BAB IV

IMPLEMENTASI DAN PENGUJIAN SISTEM

IV.1 Implementasi Sistem

Tahap implementasi sistem adalah tahap menterjemahkan perancangan sistem berdasarkan hasil analisis dalam bahasa pemrograman serta penerapan perangkat lunak pada keadaan yang sesungguhnya. Wujud dari hasil tahap implementasi ini adalah sebuah sistem yang siap untuk diuji dan digunakan.

IV.1.1 Kebutuhan Perangkat Keras (*Hardware*)

Spesifikasi perangkat keras (*hardware*) yang digunakan dalam pembangunan perangkat lunak dalam penelitian ini adalah sebagai berikut :

Tabel IV-1 Kebutuhan Perangkat Keras (*Hardware*)

No	Perangkat Keras	Keterangan
1	*Processor*	Dual core 3,2 GHz
2	RAM	2 Gb
3	*Harddisk*	160 Gb
4	Monitor	Resolusi 1366 x 768 px
5	*Keyboard* dan *Mouse*	*Keyboard* dan *Mouse*
6	*Printer*	*Printer*
7	Modem	Modem

IV.1.2 Kebutuhan Perangkat Lunak (*Software*)

Spesifikasi perangkat lunak (*Software*) yang digunakan untuk mengembangkan perangkat lunak yang dibangun adalah sebagai berikut :

Tabel IV-2 Kebutuhan Perangkat Lunak (*Software*)

No	Perangkat Lunak	Keterangan
1	Sistem Operasi	*Windows 7*
2	*Web Browser*	*Mozilla Firefox, Chrome*

No	Perangkat Lunak	Keterangan
3	PHP	Bahasa Pemrograman
4	MySQL	DBMS
5	*Adobe Dreamweaver CS3*	Editor PHP
6	*Wamp Server*	*Web Server*

IV.1.3 Implementasi Data

Pembuatan *database* dilakukan dengan menggunakan aplikasi DBMS MySQL. Berikut ini adalah tabel-tabel yang terdapat dalam perangkat lunak yang dibangun.

No	Nama Tabel	Nama File
1	pengguna	pengguna.sql
2	sumber_berita	sumber_berita.sql
3	konten_berita	konten_berita.sql
4	hasil_pembersihan	hasil_pembersihan.sql
5	hasil_pencarian	hasil_pencarian.sql

Implementasi data untuk masing-masing tabel dalam bahasa SQL adalah sebagai berikut :

1. Pengguna

   Implementasi *query* yang digunakan untuk membuat tabel pengguna adalah sebagai berikut :

Tabel IV-3 Implementasi *query* tabel pengguna

```
1  CREATE TABLE `pengguna` (
2  `id_pengguna` int(11) NOT NULL AUTO_INCREMENT,
3  `session_pengguna` varchar(100) NOT NULL,
4  PRIMARY KEY (`id_pengguna`)
5  ) ENGINE=InnoDB  DEFAULT CHARSET=latin1
```

2. Sumber_berita

   Implementasi *query* yang digunakan untuk membuat tabel sumber_berita adalah sebagai berikut :

Tabel IV-4 Implementasi *query* tabel sumber_berita

```
1  CREATE TABLE `sumber_berita` (
2  `id_sumber_berita` int(11) NOT NULL AUTO_INCREMENT,
3  `nama_sumber_berita` varchar(10) NOT NULL,
4  PRIMARY KEY (`id_sumber_berita`)
5  ) ENGINE=InnoDB  DEFAULT CHARSET=latin1
```

3. Konten_berita

   Implementasi *query* yang digunakan untuk membuat tabel konten_berita adalah sebagai berikut :

Tabel IV-5 Implementasi *query* tabel konten_berita

```
1  CREATE TABLE `konten_berita` (
2  `id_konten_berita` int(11) NOT NULL AUTO_INCREMENT,
3  `id_sumber_berita` int(11) NOT NULL,
4  `judul_konten` varchar(300) NOT NULL,
5  `isi_konten` varchar(500) NOT NULL,
6  `link` varchar(500) NOT NULL,
7  PRIMARY KEY (`id_konten_berita`),
8  KEY `FK_konten_berita` (`id_sumber_berita`),
9  CONSTRAINT `FK_konten_berita` FOREIGN KEY (`id_sumber_berita`
   ) REFERENCES `sumber_berita` (`id_sumber_berita`) ON DELETE C
   ASCADE ON UPDATE CASCADE
10 ) ENGINE=InnoDB  DEFAULT CHARSET=latin1
```

4. Hasil_pembersihan

   Implementasi *query* yang digunakan untuk membuat tabel hasil_pembersihan adalah sebagai berikut :

Tabel IV-6 Implementasi *query* tabel hasil_pembersihan

```
1  CREATE TABLE `hasil_pembersihan` (
2  `id_hasil_pembersihan` int(11) NOT NULL AUTO_INCREMENT,
3  `id_konten_berita` int(11) NOT NULL,
4  `kata_yang_didapat` varchar(100) NOT NULL,
```

```
5   `jml_kemunculan` int(11) NOT NULL,
6   PRIMARY KEY (`id_hasil_pembersihan`),
7   KEY `FK_hasil_pembersihan` (`id_konten_berita`),
8   CONSTRAINT `FK_hasil_pembersihan` FOREIGN KEY (`id_konten_ber
    ita`) REFERENCES `konten_berita` (`id_konten_berita`) ON DELE
    TE CASCADE ON UPDATE CASCADE
9   ) ENGINE=InnoDB  DEFAULT CHARSET=latin1
```

5. Hasil_pencarian

Implementasi *query* yang digunakan untuk membuat tabel hasil_ pencarian adalah sebagai berikut :

