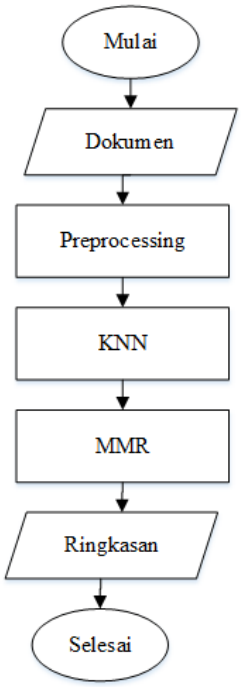
BAB 3
ANALISIS DAN PERANCANGAN

3.1. Analisis Masalah

Berdasarkan uraian pada latar belakang masalah pada penelitian ini terdapat permasalahan dalam akurasi peringkasan. Pada aplikasi *automatic summary* mampu menemukan kata yang relevan sesuai dengan kata kunci (*query*) yang diinginkan oleh pengguna. Namun dengan kata kunci (*query*) yang dimasukkan secara manual, sehingga *user* bisa memasukkan kata apa pun, karena kata yang dimasukkan tidak dibatasi [3]. Data uji coba diambil dari surat kabar berbahasa Indonesia online sejumlah 30 berita. Hasil pengujian dibandingkan dengan ringksan manual yang menghasilkan rata-rata *recall* 60%, *precision* 77%, dan *f-measure* 66% [28]. Dan hasil dari *summary* atau ringkasan mengandung informasi dari teks asli dan panjangnya tidak lebih dari setengah teks asli [26]. Sehingga pada penelitian *text summarization* ini, menggunakan dua metode, yaitu metode K-*Nearest Neighbors* (KNN) untuk mengambil setengah teks asli dan metode *Maximum Marginal Relevance* (MMR) untuk memilih kalimat yang nantinya akan terbentuk sebuah ringkasan dari artikel berbahasa Indonesia.

3.2. Analisis Sistem

Sistem *text summarization* yang akan dibangun dengan gambaran sistem yang tertera pada Gambar 3.1. Tahap sistem peringkasan teks dimulai dari penerimaan input teks dokumen hingga melakukan *preprocessing*. Hasil dari *preprocessing* akan klasifikasi terhadap kalimat yang lainnya menggunakan *K-Nearest* Neighbors (KNN) untuk menentukan kalimat yang relevan dengan kalimat lain. Dan dihitung kerelavanannya tiap kalimat dengan kata kunci dengan menggunakan *Maximum Marginal Relevance* (MMR) sehingga memperoleh hasil ringkasan.



Gambar 3. 1 Gambaran Sistem

3.2.1. Analisis Data Masukan

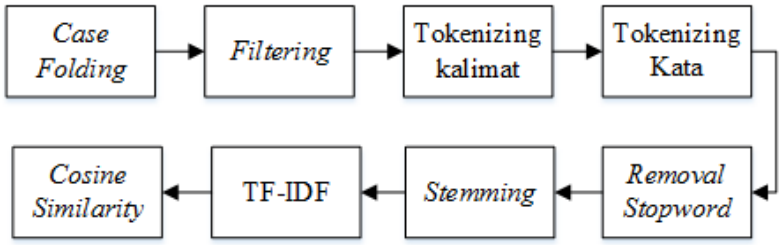
Data masukan yang diperlukan untuk menjalankan sistem ini sebelumnya dimasukkan terlebih dahulu ke dalam database. Sehingga *user* memilih berita mana yang akan di ringkas. Salah satu contoh data masukkan untuk penelitian *text summarization* dapat dilihat pada Tabel 3.1 dengan judul "Presiden Kunjungi Terminal 3 Ultimate Bandara Soekarno-Hatta".

Tabel 3. 1 Data Masukan

Presiden Kunjungi Terminal 3 Ultimate Bandara Soekarno-Hatta
Jakarta - Presiden Jokowi mengunjungi Terminal 3 Ultimate Bandara Soekarno-Hatta. Presiden meninjau langsung pembangunan terminal yang sudah hampir selesai itu. Presiden tiba di Terminal 3 Ultimate Bandara Soekarno-Hatta, Cengkareng, Rabu (11/5/2016) pukul 17:40 WIB. Saat tiba, Jokowi disambut Menteri BUMN Rini Soemarno dan Dirut Angkasa Pura II Budi Karya. Presiden langsung masuk ke kompleks terminal yang berukuran luas itu. Nampak para pekerja masih sibuk bekerja saat Jokowi datang. Jokowi lalu berkeliling ke terminal yang digadang akan bisa menampung ribuan penumpang itu. Dirut AP II lalu memberikan beberapa penjelasan kepada Jokowi mengenai terminal 3 Ultimate yang hampir selesai.

3.2.2. Analisis *Preprocessing*

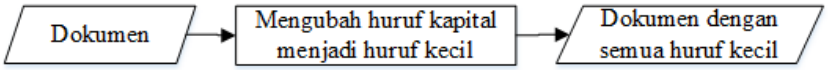
Tahap *processing* merupakan tahapan untuk mempersiapkan data masukan yang akan diolah pada tahap selanjutnya. *Preprocessing* pada penelitian ini terdiri dari beberapa tahapan, yaitu :,*filtering*, *case folding*, *tokenizing* kalimat, *tokenizing* kata, *removal stopword*, *stemming*, TF-IDF, dan *cosine similarity*. Proses *preprocessing* dapat dilihat pada Gambar 3.2.



Gambar 3. 2 Proses *Preprocessing*

3.2.2.1. *Case Folding*

Case Folding digunakan untuk mengubah semua huruf teks menjadi karakter dengan huruf kecil dan pembuangan tanda baca dan angka. Contoh kata "Jakarta" diproses dengan *case folding* sehingga menghasilkan kata "jakarta". Proses *case folding* dapat dilihat pada Gambar 3.3. Dan hasilnya dapat dilihat pada Tabel 3.2.



Gambar 3. 3 Proses *Case Folding*

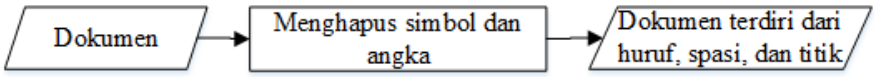
Tabel 3. 2 *Case Folding*

Sebelum	Sesudah
Jakarta - **P**residen **J**okowi mengunjungi **T**erminal 3 **U**ltimate **B**andara **S**oekarno-**H**atta. **P**residen meninjau langsung pembangunan terminal yang sudah hampir selesai itu. **P**residen tiba di **T**erminal 3 **U**ltimate **B**andara **S**oekarno-**H**atta, **C**engkareng, **R**abu	**j**akarta - **p**residen **j**okowi mengunjungi **t**erminal 3 **u**ltimate **b**andara **s**oekarno-**h**atta. **p**residen meninjau langsung pembangunan terminal yang sudah hampir selesai itu. **p**residen tiba di **t**erminal 3 **u**ltimate **b**andara **s**oekarno-**h**atta, **c**engkareng, **r**abu (11/5/2016)

(11/5/2016) pukul 17:40 **WIB**. **S**aat tiba, **J**okowi disambut **M**enteri **BUMN** **R**ini **S**oemarno dan **D**irut **A**ngkasa **P**ura **II** **B**udi **K**arya. **P**residen langsung masuk ke kompleks terminal yang berukuran luas itu. **N**ampak para pekerja masih sibuk bekerja saat **J**okowi datang. **J**okowi lalu berkeliling ke terminal yang digadang akan bisa menampung ribuan penumpang itu. **D**irut **AP II** lalu memberikan beberapa penjelasan kepada **J**okowi mengenai terminal 3 **U**ltimate yang hampir selesai.	pukul 17:40 **wib**. **s**aat tiba, **j**okowi disambut **m**enteri **bumn** **r**ini **s**oemarno dan **d**irut **a**ngkasa **p**ura **ii** **b**udi **k**arya. **p**residen langsung masuk ke kompleks terminal yang berukuran luas itu. **n**ampak para pekerja masih sibuk bekerja saat **j**okowi datang. **j**okowi lalu berkeliling ke terminal yang digadang akan bisa menampung ribuan penumpang itu. **d**irut **ap** **ii** lalu memberikan beberapa penjelasan kepada **j**okowi mengenai terminal 3 **u**ltimate yang hampir selesai.

3.2.2.2. *Filtering*

Filtering digunakan untuk pembuangan tanda baca dan angka pada teks, sehingga pada teks hanya terdapat huruf A..Z, a..z, spasi (“ ”) dan tanda baca titik (“.”). Proses *filtering* ini diterapkan pada penelitian *text summarization* bertujuan untuk memudahkan dalam *tokenizing* kalimat dan pada pembobotan kata (TF-IDF). Contoh pada kalimat “jakarta - presiden jokowi mengunjungi terminal 3 ultimate bandara soekarno-hatta.” setelah diproses *filtering* menjadi “jakarta presiden jokowi mengunjungi terminal ultimate bandara soekarno hatta.”. Proses *filtering* dapat dilihat pada Gambar 3.4. Dan hasilnya dapat dilihat pada Tabel 3.3.



Gambar 3. 4 Proses *Filtering*

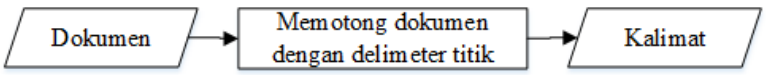
Tabel 3. 3 *Filtering*

Sebelum	Sesudah
jakarta **-** presiden jokowi mengunjungi terminal **3** ultimate bandara soekarno-hatta. presiden meninjau langsung pembangunan terminal yang sudah hampir selesai itu. presiden tiba di terminal **3** ultimate bandara	jakarta presiden jokowi mengunjungi terminal ultimate bandara soekarno hatta. presiden meninjau langsung pembangunan terminal yang sudah hampir selesai itu. presiden tiba di terminal ultimate bandara soekarno hatta

soekarno-hatta, cengkareng, rabu **(11/5/2016)** pukul **17:40** wib. saat tiba, jokowi disambut menteri bumn rini soemarno dan dirut angkasa pura ii budi karya. presiden langsung masuk ke kompleks terminal yang berukuran luas itu. nampak para pekerja masih sibuk bekerja saat jokowi datang. jokowi lalu berkeliling ke terminal yang digadang akan bisa menampung ribuan penumpang itu. dirut ap ii lalu memberikan beberapa penjelasan kepada jokowi mengenai terminal **3** ultimate yang hampir selesai.	cengkareng rabu pukul wib. saat tiba jokowi disambut menteri bumn rini soemarno dan dirut angkasa pura ii budi karya. presiden langsung masuk ke kompleks terminal yang berukuran luas itu. nampak para pekerja masih sibuk bekerja saat jokowi datang. jokowi lalu berkeliling ke terminal yang digadang akan bisa menampung ribuan penumpang itu. dirut ap ii lalu memberikan beberapa penjelasan kepada jokowi mengenai terminal ultimate yang hampir selesai.

3.2.2.3. *Tokenizing* Kalimat

Tokenizing kalimat digunakan untuk memisahkan teks ke dalam bentuk satuan kalimat. Sehingga setiap kalimat akan dijadikan sebagai ukuran untuk penghitungan pada metode TF-IDF. *Tokenizing* kalimat pada sistem ini menggunakan delimeter titik (“.”). Proses *tokenizing* kalimat dapat dilihat pada Gambar 3.5. Dan hasilnya dapat dilihat pada Tabel 3.4.



Gambar 3. 5 Proses *Tokenizing* Kalimat

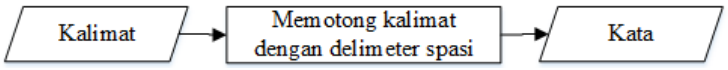
Tabel 3. 4 *Tokenizing* Kalimat

Sebelum	**Sesudah**	
Berita	**No**	**Kalimat**
jakarta presiden jokowi mengunjungi terminal ultimate bandara soekarno hatta. presiden meninjau langsung pembangunan terminal yang sudah hampir selesai itu. presiden tiba di terminal ultimate bandara soekarno hatta cengkareng rabu pukul wib. saat tiba jokowi	1	jakarta presiden jokowi mengunjungi terminal ultimate bandara soekarno hatta
	2	presiden meninjau langsung pembangunan terminal yang sudah hampir selesai itu
	3	presiden tiba di terminal ultimate bandara soekarno hatta cengkareng rabu pukul wib

disambut menteri bumn rini soemarno dan dirut angkasa pura ii budi karya. presiden langsung masuk ke kompleks terminal yang berukuran luas itu. nampak para pekerja masih sibuk bekerja saat jokowi datang. jokowi lalu berkeliling ke terminal yang digadang akan bisa menampung ribuan penumpang itu. dirut ap ii lalu memberikan beberapa penjelasan kepada jokowi mengenai terminal ultimate yang hampir selesai.	4	saat tiba jokowi disambut menteri bumn rini soemarno dan dirut angkasa pura ii budi karya
	5	presiden langsung masuk ke kompleks terminal yang berukuran luas itu
	6	nampak para pekerja masih sibuk bekerja saat jokowi datang
	7	jokowi lalu berkeliling ke terminal yang digadang akan bisa menampung ribuan penumpang itu
	8	dirut ap ii lalu memberikan beberapa penjelasan kepada jokowi mengenai terminal ultimate yang hampir selesai

3.2.2.4. *Tokenizing* kata

Tokenizing kata digunakan untuk memisahkan dari setiap kalimat menjadi satuan kata. Pemisahan kalimat menggunakan delimeter spasi (“ ”). *Tokenizing* kata dilakukan untuk menghitung kata yang sama dalam satu proses pada TF-IDF. Proses *tokenizing* dapat dilihat pada Gambar 3.6. Dan hasilnya dapat dilihat pada Tabel 3.5.



