BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Pada bagian ini akan dilakukan analisis terhadap *Information Retrieval System* (IRS) dengan menggunakan Model Ruang Vektor. Hasilnya digunakan untuk mengidentifikasi dan mengevaluasi hasil implementasi dan pengujian yang akan dibahas pada bab berikutnya. Dari proses analisis akan dapat dihasilkan berbagai macam saran perbaikan terhadap sistem yang dapat dijadikan dasar dalam perancangan *Information Retrieval System.*

3.1.1 Analisis Masalah

Dari hasil survey IDC *(Internasional Data Corporation)* dapat diketahui bahwa dokumen yang terdapat dalam dunia maya terus bertambah. Hal ini menimbulkan kesulitan dalam menemubalikkan dokumen-dokumen tersebut. Faktor utama dalam proses pencarian suatu dokumen di dunia maya adalah bagaimana dokumen tersebut ditemubalikkan sesuai atau relevan dengan *keyword* atau *query* yang di inputkan oleh *user* yang akan mencari dokumen berita tersebut.

Oleh karena itu, berbagai macam penelitian di area ini telah banyak dilakukan, salah satunya adalah melalui disiplin ilmu IR (*Information Retrieval).*IR merupakan bidang yang mengkaji metode-metode di dalam pencarian dokumen

berdasarkan representasi kebutuhan informasi berupa kata kunci,yaitu *keyword* atau *query*.

3.1.2 Masukan Perangkat Lunak

Input dari suatu sistem temu balik informasi ini adalah query dari pengguna dan koleksi dokumen. Koleksi dokumen yang digunakan dalam sistem temu balik informasi ini yaitu *content* berita IT berbahasa indonesia yang dapat dikumpulkan secara manual dari situs penyedia berita di internet.

3.1.3 Proses Yang Terjadi Didalam Perangkat Lunak

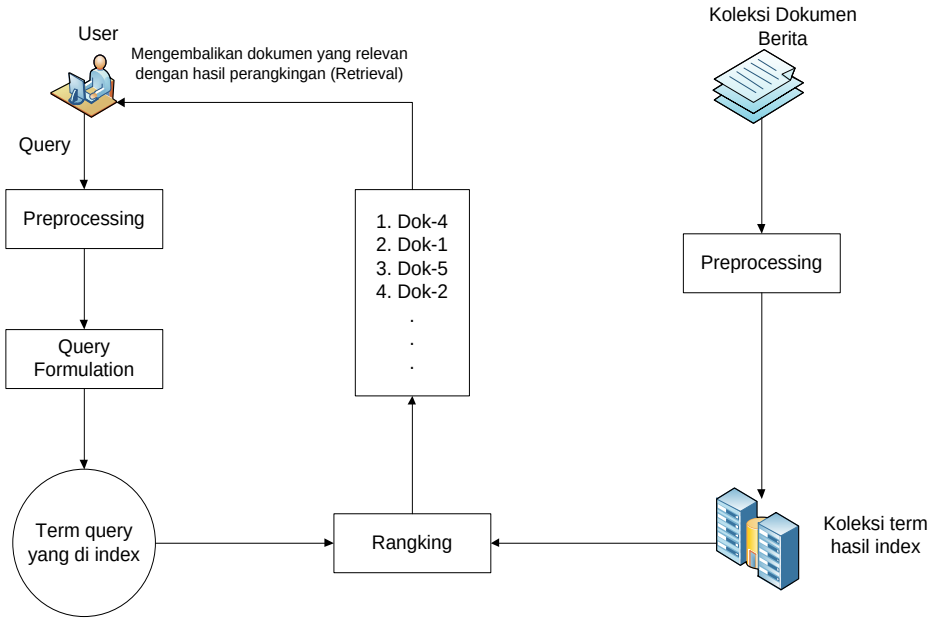
Perangkat lunak dalam skripsi ini merupakan aplikasi *Information Retrieval System* dengan menggunakan Model Ruang Vektor dan metode pembobotan *Term Frequency–Inverse Document Frequency* (TF_IDF).

Information Retrieval System ini secara umum terdapat 2 tahapan besar, yaitu *preprocessing/indexing* dan *retrieval.* Pada tahapan *preprocessing* mencakup penghapusan tag, *tokenization, filtration, stemming*, dan *weigthing.* Kemudian dilanjutkan dengan menyimpan setiap *term* yang penting ke dalam suatu *index* (dalam sistem ini menggunakan suatu tabel pada *database* MySQL).

Information Retrieval System ini akan menerima *query* dari pengguna kemudian memproses *query* tersebut dan sistem akan melakukan perhitungan kemiripan antara *query* dengan daftar berita yang tersedia, untuk menghasilkan

kumpulan dokumen yang relevan dan terurut berdasarkan ranking kerelevanannya dengan *query* dari pengguna.

Berikut ini adalah gambar dari deskripsi umum sistem :



Gambar 3.1 Proses Yang Terjadi Dalam Sistem

Dalam *preprocessing* dilakukan beberapa tahapan untuk melakukan pengindeksan dokumen berita diantarany penghapusan *tag html*, *tokenization*, *filtration*, *stemming* dan *weighting*.

Berikut ini adalah penjelasan tahapan-tahapan *preprocessing* yang akan diterapkan dalam sistem :

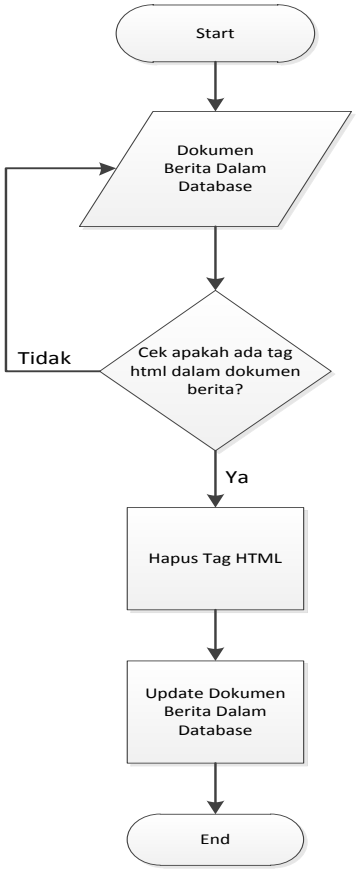
3.1.3.1 Hapus *Tag HTML*

Tahap ini menghapus semua *tag markup* dan format khusus dari dokumen, terutama pada dokumen yang mempunyai banyak *tag* dan format seperti dokumen (X)HTML.

Untuk penghapusan tag HTML menggunakan fungsi PHP strip_tags().

Pseudocode : `strip_tags("<b>google</b>");`

Dibawah ini adalah *Flowchart* alur hapus *tag HTML* :



Gambar 3.2 *Flowchart* Hapus *Tag HTML*

Berikut ini adalah Contoh penghapusan *tag HTML* pada dokumen berita :

Dokumen berita :

Dua layanan populer milik **Google**, ***Picasa*** dan ***Blogger*** tidak lama lagi akan lenyap. Bukan karena **Google** menghapusnya, tetapi karena raksasa internet itu ingin 'membuang' kedua nama produk itu untuk diganti namanya.

Sebelum *tag html* dihapus :

Dua layanan populer milik Google, Picasa dan Blogger tidak lama lagi akan lenyap. Bukan karena Google menghapusnya, tetapi karena raksasa internet itu ingin 'membuang' kedua nama produk itu untuk diganti namanya.

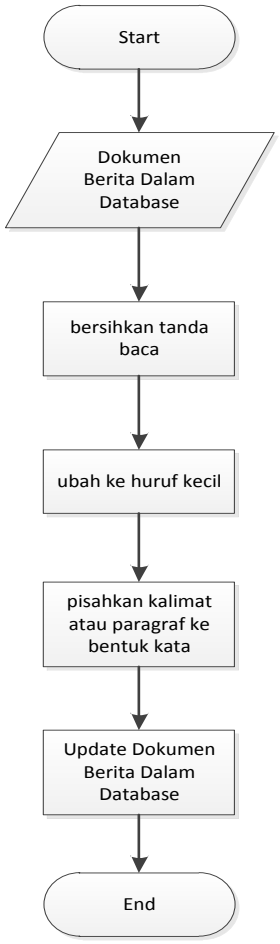
Setelah tag html dihapus :

Dua layanan populer milik Google, Picasa dan Blogger tidak lama lagi akan lenyap. Bukan karena Google menghapusnya, tetapi karena raksasa internet itu ingin 'membuang' kedua nama produk itu untuk diganti namanya.

Dari contoh diatas bisa diketahui ada 4 kata yang mengandung tag html yang telah dihapus markup tag htmlnya.

3.1.3.2 *Tokenization*

Tokenization adalah tugas memisahkan deretan kata di dalam kalimat, paragraf atau halaman menjadi *token* atau potongan kata tunggal atau *termmed word*. Tahapan ini juga menghilangkan karakter-karakter tertentu seperti tanda baca dan mengubah semua *token* ke bentuk huruf kecil (*lower case*). Dibawah ini adalah *Flowchart* alur *Tokenization* :



Gambar 3.3 *Flowchart Tokenization*

Untuk pemisahan kalimat ke bentuk kata menggunakan fungsi PHP explode().

Pseudocode :

```
explode("  ", "Dua layanan populer milik Google");
```

Contoh *tokenization* :

Input :

Dua layanan populer milik Google, Picasa dan Blogger tidak lama lagi akan lenyap. Bukan karena Google menghapusnya, tetapi karena raksasa internet itu ingin 'membuang' kedua nama produk itu untuk diganti namanya.

Output :

dua	layanan	populer	milik	google	Picasa	dan
Blogger	tidak	lama	lagi	akan	lenyap	bukan
karena	google	menghapusnya	tetapi	karena	raksasa	
internet	itu	ingin	membuang	kedua	nama	
produk	itu	untuk	diganti	namanya		

3.1.3.3 *Filtration*

Filtration adalah proses penghapusan kata buang yaitu kata sambung, kata depan, kata ganti, dll. Contoh *stop words* dalam bahasa Indonesia : yang, juga, dari, dia, kami, kamu, aku, saya, ini, itu, atau, dan, tersebut, pada, dengan, adalah, yaitu, ke, tak, tidak, di, pada, jika, maka, ada, pun, lain, saja, hanya, namun, seperti, kemudian, karena, untuk, dll. Proses *filtration* menggunakan fungsi PHP str_replace().

Dibawah ini adalah *array stopword* yang telah di inputkan beserta contoh penggunaan fungsi str_replace() :

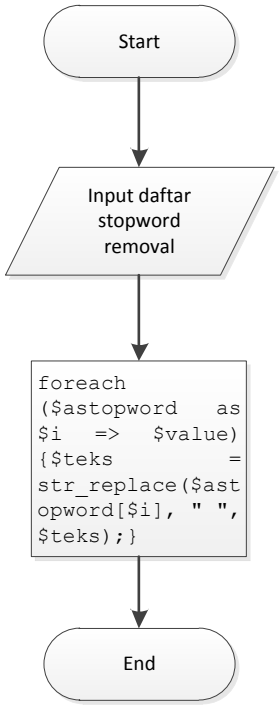
```
$astopword = array (" yang ", " juga ", " dari ", " dia ", " kami ", " kamu ", "aku", " ini ", " itu ", " atau ", " dan ", " tersebut ", " pada ", " dengan ", " adalah ", " yaitu ", " ke ", " tak "," tidak "," di ", " pada "," jika "," maka "," ada ", " pun "," lain "," saja ", " hanya "," namun "," seperti "," kemudian "," untuk "," sudah ", " belum ", " memang "," sama ", " apa "," apakah "," oleh ", " sangat "," sehingga "," memang "," dimana "," sedangkan ", " selain "," tapi "," tetapi "," tanpa "," tiada "," dengan "," lain "," akan "," lainnya "," tiada "," para ", " saat "," sebelum "," sesudah "," karena ");
```

Data array akan dibaca oleh fungsi foreach :

```
foreach ($astopword as $i => $value) {
      $teks = str_replace($astopword[$i], " ", $teks);
}
```

$teks adalah dokumen berita yana akan diproses oleh *filtration.*

Dibawah ini *flowchart* proses *filtration* :



Gambar 3.4 *Flowchart Filtration*

Contoh Sebelum *filtration* :

dua layanan populer milik google picasa dan blogger tidak lama lagi akan lenyap bukan karena google menghapusnya tetapi karena raksasa internet itu ingin membuang kedua nama produk itu untuk diganti namanya

Setelah *Filtration* :

dua layanan populer milik google picasa blogger lama lenyap google menghapus raksasa internet ingin membuang kedua nama produk diganti nama

3.1.3.4 *Stemming*

Stemming adalah pengubahan kata ke bentuk kata dasar atau penghapusan imbuhan.

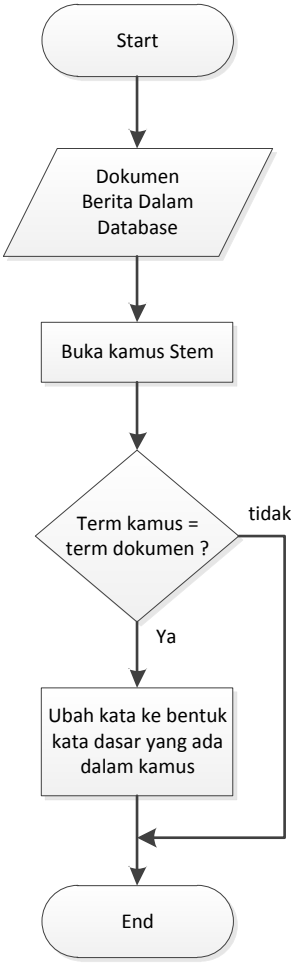
Stemming disini menggunakan kamus daftar kata berimbuhan yang mempunyai kata dasarnya dengan cara membandingkan kata-kata yang ada dalam dokumen berita dengan daftar kamus *stem*. Proses *stemming* menggunakan fungsi PHP str_replace. Berikut ini adalah contoh penggunaanya :

```
//buka tabel tbstem dan bandingkan dengan berita
$restem = mysql_query("SELECT * FROM stem ORDER BY id");

while($rowstem = mysql_fetch_array($restem)) {
      $teks=str_replace($rowstem['term'],$rowstem['stem'],
$teks);

}
```

Dibawah ini *flowchart* proses *stemming* :



Gambar 3.5 *Flowchart Stemming*

Contoh kamus *stemming* :

Tabel 3.1 Kamus *Stem*

No	Term	Stem
1	layanan	Layan
2	menghapus	Hapus
3	membuang	Buang
4	Kedua	Dua
5	Diganti	Ganti
6	namanya	Nama

Contoh Stemming :

dua layanan populer milik google picasa blogger lama lenyap google menghapus raksasa internet ingin membuang kedua nama produk diganti nama

Term yang berubah oleh proses stemming :

layanan ⇨ layan

menghapus ⇨ hapus

membuang ⇨ buang

kedua ⇨ dua

diganti ⇨ ganti

namanya ⇨ nama

Jadi berita yang telah di stemming adalah :

dua layan popular milik google picasa blogger lama lenyap google hapus raksasa internet ingin buang dua nama produk ganti nama

3.1.3.5 *Weighting*

Pembobotan kata sangat berpengaruh dalam menentukan kemiripan antara dokumen dengan *query*. Apabila bobot tiap kata dapat ditentukan dengan tepat, diharapkan hasil perhitungan kemiripan teks akan menghasilkan perangkingan dokumen yang baik.

Bobot *term* di dalam *Information Retrieval System* (W) dihitung menggunakan *tf-idf* yang didefinisikan sebagai berikut.

$$W_{=} tf \times idf \quad (3.1)$$

Diketahui :

W = Bobot *term*

tf = banyaknya kemunculan suatu kata dalam suatu dokumen.