Tabel IV-7 Implementasi *query* tabel hasil_pencarian

```
1   CREATE TABLE `hasil_pencarian` (
2   `id_hasil_pencarian` int(11) NOT NULL AUTO_INCREMENT,
3   `id_pengguna` int(11) NOT NULL,
4   `id_konten_berita` int(11) NOT NULL,
5   PRIMARY KEY (`id_hasil_pencarian`),
6   KEY `FK_hasil_pencarian_konten_berita` (`id_konten_berita`),
7   KEY `FK_hasil_pencarian_pengguna` (`id_pengguna`),
8   CONSTRAINT `FK_hasil_pencarian_konten_berita` FOREIGN KEY (`i
    d_konten_berita`) REFERENCES `konten_berita` (`id_konten_beri
    ta`) ON DELETE CASCADE ON UPDATE CASCADE,
9   CONSTRAINT `FK_hasil_pencarian_pengguna` FOREIGN KEY (`id_pen
    gguna`) REFERENCES `pengguna` (`id_pengguna`) ON DELETE CASCA
    DE ON UPDATE CASCADE
10  ) ENGINE=InnoDB  DEFAULT CHARSET=latin1
```

IV.1.4 Implementasi Antarmuka

Implementasi antarmuka dilakukan terhadap setiap tampilan perangkat lunak yang dibangun dan pengkodeannya dalam bentuk *file* program. Berikut ini adalah implementasi antarmuka yang digunakan dalam perangkat lunak ini.

Tabel IV-8 Implementasi Antarmuka

No.	Menu	Proses ke-	Nama File
1	Halaman utama	Proses 1	index.php css.css
2	Cari di Google	Proses 1.1, proses 3	searchGoogle4-v4.php css.css libFunc.php

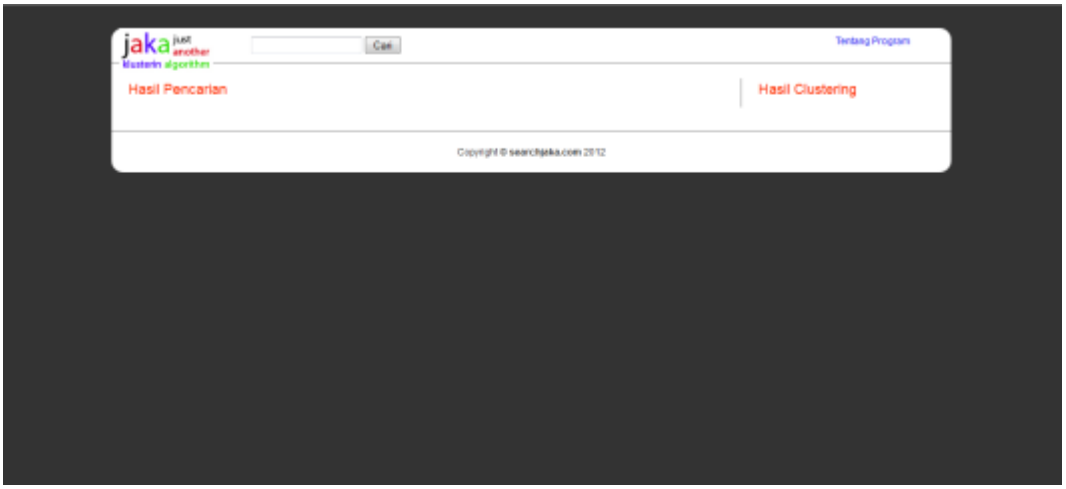
No.	Menu	Proses ke-	Nama File
			PorterStemmer.php ambil_data_clustering.php
3	Cari di Twitter	Proses 1.2, proses 3	searchTwitter4-v4.php css.css libFunc.php PorterStemmer.php ambil_data_clustering.php
4	Tentang program	Proses 4	tentangProgram.php css.css

Berikut ini adalah tampilan antarmuka halaman utama pada perangkat lunak yang dibangun. Pada antarmuka ini pengguna dapat memilih untuk melakukan pencarian di dalam Google atau Twitter.



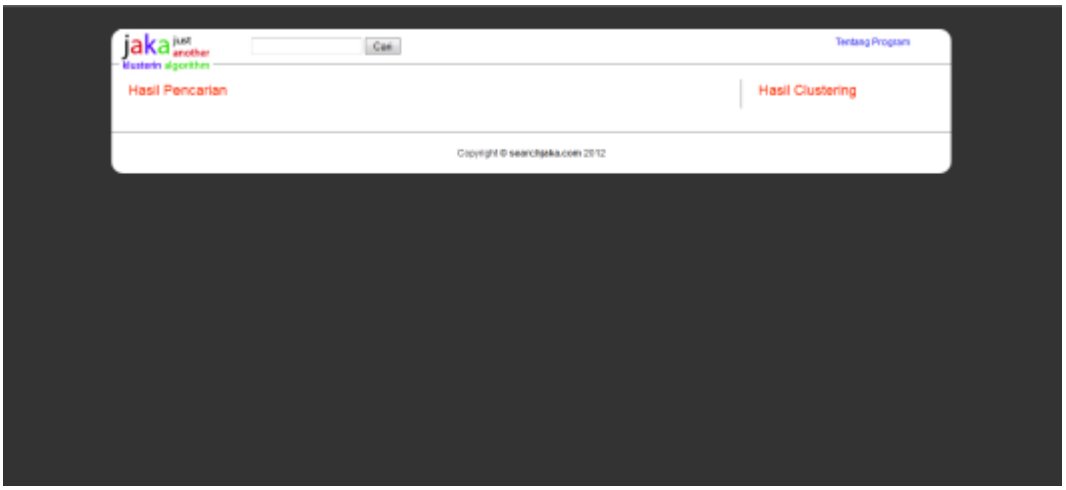
Gambar IV-1 Tampilan antarmuka halaman utama

Tampilan antarmuka di bawah ini adalah tampilan antarmuka untuk mencari dan menampilkan hasil pencarian dan hasil *clustering* dalam Google.