Gambar 3. 6 Proses *Tokeizing* Kata

Tabel 3. 5 *Tokenizing* Kata

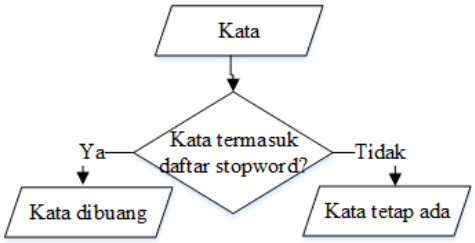
No	Sebelum	Sesudah	
1	jakarta presiden jokowi mengunjungi terminal ultimate bandara soekarno hatta	jakarta	ultimate
		presiden	bandara
		jokowi	soekarno
		mengunjungi	hatta
		terminal	
2	presiden meninjau langsung pembangunan terminal yang sudah hampir selesai itu	presiden	yang
		meninjau	sudah
		langsung	hampir
		pembangunan	selesai
		terminal	itu

3	presiden tiba di terminal ultimate bandara soekarno hatta cengkareng rabu pukul wib	presiden	soekarno
		tiba	hatta
		di	cengkareng
		terminal	rabu
		ultimate	pukul
		bandara	wib
4	saat tiba jokowi disambut menteri bumn rini soemarno dan dirut angkasa pura ii budi karya	saat	dan
		tiba	dirut
		jokowi	angkasa
		disambut	pura
		menteri	ii
		bumn	budi
		rini	karya
		soemarno	
5	presiden langsung masuk ke kompleks terminal yang berukuran luas itu	presiden	terminal
		langsung	yang
		masuk	berukuran
		ke	luas
		kompleks	itu
6	nampak para pekerja masih sibuk bekerja saat jokowi datang	nampak	Bekerja
		para	Saat
		pekerja	Jokowi
		masih	Datang
		sibuk	
7	jokowi lalu berkeliling ke terminal yang digadang akan bisa menampung ribuan penumpang itu	jokowi	akan
		lalu	bisa
		berkeliling	menampung
		ke	ribuan
		terminal	penumpang
		yang	itu
		digadang	
8	dirut ap ii lalu memberikan beberapa penjelasan kepada jokowi mengenai terminal ultimate yang hampir selesai	dirut	jokowi
		ap	mengenai
		ii	terminal
		lalu	ultimate
		memberikan	yang

		beberapa	hampir
		penjelasan	selesai
		kepada	

3.2.2.5. Removal Stopword

Removal Stopword merupakan proses penghapusan kata umum yang tidak memiliki makna atau kata yang kurang berarti dan sering muncul. Daftar *stopword* dapat dilihat pada lampiran Daftar *Stopword.* Pada sistem ini *removal stopword* digunakan agar kata umum yang tidak memiliki makna dan sering muncul tidak dihitung pada metode TF-IDF. Proses *removal stopword* dapat dilihat pada Gambar 3.7. Dan hasilnya dapat dilihat pada Tabel 3.6.



Gambar 3. 7 Proses *Removal Stopword*

Tabel 3. 6 *Removal Stopword*

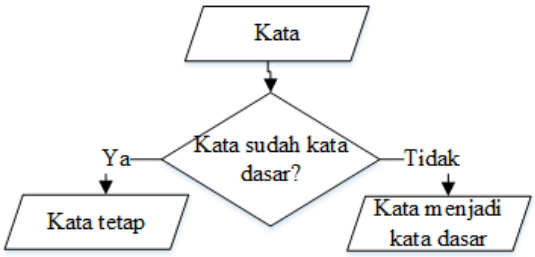
No	Sebelum		Sesudah	
1	jakarta	ultimate	jakarta	ultimate
	presiden	bandara	presiden	bandara
	jokowi	soekarno	jokowi	soekarno
	mengunjungi	hatta	mengunjungi	hatta
	terminal		terminal	
2	presiden	**yang**	presiden	hampir
	meninjau	**sudah**	meninjau	selesai
	langsung	hampir	langsung	
	pembangunan	selesai	pembangunan	
	terminal	**itu**	terminal	
3	presiden	soekarno	presiden	soekarno
	tiba	hatta	tiba	hatta
	di	cengkareng	di	cengkareng

	terminal	rabu	terminal	rabu
	ultimate	pukul	ultimate	pukul
	bandara	wib	bandara	wib
4	**saat**	dan	tiba	dan
	tiba	dirut	jokowi	dirut
	jokowi	angkasa	disambut	angkasa
	disambut	pura	menteri	pura
	menteri	ii	bumn	ii
	bumn	budi	rini	budi
	rini	karya	soemarno	karya
	soemarno			
5	presiden	terminal	presiden	terminal
	langsung	**yang**	langsung	berukuran
	masuk	berukuran	masuk	luas
	ke	luas	ke	
	kompleks	**itu**	kompleks	
6	**nampak**	bekerja	para	bekerja
	para	**saat**	pekerja	jokowi
	pekerja	jokowi	sibuk	datang
	masih	datang		
	sibuk			
7	jokowi	**akan**	jokowi	menampung
	lalu	**bisa**	berkeliling	ribuan
	berkeliling	menampung	ke	penumpang
	ke	ribuan	terminal	
	terminal	penumpang	digadang	
	yang	**itu**		
	digadang			
8	dirut	jokowi	dirut	jokowi
	ap	**mengenai**	ap	terminal
	ii	terminal	ii	ultimate
	lalu	ultimate	memberikan	hampir
	memberikan	**yang**	beberapa	selesai
	beberapa	hampir	penjelasan	
	penjelasan	selesai		
	kepada			

3.2.2.6. Stemming

Stemming merupakan proses pembuangan imbuhan pada kata menjadi kata dasar. Sehingga setiap kata memiliki resepresentasi yang sama. *Stemming* dilakukan hanya untuk penghitungan pada TF-IDF.

Stemming pada sistem ini diterapkan dengan tujuan setiap kata memiliki representasi yang sama. Dan kata tersebut hanya dilakukan untuk perhitungan TF-IDF. *Stemming* yang digunakan pada penelitian ini, menggunakan *stemming* Nazief dan Adriani. Proses *stemming* dapat dilihat pada Gambar 3.8. Dan hasil *stemming* dapat dilihat pada Tabel 3.7.



Gambar 3. 8 Proses *Stemming*

Tabel 3. 7 *Stemming*

No	Sebelum		Sesudah	
1	jakarta	ultimate	jakarta	ultimate
	presiden	bandara	presiden	bandara
	jokowi	soekarno	jokowi	soekarno
	mengunjungi	hatta	**kunjung**	hatta
	terminal		terminal	
2	presiden	terminal	presiden	terminal
	meninjau	hampir	**tinjau**	hampir
	langsung	selesai	langsung	selesai
	pembangunan		**bangun**	
3	presiden	soekarno	presiden	soekarno
	tiba	hatta	tiba	hatta
	di	cengkareng	di	cengkareng
	terminal	rabu	terminal	rabu
	ultimate	pukul	ultimate	pukul

	bandara	wib	bandara	wib
4	tiba	dan	tiba	dan
	jokowi	dirut	jokowi	dirut
	disambut	angkasa	**sambut**	angkasa
	menteri	pura	menteri	pura
	bumn	ii	bumn	ii
	rini	budi	rini	budi
	somearno	karya	somearno	karya
5	presiden	kompleks	presiden	kompleks
	langsung	terminal	langsung	terminal
	masuk	berukuran	masuk	berukuran
	ke	luas	ke	luas
6	para	**bekerja**	para	**kerja**
	pekerja	jokowi	**kerja**	jokowi
	sibuk	datang	sibuk	datang
7	jokowi	**digadang**	jokowi	**gadang**
	berkeliling	**menampung**	**keliling**	**tamping**
	ke	**ribuan**	ke	**ribu**
	terminal	**penumpang**	terminal	**tumpang**
8	dirut	jokowi	dirut	jokowi
	ap	terminal	ap	terminal
	ii	ultimate	ii	ultimate
	memberikan	hampir	**beri**	hampir
	beberapa	selesai	beberapa	selesai
	penjelasan		**jelas**	

Kemudian kata-kata dikembalikan kembali menjadi sebuah kalimat yang telah di*preprocessing.* Kalimat-kalimat ini akan dianggap dokumen pada penghitungan TF-IDF. Kalimat hasil *preprocessing* dapat dilihat pada Tabel 3.8.

Tabel 3. 8 Hasil *Stemming*

No	Kalimat
S1	jakarta presiden jokowi kunjung terminal ultimate bandara soekarno
S2	presiden tinjau langsung bangun terminal hampir selesai
S3	presiden tiba di terminal ultimate bandara soekarno hatta cengkareng rabu pukul wib

S4	tiba jokowi sambut menteri bumn rini soemarno dan dirut angkasa pura ii budi karya
S5	presiden langsung masuk ke kompleks terminal ukur luas
S6	para pekerja sibuk bekerja jokowi datang
S7	jokowi keliling ke terminal gadang tampung ribu tumpang
S8	dirut ap ii beri beberapa jelas jokowi terminal ultimate hampir selesai

3.2.2.7. Metode TF-IDF (*Term Frequency - Inverse Document Frequency*)

Proses selanjutnya yaitu penghitungan bobot kata menggunakan metode TF-IDF. Yaitu dengan menghitung kemunculan satu kata pada setiap kalimat.

Sebagai contoh, kata "jakarta".

1. Untuk mengisi kolom tf, hitung kemunculan kata "jakarta" pada setiap kalimat. Kata "jakarta" hanya muncul pada kalimat 1 saja, sehingga pada tf-S1 bernilai 1 dan tf-S lainnya diberi nilai 0.
2. Selanjutnya kolom df, jumlahkan kemunculan kata "jakarta" dalam satu berita (dari tf-S1 sampai tf-S8). Karena kata "jakarta" hanya muncul 1 kali pada kalimat 1, maka df bernilai 1.
3. Mengisi kolom idf, yaitu dengan rumus (2.1) dengan memasukkan nilai n (jumlah kata) dan df (proses no.2). Nilai n=78, dan df=1, sehingga idf kata "jakarta" = log(78/1) = 1,89209
4. Mengisi kolom w, yaitu dengan rumus (2.2) dengan memasukkan nilai tf (proses no.1) dan idf (proses no.3). Sehingga mengisi w-S1 = tf-S1.idf = 1x1,89209=1,89209 dan w-Sselanjutnya

Penghitungan TF-IDF dapat dilihat pada Tabel 3.9.