Idf = $log_2 (\frac{N}{d_{ji}})$ (3.2)

Diketahui :

N = Jumlah artikel dalam koleksi dokumen

d_{ji} = Banyaknya Dokumen (*j)* yang mengandu *term (i)*

Log_2 = digunakan untuk memperkecil pengaruhnya relatif terhadap tf_{ij}

Contoh kasus Terdapat 3 dokumen berita yaitu :

D_1 : dua layan popular milik google picasa blogger lama lenyap

D_2 : picasa blogger milik google

D_3 : nama google picasa ubah dalam cari google

Diketahui :

D = 3 (Jumlah Dokumen)

Idf = $log_2 (\frac{N}{d_{ji}})$

d_{ji} = Banyaknya Dokumen (*j)* yang mengandu *term (i)*

Tf D_1 = Banyaknya muncul kata di dokumen 1

Tf D_2 = Banyaknya muncul kata di dokumen 2

Tf D_3 = Banyaknya muncul kata di dokumen 3

W D_1 = Bobot kata di dokumen 1

W D_2 = Bobot kata di dokumen 2

W D_3 = Bobot kata di dokumen 3

Tabel 3.2 Pembobotan Kata

Kata	Tf D_1	Tf D_2	Tf D_3	d_{ji}	$\frac{N}{d_{ji}}$	Idf	W D_1	W D_2	W D_3
Dua	1	0	0	1	3/1 = 3	0.48	0.48	0	0
Layan	1	0	0	1	3/1 = 3	0.48	0.48	0	0
Popular	1	0	0	1	3/1 = 3	0.48	0.48	0	0
Milik	1	1	0	2	3/2 = 1.5	0.18	0.18	0.18	0
Google	1	1	2	4	¾ = 0.75	0.12	0.12	0.12	0.24
Picasa	1	1	1	3	3/3 = 1	0	0	0	0
Blogger	1	1	0	2	3 / 2 =1.5	0.18	0.18	0.18	0
Lama	1	0	0	1	3/1 =3	0.48	0.48	0	0
Lenyap	1	0	0	1	3/1 = 3	0.48	0.48	0	0
Nama	0	0	1	1	3/1 = 3	0.48	0	0	0.48
Ubah	0	0	1	1	3/1 = 3	0.48	0	0	0.48
Dalam	0	0	1	1	3/1 = 3	0.48	0	0	0.48
Cari	0	0	1	1	3/1 = 3	0.48	0	0	0.48

3.1.4 Keluaran Perangkat Lunak

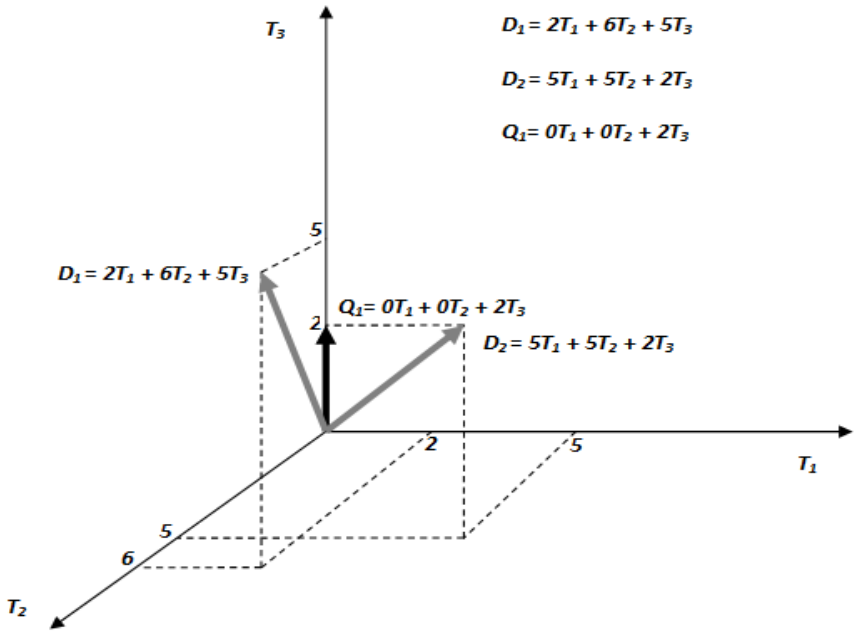
Output yang diharapkan dari *Information Retrieval System* ini adalah dokumen yang dianggap relevan oleh sistem berupa berita yang telah terurutberdasarkan relevansi dengan *query* dari pengguna.

3.1.5 Analisis Model Ruang Vektor

3.1.5.1 Ukuran Kemiripan

Setelah menghitung pembobotan dilakukan suatu perhitungan ukuran kemiripan antar dokumen dengan query. Ukuran ini memungkinkan perankingan dokumen sesuai dengan kemiripan relevansinya terhadap *query*. Setelah dokumen diranking, sejumlah tetap dokumen *top-scoring* dikembalikan kepada pengguna.

Berikut ini adalah contoh perhitungan kemiripan :



Gambar 3.6 Contoh Model Ruang Vektor dengan dua dokumun D_1 dan D_2, serta *query* Q_1. [1]

Jika dua dokumen D1 = 2T1 + 6T2 + 5T3 dan D2 = 5T1 + 5T2 + 2T3 dan query Q1 = 0T1 + 0T2 + 2T3 sebagaimana diperlihatkan pada Gambar 3.6, berikut ini adalah nilai kosinus yang diperoleh:

$$Similarity(\overrightarrow{D_1},\overrightarrow{Q_1}) = \cos\theta = \frac{(2.0 + 6.0 + 5.2)}{\sqrt{(4 + 36 + 25).(0 + 0 + 4)}} = \frac{10}{\sqrt{65.4}} = 0.62$$

$$Similarity(\overrightarrow{D_2},\overrightarrow{Q_1}) = \cos\theta = \frac{(5.0 + 5.0 + 2.2)}{\sqrt{(25 + 25 + 4).(0 + 0 + 4)}} = \frac{4}{\sqrt{54.4}} = 0.27$$

Contoh di atas memperlihatkan bahwa sesuai dengan perhitungan kosinus, dokumen D_2 lebih mirip dengan *query* daripada dokumen D_1. Terlihat sudut antara D_2 dan Q_1 lebih kecil daripada sudut antara D_1 dan Q_1.

3.1.6 Analisis dan Kebutuhan Fungsional Perangkat Lunak

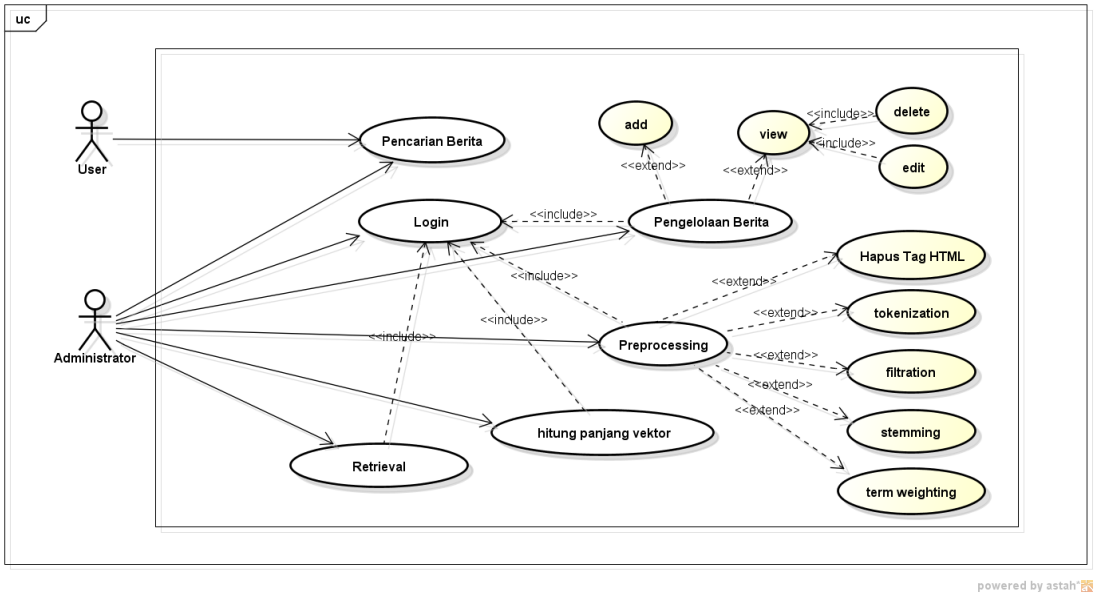
Analisis kebutuhan fungsional menggambarkan proses kegiatan yang akan diterapkan dalam sebuah sistem dan menjelaskan kebutuhan yang diperlukan sistem agar sistem dapat berjalan dengan baik.

Analisis yang dilakukan dimodelkan dengan menggunakan UML (*Unified Modeling Language*). Tahap-tahap pemodelan dalam analisis tersebut antara lain identifikasi aktor, *usecase diagram*, *skenario, activity diagram, sequence diagram*, *class diagram*.

3.1.6.1 *Use Case Diagram*

Pemodelan *use case* adalah pemodelan sistem dari perspektif pandangan pemakai akhir (*end user*). Model *use case* adalah pandangan dari luar sistem, sementara model rancangan adalah pandangan dari dalam. Model *use case* menangkap penggunaan-penggunaan sistem, sedangkan model rancangan merepresentasikan pembangunan dari sistem.

Dibawah ini adalah *Use Case Diagram Information Retrieval* :



Gambar 3.7 *Use Case Diagram Information Retrieval System*

3.1.6.2 Identifikasi Aktor

Terdapat dua aktor yang berhubungan dengan *Information Retrieval System.* Aktor dapat diidentifikasikan sebagai berikut :

a. Aktor pertama ialah *Administrator*, mempunyai hak akses untuk pengelolaan dokumen berita, melakukan *indexing*, dan menerapkan model ruang vektor pada dokumen berita juga mempunyai hak akses ke halaman *user* untuk melakukan pencarian.
b. Aktor kedua ialah *User*, hanya mempunyai hak akses di halaman *user* untuk melakukan pencarian dokumen berita.

3.1.6.3 Identifikasi *Use Case*

Tabel 3.3 Identifikasi *Use Case Information Retrieval System*

No.	*Use Case*	Deskripsi
1	*Login*	Proses ini melakukan autentikasi terhadap aktor yang menjalankan sistem untuk memastikan bahwa orang yang bersangkutan memiliki hak akses terhadap sistem.
2	*Add*	Proses bagi admin untuk menambahkan dokumen berita
3	*View*	Proses bagi admin untuk melihat dokumen berita
4	*Edit*	Proses bagi admin untuk mengubah dokumen berita
5	*Delete*	Proses bagi admin untuk menghapus dokumen berita
6	Hapus *tag html*	Proses untuk menghapus *tag html*
7	*Tokenization*	Proses untuk memecah dokumen berita menjadi potongan-potongan kata
8	*Filtration*	Proses untuk *stopword* menghilangkan kata-kata yang tidak diperlukan untuk pengindeksan dan menghilangkan tanda baca.
9	*Stemming*	Proses untuk mengubah kata ke bentuk kata dasar atau penghapusan imbuhan.
10	*Term weighting*	Proses untuk pemberian bobot tf.idf terhadap term
11	Hitung panjang vektor	Proses untuk menghitung panjang vektor dokumen
12	*Retrieval*	Proses bagi admin untuk menemubalikkan dokumen berita.
13	Pencarian Berita	Proses bagi user untuk pencarian dokumen berita

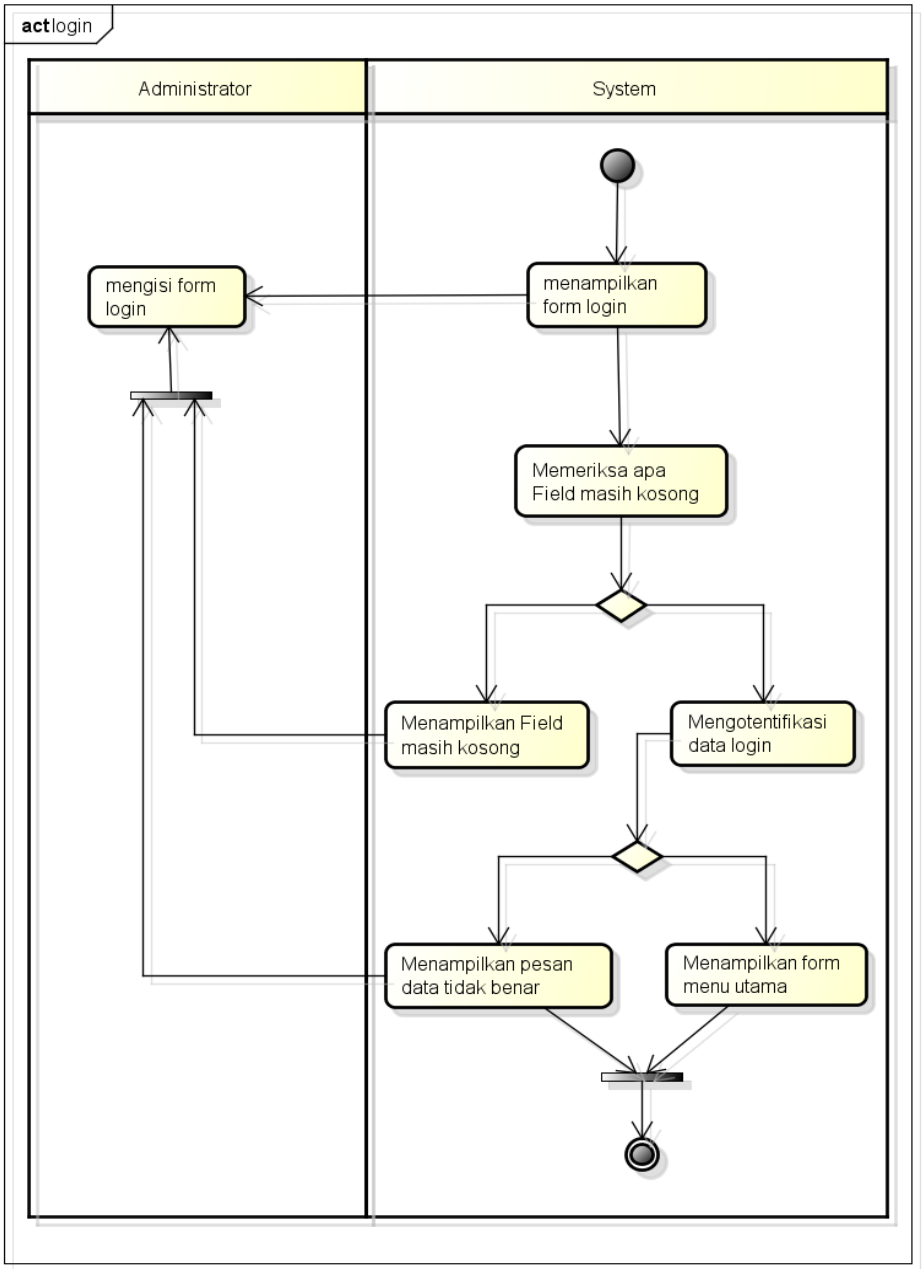
3.1.6.4 *Use Case Login*

Proses ini melakukan autentikasi terhadap aktor yang menjalankan sistem untuk memastikan bahwa orang yang bersangkutan memiliki hak akses terhadap sistem.

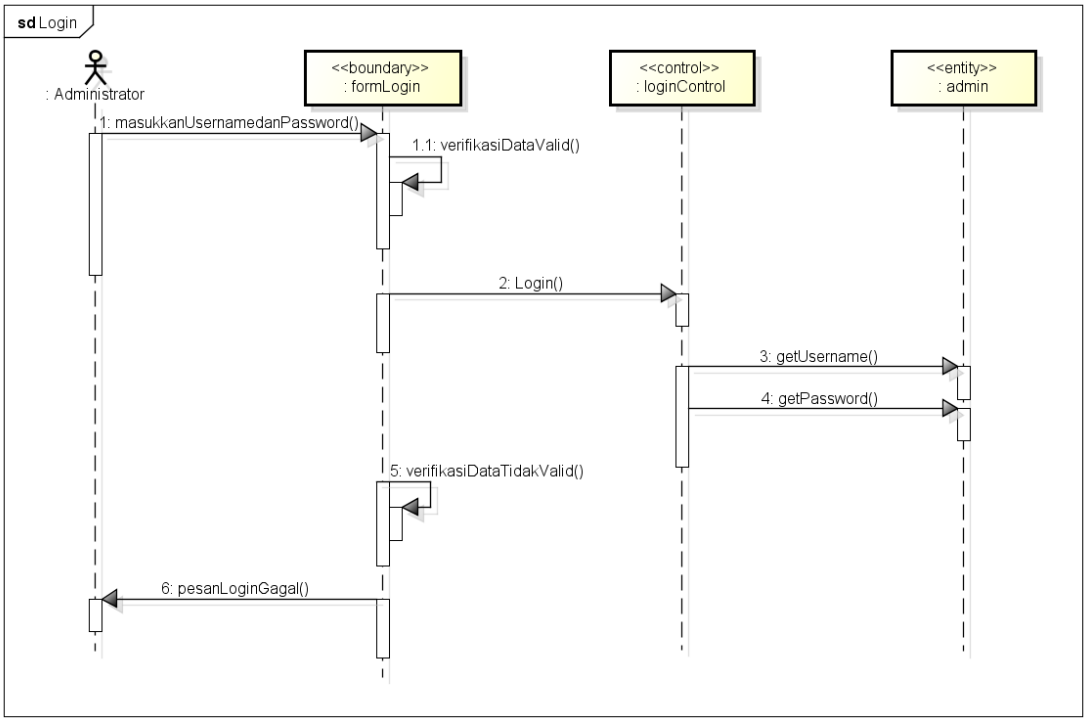
Use case login diterangkan dan dimodelkan dengan skenario, *activity diagram* dan *sequence diagram.*