Gambar IV-2 Tampilan antarmuka pencarian Google

Tampilan antarmuka di bawah ini adalah tampilan antarmuka untuk mencari dan menampilkan hasil pencarian dan hasil *clustering* dalam Twitter.



Gambar IV-3 Tampilan antarmuka pencarian Twitter

Tampilan antarmuka di bawah ini adalah tampilan antarmuka tentang program.



Gambar IV-4 Tampilan antarmuka tentang program

IV.2 Pengujian Sistem

Tahap selanjutnya setelah tahapan implementasi adalah tahap pengujian sistem. Tahap ini merupakan tahapan yang bertujuan untuk menemukan kesalahan-kesalahan atau kekurangan-kekurangan pada aplikasi yang dibangun. Pengujian ini dilakukan untuk mengetahui apakah perangkat lunak yang dibuat telah memenuhi kriteria yang sesuai dengan tujuan perancangan perangkat lunak atau belum.

IV.2.1 Rencana Pengujian

Rencana pengujian akan dilakukan dengan menguji sistem secara keseluruhan. Berikut ini adalah rencana pengujian yang akan dilakukan terhadap perangkat lunak yang dibangun.

Tabel IV-9 Rencana pengujian

Kelas Uji	Butir Uji	Jenis Pengujian
Cari di Google	Pencarian	*Black Box*
Cari di Twitter	Pencarian	*Black Box*
Penghapusan *stop word*	-	*White Box*
Stemming kata	-	*White Box*
Identifikasi *cluster*	-	*White Box*

IV.2.2 Pengujian *White Box*

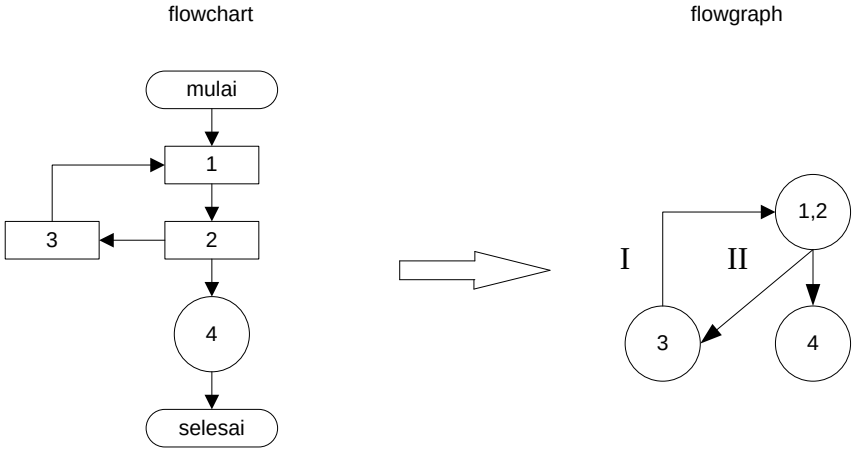
Pengujian *white box* bertujuan untuk mengetahui kinerja logika yang dibuat pada sebuah perangkat lunak. Pengujian *white box* akan digunakan pada langkah pembersihan *stop word* dan *stemming* kata, untuk mengukur kinerja logika berdasarkan *pseudocode* yang telah dibuat pada tahap analisis.

Pengujian *white box* dilakukan dengan beberapa tahap, yaitu :

1. Ubah pseudocode menjadi flowchart,
2. Ubah *flowchart* menjadi *flowgraph*,
3. Tahap pengujian, dilakukan dengan 5 cara yaitu, menghitung *region*, menghitung *Cyclomatic Complexity*, menghitung *independent path*, menggunakan *graph matriks*, menghitung *predicate node*.

IV.2.2.1 Pengujian Penghapusan *Stop Word*

Pengujian ini dilakukan untuk mengetahui apakah prosedural penghapusan *stop word* telah berjalan dengan efektif atau tidak. Berikut ini adalah langkah-langkah pengujian penghapusan *stop word* :