Tabel 3. 9 Perhitungan TF-IDF

Kata	tf									Idf	W = tf . idf							
	S1	S2	S3	S4	S5	S6	S7	S8	df	log(n/df)	S1	S2	S3	S4	S5	S6	S7	S8
jakarta	1	0	0	0	0	0	0	0	1	1,89209	1,89209	0	0	0	0	0	0	0
presiden	1	1	1	0	1	0	0	0	4	1,29003	1,29003	1,29003	1,29003	0	1,29003	0	0	0
jokowi	1	0	0	1	0	1	1	1	5	1,19312	1,19312	0	0	1,19312	0	1,19312	1,19312	1,19312
kunjung	1	0	0	0	0	0	0	0	1	1,89209	1,89209	0	0	0	0	0	0	0
terminal	1	1	1	0	1	0	1	1	6	1,11394	1,11394	1,11394	1,11394	0	1,11394	0	1,11394	1,11394
ultimate	1	0	1	0	0	0	0	1	3	1,41497	1,41497	0	1,41497	0	0	0	0	1,41497
bandara	1	0	1	0	0	0	0	0	2	1,59106	1,59106	0	1,59106	0	0	0	0	0
soekarno	1	0	1	0	0	0	0	0	2	1,59106	1,59106	0	1,59106	0	0	0	0	0
hatta	1	0	1	0	0	0	0	0	2	1,59106	1,59106	0	1,59106	0	0	0	0	0
tinjau	0	1	0	0	0	0	0	0	1	1,89209	0	1,89209	0	0	0	0	0	0
langsung	0	1	0	0	1	0	0	0	2	1,59106	0	1,59106	0	0	1,59106	0	0	0
bangun	0	1	0	0	0	0	0	0	1	1,89209	0	1,89209	0	0	0	0	0	0
hampir	0	1	0	0	0	0	0	1	2	1,59106	0	1,59106	0	0	0	0	0	1,59106
selesai	0	1	0	0	0	0	0	1	2	1,59106	0	1,59106	0	0	0	0	0	1,59106
tiba	0	0	1	1	0	0	0	0	2	1,59106	0	0	1,59106	1,59106	0	0	0	0
di	0	0	1	0	0	0	0	0	1	1,89209	0	0	1,89209	0	0	0	0	0
cengkareng	0	0	1	0	0	0	0	0	1	1,89209	0	0	1,89209	0	0	0	0	0
rabu	0	0	1	0	0	0	0	0	1	1,89209	0	0	1,89209	0	0	0	0	0
pukul	0	0	1	0	0	0	0	0	1	1,89209	0	0	1,89209	0	0	0	0	0

wib	0	0	1	0	0	0	0	0	1	1,89209	0	0	1,89209	0	0	0	0	0
sambut	0	0	0	1	0	0	0	0	1	1,89209	0	0	0	1,89209	0	0	0	0
menteri	0	0	0	1	0	0	0	0	1	1,89209	0	0	0	1,89209	0	0	0	0
bumn	0	0	0	1	0	0	0	0	1	1,89209	0	0	0	1,89209	0	0	0	0
rini	0	0	0	1	0	0	0	0	1	1,89209	0	0	0	1,89209	0	0	0	0
soemarno	0	0	0	1	0	0	0	0	1	1,89209	0	0	0	1,89209	0	0	0	0
dan	0	0	0	1	0	0	0	0	1	1,89209	0	0	0	1,89209	0	0	0	0
dirut	0	0	0	1	0	0	0	1	2	1,59106	0	0	0	1,59106	0	0	0	1,59106
angkasa	0	0	0	1	0	0	0	0	1	1,89209	0	0	0	1,89209	0	0	0	0
pura	0	0	0	1	0	0	0	0	1	1,89209	0	0	0	1,89209	0	0	0	0
ii	0	0	0	1	0	0	0	1	2	1,59106	0	0	0	1,59106	0	0	0	1,59106
budi	0	0	0	1	0	0	0	0	1	1,89209	0	0	0	1,89209	0	0	0	0
karya	0	0	0	1	0	0	0	0	1	1,89209	0	0	0	1,89209	0	0	0	0
masuk	0	0	0	0	1	0	0	0	1	1,89209	0	0	0	0	1,89209	0	0	0
ke	0	0	0	0	1	0	1	0	2	1,59106	0	0	0	0	1,59106	0	1,59106	0
kompleks	0	0	0	0	1	0	0	0	1	1,89209	0	0	0	0	1,89209	0	0	0
ukur	0	0	0	0	1	0	0	0	1	1,89209	0	0	0	0	1,89209	0	0	0
luas	0	0	0	0	1	0	0	0	1	1,89209	0	0	0	0	1,89209	0	0	0
para	0	0	0	0	0	1	0	0	1	1,89209	0	0	0	0	0	1,89209	0	0
kerja	0	0	0	0	0	2	0	0	2	1,59106	0	0	0	0	0	1,59106	0	0
sibuk	0	0	0	0	0	1	0	0	1	1,89209	0	0	0	0	0	1,89209	0	0
datang	0	0	0	0	0	1	0	0	1	1,89209	0	0	0	0	0	1,89209	0	0

keliling	0	0	0	0	0	0	1	0	1	1,89209	0	0	0	0	0	0	1,89209	0
gadang	0	0	0	0	0	0	1	0	1	1,89209	0	0	0	0	0	0	1,89209	0
tampung	0	0	0	0	0	0	1	0	1	1,89209	0	0	0	0	0	0	1,89209	0
ribu	0	0	0	0	0	0	1	0	1	1,89209	0	0	0	0	0	0	1,89209	0
tumpang	0	0	0	0	0	0	1	0	1	1,89209	0	0	0	0	0	0	1,89209	0
ap	0	0	0	0	0	0	0	1	1	1,89209	0	0	0	0	0	0	0	1,89209
beri	0	0	0	0	0	0	0	1	1	1,89209	0	0	0	0	0	0	0	1,89209
beberapa	0	0	0	0	0	0	0	1	1	1,89209	0	0	0	0	0	0	0	1,89209
jelas	0	0	0	0	0	0	0	1	1	1,89209	0	0	0	0	0	0	0	1,89209

3.2.2.8. *Cosine Similarity* Antar Kalimat

Apabila bobot kata telah diperoleh, kemudian hitung vektornya terlebih dahulu, untuk bisa menghitung similaritas antar kalimat dan query, dan similaritas kalimat dan teks.

Proses penghitungan $\sum$ TF-IDF^2 terhadap kalimat lain, yaitu dengan mengambil nilai w atau TF-IDF yang telah dihitung sebelumnya, kemudian di pangkat dua. $\sum$ TF-IDF^2 dapat dilihat pada Tabel 3.10.

Tabel 3. 10 Perhitungan $\sum$ TF-IDF^2 S1

Kata	TF-IDF	TF-IDF^2
jakarta	1,89209	3,58000
presiden	1,29003	1,66418
jokowi	1,19312	1,42354
kunjung	1,89209	3,58000
terminal	1,11394	1,24086
ultimate	1,41497	2,00214
bandara	1,59106	2,53147
soekarno	1,59106	2,53147
hatta	1,59106	2,53147
$\sum$		21,08514

Perhitungan kalimat selanjutnya dapat dilihat pada lampiran Tabel Perhitungan $\sum$ TF-IDF^2. Setelah mendapatkan nilai TF-IDF^2 pada setiap kalimat, kemudian hitung vektor setiap kalimat tersebut dengan mengakarkan jumlah TF-IDF^2 setiap kalimat. Berikut adalah perhitungan vektor dari setiap kalimat, dapat dilihat pada Tabel 3.11.

Tabel 3. 11 Perhitungan Vektor

Kalimat	$\sum$ TF-IDF^2	Vektor = $\sqrt{\sum}$ TF-IDF^2
S1	21,08514	4,59186
S2	17,65946	4,20232
S3	32,93309	5,73874
S4	44,81800	6,69462

S5	22,28800	4,72102
S6	14,69502	3,83341
S7	23,09589	4,80582
S8	29,11244	5,39559

Setelah mendapatkan vektor, kemudian tiap kalimat dikalikan dengan kalimat yang lain untuk selanjutnya digunakan pada similarity.

Contoh perkalian kalimat 1 dengan kalimat lain dengan kata “jakarta”, ambil w atau TF-IDF dari kata “jakarta” pada w-S1 dan w-S lainnya, selanjutnya kalikan w-S1 dengan w-S2, w-S1 dengan w-3, dan seterusnya.

Hasil perhitungan dapat dilihat pada Tabel 3.12.

Tabel 3. 12 Hasil Perkalian Antara S1 dan Kalimat Lain (S1 * Si)

Kata	w=tf.idf								S1 * Si						
	S1	S2	S3	S4	S5	S6	S7	S8	S2	S3	S4	S5	S6	S7	S8
jakarta	1,89209	0	0	0	0	0	0	0	0	0	0	0	0	0	0
presiden	1,29003	1,29003	1,29003	0	1,29003	0	0	0	1,66418	1,66418	0	1,66418	0	0	0
jokowi	1,19312	0	0	1,19312	0	1,19312	1,19312	1,19312	0	0	1,42354	0	1,42354	1,42354	1,42354
kunjung	1,89209	0	0	0	0	0	0	0	0	0	0	0	0	0	0
terminal	1,11394	1,11394	1,11394	0	1,11394	0	1,11394	1,11394	1,24086	1,24086	0	1,24086	0	1,24086	1,24086
ultimate	1,41497	0	1,41497	0	0	0	0	1,41497	0	2,00214	0	0	0	0	2,00214
bandara	1,59106	0	1,59106	0	0	0	0	0	0	2,53147	0	0	0	0	0
soekarno	1,59106	0	1,59106	0	0	0	0	0	0	2,53147	0	0	0	0	0
hatta	1,59106	0	1,59106	0	0	0	0	0	0	2,53147	0	0	0	0	0
∑									2,90504	12,50159	1,42354	2,90504	1,42354	2,66440	4,66654

Setelah memperoleh hasil perkalian kalimat 1 dengan kalimat lainnya, selanjutnya menghitung *cosine similarity* dengan menggunakan rumus

$$Sim(S1, Si) = \frac{\sum(S1.Si)}{\sqrt{\sum S1^2}\sqrt{\sum Si^2}}$$

Tabel 3. 13 Perhitungan *Cosine Similarity* Kalimat 1 (S1)

Si	$\sum(S1.Si)$	$\sqrt{\sum S1^2}$	$\sqrt{\sum Si^2}$	$Sim(S1, Si)$
S2	2,90504	4,59286	4,20232	0,15055
S3	12,50159	4,59286	5,73874	0,47442
S4	1,42354	4,59286	6,69462	0,04631
S5	2,90504	4,59286	4,72102	0,13401
S6	1,42354	4,59286	3,83341	0,13401
S7	2,66440	4,59286	4,80582	0,12074
S8	4,66654	4,59286	5,39559	0,18835
∑				1,19524

Perhitungan perkalian kalimat selanjutnya dapat dilihat pada lampiran Tabel perkalian antar kalimat. Dan perhitungan *cosine similarity* selanjutnya dapat dilihat pada lampiran Tabel *cosine similarity*. Hasil perhitungan *cosine similarity* antarkalimat dapat dilihat pada Tabel 3.14

Tabel 3. 14 Hasil *Cosine Similarity* Antar Kalimat

	S1	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**
S1	1	0,15055	0,47442	0,04631	0,13401	0,08087	0,12074	0,18835
S2	0,15055	1	0,12046	0	0,27403	0	0,06144	0,27802
S3	0,47442	0,12046	1	0,06589	0,10723	0,06471	0,04499	0,10473
S4	0,04631	0,08998	0	1	0	0	0,04425	0,17957
S5	0,13401	0,14643	0,20066	0	1	0	0,16627	0,04871
S6	0,08087	0	0	0,05547	0	1	0,07727	0,06882
S7	0,12074	0,31214	0,04499	0,04425	0,16627	0,07727	1	0,10275
S8	0,18835	0,05473	0,10473	0,17957	0,04871	0,06882	0,10275	1

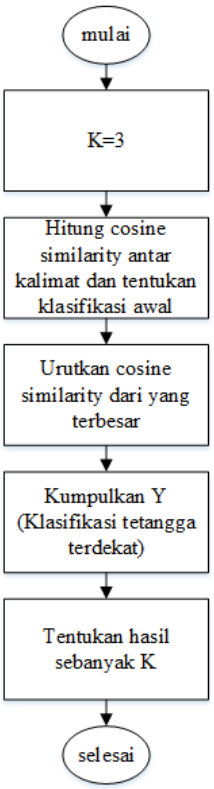
Hasil total *cosine similarity* kalimat satu terhadap kalimat yang lain dapat dilihat pada Tabel 3.15.

Tabel 3. 15 Hasil Total *Cosine Similarity* Kalimat Satu Terhadap Kalimat yang Lain

Kalimat	***cosine similarity***
S1	1,19524
S2	0,87428
S3	0,94527
S4	0,39149
S5	0,73024
S6	0,29168
S7	0,61771
S8	0,97097

3.2.3. Metode KNN (K-Nearest Neighbors)

Setelah menghitung dan mendapatkan nilai *cosine similarity* antarkalimat, selanjutnya proses KNN, untuk memilah kalimat yang akan diproses pada proses MMR.



Gambar 3. 9 Proses KNN

Langkah pertama adalah menentukan parameter K (jumlah tetangga paling dekat). Pada penelitian *text summarization* ini nilai K = 3 dapat dilihat pada Lampiran F Menentukan Nilai K Pada KNN. Karena dokumen yang dipakai adalah dokumen kecil (artikel berita). Sehingga apabila diambil nilai K-nya terlalu besar maka lingkup tetangganya besar.

Selanjutnya menghitung jarak masing-masing objek. Perhitungan jarak menggunakan *cosine similarity* yang telah dilakukan pada proses sebelumnya. Untuk mengetahui relevan dan tidak relevan sebagai tetangganya, maka tentukan terlebih dahulu nilai relevan dan tidak relevan sebuah kalimat. Dikatakan relevan apabila nilai *cosine similarity*-nya melebihi nilai rata-rata *cosine similarity*. Dan dikatakan tidak relevan apabila nilai *cosine similarity*-nya kurang dari nilai rata-rata *cosine similarity*. Hasil relevansi dapat dilihat pada Tabel 3.16.