Tabel 3.4 Skenario *Use Case* Login

Identifikasi	
Nama	*Login*
Tujuan	Masuk ke dalam sistem
Deskripsi	Proses *login* merupakan proses autentifikasi untuk menggunakan sistem.
Aktor	*Administrator*
Skenario Utama	
Kondisi awal	*Form login* ditampilkan
Aksi actor	Reaksi sistem
	1) Menampilkan *form login*
2) Mengisi *Form Login*	2) Cek *field* masih kosong
	3) Mengautentifikasi data *login* dengan data *username* pada basis data
	4) Bila cocok sistem menampilkan halaman menu utama
Skenario Alternatif (Auntentifikasi Gagal)	
Aksi Aktor	Reaksi Sistem
	1) Menampilkan Pesan bahwa data *username* tidak benar
2) Mengisi kembali *Form Login*	3) Mengautentifikasi data *username* dan *password* pada basis data
	4) Bila cocok sistem menampilkan halaman menu utama
Kondisi akhir	Menu utama ditampilkan



Gambar 3.8 *Activity Diagram* Login



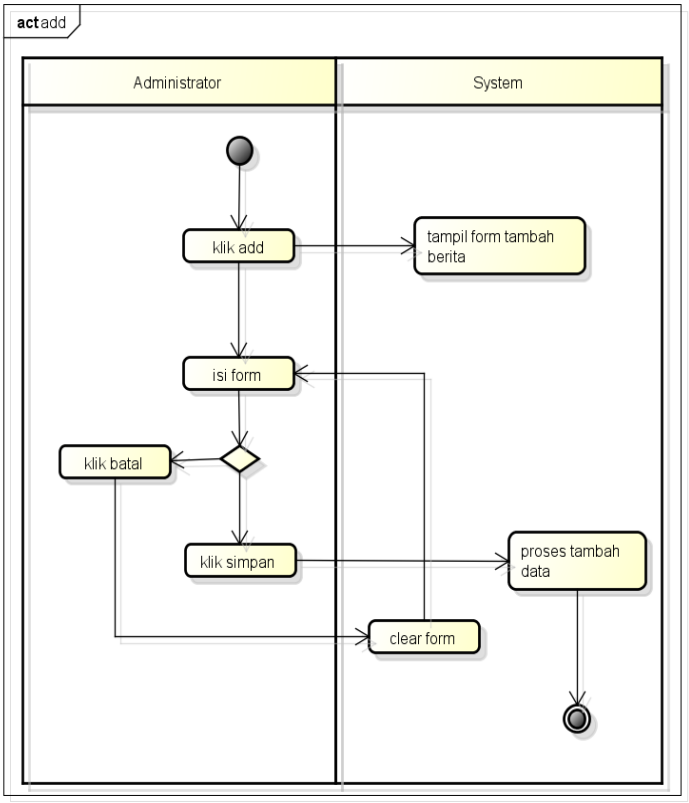
Gambar 3.9 ***Sequence Diagram Login***

3.1.6.5 *Use Case Add*

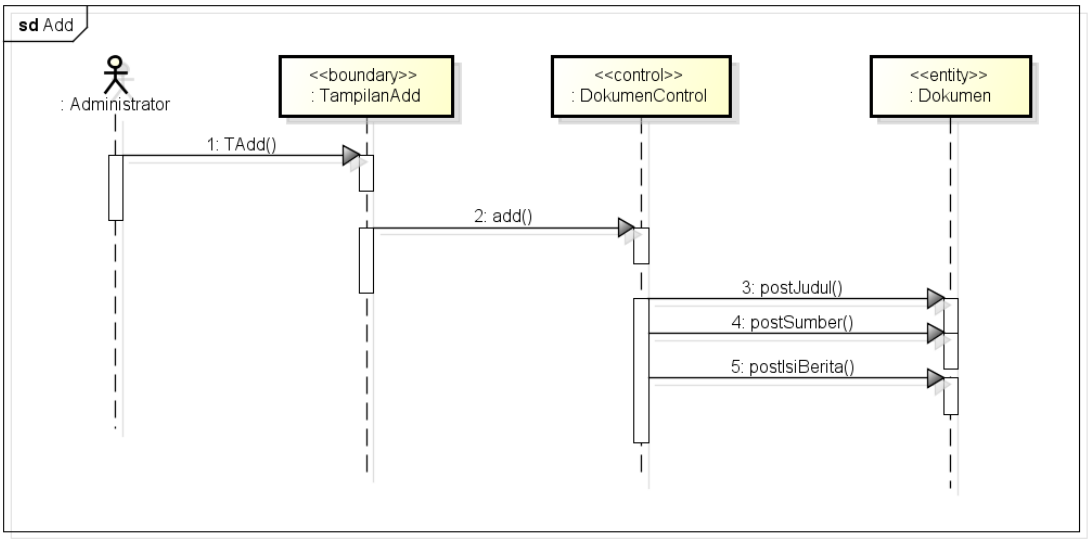
Proses ini melakukan penambahan dokumen berita. *Use case add* diterangkan dan dimodelkan dengan skenario, *activity diagram* dan *sequence diagram.*

Tabel 3.5 Skenario *Use Case Add*

Identifikasi	
Nama	*Add*
Tujuan	Menambah dokumen berita
Deskripsi	Proses *Add* merupakan proses untuk penambahan dokumen berita ke dalam *Information Retrieval System*
Aktor	*Administrator*
Skenario Utama	
Kondisi awal	Klik tombol add
Aksi aktor	Reaksi sistem
1. Klik add	2. Menampilkan *Form* tambah dokumen berita.
3. Mengisi *Form* Dokumen berita	
4. Klik tombol simpan	5. Memproses penambahan data
Skenario Alternatif (Auntentifikasi Gagal)	
Aksi Aktor	Reaksi Sistem
1. Klik tombol batal	2. Kosongkan *form*
Kondisi akhir	Tampil Sukses



Gambar 3.10 *Activity Diagram* Add



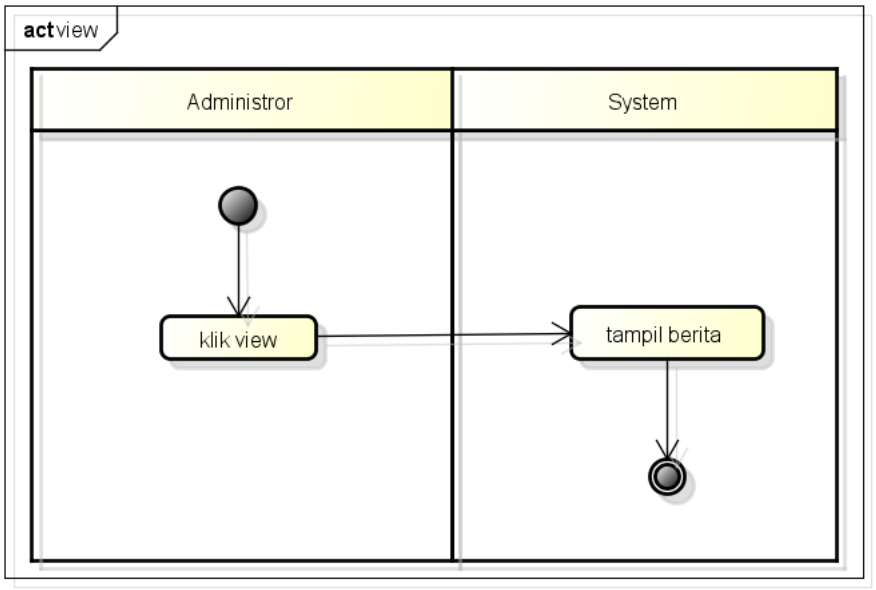
Gambar 3.11 *Sequence Diagram Add*

3.1.6.6 *Use Case View*

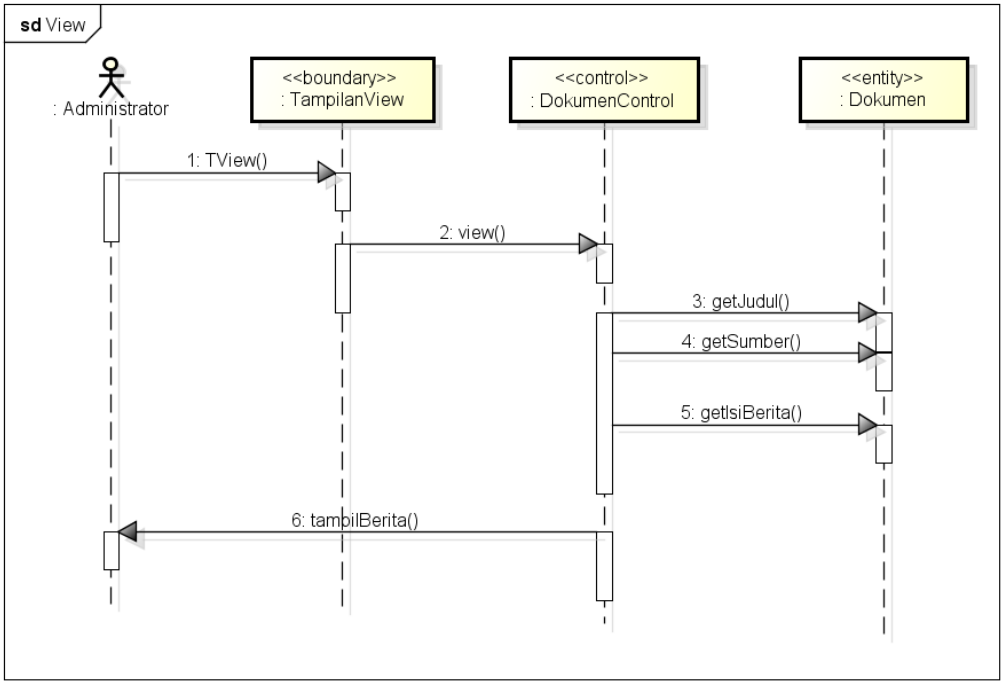
Proses ini menampilkan dokumen berita. *Use case* view diterangkan dan dimodelkan dengan skenario, *activity diagram* dan *sequence diagram.*

Tabel 3.6 Skenario *Use Case* View

Identifikasi	
Nama	*View*
Tujuan	Menampilkan dokumen berita
Deskripsi	Proses *View* untuk menampilkan berita.
Aktor	*Administrator*
Skenario Utama	
Kondisi awal	Klik tombol *view*
Aksi actor	Reaksi sistem
1. Klik view	2. Tampilkan dokumen berita
Kondisi akhir	Admin berhasil *view* berita



Gambar 3.12 *Activity Diagram View*



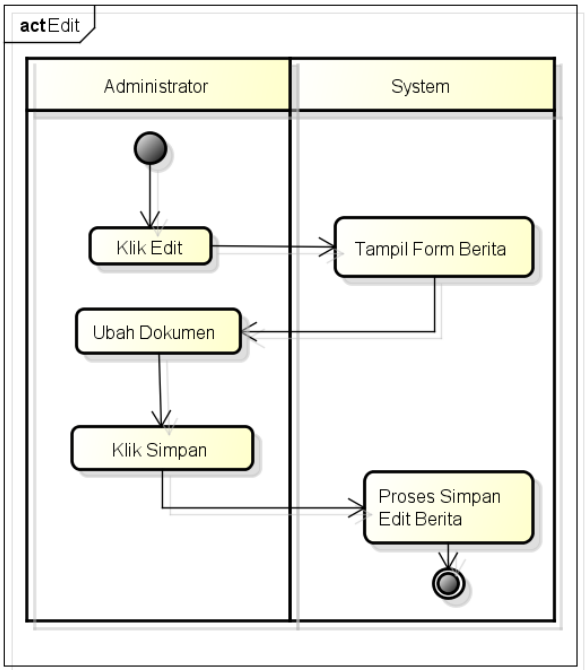
Gambar 3.13 *Sequence Diagram View*

3.1.6.7 *Use Case Edit*

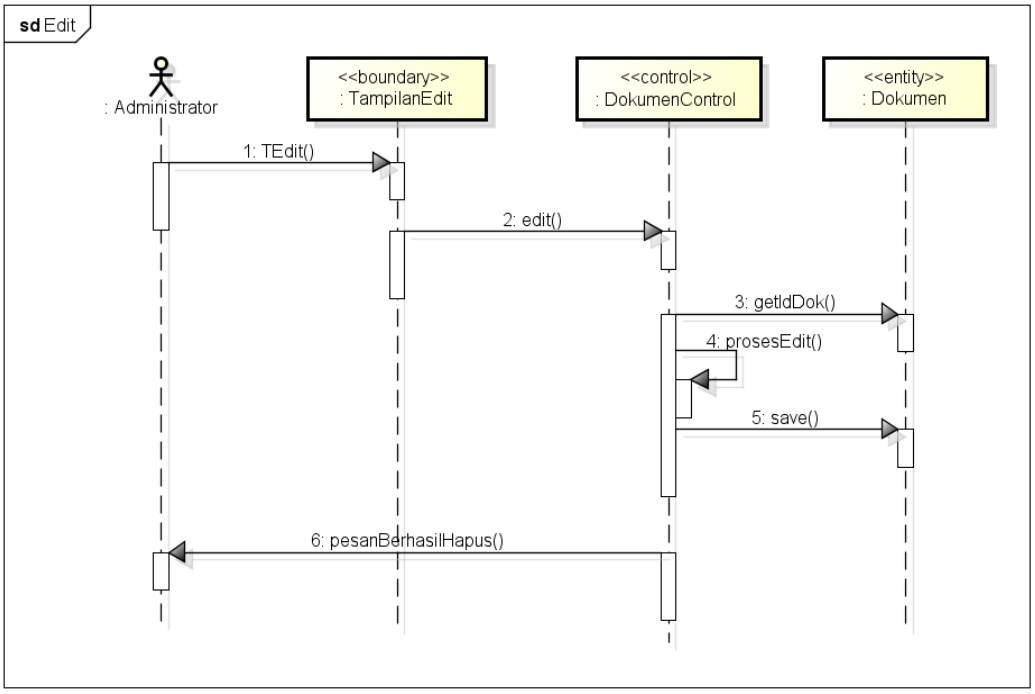
Proses ini untuk mengubah dokumen berita. *Use case edit* diterangkan dan dimodelkan dengan skenario, *activity diagram* dan *sequence diagram.*

Tabel 3.7 Skenario *Use Case Edit*

Identifikasi	
Nama	*Edit*
Tujuan	Mengubah dokumen berita
Deskripsi	Proses untuk mengubah dokumen berita
Aktor	*Administrator*
Skenario Utama	
Kondisi awal	Klik tombol *view*
Aksi actor	Reaksi sistem
1. Klik *Edit*	2. Tampilkan *form* Edit Berita
3. Ubah Dokumen Berita	
4. Klik Simpan	5. Simpan Hasil Edit
Kondisi akhir	Admin berhasil mengubah dokumen berita



Gambar 3.14 ***Activity Diagram Edit***



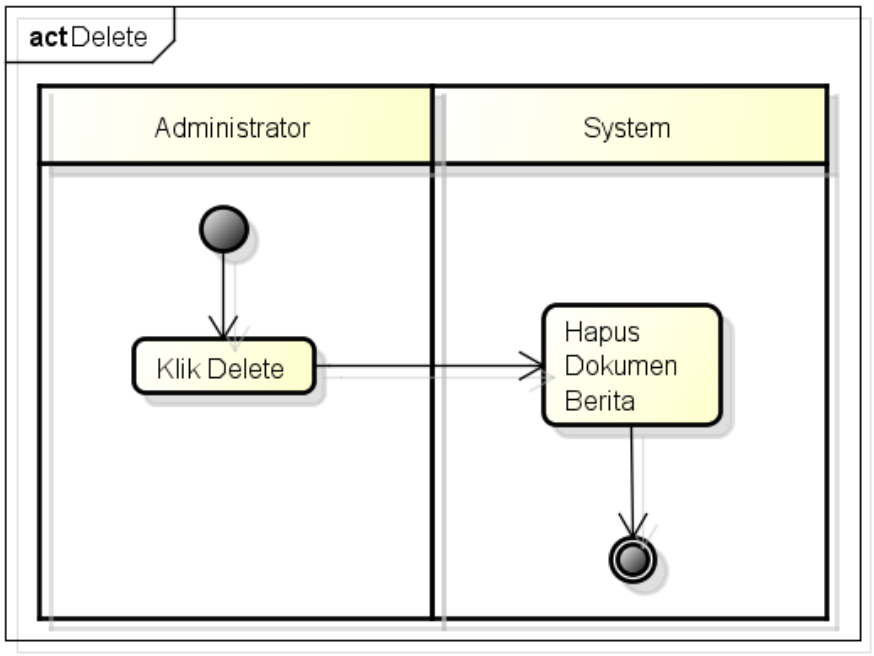
Gambar 3.15 ***Sequence Diagram Edit***

3.1.6.8 *Use Case Delete*

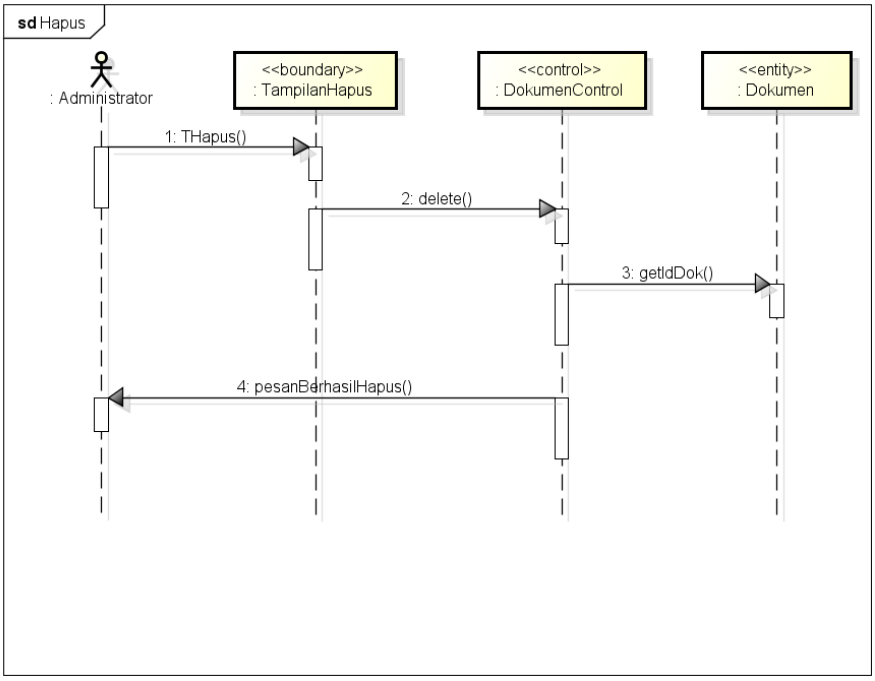
Proses ini untuk mengubah dokumen berita. *Use case* edit diterangkan dan dimodelkan dengan skenario, *activity diagram* dan *sequence diagram.*