Gambar IV-5 *Flowchart – flowgraph* penghapusan *stop word*

1. *Region* = 2
2. $V_{(G)}$ = edge – node + 2

   = 3 – 3 + 2

   = 2
3. *Independent path*

   1 = 1, 2, 4

   2 = 1, 2, 3, 1, 2, 4
4. *Graph* matriks

Tabel IV-10 *Graph* matriks penghapusan stop word

Node	1	2	3	4	Sum
1		1			0
2			1	1	1
3	1				0
4					0
Total					1

$V_{(G)}$ = jumlah *graph* matriks + 1

= 1 + 1

= 2

5. *Predicate node*

   $V_{(G)}$ = jumlah node yang memiliki lebih dari 1 jalur + 1

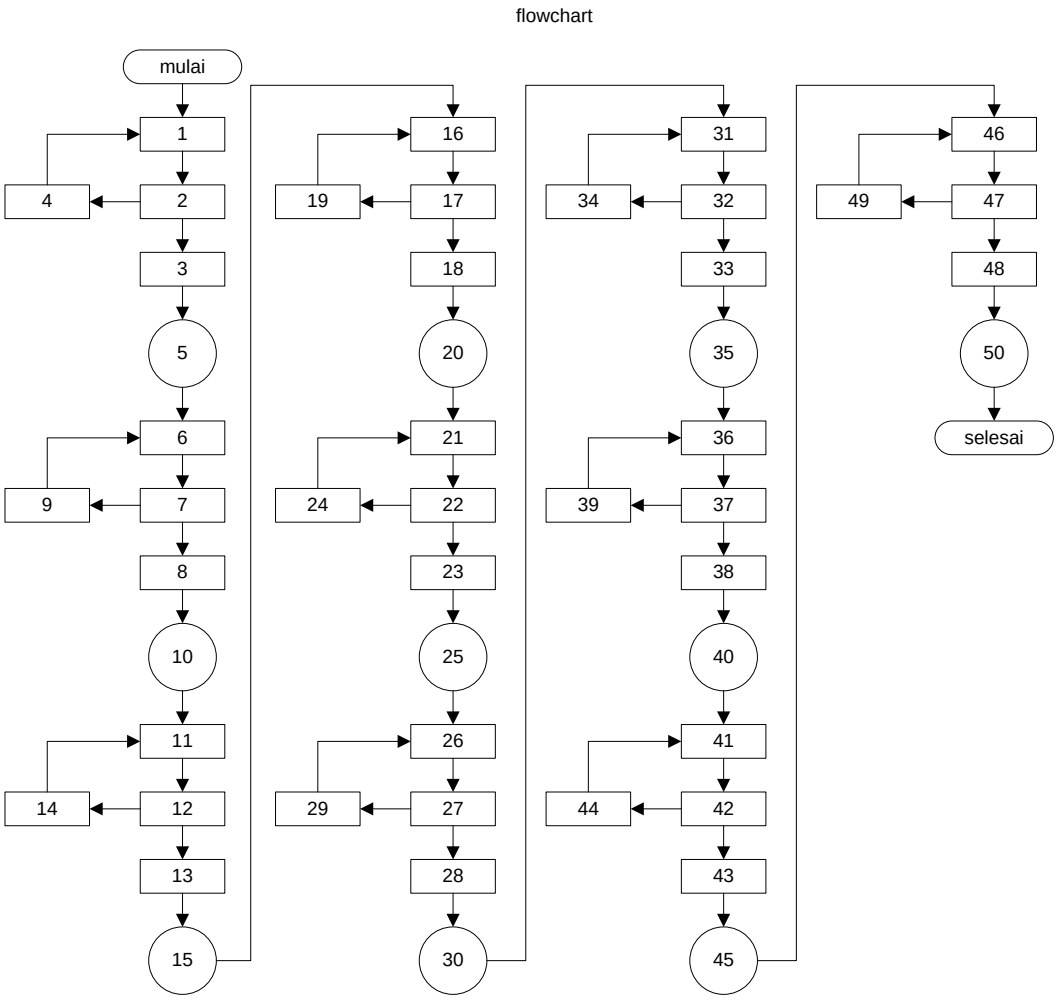
   = 1 + 1

   = 2

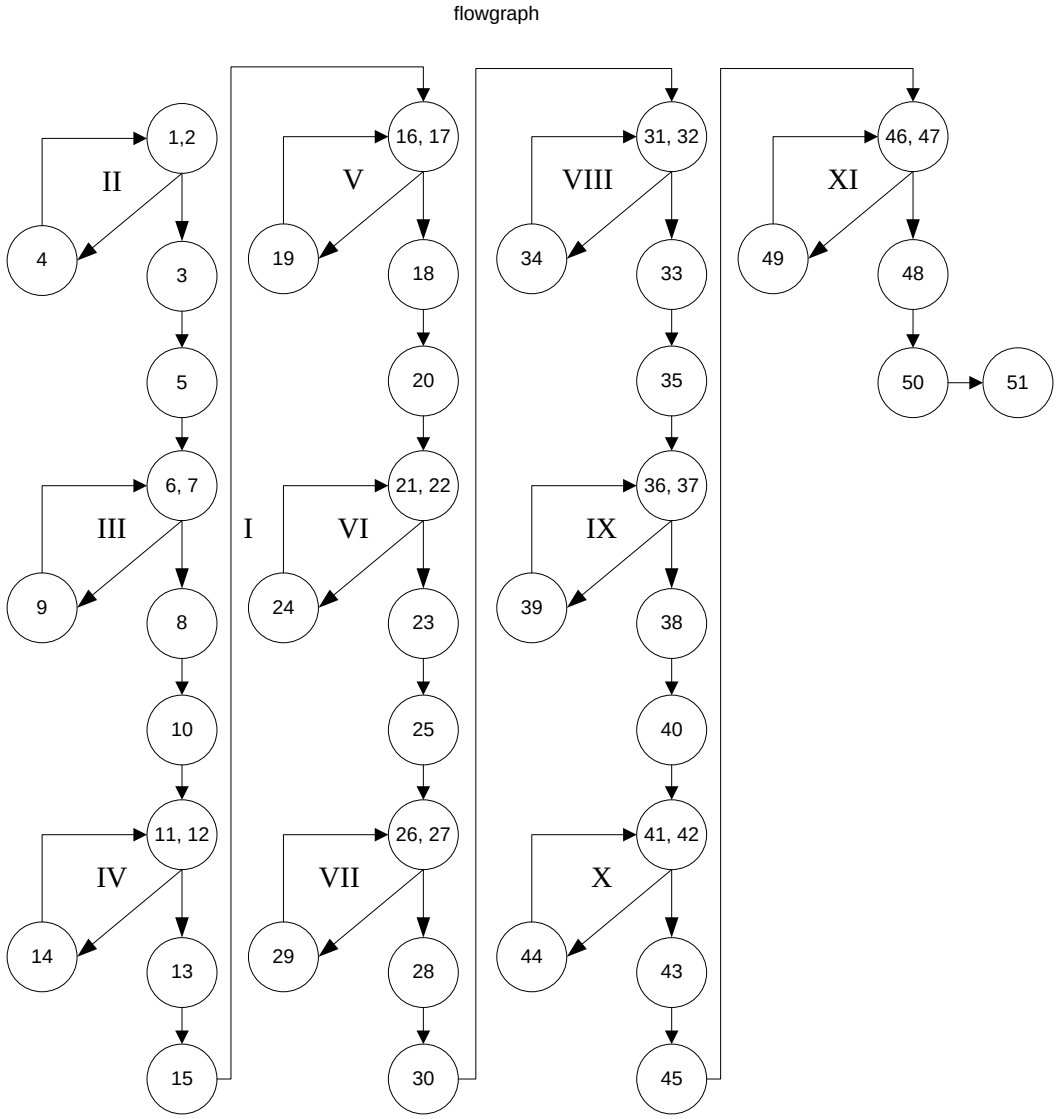
Berdasarkan pengujian, setiap metode menghasilkan nilai yang sama, yaitu 2, maka dapat disimpulkan bahwa pengujian *white box* untuk penghapusan *stop word* berjalan dengan baik.

