi ini membantu pendistribusian dokumen karya penulis ?

Tabel 4.34 Persentase Kuesioner Pertanyaan 5

Pertanyaan	No	Keterangan	Responden	Skor	Persentase(%)
5	1.	Sangat membantu	8	40	(67/75)*100=89,3
	2.	Membantu	6	24	
	3.	Cukup membantu	1	3	
	4.	Tidak membantu	0	0	
	5.	Sangat tidak membantu	0	0	
Jumlah			15	67	

Hasil presentase dari Tabel 4.34 adalah sebanyak 89,3%. Nilai perhitungan tersebut berarti aplikasi yang telah dibangun masuk ke kriteria interpretasi sangat membantu pendistribusian dokumen karya penulis.

Berdasarkan pengujian *beta* diatas dapat diambil kesimpulan bahwa:

1. Aplikasi ini sangat mudah dipelajari dan digunakan oleh para penulis.
2. Tampilan dari aplikasi ini cukup menarik.
3. Aplikasi ini cukup interaktif.

4. Aplikasi ini sangat membantu penulis dalam menampilkan informasi yang dibutuhkan.
5. Aplikasi ini sangat membantu dalam pendistribusian dokumen ke pihak Balai Bahasa Bandung.