Tabel 3.8 Skenario *Usecase Delete*

Identifikasi	
Nama	*Delete*
Tujuan	Menghapus dokumen berita
Deskripsi	Proses untuk menghapus dokumen berita
Aktor	*Administrator*
Skenario Utama	
Kondisi awal	Klik tombol *delete*
Aksi actor	Reaksi sistem
1. Klik *delete*	2. Hapus Dokumen berita
Kondisi akhir	Admin berhasil menghapus dokumen berita



Gambar 3.16 *Activity Diagram Delete*



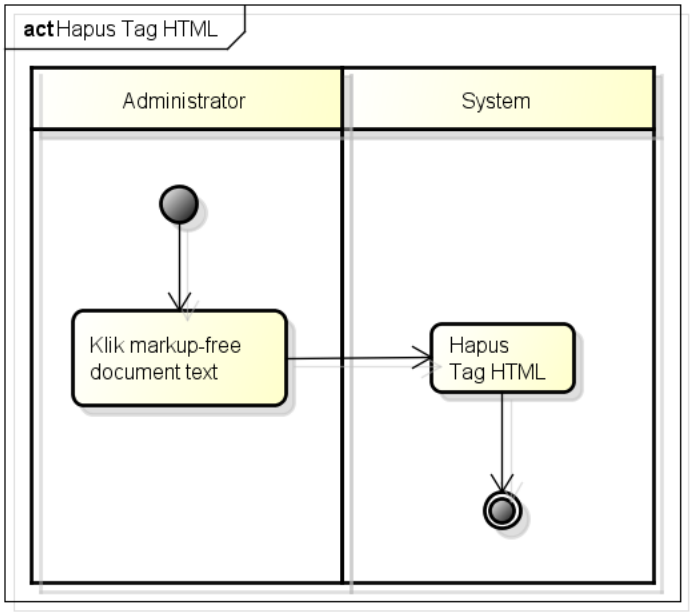
Gambar 3.17 *Sequence Diagram Delete*

3.1.6.9 *Use Case* Hapus *Tag HTML*

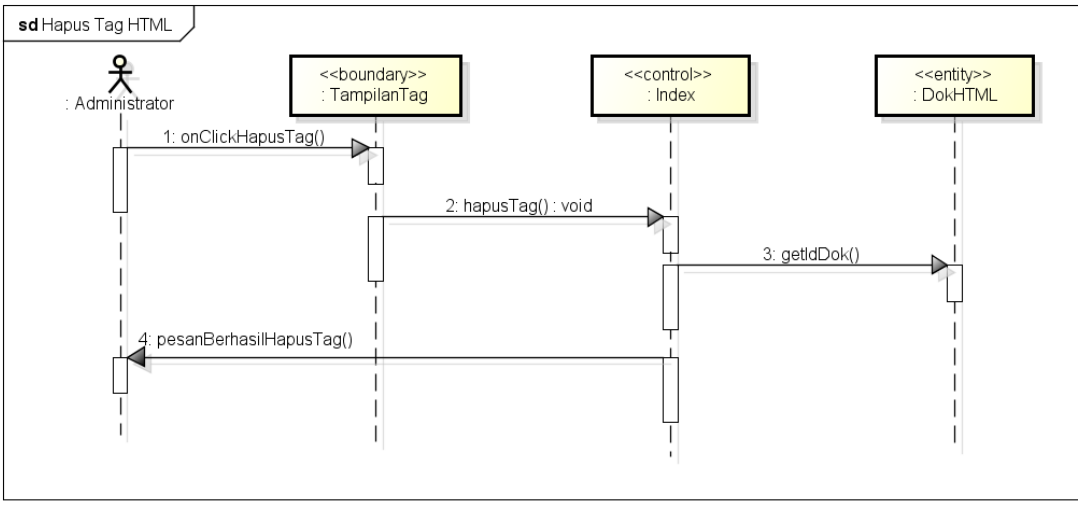
Proses untuk menghitung panjang vektor dokumen berita. *Use case* hitung panjang vektor diterangkan dan dimodelkan dengan skenario, *activity diagram* dan *sequence diagram*

Tabel 3.9 Skenario *Use Case* Hapus *Tag HTML*

Identifikasi	
Nama	Hapus *Tag HTML*
Tujuan	Menghapus *Tag HTML*
Deskripsi	Proses untuk menghapus *Tag HTML*
Aktor	*Administrator*
Skenario Utama	
Kondisi awal	Klik tombol *Markup-free document text*
Aksi actor	Reaksi sistem
1. Klik *Markup-free document text*	2. Hapus *Tag HTML*
Kondisi akhir	Admin berhasil menghapus dokumen berita



Gambar 3.18 ***Activity Diagram*** **Hapus** ***Tag HTML***



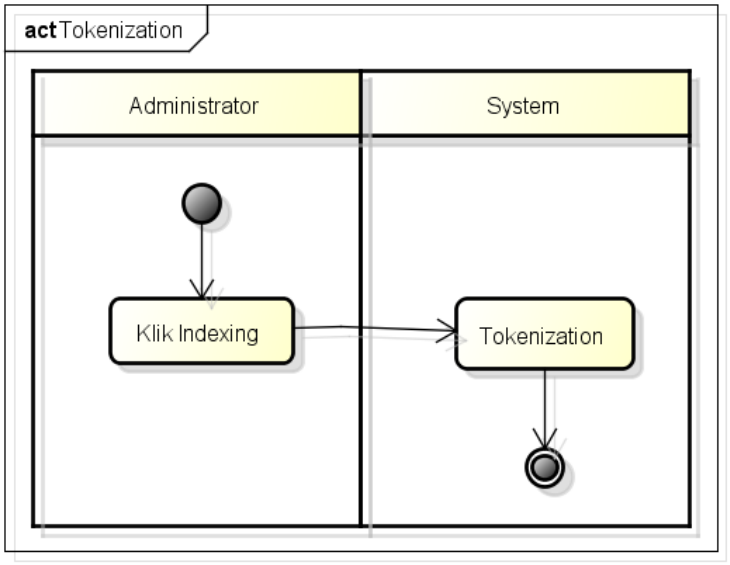
Gambar 3.19 ***Sequence Diagram*** **Hapus** ***Tag HTML***

3.1.6.10 *Use Case Tokenization*

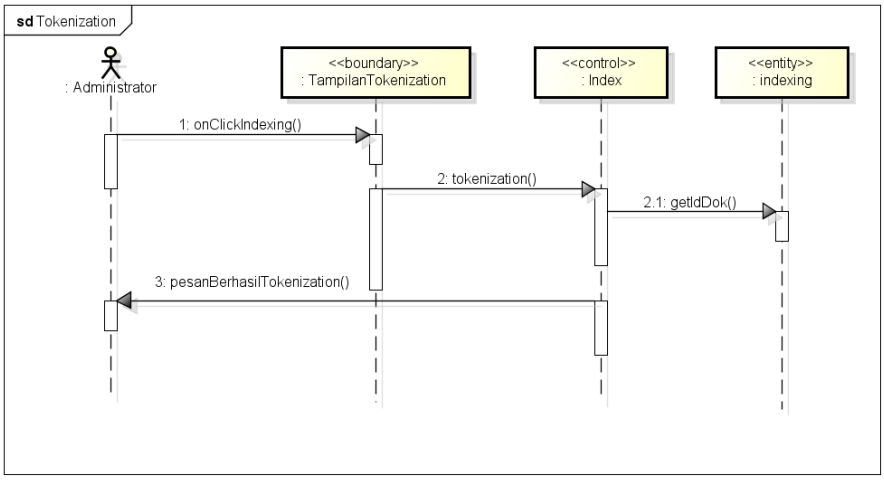
Proses untuk memecah dokumen berita menjadi potongan-potongan kata. *Use case Tokenization* diterangkan dan dimodelkan dengan skenario, *activity diagram* dan *sequence diagram.*

Tabel 3.10 Skenario *Use Case Tokenization*

Identifikasi	
Nama	*Tokenization*
Tujuan	Memecah dokumen berita menjadi potongan-potongan kata
Deskripsi	Proses untuk memecah dokumen berita menjadi potongan-potongan kata.
Aktor	*Administrator*
Skenario Utama	
Kondisi awal	Dokumen berita belum terpecah menjadi potongan-potongan kata
Aksi aktor	Reaksi sistem
1. Klik *Indexing*	2. Dokumen berita dipecah menjadi potongan-potongan kata
Kondisi akhir	Admin berhasil memecah dokumen berita menjadi setiap potongan kata.



Gambar 3.20 *Activity Diagram Tokenization*



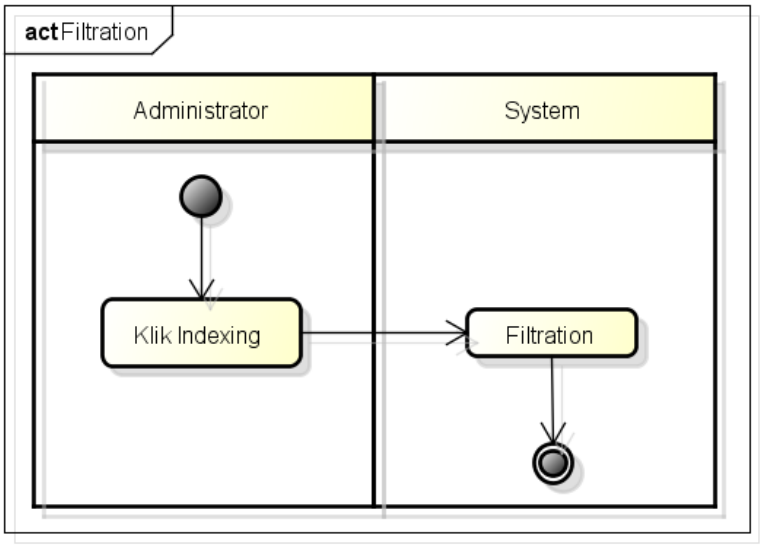
Gambar 3.21 *Sequence Diagram Tokenization*

3.1.6.11 *Use Case Filtration*

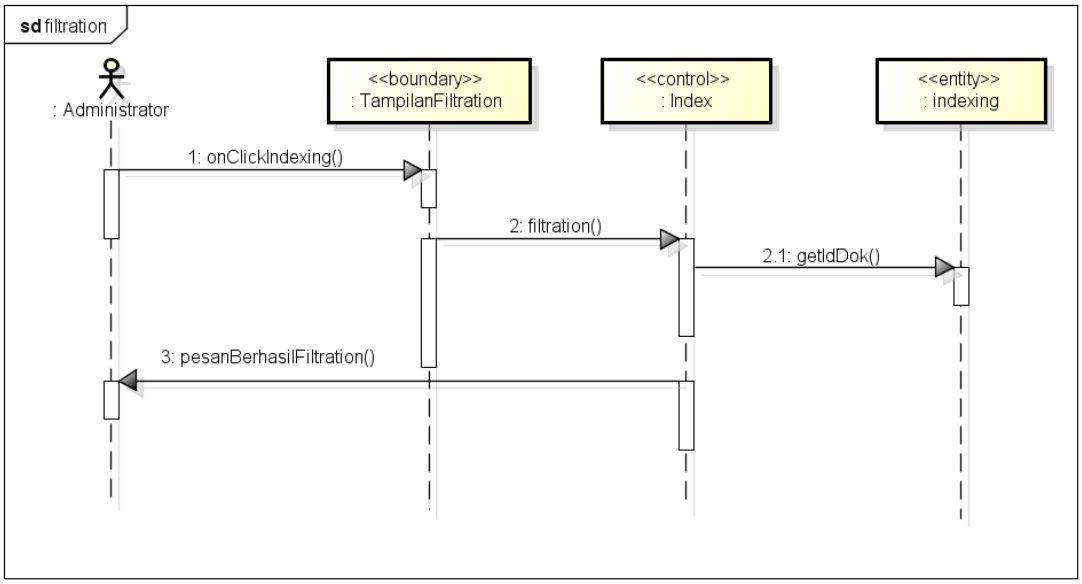
Proses untuk *stopword* menghilangkan kata-kata yang tidak diperlukan dalam pengindeksan dan menghilangkan tanda baca. *Use case filtration* diterangkan dan dimodelkan dengan skenario, *activity diagram* dan *sequence diagram.*

Tabel 3.11 *Skenario Use Case Filtration*

Identifikasi	
Nama	*Filtration*
Tujuan	Menghilangkan kata-kata yang tidak diperlukan dalam pengindeksan dan menghilangkan tanda baca
Deskripsi	Proses untuk *stopword* menghilangkan kata-kata yang tidak diperlukan dalam pengindeksan dan menghilangkan tanda baca
Aktor	*Administrator*
Skenario Utama	
Kondisi awal	Masih ada kata-kata yang tidak diperlukan dan tanda baca.
Aksi aktor	Reaksi sistem
1. Klik *Indexing*	2. Hilangkan tanda baca dan kata-kata yang tidak diperlukan dalam pengindeksan.
Kondisi akhir	Admin berhasil menghapus tanda baca dan menghilangkan kata-kata yang tidak perlu dalam indeks



Gambar 3.22 ***Activity Diagram Filtration***



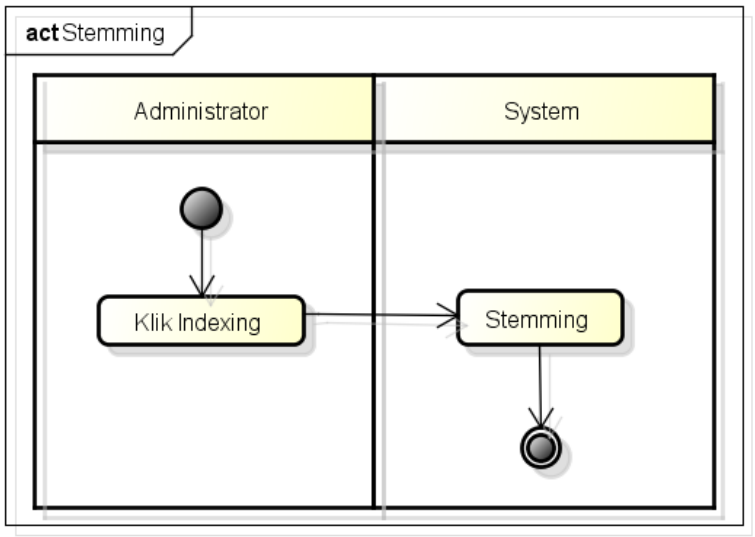
Gambar 3.23 ***Sequence Diagram Filtration***