IV.2.2.2 Pengujian *Stemming* Kata

Pengujian ini dilakukan untuk mengetahui apakah prosedural penghapusan *stemming* telah berjalan dengan efektif atau tidak. Berikut ini adalah langkah-langkah pengujian penghapusan *stemming* :



Gambar IV-6 *Flowchart stemming* kata



Gambar IV-7 *Flowgraph stemming* kata

1. *Region* = 11
2. $V_{(G)}$ = edge – node + 2

   = 50 – 41 + 2

   = 11

3. *Independent path*

1 = 1, 2, 3, 5, 6, 7, 8, 10, 11, 12, 13, 15, 16, 17, 18, 20, 21, 22, 23, 25, 26, 27, 28, 30, 31, 32, 33, 35, 36, 37, 38, 40, 41, 42, 43, 45, 46, 47, 48, 50, 51

2 = 1, 2, 4, 1, 2, 3, 5, 6, 7, 8, 10, 11, 12, 13, 15, 16, 17, 18, 20, 21, 22, 23, 25, 26, 27, 28, 30, 31, 32, 33, 35, 36, 37, 38, 40, 41, 42, 43, 45, 46, 47, 48, 50, 51

3 = 1, 2, 3, 5, 6, 7, 9, 6, 7, 8, 10, 11, 12, 13, 15, 16, 17, 18, 20, 21, 22, 23, 25, 26, 27, 28, 30, 31, 32, 33, 35, 36, 37, 38, 40, 41, 42, 43, 45, 46, 47, 48, 50, 51

4 = 1, 2, 3, 5, 6, 7, 8, 10, 11, 12, 14, 11, 12, 13, 15, 16, 17, 18, 20, 21, 22, 23, 25, 26, 27, 28, 30, 31, 32, 33, 35, 36, 37, 38, 40, 41, 42, 43, 45, 46, 47, 48, 50, 51

5 = 1, 2, 3, 5, 6, 7, 8, 10, 11, 12, 13, 15, 16, 17, 19, 16, 17, 18, 20, 21, 22, 23, 25, 26, 27, 28, 30, 31, 32, 33, 35, 36, 37, 38, 40, 41, 42, 43, 45, 46, 47, 48, 50, 51

6 = 1, 2, 3, 5, 6, 7, 8, 10, 11, 12, 13, 15, 16, 17, 18, 20, 21, 22, 24, 21, 22, 23, 25, 26, 27, 28, 30, 31, 32, 33, 35, 36, 37, 38, 40, 41, 42, 43, 45, 46, 47, 48, 50, 51

7 = 1, 2, 3, 5, 6, 7, 8, 10, 11, 12, 13, 15, 16, 17, 18, 20, 21, 22, 23, 25, 26, 27, 29, 26, 27, 28, 30, 31, 32, 33, 35, 36, 37, 38, 40, 41, 42, 43, 45, 46, 47, 48, 50, 51

8 = 1, 2, 3, 5, 6, 7, 8, 10, 11, 12, 13, 15, 16, 17, 18, 20, 21, 22, 23, 25, 26, 27, 28, 30, 31, 32, 34, 21, 32, 33, 35, 36, 37, 38, 40, 41, 42, 43, 45, 46, 47, 48, 50, 51

9 = 1, 2, 3, 5, 6, 7, 8, 10, 11, 12, 13, 15, 16, 17, 18, 20, 21, 22, 23, 25, 26, 27, 28, 30, 31, 32, 33, 35, 36, 37, 39, 36, 37, 38, 40, 41, 42, 43, 45, 46, 47, 48, 50, 51

10 = 1, 2, 3, 5, 6, 7, 8, 10, 11, 12, 13, 15, 16, 17, 18, 20, 21, 22, 23, 25, 26, 27, 28, 30, 31, 32, 33, 35, 36, 37, 38, 40, 41, 42, 44, 41, 42, 43, 45, 46, 47, 48, 50, 51

11 = 1, 2, 3, 5, 6, 7, 8, 10, 11, 12, 13, 15, 16, 17, 18, 20, 21, 22, 23, 25, 26, 27, 28, 30, 31, 32, 33, 35, 36, 37, 38, 40, 41, 42, 43, 45, 42, 43, 45, 46, 47, 49, 46, 47, 48, 50, 51

4. *Graph* matriks

Tabel IV-11 *Graph* matriks *stemming* kata

Node	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	Sum
1		1																																			0
2			1	1																																	1
3					1																																0
4	1																																				0
5						1																															0
6							1																														0
7								1	1																												1
8										1																											0
9						1																															0
10											1																										0
11												1																									0
12													1	1																							1
13															1																						0
14											1																										0
15																1																					0
16																	1																				0
17																		1	1																		1
18																			1																		0
19																1																					0
20																					1																0
21																						1															0
22																							1	1													1
23																									1												0
24																					1																0
25																										1											0
26																											1										0
27																												1	1								1
28																													1								0
29																										1											0
30																														1							0
31																															1						0
32																																1	1				1
33																																	1				0
34																														1							0
35																																			1		0
36																																				1	0

