BAB 3
ANALISIS DAN PERANCANGAN

3.1 Analisis Masalah

Analisis masalah adalah suatu gambaran masalah yang diangkat dalam penulisan skripsi tentang Peringkasan Teks dalam dokumen berbahasa Indonesia. Analisis masalah ini menjelaskan proses identifikasi masalah serta evaluasi mengenai sistem peringkasan teks otomatis dalam dokumen berbahasa Indonesia. Berdasarkan uraian pada latar belakang masalah pada penelitian ini terdapat permasalahan untuk melakukan peringkasan teks otomatis menggunakan *Hidden Markov Model* yang nantinya akan menghasilkan sebuah ringkasan dari artikel pada sebuah dokumen berbahasa Indonesia.

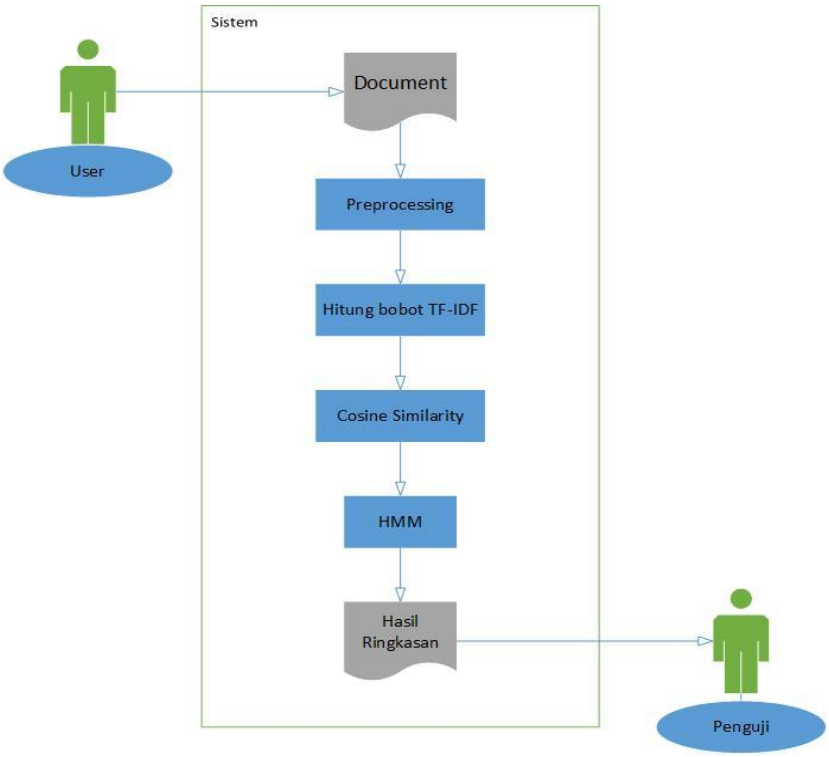
3.2 Analisis Sistem

Analisis sistem didefinisikan sebagai penguraian dari sistem utama kedalam sub-sub sistem dengan tujuan untuk mengidentifikasikan permasalahan-permasalahan yang ada dan kebutuhan-kebutuhan yang diperlukan agar dapat diusulkan dan diciptakan sistem yang lebih baik. Dalam mengimplementasikan *Hidden Markov Model* pada peringkasan teks akan dibangun dengan gambaran sistem yang akan tertera pada Gambar 3.1 .

Analisis sistem peringkasan teks yang dibangun memiliki 4 tahapan yaitu praproses, hitung bobot kata, pentuan ringkasan, dan uji akurasi. Untuk penjelasan setiap tahap pada proses peringkasan sebagai berikut :

1. Tahap pertama yaitu tahap pengumpulan data dokumen berita yang telah dilakukan secara manual dari media *online* seperti kompas, tempo, merdeka, republika, liputan6, dan tribunnews. Kemudian data tersebut dilakukan proses *preprocessing* yang merupakan tahap dimana data yang akan digunakan sebagai data masukan dibersihkan dari *noise* atau dari hal yang tidak mempunyai pengaruh dalam ringkasan. *Preprocessing* yang dilakukan dalam tahap ini antara lain : pemecah kalimat, *case folding, filtering, tokenizing, stopword removal* dan *stemming.*

2. Tahap kedua yaitu melakukan perhitungan bobot menggunakan metode *tf-idf*, dengan menghitung TF(*Term Frequency*), DF(*Document Frequenc*), dan dihitung *weight* (W) dari masing-masing *term.*
3. Tahap ketiga adalah menentukan hasil ringkasan dengan menggunakan *Hidden Markov Model*. Pada metode ini dihitung nilai *cosine similarity* antar kalimat atau kemiripannya pada dokumen masukan. Proses ini dilakukan pada semua kalimat terhadap kalimat yang lainnya yang akan menghasilkan sebuah ringkasan. Hasil ringkasan adalah kumpulan dari kalimat-kalimat hasil persentase yang paling tinggi.



Gambar 3. 1 Gambar Alur Sistem

4. Tahapan ke empat, merupakan kelanjutan dari hasil peringkasan yaitu proses untuk menguji akurasi (*Precession, Recall, F-measure*) hasil ringkasan dari aplikasi dengan hasil ringkasan dari manusia.

3.2.1 Analisis Data Masukan (*Input Data*)

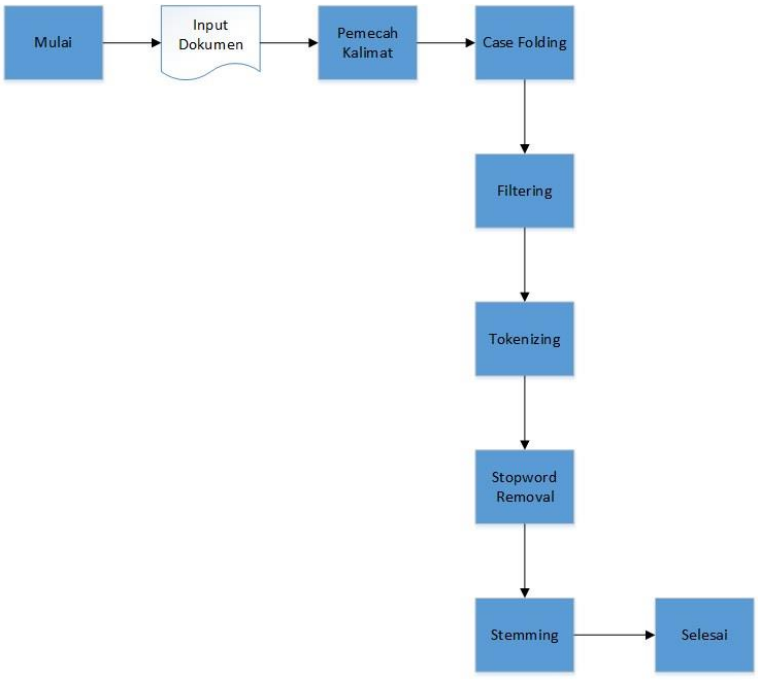
Analisis dapa pada sistem peringksan teks dokumen bahasa Indonesia menjelaskan proses data masukan. Peringkasan teks otomatis berita yang dibuat

merupakan sistem peringkasan dengan input *data trining* sekaligus *testing* berupa *single* dokumen untuk menghasilkan ringkasan (*summary*). Gambaran sistem menunjukkan skema proses peringkasan secara umum.

Berikut alur proses data masukan pada sistem peringkasan teks :

1. Teks dokumen yang dimasukkan berupa teks dengan format (file.txt) oleh *user* yang akan diringkas, data yang masuk tersebut adalah *data training* dan *testing* dokumen berita dalam berbahasa Indonesia. Dokumen yang akan diproses sesuai Tabel 3.1.
2. Sistem akan melakukan *text preprocessing* dokumen yang terdiri dari tahap pemecahan kalimat, *case folding, filtering, tokenizing, stop word removal* dan *stemming* yang akan dibahas pada analisis berikutnya.

Berikut *flowchart* dari proses *input data* yang dapat dilihat pada gambar 3.3:



Gambar 3. 2 *Flowchart* Data Input

Berikut disajikan dokumen artikel *input* yang akan diringkas, dapat dilihat pada tabel 3.1.

Tabel 3. 1 Data *Testing*

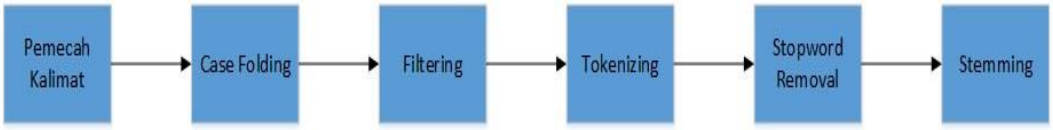
Artikel Berita
Sekjen FPI DKI Jakarta Novel Bamukmin menuding Calon Gubernur DKI Jakarta nomor urut dua Basuki Tjahaja Purnama alias Ahok telah membunuh dua anggotanya, dalam persidangan kasus dugaan penistaan agama kemarin, Selasa (3/1). Novel yang menjadi saksi pelapor dari Jaksa Penuntut Umum (JPU) ini menganggap Ahok harus bertanggung jawab akibat demo ricuh FPI dengan polisi medio Oktober 2014 silam di depan Gedung DPRD DKI Jakarta. "Kamu telah membunuh dua anak buah saya," kata Ahok menirukan Novel di Rumah Pemenangan Lembang, Menteng, Jakarta Pusat, Kamis (5/1). Ahok mengaku enggan ambil pusing atas tudingan Novel tersebut. Dia malah ingin bergurau agar Novel tambah emosi, namun Ahok recana tersebut tidak jadi dilakukan untuk menghormati majelis hakim. "Tadinya saya mau bilang, kan kata dia (Novel) dua, saya mau tawarkan dua apa satu, dua atau satu, atau dua atau tiga. Kalau sampai dia (Novel) bilang dua, top," tutupnya.

Data masukan yang ada pada table 3.1 didapat dari artikel berita yang dimuat oleh situs www.merdeka.com dengan judul "**Ini Tanggapan Ahok Dituding Novel Bunuh Dua Anggota FPI**" yang diakses pada tgl 05 Jan 2017 pada pukul 22.30 WIB dan berikut alamat *website*-nya:

https://www.merdeka.com/jakarta/ini-tanggap-ahok-dituding-novel-bunuh-dua-anggota-fpi.html

3.2.2 Analisis *Preprocessing*

Analisis *preprocessing* menjelaskan praproses yang dilakukan untuk mendapatkan fitur kata-kata yang bersih sehingga tidak akan mengangung *noise* yang akan menggunakan hasil untuk proses selanjutnya. Berikut tahapannya dapat dilihat pada gambar 3.3.



Gambar 3. 3 Tahapan Preprocessing

1. Pemecahan Kalimat

   Pada tahap pemecahan kalimat adalah memecah string dokumen menjadi kumpulan kalimat-kalimat berdasarkan tanda titik "." sebagai pemisah (*delimiter*) untuk memotong *string* dokumen. Tabel 3.2 adalah hasil proses pemecahan dokumen menjadi kumpulan kalimat-kalimat sebagai berikut :

Tabel 3. 2 Pemecahan Kalimat

No.	Kalimat
S1	Sekjen FPI DKI Jakarta Novel Bamukmin menuding Calon Gubernur DKI Jakarta nomor urut dua Basuki Tjahaja Purnama alias Ahok telah membunuh dua anggotanya, dalam persidangan kasus dugaan penistaan agama kemarin, Selasa (3/1).
S2	Novel yang menjadi saksi pelapor dari Jaksa Penuntut Umum (JPU) ini menganggap Ahok harus bertanggung jawab akibat demo ricuh FPI dengan polisi medio Oktober 2014 silam di depan Gedung DPRD DKI Jakarta.
S3	"Kamu telah membunuh dua anak buah saya," kata Ahok menirukan Novel di Rumah Pemenangan Lembang, Menteng, Jakarta Pusat, Kamis (5/1).

No.	Kalimat
S4	Ahok mengaku enggan ambil pusing atas tudingan Novel tersebut.
S5	Dia malah ingin bergurau agar Novel tambah emosi, namun Ahok recana tersebut tidak jadi dilakukan untuk menghormati majelis hakim.
S6	"Tadinya saya mau bilang, kan kata dia (Novel) dua, saya mau tawarkan dua apa satu, dua atau satu, atau dua atau tiga.
S7	Kalau sampai dia (Novel) bilang dua, top," tutupnya.

2. *Case Folding*

Pada tahap ini, kalimat yang telah dipecah akan dijadikan satu kesatuan bentuk yaitu merubah seluruh karakter kebentuk huruf kecil. Hasil proses *case folding* dapat dilihat pada tabel 3.3 sebagai berikut.

Tabel 3. 3 Case Folding

No.	Kalimat
S1	sekjen fpi dki jakarta novel bamukmin menuding calon gubernur dki jakarta nomor urut dua basuki sjahaja purnama alias ahok telah membunuh dua anggotanya, dalam persidangan kasus dugaan penistaan agama kemarin, selasa (3/1).
S2	novel yang menjadi saksi pelapor dari jaksa penuntut umum (jpu) ini menganggap ahok harus bertanggung jawab akibat demo ricuh fpi dengan polisi pada oktober 2014 silam di depan gedung dprd dki Jakarta.
S3	“kamu telah membunuh dua anak buah saya,” kata ahok menirukan novel di rumah pemenangan lembang, menteng Jakarta pusat, kamis (5/1).
S4	ahok mengaku enggan ambil pusing atas tudingan novel tersebut.
S5	dia malah ingin bergurau agar novel tambah emosi, namun ahok rencana tersebut tidak jadi dilakukan untuk menghormati majelis hakim.

No.	Kalimat
S6	“tadinya saya mau bilang, kan kata dia (novel) dua, saya mau tawarkan dua apa satu, dua atau satu, atau dua atau tiga.
S7	kalua sampai dia (novel) bilang dua, top,” tutupnya.

3. *Filtering*

Setelah tahap *case folding* dilakukan kemudian berlanjut ketahap *filitering*, yaitu proses menghilangkan karakter selain karakter huruf “a” sampai “z” dan spasi. Hasil proses *filtering* dapat dilihat pada tabel 3.4 sebagai berikut.

Tabel 3. 4 Filtering

No.	Kalimat
S1	sekjen fpi dki jakarta novel bamukmin menuding calon gubernur dki jakarta nomor urut dua basuki sjahaja purnama alias ahok telah membunuh dua anggotanya dalam persidangan kasus dugaan penistaan agama kemarin selasa
S2	novel yang menjadi saksi pelapor dari jaksa penuntut umum jpu ini menganggap ahok harus bertanggung jawab akibat demo ricuh fpi dengan polisi pada oktober silam di depan gedung dprd dki Jakarta
S3	kamu telah membunuh dua anak buah saya kata ahok menirukan novel di rumah pemenangan lembang menteng jakarta pusat kamis
S4	ahok mengaku enggan ambil pusing atas tudingan novel tersebut
S5	dia malah ingin bergurau agar novel tambah emosi namun ahok rencana tersebut tidak jadi dilakukan untuk menghormati majelis hakim
S6	tadinya saya mau bilang kan kata dia novel dua saya mau tawarkan dua apa satu dua atau satu atau dua atau tiga
S7	kalua sampai dia novel bilang dua top tutupnya

4. *Tokenizing*

Proses pemisahan kata dalam kalimat sehingga terbentuk sebuah kata tunggal yang diambil pada setiap kalimat, menghilangkan pemisah (*delimiter*) yang menyusunnya berupa kalimat spasi. Berikut hasil dari *tokenizing* terdapat pada tabel 3.5.

Tabel 3. 5 *Tokenizing*

S1	S2	S3	S4	S5	S6	S7
sekjen	novel	kamu	ahok	dia	tadinya	kalau
fpi	yanng	telah	mengaku	malah	saya	sampai
dki	mejadi	membunuh	enggan	ingin	mau	dia
jakarta	saksi	dua	ambil	bergurau	bilang	novel
novel	pelapor	anak	pusing	agar	kan	bilang
bamukmin	dari	buah	atas	novel	kata	dua
menuding	jaksa	saya	tudingan	tambah	dia	top
calon	penuntut	kata	novel	emosi	novel	tutupnya
gubernur	umum	ahok	tersebut	namun	dua	
dki	jpu	menirukan		ahok	saya	
jakarta	ini	noval		rencana	mau	
nomor	menganggap	di		tersebut	tawarkan	
urut	ahok	rumah		tidak	dua	
dua	harus	pemenangan		jadi	apa	
basuki	bertanggung	lembang		di	satu	
sjahaja	jawab	menteng		lakukan	dua	
purnama	akibat	jakarta		untuk	atau	
alias	demo	pusat		menghormati	satu	
ahok	ricuh	kamis		majelis	atau	
telah	fpi			hakim	dua	
membunuh	dengan				atau	
dau	polisi				tiga	
anggotanya	pada					
dalam	oktober					
persidangan	silam					
kasus	di					
dugaan	depan					
penistaan	gedung					
agama	dprd					
kemarin	dki					

S1	S2	S3	S4	S5	S6	S7
selasa	jakarta					

5. *Stopword Removal*

Yaitu proses untuk menghapus kata-kata yang tidak penting. Hasil dari *Tokenizing* kemudian dilakukan proses *stop word removal.* Berikut adalah hasil dari proses *stop word removal* pada tabel 3.6.

Tabel 3. 6 *Stopword Removal*

Hasil *Stopword Removal*			
agama	dugaan	menuding	tanggapan
ahok	emosi	nomor	tawarkan
akibat	enggan	novel	tjahaja
alias	fpi	oktober	top
ambil	gedung	pelapor	tudingan
anak	gubernur	pemenangan	urut
anggota	hakim	penistaan	
anggotanya	jakarta	penuntut	
bamukmin	jaksa	persidangan	
basuki	jpu	polisi	
bergurau	kamis	purnama	
bertanggung	kemarin	pusat	
bilang	lembang	pusing	
buah	majelis	recana	
bunuh	membunuh	ricuh	
calon	mengaku	rumah	
demo	menganggap	saksi	
dituding	menghormati	sekjen	
dki	menirukan	selasa	
dprd	menteng	silam	

6. *Stemming*

Proses ini untuk menentukan kata dasar yang belum diproses melalui *stop word* dan tidak terdapat pada *database.* Hasil dari *stop word removal* kemudian dilakukan proses *stemming.* Berikut hasil proses *stemming* dapat dilihat pada tabel 3.7.

Tabel 3. 7 *Stemming*

Hasil *Stemming*			
agama	enggan	nista	tuding
ahok	fpi	nomor	tuntut
akibat	gedung	novel	urut
alias	gubernur	oktober	
ambil	gurau	polisi	
anak	hakim	purnama	
anggap	hormat	pusat	
anggota	jakarta	pusing	
bamukmin	jaksa	recana	
basuki	jpu	ricuh	
bilang	kamis	rumah	
buah	kemarin	saksi	
bunuh	---	sekjen	
---	lapor	sidang	
calon	---	silam	
demo	lembang	tanggap	
dki	majelis	tanggung	
dprd	menang	tawar	
duga	mengaku	tjahaja	
emosi	menteng	top	

Berikut adalah kumpulan kalimat hasil *stemming* yang kaan dihitung bobot tiap kata terhadap setiap kalimatnya yang akan diproses pada algoritma *tf-idf* dapat dilihat pada tabel 3.8.

Tabel 3. 8 Kalimat Hasil Stemming

No.	Kalimat
S1	sekjen fpi dki Jakarta novel bamukmin tuding calon gubernur dki jakarta nomor urut basuki tjahaja purnama alias ahok bunuh anggota sidang duga nista agama kemarin selasa
S2	novel saksi lapor jaksa tuntut jpu anggap ahok tanggung akibat demo ricuh fpi polisi oktober silam gedung dprd dki jakarta
S3	bunuh anak buah ahok novel rumah menang lembang menteng jakarta pusat kamis
S4	ahok mengaku enggan ambil pusing tuding novel
S5	gurau novel emosi ahok rencana hormat majelis hakim
S6	bilang novel tawar
S7	novel bilang top

3.2.3 Extraction Features

Extraction features merupakan proses pengolahan kalimat menggunakan beberapa fitur sehingga menghasilkan nilai-nilai yang nantinya akan digunakan untuk meringkas dokumen. *Extraction features* akan dilakukan dari data hasil *stemming. Extraction features* yang digunakan meliputi kemunculan kata, bobot kata, perkalian bobot judul dengan bobot kalimat, panjang kalimat, dan *cosine similarity.*

a. Kemunculan Kata

Merupakan kumunculan sebuah kata yang terdapat didalam sebuah dokumen (kalimat). Jika kata yang dicari ada pada dokumen (kalimat), maka akan diberikan nilai 1 jika tidak akan diberikan nilai 0. Berikut ditampilkan frekuensi kemunculan kata pada seluruh kalimat.

Tabel 3. 9 Kemunculan Kata Pada Kalimat

KATA	tf							df
	S1	S2	S3	S4	S5	S6	S7	
agama	1	0	0	0	0	0	0	1
ahok	1	1	1	1	1	0	0	6
akibat	0	1	0	0	0	0	0	1
alias	1	0	0	0	0	0	0	1
ambil	0	0	0	1	0	0	0	1
anak	0	0	1	0	0	0	0	1
anggap	0	1	0	0	0	0	0	1
anggota	1	0	0	0	0	0	0	2
bamukmin	1	0	0	0	0	0	0	1
basuki	1	0	0	0	0	0	0	1
bilang	0	0	0	0	0	1	1	2
buah	0	0	1	0	0	0	0	1
bunuh	1	0	0	1	0	0	0	3
calon	1	0	0	0	0	0	0	1
demo	0	1	0	0	0	0	0	1
dki	2	1	0	0	0	0	0	3
dprd	0	1	0	0	0	0	0	1
duga	1	0	0	0	0	0	0	1
emosi	0	0	0	0	1	0	0	1
enggan	0	0	0	1	0	0	0	1
fpi	1	1	0	0	0	0	0	3
gedung	0	1	0	0	0	0	0	1
gubernur	1	0	0	0	0	0	0	1
gurau	0	0	0	0	1	0	0	1
hakim	0	0	0	0	1	0	0	1
hormat	0	0	0	0	1	0	0	1
jakarta	2	1	1	0	0	0	0	4
jaksa	0	1	0	0	0	0	0	1
jpu	0	1	0	0	0	0	0	1

KATA	tf							df
	S1	S2	S3	S4	S5	S6	S7	
kamis	0	0	1	0	0	0	0	1
kemarin	1	0	0	0	0	0	0	1
lapor	0	1	0	0	0	0	0	1
lembang	0	0	1	0	0	0	0	1
majelis	0	0	0	0	1	0	0	1
menang	0	0	1	0	0	0	0	1
mengaku	0	0	0	1	0	0	0	1
menteng	0	0	1	0	0	0	0	1
nista	1	0	0	0	0	0	0	1
nomor	1	0	0	0	0	0	0	1
novel	1	1	1	1	1	1	1	8
oktober	0	1	0	0	0	0	0	1
polisi	0	1	0	0	0	0	0	1
purnama	1	0	0	0	0	0	0	1
pusat	0	0	1	0	0	0	0	1
pusing	0	0	0	1	0	0	0	1
recana	0	0	0	0	1	0	0	1
ricuh	0	1	0	0	0	0	0	1
rumah	0	0	1	0	0	0	0	1
saksi	0	1	0	0	0	0	0	1
sekjen	1	0	0	0	0	0	0	1
sidang	1	0	0	0	0	0	0	1
silam	0	1	0	0	0	0	0	1
tanggap	0	0	0	0	0	0	0	1
tanggung	0	1	0	0	0	0	0	1
tawar	0	0	0	0	0	1	0	1
tjahaja	1	0	0	0	0	0	0	1
top	0	0	0	0	0	0	1	1
tuding	1	0	0	1	0	0	0	3
tuntut	0	1	0	0	0	0	0	1
urut	1	0	0	0	0	0	0	1

Keterangan :

tf = banyaknya kata yang dicari pada sebuah dokumen

df = jumlah banyak kata yang muncul dari setiap dokumen (kalimat)

Q = judul teks

S_{1-n} = kalimat ke-1 sampai ke-n

b. Bobot Kata

Pemberian bobot pada masing-masing kata didefinisikan melalui perhitungan frekuens kemunculan dokumen yang mengandung sebuah kata (*term)* yaitu DF (*Document Frequency*), penghitungan frekuensi kemunculan *term* didalam dokumen yaitu TF (*Term Frequency*) dan perhitungan jumlah dokumen yang mengandung sebuah *term* yang dicari dari kumpulan dokumen yang ada yaitu IDF (*Inverse Document Frequency*). Sebelum dapat menghitung bobot, terlebih dahulu dicari D/df yang nilainya diambil pada tabel 3.9 diatas. Dicontohkan pada kata "agama" sebagai berikut:

D/df = 8/1 = 8

Ket :

D = jumlah kalimat yang terbentuk dari hasil *preprocessing* pemecahan kalimat
df = jumlah banyak kata yang muncul dari setiap dokumen (kalimat)
perhitungan akan terus dilakukan hingga pada kata terakhir.

Kemudian mencari nilai IDF log (D/df) yang dicontohkan pada kata agama sebagai berikut :
IDF log(D/df) = log 8 = 0.903089987
Ket :
IDF = jumlah dokumen yang mengandung sebuah *term* yang dicari dari kumpulan dokumen yang ada.
Perhitungan akan terus dilakukan hingga pada kata terakhir.

Setelah kedua nilai tersebut diperoleh, barulah dapat mencari nilai bobot kata pada setiap kalimat. Dicontohkan pada kata “agama” untuk kalimat judul (Q) seperti dibawah ini.

(D/df) * (IDF log(D/df)) = 8 * 0.903089987 = 0

Hasil perhitungan bobot pada *term* tiap dokumen dapat dilihat pada tabel 3.10.