3.1.6.12 *Use Case Stemming*

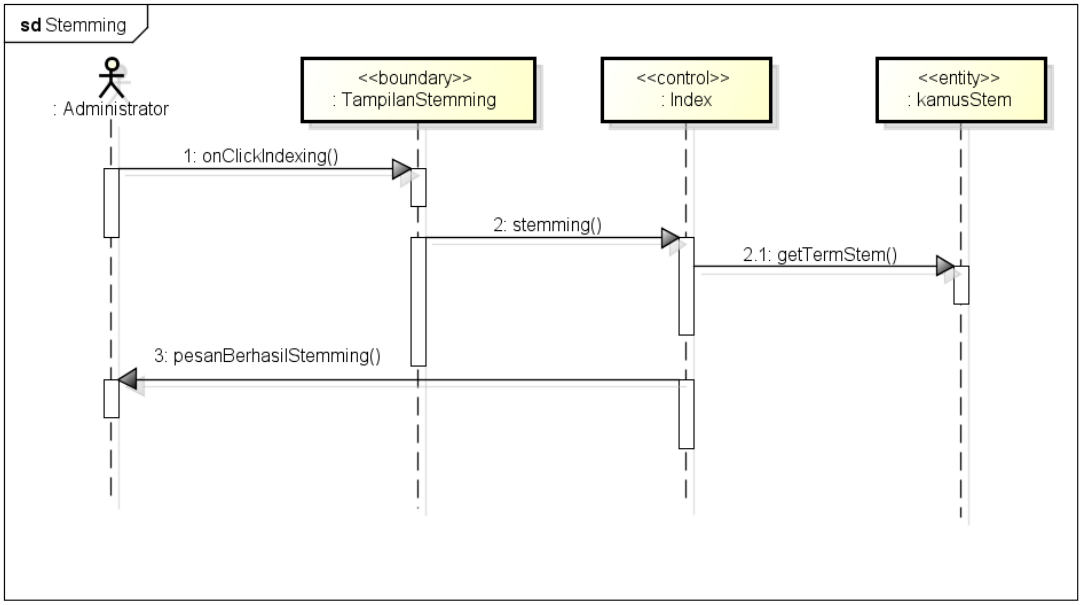
Proses untuk mengubah kata ke bentuk kata dasar atau penghapusan imbuhan. *Use case stemming* diterangkan dan dimodelkan dengan skenario, *activity diagram* dan *sequence diagram.*

Tabel 3.12 Skenario *Use Case Stemming*

Identifikasi	
Nama	*Stemming*
Tujuan	untuk mengubah kata ke bentuk kata
Deskripsi	Proses untuk mengubah kata ke bentuk kata dasar atau penghapusan imbuhan.
Aktor	*Administrator*
Skenario Utama	
Kondisi awal	Kata yang mengandung imbuhan belum diubah ke bentuk kata dasar
Aksi aktor	Reaksi sistem
1. Klik Indexing	2. Ubah kata yang mengandung imbuhan ke bentuk kata dasar
Kondisi akhir	Admin berhasil mengubah kata yang mengandung imbuhan ke bentuk kata dasar



Gambar 3.24 *Activity Diagram Stemming*



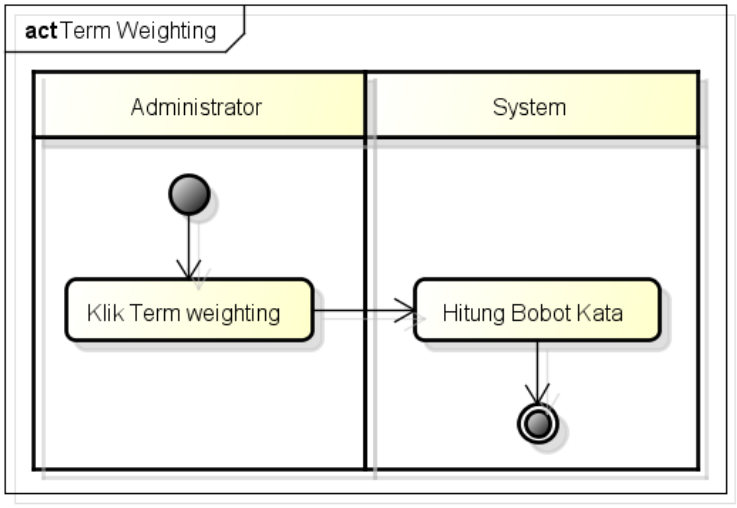
Gambar 3.25 *Sequence Diagram Stemming*

3.1.6.13 *Use Case Term Weighting*

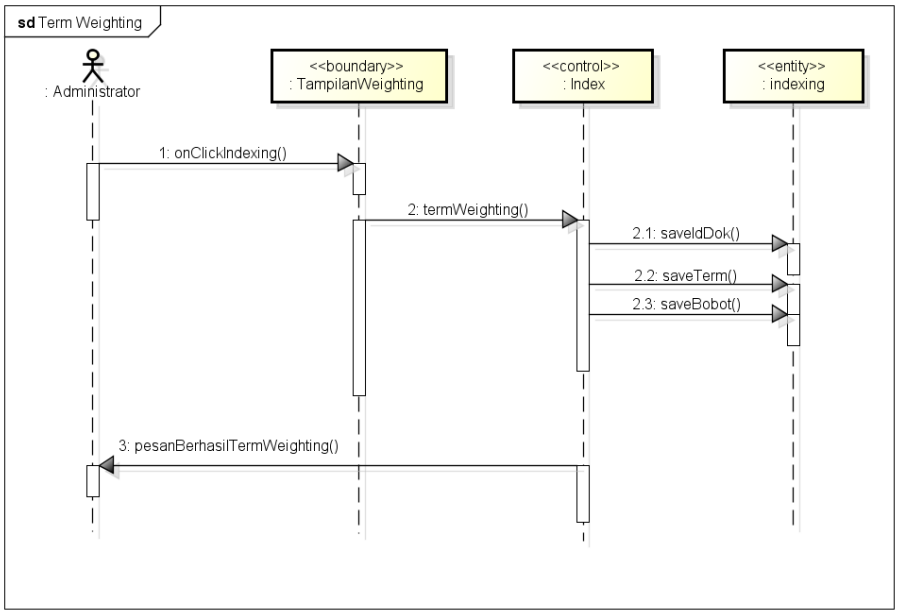
Proses untuk menghitung bobot kata. *Usecase term weighting* diterangkan dan dimodelkan dengan skenario, *activity diagram* dan *sequence diagram.*

Tabel 3.13 Skenario *Use Case Term Weighting*

Identifikasi	
Nama	*Term Weighting*
Tujuan	untuk menghitung bobot kata
Deskripsi	Proses untuk menghitung bobot kata
Aktor	*Administrator*
Skenario Utama	
Kondisi awal	Nilai bobot nol
Aksi aktor	Reaksi sistem
1. Klik Term Weighting	2. Hitung Bobot Setiap Kata
Kondisi akhir	Admin berhasil menghitung bobot setiap kata



Gambar 3.26 ***Activity Diagram Term Weighting***



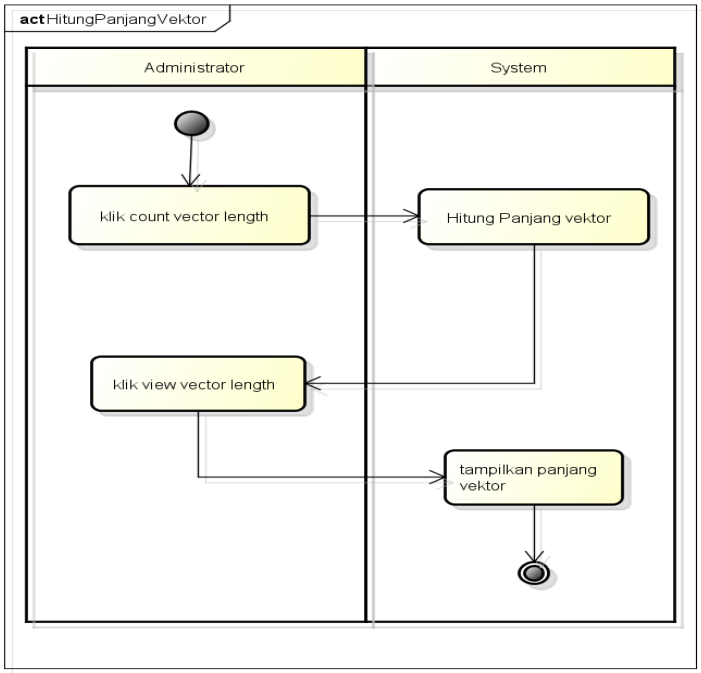
Gambar 3.27 ***Sequence Diagram Term Weighting***

3.1.6.14 *Use Case* Hitung Panjang Vektor

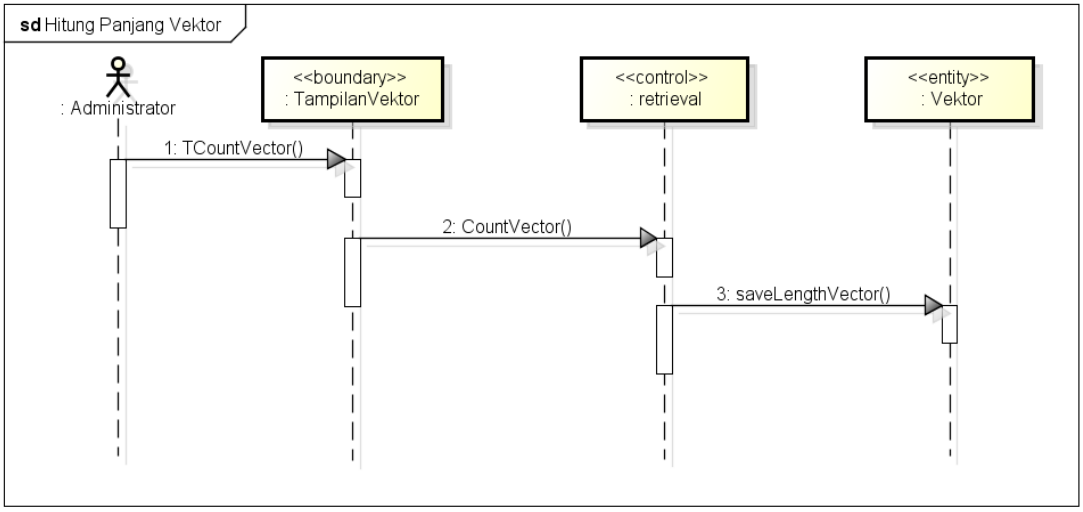
Proses untuk menghitung panjang vektor dokumen berita. *Use case* hitung panjang vektor diterangkan dan dimodelkan dengan skenario, *activity diagram* dan *sequence diagram.*

Tabel 3.14 Skenario *Use Case* Hitung Panjang Vektor

Identifikasi	
Nama	Hitung Panjang Vektor
Tujuan	Menghitung Panjang Vektor
Deskripsi	Menghitung Panjang Vektor Dokumen berita dan menampilkannya.
Aktor	*Administrator*
Skenario Utama	
Kondisi awal	Tombol belum diklik
Aksi aktor	Reaksi sistem
1. klik *Count Vector Length*	2. Hitung Panjang Vektor
1. klik *view vector length*	2. Tampilkan panjang vektor
Kondisi akhir	Menampilkan panjang vektor



Gambar 3.28 *Activity Diagram* Hitung Panjang Vektor



Gambar 3.29 *Sequence Diagram* Hitung Panjang Vektor

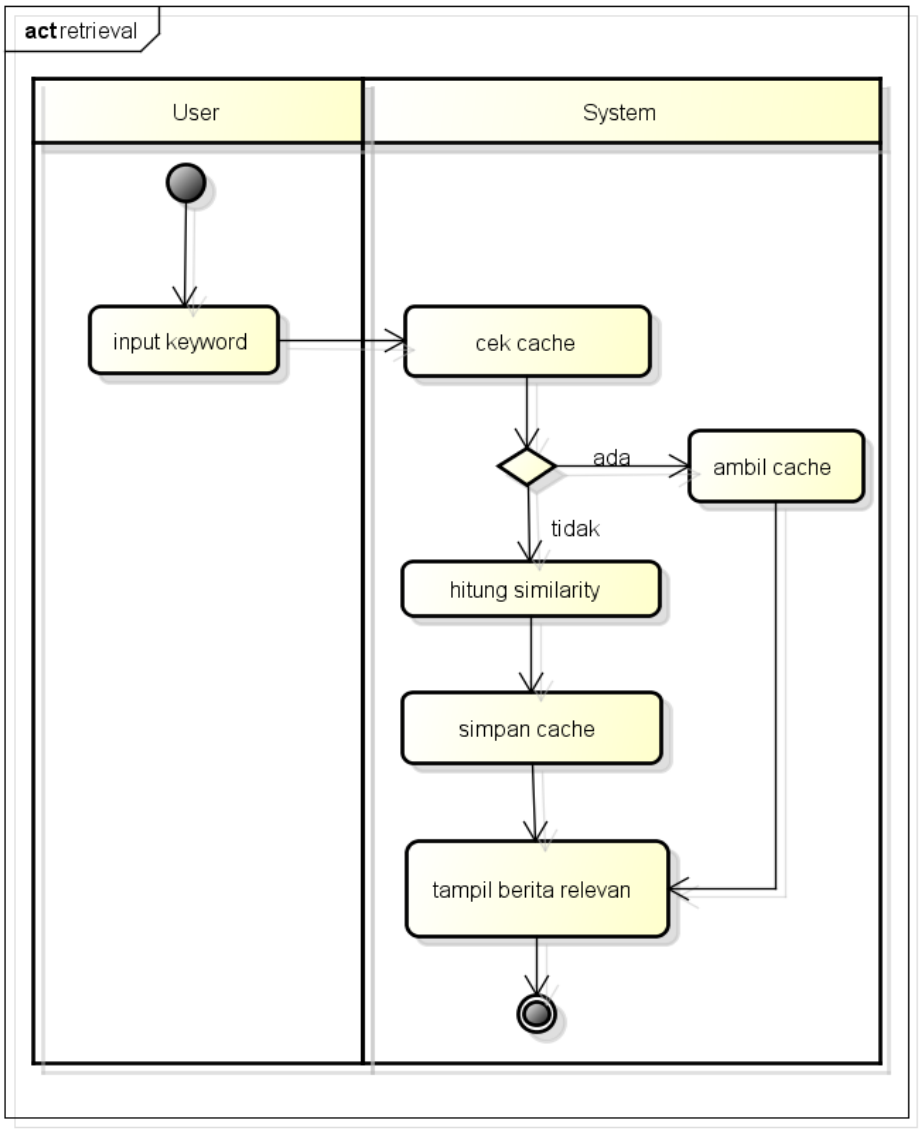
3.1.6.15 *Use Case Retrieval*

Proses yang digunakan oleh administrator untuk menemubalikkan dokumen berita. Fungsi ini sama seperti fungsi pencarian berita yang melakukan model ruang vektor untuk perhitungan kemiripan antara *query* dan dokumen. *Use Case* retrieval diterangkan dan dimodelkan dengan skenario, *activity diagram* dan *sequence diagram.*

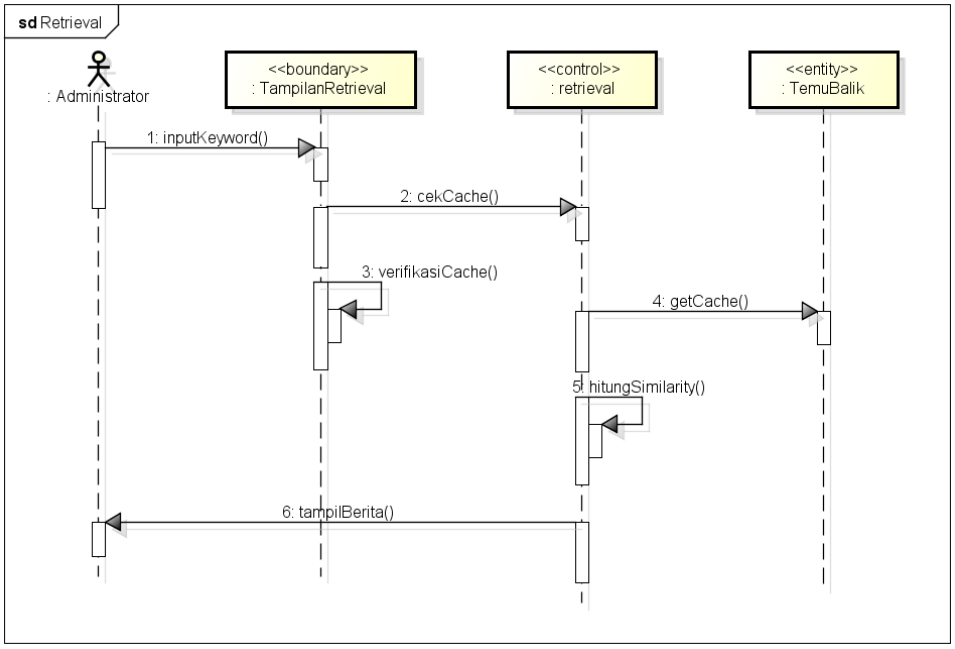
Tabel 3.15 Skenario *Use Case* Retrieval