Tabel 3. 16 Relevansi Kalimat

Kalimat	*cosine similarity*	Klasifikasi
S1	1,19524	Relevan
S2	0,87428	Relevan
S3	0,94527	Relevan
S4	0,39149	Tidak Relevan
S5	0,73024	Tidak Relevan
S6	0,29168	Tidak Relevan
S7	0,61771	Tidak Relevan
S8	0,97097	Relevan
Rata-rata	0,75211	

Kemudian urutkan objek yang mempunyai jarak terkecil dengan mengurutkan secara *descending* atau dari yang terbesar ke yang terkecil. Hasil urutan dapat dilihat pada Tabel 3.17.

Tabel 3. 17 *Cosine Similarity* Telah Terurut

Ranking	Kalimat	*cosine similarity*	Klasifikasi
1	S1	1,19524	Relevan
2	S8	0,97097	Relevan
3	S3	0,94527	Relevan
4	S2	0,87428	Relevan
5	S5	0,73024	Tidak Relevan
6	S7	0,61771	Tidak Relevan
7	S4	0,39149	Tidak Relevan
8	S6	0,29168	Tidak Relevan

Selanjutnya kumpulkan Y (klasifikasi tetangga terdekat) dan tentukan hasil berdasarkan K.

Tabel 3. 18 Hasil KNN

Kalimat	Tetangga	Klasifikasi Tetangga	Hasil
S1	S8, S3, S2	Relevan, Relevan, Relevan	Relevan

S2	S5, S7, S4	Tidak Relevan, Tidak Relevan, Tidak Relevan	Tidak Relevan
S3	S2, S5, S7	Relevan, Tidak Relevan, Tidak Relevan	Tidak Relevan
S4	S6, S1, S8	Relevan, Relevan, Relevan	Relevan
S5	S7, S4, S6	Tidak Relevan, Tidak Relevan, Tidak Relevan	Tidak Relevan
S6	S1, S8, S3	Relevan, Relevan, Relevan	Relevan
S7	S4, S6, S1	Tidak Relevan, Tidak Relevan, Relevan	Tidak Relevan
S8	S3, S2, S5	Relevan, Relevan, Tidak Relevan	Relevan

Setelah mendapatkan hasil KNN dari setiap kalimat, selanjutnya ambil kalimat yang mempunyai hasil relevan untuk diproses pada MMR. Pada penelitian ini hasil kalimat relevan berjumlah 4 kalimat, yaitu S1, S4, S6, dan S8. Kalimat yang akan diproses dapat dilihat pada Tabel 3.19.

Tabel 3. 19 Kalimat yang Akan Diproses

Kalimat	***cosine similarity***
S1	1,19524
S4	0,39149
S6	0,29168
S8	0,97097

3.2.4. Metode MMR (*Maximum Marginal Relevance*)

Dalam *Maximum Marginal Relevance* terdapat proses *cosine similarity query* terhadap kalimat untuk mengetahui kesamaan antara *query* dan kalimat, dan proses perhitungan untuk memperoleh hasil ringkasan

3.2.4.1. *Cosine Similarity Query* Terhadap Kalimat

Setelah memperoleh *cosine similarity* pada masing-masing kalimat. Selanjutnya penghitungan *cosine similarity* pada *query*. *Query* juga di preprocessing, sehingga judul ikut dibandingkan dengan kalimat yang telah dipilih.

- *Query*

  Query pada penelitian ini diambil dari judul berita, yaitu

Presiden Kunjungi Terminal 3 Ultimate Bandara Soekarno-Hatta

- *Filtering*

  Filtering dengan menghapus angka dan simbol pada *query*. Sehingga *query* hanya terdiri dari huruf, spasi, dan titik.

Tabel 3. 20 *Filtering Query*

Sebelum	Sesudah
Presiden Kunjungi Terminal **3** Ultimate Bandara Soekarno-Hatta	Presiden Kunjungi Terminal Ultimate Bandara Soekarno Hatta

- *Case Folding*

  Case folding dengan mengubah semua huruf *query* menjadi huruf kecil atau *lowcase*.

Tabel 3. 21 *Case Folding Query*

Sebelum	Sesudah
Presiden **K**unjungi **T**erminal **U**ltimate **B**andara **S**oekarno **H**atta	**p**residen **k**unjungi **t**erminal **u**ltimate **b**andara **s**oekarno **h**atta

- *Tokenizing* kata

  Tokenizing kata, yaitu dengan membagi kalimat dengan delimeter spasi " ". Hasil *tokenizing* kata dapat dilihat pada Tabel 3.22

Tabel 3. 22 *Tokenizing* Kata *Query*

Sebelum	Sesudah	
presiden kunjungi terminal ultimate bandara soekarno hatta	presiden	kunjungi
	terminal	ultimate
	bandara	soekarno
	hatta	

- *Removal Stopword*

  Removal stopword, yaitu dengan membuang kata yang tidak bermakna atau sering muncul. Hasil *removal stopword* dapat dilihat pada Tabel 3.21

Tabel 3. 23 *Removal Stopword Query*

Sebelum		**Sesudah**	
presiden	kunjungi	presiden	kunjungi
terminal	ultimate	terminal	ultimate
bandara	soekarno	bandara	soekarno
hatta		hatta	

- *Stemming*

  Stemming, yaitu dengan membuang imbuhan pada *query*, sehingga *query* terdiri dari kata dasar. Hasil *stemming* kata dapat dilihat pada Tabel 3.22.

Tabel 3. 24 *Stemming Query*

Sebelum		**Sesudah**	
presiden	**kunjungi**	presiden	**kunjung**
terminal	ultimate	terminal	ultimate
bandara	soekarno	bandara	soekarno
hatta		hatta	

Setelah *preprocessing*, lakukan TF-IDF terhadap kata yang sudah ada untuk mendapatkan nilai vektor *query*. Perhitungan $\sum$ TF-IDF^2 *query* dapat dilihat pada Tabel 3.23.

Tabel 3. 25 Perhitungan $\sum$ TF-IDF^2 *Query*

Kata	**TF-IDF**	**TF-IDF^2**
presiden	1,29003	1,66418
kunjung	1,89209	3,58000
terminal	1,11394	1,24086
ultimate	1,41497	2,00214
bandara	1,59106	2,53147
soekarno	1,59106	2,53147
hatta	1,59106	2,53147
$\sum$		16,08160
Vektor atau $\sqrt{\sum}$		4,01019

Selanjutnya hitung *cosine similarity query* terhadap masing-masing kalimat. Hasil perkalian *query* dengan kalimat lainnya dapat dilihat pada tabel 3.26.

Tabel 3. 26 Hasil Perkalian Antara *Query* dan Kalimat Lain (query * Si)

Kata	Query	S1	S4	S6	S8	QS1	QS4	QS6	QS8
presiden	1,29003	1,29003	0	0	0	1,66418	0	0	0
kunjung	1,89209	1,89209	0	0	0	3,58000	0	0	0
terminal	1,11394	1,11394	0	0	1,11394	1,24086	0	0	1,24086
ultimate	1,41497	1,41497	0	0	1,41497	2,00214	0	0	2,00214
bandara	1,59106	1,59106	0	0	0	2,53147	0	0	0
soekarno	1,59106	1,59106	0	0	0	2,53147	0	0	0
hatta	1,59106	1,59106	0	0	0	2,53147	0	0	0
∑						16,08159	0	0	3,24300

Berikut adalah perhitungan *cosine similarity query* terhadap kalimat lainnya, dapat dilihat pada Tabel 3.27.

Tabel 3. 27 Perhitungan *Cosine Similarity* Antara *Query* dengan Kalimat Lain

Si	$\sum(Q.Si)$	$\sqrt{\sum Q^2}$	$\sqrt{\sum Si^2}$	$Sim(Q,Si)$
S1	16,08159	4,01019	4,59186	0,87332
S4	0	4,01019	6,69462	0
S6	0	4,01019	3,83341	0
S8	3,24300	4,01019	5,39559	0,14988

3.2.4.2. Perhitungan Iterasi

Setelah perhitungan *cosine similarity* diperoleh, maka tahap berikutnya adalah menghitung MMR. Perhitungan MMR dilakukan dengan iterasi mengkombinasikan 2 matrik *cosine similarity* antara *query* terhadap kalimat lain, dan antara kalimat satu dengan kalimat yang lain. Pada prinsipnya cara kerja MMR adalah mengambil kalimat dengan nilai tertinggi dari setiap iterasi. Iterasi akan berhenti, jika nilai maksimum MMR sama dengan nol (0). Nilai λ yang digunakan untuk peringkasan dokumen paling efektif λ = 0,3 untuk memberi penekanan lebih,

dan kemudian meningkatkannya λ = 0,7 untuk fokus pada dokumen yang paling relevan [30]. Sehingga pada penelitian *text summarization* ini nilai λ = 0,7 dapat dilihat pada Lampiran E Menentukan Nilai λ Pada MMR.

Sebelumnya telah diproses KNN. Dan berikut adalah *cosinus similarity* kalimat yang akan diproses dapat dilihat pada Tabel 3.28.

Tabel 3. 28 *Cosine similarity* Kalimat yang Akan Diproses

	S1	**S4**	**S6**	**S8**
S1	1	0,04631	0,08087	0,18835
S4	0,04631	1	0	0,17957
S6	0,08087	0,05547	1	0,06882
S8	0,18835	0,17957	0,06882	1

1. Iterasi ke-1

   Pada iterasi ke-1 ini adalah tahap pertama sehingga Sim(Si,Sj) belum ada , maka dari itu nilai Sim(Si,Sj) bernilai nol (0).

   Berikut adalah perhitungan pada iterasi ke-1.

   $MMR(Si) = \left(\lambda * \text{Sim}(Si, Q)\right) - ((1 - \lambda) * \text{maxSim}(Si, Sj))$

Tabel 3. 29 Perhitungan MMR Iterasi ke-1

$\boldsymbol{Si}$	$\boldsymbol{\lambda}$	$\textbf{Sim}(\boldsymbol{Si}, \boldsymbol{Q})$	$\textbf{maxSim}(\boldsymbol{Si}, \boldsymbol{Sj})$	$\boldsymbol{MMR}(\boldsymbol{Si})$
S1	0,7	0,87332	0	**0,61133**
S4	0,7	0	0	0
S6	0,7	0	0	0
S8	0,7	0,14988	0	0,10492

Hasil dari perhitungan iterasi ke-1, diperoleh nilai maksimum = 0,61133 pada S1. Sehingga, kalimat 1 akan dipastikan menjadi ringkasan dan akan menjadi acuan untuk perhitungan iterasi selanjutnya.

2. Iterasi ke-2

   Pada iterasi ke-2, nilai maksimum iterasi ke-1 akan digunakan untuk menghitung *similarity* pada maxSim(S_i, S_j) yaitu maxSim(S_i, S_1). Nilai yang akan digunakan dapat dilihat pada Tabel 3.30

Tabel 3. 30 *Cosine Similarity* Kalimat 1 (S1)

	S1
S4	0,04631
S6	0,08087
S8	0,18835

Berikut ini adalah perhitungan untuk iterasi ke-2.

$MMR(Si) = \left(\lambda * \text{Sim}(Si, Q)\right) - ((1 - \lambda) * \text{maxSim}(Si, S1))$

Tabel 3. 31 Perhitungan MMR Iterasi ke-2

Si	λ	$\text{Sim}(Si, Q)$	$\text{maxSim}(Si, Sj)$	$MMR(Si)$
S2	0,7	0	0,04631	0
S3	0,7	0	0,08087	0
S8	0,7	0,14988	0,18835	**0,04841**

Hasil dari perhitungan iterasi ke-2, diperoleh nilai maksimum = 0,04841 pada S8. Sehingga, kalimat 8 akan dipastikan menjadi ringkasan dan akan menjadi acuan untuk perhitungan iterasi selanjutnya.

3. Iterasi ke-3

   Pada iterasi ke-3, untuk menghitung *similarity* pada maxSim(S_i, S_j), dicari dengan membandingkan nilai maksimum *similarity* antara S1 dengan S3. Nilai yang akan digunakan dapat dilihat pada Tabel 3.32.