Node	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	Sum
37																																					1	1															1
38																																						1															0
39																																			1																	0
40																																									1										0
41																																										1									0
42																																											1	1							1
43																																												1							0
44																																										1									0
45																																													1						0
46																																														1					0
47																																															1	1			1
48																																																1		0
49																																													1					0
50																																																	1	0
51																																																		0
Total																																																		10

$V_{(G)}$ = jumlah *graph* matriks + 1

= 10 + 1

= 11

5. *Predicate node*

$V_{(G)}$ = jumlah node yang memiliki lebih dari 1 jalur + 1

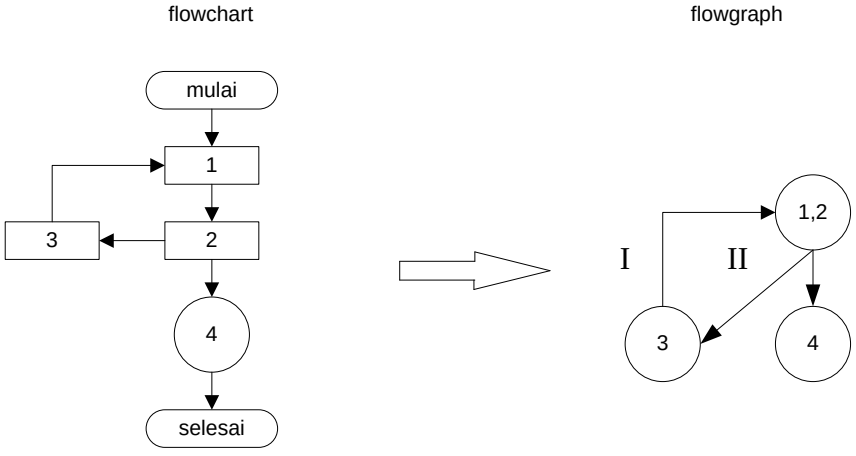
= 10 + 1

= 11

Berdasarkan pengujian, setiap metode menghasilkan nilai yang sama, yaitu 11, maka dapat disimpulkan bahwa pengujian *white box* untuk *stemming* kata berjalan dengan baik.

IV.2.2.3 Pengujian Identifikasi Cluster

Pengujian ini dilakukan untuk mengetahui apakah prosedural identifikasi *cluster* telah berjalan dengan efektif atau tidak. Berikut ini adalah langkah-langkah pengujian identifikasi *cluster* :



Gambar IV-8 *Flowchart - flowgraph* identifikasi *cluster*

1. *Region* = 2
2. $V_{(G)}$ = edge – node + 2

   = 3 – 3 + 2

   = 2
3. *Independent path*

   1 = 1, 2, 4

   2 = 1, 2, 3, 1, 2, 4
6. *Graph* matriks

Tabel IV-12 *Graph* matriks identifikasi cluster

Node	1	2	3	4	Sum
1		1			0
2			1	1	1
3	1				0
4					0
Total					1

$V_{(G)}$ = jumlah *graph* matriks + 1

= 1 + 1

= 2

7. *Predicate node*

$V_{(G)}$ = jumlah node yang memiliki lebih dari 1 jalur + 1

= 1 + 1

= 2

Berdasarkan pengujian, setiap metode menghasilkan nilai yang sama, yaitu 2, maka dapat disimpulkan bahwa pengujian *white box* untuk identifikasi *cluster* berjalan dengan baik.

IV.2.3 Pengujian *Black Box*

Berikut ini adalah kasus dan hasil pengujian *black box* dalam form pencarian Google pada perangkat lunak yang dibangun.

Tabel IV-13 Pengujian *black box form* pencarian Google

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Form pencarian diisi dengan : chelsea	Data hasil pencarian berdasarkan kata kunci "chelsea" muncul	Muncul hasil pencarian berdasarkan kata kunci "chelsea".	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Form pencarian diisi dengan : 12	Data hasil pencarian berdasarkan kata kunci "12" muncul	Muncul hasil pencarian berdasarkan kata kunci "12".	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Form pencarian diisi dengan : ?	Data hasil pencarian berdasarkan kata kunci "?" muncul	Muncul hasil pencarian berdasarkan kata kunci "?".	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Kosong)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Form pencarian kosong	Muncul pesan kesalahan "Form pencarian belum terisi. Silahkan isi dahulu sebelum melakukan pencarian."	Dapat menampilkan pesan kesalahan "Form pencarian belum terisi. Silahkan isi dahulu sebelum melakukan pencarian."	[✓] Diterima [ ] Ditolak

Adapun kasus dan hasil pengujian *black box* dalam form pencarian Twitter pada perangkat lunak yang dibangun adalah sebagai berikut.

Tabel IV-14 Pengujian *black box form* pencarian Twitter

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Form pencarian diisi dengan : chelsea	Data hasil pencarian berdasarkan kata kunci "chelsea" muncul	Muncul hasil pencarian berdasarkan kata kunci "chelsea".	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Form pencarian diisi dengan : 12	Data hasil pencarian berdasarkan kata kunci "12" muncul	Muncul hasil pencarian berdasarkan kata kunci "12".	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Form pencarian diisi dengan : ?	Data hasil pencarian berdasarkan kata kunci "?" muncul	Muncul hasil pencarian berdasarkan kata kunci "?".	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Kosong)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Form pencarian kosong	Muncul pesan kesalahan "Form pencarian belum terisi. Silahkan isi dahulu sebelum melakukan pencarian."	Dapat menampilkan pesan kesalahan "Form pencarian belum terisi. Silahkan isi dahulu sebelum melakukan pencarian."	[✓] Diterima [ ] Ditolak