Identifikasi	
Nama	Pencarian Berita
Tujuan	Mencari berita
Deskripsi	Melakukan pencarian berita yang relevan.
Aktor	*Administrator,user*
Skenario Utama	
Kondisi awal	*Form* pencarian
Aksi actor	Reaksi sistem
1. masukan *keyword/query*	2. Sistem melakukan pengecekan pada daftar cache
	3. menyatakan *query* ada pada daftar *cache*
	4. menampilkan berita relevan
Kondisi akhir	Berita yang relevan dengan query ketemu.
Skenario Alternatif (Auntentifikasi Gagal)	

Aksi Aktor	Reaksi Sistem
1. masukan *keyword/query*	2. Sistem melakukan pengecekan pada daftar cache
	3. menyatakan *query* tidak ada pada daftar *cache*
	4. perhitungan kemiripan antara *keyword/query* dengan dokumen berita
	5. hasil perhitungan disimpan pada daftar cache
	6. menampilkan berita relevan



Gambar 3.30 *Activity Diagram* Retrieval



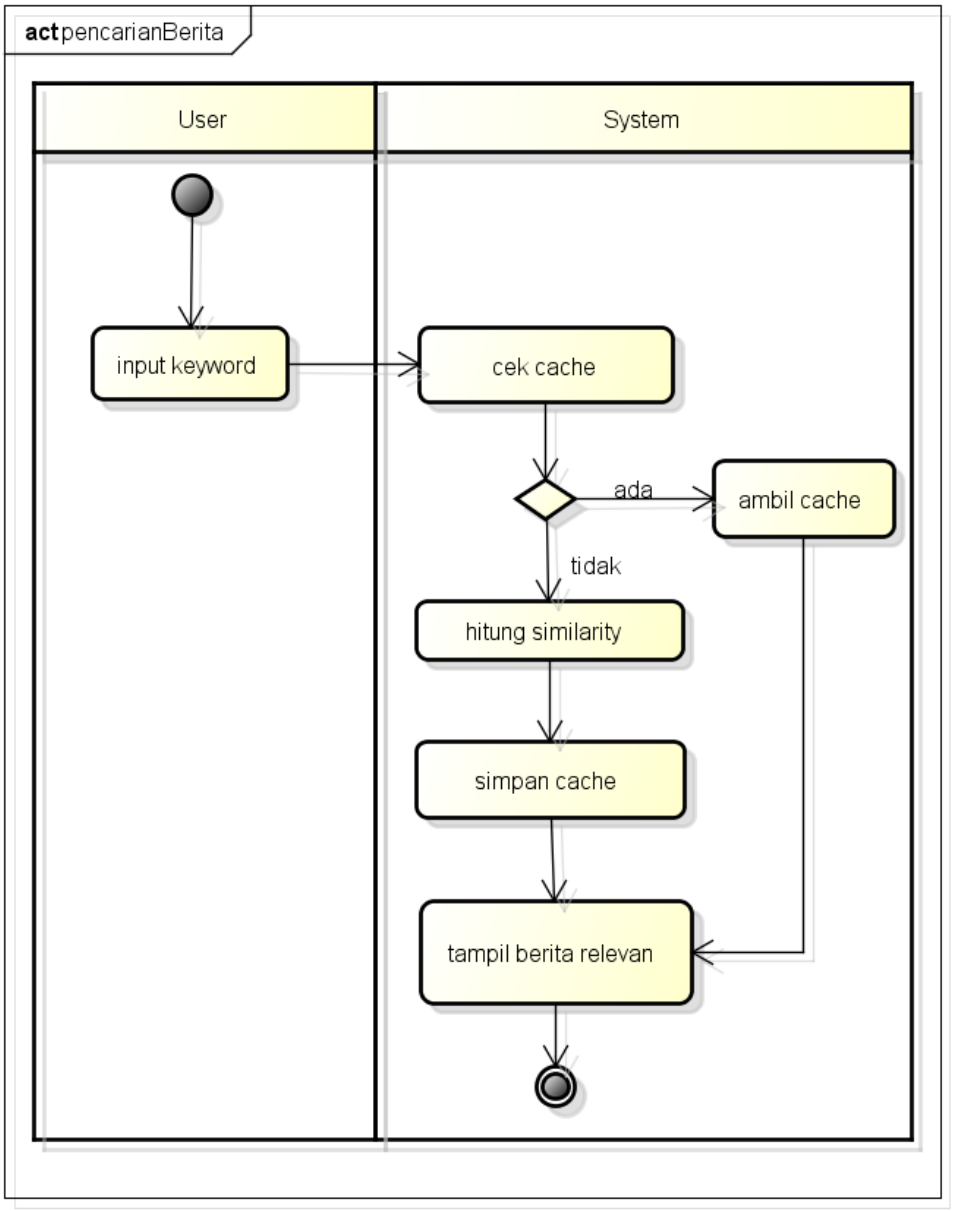
Gambar 3.31*Sequence Diagram Retrieval*

3.1.6.16 *Use Case* Pencarian Berita

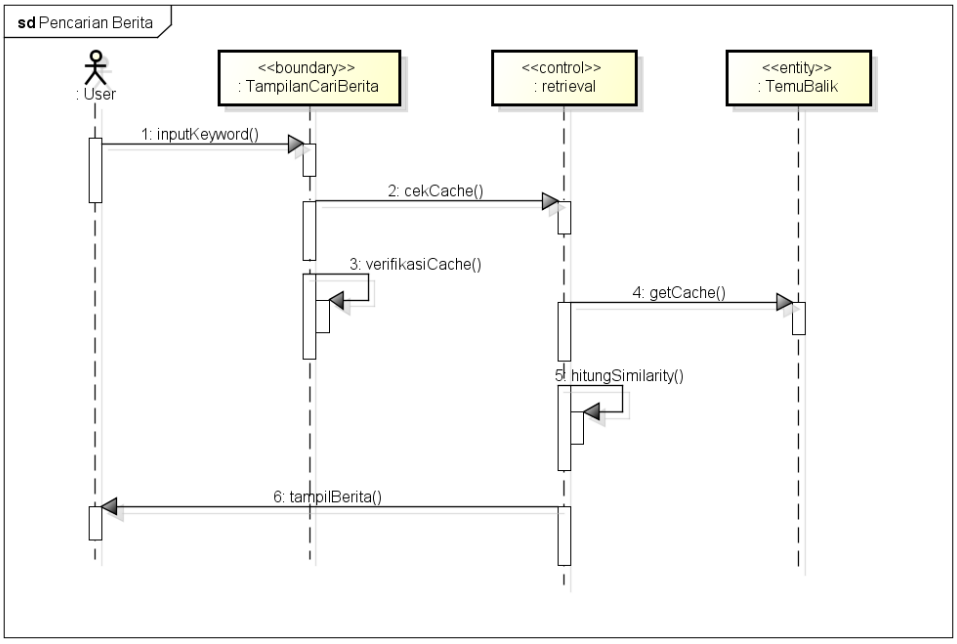
Proses yang digunakan aktor *user* dan *administrator* untuk mencari berita yang relevan. Dalam proses ini dilakukan model ruang vektor untuk perhitungan kemiripan antara *query* dan dokumen. *Use case* pencarian berita diterangkan dan dimodelkan dengan skenario, *activity diagram* dan *sequence diagram.*

Table 3.16 Skenario *Use Case* Pencarian berita

Identifikasi	
Nama	Pencarian Berita
Tujuan	Mencari berita
Deskripsi	Melakukan pencarian berita yang relevan.
Aktor	*Administrator,user*
Skenario Utama	
Kondisi awal	Form pencarian
Aksi aktor	Reaksi sistem
1. masukan *keyword/query*	2. Sistem melakukan pengecekan pada daftar *cache*
	3. menyatakan *query* ada pada daftar *cache*
	4. menampilkan berita relevan
Kondisi akhir	Berita yang relevan dengan *query* ketemu.
Skenario Alternatif (Auntentifikasi Gagal)	
Aksi Aktor	Reaksi Sistem
1. masukan *keyword/query*	2. Sistem melakukan pengecekan pada daftar *cache*
	3. menyatakan *query* tidak ada pada daftar *cache*
	4. perhitungan kemiripan antara *keyword/query* dengan dokumen berita
	5. hasil perhitungan disimpan pada daftar *cache*
	6. menampilkan berita relevan
1. masukan *keyword/query*	2. Tidak ada dokumen berita yang sesuai dengan *keyword.*

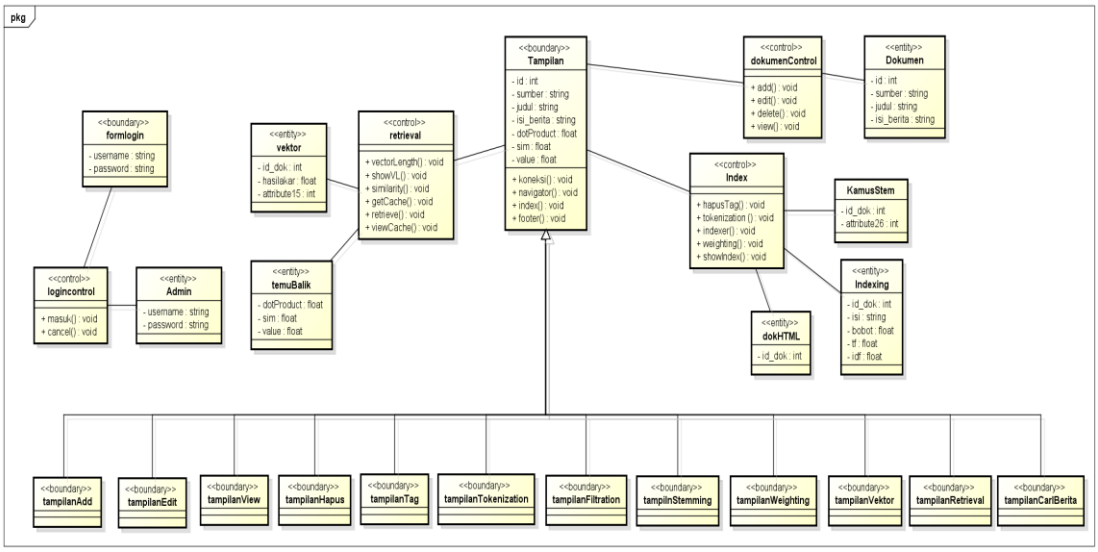


Gambar 3.32 *Activity Diagram* Pencarian Berita



Gambar 3.33 *Sequence Diagram* Cari Berita

3.1.6.17 *Class Diagram*



Gambar 3.34 *Class Diagram*

3.1.7 Analisis dan Kebutuhan Non Fungsional

Analisis dan kebutuhan non fungsional meliputi analisis dan kebutuhan perangkat keras serta analisis dan kebutuhan perangkat lunak.

3.1.7.1 Analisis dan Kebutuhan Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan dalam *Information Retrieval System* ini adalah :

- Sistem Operasi Microsoft Windows 7 Professional
- *Software* wampp sebagai Web Server
- PHP sebagai bahasa pemrograman web
- Mysql sebagai sistem manajemen basis data
- *Software utility Dreamweaver* CS5.

3.1.7.2 Analisis dan Kebutuhan Perangkat Keras

Spesifikasi perangkat keras yang digunakan dalam *Information Retrieval System* ini adalah :

- *Processor* : AMD Turion™ X2 *Dual-Core Mobile* RM-74(2 CPUs) 2.2 GHz.
- RAM : 3 Giga
- VGA : Onboard
- HDD : 320 Giga

- Monitor : 15”

3.2 Perancangan Sistem

Perancangan akan dimulai setelah tahap analisis terhadap sistem selesai dilakukan. Perancangan dapat didefinisikan sebagai proses aplikasi berbagai teknik dan prinsip bagi tujuan pendefinisian suatu perangkat, suatu proses atau sistem dalam detail yang memadai untuk memungkinkan realisasi fisiknya. Perancangan digambarkan sebagai proses multi-langkah dimana representasi struktur data, struktur program, karakteristik *interface*, dan detail prosedur, disintesis dari persyaratan informasi.

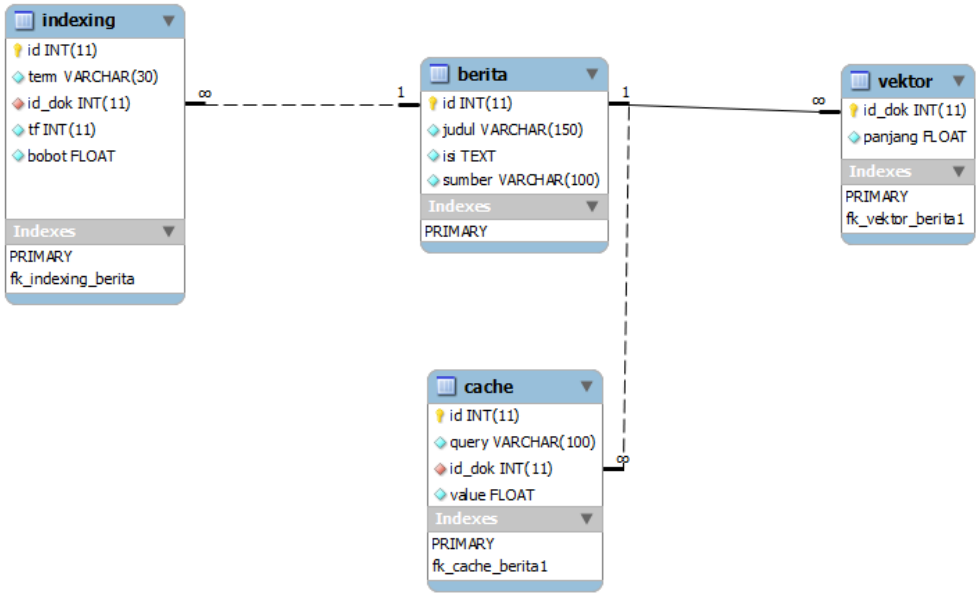
3.2.1 Perancangan Basis Data

Perancangan data mentransformasi model domain informasi yang dibuat selama analisis ke dalam struktur data yang akan diperlukan untuk mengimplementasi perangkat lunak.

3.2.1.1 Skema Relasi

Proses relasi antar file merupakan gabungan antar file yang mempunyai kunci utama yang sama, sehingga file-file tersebut menjadi satu kesatuan yang dihubungkan oleh *field* kunci tersebut. Pada proses ini elemen-elemen data dikelompokkan menjadi

satu file database beserta entitas dan hubungannya. Berikut adalah skema relasi antar tabel yang ada di aplikasi *Information Retrieval System* :



Gambar 3.35 Skema Relasi

3.2.1.2 Struktur Tabel

Tabel adalah sekumpulan data atau informasi spesifik tentang subjek tertentu yang disusun dalam bentuk kolom dan baris. Tabel adalah komponen utama dan pertama dari sebuah *database*. Struktur tabel yang terdapat dalam *database Information Retrieval System* sebagai berikut :

a. Tabel Berita

Tabel Berita digunakan untuk menyimpan dokumen-dokumen berita.

Tabel 3.17 Tabel Berita

Nama field	Tipe	Size	Keterangan
id	Integer	11	*Primary key*
judul	varchar	150	
isi	Text		
sumber	varchar	100	

b. Tabel *Indexing*

Tabel *Indexing* digunakan untuk menyimpan data hasil *preprocessing* dan *weighting* (pembobotan).

Tabel 3.18 Tabel Indexing

Nama field	Tipe	Size	Keterangan
id	Integer	11	*primary key*
term	Varchar	30	
id_dok	Integer	11	*foreign key, reference* berita(id)
tf	Integer	11	
bobot	Float		

c. Tabel *Stem*

Tabel *Stem* digunakan untuk menyimpan kamus atau koleksi *term-term* yang akan diubah kebentuk kata dasarnya.

Tabel 3.19 Tabel *Stem*

Nama field	Tipe	Size	Keterangan
id	Integer	11	primary key
term	Varchar	50	
Stem	Varchar	50	

d. Tabel Vektor

Tabel Vektor digunakan untuk menyimpan hasil perhitungan panjang vektor dari tiap dokumen berita.

Tabel 3.20 Tabel Vektor

Nama field	Tipe	Size	Keterangan
id_dok	Integer	11	foreign key, reference berita(id)
Panjang	Float		

e. Tabel *Cache*

Tabel *Cache* digunakan untuk menyimpan history *query* atau *keyword* dan untuk menyimpan nilai kemiripan antara *query* dan dokumen berita.

