BAB IV

IMPLEMENTASI DAN PENGUJIAN SISTEM

Pada bab ini akan dilakukan implementasi dan pengujian terhadap sistem. Tahapan ini dilakukan setelah perancangan selesai dilakukan dan selanjutnya akan diimplementasikan pada bahasa pemrograman yang akan digunakan. Setelah implementasi maka dilakukan pengujian sistem yang baru dimana akan dilihat kekurangan-kekurangan pada aplikasi yang baru untuk pengembangan sistem selanjutnya.

4.1 Implementasi

Seteleh menganalisis dan merancang sistem yang akan dibanguan secara rinci, maka selanjutnya menuju tahap implementasi.

Tujuan implementasi adalah untuk mengkonfirmasikan modul program perancangan kepada para pengguna sistem, sehingga pengguna dapat memberikan masukkan terhadap pembangunan sistem.

4.1.1 Kebutuhan Perangkat Keras

Perangkat keras yang diperlukan untuk mengimplementasikan perangkat lunak dari Aplikasi *Information Retrieval System* adalah sebagai berikut :

Tabel 4.1 Perangkat Keras Yang Digunakan

No	Perangkat Keras	Spesifikasi
1	*Processor*	AMD Turion™ X2 Dual-Core Mobile RM-74(2 CPUs) 2.2 GHz.
2	RAM	3 Giga
3	VGA	Onboard
4	HDD	320 Giga
5	Monitor	15 “

4.1.2 Kebutuhan Perangkat Lunak

Perangkat lunak yang digunakan untuk mengimplementasikan aplikasi *Information Retrieval System* ini adalah sebagai berikut :

Tabel 4.2 Perangkat Lunak Yang Digunakan

No	Perangkat Lunak	Spesifikasi
1	Sistem Operasi	Microsoft Windows 7 Professional
2	Bahasa Pemrograman	PHP Ver. 5.3.0
3	*Web Server*	Wamp Ver. 2.0
4	*Database*	MySQL Ver. 5.1.36
5	*Browser*	Mozilla Firefox Ver. 4.1
6	Editor	Adobe Dreamweaver CS5
7	*UML Modeler*	Astah Community Ver. 6.3

4.1.3 Implementasi Basis Data

Pembuatan *Database* dilakukan dengan menggunakan aplikasi *Database Management System* (*DBMS*) yaitu *MySQL* Ver. 5.1.36. Implementasi *database* dalam bahasa SQL adalah sebagai berikut :

Tabel 4.3 Implementasi *Database*

No	Nama Tabel	SQL
1	Admin	CREATE TABLE IF NOT EXISTS `admin` (`username` varchar(20) NOT NULL, `password` varchar(50) NOT NULL, PRIMARY KEY (`username`)) ENGINE=MyISAM DEFAULT CHARSET=latin1;
2	Berita	CREATE TABLE IF NOT EXISTS `berita` (`id` int(11) NOT NULL AUTO_INCREMENT, `judul` varchar(150) NOT NULL, `isi` text NOT NULL, `sumber` varchar(100) NOT NULL, PRIMARY KEY (`id`)) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=22 ;
3	Cache	CREATE TABLE IF NOT EXISTS `cache` (`id` int(11) NOT NULL AUTO_INCREMENT, `query` varchar(100) NOT NULL, `id_dok` int(11) NOT NULL, `value` float NOT NULL, PRIMARY KEY (`id`)) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=4 ;
4	Indexing	CREATE TABLE IF NOT EXISTS `indexing` (`id` int(11) NOT NULL AUTO_INCREMENT, `term` varchar(30) NOT NULL, `id_dok` int(11) NOT NULL, `tf` int(11) NOT NULL, `bobot` float NOT NULL, PRIMARY KEY (`id`)) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=2239 ;
5	Stem	CREATE TABLE IF NOT EXISTS `stem` (`id` int(11) NOT NULL AUTO_INCREMENT, `term` varchar(50) NOT NULL, `stem` varchar(50) NOT NULL,

		PRIMARY KEY (`id`)) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=274 ;
6	Vektor	CREATE TABLE IF NOT EXISTS `vektor` (`id_dok` int(11) NOT NULL, `panjang` float NOT NULL, PRIMARY KEY (`id_dok`)) ENGINE=MyISAM DEFAULT CHARSET=latin1;

4.1.4 Implementasi Modul Program

4.1.4.1 Library

Tabel 4.4 Class Tampilan

No	Fungsi	Keterangan
1	koneksi()	Fungsi untuk mengkoneksikan perintah-perintah sql ke web server
2	navigator()	Fungsi untuk menampilkan menu di halaman administrator
3	microtime_float()	Fungsi untuk menghitung waktu proses eksekusi program atau data.
4	index()	Fungsi untuk memanggil fungsi-fungsi yang ada di class lain agar diproses di halaman index administrator
5	footer()	Fungsi untuk menampilkan footer web

Tabel 4.5 Class Dokumen

No	Fungsi	Keterangan
1	add()	Fungsi untuk menambahkan konten berita
2	view()	Fungsi untuk menampilkan konten berita
3	edit()	Fungsi untuk mengubah konten berita
4	delete()	Fungsi untuk menghapus konten berita

Tabel 4.6 Class Indeks

No	Fungsi	Keterangan
1	hapusTag()	Fungsi untuk menghapus tag html (*markup-free document text*)
2	preprocessing()	Fungsi untuk membersihkan tanda baca, mengubah text dokumen ke bentuk huruf kecil, *stop word removal* dan stemming
3	indexer()	Fungsi untuk menerapkan *preprocessing* dan *tokenization* serta memberikan nilai tf pada kata setiap *term*.
4	weighting()	Fungsi untuk memberikan bobot pada setiap *term*
5	showIndex()	Fungsi untuk menampilkan *term-term* yang sudah di indeks