IV.2.4 Pengujian *Beta*

Pengujian *beta* merupakan pengujian yang dilakukan secara objektif dan dilakukan dengan tujuan untuk mengetahui sejauh mana kualitas dari perangkat lunak yang dibangun. Untuk itu, pengujian *beta* dilakukan dengan cara membuat kuesioner mengenai kepuasan pengguna dengan kandungan poin syarat *user friendly*. Kuesioner disebarkan menggunakan teknik *Simple Random Sampling*

dimana anggota *sample* dari populasi dilakukan secara acak tanpa memperhatikan strata yang ada dalam populasi. Kuesioner disebarkan kepada 30 reponden yang memiliki tingkat pendidikan minimal SMA. Dari hasil kuesioner tersebut akan dilakukan perhitungan agar diambil kesimpulan terhadap penilaian penerapan sistem yang dibangun. Kuesioner ini terdiri dari 6 pertanyaan dan menggunakan skala likert dengan skala 1 sampai 5, dengan struktur pertanyaan dan ketentuan skala sebagai berikut :

1. Apakah Anda tidak memiliki kesulitan dalam menggunakan *website* searchjaka.com?

   A. Sangat Setuju
   B. Setuju
   C. Cukup Setuju
   D. Kurang Setuju
   E. Tidak Setuju

2. Apakah Anda setuju website searchjaka.com memiliki navigasi yang mudah?

   A. Sangat Setuju
   B. Setuju
   C. Cukup Setuju
   D. Kurang Setuju
   E. Tidak Setuju

3. Apakah Anda setuju bahwa *website* searchjaka.com ini menampilkan informasi hasil pencarian yang sama seperti Anda melakukan pencarian di Google atau Twitter?

   A. Sangat Setuju
   B. Setuju
   C. Cukup Setuju
   D. Kurang Setuju
   E. Tidak Setuju

4. Apakah Anda setuju dengan adanya *website* searchjaka.com dapat membantu Anda untuk menemukan hasil pencarian dengan lebih cepat?

   A. Sangat Setuju
   B. Setuju
   C. Cukup Setuju
   D. Kurang Setuju
   E. Tidak Setuju

5. Apakah Anda setuju bahwa *website* searchjaka.com memiliki poin-poin pada bagian hasil *clustering* yang sudah cukup mewakili hasil pencarian secara keseluruhan?

   A. Sangat Setuju
   B. Setuju
   C. Cukup Setuju
   D. Kurang Setuju
   E. Tidak Setuju

Berdasarkan data hasil kuesioner tersebut, dapat dicari presentase masing-masing jawaban dengan menggunakan rumus skala likert sebagai berikut :

$$P = \frac{total\ skor}{skor\ ideal}\ x\ 100\ \% \tag{4.1}$$

Keterangan :

P = Nilai presentase yang dicari
total skor = Jumlah frekuensi dikalikan dengan nilai yang ditetapkan jawaban
skor ideal = Nilai tertinggi dikalikan dengan jumlah sampel
100% = Bilangan tetap

Berikut adalah hasil persentase masing-masing jawaban yang sudah dihitung nilainya dengan menggunakan rumus diatas.

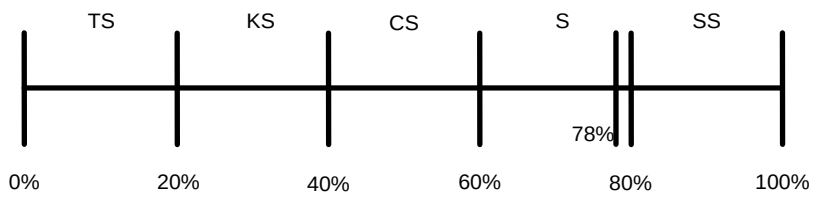
1. Apakah Anda tidak memiliki kesulitan dalam menggunakan *website* searchjaka.com?

   A. Sangat Setuju
   B. Setuju
   C. Cukup Setuju
   D. Kurang Setuju
   E. Tidak Setuju

Pertanyaan	No.	Keterangan	Skor	Frekuensi	Total skor
1	1	Sangat setuju	5	6	30
	2	Setuju	4	15	60
	3	Cukup setuju	3	9	27
	4	Kurang setuju	2	0	0
	5	Tidak setuju	1	0	0
Jumlah				**30**	**117**

$$P = \frac{117}{150}\, x\, 100\ \% = 78\ \%$$

Berdasarkan hasil presentase diatas maka dapat disimpulkan bahwa penilaian terhadap kemudahan penggunaan perangkat lunak 78% dari yang diharapkan 100% atau bisa dikategorikan sebagai setuju bahwa responden tidak memiliki kesulitan dalam menggunakan perangkat lunak. Berikut kriteria interpretasi skor :



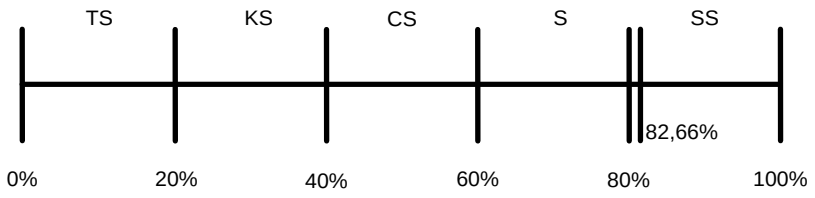
2. Apakah Anda setuju website searchjaka.com memiliki navigasi yang mudah?