Tabel 3. 10 Nilai Bobot Kata Pada Kalimat

KATA	D/df	IDF log (D/df)	W (bobot)							
			S1	S2	S3	S4	S5	S6	S7	
agama	8	0.903089987	0.90309	0	0	0	0	0	0	
ahok	1.3333	0.124938737	0.12494	0.12494	0.12494	0.12494	0.12494	0	0	
akibat	8	0.903089987	0	0.90309	0	0	0	0	0	
alias	8	0.903089987	0.90309	0	0	0	0	0	0	
ambil	8	0.903089987	0	0	0	0.90309	0	0	0	
anak	8	0.903089987	0	0	0.90309	0	0	0	0	
anggap	8	0.903089987	0	0.90309	0	0	0	0	0	
anggota	4	0.602059991	0.60206	0	0	0	0	0	0	
bamukmin	8	0.903089987	0.90309	0	0	0	0	0	0	
basuki	8	0.903089987	0.90309	0	0	0	0	0	0	

KATA	D/df	IDF log (D/df)	W (bobot)							
			S1	S2	S3	S4	S5	S6	S7	
bilang	4	0.602059991	0	0	0	0	0	0.60206	0.60206	
buah	8	0.903089987	0	0	0.90309	0	0	0	0	
bunuh	2.6667	0.425968732	0.42597	0	0	0.42597	0	0	0	
calon	8	0.903089987	0.90309	0	0	0	0	0	0	
demo	8	0.903089987	0	0.90309	0	0	0	0	0	
dki	2.6667	0.425968732	0.85194	0.42597	0	0	0	0	0	
dprd	8	0.903089987	0	0.90309	0	0	0	0	0	
duga	8	0.903089987	0.90309	0	0	0	0	0	0	
emosi	8	0.903089987	0	0	0	0	0.90309	0	0	
enggan	8	0.903089987	0	0	0	0.90309	0	0	0	
fpi	2.6667	0.425968732	0.42597	0.42597	0	0	0	0	0	
gedung	8	0.903089987	0	0.90309	0	0	0	0	0	
gubernur	8	0.903089987	0.90309	0	0	0	0	0	0	
gurau	8	0.903089987	0	0	0	0	0.90309	0	0	

KATA	D/df	IDF log (D/df)	W (bobot)							
			S1	S2	S3	S4	S5	S6	S7	
hakim	8	0.903089987	0	0	0	0	0.90309	0	0	
hormat	8	0.903089987	0	0	0	0	0.90309	0	0	
jakarta	2	0.301029996	0.60206	0.30103	0.30103	0	0	0	0	
jaksa	8	0.903089987	0	0.90309	0	0	0	0	0	
jpu	8	0.903089987	0	0.90309	0	0	0	0	0	
kamis	8	0.903089987	0	0	0.90309	0	0	0	0	
kemarin	8	0.903089987	0.90309	0	0	0	0	0	0	
lapor	8	0.903089987	0	0.90309	0	0	0	0	0	
lembang	8	0.903089987	0	0	0.90309	0	0	0	0	
majelis	8	0.903089987	0	0	0	0	0.90309	0	0	
menang	8	0.903089987	0	0	0.90309	0	0	0	0	
mengaku	8	0.903089987	0	0	0	0.90309	0	0	0	
menteng	8	0.903089987	0	0	0.90309	0	0	0	0	
nista	8	0.903089987	0.90309	0	0	0	0	0	0	

KATA	D/df	IDF log (D/df)	W (bobot)							
			S1	S2	S3	S4	S5	S6	S7	
nomor	8	0.903089987	0.90309	0	0	0	0	0	0	
novel	1	0	0	0	0	0	0	0	0	
oktober	8	0.903089987	0	0.90309	0	0	0	0	0	
polisi	8	0.903089987	0	0.90309	0	0	0	0	0	
purnama	8	0.903089987	0.90309	0	0	0	0	0	0	
pusat	8	0.903089987	0	0	0.90309	0	0	0	0	
pusing	8	0.903089987	0	0	0	0.90309	0	0	0	
recana	8	0.903089987	0	0	0	0	0.90309	0	0	
ricuh	8	0.903089987	0	0.90309	0	0	0	0	0	
rumah	8	0.903089987	0	0	0.90309	0	0	0	0	
saksi	8	0.903089987	0	0.90309	0	0	0	0	0	
sekjen	8	0.903089987	0.90309	0	0	0	0	0	0	
sidang	8	0.903089987	0.90309	0	0	0	0	0	0	
silam	8	0.903089987	0	0.90309	0	0	0	0	0	
tangga p	8	0.903089987	0	0	0	0	0	0	0	

KATA	D/df	IDF log (D/df)	W (bobot)							
			S1	S2	S3	S4	S5	S6	S7	
tangg ung	8	0.90308 9987	0	0.903 09	0	0	0	0	0	
tawar	8	0.90308 9987	0	0	0	0	0	0.903 09	0	
tjahaja	8	0.90308 9987	0.9030 9	0	0	0	0	0	0	
top	8	0.90308 9987	0	0	0	0	0	0	0.903 09	
tuding	2.6667	0.42596 8732	0.4259 7	0	0	0.425 97	0	0	0	
tuntut	8	0.90308 9987	0	0.903 09	0	0	0	0	0	
urut	8	0.90308 9987	0.9030 9	0	0	0	0	0	0	
		Jumlah	17.005 3	14.82 43	7.650 69	4.589 24	5.543 48	1.505 15	1.505 15	

Keterangan :

W = bobot kata pada kalimat

c. *Cosine Similarity*

Cosine similarity digunakan untuk menghitung pendekatan relevansi *query* terhadap dokumen. Semakin banyak *term* yang muncul dalam dokumen, maka *term* tersebut dipandang sebagai *vector query*. Mencari nilai kemiripan antar kalimat dengan kalimat yang lain dengan menggunakan persmaan (**2.4**) untuk mendapatkan bobot *similarity* kalimat. Hasilnya dapat dilihat pada tabel 3.11 Bobot *Similarity* Kalimat dibawah ini. Sebagai contoh dilakukan perhitungan pada kalimat pertama (S1) dengan kalimat ke tiga (S2) sebagai berikut:

$$\boldsymbol{Sim}\left(\boldsymbol{S_{i,}}, \boldsymbol{S_{2,}}\right)$$

$$= \frac{(0.12494*0.12494)+(0.85194*0.42597)+(0.42597*0.42597)+(0.60206*0.30103)}{\sqrt{\begin{matrix}0.90309^2+0.12494^2+0.90309^2+0.60206^2+0.90309^2+0.90309^2+0.42597^2+0.90309^2+0.85194^2+\\ 0.90309^2+0.90309^2+0.42597^2+0.90309^2+0.60206^2+0.90309^2+0.90309^2+0.90309^2+0.90309^2\\ +0.90309^2+0.90309^2+0.90309^2+0.42597^2+0.90309^2\end{matrix}} * \sqrt{\begin{matrix}0.12494^2+0.90309^2+0.90309^2+0.90309^2+0.42597^2+0.90309^2+0.42597^2+0.90309^2+0.30103^2\\ +0.90309^2+0.90309^2+0.90309^2+0.90309^2+0.90309^2+0.90309^2+0.90309^2+0.90309^2+0.90309^2\\ +0.90309^2\end{matrix}}}$$

$$= 0.403040022$$

Tabel 3. 11 Bobot Similarity Kalimat

	S1	S2	S3	S4	S5	S6	S7
S1	1.60	0.40	0.09	0.17	0.01	0.00	0.00
S2	0.26	0.23	0.06	0.01	0.01	0.00	0.00
S3	0.09	0.06	1.13	0.01	0.01	0.00	0.00
S4	0.17	0.01	0.01	0.70	0.01	0.00	0.00
S5	0.01	0.01	0.01	0.01	0.82	0.00	0.00
S6	0.00	0.00	0.00	0.00	0.00	0.24	0.11
S7	0.00	0.00	0.00	0.00	0.00	0.11	0.24

3.2.3 Analisis Metode Hidden Markov Model

a. *Hidden Markov Model*

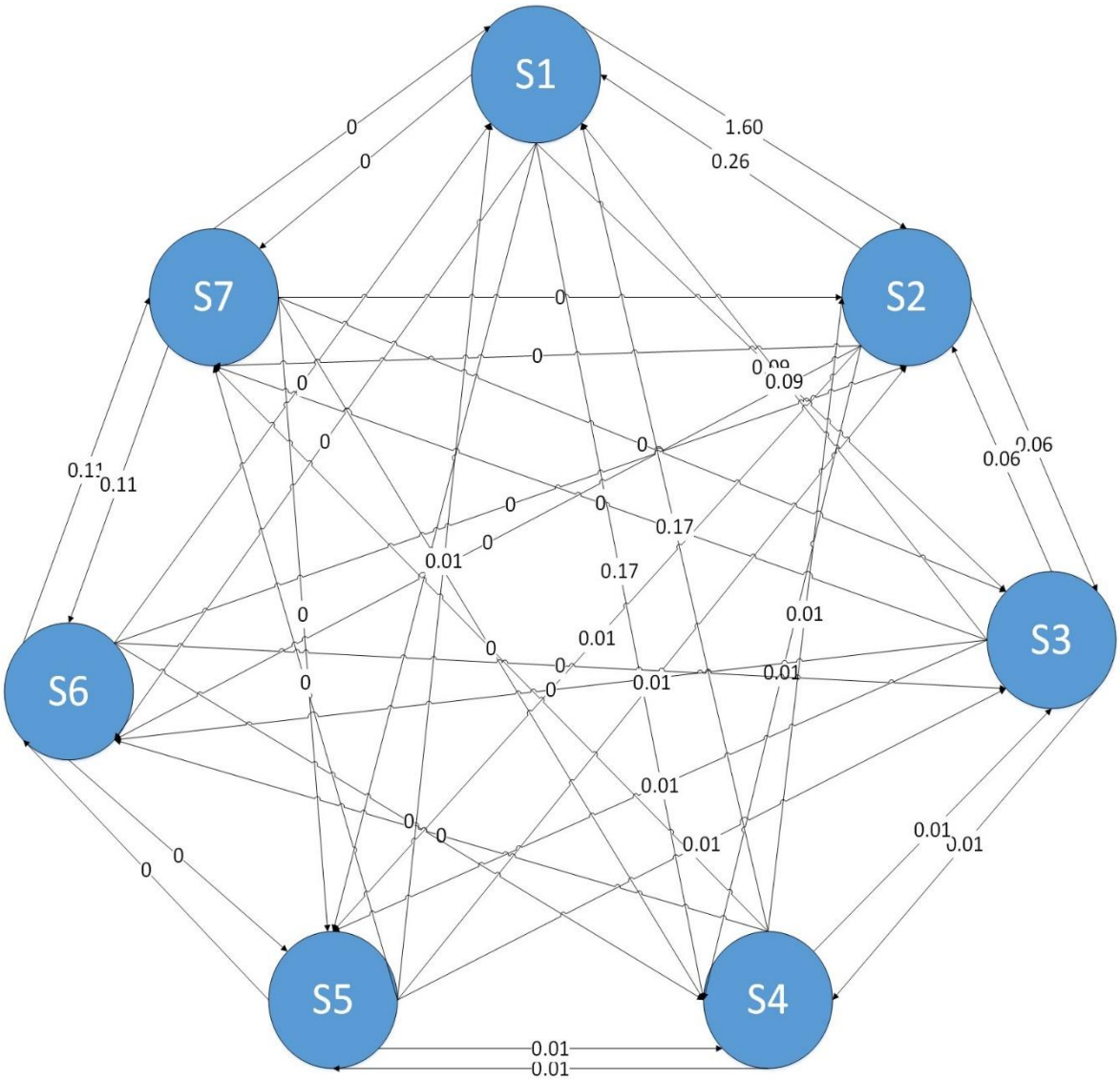
Pada proses hmm, akan digambarkan sebuah *state* yang menggambarkan kejadian yang saling terhubung dan masing-masing *state* memiliki nilai probabilitas. Pada penelitian ini yang menjadi *state* yang dapat diamati (*observation*) adalah "kalimat" dan yang menjadi pengamatan tersembunyi (*hidden*) adalah "kalimat relevan" dan "kalimat tidak relevan".

Pada tahapan ini dilakukan pembentukan notasi HMM yaitu $\lambda = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$ dengan terlebih dahulu menentukan 3 nilai parameter hitungan pada notasi tersebut yaitu nilai dari *similarity* kalimat untuk parameter A, nilai transpos *similarity kalimat* untuk parameter B, dan untuk nilai π = 0.5 . Pada nilai A dan

B akan membentuk sebuah matriks yang mencerminkan nilai perpindahan setiap *state.* Berikut ini adalah nilai probabilitas transisi (A) antar *state* yang didapatkan dari hasil perhitungan *similarity* (kemiripan) antar kalimat pada tabel 3.14. Probabilitas transisi yaitu menjelaskan peluang perpindahan dari *state* saat ini ke-*state* berikutnya.

$$A = a_{ij} = \begin{bmatrix} 1.60 & 0.40 & 0.09 & 0.17 & 0.01 & 0 & 0 \\ 0.26 & 0.23 & 0.06 & 0.01 & 0.01 & 0 & 0 \\ 0.09 & 0.06 & 1.13 & 0.01 & 0.01 & 0 & 0 \\ 0.17 & 0.01 & 0.01 & 0.70 & 0.01 & 0 & 0 \\ 0.01 & 0.01 & 0.01 & 0.01 & 0.82 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.24 & 0.11 \\ 0 & 0 & 0 & 0 & 0 & 0.11 & 0.24 \end{bmatrix}$$

Nilai A tersebut akan diilustrasikan pada gambar 3.4 dibawah ini.

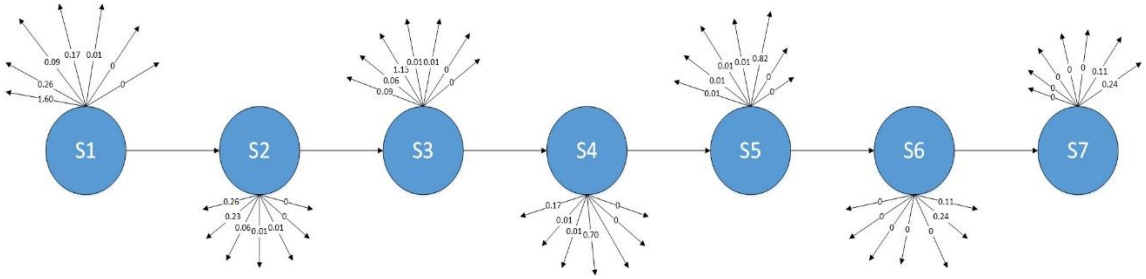


Gambar 3. 4 Ilustrasi Antar State Observation

Sedangkan untuk nilai parameter B atau nilai probabilitas emisi adalah hasil perhitungan dari tabel 3.14 terkhusus pada *state j*. Probabilitas emisi yaitu menjelaskan peluang dari *output* yang mungkin berbeda yang mungkin dari suatu *state*.

$$B = b_{jk} = \begin{bmatrix} 1.60 & 0.40 & 0.09 & 0.17 & 0.01 & 0 & 0 \\ 0.26 & 0.23 & 0.06 & 0.01 & 0.01 & 0 & 0 \\ 0.09 & 0.06 & 1.13 & 0.01 & 0.01 & 0 & 0 \\ 0.17 & 0.01 & 0.01 & 0.70 & 0.01 & 0 & 0 \\ 0.01 & 0.01 & 0.01 & 0.01 & 0.82 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.24 & 0.11 \\ 0 & 0 & 0 & 0 & 0 & 0.11 & 0.24 \end{bmatrix}$$

Nilai B akan diilustrasikan pada gambar 3.5 dibawah ini.



Gambar 3. 5 Ilustrasi Nilai Probabilitas Emisi

b. Algoritma *Forward*

Pada tahap ini akan dilakukan perhitungan nilai probabilitas pada observasi yang diberikan. Observasi yang dimaksud adalah kalimat-kalimat yang akan dipilih menjadi kalimat ringkasan. Dengan menggunakan 3 tahapan yang telah dijelaskan sebelumnya.

1. Tahap Inisialisasi

   Menggunakan persamaan **(2.7)**

   $\alpha_1(i) = \pi(i) b_i(O_1)$

   Pada tahapan ini aka ada dua buah α masing-masing untuk "Kalimat Relevan atau $\alpha_1(1)$" dan "Kalimat Tidak Relevan atau $\alpha_1(2)$", yaitu sebagai berikut :

   - Kalimat Relevan

     $\alpha_1(1) = (0.5) * (1.60) = 0.8$

perhitungan dilakukan sebanyak kalimat dan dapat dilihat pada tabel hasil 3.12 dibawah ini.

Tabel 3. 12 Hasil Tahap Inisialisasi $\alpha_1(1)$

Kalimat	Hasil
S1	0.080182
S2	0.013207
S3	0.004281
S4	0.008348
S5	0.000608
S6	0
S7	0
Jumlah	0.106626

- Kalimat Tidak Relevan

  $\alpha_1(2) = (0.5) * (1.60) = 0.8$

  perhitungan dilakukan sebanyak kalimat dan dapat dilihat pada tabel hasil 3.13 dibawah ini.

Tabel 3. 13 Hasil Tahap Inisialisasi $\alpha_1(2)$

Kalimat	Hasil
S1	0.080182
S2	0.013207
S3	0.004281
S4	0.008348
S5	0.000608
S6	0
S7	0
Jumlah	0.106626

Maka nilai untuk :

$\alpha_1(1) = 0.106626$

$\alpha_1(2) = 0.106626$

2. Tahapan Induksi

Menggunakan persamaan **(2.8)**

$\alpha_{t+1} = \{\Sigma_{i=1}^{N} \alpha_t(i)\alpha_{ij}\} b_j(O_{t+1})$

Pada tahapan ini akan dihitung $\alpha_2(j), \alpha_3(j), hingga\ \alpha_7(j)$

Tahap induksi dimulai dari t = 2

- $\alpha_2(1) = \{\big(\alpha_1(1) * \alpha_{ij}\big) + (\alpha_1(2) * \alpha_{ij})\} b_2(O_2)$

$= \{(0.106626 * 1.60) + (0.106626 * 0.26)\} * 0.26$

$= 0.05$

Hasil perhitungan untuk $\alpha_2(1)$ untuk seluruh kalimat dapat dilihat tabel 3.14 dibawah ini.

Tabel 3. 14 Hasil Untuk $\alpha_2(1)$ pada t = 2

Kalimat	Hasil
S1	0.052602
S2	0.011836
S3	0.000971
S4	0.000232
S5	3.15E-05
S6	0
S7	0
Jumlah	0.065674

- $\alpha_2(2) = \{\big(\alpha_1(1) * \alpha_{ij}\big) + (\alpha_1(2) * \alpha_{ij})\} b_2(O_2)$

$= \{(0.106626 * 0.26) + (0.106626 * 0.08)\} * 0.26$

$= 0.009$

Hasil perhitungan untuk $\alpha_2(2)$ untuk seluruh kalimat dapat dilihat tabel 3.15 dibawah ini.

Tabel 3. 15 Hasil Untuk $\alpha_2(2)$ pada t = 2

Kalimat	Hasil
S1	0.00985
S2	0.006583
S3	0.006024
S4	2.76E-05
S5	2.76E-05
S6	0
S7	0
Jumlah	0.022512

Tahap induksi pada saat t = 3

- $\alpha_3(1) = \{(\alpha_2(1) * \alpha_{ij}) + (\alpha_2(2) * \alpha_{ij})\} b_3(O_3)$

$$= \{(0.065674 * 0.26) + (0.022512 * 0.08)\} * 0.08$$

$$= 0.001$$

Hasil perhitungan untuk $\alpha_3(1)$ untuk seluruh kalimat dapat dilihat tabel 3.16 dibawah ini.

Tabel 3. 16 Hasil Untuk $\alpha_3(1)$ pada t = 3

Kalimat	Hasil
S1	0.00165
S2	0.001004
S3	0.033532
S4	1.3E-05
S5	1.3E-05
S6	0
S7	0

Kalimat	Hasil
Jumlah	0.036213

- $\alpha_3(2) = \{(\alpha_2(1) * \alpha_{ij}) + (\alpha_2(2) * \alpha_{ij})\} b_3(O_3)$

$$= \{(0.065674 * 0.08) + (0.022512 * 0.16)\} * 0.08$$

$$= 0.0008$$

Hasil perhitungan untuk $\alpha_3(2)$ untuk seluruh kalimat dapat dilihat tabel 3.17 dibawah ini.

Tabel 3. 17 Hasil Untuk $\alpha_3(2)$ pada t = 3

Kalimat	Hasil
S1	0.000803
S2	0.000268
S3	0.084712
S4	0.000202
S5	1.3E-05
S6	0
S7	0
Jumlah	0.085998

Tahapan induksi pada t = 4

- $\alpha_4(1) = \{(\alpha_3(1) * \alpha_{ij}) + (\alpha_3(2) * \alpha_{ij})\} b_4(O_4)$

$$= \{(0.036213 * 0.08) + (0.085998 * 0.16)\} * 0.16$$

$$= 0.002$$

Hasil perhitungan untuk $\alpha_4(1)$ untuk seluruh kalimat dapat dilihat tabel 3.18 dibawah ini.

Tabel 3. 18 Hasil Untuk $\alpha_4(1)$ pada t = 4

Kalimat	Hasil

S1	0.002915092
S2	3.99174E-05
S3	0.000511844
S4	0.04279776
S5	1.80658E-05
S6	0
S7	0
Jumlah	0.046282679

- $\alpha_4(2) = \{(\alpha_3(1) * \alpha_{ij}) + (\alpha_3(2) * \alpha_{ij})\} b_4(O_4)$

  $= \{(0.036213 * 0.16) + (0.085998 * 0.01)\} * 0.16$

  $= 1.05556E - 06$

  Hasil perhitungan untuk $\alpha_4(2)$ untuk seluruh kalimat dapat dilihat tabel 3.19 dibawah ini.

Tabel 3. 19 Hasil Untuk $\alpha_4(2)$ pada t = 4

Kalimat	Hasil
S1	1.05556E-06
S2	5.59719E-09
S3	5.59719E-09
S4	1.87071E-05
S5	3.75677E-07
S6	0
S7	0
Jumlah	2.01495E-05

Tahap induksi pada t = 5

- $\alpha_5(1) = \{(\alpha_4(1) * \alpha_{ij}) + (\alpha_4(2) * \alpha_{ij})\} b_5(O_5)$

  $= \{(0.046282679 * 0.16) + (2.01495\text{E} - 05 * 0.01)\} * 0.01$

  $= 9.39583E - 05$

Hasil perhitungan untuk $\alpha_5(1)$ untuk seluruh kalimat dapat dilihat tabel 3.20 dibawah ini.

Tabel 3. 20 Hasil Untuk $\alpha_5(1)$ pada t = 5

Kalimat	Hasil
S1	9.39583E-05
S2	6.84471E-06
S3	6.84471E-06
S4	0.000395537
S5	0.000472628
S6	0
S7	0
Jumlah	0.000975812

- $\alpha_5(2) = \{(\alpha_4(1) * \alpha_{ij}) + (\alpha_4(2) * \alpha_{ij})\} b_5(O_5)$

$$= \{(0.046282679 * 0.01) + (2.01495\text{E} - 05 * 0)\} * 0.01$$

$$= 6.84173E - 06$$

Hasil perhitungan untuk $\alpha_5(2)$ untuk seluruh kalimat dapat dilihat tabel 3.21 dibawah ini.

Tabel 3. 21 Hasil Untuk $\alpha_5(2)$ pada t = 5

Kalimat	Hasil
S1	6.84173E-06
S2	6.84173E-06
S3	6.84173E-06
S4	6.84173E-06
S5	0.030821625
S6	0
S7	0
Jumlah	0.030848992

Tahap induksi pada t = 6

- $\alpha_6(1) = \{(\alpha_5(1) * \alpha_{ij}) + (\alpha_5(2) * \alpha_{ij})\} b_6(O_6)$

$$= \{(0.000975812 * 0.01) + (0.030848992 * 0)\} * 0.01$$

$$= 0$$

Hasil perhitungan untuk $\alpha_6(1)$ untuk seluruh kalimat dapat dilihat tabel 3.22 dibawah ini.

Tabel 3. 22 Hasil Untuk $\alpha_6(1)$ pada t = 6

Kalimat	Hasil
S1	0
S2	0
S3	0
S4	0
S5	0
S6	0.001817
S7	0.000365
Jumlah	0.002182

- $\alpha_6(2) = \{(\alpha_5(1) * \alpha_{ij}) + (\alpha_5(2) * \alpha_{ij})\} b_6(O_6)$

$$= \{(0.000975812 * 0) + (0.030848992 * 0)\} * 0.01$$

$$= 0$$

Hasil perhitungan untuk $\alpha_6(2)$ untuk seluruh kalimat dapat dilihat tabel 3.23 dibawah ini.

Tabel 3. 23 Hasil Untuk $\alpha_6(2)$ pada t = 6

Kalimat	Hasil
S1	0
S2	0
S3	0
S4	0

Kalimat	Hasil
S5	0
S6	0.000872
S7	0.000826
Jumlah	0.001697

Tahap induksi pada t = 7

- $\alpha_7(1) = \{(\alpha_6(1) * \alpha_{ij}) + (\alpha_6(2) * \alpha_{ij})\} b_7(O_7)$

$$= \{(0.002182 * 0) + (0.001697 * 0)\} * 0$$

$$= 0$$

Hasil perhitungan untuk $\alpha_7(1)$ untuk seluruh kalimat dapat dilihat tabel 3.24 dibawah ini.

Tabel 3. 24 Hasil Untuk $\alpha_7(1)$ pada t = 7

Kalimat	Hasil
S1	0
S2	0
S3	0
S4	0
S5	0
S6	7.76E-05
S7	0.000158
Jumlah	0.000235

- $\alpha_7(2) = \{(\alpha_6(1) * \alpha_{ij}) + (\alpha_6(2) * \alpha_{ij})\} b_7(O_7)$

$$= \{(0.002182 * 0) + (0.001697 * 1)\} * 0$$

$$= 0$$

Hasil perhitungan untuk $\alpha_7(2)$ untuk seluruh kalimat dapat dilihat tabel 3.25 dibawah ini.

Tabel 3. 25 Hasil Untuk $\alpha_7(2)$ pada t = 7

Kalimat	Hasil
S1	0
S2	0
S3	0
S4	0
S5	0
S6	0.00021
S7	0.00054
Jumlah	0.000751

3. Tahap Terminasi

   Menggunakan persamaan **(2.9)**

   $P(O|\lambda) = \Sigma_{i=1}^{N} \alpha_T(i)$

   $$P(O|\lambda) = 0.000235 + 0.000751 = 0.000985971$$

   Maka probabilitas observasi yaitu $P(O|\lambda) = 0.000985971$

c. Algoritma *Viterbi*

Pada tahap ini algoritma *viterbi* digunakan untuk mencari barisan keadaan tersembunyi yang paling optimal dari suatu barisan observasi. Barisan observasi tersebut merepresentasikan barisan kalimat yang akan dikelompokkan berdasarkan suatu keadaan *hidden* (tersembunyi) yaitu "Kalimat Relevan" dan "Kalimat Tidak Relevan". Kedua keadaan *hidden* tersebut akan menghasilkan nilai probabilitas yang nantinya akan dipilih nilai probabilitas mana yang paling optimal dari keduanya, nilai yang paling optimal akan dipilih sebagai kalimat yang menjadi ringkasan pada teks. Penyelesaian algoritma *viterbi* dapat diselesaikan dengan empat tahap, yaitu :

1. Tahap Inisialisasi

   Dengan $O_1 = 1(Kalimat\ 1)$

   Menggunakan persamaan **(2.11)**

   $\delta_1(i) = b_i(O_1)\pi(i)$

   pada tahap ini $\psi_1(i) = 0$

   Pada tahapan ini aka ada dua buah δ masing-masing untuk "Kalimat Relevan atau $\delta_1(1)$" dan "Kalimat Tidak Relevan atau $\delta_1(2)$", yaitu sebagai berikut :

   - Kalimat Relevan

     $\delta_1(1) = (0.5) * (1.60) = 0.8$

     perhitungan dilakukan sebanyak kalimat dan dapat dilihat pada tabel hasil 3.26 dibawah ini.