Tabel 4.7 Class Retrieval

No	Fungsi	Keterangan
1	vectorLength()	Fungsi untuk menghitung panjang vektor dokumen
2	showVL()	Fungsi untuk menampilkan hasil perhitungan panjang vektor dokumen
3	similarity()	Fungsi untuk menghitung kemiripan antara *keyword* dan isi dokumen berita dan menyimpan *keyword* di tabel *cache*
4	getCache()	Menampilkan dokumen berita jika ada *keyword* yang telah di simpan di tabel cache
5	retrieval()	Fungsi untuk menerapkan fungsi *preprocessing()* pada *keyword* dan menampilkan dokumen berita yang diproses oleh fungsi getCache()
6	viewCache()	Fungsi untuk menampilkan *cache*
7	clear()	Fungsi untuk menghapus semua data *cache*

4.1.4.2 *Pseudocode*

a. Markup-free Document Text / Hapus Tag Html

```
$doc     =  mysql_query("SELECT  id,judul,isi  FROM
         berita WHERE isi LIKE '%<%' or isi LIKE
         '%&%'") or die(mysql_error());
$jml     = mysql_num_rows($doc);
if($jml == 0){
    echo "<div class=hurufgagal align=center>Tidak
    ada dokumen yang mengandung tag html</div>";
}else{

    echo  "Tag  html  telah  dihapus  di  ".$jml."
    dokumen";

}//endif

while($data    = mysql_fetch_array($doc)){

    $id  = $data['id'];
    $isi = $data['isi'];
    $del =                preg_replace("/<[^>]+>/s",
"",preg_replace("/&.*?;/s", "", $isi));
    $q   =    mysql_query("UPDATE    berita    SET
isi='$del' WHERE id=$id") or die(mysql_error());

}//endwhile
```

b. Tokenization

```
$berita = explode(" ", trim($isi));
```

c. Filtration / Stop Word Removal

```
$teks = str_replace("'", " ", $teks);
$teks = str_replace("-", " ", $teks);
$teks = str_replace(")", " ", $teks);
$teks = str_replace("(", " ", $teks);
$teks = str_replace("\"", " ", $teks);
$teks = str_replace("/", " ", $teks);
$teks = str_replace("=", " ", $teks);
$teks = str_replace(".", " ", $teks);
$teks = str_replace(",", " ", $teks);
$teks = str_replace(":", " ", $teks);
```

```
$teks = str_replace(";", " ", $teks);
$teks = str_replace("!", " ", $teks);
$teks = str_replace("?", " ", $teks);


$teks = strtolower(trim($teks));

$astopword = array (" yang ", " juga ", " dari ", " dia
", " kami ", " kamu ", "aku", " ini ", " itu "," atau
", " dan ", " tersebut ", " pada ", " dengan ", "
adalah ", " yaitu ", " ke "," tak "," tidak "," di ", "
pada "," jika "," maka "," ada ", " pun "," lain ","
saja "," hanya "," namun "," seperti "," kemudian ","
untuk "," sudah ", " belum ", " memang "," sama "," apa
"," apakah "," oleh ", " sangat "," sehingga "," memang
"," dimana "," sedangkan ", " selain "," tapi "," tanpa
"," tiada "," dengan ");
foreach ($astopword as $i => $value) {
    $teks = str_replace($astopword[$i], " ", $teks);
}
```

d. Stemming

```
$restem = mysql_query("SELECT * FROM stem ORDER BY
id");

while($rowstem = mysql_fetch_array($restem)) {
    $teks=str_replace($rowstem['term'],$rowstem['st
em'], $teks);
}
```

e. Term Weighting

```
$q  =  mysql_query("SELECT  DISTINCT  id_dok  FROM
indexing");
$n  = mysql_num_rows($q);

$rBobot=mysql_query("SELECT id,term,id_dok,tf,bobot
FROM indexing ORDER BY id");
$nRows   = mysql_num_rows($rBobot);
 echo  "Terdapat  ".$nRows."  Term  yang  diberikan
 bobot <br>";


while($dBobot = mysql_fetch_array($rBobot)){
```

```
$term    = $dBobot['term'];
$tf      = $dBobot['tf'];
$id      = $dBobot['id'];

$N       = mysql_query("SELECT COUNT(*) as N FROM
indexing WHERE term='$term'");
$d       = mysql_fetch_array($N);
$Nterm   = $d['N'];

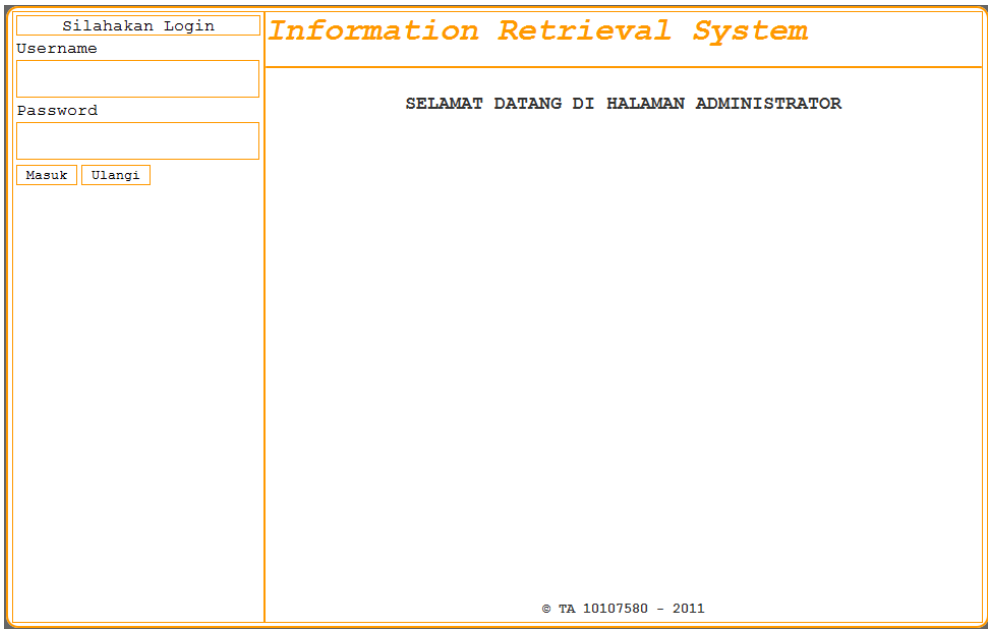
$idf     = log($n/$Nterm);
$bobot   = $tf * $idf;

mysql_query("UPDATE indexing SET bobot=$bobot WHERE
id=$id");

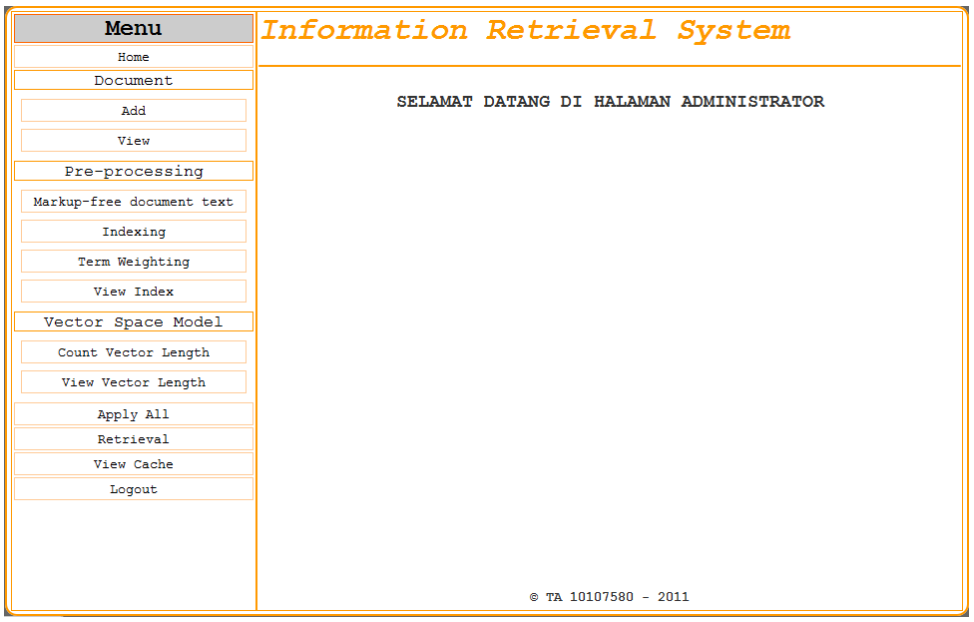
}//endwhile
```