Tabel 3. 32 *Cosine Similarity* Kalimat 1 (S1) dan Kalimat 8 (S8)

	S1	**S8**
S4	0,04631	**0,17957**
S6	**0,08087**	0,06882

Perbandingan nilai *similarity* maksimum (S_i, S_1) dan (S_i, S_3) adalah sebagai beikut :

- *Similarity* maksimum (S_i, S_1) :

  Sim(S_4, S_1) = 0,04631

  Sim(S_6, S_1) = **0,08087**

  Nilai *similarity* maksimum (S_i, S_1) adalah 0,08087
- *Similarity* maksimum (S_i, S_8) :

  Sim(S_4, S_8) = **0,17957**

  Sim(S_6, S_8) = 0,06882

  Nilai *similarity* maksimum (S_i, S_8) adalah 0,17957
- Jadi, *similarity* maksimum (S_i, S_1) dan (S_i, S_3) adalah 0,17957 pada S8

Berikut ini adalah perhitungan untuk iterasi ke-3.

$$MMR(Si) = \left(\lambda * \text{Sim}(Si, Q)\right) - ((1 - \lambda) * \text{maxSim}(Si, S8))$$

Tabel 3. 33 Perhitungan MMR Iterasi ke-3

Si	λ	$\text{Sim}(Si, Q)$	$\text{maxSim}(Si, Sj)$	$MMR(Si)$
S4	0,7	0	0,17957	0
S6	0,7	0	0,06882	0

Dari hasil perhitungan pada iterasi ke-3, diperoleh nilai maksimum MMR (Si) $\leq$ 0 pada S4 dan S6, sehingga tidak ada kalimat yang dipilih sebagai ringkasan dan iterasi perhitungan akan berhenti.

Setelah diperolehnya nilai MMR pada kalimat terpilih, selanjutnya perankingan MMR dari nilai terbesar ke yang terkecil. Dan dikarenakan pada iterasi-3 mendapatkan maksimalnya 0, sehingga hasil ringkasan hanya sampai iterasi-2.

Perangkingan kalimat dari hasil perhitungan MMR dapat dilihat pada Tabel 3.34

Tabel 3. 34 Hasil perankingan Kalimat

Ranking	Kalimat	Max MMR
1	1	0,61133

2	8	0,04841

3.4. Hasil Ringkasan

Dari hasil perangkingan pada Tabel 3.34 dapat diketahui kalimat yang menjadi ringkasan berdasarkan maksimum MMR dapat dilihat pada Tabel 3.35.

Tabel 3. 35 Ringkasan

Kalimat ke-	Kalimat
1	Jakarta - Presiden Jokowi mengunjungi Terminal 3 Ultimate Bandara Soekarno-Hatta
8	Dirut AP II lalu memberikan beberapa penjelasan kepada Jokowi mengenai terminal 3 Ultimate yang hampir selesai

Sehingga ringkasan yang diperoleh dapat dilihat pada Tabel 3.36.

Tabel 3. 36 Hasil Ringkasan

Sebelum	Sesudah
Jakarta - Presiden Jokowi mengunjungi Terminal 3 Ultimate Bandara Soekarno-Hatta. Presiden meninjau langsung pembangunan terminal yang sudah hampir selesai itu. Presiden tiba di Terminal 3 Ultimate Bandara Soekarno-Hatta, Cengkareng, Rabu (11/5/2016) pukul 17:40 WIB. Saat tiba, Jokowi disambut Menteri BUMN Rini Soemarno dan Dirut Angkasa Pura II Budi Karya. Presiden langsung masuk ke kompleks terminal yang berukuran luas itu. Nampak para pekerja masih sibuk bekerja saat Jokowi datang. Jokowi lalu berkeliling ke terminal yang digadang akan bisa menampung ribuan penumpang itu. Dirut AP II lalu memberikan beberapa penjelasan kepada Jokowi mengenai terminal 3 Ultimate yang hampir selesai.	Jakarta - Presiden Jokowi mengunjungi Terminal 3 Ultimate Bandara Soekarno-Hatta. Dirut AP II lalu memberikan beberapa penjelasan kepada Jokowi mengenai terminal 3 Ultimate yang hampir selesai.

3.5.Analisis Pengujian

Pada tahap analisis pengujian, hasil ringkasan yang diperoleh dengan menggunakan metode KNN (K-*Nearest Neighbors*) dan MMR (*Maximum Marginal Relevance*) akan dibandingkan dengan ringkasan menggunakan metode MMR (*Maximum Marginal Relevance*). Hasil perbandingkan ringkasan dapat dilihat pada Tabel 3.37.

Tabel 3. 37 Perbandingan Hasil Ringkasan

Hasil Ringkasan		
KNN + MMR	**MMR**	**Manual**
Jakarta - Presiden Jokowi mengunjungi Terminal 3 Ultimate Bandara Soekarno-Hatta. Dirut AP II lalu memberikan beberapa penjelasan kepada Jokowi mengenai terminal 3 Ultimate yang hampir selesai.	Jakarta - Presiden Jokowi mengunjungi Terminal 3 Ultimate Bandara Soekarno-Hatta. Presiden meninjau langsung pembangunan terminal yang sudah hampir selesai itu. Presiden tiba di Terminal 3 Ultimate Bandara Soekarno-Hatta, Cengkareng, Rabu (11/5/2016) pukul 17:40 WIB. Presiden langsung masuk ke kompleks terminal yang berukuran luas itu. Dirut AP II lalu memberikan beberapa penjelasan kepada Jokowi mengenai terminal 3 Ultimate yang hampir selesai.	Jakarta - Presiden Jokowi mengunjungi Terminal 3 Ultimate Bandara Soekarno-Hatta. Presiden meninjau langsung pembangunan terminal yang sudah hampir selesai itu. Presiden tiba di Terminal 3 Ultimate Bandara Soekarno-Hatta, Cengkareng, Rabu (11/5/2016) pukul 17:40 WIB. Dirut AP II lalu memberikan beberapa penjelasan kepada Jokowi mengenai terminal 3 Ultimate yang hampir selesai.

Untuk mengukur pengujian pada penelitian *text summarization* ini menggunakan perhitungan nilai performansi (*recall*, *precision* dan *f-measure*) dengan membandingkan ringkasan yang dihasilkan KNN dan MMR dengan manuak, dan ringkasan pembandingnya menggunakan MMR dengan manual adalah sebagi berikut.

1. Perbandingan ringkasan KNN dan MMR dengan ringkasan manual

   Didapatkan nilai parameter tp, fp, fn, dan tn, yaitu:

   a. tp (banyak kalimat yang ada di KNN+MMR dan manual) = 2
   b. fp (banyak kalimat yang hanya ada di KNN+MMR) = 0
   c. fn (banyak kalimat yang hanya ada di manual) = 2
   d. tn (banyak kalimat yang tidak ada di KNN+MMR) = 6

   Berdasarkan persamaan (2.5) dalam menghitung nilai *recall***,** maka nilai *recall* yang dihasilkan yaitu:

   $$recall = \frac{tp}{tp + fn} = \frac{2}{2 + 2} = \frac{2}{4} = 0{,}5$$

   $$presentase\ recall = \ 0{,}5\ x\ 100\% = 50\%$$

   Berdasarkan persamaan (2.6) dalam menghitung nilai *precision***,** maka nilai *precision* yang dihasilkan yaitu:

   $$precision = \frac{tp}{tp + fp} = \frac{2}{2 + 0} = \frac{2}{2} = 1$$

   $$presentase\ precision = \ 1\ x\ 100\% = 100\%$$

   Berdasarkan persamaan (2.7) dalam menghitung nilai *f-measure***,** maka nilai *f-measure* yang dihasilkan yaitu:

   $$F - meansure = \frac{2\ x\ recall\ x\ precision}{recall + \ precision} = \frac{2\ x\ 0{,}5\ x\ 1}{0{,}5 + 1} = \frac{1}{1{,}5} = 0{,}67$$

   $$presentase\ F - meansure = \ 0{,}67\ x\ 100\% = 67\%$$

   Sehingga akurasi yang dihasilkan yaitu :

   $$Akurasi = \frac{tp + tn}{tp + fp + fn + tn} = \frac{2 + 6}{2 + 0 + 2 + 6} = \frac{8}{10} = 0{,}8$$

   $$presentase\ Akurasi = \ 0{,}8\ x\ 100\% = 80\%$$

2. Perbandingan ringkasan MMR dengan ringkasan manual

   Didapatkan nilai parameter tp, fp, fn, dan tn, yaitu:

a. tp (banyak kalimat yang ada di MMR dan manual) = 3
b. fp (banyak kalimat yang hanya ada di MMR) = 1
c. fn (banyak kalimat yang hanya ada di manual) = 0
d. tn (banyak kalimat yang tidak ada di MMR) = 2

Berdasarkan persamaan (2.5) dalam menghitung nilai *recall***,** maka nilai *recall* yang dihasilkan yaitu:

$$recall = \frac{tp}{tp + fn} = \frac{3}{3 + 0} = \frac{3}{3} = 1$$

$$presentase\ recall = 1\ x\ 100\% = 100\%$$

Berdasarkan persamaan (2.6) dalam menghitung nilai *precision***,** maka nilai *precision* yang dihasilkan yaitu:

$$precision = \frac{tp}{tp + fp} = \frac{3}{3 + 1} = \frac{3}{4} = 0{,}75$$

$$presentase\ precision = 0{,}75\ x\ 100\% = 75\%$$

Berdasarkan persamaan (2.7) dalam menghitung nilai *f-measure***,** maka nilai *f-measure* yang dihasilkan yaitu:

$$F - meansure = \frac{2\ x\ recall\ x\ precision}{recall + precision} = \frac{2\ x\ 1\ x\ 0{,}75}{1 + 0{,}75} = \frac{1{,}5}{1{,}75} = 0{,}86$$

$$presentase\ F - meansure = 0{,}86\ x\ 100\% = 86\%$$

Sehingga akurasi yang dihasilkan yaitu :

$$Akurasi = \frac{tp + tn}{tp + fp + fn + tn} = \frac{3 + 2}{3 + 1 + 0 + 2} = \frac{5}{6} = 0{,}83$$

$$presentase\ Akurasi = 0{,}83\ x\ 100\% = 83\%$$

3. Kesimpulan perbandingan ringkasan KNN dan MMR dengan ringkasan MMR

Setelah dihitung perbandingan KNN dan MMR dengan manual, dan MMR dengan manual, maka hasil keseluruhan dari perhitungan yang dilakukan dapat dilihat pada Tabel 3.38

Tabel 3. 38 Nilai *Recall*, *Precision*, F-*measure*, dan Akurasi

MMR+KNN				**MMR**			
Recall	***Precision***	***F-measure***	***Akurasi***	***Recall***	***Precision***	***F-measure***	***Akurasi***
50%	100%	67%	80%	100%	75%	86%	83%

3.6.Analisis Kebutuhan Non Fungsional

Kebutuhan non fungsional pada penelitian ini terdiri dari kebutuhan perangkat keras, kebutuhan perangkat lunak, dan kebutuhan pengguna.

3.6.1. Analisis Kebutuhan Perangkat Keras

Kebutuhan perangkat keras yang digunakan dalam pembangunan *text summarization* menggunakan metode MMR pada artikel berbahasa Indonesia adalah sebagai berikut:

Tabel 3. 39 Kebutuhan Perangkat Keras

Perangkat Keras	**Spesifikasi**
Processor	Intel(R) core(TM) i3
RAM	2048 MB
Monitor	15 inch
Keyboard	Standar
Mouse	Standar

3.6.2. Analisis Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak yang digunakan dalam pembangunan *text summarization* menggunakan metode MMR pada artikel berbahasa Indonesia adalah sebagai berikut:

Tabel 3. 40 Kebutuhan Perangkat Lunak

Perangkat Lunak	**Spesifikasi**
Sistem Operasi	Windows 7 Ultimate 32 bit
Browser	Mozila Firefox
Web Server	XAMPP
HTML Editor	Macromedia Dreamweaver 8

3.6.3. Analisis Kebutuhan Pengguna

Kebutuhan pengguna pada *text summarization* menggunakan metode MMR pada artikel berbahasa Indonesia adalah sebagai berikut:

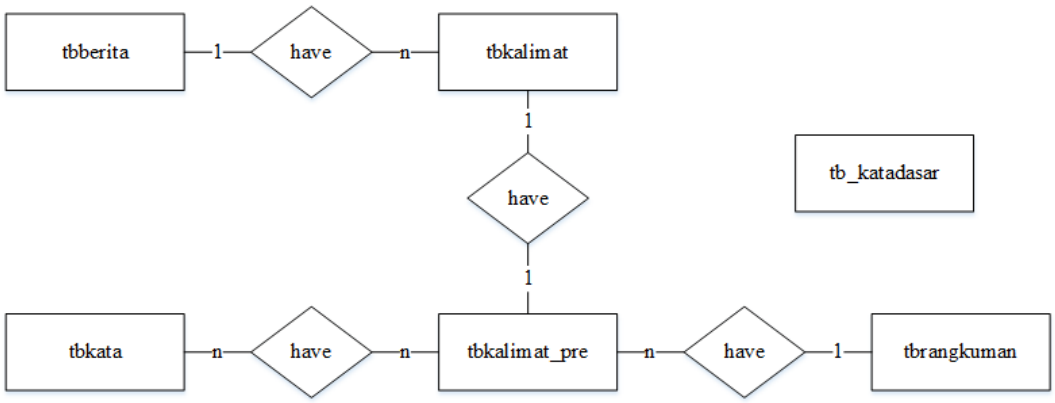
a. Menguasai penggunaan komputer.
b. Mengerti secara teknis *tools* dan *software* pendukung dalam menjalankan aplikasi.
c. Mengerti tahap-tahap dalam menjalankan aplikasi *text summarization*.