Tabel 3. 26 Hasil Tahap Inisialisasi $\delta_1(1)$

Kalimat	Hasil
S1	0.080182
S2	0.013207
S3	0.004281
S4	0.008348
S5	0.000608
S6	0
S7	0
Jumlah	0.106626

   - Kalimat Tidak Relevan

     $\delta_1(2) = (0.5) * (1.60) = 0.8$

     perhitungan dilakukan sebanyak kalimat dan dapat dilihat pada tabel hasil 3.27 dibawah ini.

Tabel 3. 27 Hasil Tahap Inisialisasi $\delta_1(2)$

Kalimat	Hasil
S1	0.080182
S2	0.013207
S3	0.004281
S4	0.008348
S5	0.000608
S6	0
S7	0
Jumlah	0.106626

Maka nilai untuk :

$\delta_1(1) = 0.106626$

$\delta_1(2) = 0.106626$

$\psi_1(1) = \psi_1(2) = 0$

2. Tahap Rekursi

Untuk t = 2 , $O_2 = 2$ $(Kalimat\ 2)$

Menggunakan persamaan **(2.13)**

$\delta_t(j) = \max_{1 \le i \le N} \{a_{ij}\delta_{t-1}(i)\}$

Pada tahapan ini akan dihitung $\delta_2(j), \delta_3(j), hingga\ \delta_7(j)$

Tahap rekursi dimulai dari t = 2

- $\delta_2(1) = max\ \{(\delta_1(1) * \alpha_{ij}), (\delta_1(2) * \alpha_{ij})\} b_2(O_2)$

$= max\{(0.106626 * 1.60), (0.106626 * 0.26)\} * 0.26$

$= max\{(0.1706016), (0.02772276)\} * 0.26$

$= 0.04435642$

Hasil perhitungan untuk $\delta_2(1)$ untuk seluruh kalimat dapat dilihat tabel 3.28 dibawah ini.

Tabel 3. 28 Hasil Untuk $\delta_2(1)$ pada t = 2

X1	X2	O	Terbesar
0.170989	0.028163	0.264132	0.045163693
0.028163	0.024133	0.226334	0.006374306
0.009129	0.006588	0.061789	0.000564067
0.017803	0.001296	0.012158	0.000216454
0.001296	0.001296	0.012158	1.57619E-05
0	0	0	0
0	0	0	0
Jumlah			0.052334282

- $\delta_2(2) = max\{(\delta_1(1) * \alpha_{ij}), (\delta_1(2) * \alpha_{ij})\} b_2(O_2)$

  $= max\{(0.106626 * 0.26), (0.106626 * 0.08)\} * 0.26$

  $= max\{(0.028163), (0.009129)\} * 0.26$

  $= 0.007438807$

  Hasil perhitungan untuk $\delta_2(2)$ untuk seluruh kalimat dapat dilihat tabel 3.29 dibawah ini.

Tabel 3. 29 Hasil Untuk $\delta_2(2)$ pada t = 2

X1	X2	O	Terbesar
0.028163	0.009129	0.264132	0.007438807
0.024133	0.006588	0.226334	0.005462136
0.006588	0.120876	0.061789	0.007468784
0.001296	0.001296	0.012158	1.57619E-05
0.001296	0.001296	0.012158	1.57619E-05
0	0	0	0
0	0	0	0
Jumlah			0.020401251

Tahap rekursi dimulai dari t = 3

- $\delta_3(1) = max\ \{(\delta_2(1) * \alpha_{ij}), (\delta_2(2) *\ \alpha_{ij})\} b_3(O_3)$

$= max\{(0.065674 * 1.60)\ , (0.022512 * 0.26)\} * 0.085617$

$= max\{(0.105317)\ , (0.005946)\} * 0.085617$

$= 0.009016932$

Hasil perhitungan untuk $\delta_3(1)$ untuk seluruh kalimat dapat dilihat tabel 3.30 dibawah ini.

Tabel 3. 30 Hasil Untuk $\delta_3(1)$ pada t = 3

X1	X2	O	Terbesar
0.105317	0.005946	0.085617	0.009016932
0.017347	0.005095	0.061789	0.001071822
0.005623	0.001391	1.133649	0.006374269
0.010965	0.000274	0.012158	0.00013332
0.000798	0.000274	0.012158	9.70823E-06
0	0	0	0
0	0	0	0
Jumlah			0.016606051

- $\delta_3(2) = max\ \{(\delta_2(1) * \alpha_{ij}), (\delta_2(2) *\ \alpha_{ij})\} b_3(O_3)$

$= max\{(0.065674 * 1.60)\ , (0.022512 * 0.26)\} * 0.085617$

$= max\{(0.017347)\ , (0.001927)\} * 0.08$

$= 0.001485158$

Hasil perhitungan untuk $\delta_2(2)$ untuk seluruh kalimat dapat dilihat tabel 3.31 dibawah ini.

Tabel 3. 31 Hasil Untuk $\delta_3(2)$ pada t = 3

X1	X2	O	Terbesar
0.017347	0.001927	0.085617	0.001485158
0.014864	0.001391	0.061789	0.000918443

X1	X2	O	Terbesar
0.004058	0.025521	1.133649	0.028931937
0.000798	0.000274	0.012158	9.70823E-06
0.000798	0.000274	0.012158	9.70823E-06
0	0	0	0
0	0	0	0
Jumlah			0.031354954

Tahapan induksi pada t = 4

- $\delta_4(1) = max\{(\delta_3(1) * \alpha_{ij}), (\delta_3(2) * \alpha_{ij})\} b_4(O_4)$

$= max\{(0.036213 * 0.08), (0.085998 * 0.16)\} * 0.16$

$= max\{(0.0031), (0.014359)\} * 0.16$

$= 0.002397424$

Hasil perhitungan untuk $\delta_4(1)$ untuk seluruh kalimat dapat dilihat tabel 3.32 dibawah ini.

Tabel 3. 32 Hasil Untuk $\delta_4(1)$ pada t = 4

X1	X2	O	Terbesar
0.0031	0.014359	0.166966	0.002397424
0.002238	0.001046	0.012158	2.72048E-05
0.041053	0.001046	0.012158	0.000499131
0.00044	0.060447	0.702896	0.042488284
0.00044	0.001046	0.012158	1.27126E-05
0	0	0	0
0	0	0	0
Jumlah			0.045424756

- $\delta_4(2) = max\{(\delta_3(1) * \alpha_{ij}), (\delta_3(2) * \alpha_{ij})\} b_4(O_4)$

$= max\{(0.036213 * 0.16), (0.085998 * 0.01)\} * 0.16$

$= max\{(0.006046), (0.001046)\} * 0.16$

$$= 0.001009533$$

Hasil perhitungan untuk $\alpha_4(2)$ untuk seluruh kalimat dapat dilihat tabel 3.33 dibawah ini.

Tabel 3. 33 Hasil Untuk $\delta_4(2)$ pada t = 4

X1	X2	O	Terbesar
0.006046	0.001046	0.166966	0.001009533
0.00044	0.001046	0.012158	1.27126E-05
0.00044	0.001046	0.012158	1.27126E-05
0.025454	0.001046	0.702896	0.017891427
0.00044	0.070179	0.012158	0.000853255
0	0	0	0
0	0	0	0
Jumlah			0.019779641

Tahap induksi pada t = 5

- $\delta_5(1) = max\{(\delta_4(1) * \alpha_{ij}), (\delta_4(2) * \alpha_{ij})\} b_5(O_5)$

$$= max\{(0.046282679 * 0.16), (2.01495\text{E} - 05 * 0.01)\} * 0.01$$

$$= max\{(0.007728), (2.44984\text{E} - 07)\} * 0.01$$

$$= 9.39583E - 05$$

Hasil perhitungan untuk $\delta_5(1)$ untuk seluruh kalimat dapat dilihat tabel 3.34 dibawah ini.

Tabel 3. 34 Hasil Untuk $\delta_5(1)$ pada t = 5

X1	X2	O	Terbesar
0.007728	2.44984E-07	0.012158	9.39553E-05
0.000563	2.44984E-07	0.012158	6.84173E-06
0.000563	2.44984E-07	0.012158	6.84173E-06
0.032532	2.44984E-07	0.012158	0.000395534
0.000563	1.6443E-05	0.816053	0.000459209

X1	X2	O	Terbesar
0	0	0	0
0	0	0	0
Jumlah			0.000962382

- $\delta_5(2) = max\{(\delta_4(1) * \alpha_{ij}), (\delta_4(2) * \alpha_{ij})\} b_5(O_5)$

$= max\{(0.046282679 * 0.01), (2.01495\text{E} - 05 * 0)\} * 0.01$

$= max\{(0.000563), (0)\} * 0.01$

$= 6.84173E - 06$

Hasil perhitungan untuk $\delta_5(2)$ untuk seluruh kalimat dapat dilihat tabel 3.35 dibawah ini.

Tabel 3. 35 Hasil Untuk $\delta_5(2)$ pada t = 5

X1	X2	O	Terbesar
0.000563	0	0.012158	6.84173E-06
0.000563	0	0.012158	6.84173E-06
0.000563	0	0.012158	6.84173E-06
0.000563	0	0.012158	6.84173E-06
0.037769	0	0.816053	0.030821625
0	4.89047E-06	0	0
0	2.19079E-06	0	0
Jumlah			0.030848992

Tahap induksi pada t = 6

- $\delta_6(1) = max\{(\delta_5(1) * \alpha_{ij}), (\delta_5(2) * \alpha_{ij})\} b_6(O_6)$

$= max\{(0.000975812 * 0.01), (0.030848992 * 0)\} * 0$

$= max\{(1.18642\text{E} - 05), (0)\} * 0$

$= 0$

Hasil perhitungan untuk $\delta_6(1)$ untuk seluruh kalimat dapat dilihat tabel 3.36 dibawah ini.

Tabel 3. 36 Hasil Untuk $\delta_6(1)$ pada t = 6

X1	X2	O	Terbesar
1.18642E-05	0	0	0
1.18642E-05	0	0	0
1.18642E-05	0	0	0
1.18642E-05	0	0	0
0.000796315	0	0	0
0	0.007487	0.242709	0.001817
0	0.003354	0.108727	0.000365
Jumlah			0.002182

- $\delta_6(2) = max\{(\delta_5(1) * \alpha_{ij}), (\delta_5(2) * \alpha_{ij})\} b_6(O_6)$

  $= max\{(0.000975812 * 0), (0.030848992 * 0)\} * 0$

  $= max\{(0), (0)\} * 0$

  $= 0$

  Hasil perhitungan untuk $\delta_6(2)$ untuk seluruh kalimat dapat dilihat tabel 3.37 dibawah ini.

Tabel 3. 37 Hasil Untuk $\delta_6(2)$ pada t = 6

X1	X2	O	Terbesar
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0.000236839	0.003354	0.242709	0.000814
0.000106097	0.007487	0.108727	0.000814
Jumlah			0.001628

Tahap induksi pada t = 7

- $\delta_7(1) = max\{(\delta_6(1) * \alpha_{ij}), (\delta_6(2) * \alpha_{ij})\} b_7(O_7)$

$$= max\{(0.002182 * 0), (0.001697 * 0)\} * 0$$

$$= max\{(0), (0)\} * 0$$

$$= 0$$

Hasil perhitungan untuk $\delta_7(1)$ untuk seluruh kalimat dapat dilihat tabel 3.38 dibawah ini.

Tabel 3. 38 Hasil Untuk $\delta_7(1)$ pada t = 7

X1	X2	O	Terbesar
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0.00053	0.000185	0.108727	5.7579E-05
0.000237	0.000412	0.242709	9.99765E-05
Jumlah			0.000157556

- $\delta_7(2) = max\{(\delta_6(1) * \alpha_{ij}), (\delta_6(2) * \alpha_{ij})\} b_7(O_7)$

$$= max\{(0.002182 * 0), (0.001697 * 1)\} * 0$$

$$= max\{(0), (0.001697)\} * 0$$

$$= 0$$

Hasil perhitungan untuk $\delta_7(2)$ untuk seluruh kalimat dapat dilihat tabel 3.39 dibawah ini.

Tabel 3. 39 Hasil Untuk $\delta_7(2)$ pada t = 7

X1	X2	O	Terbesar
0	0.001697	0	0
0	0.001697	0	0
0	0.001697	0	0

X1	X2	O	Terbesar
0	0.001697	0	0
0	0.001697	0	0
0.000185	0.001697	0.108727	0.000184528
0.000412	0.001697	0.242709	0.000411919
Jumlah			0.000596446

3. Tahapan Terminasi

   Menggunakan persamaan **(2.14 dan 2.15)**

$$\boldsymbol{P^* = max\{\delta_7(1)\ , \delta_7(2)\}}$$

$$\boldsymbol{= max\{}(0.000157556), (0.000596446)\boldsymbol{\}}$$

$$\boldsymbol{= 0.000596446}$$

$$\boldsymbol{X_7^* = arg\ max\{\delta_7(1), \delta_7(2)\} = 2\ (Kalimat\ Tidak\ Relevan)}$$

4. Tahap *Backtracking*

   Menggunakan persamaan **(2.16)**

   $X_t^* = \psi_{t+1}\ (X_{t+1}^*), \qquad dimana\ \ t = T-1, T-2, \dots, 1$

   $X_6^* = \psi_7\ (X_7^*) = 2\ (Kalimat\ Tidak\ Relevan)$

   $X_5^* = \psi_6\ (X_6^*) = 1\ (Kalimat\ Relevan)$

   $X_4^* = \psi_5\ (X_5^*) = 2\ (Kalimat\ Tidak\ Relevan)$

   $X_3^* = \psi_4\ (X_4^*) = 1\ (Kalimat\ Relevan)$

   $X_2^* = \psi_3\ (X_3^*) = 2\ (Kalimat\ Tidak\ Relevan)$

   $X_1^* = \psi_2\ (X_2^*) = 1\ (Kalimat\ Relevan)$

d. Pembentukan Ringkasan

Pada tahap ini sistem akan menentukan kalimat yang menjadi ringkasan dengan cara memilih nilai probabilitas *state* dari hasil perhitungna algoritma *viterbi* yang diperlihatkan pada tahap keempat (*backtracking*) yang bernilai paling optimal. Berikut disajikan tabel hasil perhitungan algoritma *viterbi* yang telah dikelompokkan kedalam tabel dibawah ini.

Tabel 3. 40 Penentuan Nilai State Paling Optimal

	S1	Check List	S2	Check List	S3	Check List	S4	Check List
Kalimat Relevan	0.106626	x	0.052334	✓	0.016606		0.045425	✓
Kalimat Tidak Relevan	0.106626	x	0.015374		0.031355	✓	0.01978	

S5	Check List	S6	Check List	S7	Check List
0.000962		0.002182	✓	0.000158	
0.030849	✓	0.001628		0.000596	✓

Pada tabel 3.43 nilai yang diberi tanda centang (*checklist*) merupakan nilai *state* yang paling optimal, nilai yang diberi tanda silang tidak menjadi pilihan untuk kalimat yang dijadikan ringkasan karena proses dalam algoritma *viterbi* perpindahan *state* awal ke-*state* berikutnya bernilai 0 (nol). Untuk pemilihan kalimat yang menjadi ringkasan diambil dari kelompok "Kalimat Relevan" yang diberi tanda centang (*checklist*) pada tabel 3.41. Berikut disajikan tabel kalimat relevan yang terpilih menjadi kalimat ringkasan dibawah ini.

Tabel 3. 41 Kalimat Ringkasan

Kalimat Relevan		
	Nilai	Kalimat
S2	0.052334282	Novel yang menjadi saksi pelapor dari Jaksa Penuntut Umum (JPU) ini menganggap Ahok harus bertanggung jawab akibat demo ricuh FPI dengan polisi medio Oktober 2014 silam di depan Gedung DPRD DKI Jakarta.

Kalimat Relevan		
	Nilai	Kalimat
S4	0.045424756	Ahok mengaku enggan ambil pusing atas tudingan Novel tersebut.
S6	0.00218193	"Tadinya saya mau bilang, kan kata dia (Novel) dua, saya mau tawarkan dua apa satu, dua atau satu, atau dua atau tiga.

3.3 Spesifikasi Kebutuhan Perangkat Lunak

Spesifikasi kebutuhan perangkat lunak untuk implementasi metode *Hidden Markov Model* untuk menghasilkan rangkuman dari artikel berbahasa Indonesia, terdiri dari Spesifikasi Kebutuhan Perangkat Lunak-Fungsional (SKPL-F) dan Spesifikasi Kebutuhan Perangkat Lunak-Non Fungsionel (SKPL-NF). Berikut adalah tabel dari SKPL-F dan SKPL-NF dapat dilihat pada tabel 3.42 dan tabel 3.43 dibawah ini.

Tabel 3. 42 Spesifikasi Kebutuhan Perangkat Lunak Fungsionalitas

Kode	Kebutuhan
SKPL-F-01	Sistem dapat menginputkan file dokumen.
SKPL-F-02	Sistem dapat menampilkan dokumen asli untuk diringkas.
SKPL-F-03	Sistem dapat melakukan tahap *Preprocessing.*
SKPL-F-04	Sistem dapat melakukan pemecahan kalimat.
SKPL-F-05	Sistem dapat melakukan case folding.
SKPL-F-06	Sistem dapat melakukan *filtering* kalimat.
SKPL-F-07	Sistem dapat melakukan *Tokenizing*.
SKPL-F-08	Sistem dapat melakukan *Stop Word Removal.*

Kode	Kebutuhan
SKPL-F-09	Sistem dapat melakukan *Stemming.*
SKPL-F-10	Sistem dapat melakukan perhitungan pembobotan TF-IDF.
SKPL-F-11	Sistem dapat melakukan perhitungan pembobotan *Cosine Similarity.*
SKPL-F-12	Sistem dapat melakukan perhitungan *Query Relevance.*
SKPL-F-13	Sistem dapat melakukan perhitungan *Similarity* kalimat.
SKPL-F-14	Sistem dapat melakukan perhitungan probabilitas HMM.
SKPL-F-15	Sistem dapat melakukan perhitungan probabilitas Forward.
SKPL-F-16	Sistem dapat menyajikan hasil ringkasan.
SKPL-F-17	Sistem dapat menampilkan hasil evaluasi ringkasan.
SKPL-F-18	Sistem dapat melakukan perhitungan *Precession.*
SKPL-F-19	Sistem dapat melakukan perhitungan *Recall.*
SKPL-F-20	Sistem dapat melakukan perhitungan *F-measure.*

Tabel 3. 43 Spesifikasi Kebutuhan Perangkat Lunak Non Fungsional

Kode	Kebutuhan
SKPL-NF-01	Perangkat lunak yang digunakan oleh OS Windows 10 Pro 64 Bit, NetBeans IDE, dan JDK.

3.4 Analisis Kebetuhan Non-Fungsional

Analisis non-sungsional yang dilakukan dibagi kedalam 2 tahap, yaitu analisis kebutuhan perangkat keras dan analisis kebutuhan perangkat lunak.

3.4.1 Analisis Kebetuhan Perangkat Keras

Analisis kebutuhan perangkat keras pada penelitian ini merupakan kebutuhan perangkat keras minimum untuk menjalankan aplikasi. Adapun perangkat keras yang digunakan sebagai berikut:

1. Laptop dengan *processor* 800 MHz *Intel Pentium* III (*recommended: Intel Core i5 or equivalent*) .
2. RAM 512 MB (*recommended* : 2 GB(32-bit), 4 GB(64-bit)).
3. *Harddisk* 700 MB (*recommended* : 1.5 GB *of free disk space*).
4. *Minimum* Resolusi layar 1024 x 768 *pixel.*
5. IBM *keyboard* dan *mouse.*

3.4.2 Analisis Kebutuhan Perangkat Lunak

Analisis kebutuhan perangkat lunak pada penelitian ini merupakan *tools* yang digunakan dalam pembangunan aplikasi. Adapun *tools* yang digunakan sebagai berikut :

1. OS Windows 10 Pro 64 bit.
2. NetBeans IDE.
3. *Java Development Kit* dan *Java Runtime Environtment.*

3.5 Analisis Pengguna

Karakteristik pengguna yang dapat menjalankan sistem yang akan dibangun hanya terdapat satu jenis *user* yaitu seorang penguji. Penguji bekerja untuk menjalankan serta mengetahui hasil dari aplikasi yang dijalankan. Adapun spesifikasi pengguna yang dibutuhkan adalah sebagai berikut :

1. Menguasai penggunaan computer.
2. Mengerti secara teknis *tools* dan *software* pendukung dalam menjalankan aplikasi.
3. Mengerti tahap-tahapan dalam menjalankan aplikasi.
4. Memahami preses dan kebutuhan dalam menggunakan aplikasi.

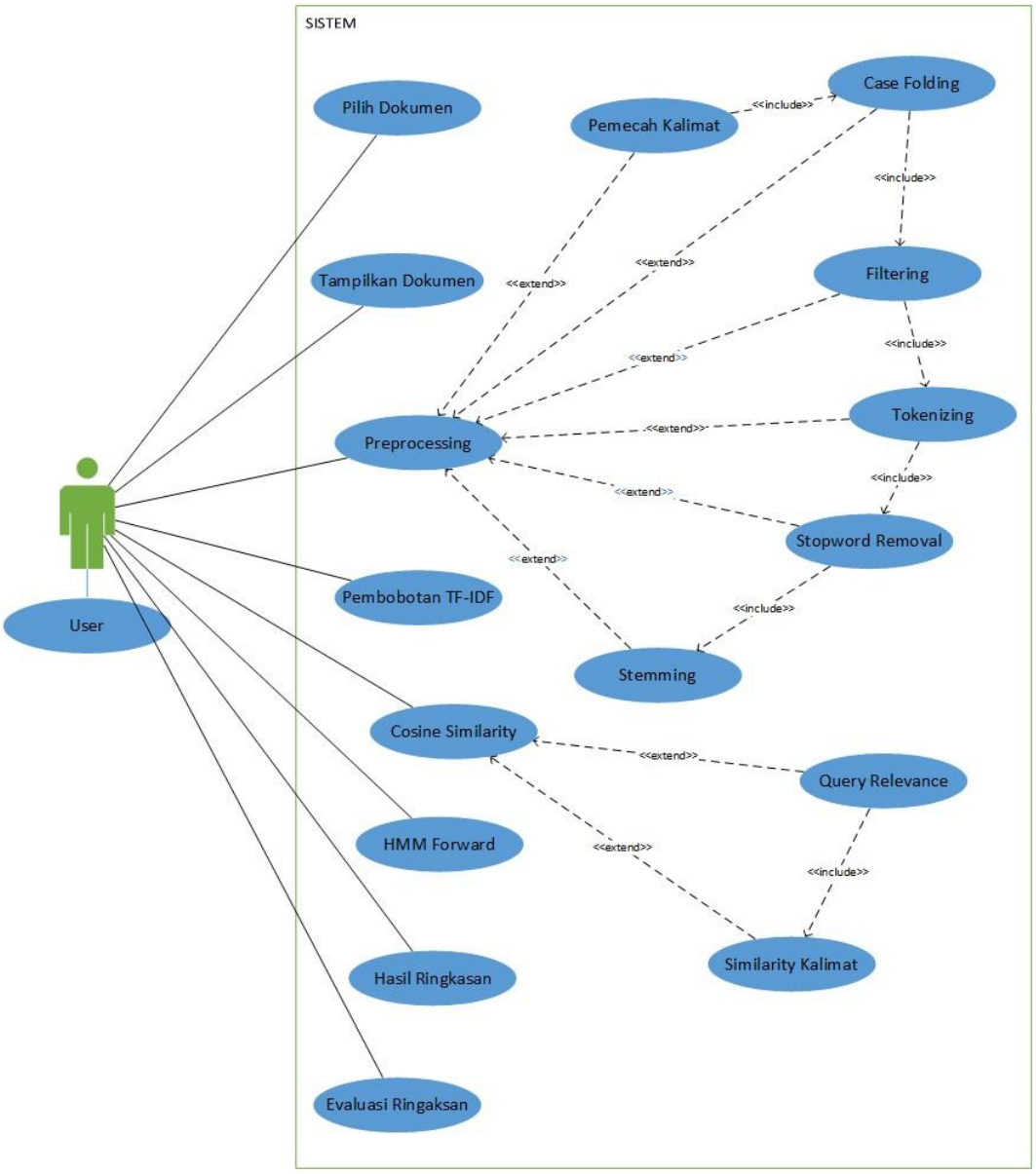
3.6 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional bertujuan untuk menganalisis proses yang diterapkan dalam sistem dan menjelasksn kebutuhan yang diperlukan. Menganalisis operasional sistem dengan mendefinisikan skenario penggunaan aplikasi. Analisis yang dilakukan dimodelkan dengan menggunakan UML (*Unified Modelling Language*).

UML adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara visual (Braun, *el. al* 2011). Juga merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *software* yang terkait dengan objek (Whitten, *el. al.* 2004). Bagian-bagian yang dilakukan dalam analisis tersebut antara lain *Use Case Diagram, Use Case Scenario, Activity Diagram, Class Diagram,* dan *Sequence Diagram.*

3.6.1 Use Case Diagram

Use case diagram adalah gambaran umum sistem dari sudut pandang pengguna sistem. Tujuan dari *use case* adalah untuk menggambarkan apa yang sistem dapat lakukan. *Use case diagram* dibentuk dari scenario tentang kegunaan sistem yang dinotasikan dengan sebuah *use case.* Setiap scenario menjelaskan suatu alur kegiatan, dapat diinisialisasi oleh pengguna sistem yang disebut actor. Adapun *use case diagram* dapat dilihat pada gamvar 3.6 berikut :



Gambar 3. 6 Use Case Diagram APT

Berikut ini merupakan deskripsi actor yang dapat dilihat pada tabel 3.44 Berikut :

Tabel 3. 44 Deskripsi Aktor

Aktor	Deskripsi
User	*User* dapat melakukan peng-*input*-an data masukan, serta melihat data hasil proses *preprocessing*, pembobotan tf-idf,

	perhitungan *Cosine Similarity*, pembentukan ringkasan, dan evaluasi ringkasan.

Berikut merupakan deskripsi *use case* yang dapat dilihat pada tabel 3.45 Berikut:

Tabel 3. 45 Deskripsi *Use Case*

NO.	*Use Case*	Deskripsi
1.	Pilih Dokumen	Fungsionalitas ini digunakan sistem untuk memilih *file* berupa dokumen dengan format *file.txt*
2.	Tampilkan Dokumen	Fungsionalitas ini digunakan sistem untuk menampilkan *file* dokumen yang dipilih.
3.	*Preprocessing*	Fungsionalitas ini digunakan sistem untuk memecahakn data dokumen, sehingga layak untuk diproses selanjutnya.
4.	Pemecah Kalimat	Fungsionalitas ini digunakan sistem untuk memisahkan data dokumen menjadi data perkalimat.
5.	*Case Folding*	Fungsionalitas ini digunakan sistem untuk mengubuh data kalimat yang memiliki huruf capital menjadi huruf kecil.
6.	*Filtering*	Fungsionalitas ini digunakan sistem untuk menghilangkan tanda baca pada kalimat selain spasi dan karakter “a”-“z”..
7.	*Tokenizing*	Fungsionalitas ini digunakan sistem untuk memecahkan data kalimat dan memisahkannya menjadi data perkata.