4.1.5 Implementasi Antarmuka

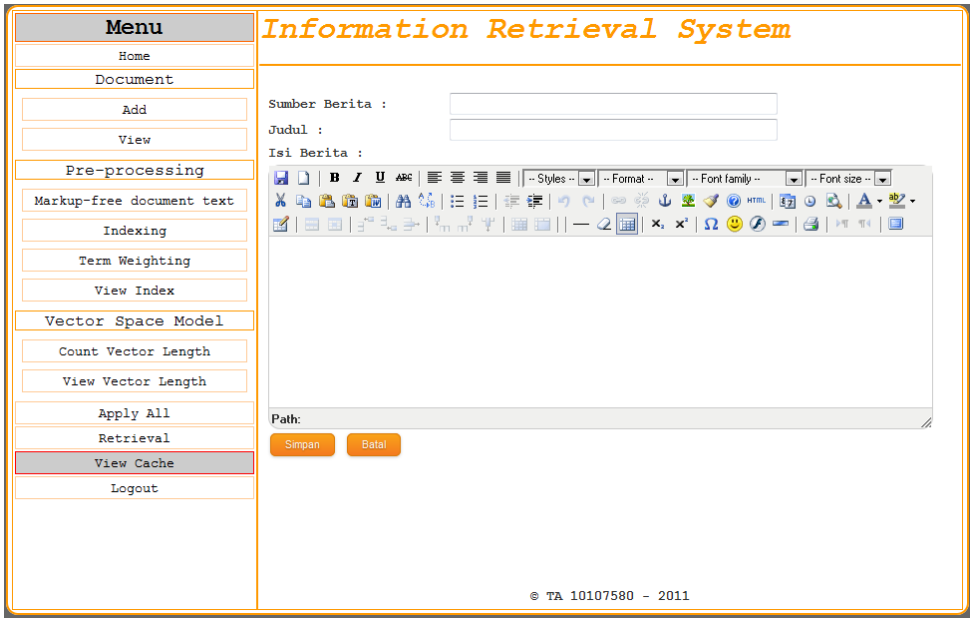
4.1.5.1 Administrator



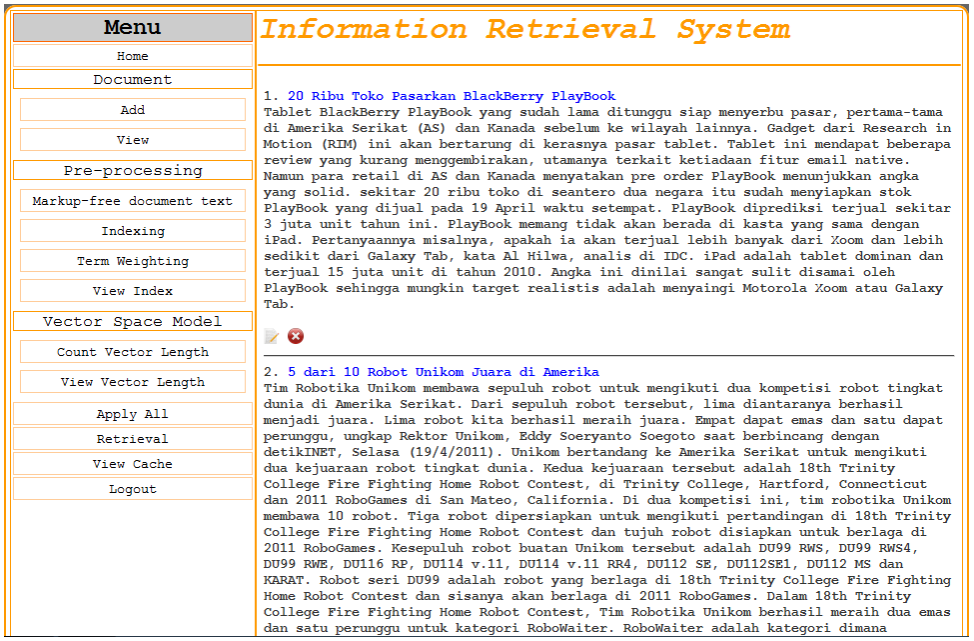
Gambar 4.1 Halaman Login



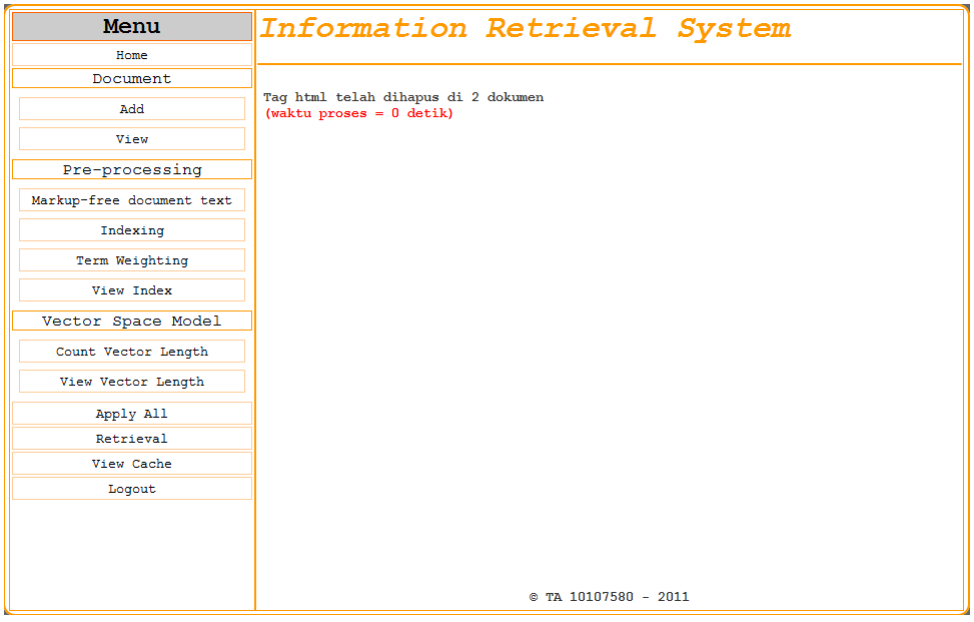