Tabel 3.21 Tabel *Cache*

Nama field	Tipe	Size	Keterangan
Id	Integer	11	*primary key*
Query	Varchar	100	
id_dok	Integer	11	*foreign key, reference* berita(id)
Value	Float		

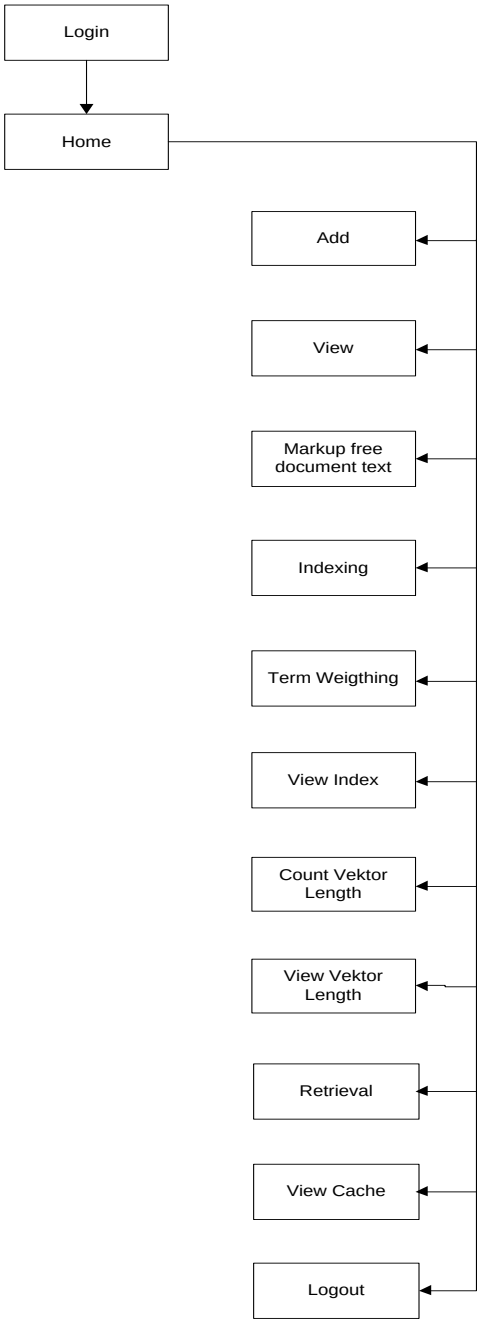
f. Tabel Admin

Tabel Admin digunakan untuk menyimpan data login *administrator*.

Tabel 3.22 Tabel Admin

Nama field	Tipe	Size	Keterangan
username	Varchar	20	*primary key*
password	Varchar	50	

3.2.2 Perancangan Struktur Menu

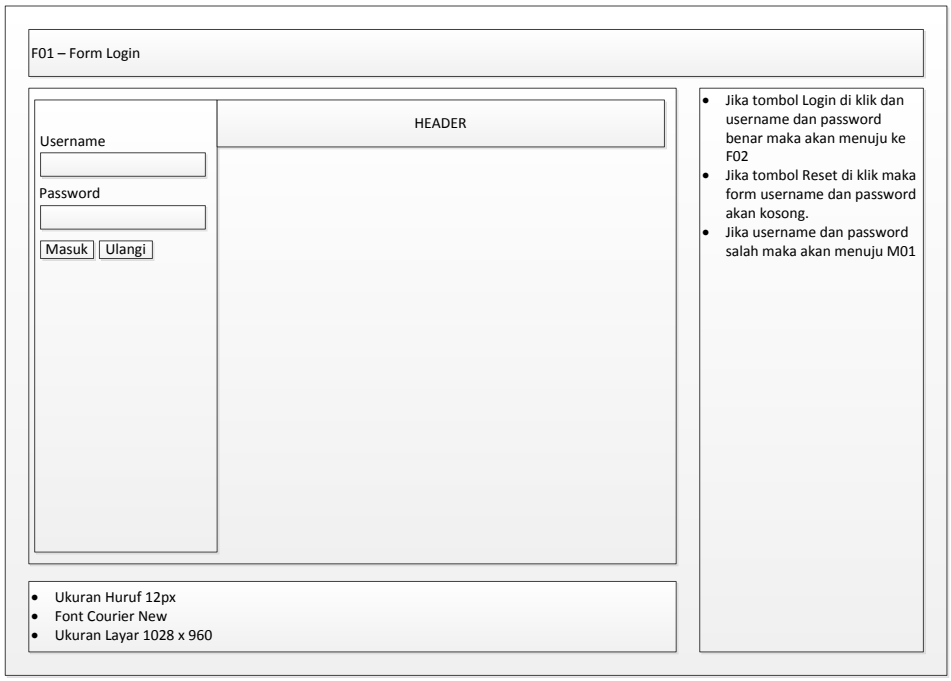


Gambar 3.36 Struktur Menu

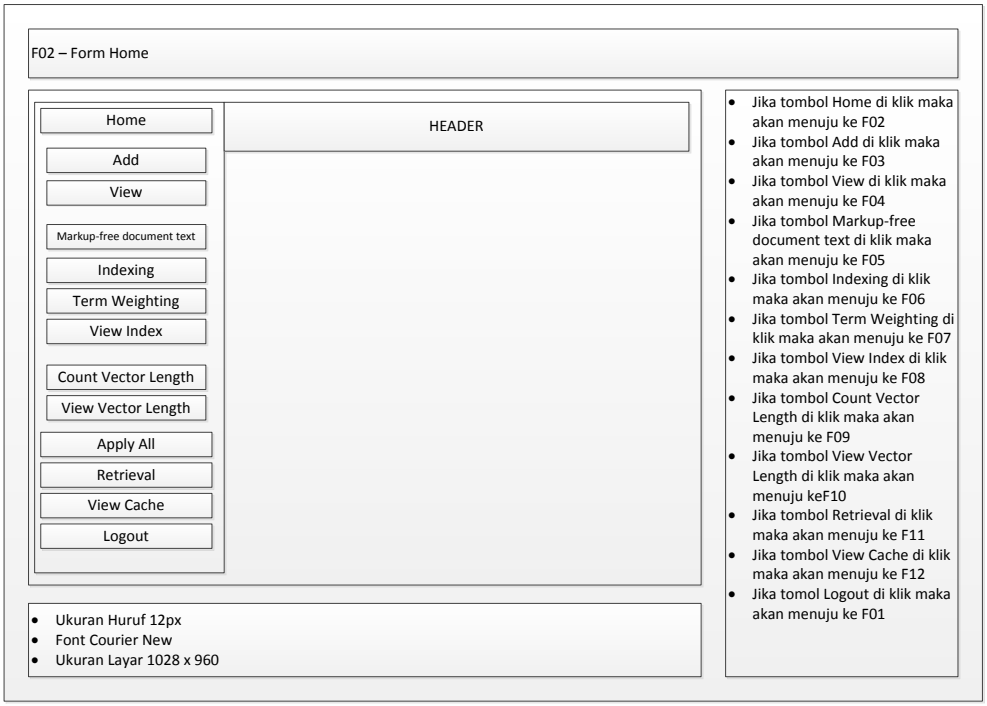
3.2.3 Perancangan Antar Muka

Perancangan antarmuka (*interface*) menggambarkan bagaimana perangkat lunak berkomunikasi dengan dirinya sendiri, dengan sistem yang berinteroperasi dengannnya, dan dengan manusia yang menggunakannya.

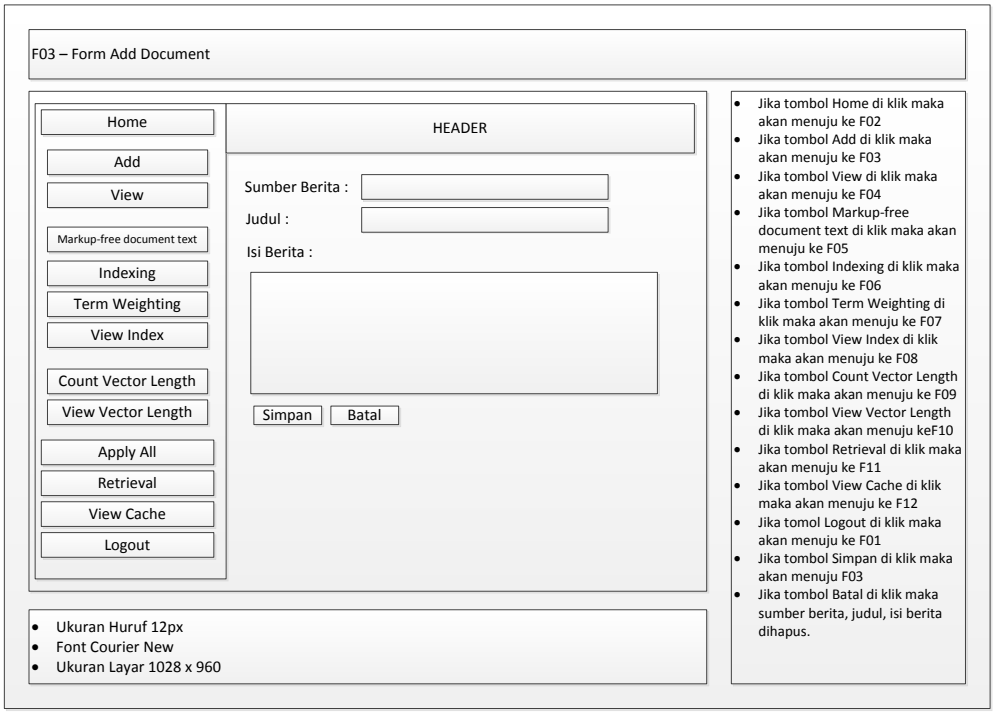
3.2.3.1 Perancangan Tampilan



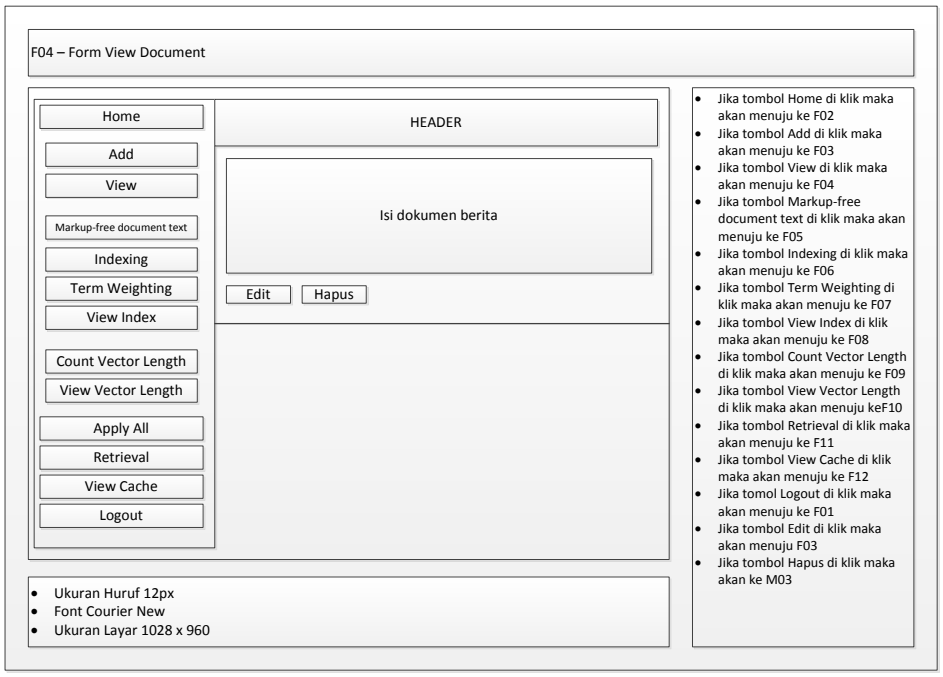
Gambar 3.37 Tampilan Halaman Login



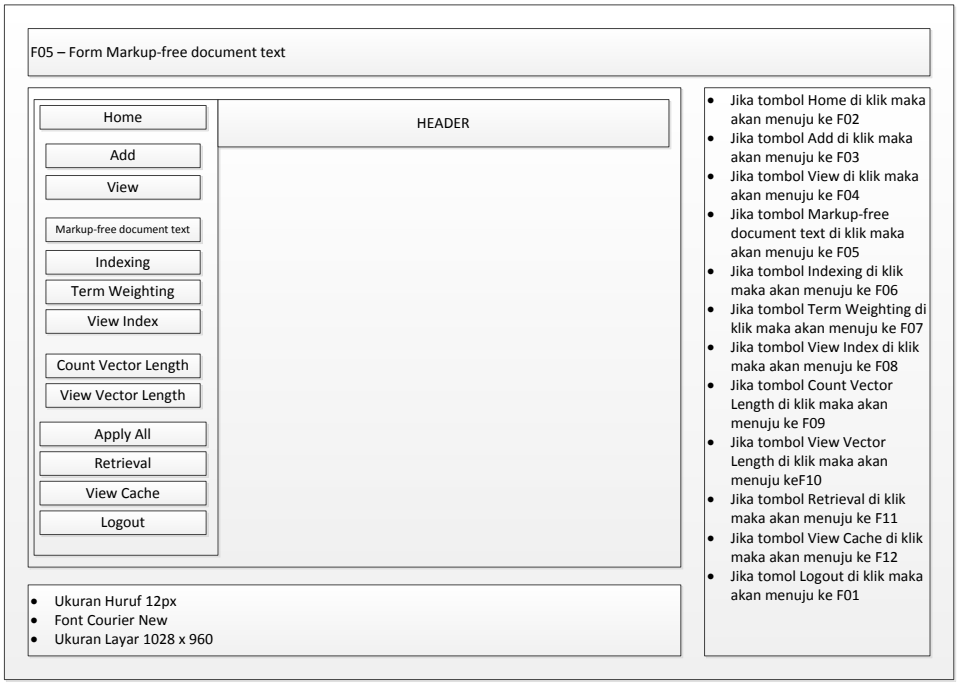
Gambar 3.38 Tampilan Halaman Home



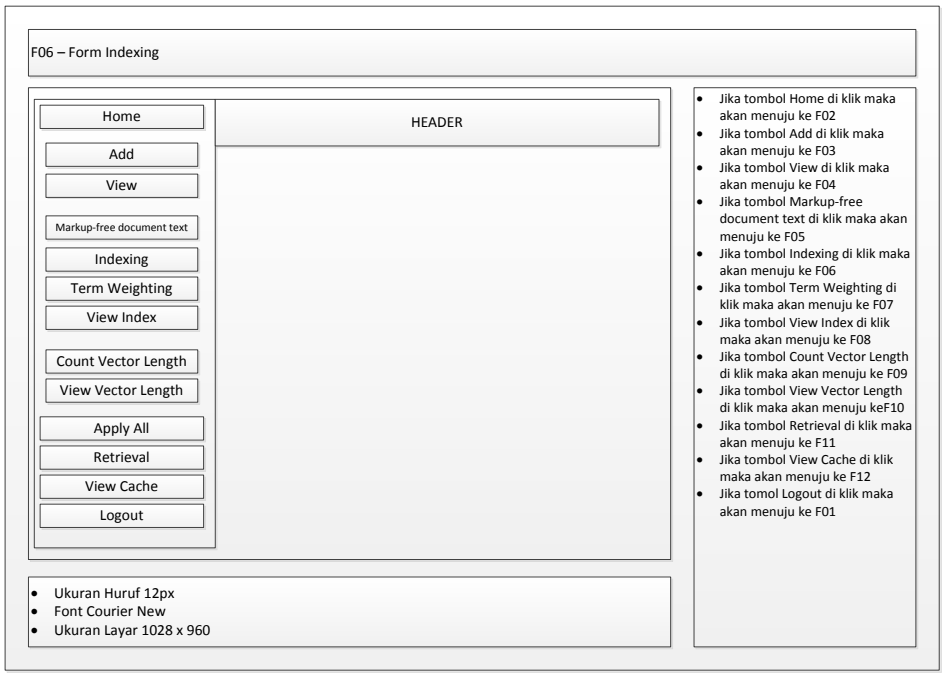
Gambar 3.39 Tampilan Halaman Tambah Dokumen Berita



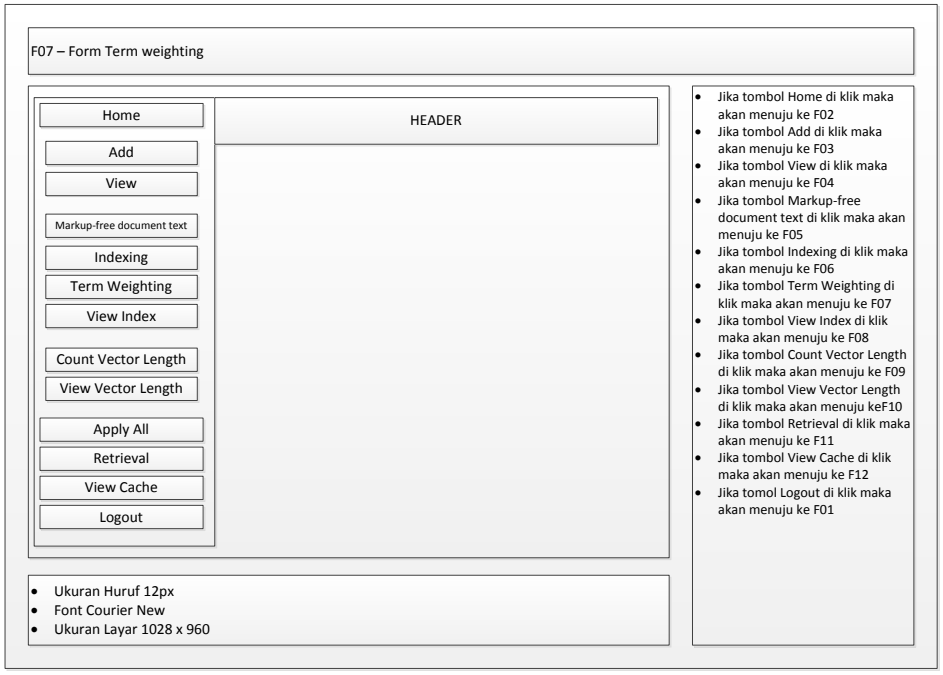
Gambar 3.40 Tampilan Halaman View Document