3.7.Analisis Kebutuhan Fungsional

Kebutuhan fungsional pada penelitian ini, terdiri dari ERD, diagram konteks, DFD level 1, DFD level 2, deskripsi proses, dan kamus data.

3.7.1.*Entity Relationship Diagram* (ERD)

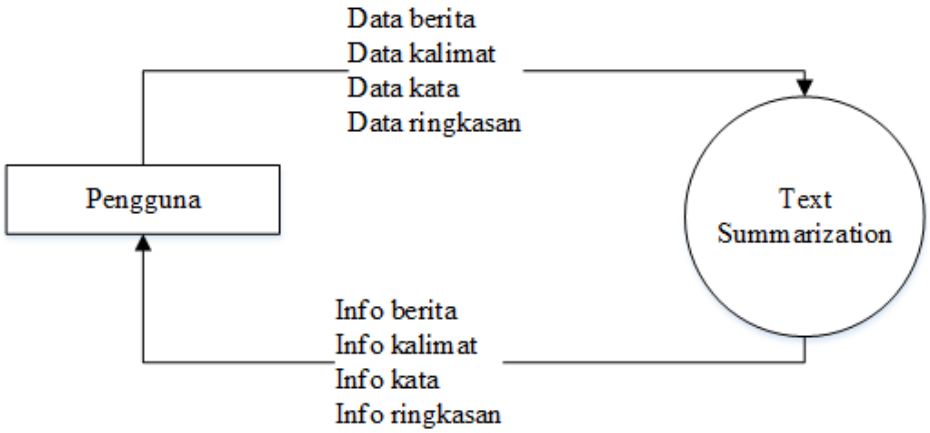
Entity relationship diagram adalah diagram yang menggambarkan hubungan antar data dalam basis data pada sistem *text summarization* arikel berbahasa Indonesia. Tabel digambarkan dengan kotak, sedangkan relasi digambarkan dengan *diamonds*. *Entity relationship diagram* pada *text summarization* arikel berbahasa Indonesia dapat dilihat pada Gambar 3.10.



Gambar 3. 10 ERD

3.7.2. Diagram Konteks

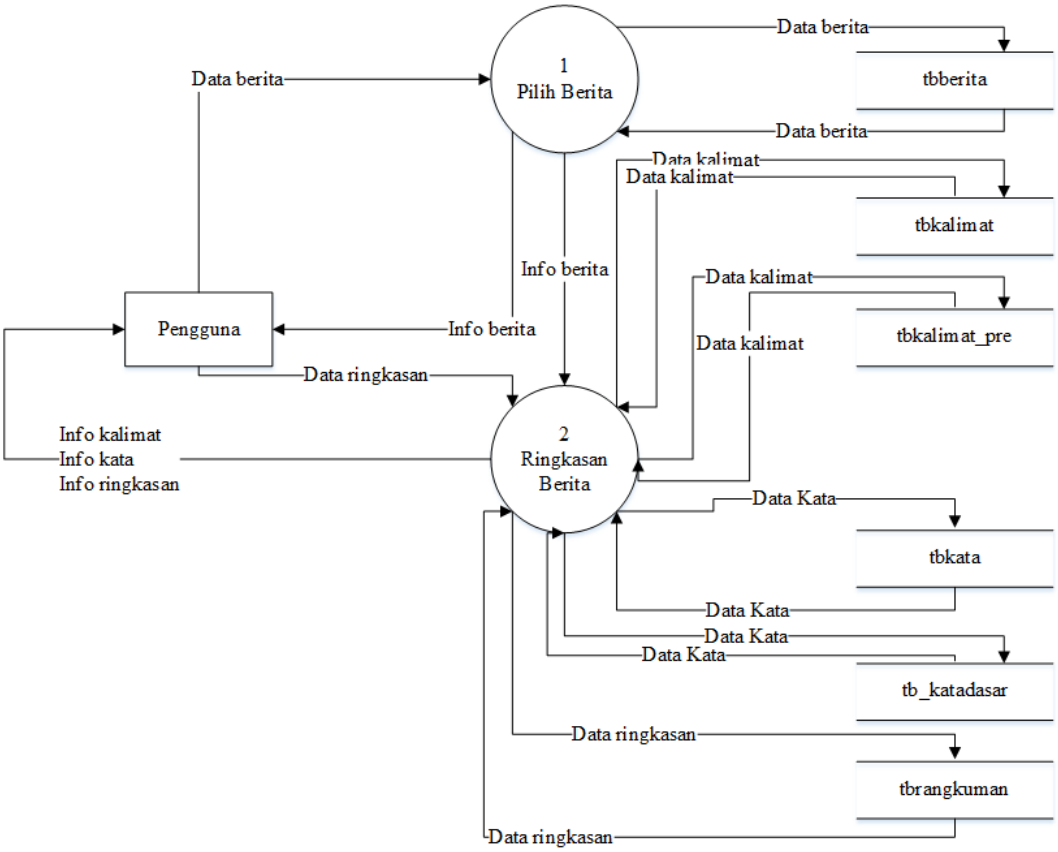
Diagram konteks adalah suatu diagram yang menggambarkan ruang lingkup *text summarization* artikel berbahasa Indonesia. Sistem digambarkan dengan bulatan, sedangkan lingkungan diwakili oleh entitas luar yang digambarkan dengan persegi. Gambar 3.11 menggambarkan diagram konteks yang terdapat sistem *text summarization*.



Gambar 3. 11 Diagram konteks

3.7.3. *Data Flow Diagram* (DFD) Level 1

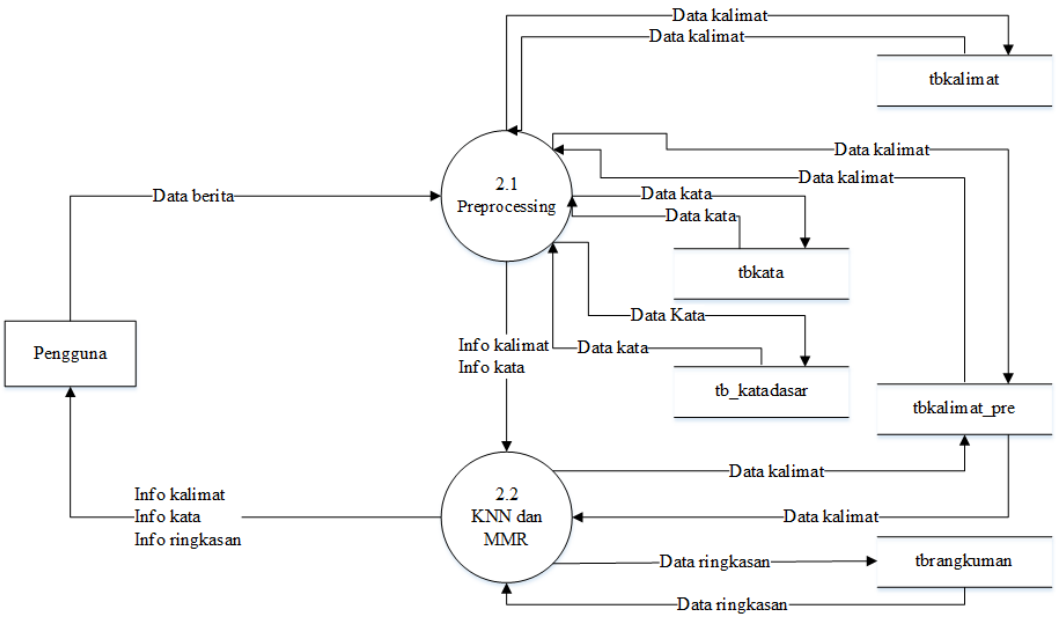
Data flow diagram level 1 berfungsi untuk menjelaskan aliran data yang terdapat di Sistem *Text Sumarization*. DFD level 1 dapat dilihat pada Gambar 3.12.



Gambar 3. 12 DFD Level 1

3.6.4. *Data Flow Diagram* (DFD) Level 2

Data flow diagram level 2 berfungsi untuk menjelaskan aliran data yang terdapat pada proses 2 Ringkasan Berita. DFD level 2 dapat dilihat pada Gambar 3.13.



Gambar 3. 13 DFD Level 2 Proses 2 Ringkasan Berita

3.7.5. Deskripsi Proses

Deskripsi proses digunakan untuk menggambarkan proses model aliran yang terdapat pada DFD. Spesifikasi proses dari DFD yang telah dibuat dapat dijelaskan pada tabel deskripsi proses level 1 dan level 2. Deskripsi proses dapat dilihat pada Tabel 3.41.

Tabel 3. 41 Deskripsi Proses

No	Proses	Keterangan
1.	*Process Number*	1
	Process Name	Pilih berita
	Source	Pengguna
	Input	Data berita
	Output	Info berita
	Destination	Pengguna
	Logic Process	1. Sistem menampilkan pilihan berita 2. Pengguna memilih dan mengklik berita yang akan diringkas 3. Sistem akan menampilkan hasil ringakasn
2.	*Process Number*	2.1
	Process Name	*Preprocessing*

	Source	Pengguna
	Input	Data berita, data kalimat, data kata
	Output	Info berita, info kalimat, info kata
	Destination	Pengguna
	Logic Process	1. Sistem telah mengambil data berita yang telah dipilih 2. Sistem akan mengubah semua huruf menadi huruf kecil (*case folding)* 3. Sistem akan menghapus semua simbol dan angka, kecuali huruf, spasi, dan titik (*filtering*) 4. Sistem akan memisahkan teks menjadi satuan kalimat dengan delimeter titik (*tokenizing* kalimat) 5. Sistem akan memisahkan kalimat menjadi sebuah kata dengan delimeter spasi (*tokenizing* kata) 6. Sistem akan mengecek dan menghapus kata yang termasuk ke dalam daftar *stopword* (*removal stopword*) 7. Sistem akan mengecek dan mengganti kata menjadi kata dasar (*stemming*) 8. Sistem akan menghitung bobot setiap kata (tf-idf) 9. Sistem akan menghitung similaritas antar kalimat dan *query* dengan kalimat
3.	*Process Number*	2.2
	Process Name	KNN dan MMR
	Source	Pengguna
	Input	Data kalimat, data ringkasan
	Output	Info kalimat, info ringkasan
	Destination	Pengguna
	Logic Process	1. Sistem akan mengambil similaritas antar kalimat dan menghitung rata-rata similaritas 2. Sistem akan mengklasifikasikan kalimat ke dalam dua kelas 3. Sistem akan mengambil kalimat yang termasuk ke dalam kelas relevan saja 4. Sistem akan menghitung similarity kalimat dengan *query* 5. Sistem akan menghitung dan menentukan kalimat tersebut termasuk ke dalam ringkasan atau tidak 6. Sistem akan mengurutkan kalimat yang akan menjadi ringkasn

3.7.6. Kamus Data

Kamus data merupakan deskripsi formal dari seluruh elemen atau aliran data yang terdapat dalam *data flow diagram*. Kamus data *text summarization* dapat dilihat pada Tabel 3.42, Tabel 3.43, Tabel 3.44, Tabel 3.45, Tabel 3.46, Tabel 3.47, Tabel 3.48, dan Tabel 3.49.

Tabel 3. 42 Data Berita

Nama Aliran Data	Data berita
Digunakan pada	Pilih berita (1), *Preprocessing* (2.1)
Deskripsi	Data yang akan di proses dalam peringkasan
Struktur data	Teks
Penjelasan per struktur	[A..Z], [a..z], [0..9], symbol

Tabel 3. 43 Info berita

Nama Aliran Data	Info berita
Digunakan pada	Pilih berita (1), *Preprocessing* (2.1)
Deskripsi	Data yang dipilih untuk di proses dalam peringkasan
Struktur data	Teks
Penjelasan per struktur	[A..Z], [a..z], [0..9], symbol

Tabel 3. 44 Data Kalimat

Nama Aliran Data	Data kalimat
Digunakan pada	*Preprocessing* (2.1), KNN dan MMR (2.2)
Deskripsi	Data yang akan diproses untuk pemisahan kalimat, perhitungan kalimat
Struktur data	Teks
Penjelasan per struktur	[A..Z], [a..z], [0..9], symbol

Tabel 3. 45 Info Kalimat

Nama Aliran Data	Info kalimat
Digunakan pada	*Preprocessing* (2.1), KNN dan MMR (2.2)
Deskripsi	Data hasil pemrosesan pemisahan kalimat, nilai kalimat
Struktur data	Teks

Penjelasan per struktur	[A..Z], [a..z], [0..9], symbol

Tabel 3. 46 Data Kata

Nama Aliran Data	Data kata
Digunakan pada	*Preprocessing* (2.1)
Deskripsi	Data yang diproses dalam pemisahan kata dan pemilihan kata
Struktur data	Teks
Penjelasan per struktur	[A..Z], [a..z], [0..9], symbol

Tabel 3. 47 Info Kata

Nama Aliran Data	Info kata
Digunakan pada	*Preprocessing* (2.1)
Deskripsi	Data hasil pemrosesan pemisahan kata dan pemilihan kata
Struktur data	Teks
Penjelasan per struktur	[A..Z], [a..z], [0..9], symbol

Tabel 3. 48 Data Ringkasan

Nama Aliran Data	Data ringkasan
Digunakan pada	KNN dan MMR (2.2)
Deskripsi	Data yang akan dipilih untuk dijadikan ringkasan
Struktur data	Teks
Penjelasan per struktur	[A..Z], [a..z], [0..9], symbol

Tabel 3. 49 Info Ringkasan

Nama Aliran Data	Info ringkasan
Digunakan pada	KNN dan MMR (2.2)
Deskripsi	Data hasil dari peringkasan
Struktur data	Teks
Penjelasan per struktur	[A..Z], [a..z], [0..9], symbol

3.8.Perancangan Sistem

Perancangan sistem pada penelitian ini terdiri dari perancangan basis data, perancangan struktur menu, perancangan antarmuka, dan jaringan semantik.