Gambar 4.2 Halaman Utama Administrator



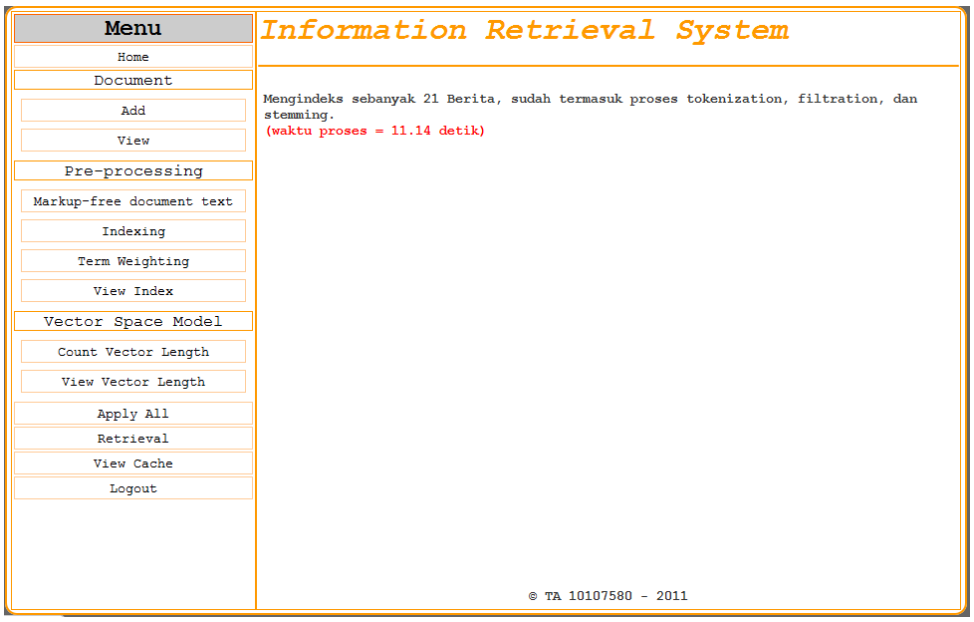
Gambar 4.3 Halaman Tambah Dokumen Berita



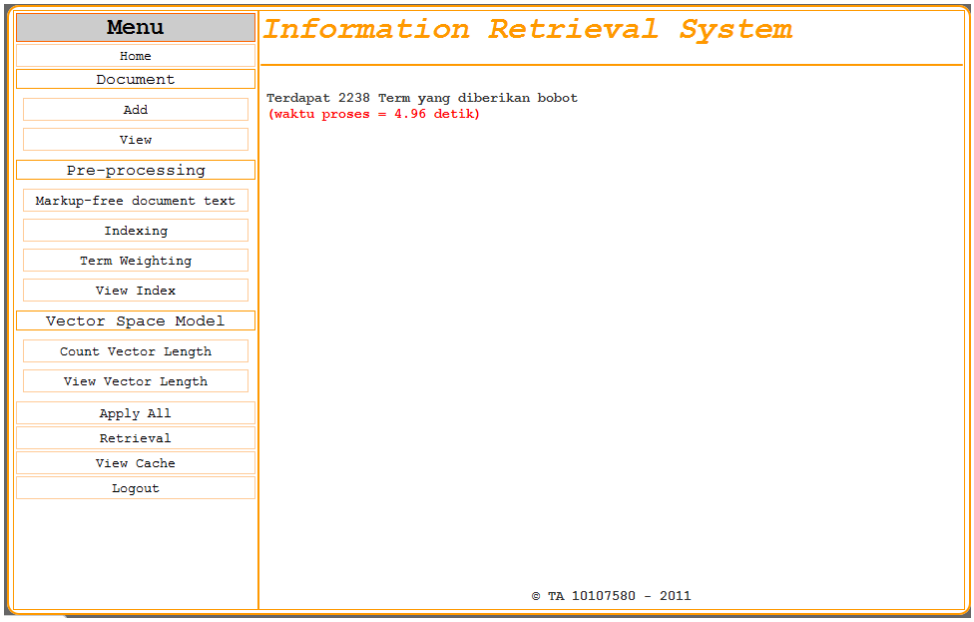
Gambar 4.4 Halaman Tampil Dokumen Berita