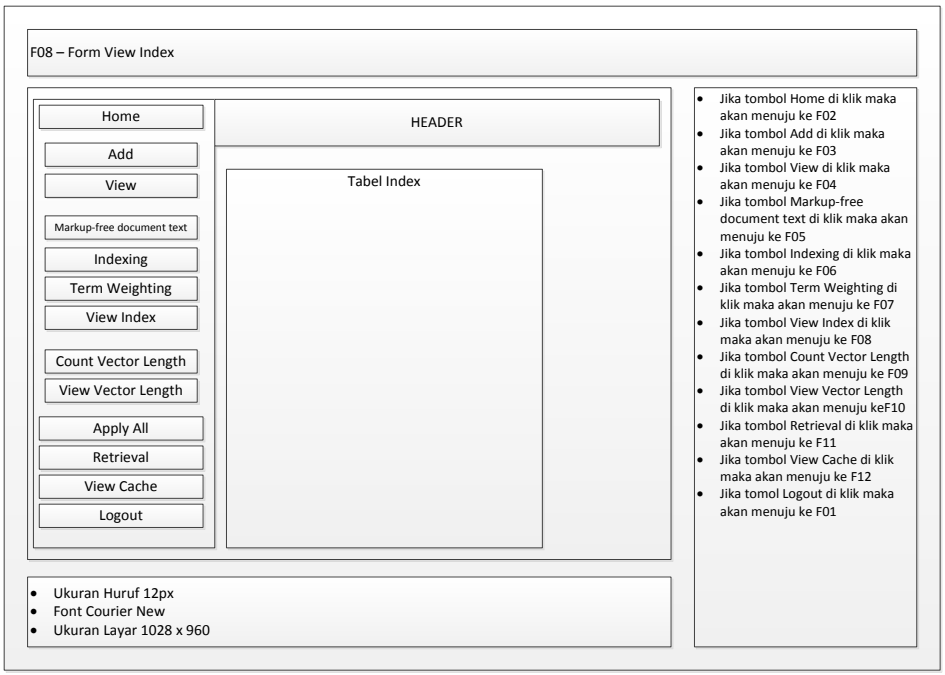
Gambar 3.41 Tampilan Halaman Hapus Tag HTML



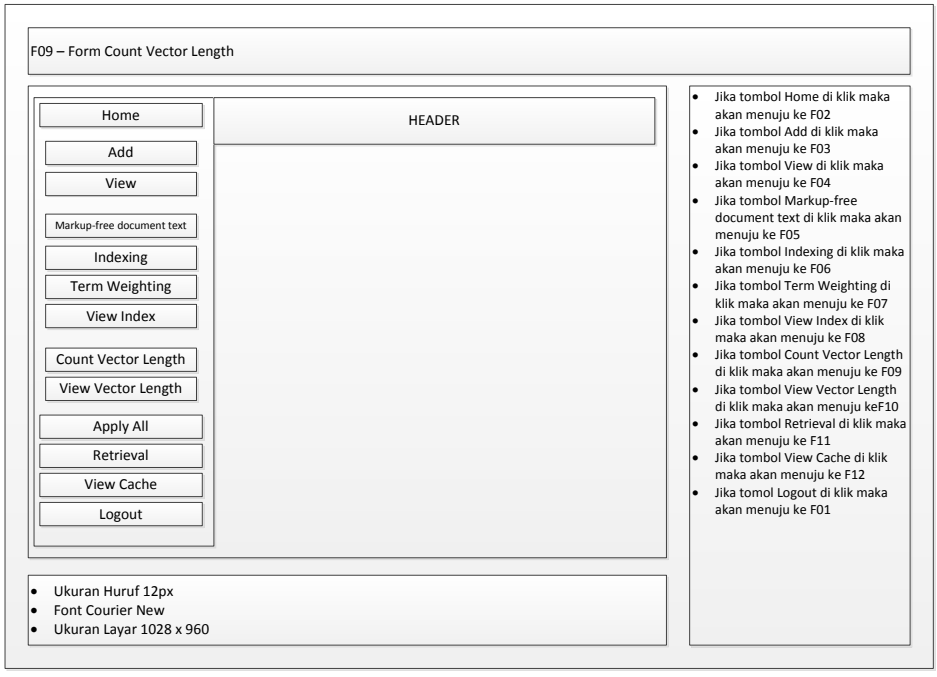
Gambar 3.42 Tampilan Halaman Indexing



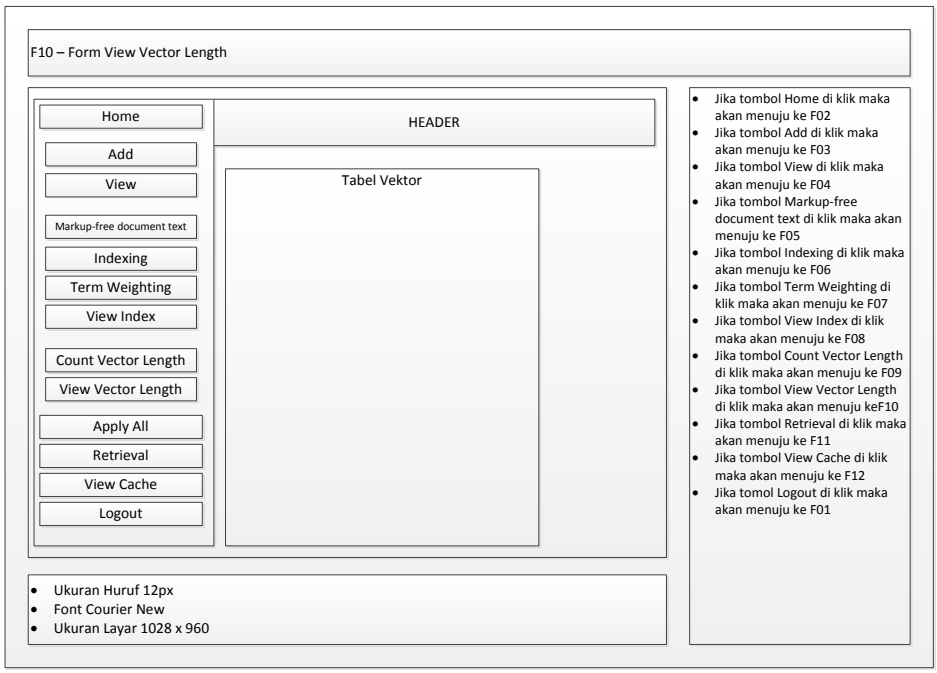
Gambar 3.43 Tampilan Halaman Pembobotan Kata



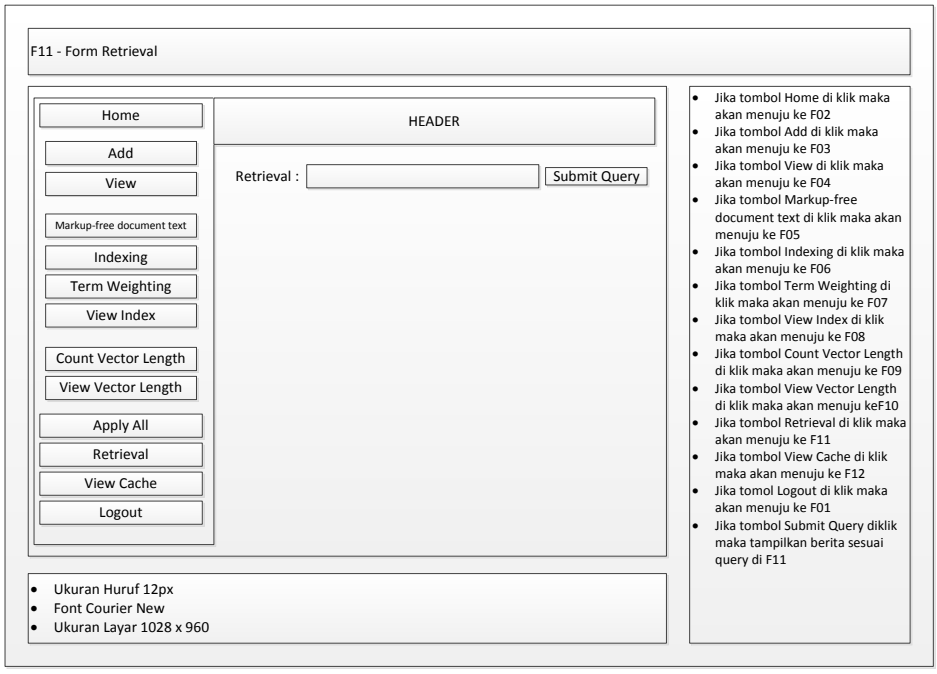
Gambar 3.44 Tampilan Halaman View Index



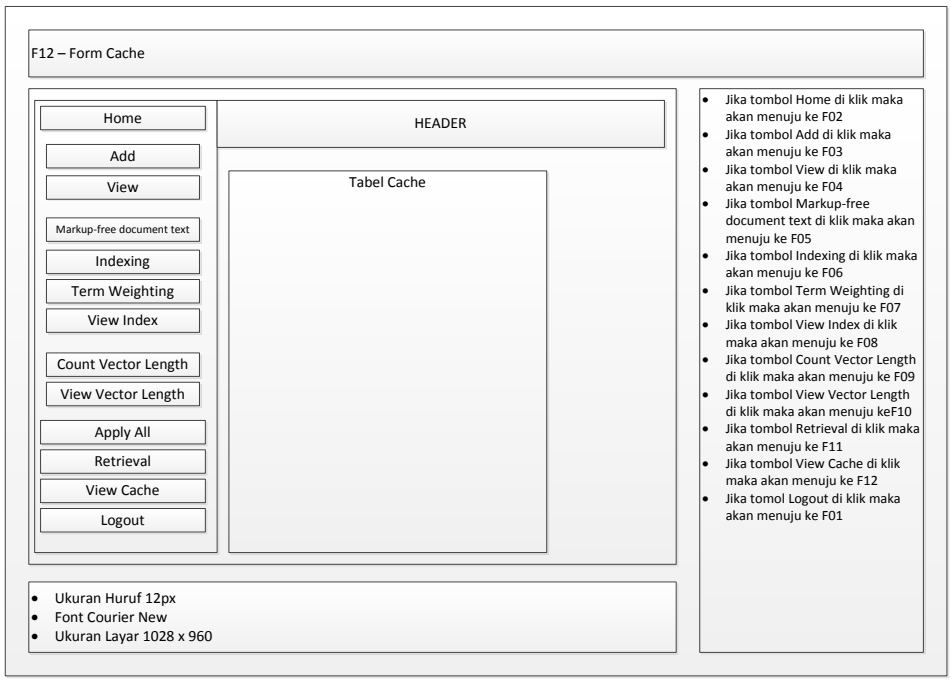
Gambar 3.45 Tampilan Halaman Hitung Panjang Vektor



Gambar 3.46 Tampilan Halaman Lihat Panjang Vektor

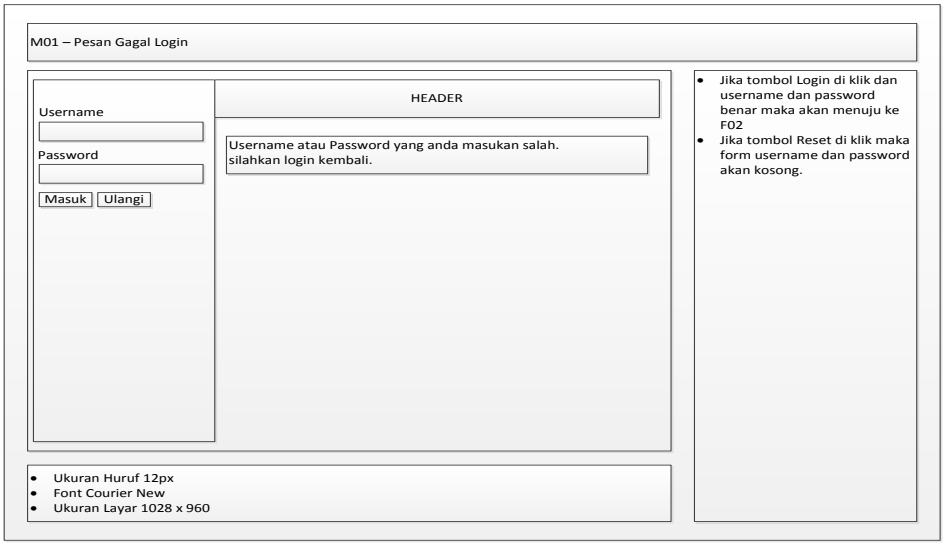


Gambar 3.47 Tampilan Halaman Retrieval

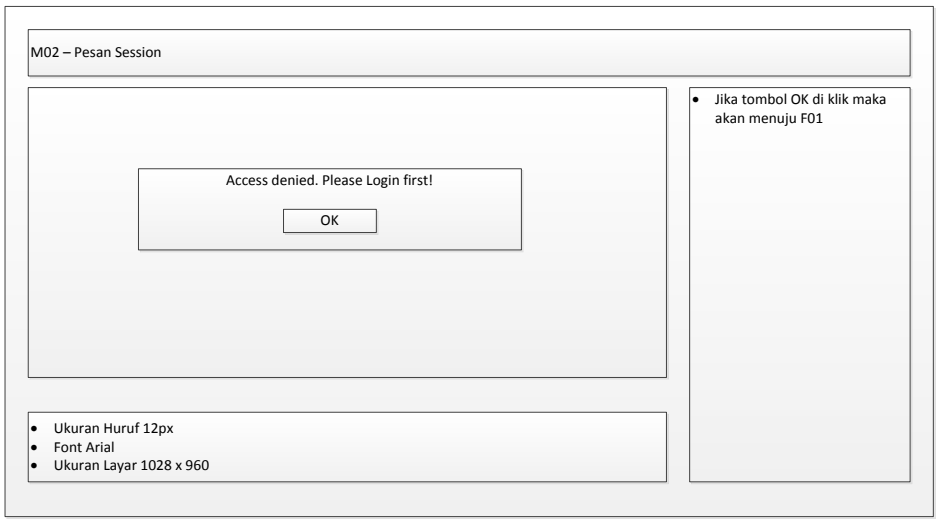


Gambar 3.48 Tampilan Halaman Cache

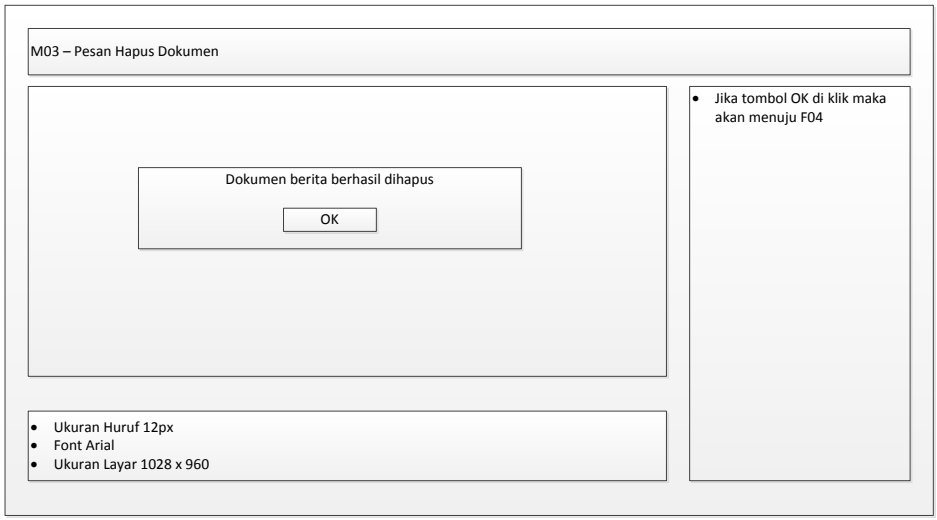
3.2.3.2 Perancangan Pesan



Gambar 3.49 Tampilan Pesan M01



Gambar 3.50 Tampilan Pesan M02



Gambar 3.51 Tampilan Pesan M03