3.8.1.Perancangan Basis Data

Perancangan basis data pada penelitian ini terdiri dari beberapa struktur tabel.

3.8.1.1. Struktur Tabel

Struktur tabel pada perancangan basis data dapat dilihat pada Tabel 3.50, Tabel 3.51, Tabel 3.52, Tabel 3.53, Tabel 3.54, Tabel 3.55, dan Tabel 3.56.

Tabel 3. 50 Struktur Tabel tbberita

No	Atribut	Tipe Data	Keterangan
1.	id_berita	Int	Primary Key
2.	judul	Varchar(100)	
3.	berita	Varchar(5000)	

Tabel 3. 51 Struktur Tabel tbkalimat

No	Atribut	Tipe Data	Keterangan
1.	id_kalimat	Int	Primary Key
2.	kalimat	Varchar(1000)	
3.	id_berita	Int	Foreign Key

Tabel 3. 52 Struktur Tabel tbkalimat_pre

No	Atribut	Tipe Data	Keterangan
1.	id_kalimat_pre	Int	Primary Key
2.	id_kalimat	Int	Foreign Key
3.	kalimat_pre	Varchar(1000)	
4.	vektor	Float	
5.	bobot	Float	
6.	simcos	Float	

Tabel 3. 53 Struktur Tabel tbkata

No	Atribut	Tipe Data	Keterangan
1.	id_kata	Int	Primary Key
2.	id_kalimat_pre	Int	Foreign Key
3.	kata	Varchar(30)	

4.	count	Int	
5.	idf	Float	
6.	bobot	Float	

Tabel 3. 54 Struktur Tabel tb_katadasar

No	Atribut	Tipe Data	Keterangan
1.	id_katadasar	Int	Primary Key
2.	katadasar	Varchar(50)	
3.	tipe_katadasar	Varchar(50)	

Tabel 3. 55 Struktur Tabel tbrangkuman

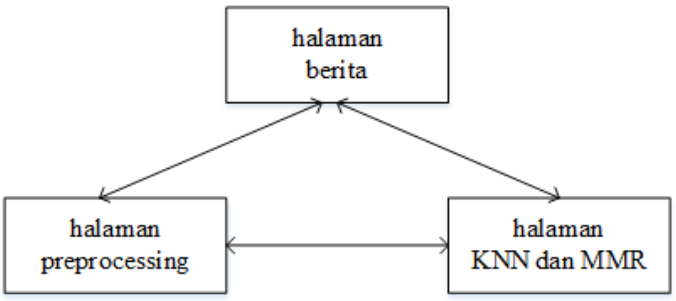
No	Atribut	Tipe Data	Keterangan
1.	id_rangkuman	Int	Primary Key
2.	id_kalimat_pre	Int	Foreign Key
3.	bobotQuery	Float	
4.	vektorQuery	Float	
5.	vektor	Float	
6.	hasil	Float	

Tabel 3. 56 Struktur Tabel tbw (Transaksi tbkata dengan tbkalimat_pre)

No	Atribut	Tipe Data	Keterangan
1.	id_kalimat_pre1	Int	Foreign Key
2.	id_kalimat_pre2	Int	Foreign Key
3.	id_kata	Int	Foreign Key
4.	bobot1	Float	
5.	bobot2	Float	
6.	hasil_bobot	Float	

3.8.2. Perancangan Struktur Menu

Struktur menu yang digunakan dalam pembangunan *text summarization* menggunakan metode MMR pada artikel berbahasa Indonesia. Struktur menu penelitian ini dapat dilihat pada Gambar 3.14.



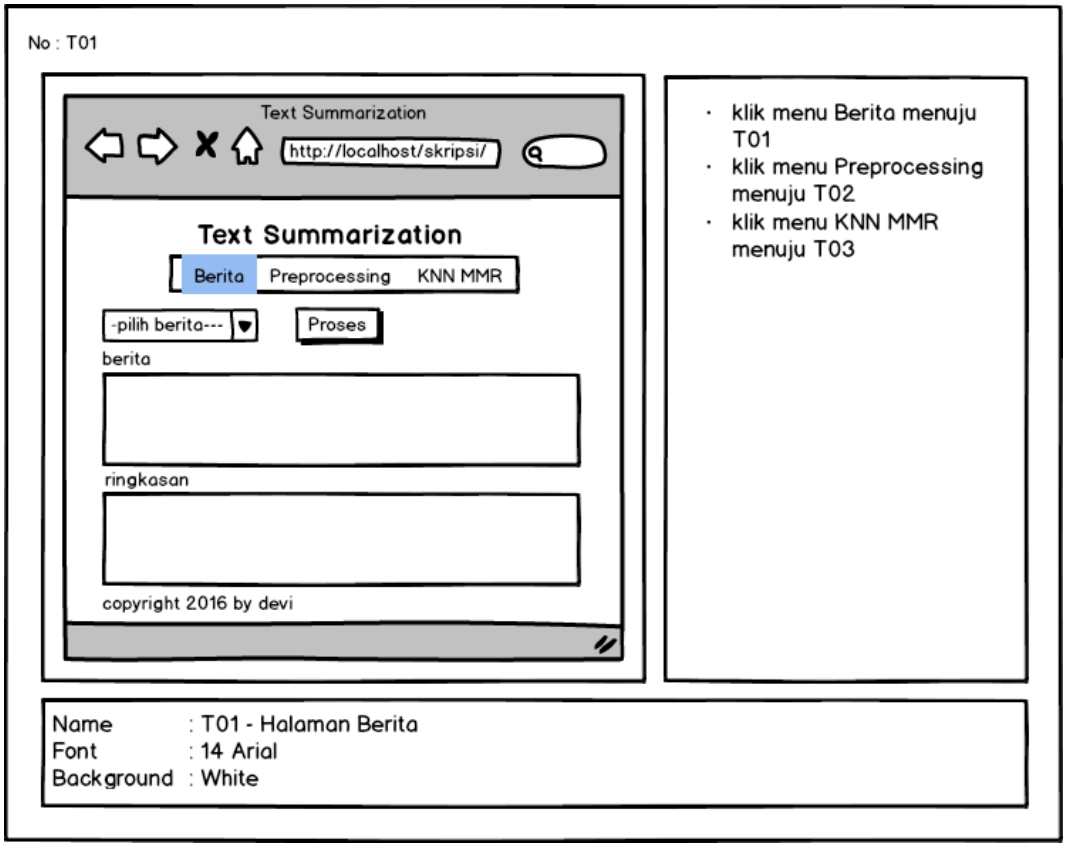
Gambar 3. 14 Struktur Menu

3.8.3.Perancangan Antarmuka

Perancangan antarmuka mendeskripsikan rencana tampilan yang akan digunakan pada aplikasi yang akan dibangun. Perancangan antarmuka pada penelitian ini terdiri dari antarmuka halaman beranda, antarmuka halaman berita, antarmuka halaman kalimat, dan antarmuka halaman kata.

3.8.3.1. Antarmuka Halaman Berita (T01)

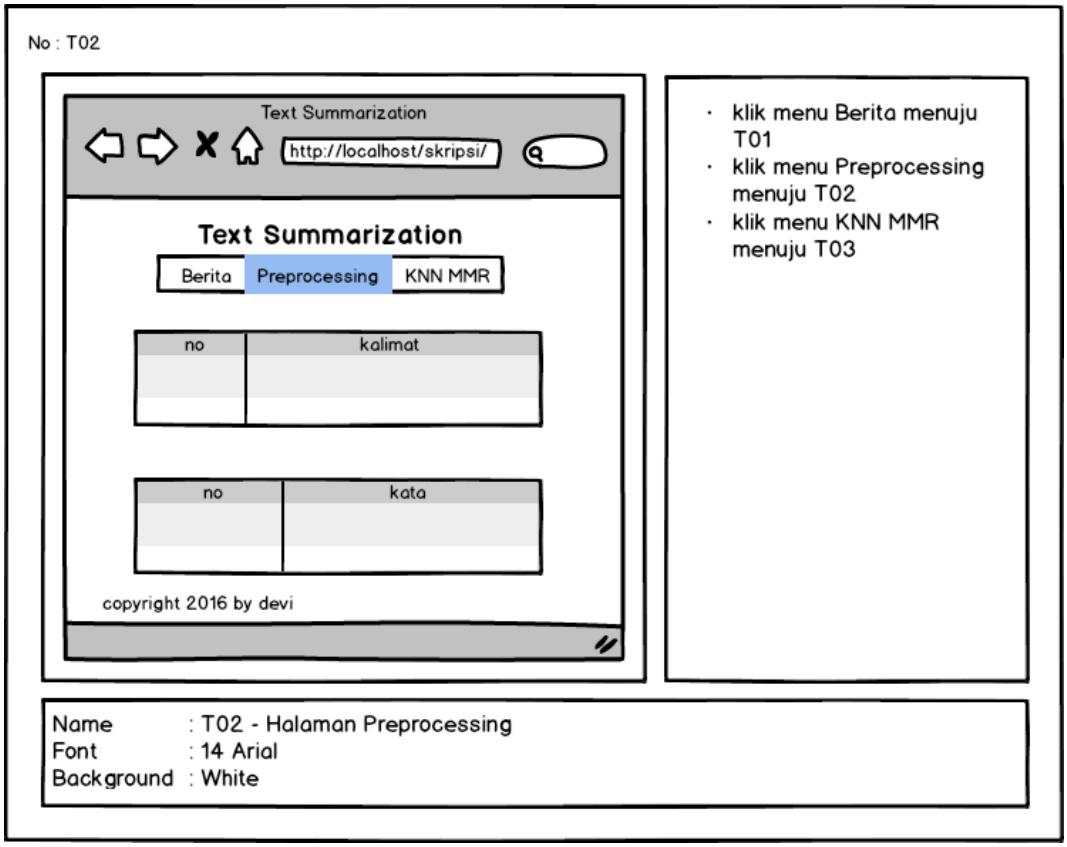
Perancangan T01 merupakan halaman yang menampilkan teks asli dan teks ringkasan. Perancangan antarmuka halaman berita dapat dilihat pada Gambar 3.15



Gambar 3. 15 Perancangan Halaman Berita

3.8.3.2. Antarmuka Halaman *Preprocessing* (T02)

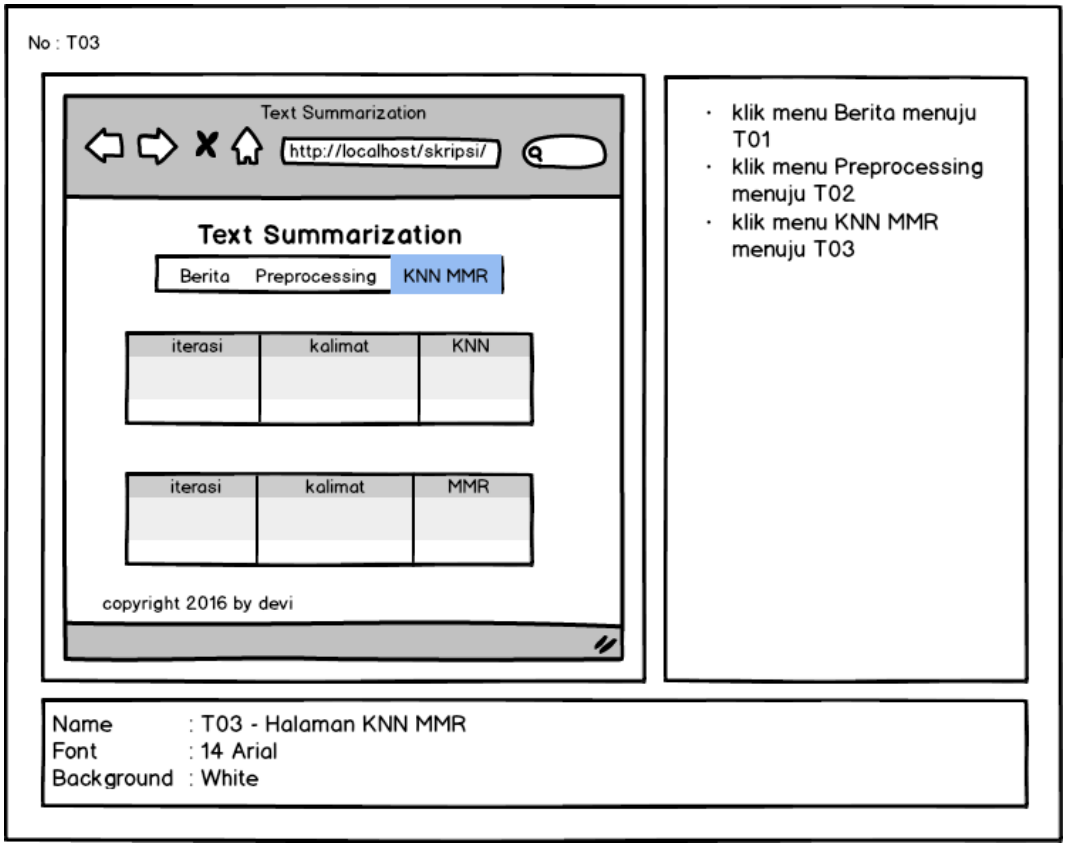
Perancangan T02 merupakan halaman hasil dari *preprocessing*. Perancangan antarmuka halaman *prepocessing* dapat dilihat pada Gambar 3.16.