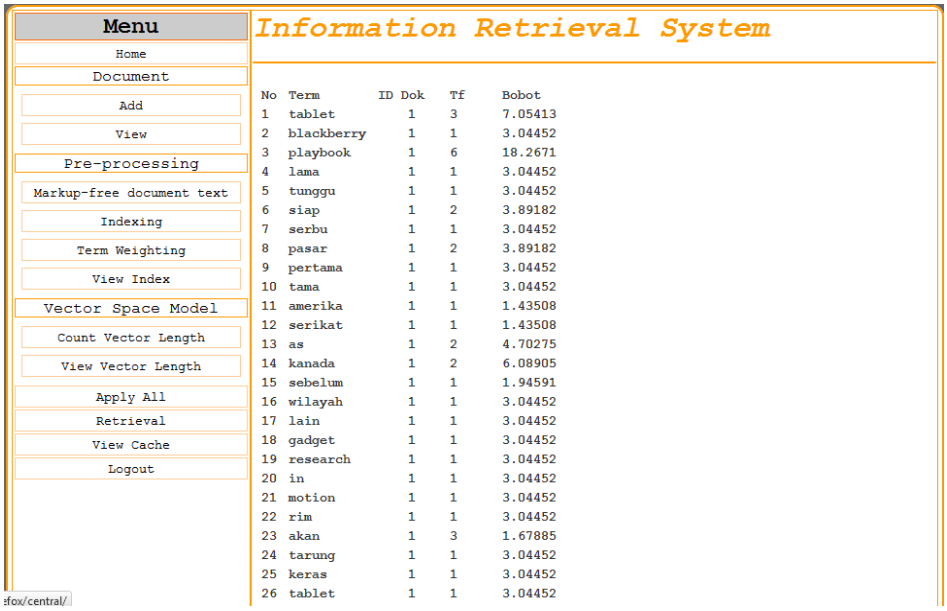
Gambar 4.5 Halaman Hapus Tag HTML



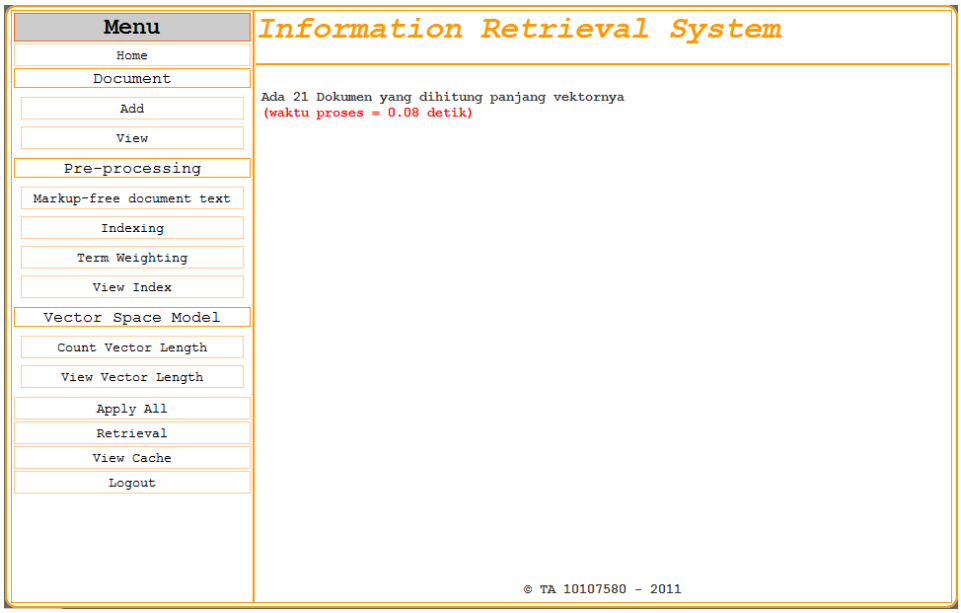
Gambar 4.6 Halaman Indexing



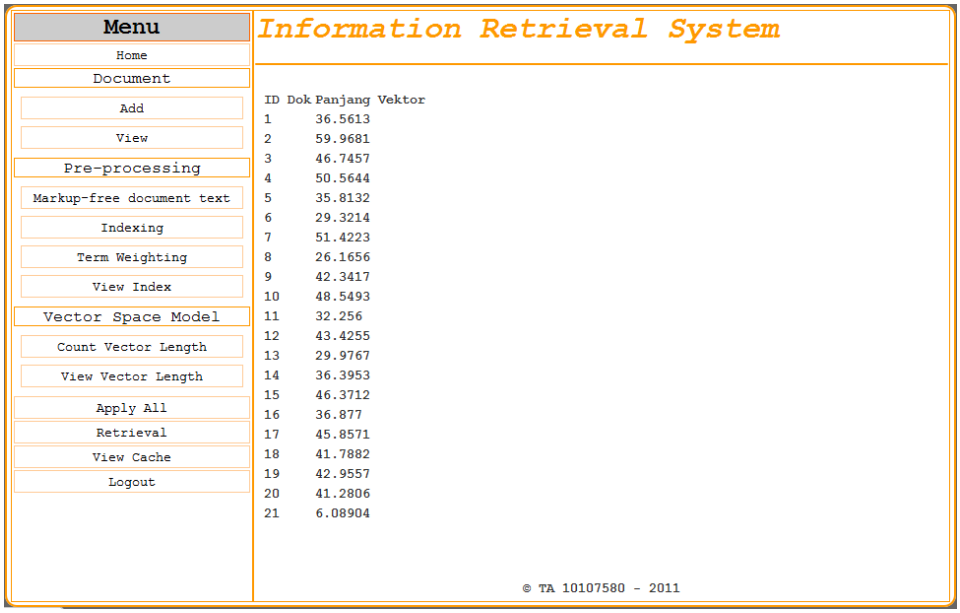