NO.	*Use Case*	Deskripsi
8.	*Stopword Removal*	Fungsionalitas ini digunakan sistem untuk menghilangkan kata-kata yang masuk kedalam daftar *stopword.*
9.	*Stemming*	Fungsionalitas ini digunakan sistem untuk mengubah kata yang memiliki imbuhan kedalam kata dasarnya.
10.	Pembobotan TF-IDF	Fungsionalitas ini digunakan sistem untuk melakukan perhitungan pembobotan TF-IDF.
11.	Pembobotan *Cosine Similarity*	Fungsionalitas ini digunakan sistem untuk melakukan perhitungan *Cosine Similarity.*
12.	*Query Relevance*	Fungsionalitas ini digunakan sistem untuk melakukan perhitungan *Query Relevance.*
13.	*Similarity* Kalimat	Fungsionalitas ini digunakan sistem untuk melakukan perhitungan *Similarity* kalimat.
14.	HMM *Forward*	Fungsionalitas ini digunakan sistem untuk melakukan input nilai kenotasi HMM Forward.
16.	Penyajian Ringkasan	Fungsionalitas ini digunakan sitem untuk menampilkan hasil ringkasan.
17.	Evaluasi Ringkasan	Fungsionalitas ini digunakan sistem untuk menampilkan perhitungan hasil evaluasi ringkasan.

3.6.2 Use Case Scenario

Bagian ini menjelaskan scenario untuk tiap *use case* yang menggambarkan urutan intraksi actor dengan *use case* tersebut dari awal sampai akhir.

Tabel 3. 46 *Use Case Scenario* Pilih Dokumen

Nama *Use Case*	Pilih Dokumen
Tujuan	*Input data.*
Deskripsi	Fungsionalitas ini digunakan sistem untuk memilih *file* berupa dokumen dengan *file.txt*
Aktor	*User*
Related Case	-
Kondisi Sukses	Data masuk ke proses selanjutna.
Kondisi Gagal	Data masukan gagal masuk keproses selanjutnya.
Include Case	-
Trigger	Menekan tombol Pilih.
Skenario Utama	
Aksi Aktor	Reaksi Sistem
1. Memilih *file input* sesuai dengan format *file* yang dipilih (*choose file*).	
2. Aktor menekan tombol Pilih.	
	3. Data masukan akan dilanjutkan pada proses berikutnya.
Skenario Alternatif	
Aksi Aktor	Reaksi Sistem

1. Aktor menekan tombol Pilih.	
	2. *File* data masukan belum dipilih.

Tabel 3. 47 *Use Case Scenario* Tampilkan Dokumen

Nama *Use Case*	Tampilkan Dokumen
Tujuan	Menampilkan *data.*
Deskripsi	Fungsionalitas ini digunakan sistem untuk menampilkan *file* dokumen yang dipilih.
Aktor	*User*
Related Case	-
Kondisi Sukses	Dokumen ditampilkan.
Kondisi Gagal	Dokumen masukan gagal ditampilkan.
Include Case	-
Trigger	Menekan tombol Pilih.
Skenario Utama	
Aksi Aktor	Reaksi Sistem
1. Memilih data masukan sesuai dengan format *file* yang dipilih.	
2. Aktor menekan tombol proses.	
	3. Data masukan masuk kemudian akan ditampilkan.
Skenario Alternatif	

Aksi Aktor	Reaksi Sistem
1. Aktor menekan tombol Pilih.	
	2. *File* data masukan belum dipilih.

Tabel 3. 48 *Use Case Scenario Preprocessing*

Nama *Use Case*	*Preprocessing*
Tujuan	Proses *data.*
Deskripsi	Fungsionalitas ini digunakan sistem untuk memecahkan data dokumen, sehingga layak untuk diproses selanjutnya.
Aktor	*User*
Related Case	-
Kondisi Sukses	Data masukan berhasil diproses.
Kondisi Gagal	Dokumen masukan gagal diproses.
Include Case	Pemecah Kalimat, *Case Folding, Filtering, Tokenizing, Stopword Removal, Stemming.*
Trigger	Terdapat data masukan dan menekan tombol Lanjutkan kemudian pilih menu *Preprocessing*..
Skenario Utama	
Aksi Aktor	Reaksi Sistem
1. Aktor menekan tombol Proses.	

	2. Sistem melakukan proses *preprocessing.*.
	3. Data masukan berhasil di-*preprocessing.*
	4. Menampilkan hasil *preprocessing*
Skenario Alternatif	
Aksi Aktor	Reaksi Sistem
1. Aktor menekan tombol Proses.	
	2. Data masukan gagal di-*preproses*.

Tabel 3. 49 *Use Case Scenario* Pemecahan Kalimat

Nama *Use Case*	Pemecah Kalimat
Tujuan	Menjadikan data perkalimat..
Deskripsi	Fungsionalitas ini digunakan sistem untuk memisahkan data dokumen menjadi data perkalimat.
Aktor	*User*
Related Case	*Preprocessing.*
Kondisi Sukses	Data masukan berhasil melalui proses *preprocessing* pemisahan kalimat.
Kondisi Gagal	Data masukan gagal melalui proses *preprocessing* pemisahn kalimat.
Include Case	-

Trigger	Terdapat data masukan yang akan diproses pemisahan kalimat..
Skenario Utama	
Aksi Aktor	Reaksi Sistem
	1. Sistem melakukan *preprocessing* pemecahan kalimat.
	2. Data masukan berhasil melaluai proses *preprocessing* pemecahan kalimat.
	3. Manampilkan hasil *preprocessing* pemecahan kalimat.
Skenario Alternatif	
Aksi Aktor	Reaksi Sistem
	1. Data masukan gagal melalui proses *preprocessing* pemisahan kalimat.

Tabel 3. 50 *Use Case Scenario Case Folding*

Nama *Use Case*	*Case Folding*
Tujuan	Mengubah huruf capital menjadi huruf kecil.
Deskripsi	Fungsionalitas ini digunakan sistem untuk mengubuh data kalimat yang memiliki huruf capital menjadi huruf kecil.

Aktor	*User*
Related Case	*Preprocessing.*
Kondisi Sukses	Data masukan berhasil melalui proses *preprocessing case folding*.
Kondisi Gagal	Data masukan gagal melalui proses *preprocessing case folding*.
Include Case	-
Trigger	Terdapat data hasil proses pemisahan kalimat..
Skenario Utama	
Aksi Aktor	Reaksi Sistem
	1. Sistem melakukan proses *preprocessing case folding*
	2. Data masukan berhasil melaluai proses *preprocessing case folding*.
	3. Menampilkan hasil *preprocessing case folding.*
Skenario Alternatif	
Aksi Aktor	Reaksi Sistem
	1. Data masukan gagal melalui proses *preprocessing case folding*.

Tabel 3. 51 *Use Case Scenario Filtering*

Nama *Use Case*	*Filtering*

Tujuan	Menghilangkan kata selain “a” sampai “z” dan hanya menerima spasi.
Deskripsi	Fungsionalitas ini digunakan sistem untuk menghilangkan tanda baca pada kalimat selain spasi dan karakter “a”-“z”..
Aktor	*User*
Related Case	*Preprocessing.*
Kondisi Sukses	Data masukan berhasil melalui proses *preprocessing filtering*.
Kondisi Gagal	Data masukan gagal melalui proses *preprocessing filtering*.
Include Case	-
Trigger	Terdapat data hasil proses *case folding.*.
Skenario Utama	
Aksi Aktor	Reaksi Sistem
	1. Sistem melakukan proses *preprocessing filtering.*
	2. Data masukan berhasil melaluai proses *preprocessing filtering*.
	3. Manpilkan hasil *preprocessing filering.*
Skenario Alternatif	

Aksi Aktor	Reaksi Sistem
	1. Data masukan gagal melalui proses *preprocessing filtering*.

Tabel 3. 52 *Use Case Scenario Tokenizing*

Nama *Use Case*	*Tokenizing*
Tujuan	Memisahkan data kalimat menjadi data perkata.
Deskripsi	Fungsionalitas ini digunakan sistem untuk memecahkan data kalimat dan memisahkannya menjadi data perkata.
Aktor	*User*
Related Case	*Preprocessing.*
Kondisi Sukses	Data masukan berhasil melalui proses *preprocessing tokenizing*.
Kondisi Gagal	Data masukan gagal melalui proses *preprocessing tokenizing*.
Include Case	-
Trigger	Terdapat data hasil proses *case folding*..
Skenario Utama	
Aksi Aktor	Reaksi Sistem
	1. Sistem melakukan proses *preprocessing tokenizing.*

	2. Data masukan berhasil melaluai proses *preprocessing tokenizing*.
	3. Menampilkan hasil *preprocessing tokenizing.*
Skenario Alternatif	
Aksi Aktor	Reaksi Sistem
	1. Data masukan gagal melalui proses *preprocessing tokenizing*.

Tabel 3. 53 *Use Case Scenario Stopword Removal*

Nama *Use Case*	*Stopword Removal*
Tujuan	Menghilangkan kata yang terdapat dalam kamus *stopword.*
Deskripsi	Fungsionalitas ini digunakan sistem untuk menghilangkan kata-kata yang masuk kedalam daftar *stopword.*
Aktor	*User*
Related Case	*Preprocessing.*
Kondisi Sukses	Data masukan berhasil melalui proses *preprocessing stopword removal*.
Kondisi Gagal	Data masukan gagal melalui proses *preprocessing stopword removal*.
Include Case	-

Trigger	Terdapat data hasil proses *case tokenizing*..
Skenario Utama	
Aksi Aktor	Reaksi Sistem
	1. Sistem melakukan proses *preprocessing stropword removal.*
	2. Data masukan berhasil melaluai proses *preprocessing stopword removal*.
	3. Menampilkan hasil *preprocessing stopword removal.*
Skenario Alternatif	
Aksi Aktor	Reaksi Sistem
	1. Data masukan gagal melalui proses *preprocessing stopword removal*.

Tabel 3. 54 *Use Case Scenario Stemming*

Nama *Use Case*	*Stemming*
Tujuan	Mengubah kata berimbuhan menjadi kata dasar.
Deskripsi	Fungsionalitas ini digunakan sistem untuk mengubah kata yang memiliki imbuhan kedalam kata dasarnya.
Aktor	*User*

Related Case	*Preprocessing.*
Kondisi Sukses	Data masukan berhasil melalui proses *preprocessing stemming*.
Kondisi Gagal	Data masukan gagal melalui proses *preprocessing stemming*.
Include Case	-
Trigger	Terdapat data hasil proses *stopword removal.*.
Skenario Utama	
Aksi Aktor	Reaksi Sistem
	1. Sistem melakukan proses *preprocessing stemming.*
	2. Data masukan berhasil melaluai proses *preprocessing stemming*.
	3. Menampilkan hasil *preprocessing stemming.*
Skenario Alternatif	
Aksi Aktor	Reaksi Sistem
	1. Data masukan gagal melalui proses *preprocessing stemming*.

Tabel 3. 55 *Use Case Scenario* Pembobotan TF-IDF

Nama *Use Case*	Pembobotan TF-IDF
Tujuan	Memberii nilai bobot pada tiap kata.

Deskripsi	Fungsionalitas ini digunakan sistem untuk melakukan perhitungan pembobotan TF-IDF.
Aktor	*User*
Related Case	-.
Kondisi Sukses	Data hasil *preprocessing* berhasil diberi nilai bobot tf-idf.
Kondisi Gagal	Data hasil *preprocessing* gagal diberi nilai bobot tf-idf.
Include Case	-
Trigger	Terdapat data hasil proses *preprocessing* dan menekan tombol menu pembobotan tf-idf.
Skenario Utama	
Aksi Aktor	Reaksi Sistem
	1. Sistem melakukan proses pebobotan tf-idf.
	2. Data masukan berhasil melaluai proses pembobotan tf-idf.
	3. Menampilkan hasil pembobotan tf-idf.
Skenario Alternatif	
Aksi Aktor	Reaksi Sistem

	1. Data masukan gagal melalui proses pembobotan tf-idf.

Tabel 3. 56 *Use Case Scenario* Pembobotan *Cosine Similarity*

Nama *Use Case*	Pembobotan *Cosine Simility*
Tujuan	Mendapatkan nilai bobot *query relevance* dan *similarity* kalimat.
Deskripsi	Fungsionalitas ini digunakan sistem untuk melakukan perhitungan *Cosine Similarity.*
Aktor	*User*
Related Case	-.
Kondisi Sukses	Data hasil *pembobotan tf-idf* berhasil didapatkan nilai bobot *query relevance* dan *similarity* kaliamt.
Kondisi Gagal	Data hasil *pembobotan tf-idf* gagal didapatkan nilai bobot *query relevance* dan *similarity* kaliamt
Include Case	*query relevance*, *similarity* kalimat.
Trigger	Terdapat data hasil proses pembobotan tf-idf dan menekan tombol menu perhitungan *cosine similarity*.
Skenario Utama	
Aksi Aktor	Reaksi Sistem

	1. Sistem melakukan perhitungan *cosine similarity.*
	2. Data hasil pembobotan tf-idf berhasil dihitung dan didapatkan nilai bobot *query relevance* dan *similarity kalimat.*
	3. Menampilkan hasil perhitungan *cosine similarity.*
Skenario Alternatif	
Aksi Aktor	Reaksi Sistem
	1. Data hasil pembobotan tf-idf gagal dihitung untuk mendapatkan nilai bobot *query relevance* dan *similarity kalimat..*

Tabel 3. 57 *Use Case Scenario Similarity* Kalimat

Nama *Use Case*	*Simility* Kalimat
Tujuan	Mendapat nilai bobot *similarity* kalimat.
Deskripsi	Fungsionalitas ini digunakan sistem untuk melakukan perhitungan *Similarity* kalimat.
Aktor	*User*
Related Case	Pembobotan *Cosine Similarity*

Kondisi Sukses	Data hasil pemobotan tf-idf berhasil dihitung dan didapatkan nilai bobot *similarity* kalimat.
Kondisi Gagal	Data hasil pemobotan tf-idf gagal dihitung dan didapatkan nilai bobot *similarity* kalimat.
Include Case	-
Trigger	Terdapat data hasil pembobotan tf-idf.
Skenario Utama	
Aksi Aktor	Reaksi Sistem
	1. Sistem melakukan perhitungan *similarity* kalimat.
	2. Data hasil pembobotan tf-idf berhasil dihitung dan didapatkan nilai bobot *similarity* kalimat.
	3. Menampilkah hasil perhitungan *similarity* kalimat.
Skenario Alternatif	
Aksi Aktor	Reaksi Sistem
	1. Data hasil pembobotan tf-idf gagal dihitung untuk mendapatkan nilai *similarity* kalimat.

Tabel 3. 58 *Use Case Scenario* HMM *Forward*

Nama *Use Case*	HMM *Forward*
Tujuan	Mendapatkan nilai pembobotan tf-idf dan *coseine similarity*.
Deskripsi	Fungsionalitas ini digunakan sistem untuk melakukan input nilai kenotasi HMM Forward.
Aktor	*User*
Related Case	-.
Kondisi Sukses	Data hasil *pembobotan tf-idf* dan *cosine similarity* berhasil dimasukkan kenotasi hmm.
Kondisi Gagal	Data hasil *pembobotan tf-idf* dan *cosine similarity* gagal dimasukkan kenotasi hmm.
Include Case	*query relevance, similarity* kalimat, probabilitas *forward*.
Trigger	Terdapat data hasil proses pembobotan tf-idf, pembobotan *cosine similarity* dan menekan tombol menu HMM Forward.
Skenario Utama	
Aksi Aktor	Reaksi Sistem
	1. Sistem melakukan proses pembentukan notasi HMM.

	2. Data hasil pembobotan tf-idf, nilai *query relevance*, nilai *similarity* kalimat berhasil dimasukan kenotasi HMM .
	3. Sistem melakukan perhitungan probabilitas *forward* dari notasi HMM.
	4. Data hasil probabilitas *forward* berhasil dihitung.
	5. Menampilkan hasil perhitungan HMM *forward.*
Skenario Alternatif	
Aksi Aktor	Reaksi Sistem
	1. Data hasil pembobotan tf-idf, nilai *query relevance*, nilai *similarity* kalimat gagal dimasukan kenotasi HMM dan data hasil probabilitas *forward* gagal dihitung.

Tabel 3. 59 *Use Case Scenario* Penyajian Ringkasan

Nama *Use Case*	Penyajian Ringkasan
Tujuan	Menampilkan hasil ringkasan.
Deskripsi	Fungsionalitas ini digunakan sitem untuk menampilkan hasil ringkasan.
Aktor	*User*

Related Case	-
Kondisi Sukses	Hasil ringkasan ditampilkan.
Kondisi Gagal	Hasil ringkasan gagal ditampilkan.
Include Case	-
Trigger	Menekan tombol proses.
Skenario Utama	
Aksi Aktor	Reaksi Sistem
1. Aktor menekan tombol proses.	
	2. Hasil ringkasan berhasil ditampilkan.
Skenario Alternatif	
Aksi Aktor	Reaksi Sistem
1. Aktor menekan tombol proses.	
	2. Hasil ringkasan gagal ditampilkan.

Tabel 3. 60 Use Case Scenario Evaluasi Ringkasan

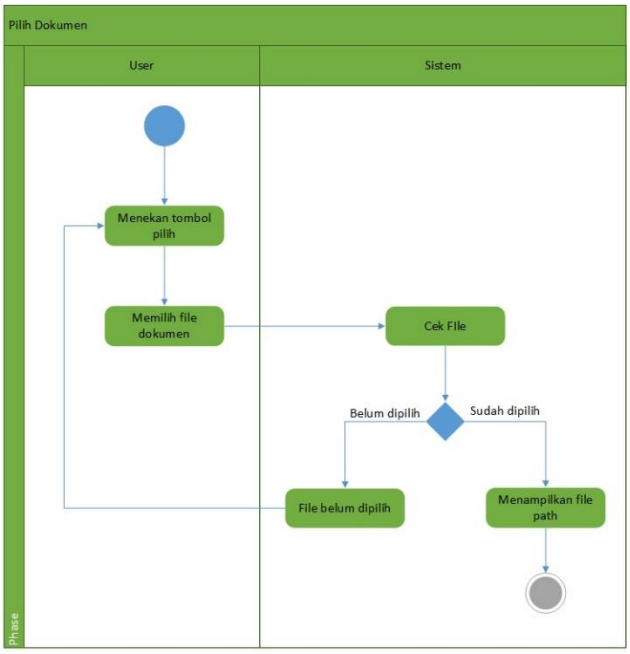
Nama *Use Case*	Evaluasi Ringkasan
Tujuan	Mendapatkan hasil evaluasi ringkasan.
Deskripsi	Fungsionalitas ini digunakan sistem untuk menampilkan perhitungan hasil evaluasi ringkasan.
Aktor	*User*

Related Case	-.
Kondisi Sukses	Berhasil mendapatkan hasil evaluasi ringkasan.
Kondisi Gagal	Gagal mendapatkan hasil evaluasi ringkasan.
Include Case	*precession, recall, f-measure.*
Trigger	Terdapat data hasil pembentukan ringkasan, menekan tombol menu evaluasi ringkasan, kemudian memilih *file* ringkasan manual dan klik Proses.
Skenario Utama	
Aksi Aktor	Reaksi Sistem
1. Aktor menekan tombol Proses.	
	2. Melakukan perhitungan nilai *precession.*
	3. Melakukan perhitungan nilai *recall.*
	4. Melakukan perhitungan nilai *f-measure.*
	5. Berhasil mendapatkan hasil evaluasi ringkasan
	6. Menampilkan hasil perhitungan evaluasi ringkasan.
Skenario Alternatif	

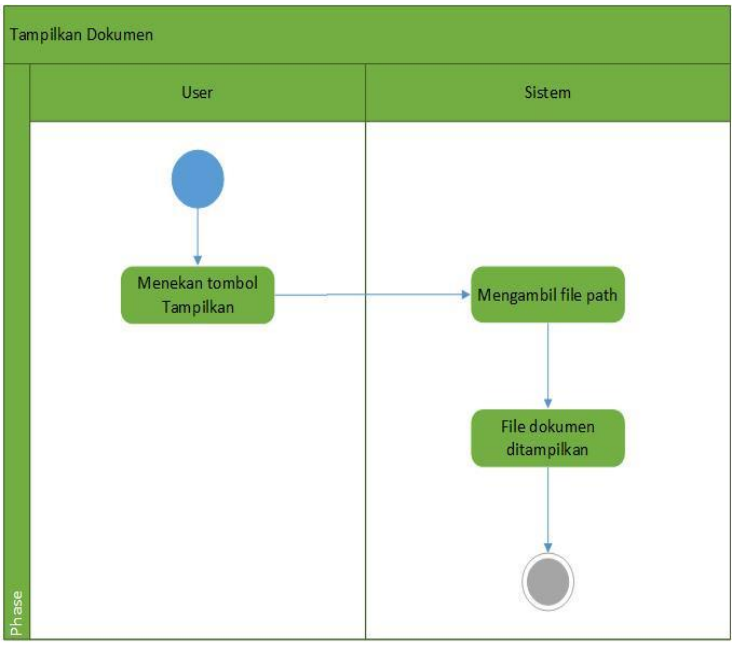
Aksi Aktor	Reaksi Sistem
1. Aktor menekan tombol Proses.	
	2. Gagal mendapatkan hasil evaluasi ringkasan

3.6.3 Activity Diagram

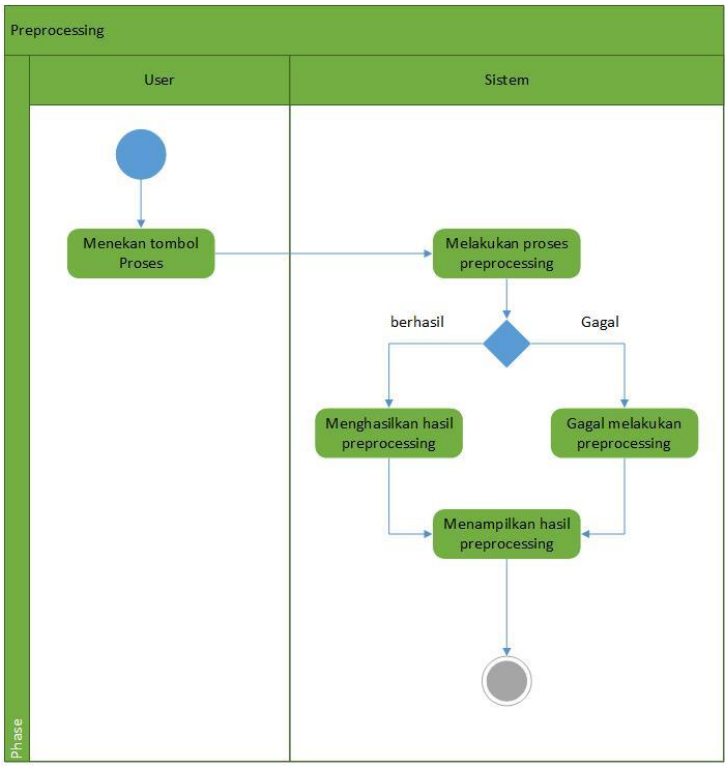
Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang dirancang, menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum, menggambarkan masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana alir berakhir. Adapun *activity diagram* dari masing-masing *use case scenario.*



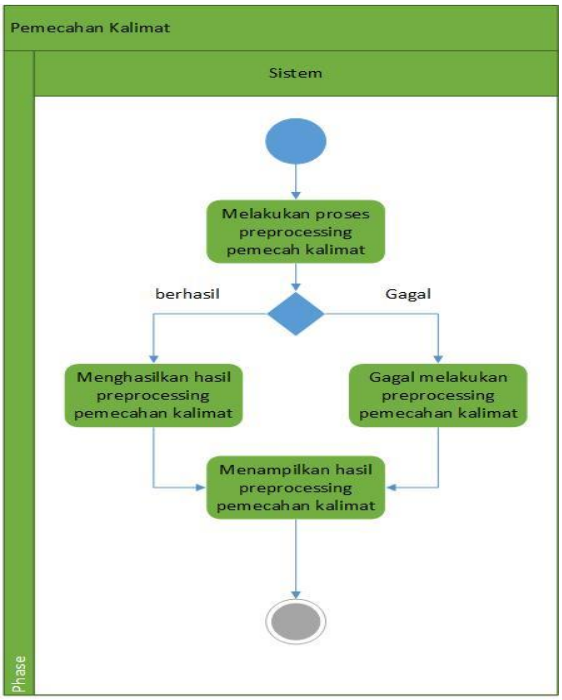
Gambar 3. 7 *Activity Diagram* Pilih Dokumen



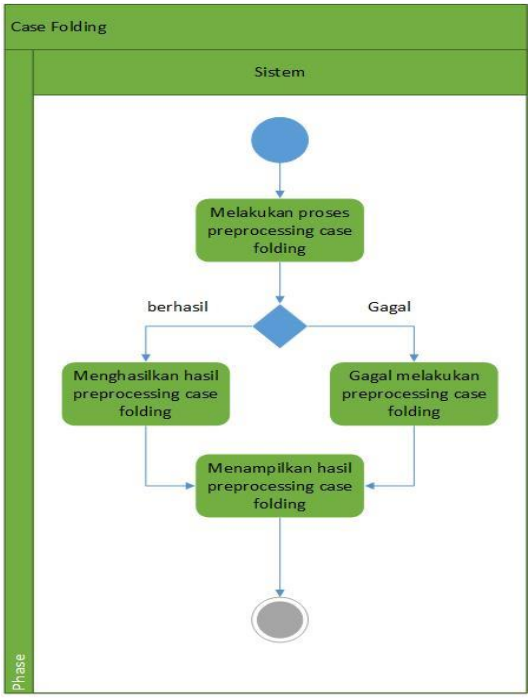
Gambar 3. 8 *Activity Diagram* Tampilkan Dokumen



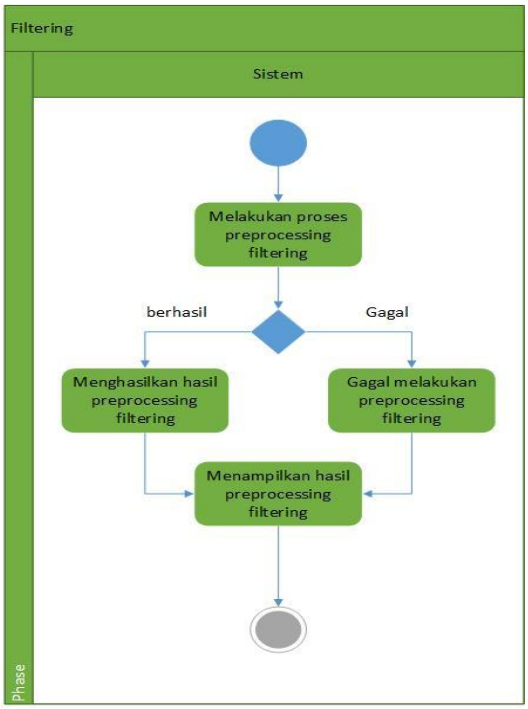
Gambar 3. 9 *Activity Diagram Preprocessing*