Gambar 3. 16 Perancangan Halaman *Preprocessing*

3.8.3.3. Antarmuka Halaman KNN MMR (T03)

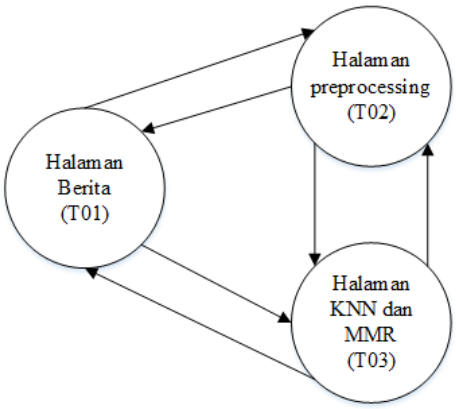
Perancangan T03 merupakan halaman menampilkan hasil dari KNN MMR. Perancangan antarmuka halaman KNN MMR dapat dilihat pada Gambar 3.17.



Gambar 3. 17 Perancangan Halaman KNN MMR

3.8.4. Jaringan Semantik

Jaringan semantik menggambarkan keterhubungan navigasi menu dari satu antarmuka ke antarmuka lain. Jaringan semantik yang terbentuk pada sistem ini dapat dilihat pada Gambar 3.18.

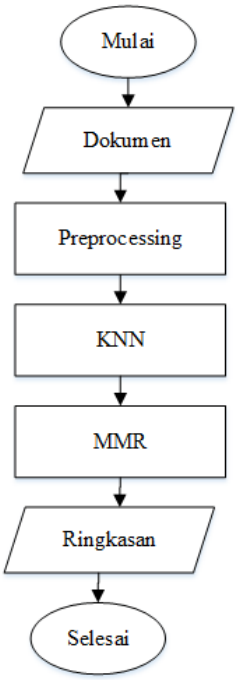


Gambar 3. 18 Jaringan Sematik

3.8.5.Perancangan Prosedural

3.8.5.1.Perancangan Algoritma Utama

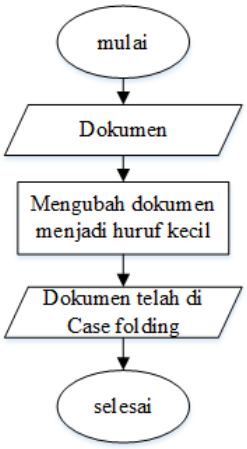
User akan memilih berita (dokumen) yang akan di ringkas. Selanjutnya sistem akan memrosesnya dengan dua tahapan, yaitu *preprocessing* dan MMR. Pada tahap *preprocessing* dokumen yang dimasukkan diproses dengan *filtering*, *case folding*, *tokenizing* kalimat, *tokenizing* kata, *removal stopword*, *stemming*, TF-IDF, *cosine similarity*, dan KNN. Setelah dilakukannya tahap *preprocessing* kemudian sistem masuk ke tahap MMR, yaitu dengan menghitung kerelevanan antara *query* dan antarkalimat. Dan sistem akan menghasilkan sebuah ringkasan.



Gambar 3. 19 *Flowchart* Sistem *Text Summarization* Menggunakan KNN dan MMR

3.8.5.2.Perancangan Algoritma *Case Folding*

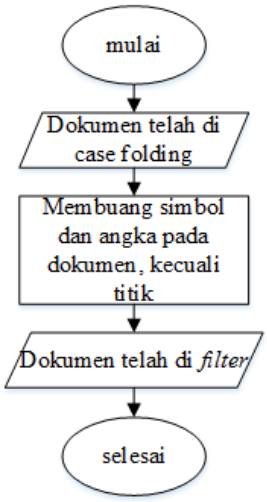
Proses *case folding* ini dimulai dari dokumen yang telah *user* pilih. Kemudian sistem akan mengubah semua huruf pada dokumen menjadi huruf kecil. Sehingga sistem akan menghasilkan dokumen dengan huruf *lowcase*.



Gambar 3. 20 *Flowchart Case Folding*

3.8.5.3.Perancangan Algoritma *Filtering*

Selanjutnya proses *filtering*, dengan masukkannya dokumen yang telah di *filter* pada proses sebelumnya. Pada proses ini, sistem akan menghapus semua simbol dan angka pada dokumen, kecuali titik. Sehingga sistem akan menghasilkan dokumen yang isinya hanya terdiri dari huruf (A-Z, a-z), spasi (“ ”), dan titik (“.”).

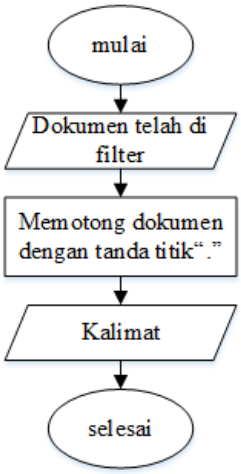


Gambar 3. 21 *Flowchart Filtering*

3.8.5.4.Perancangan Algoritma *Tokenizing* Kalimat

Selanjutnya proses *tokenizing* kalimat, dengan masukkannya dokumen yang telah di *case foldding* pada prses sebelumnya. Pada proses ini, sistem akan

membagi-bagi dokumen dengan *delimeter* titik (“.”) menjadi satuan kalimat. Sehingga sistem akan menghasilkan kalimat-kalimat.



Gambar 3. 22 *Flowchart Tokenzing* Kalimat

3.8.5.5.Perancangan Algoritma *Tokenizing* Kata

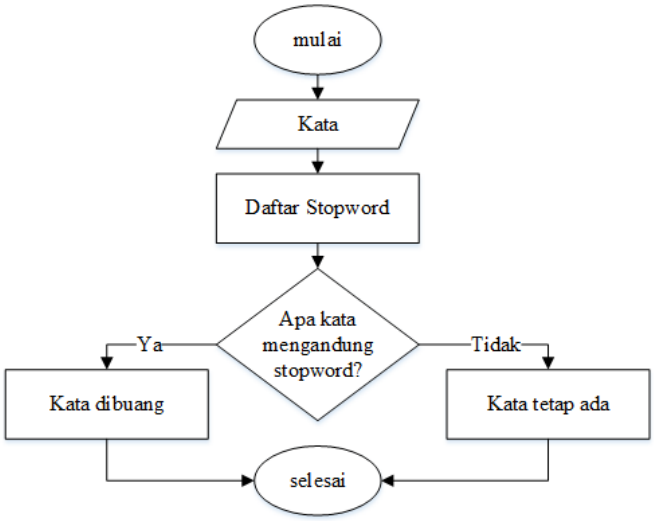
Selanjutnya proses *tokenizing* kata, dengan masukkannya kalimat yang telah diproses sebelumnya. Pada proses ini, sistem akan membagi-bagi kalimat dengan *delimeter* spasi (“ ”) menjadi satuan kata. Sehingga sistem akan menghasilkan kata-kata.



Gambar 3. 23 *Flowchart Tokenizing* Kata

3.8.5.6.Perancangan Algoritma *Removal Stopword*

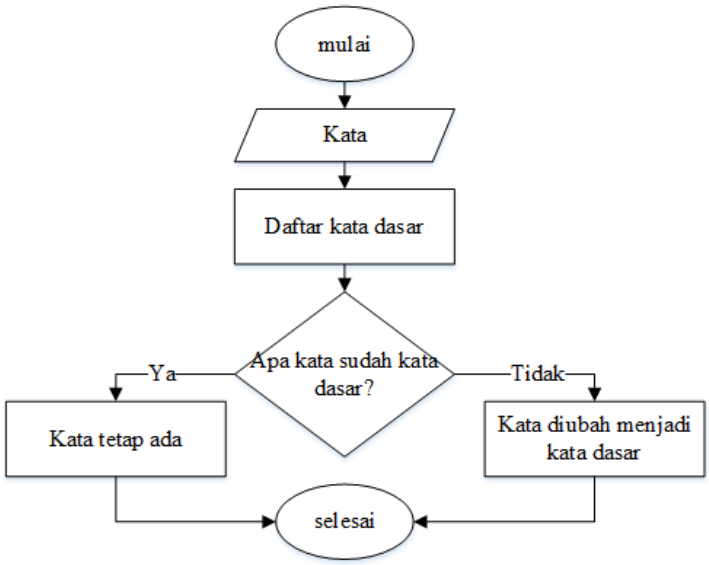
Selanjutnya proses *removal stopword*, dengan masukkannya kata yang telah di proses sebelumnya. Pada proses ini, sistem akan mengecek kata dengan daftar *stopword* bisa dilihat pada lampiran daftar *stopword*. Apabila kata tersebut termasuk kedalam daftar *stopword* maka sistem akan membuang kata tersebut. Namun apabila kata tidak termasuk ke dalam daftar *stopword*, maka sistem akan tetap menyimpan kata tersebut, sehingga kata masih tetap ada.



Gambar 3. 24 *Flowchart Removal Stopword*

3.8.5.7.Perancangan Algoritma *Stemming*

Selanjutnya proses *stemming*, dengan masukkannya kata yang telah terseleksi pada proses sebelumnya. Pada proses ini, sistem akan mengecek kata dengan daftar kata dasar. Apabila kata tersebut termasuk kedalam daftar kata dasar maka sistem akan tetap menyimpan kata tersebut, sehingga kata tetap. Namun apabila kata tersebut tidak termasuk ke dalam daftar kata dasar maka sistem akan mengubah menjadi kata dasar.

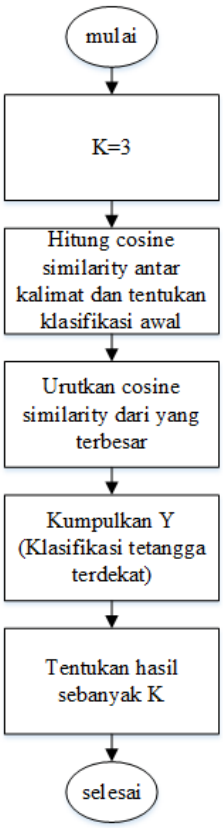


Gambar 3. 25 *Flowchart Stemming*

3.8.5.8.Perancangan Algoritma KNN (*K-Nearest Neighbors*)

Pada tahap KNN (K-*Nearest Neighbors*) ini tujuannya adalah untuk memilih kalimat mana yang relevan dengan kalimat lain. Sehingga kalimat yang terpilih akan diproses pada proses selanjutnya.

Proses KNN dimulai dari menentukan nilai K. Pada penelitian *text summarization* ini parameter K bernilai 3 (K=3). Selanjutnya perhitungan similaritas antar kalimat dengan menggunakan *cosine similarity*, dan penetuan klasifikasi awal ditentukan dari rata-rata *cosine similarity*, yaitu apabila diatas rata-rata *cosine similarity* maka masuk ke dalam klasifikasi relevan dan apabila dibawah rata-rata *cosine similarity* maka masuk ke dalam klasifikasi tidak relevan. Kemudian *cosine similarity* diurutkan secara *descending* (dari terbesar sampai terkecil). Selanjutnya tiap kalimat ambil 3 kalimat terdekatnya untuk menghasilkan klasifikasi yang baru.



Gambar 3. 26 *Flowchart* KNN