Gambar 4.7 Halaman *Term Weighting*



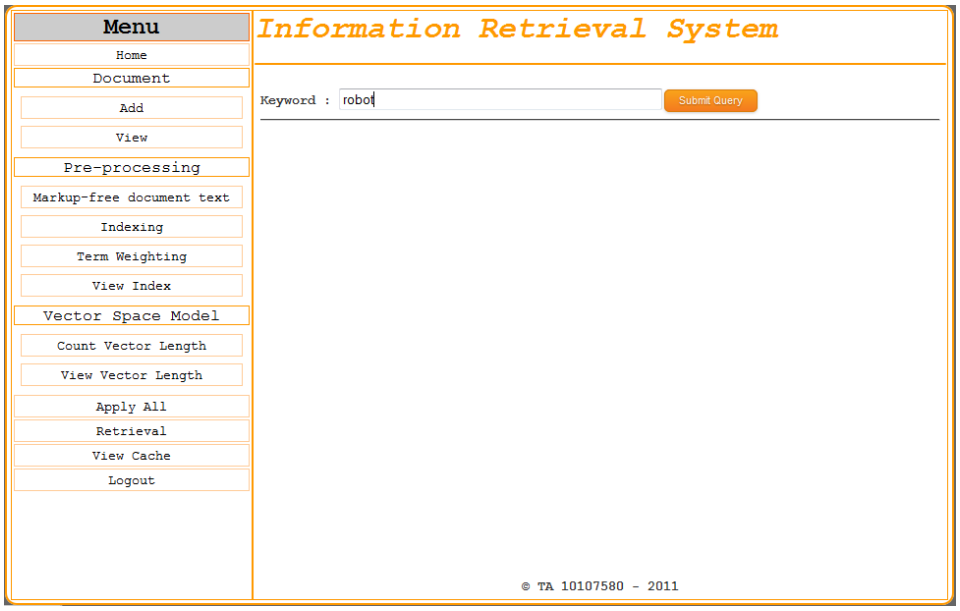
Gambar 4.8 Halaman Tampil Index



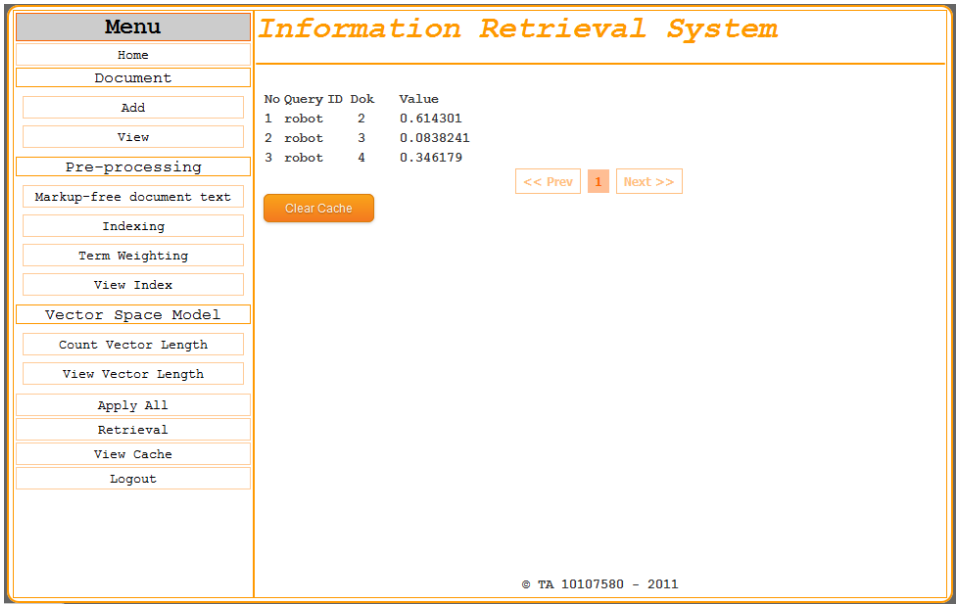
Gambar 4.9 Halaman Hitung Panjang Vektor