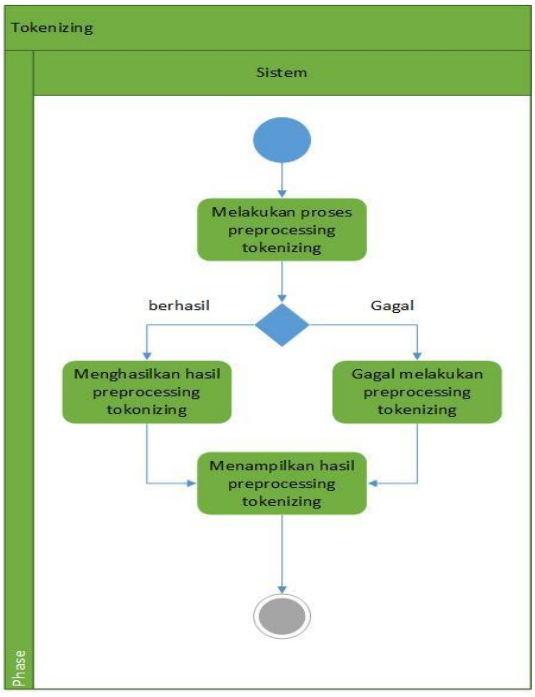
Gambar 3. 10 *Activity Diagram* Pemecah Kalimat



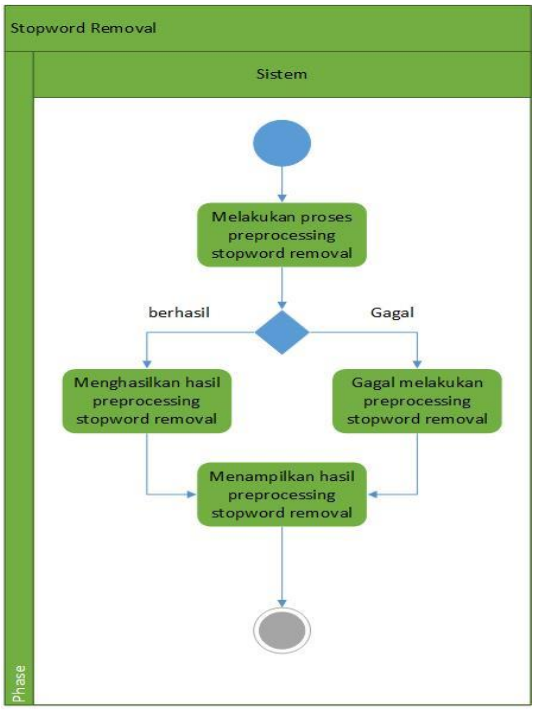
Gambar 3. 11 *Activity Diagram Case Folding*



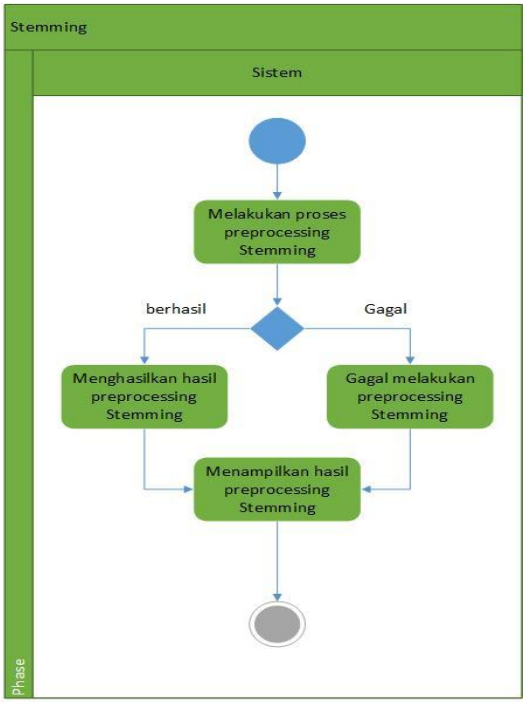
Gambar 3. 12 ***Activity Diagram Filtering***



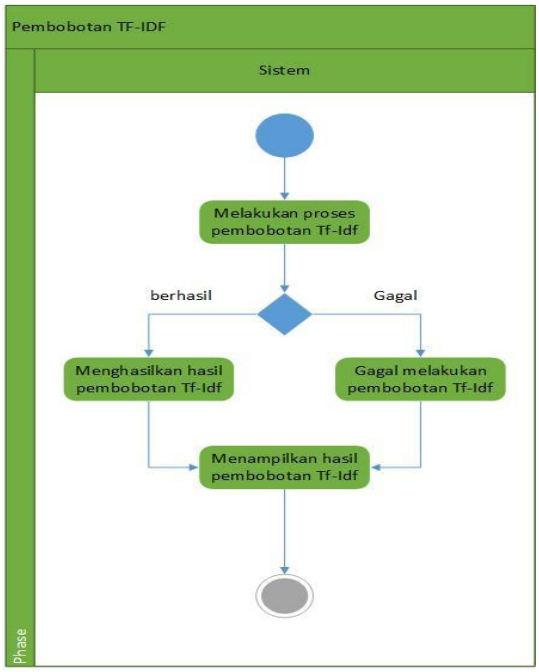
Gambar 3. 13 ***Activity Diagram Tokenizing***



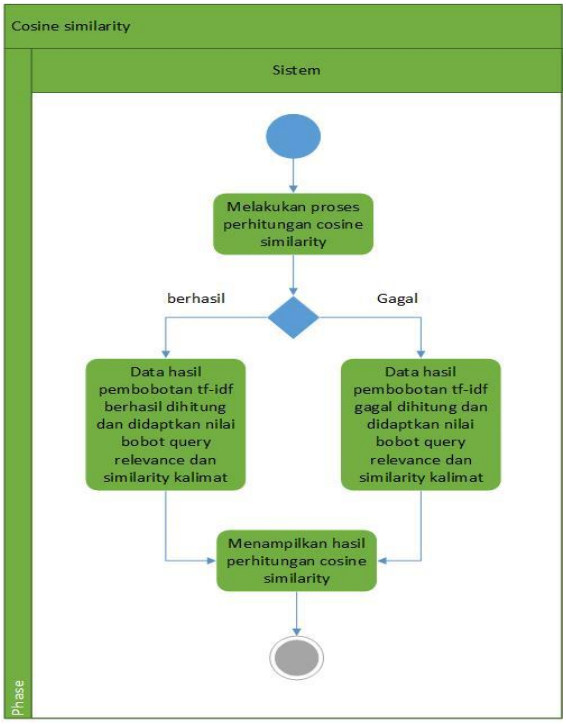
Gambar 3. 14 ***Activity Diagram Stopword Removal***



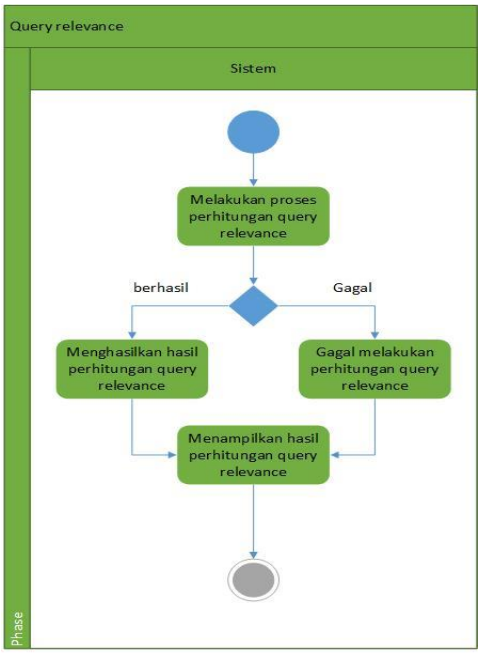
Gambar 3. 15 ***Activity Diagram Stemming***



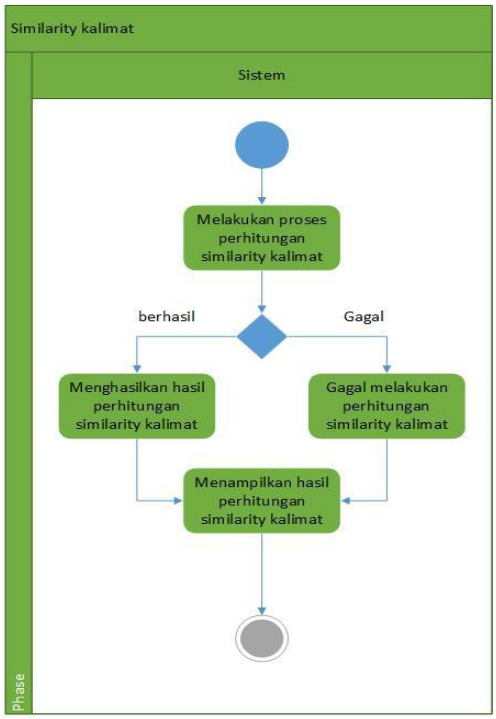
Gambar 3. 16 *Activity Diagram* Pembobotan TF-IDF



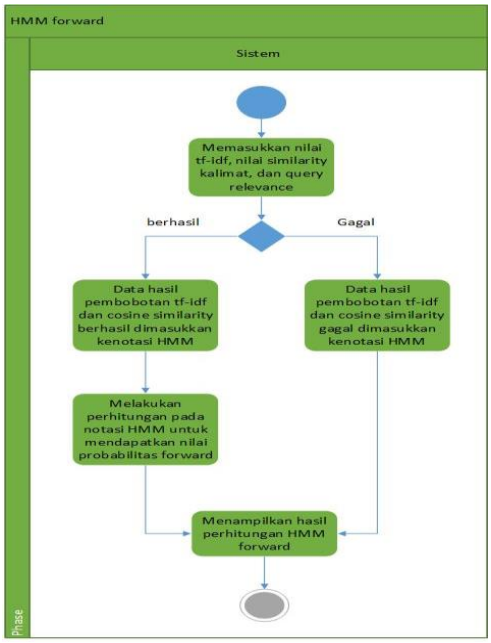
Gambar 3. 17 *Activity Diagram Pembobotan Cosine Similarity*



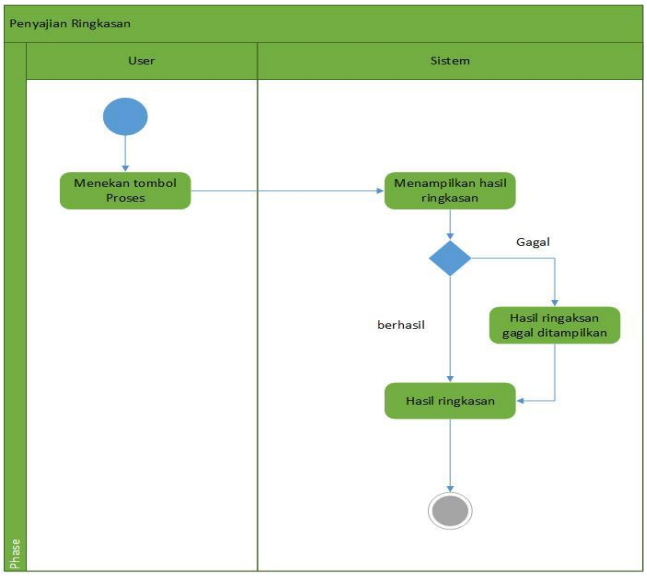
Gambar 3. 18 ***Activity Diagram Query Relevance***



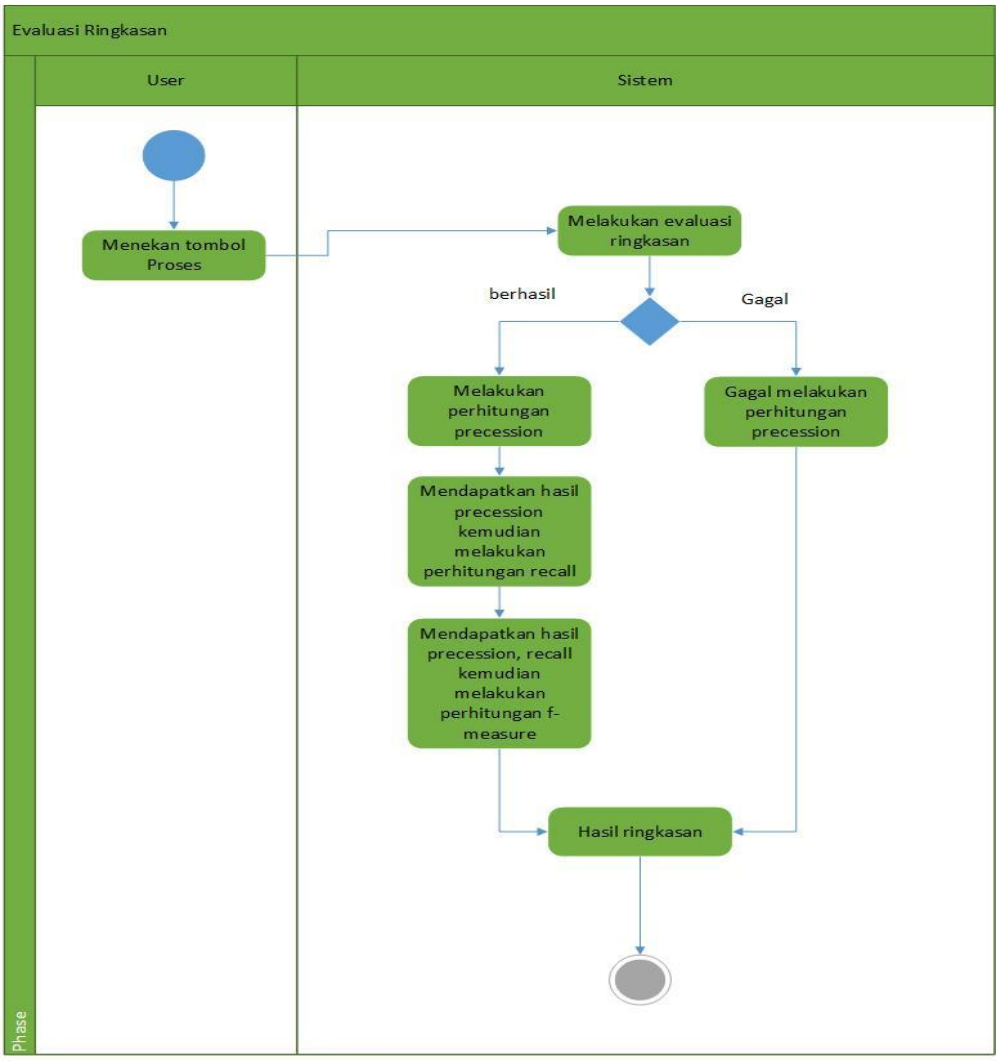
Gambar 3. 19 ***Activity Diagram Similarity*** **Kalimat**



Gambar 3. 20 *Activity Diagram HMM Forward*



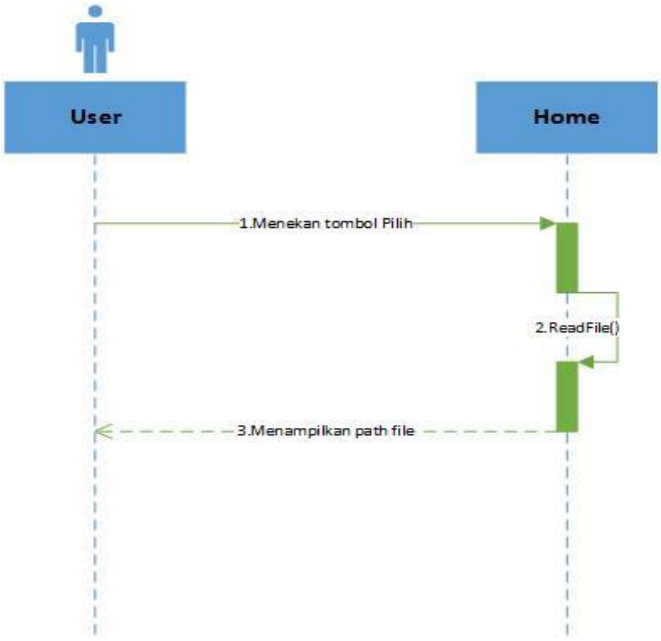
Gambar 3. 21 *Activity Diagram* Penyajian Ringkasan



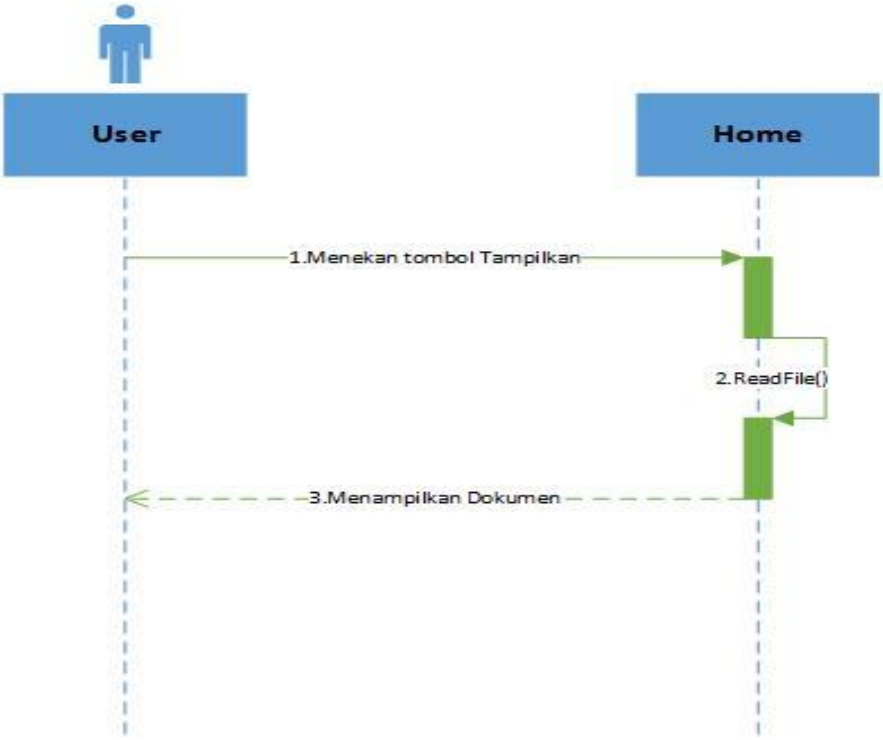
Gambar 3. 22 Activity Diagram Evaluasi Ringkasan

3.6.4 Sequence Diagram

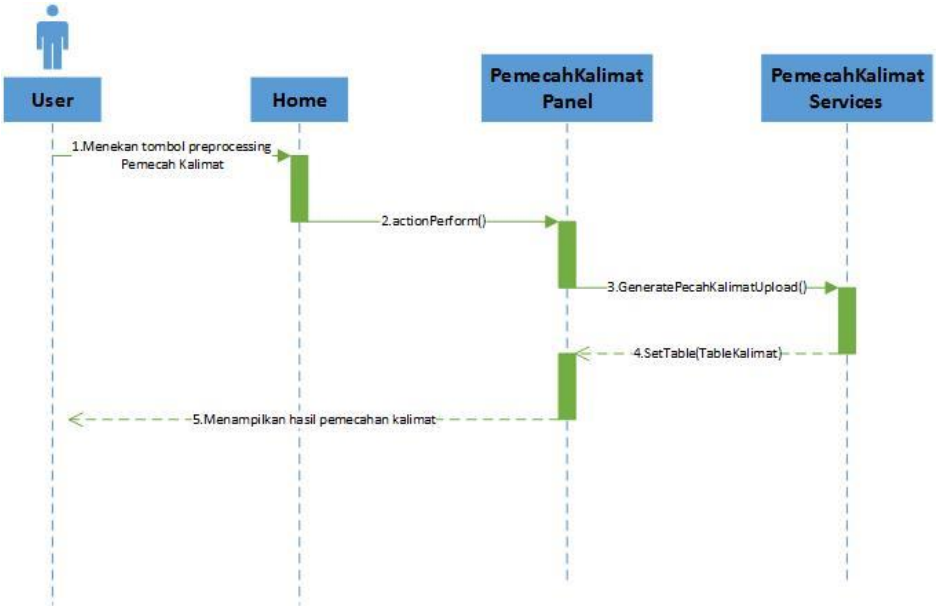
Berikut ini adalah *sequence diagram* yang terdapat pada pembangunan aplikasi yang dibangun.



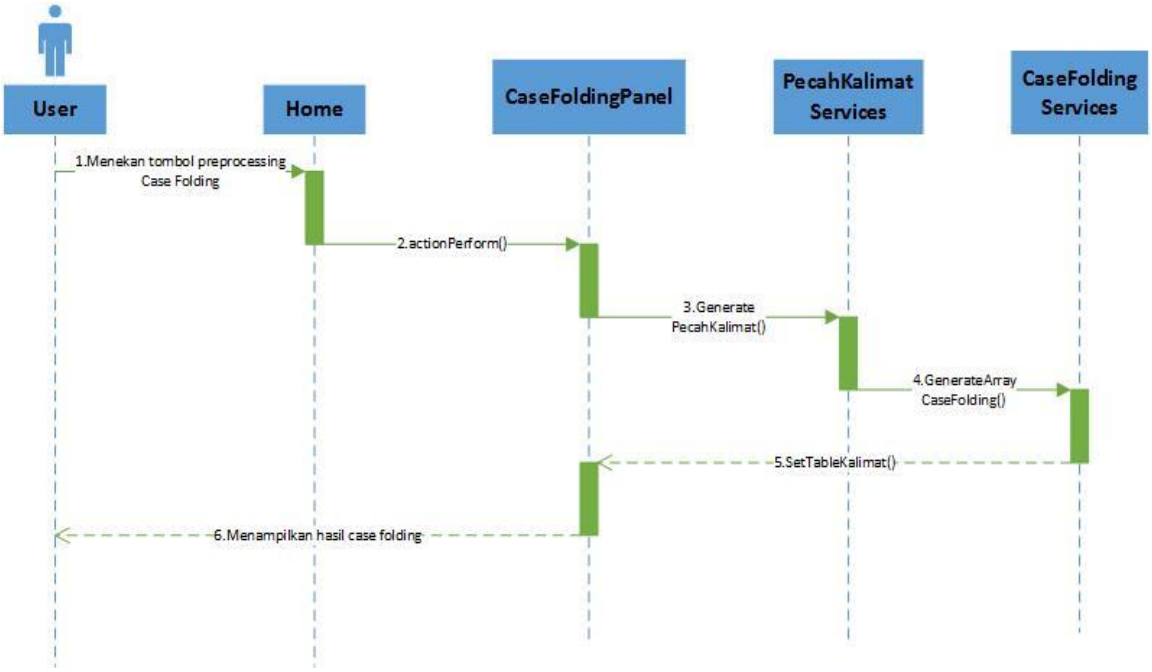
Gambar 3. 23 *Sequence Diagram* Pilih Dokumen



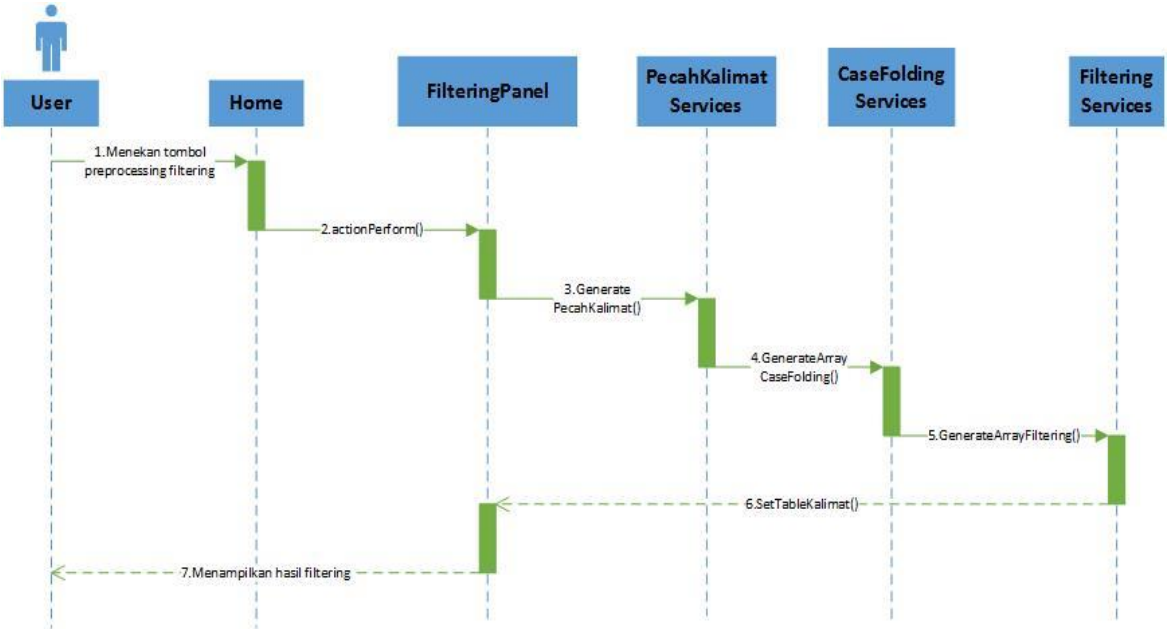
Gambar 3. 24 *Sequence Diagram* Tampilkan Dokumen