A. Sangat Setuju
B. Setuju
C. Cukup Setuju
D. Kurang Setuju
E. Tidak Setuju

Pertanyaan	No.	Keterangan	Skor	Frekuensi	Total skor
2	1	Sangat setuju	5	8	40
	2	Setuju	4	18	72
	3	Cukup setuju	3	4	12
	4	Kurang setuju	2	0	0
	5	Tidak setuju	1	0	0
Jumlah				**30**	**124**

$$P = \frac{124}{150}\ x\ 100\ \% = 82{,}66\ \%$$

Berdasarkan hasil presentase diatas maka dapat disimpulkan bahwa penilaian terhadap navigasi perangkat lunak memperoleh 82,66% dari yang diharapkan 100% atau bisa dikategorikan sebagai sangat setuju bahwa perangkat lunak memiliki navigasi yang mudah dimengerti. Berikut kriteria interpretasi skor :



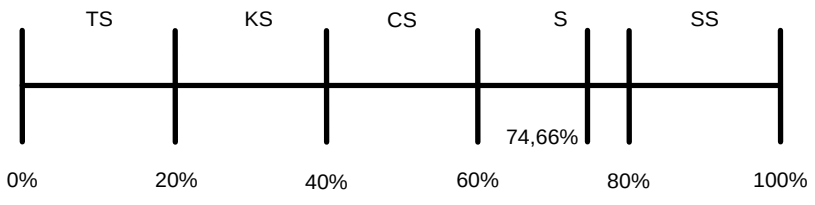
3. Apakah Anda setuju bahwa *website* searchjaka.com ini menampilkan informasi hasil pencarian yang sama seperti Anda melakukan pencarian di Google atau Twitter?

   A. Sangat Setuju
   B. Setuju
   C. Cukup Setuju
   D. Kurang Setuju
   E. Tidak Setuju

Pertanyaan	No.	Keterangan	Skor	Frekuensi	Total skor
3	1	Sangat setuju	5	3	15
	2	Setuju	4	16	64
	3	Cukup setuju	3	11	33
	4	Kurang setuju	2	0	0
	5	Tidak setuju	1	0	0
Jumlah				**30**	**112**

$$P = \frac{112}{150}\ x\ 100\ \% = 74{,}66\ \%$$

Berdasarkan hasil presentase diatas maka dapat disimpulkan bahwa penilaian terhadap informasi yang ditampilkan perangkat lunak memperoleh 74,66% dari yang diharapkan 100% atau bisa dikategorikan sebagai setuju bahwa hasil pencarian informasi dalam perangkat lunak sama dengan hasil pencarian baik dalam Google maupun Twitter. Berikut kriteria interpretasi skor :



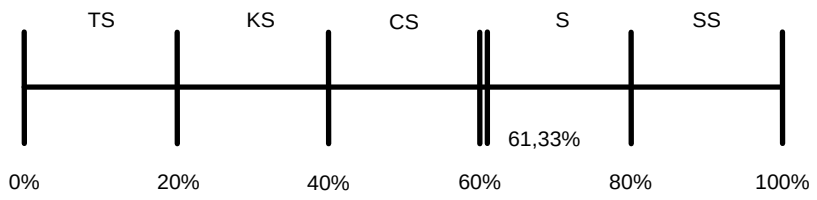
4. Apakah Anda setuju dengan adanya *website* searchjaka.com dapat membantu Anda untuk menemukan hasil pencarian dengan lebih cepat?

   A. Sangat Setuju
   B. Setuju
   C. Cukup Setuju
   D. Kurang Setuju
   E. Tidak Setuju

Pertanyaan	No.	Keterangan	Skor	Frekuensi	Total skor
4	1	Sangat setuju	5	2	10
	2	Setuju	4	8	32
	3	Cukup setuju	3	10	30
	4	Kurang setuju	2	10	20
	5	Tidak setuju	1	0	0
Jumlah				**30**	**92**

$$P = \frac{92}{150}\ x\ 100\ \% = 61{,}33\ \%$$

Berdasarkan hasil presentase diatas maka dapat disimpulkan bahwa penilaian terhadap temuan hasil pencarian perangkat lunak memperoleh 61,33% dari yang diharapkan 100% atau bisa dikategorikan sebagai setuju bahwa pencarian informasi dapat dilakukan dengan lebih cepat dalam perangkat lunak yang dibangun dibandingkan dalam Google maupun Twitter. Berikut kriteria interpretasi skor :



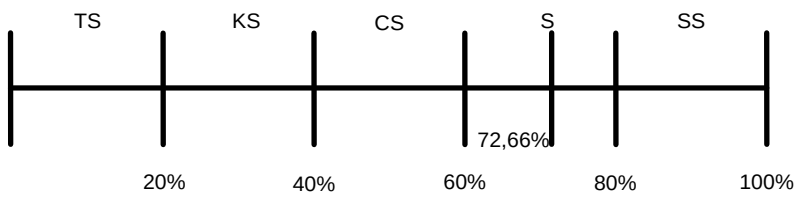
5. Apakah Anda setuju bahwa *website* searchjaka.com memiliki poin-poin pada bagian hasil *clustering* yang sudah cukup mewakili hasil pencarian secara keseluruhan?

A. Sangat Setuju
B. Setuju
C. Cukup Setuju
D. Kurang Setuju
E. Tidak Setuju

Pertanyaan	No.	Keterangan	Skor	Frekuensi	Total skor
5	1	Sangat setuju	5	2	10
	2	Setuju	4	8	32
	3	Cukup setuju	3	17	61
	4	Kurang setuju	2	3	6
	5	Tidak setuju	1	0	0
Jumlah				**30**	**109**

$$P = \frac{109}{150}\ x\ 100\ \% = 72{,}66\ \%$$

Berdasarkan hasil presentase diatas maka dapat disimpulkan bahwa penilaian terhadap poin-poin hasil clustering perangkat lunak memperoleh 72,66% dari yang diharapkan 100% atau bisa dikategorikan sebagai setuju bahwa hasil *clustering* cukup mewakili hasil pencarian secara keseluruhan. Berikut kriteria interpretasi skor :



IV.2.5 Kesimpulan Pengujian *Beta*

Berdasarkan pengujian di atas, dapat disimpulkan bahwa perangkat lunak yang dibangun telah cukup memenuhi tujuan awal pembangunan, yaitu dapat mengelompokan dokumen hasil pencarian *online* dengan cara mengimplementasikan algoritma STC secara langsung.