Gambar 4.10 Halaman Lihat Panjang Vektor



Gambar 4.11 Halaman Cari Dokumen Berita

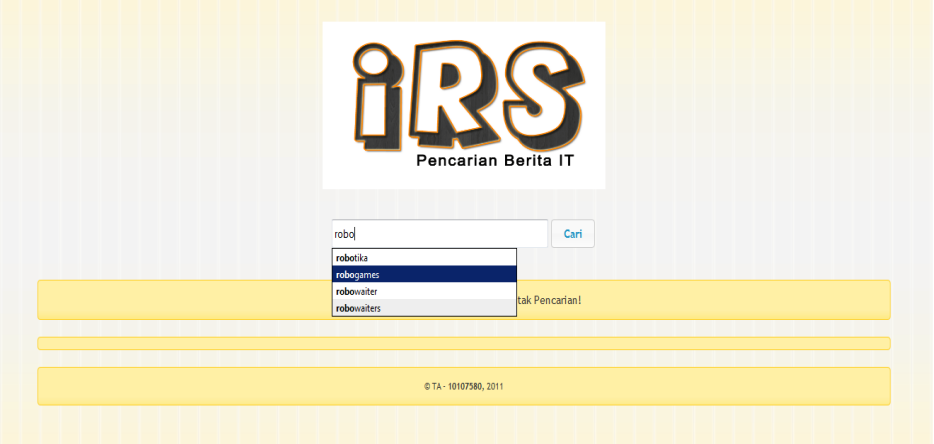


Gambar 4.12 Halaman Cache

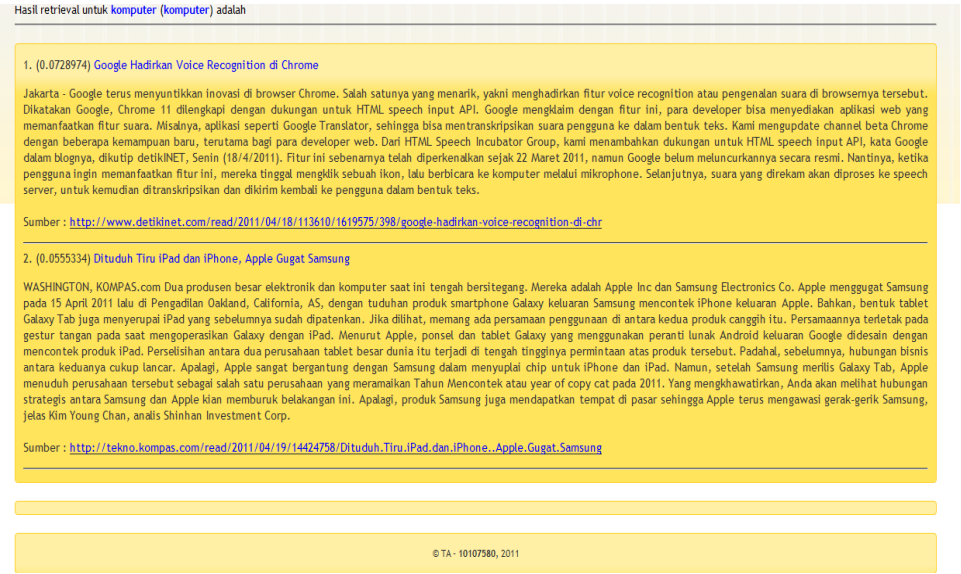
4.1.5.2 Pengguna



Gambar 4.13 Halaman Utama Pengguna



Gambar 4.14 Halaman *Autocomplete Searching*



Gambar 4.15 Halaman Hasil Pencarian

4.2 Pengujian

Pengujian perangkat lunak adalah elemen kritis dari jaminan kwalitas perangkat lunak dan merepresentasikan kajian pokok dari spesifikasi, perancangan, dan pengkodean. Metode pengujian yang digunakan untuk menguji sistem ini adalah pengujian *Black Box.* Pengujian *Black Box* berfokus pada persyaratan fungsional perangkat lunak.

4.2.1 Rencana Pengujian Aplikasi

Rencana pengujian akan dilakukan dengan menguji sistem secara alpha. Pengujian perangkat lunak aplikasi *Information Retrieval System* ini menggunakan data uji dokumen berita IT. Rencana pengujian selengkapnya dapat dilihat pada tabel dibawah ini :

Tabel 4.8 Rencana Pengujian Aplikasi *Information Retrieval System*

Item Uji	Detail Pengujian	Jenis Pengujian
Login	Verifikasi *username*	*Black Box*
	Verifikasi *password*	*Black Box*
Pengolahan Dokumen Berita	Tambah Dokumen Berita	*Black Box*
	Edit Dokumen Berita	*Black Box*
	Hapus Dokumen Berita	*Black Box*
Preprocessing	Hapus *Tag* HTML	*Black Box*
	Tokenization, Filtration dan Stemming	*Black Box*
	Menghitung bobot kata	*Black Box*
Vektor	Menghitung Panjang Vektor	*Black Box*
Retrieval	Mencari isi dokumen berita yang relevan berdasarkan perangkingan	*Black Box*
Cache	Menyimpan keyword	*Black Box*

4.2.2 Kasus dan Hasil Pengujian

4.2.2.1 Pengujian Login

Tabel 4.9 Verifikasi *Username* dan *Password*

Kasus dan Hasil Uji (Data Normal)	
Data masukan	*username* : admin , *password* : admin
Yang diharapkan	*username* dan *password* diterima. Masuk ke halaman utama.
Pengamatan	*username* dan *password* diterima dan masuk ke halaman utama, sesuai yang diharapkan
Kesimpulan	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)	
Data masukan	*username*: testuser ; password : testpass
Yang diharapkan	Kembali ke menu login. Menampilkan pesan gagal: "*username* atau *password* yang anda masukan salah, silahkan *login* kembali"
Pengamatan	Kembali ke menu login dan tampil pesan gagal. Sesuai dengan yang diharapkan.
Kesimpulan	[ ] Diterima [√] Ditolak

Tabel 4.10 Percobaan Masuk Ke Dalam Sistem Tanpa *Login*

Kasus dan Hasil Uji (Data Normal)	
Data masukan	User mengetik alamat salah satu halaman sistem tanpa melalui proses login
Yang diharapkan	Kembali ke menu login. Menampilkan pesan gagal: "*Access denied. Please Login First*"
Pengamatan	User mengetik : 'http://localhost/TA/admin/home.php' pada browser tanpa melakukan proses login, sistem kembali ke menu login dan menampilkan pesan gagal sesuai dengan yang diharapkan.
Kesimpulan	[ ] Diterima [√] Ditolak

4.2.2.2 Pengujian Pengolahan Dokumen Berita

Tabel 4.11 Pengujian Tambah Dokumen Berita

Kasus dan Hasil Uji (Data Normal)	
Data masukan	Sumber Berita, Judul Berita, dan Isi Berita diisi lengkap.
Yang diharapkan	Sumber Berita, Judul Berita, dan Isi Berita disimpan. Kembali ke form tambah dokumen berita. Tampil pesan sukses "Dokumen Berita Telah Disimpan!".
Pengamatan	Data tersimpan, tampilan kembali ke form tambah dokumen berita, dan tampil pesan sukses, sesuai yang diharapkan.
Kesimpulan	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)	
Data masukan	Sumber berita kosong, judul berita kosong, isi berita kosong
Yang diharapkan	Kembali ke form tambah data user simpan. Tampil pesan gagal : "Dokumen berita gagal disimpan, semua form harus diisi".
Pengamatan	Tampilan kembali ke form tambah dokumen berita dan tampil pesan gagal, sesuai yang diharapkan.
Kesimpulan	[ ] Diterima [√] Ditolak

Tabel 4.12 Pengujian *Edit* Dokumen Berita

Kasus dan Hasil Uji (Data Normal)	
Data masukan	Sumber Berita, Judul Berita, dan Isi Berita diisi lengkap.
Yang diharapkan	Dokumen Berita dapat diubah dan disimpan kembali. Tampil pesan sukses "Dokumen Berita Berhasil Diedit!".
Pengamatan	Data berhasil diedit dan tersimpan kembali, tampilan kembali ke form tambah dokumen berita, dan tampil pesan sukses, sesuai yang diharapkan.
Kesimpulan	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)	
Data masukan	Sumber Berita, Judul Berita, dan Isi Berita hilang/terhapus
Yang diharapkan	Kembali ke form tambah data user simpan. Tampil pesan gagal : "Dokumen berita gagal diedit".
Pengamatan	Tampilan kembali ke form tampil dokumen berita dan tampil pesan gagal, sesuai yang diharapkan.
Kesimpulan	[ ] Diterima [√] Ditolak