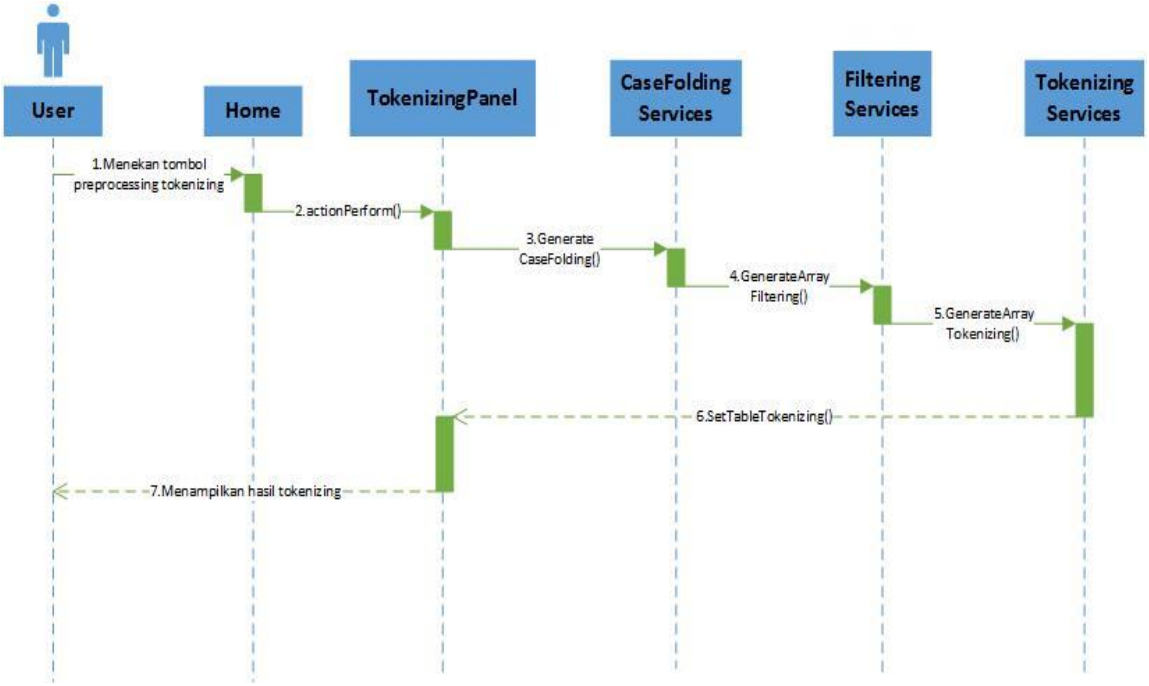
Gambar 3. 25 *Sequence Diagram Preprocessing* Pemecah Kalimat



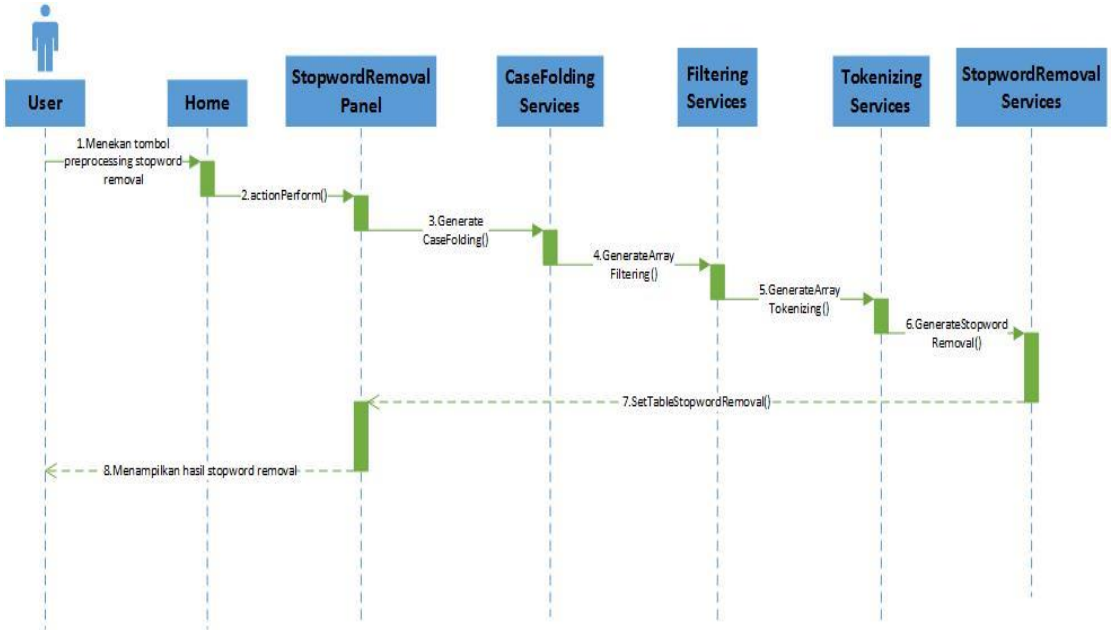
Gambar 3. 26 *Sequence Diagram Preprocessing Case Folding*



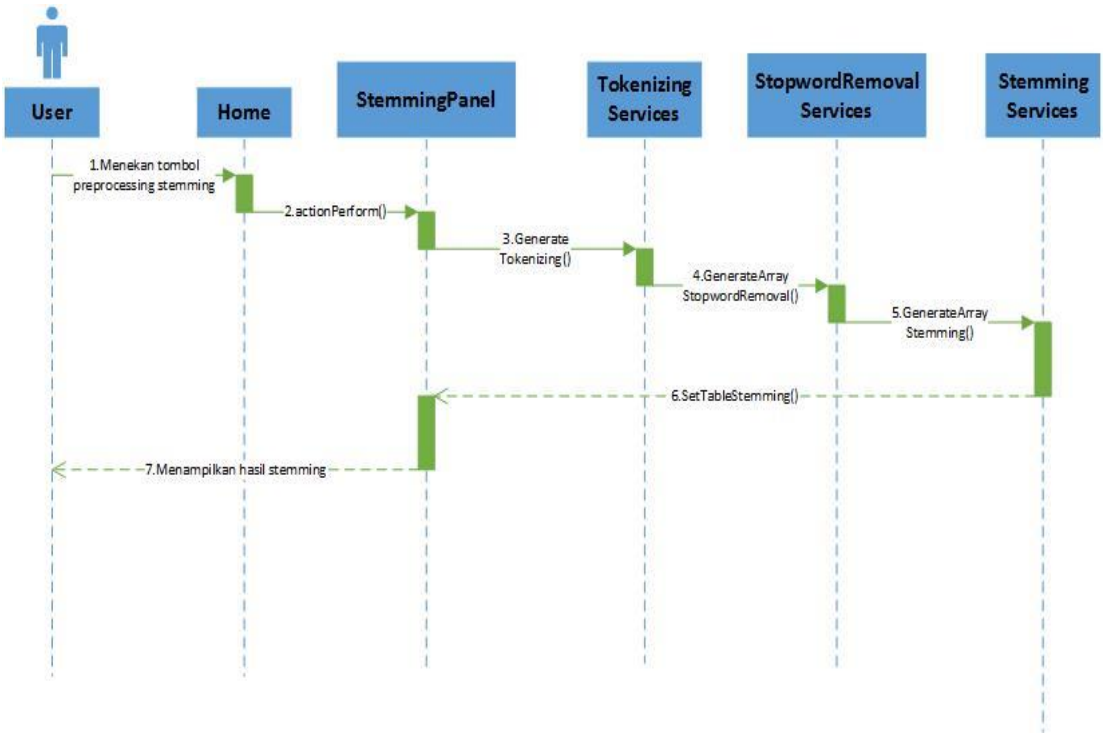
Gambar 3. 27 ***Sequence Diagram Preprocessing Filtering***



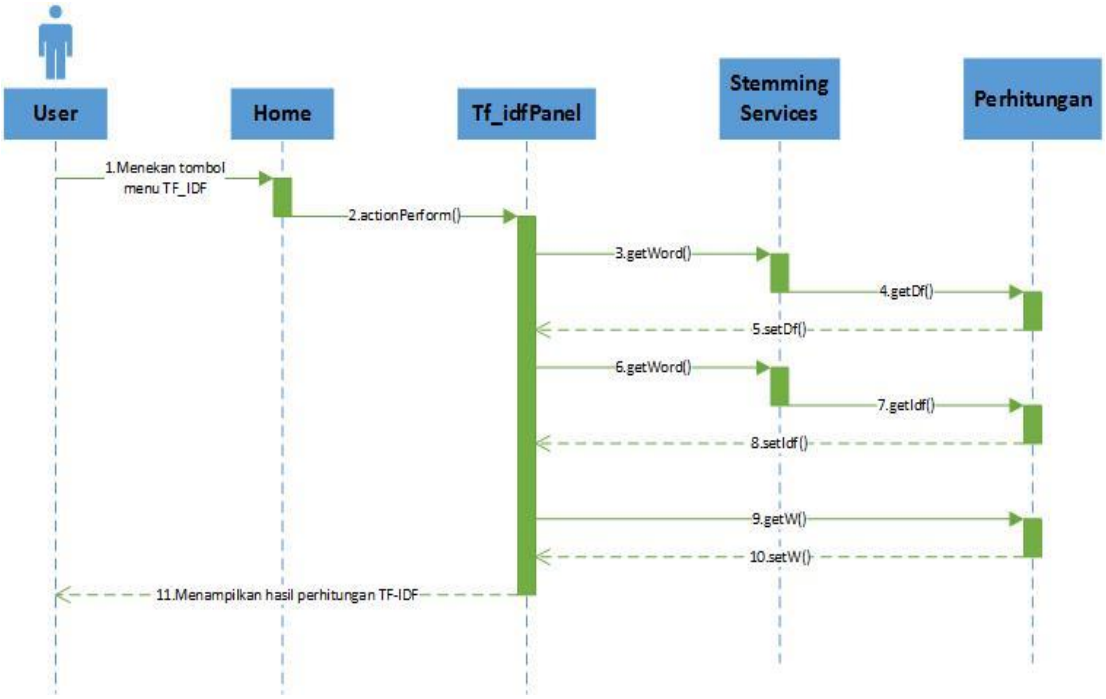
Gambar 3. 28 ***Sequence Diagram Preprocessing Tokenizing***



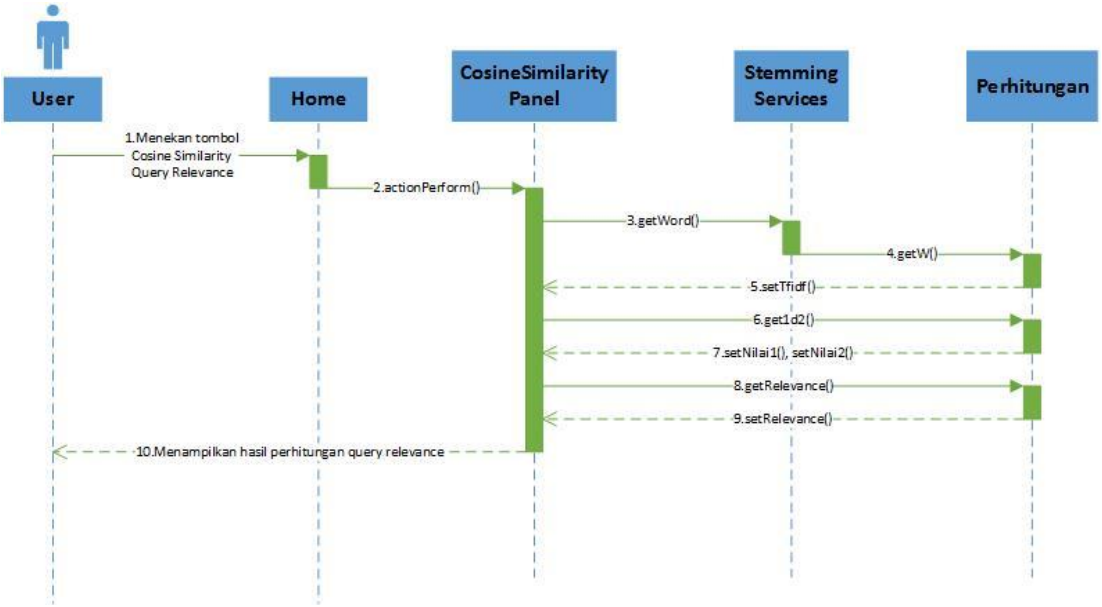
Gambar 3. 29 ***Sequence Diagram Stopword Removal***



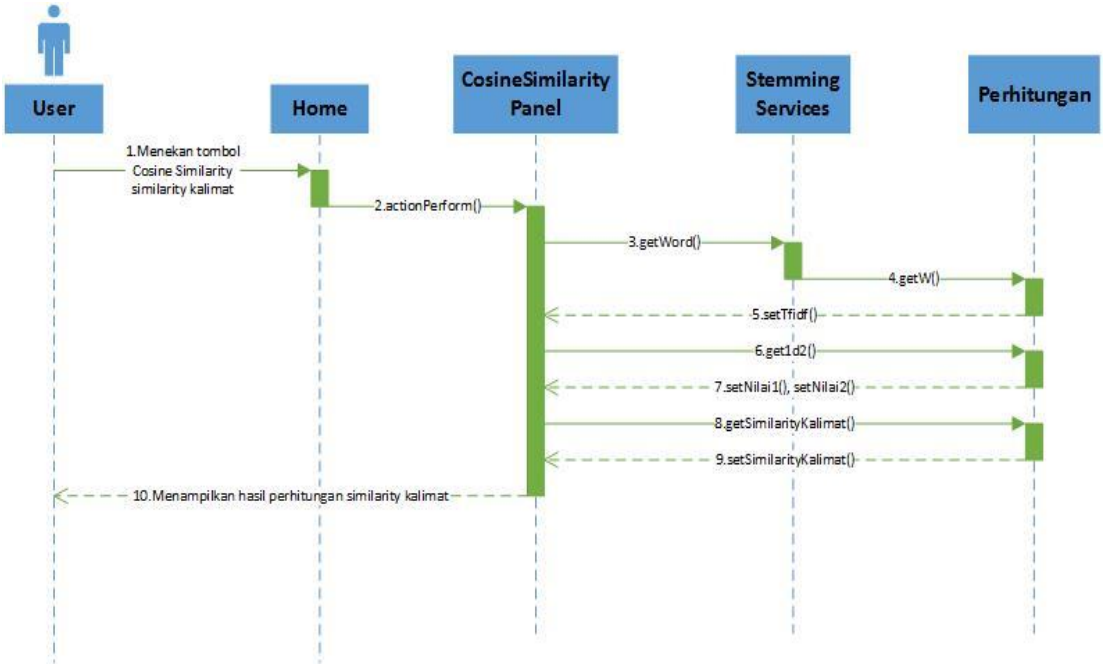
Gambar 3. 30 ***Sequence Diagram Preprocessing Stemming***



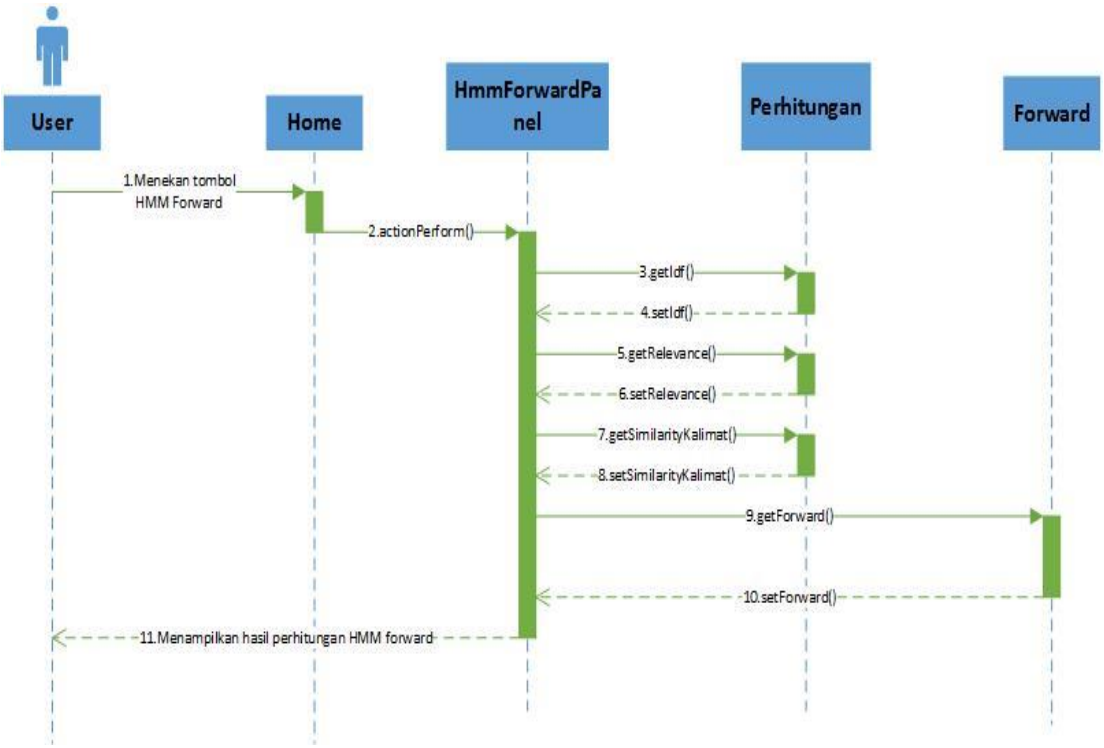
Gambar 3. 31 Sequence Diagram Preprocessing TF-IDF



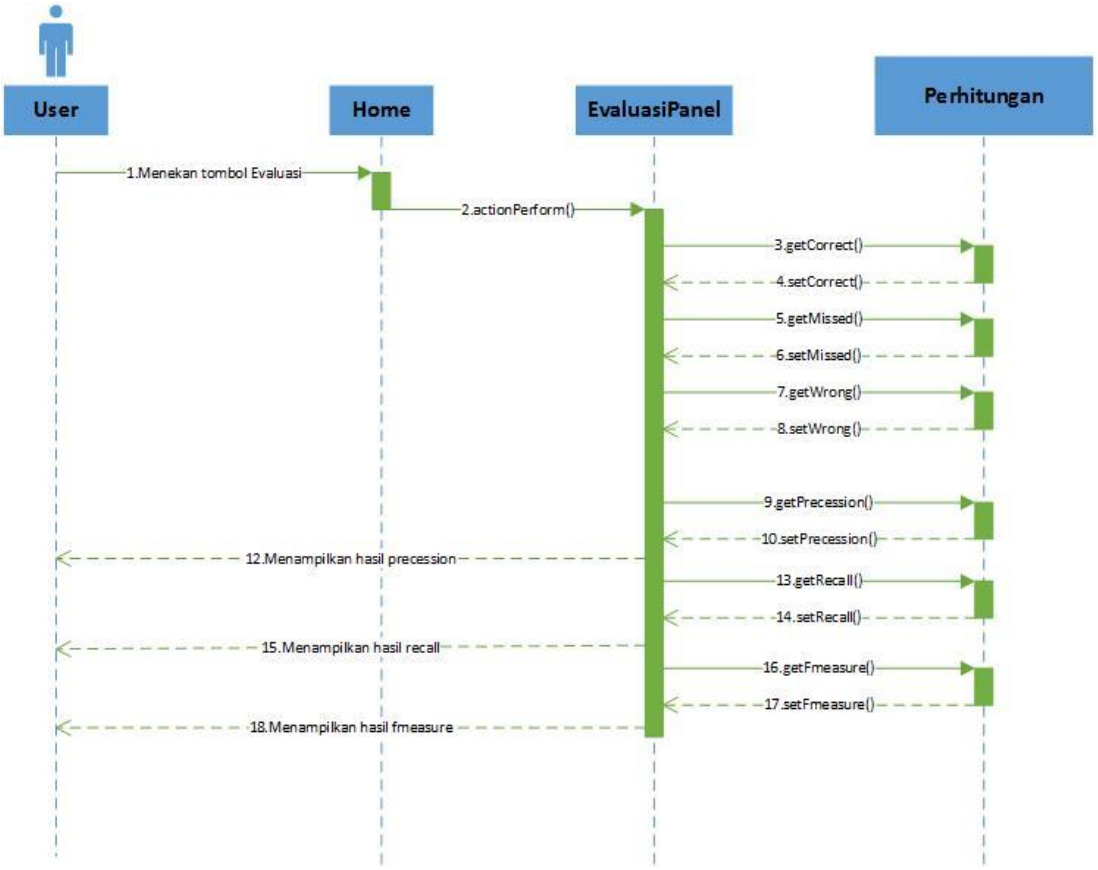
Gambar 3. 32 *Sequence Diagram Preprocessing Query Relevance*



Gambar 3. 33 *Sequence Diagram Similarity* Kalimat



Gambar 3. 34 *Sequence Diagram HMM Forward*



Gambar 3. 35 *Sequence Diagram* Evaluasi Ringkasan

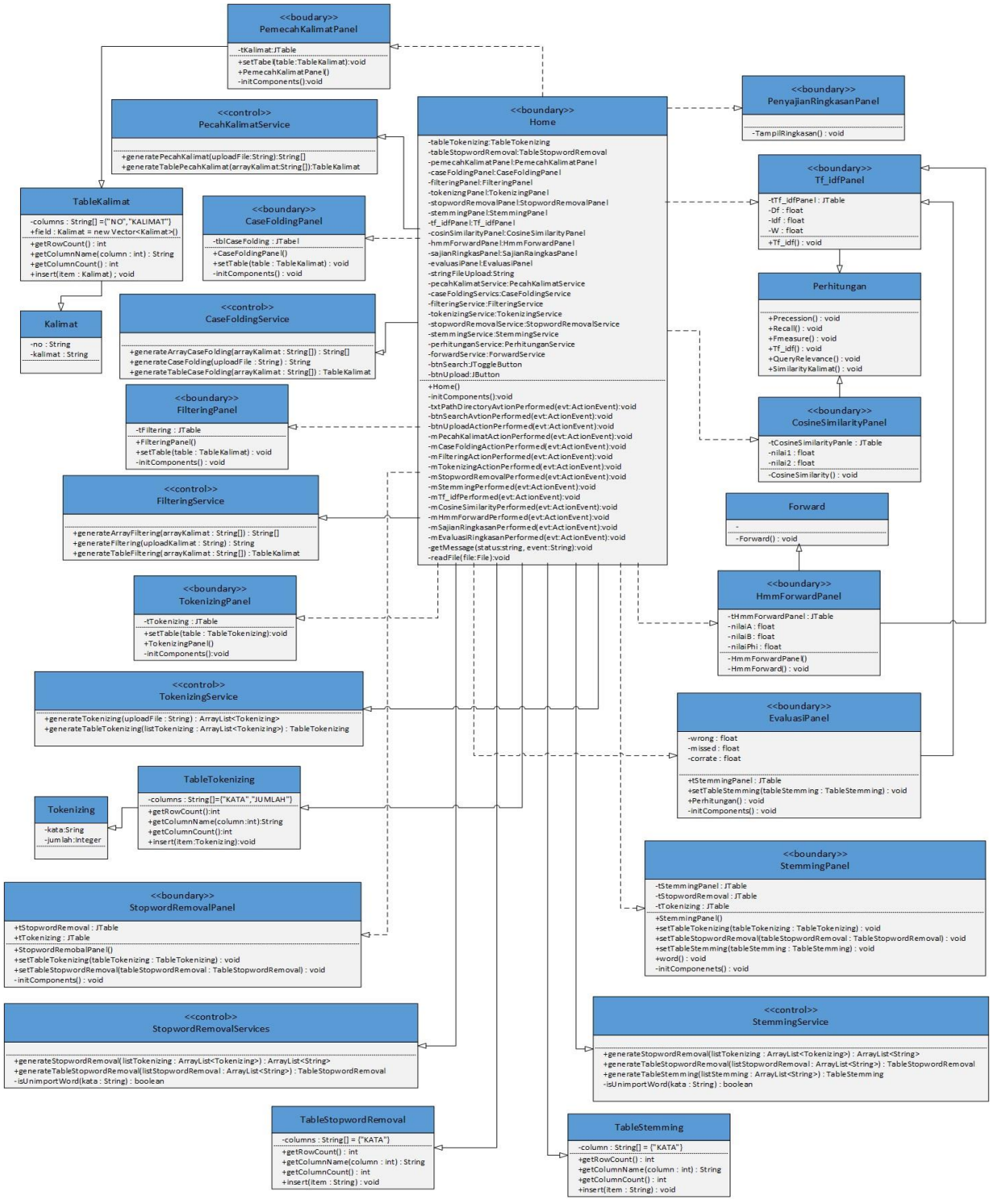
3.6.5 Class Diagram

Class diagram merupakan sebuah spesifikasi yang jika diimajinasikan akan membentuk sebuah objek dan merupakan intu dari desain berorientasi objek. Berikut adalah penjelasan berdasarkan jenis pada *class diagram* pada tabel 3.61 Jenis *Class* dan bentuk *class diagram* pada gambar 3.36 Berikut:

Tabel 3. 61 Jenis *Class*

Jenis	Nama *Class*
Model	TableKalimat
	Kalimat
	TableTokenizing
	Tokenizing
	TableStopwordRemoval
	TableStemming
	Forward
	Perhitungan
View	Home
	PemecahKalimatPanel
	CaseFoldingPanel
	FilteringPanel
	TokenizingPanel
	StopwordRemobalPanel
	TokenizingPanel
	Tf_IdfPanel
	CosineSimilarityPanel
	HmmForwardPanel
	PenyajianRingkasanPanel

Jenis	Nama *Class*
	EvaluasiPanel
Controller	PecahKalimatService
	CaseFoldingService
	FilteringService
	TokenizingService
	StopwordRemovalService
	StemmingService



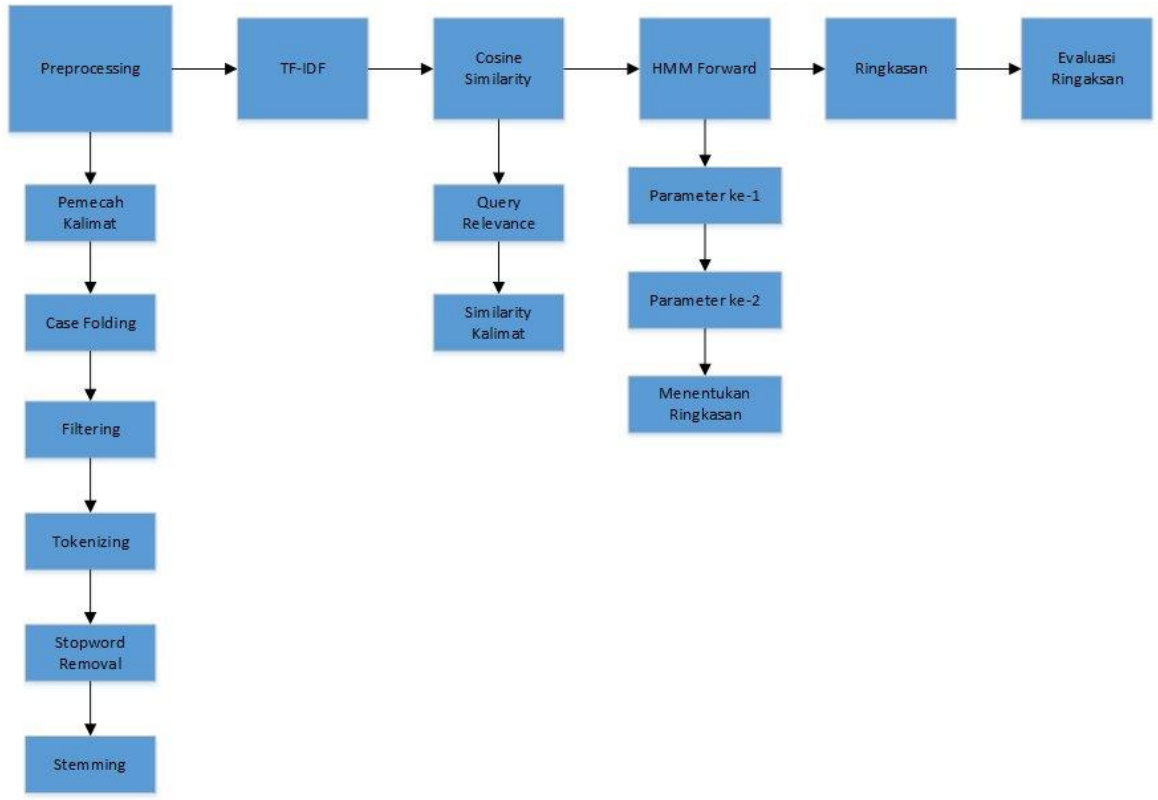
Gambar 3. 36 *Class Diagram*

3.7 Perancangan Sistem

Perancangan sistem merupakan penggambaran dan perencanaan sistem untuk mengimplementasikan hasil analisis yang dilakukan sebelumnya. Perancangan sistem yang dibuat terdiri dari perancangan basis data serta perancangan antarmuka.