Tabel 4.13 Pengujian Hapus Dokumen Berita

Kasus dan Hasil Uji (Data Normal)	
Data masukan	Sumber Berita, Judul Berita, dan Isi Berita diisi lengkap.
Yang diharapkan	Akan muncul pesan dialog "Dokumen berita berhasil dihapus"
Pengamatan	muncul pesan dialog "Dokumen berita berhasil dihapus" dan data yang dihapus tidak ada.
Kesimpulan	[√] Diterima [ ] Ditolak

4.2.2.3 Pengujian *Preprocessing*

Tabel 4.14 Pengujian Hapus Tag HTML

Kasus dan Hasil Uji (Data Normal)	
Data masukan	Dokumen Berita
Yang diharapkan	Akan Muncul Pesan sukses "Tag HTML telah dihapus di *n* dokumen"
Pengamatan	Dokumen berita yang mengandung tag html berhasil dihapus
Kesimpulan	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)	
Data masukan	Dokumen berita tidak mengandung *tag html*
Yang diharapkan	Akan muncul pesan "Tidak ada dokumen berita yang mengandung tag html"
Pengamatan	muncul pesan "Tidak ada dokumen berita yang mengandung tag html"
Kesimpulan	[ ] Diterima [√] Ditolak

Tabel 4.15 Pengujian *Tokenization, Filtration, dan Stemming*

Kasus dan Hasil Uji (Data Normal)	
Data masukan	Dokumen Berita
Yang diharapkan	Akan muncul pesan "Mengindeks sebanyak *n* Berita, sudah termasuk proses *tokenization, filtration, dan stemming."*
Pengamatan	Semua dokumen berita sudah terindeks di halaman View Index
Kesimpulan	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)	
Data masukan	Dokumen berita yang berisi kata dasar
Yang diharapkan	Akan muncul *term-term* yang tidak di proses oleh *tokenization, filtration, dan stemming.*
Pengamatan	muncul muncul *term-term* yang tidak di proses oleh *tokenization, filtration, dan stemming.*
Kesimpulan	[ ] Diterima [√] Ditolak

Tabel 4.16 Pengujian Hitung Bobot Kata

Kasus dan Hasil Uji (Data Normal)	
Data masukan	Dokumen Berita
Yang diharapkan	Akan muncul pesan "Terdapat Term yang diberikan bobot"
Pengamatan	Semua kata sudah diberi bobot dan hasil pembobotan ada di halaman View Index
Kesimpulan	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)	
Data masukan	Dokumen berita yang *term*nya tidak ada dalam kamus
Yang diharapkan	Akan muncul pesan "*Term-term* gagal diberi bobot"
Pengamatan	muncul pesan "*Term-term* gagal diberi bobot"
Kesimpulan	[ ] Diterima [√] Ditolak

4.2.2.4 Pengujian Vektor

Tabel 4.17 Pengujian Hitung Panjang Vektor

Kasus dan Hasil Uji (Data Normal)	
Data masukan	Dokumen Berita
Yang diharapkan	Akan muncul pesan "Ada *n* Dokumen yang dihitung panjang vektornya"
Pengamatan	Semua kata sudah dihitung panjang vektornya dan hasil perhitungan dapat dilihat di halaman *View Vector Length*
Kesimpulan	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)	
Data masukan	Dokumen berita yang *term*nya tidak ada dalam kamus
Yang diharapkan	Akan menampilkan nilai hitung panjang vektor tiap dokumen nol
Pengamatan	muncul nilai hitung panjang vektor tiap dokumen nol
Kesimpulan	[ ] Diterima [√] Ditolak

4.2.2.5 Pengujian *Retrieval*

Tabel 4.18 Pengujian Temu Balik Informasi

Kasus dan Hasil Uji (Data Normal)	
Data masukan	Masukkan *Keyword* yang ada dokumen
Yang diharapkan	Dokumen berita yang relevan dengan *keyword*
Pengamatan	Dokumen berita tampil sesuai dengan keyword yang dimasukkan
Kesimpulan	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)	
Data masukan	Masukkan *Keyword* yang tidak ada dalam dokumen
Yang diharapkan	Akan muncul pesan "Tidak ada dokumen berita yang sesuai dengan *keyword*"
Pengamatan	muncul pesan "Tidak ada dokumen berita yang sesuai dengan *keyword*"
Kesimpulan	[ ] Diterima [√] Ditolak

4.2.2.6 Pengujian *Cache*

Tabel 4.19 Pengujian *cache*

Kasus dan Hasil Uji (Data Normal)	
Data masukan	Masukkan *Keyword*
Yang diharapkan	*Keyword* yang relevan dengan dokumen berita akan tersimpan di cache
Pengamatan	*Keyword* yang relevan dengan dokumen berita tersimpan dan ada di halaman *view cache*
Kesimpulan	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)	
Data masukan	Masukkan *Keyword*
Yang diharapkan	*Keyword* yang tidak relevan dengan dokumen berita akan tersimpan di cache
Pengamatan	*Keyword* yang tidak relevan dengan dokumen berita tidak disimpan dan tidak ada di halaman *view cache*
Kesimpulan	[ ] Diterima [√] Ditolak

4.2.3 Kesimpulan Hasil Pengujian

Berdasarkan hasil pengujian dengan kasus uji diatas dapat ditarik kesimpulan sebgai berikut :

1. Jika data yang dimasukan benar maka sistem secara fungsional akan mengeluarkan keluaran sesuai dengan harapan.
2. Jika data yang dimasukan salah maka sistem akan mengeluarkan pesan kesalahan dan data keluaran tidak akan sesuai dengan harapan.

Dengan kata lain bahwa pembangunan Sistem ini bebas kesalahan dan secara fungsional mengeluarkan hasil yang sesuai dengan yang diharapkan.