3.7.1 Struktur Menu

Perancangan strukur menu merupakan gambar alur pemakaian sistem, sehingga sistem yang dibangun mudah dipahami dan mudah digunakan. Berikut ini adalah perancangan sturktur menu yang akan diterapakan pada aplikasi. Adapun perancangan struktur menu dapat dilihat pada gambar 3.37 berikut:



Gambar 3. 37 Struktur Menu

3.8 Perancangan Antarmuka

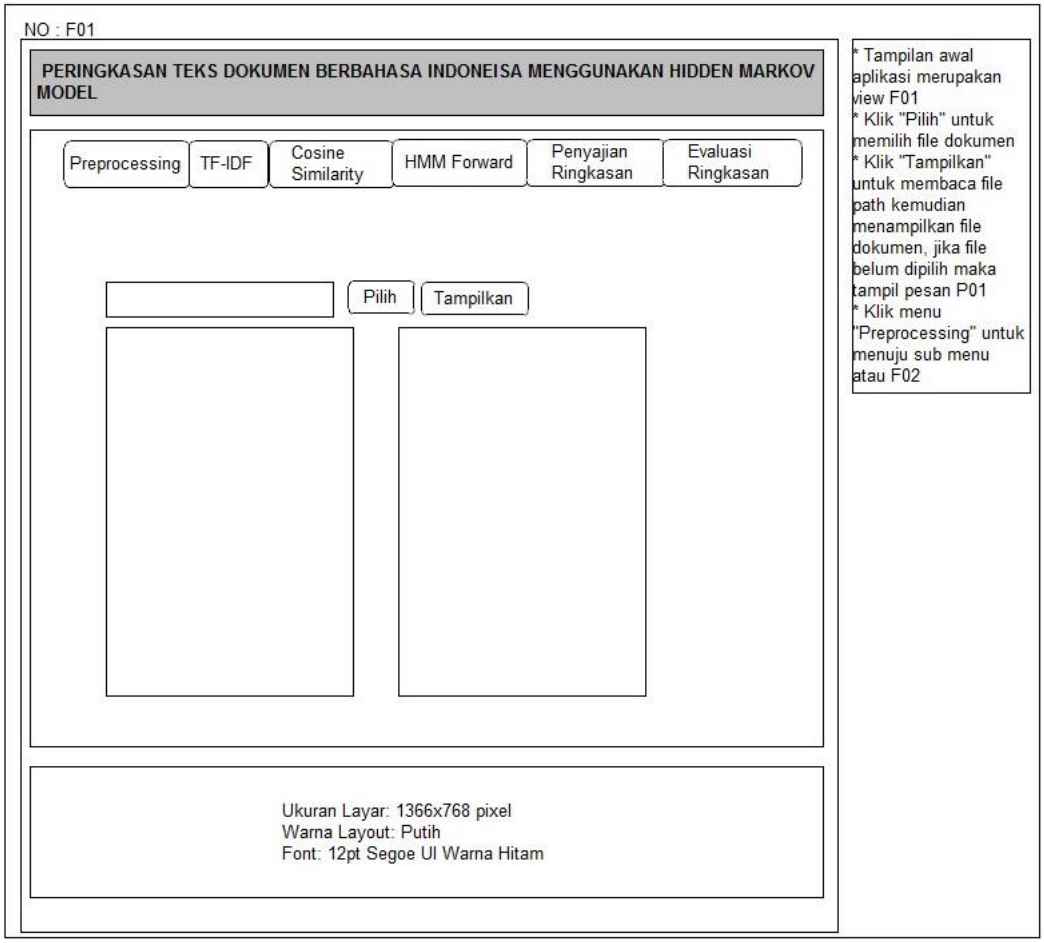
Perancangan antarmuka menjelaskan rencana tampilan yang akan digunakan pada aplikasi yang akan dibangun. Perancangan antarmuka yang ada terdiri dari perancangan *form*, perancangan pesan, dan jaringan semantic.

3.8.1 Perancangan *Form*

Perancangan antarmuka *form* merupakan rancangan dalam menentukan tata letak dari tampilan sistem yang akan dibuat. Rancangan tersebut dibuat sebagai perancangan *form* pada sistem. Berikut adalah perancangan *form* pada aplikasi yang akan dibangun.

1. F01 – *Form* Pemilihan *File*

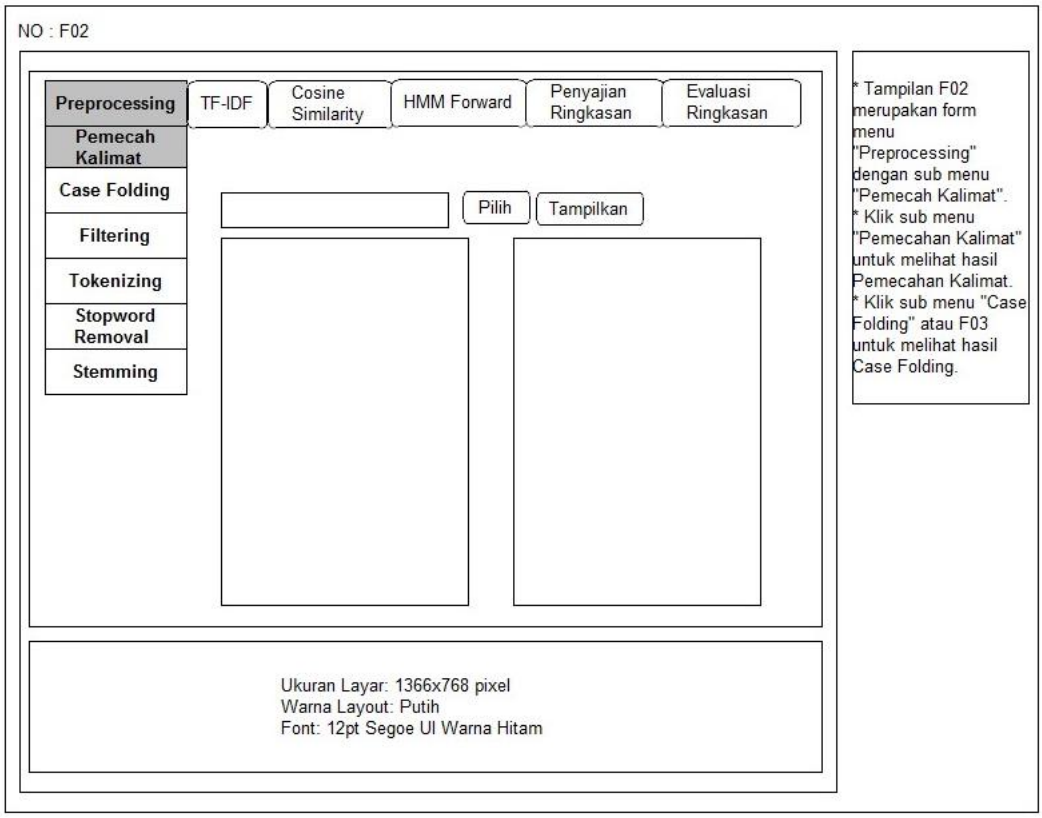
   Perancangan F01 merupakan halaman yang dijadikan halaman utama pada saat pertama kali menjalanakan program. Tampilan *default* awal program diawali tampilan pemilihan *file*. Perancangan F01 dapat dilihat pada gambar 3.38 berikut :



Gambar 3. 38 *Form* Pemilihan *File*

2. F02 – *Form* Proses *Preprocessing* Pemecahan Kalimat

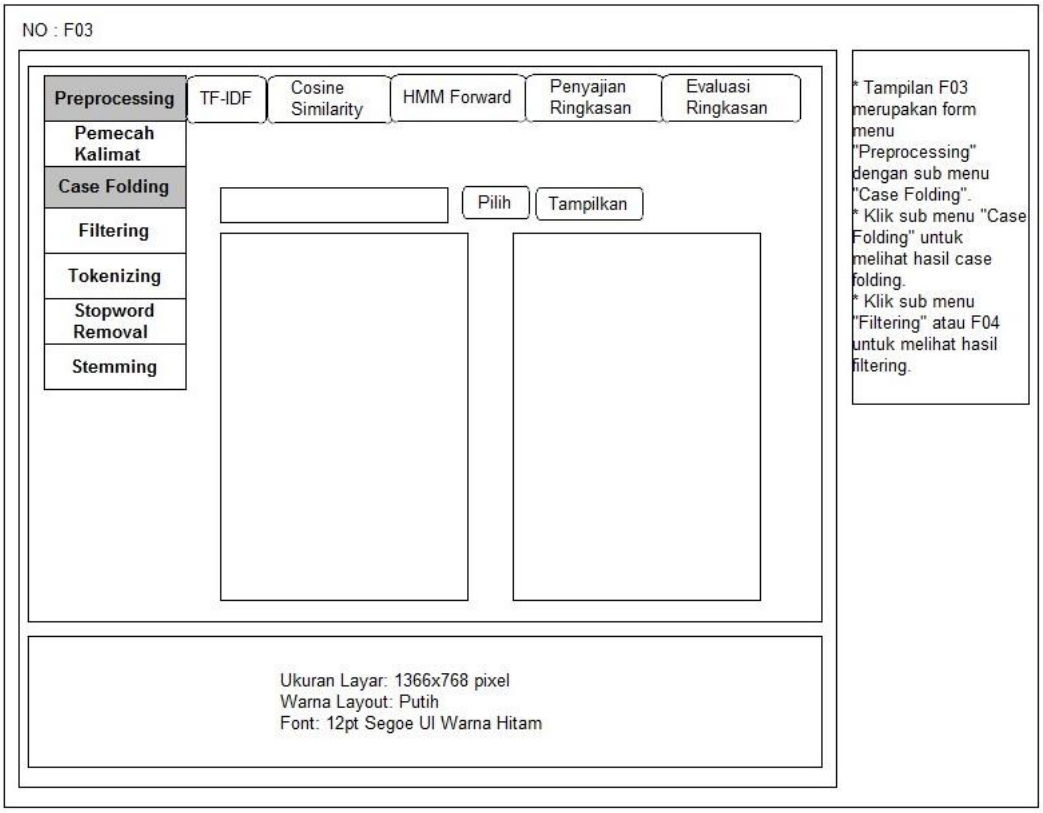
Pada perancangan *from* F02 merupakan halaman sub menu pada menu proses *preprocessing*. Perancangan F02 dapat dilihat pada gambar 3.39 berikut :



Gambar 3. 39 *Form Proses Preprocessing* Pemecahan Kalimat

3. F03 – *Form* Proses *Preprocessing Case Folding*

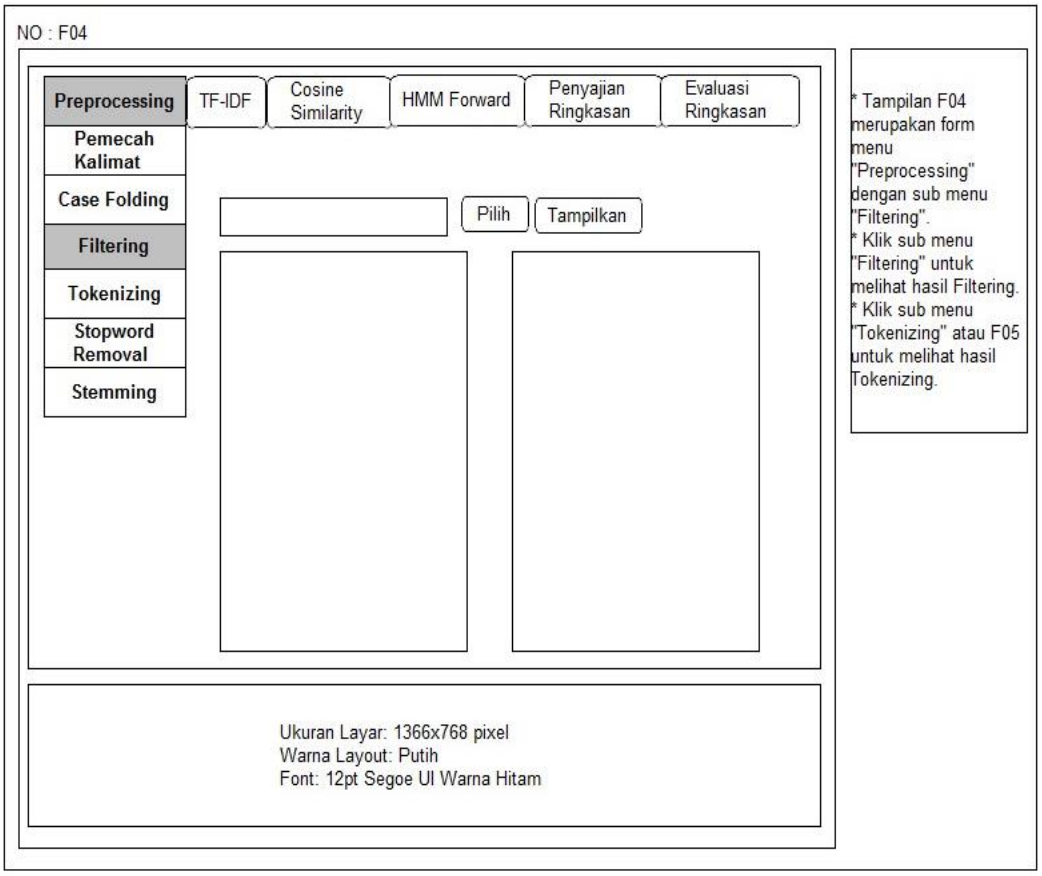
   Pada perancangan *from* F03 merupakan halaman sub menu pada menu proses *preprocessing*. Perancangan F03 dapat dilihat pada gambar 3.40 berikut :



Gambar 3. 40 *Form Proses Preprocessing Case Folding*

4. F04 – *Form* Proses *Preprocessing Filtering*

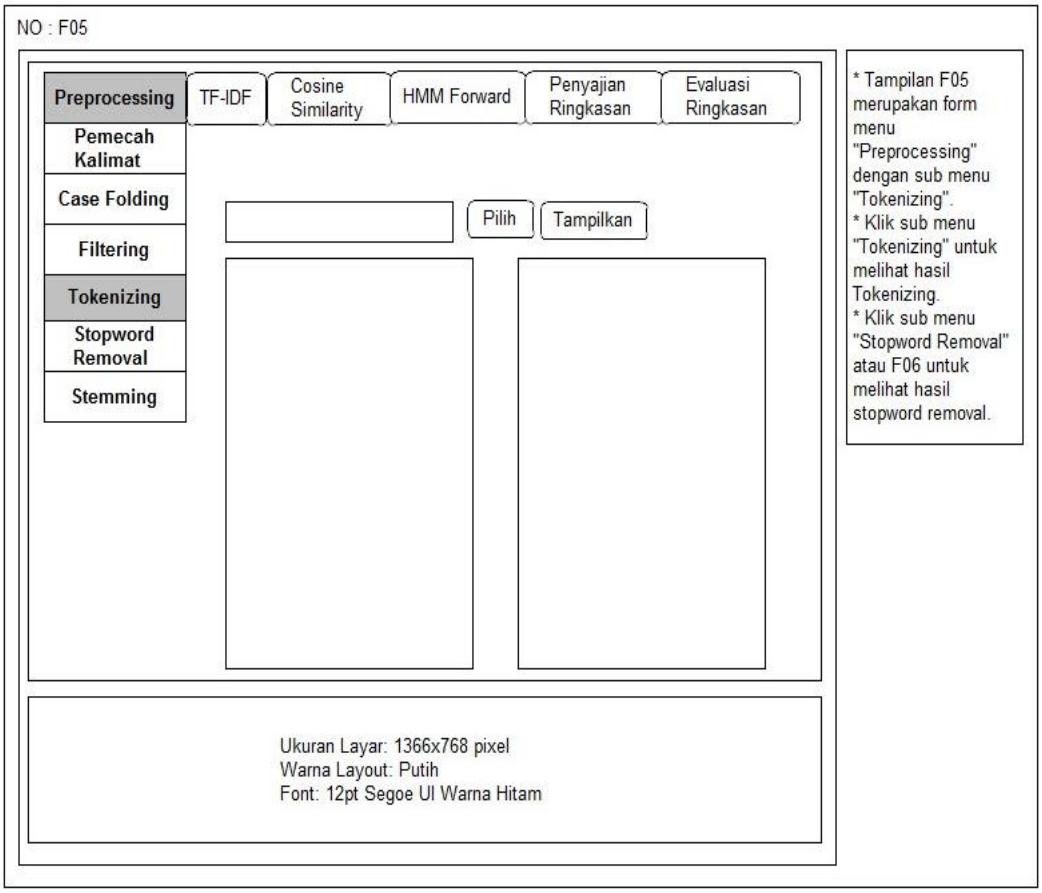
   Pada perancangan *from* F04 merupakan halaman sub menu pada menu proses *preprocessing*. Perancangan F04 dapat dilihat pada gambar 3.41 berikut :



Gambar 3. 41 *Form Proses Preprocessing Filtering*

5. F05 – *Form* Proses *Preprocessing Tokenizing*

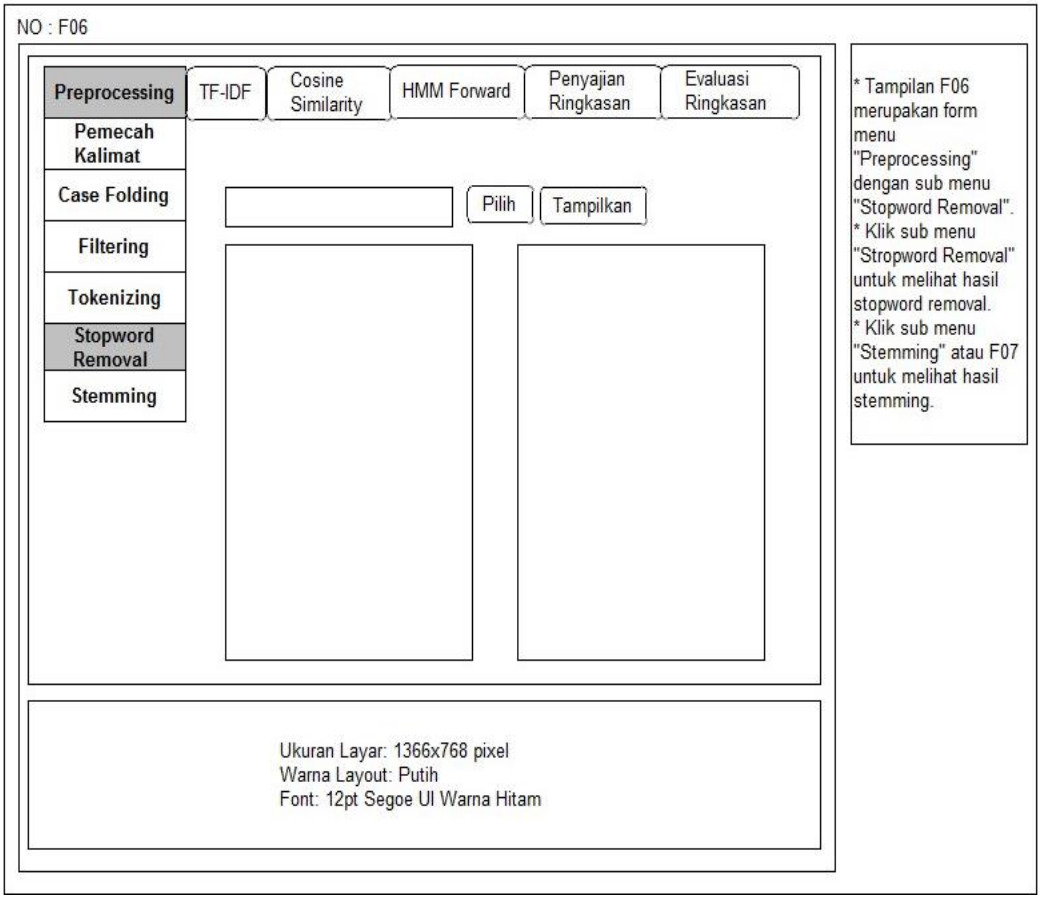
   Pada perancangan *from* F05 merupakan halaman sub menu pada menu proses *preprocessing*. Perancangan F05 dapat dilihat pada gambar 3.42 berikut :



Gambar 3. 42 *Form Proses Preprocessing Tokenizing*

6. F06 – *Form* Proses *Preprocessing Stopword Removal*

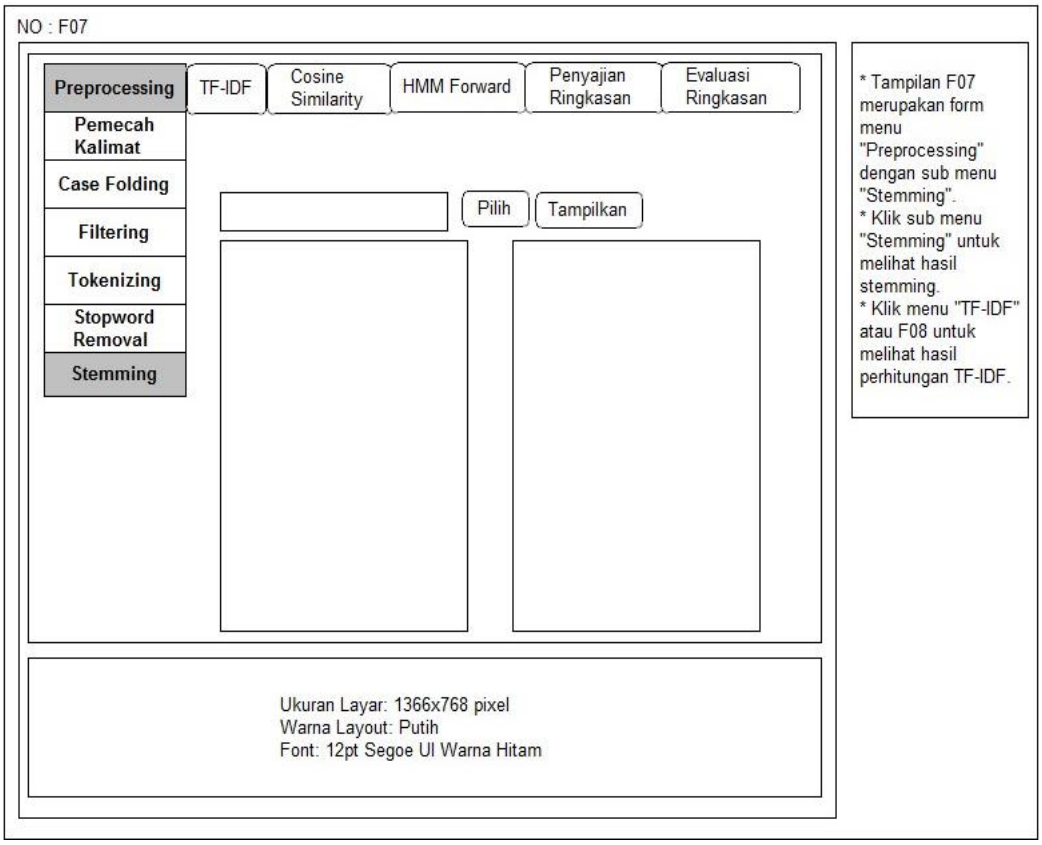
Pada perancangan *from* F06 merupakan halaman sub menu pada menu proses *preprocessing*. Perancangan F06 dapat dilihat pada gambar 3.43 berikut :



Gambar 3. 43 *Form Proses Preprocessing Stopword Removal*

7. F07 – *Form* Proses *Preprocessing Stemming*

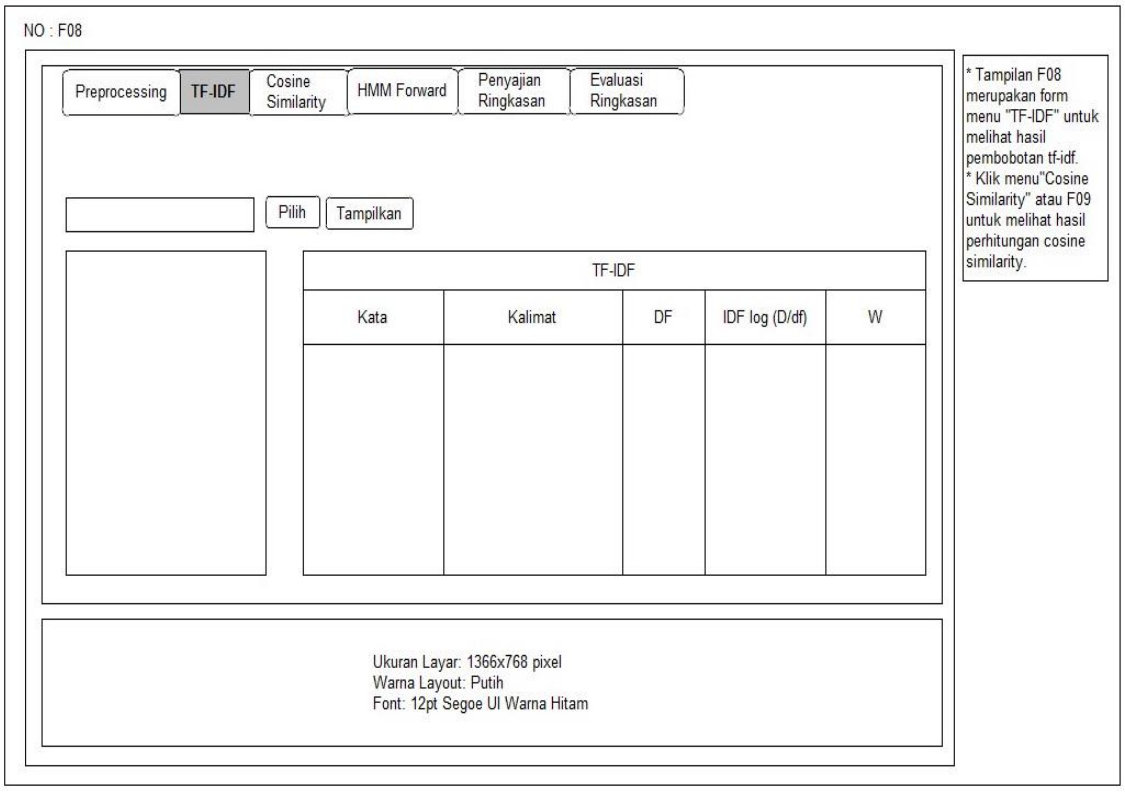
   Pada perancangan *from* F07 merupakan halaman sub menu pada menu proses *preprocessing*. Perancangan F07 dapat dilihat pada gambar 3.44 berikut:



Gambar 3. 44 *Form Proses Preprocessing Stemming*

8. F08 – *Form* Proses Perhitungan TF-IDF

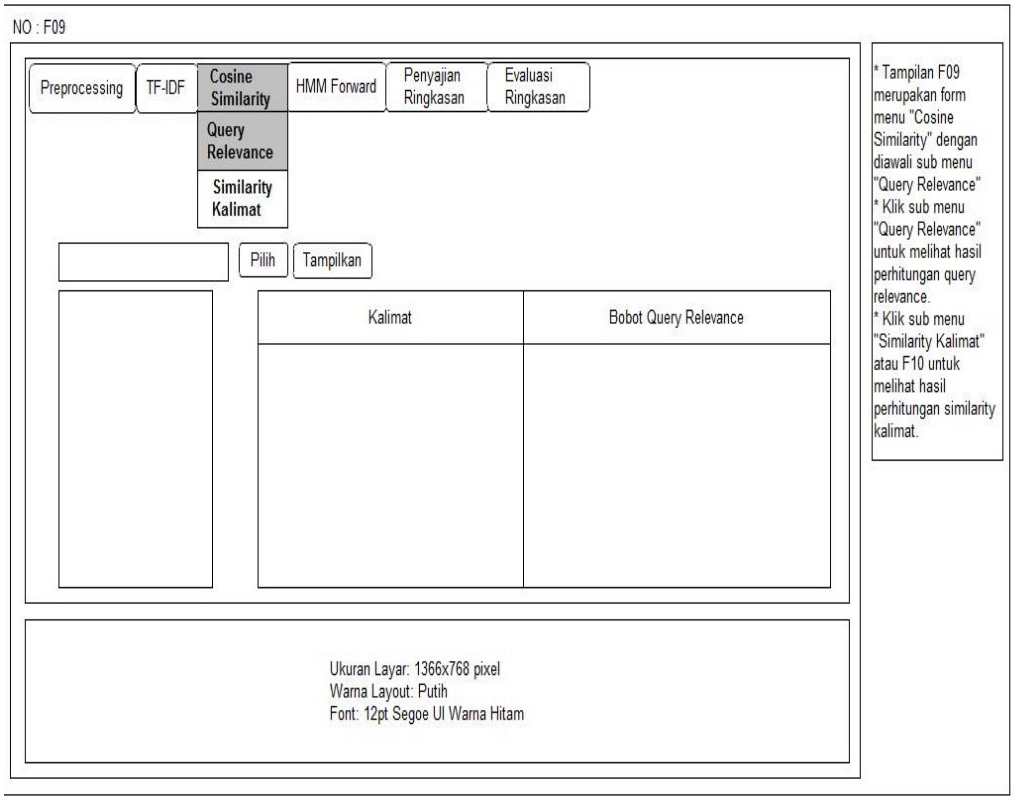
   Pada perancangan *form* F08 merupakan halaman yang digukanan unuk melihat hasil perhitungan pembobotan TF-IDF. Perancangan F08 dapat dilihat pada gambar 3.45 berikut :



Gambar 3. 45 *Form Proses* Perhitungan TF-IDF

9. F09 – *Form* Proses *Cosine Similarity Query Relevance*

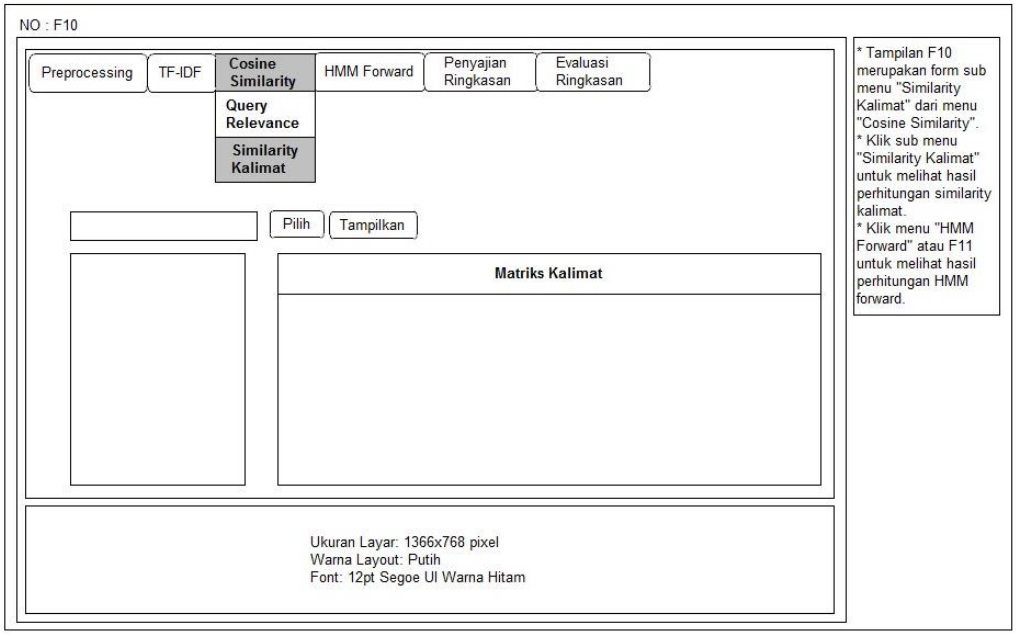
   Pada perancangan *from* F09 merupakan halaman sub menu pada menu proses *cosine similarity*. Perancangan F09 dapat dilihat pada gambar 3.46 berikut:



Gambar 3. 46 *Form Proses Cosine Similarity Query Relevance*

10. F10 – *Form* Proses *Cosine Similarity Similarity* Kalimat

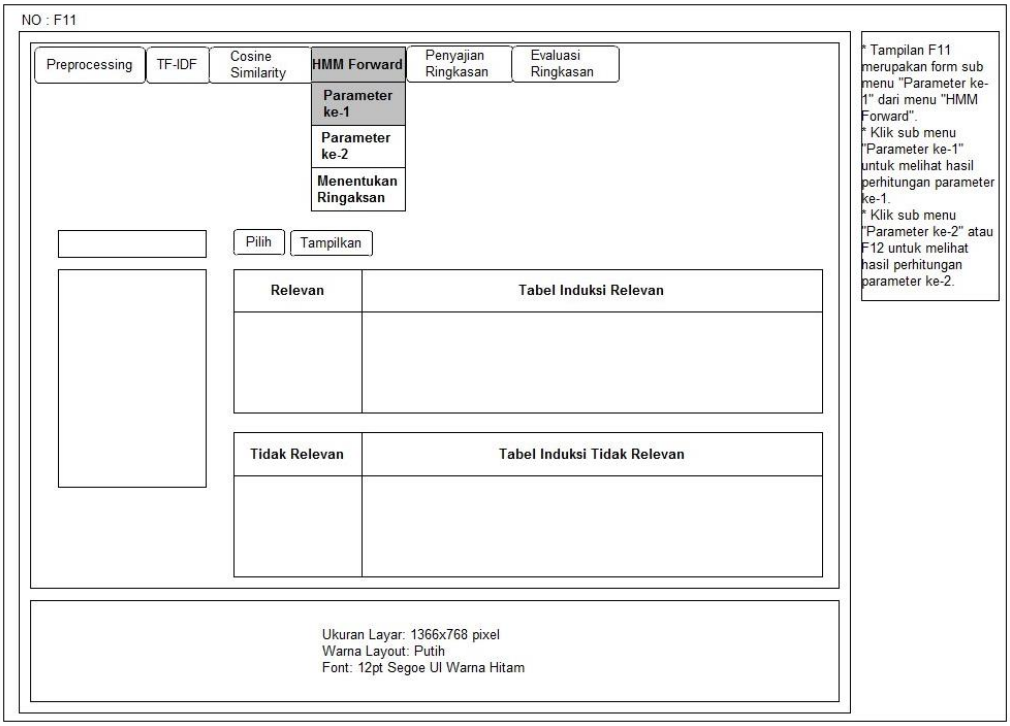
Pada perancangan *form* F10 merupakan halaman sub menu pada menu proses *cosine similarity.* Perancangan F10 dapat dilihat pada gambar 3.47 berikut:



Gambar 3. 47 *Form Proses Cosine Similarity Similarity* Kalimat

11. F11 – *Form* Proses HMM *Forward* Parameter ke-1

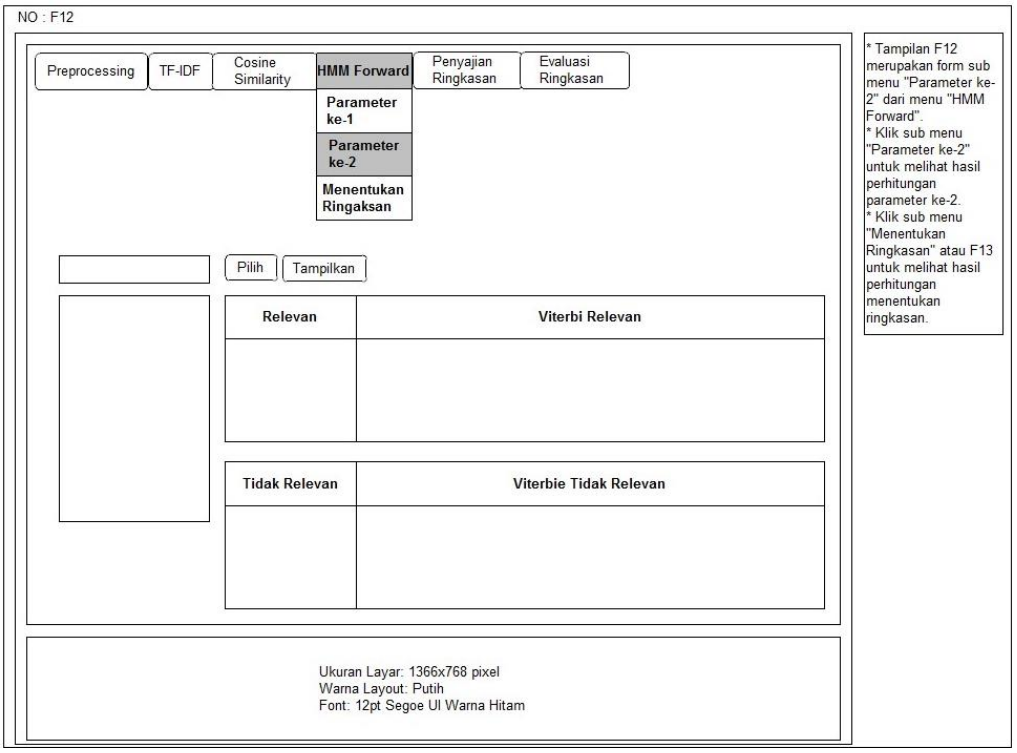
Pada perancangan *form* F11 merupakan halaman sub menu pada menu HMM *Forward.* Perancangan F11 dapat dilihat pada gambar 3.48 berikut:



Gambar 3. 48 *Form Proses HMM Forward* Parameter ke-1

12. F12 – *Form* Proses HMM *Forward* Parameter ke-2

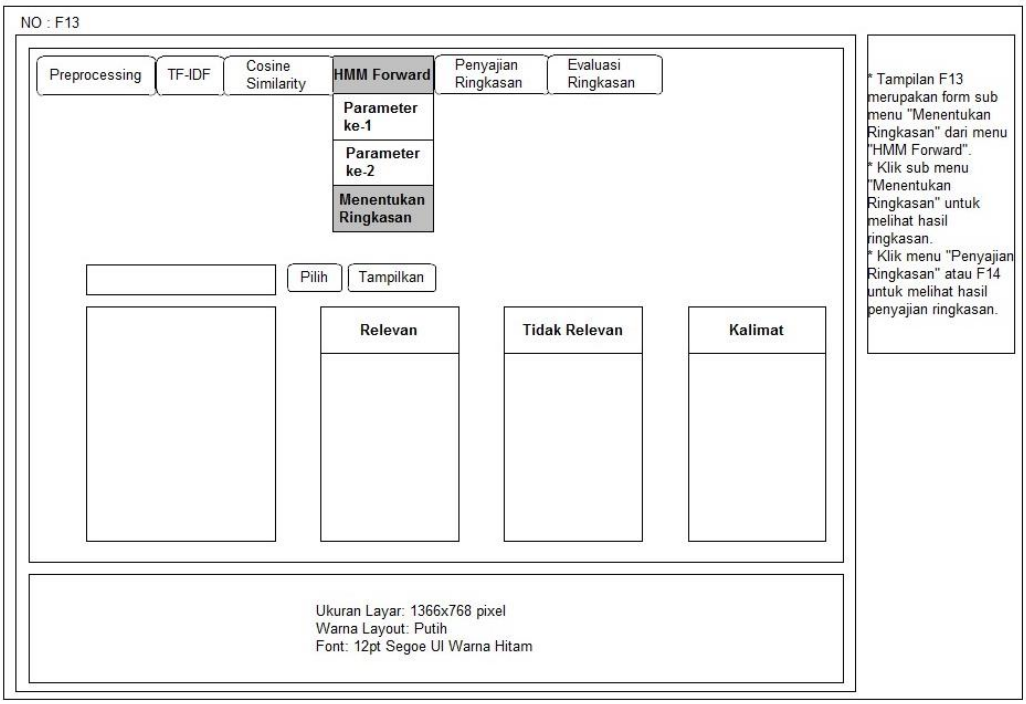
Pada perancangan *form* F12 merupakan halaman sub menu pada menu HMM *Forward.* Perancangan F12 dapat dilihat pada gambar 3.49 berikut:



Gambar 3. 49 *Form Proses HMM Forward* Parameter ke-2

13. F13 – *Form* Proses HMM *Forward* Menentukan Ringkasan

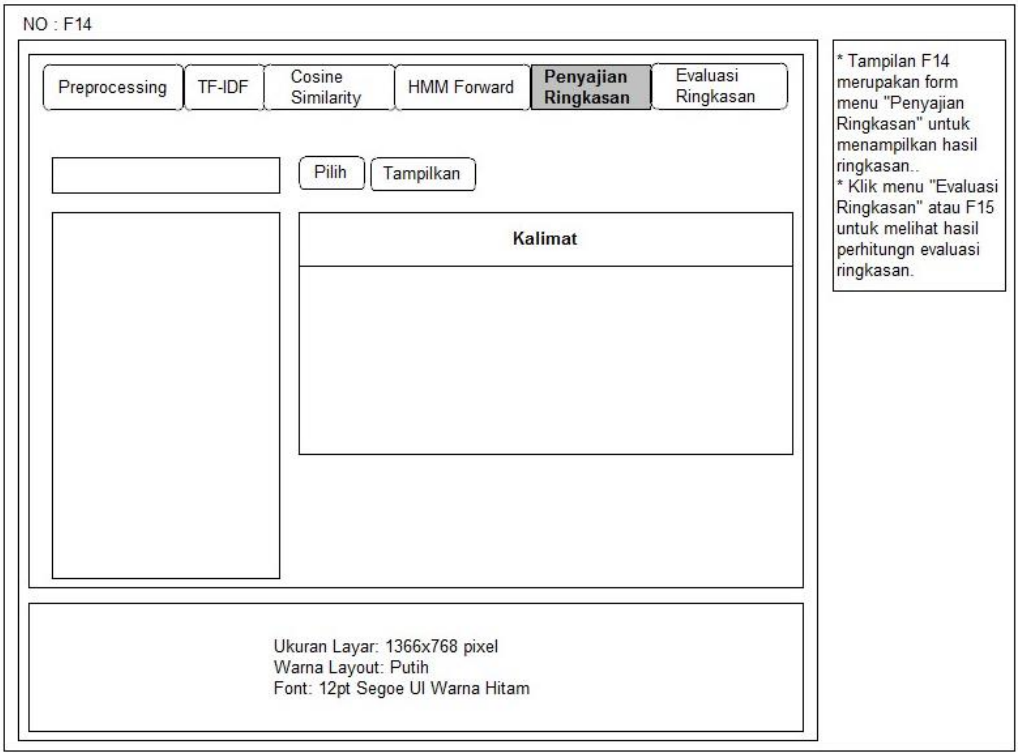
Pada perancangan *form* F13 merupakan halaman sub menu pada menu HMM *Forward.* Perancagan F13 dapat dihilat pada gambar 3.50 berikut:



Gambar 3. 50 *Form Proses HMM Forward* Menentukan Ringkasan

14. F14 – *Form* Proses Penyajian Ringkasan

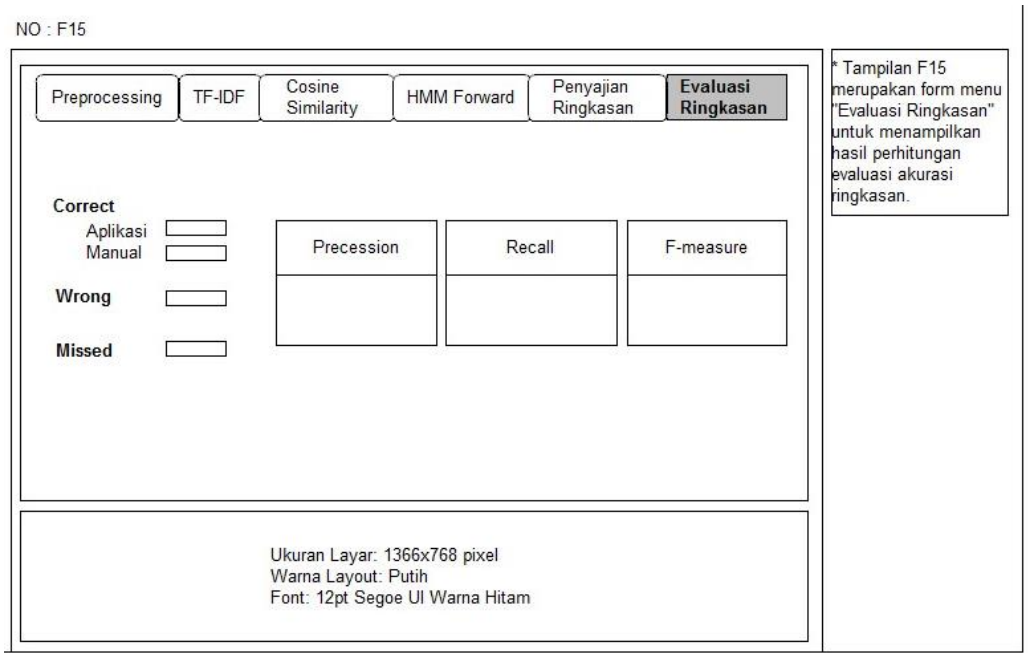
Pada perancangan F14 merupakan halaman penyajian ringkasan dari hasil perhitungan menggunakan metode HMM. Perangan F14 dapat dilihat pada gambar 3.51 berikut:



Gambar 3. 51 *Form Proses* Penyajian Ringkasan

15. F15 – Form Proses Evaluasi Ringkasan

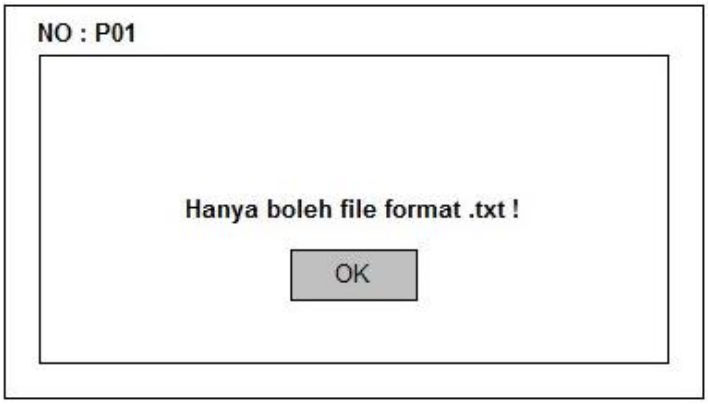
Pada perancangan F15 merupakan halaman evaluasi ringkasan yang memperlihatkan hasil perbandingan antara ringkasan dari program dengan ringkasan dari manusia. Perancangan F15 dapat dilihat pada gambar 3.52 berikut:



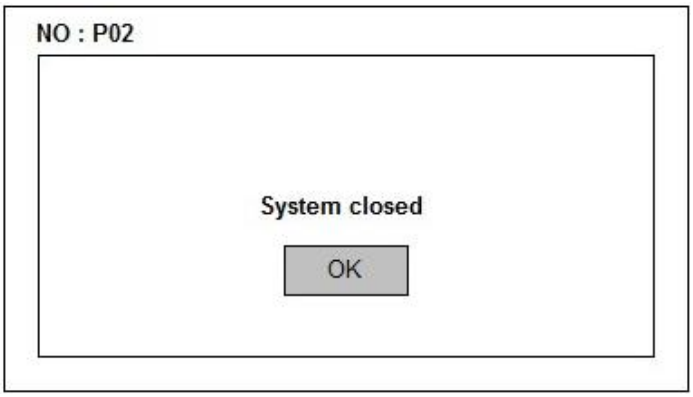
Gambar 3. 52 *Form Proses* Evaluasi Ringkasan

3.8.2 Perancangan Pesan

Perancangan pesan merupakan validasi dan pesan yang akan disampaikan oleh sistem dalam keadaan-keadaan tertentu. Adapun perancangan pesan dapat dilihat pada gambar gambar 3.53 berikut :



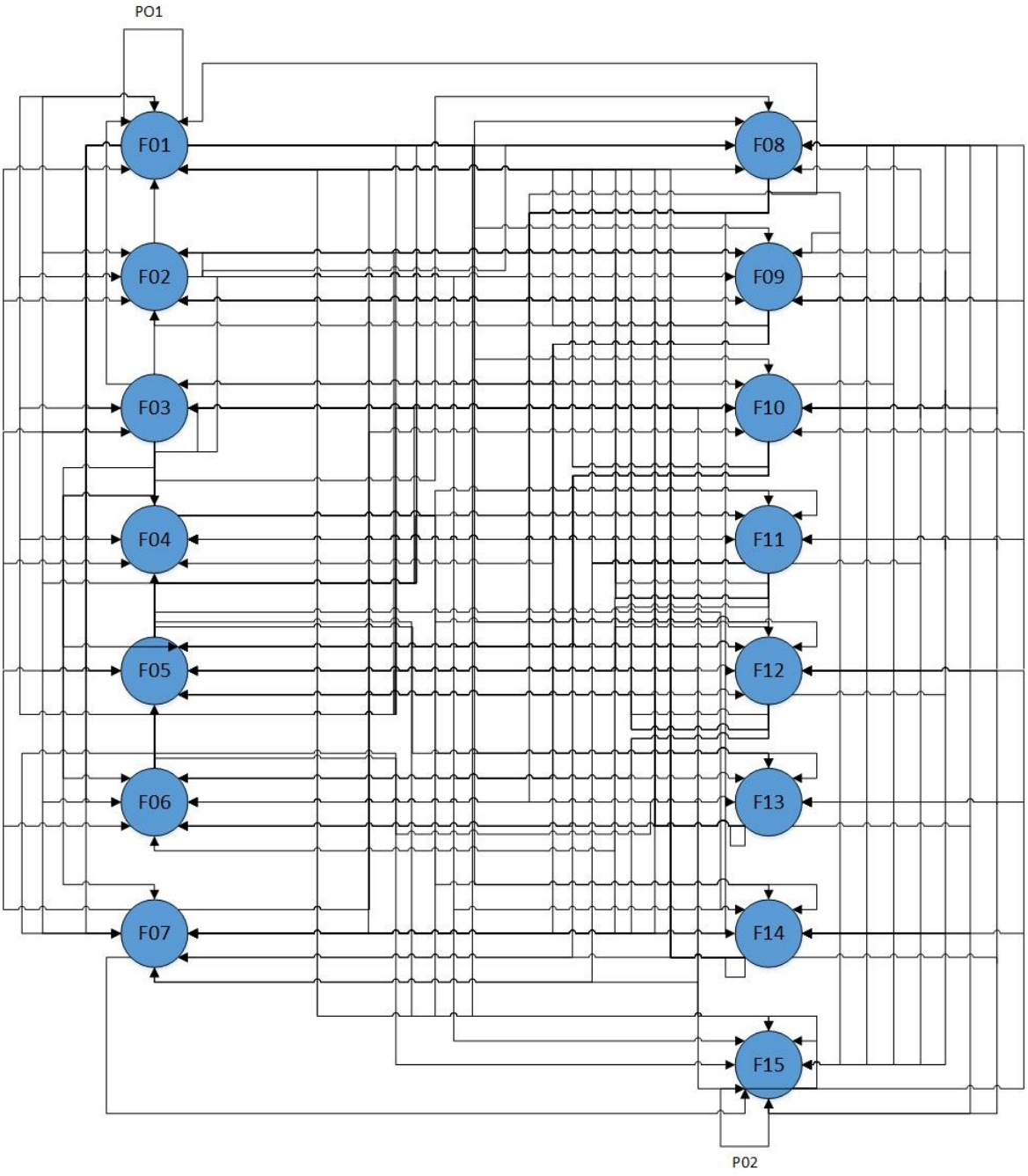
Gambar 3. 53 Perancangan Pesan P01 Salah Pilih *File*



Gambar 3. 54 Perancangan Pesan P02 Keluar Aplikasi

3.8.3 Jaringan Semantik

Jaringan semantic menggambarkan keterhubungan navigasi menu dari satu antarmuka ke antarmuka lain. Jaringan semantik yang terbentuk pada sistem ini dapat dilhat pada gambar 3.54 berikut :



Gambar 3. 55 Jaringan Semantik APT

KETERANGAN

F01 : *Form* Pemilihan *File*

F02 : *Form* Proses *Preprocessing* Pemecahan Kalimat

F03 : *Form* Proses *Preprocessing Case Folding*

F04 : *Form* Proses *Preprocessing Filtering*

F05 : *Form* Proses *Preprocessing Tokenizing*

F06 : *Form* Proses *Preprocessing Stopword Removal*

F07 : *Form* Proses *Preprocessing Stemming*

F08 : *Form* Proses Perhitungan TF-IDF

F09 : *Form* Proses *Cosine Similarity Query Relevance*

F10 : *Form* Proses *Cosine Similarity Similarity* Kalimat

F11 : *Form* Proses HMM *Forward* Parameter ke-1

F12 : *Form* Proses HMM *Forward* Parameter ke-2

F13 : *Form* Proses HMM *Forward* Menentukan Ringkasan

F14 : *Form* Proses Penyajian Ringkasan

F15 : *Form* Proses Evaluasi Ringkasan

P01 : Salah Pilih *File*