BAB 3
ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Masalah

Pelaksanaan skripsi pada program studi Teknik Informatika UNIKOM setiap tahunnya, berdampak pada laporan skripsi yang semakin bertambah dan mengakibatkan penumpukan dokumen yang banyak dalam penyimpanan. Kondisi tersebut bisa memberi kemudahan dalam mendapatkan informasi, namun disisi lain informasi akan semakin bervariasi sehingga kesalahan sering terjadi dalam hal menentukan kategori laporan sesuai topik keilmuan. Penentuan keilmuan dokumen skripsi bisa saja diklasifikasi secara manual, namun dengan jumlah dokumen yang banyak akan menimbulkan kesalahan (*human error)* dalam menentukan kelompok keilmuan pada laporan skripsi serta lebih lama dalam proses pengerjaannya. Faktor pemicu kesalahan dalam menentukan kelompok keilmuan karena masih berkaitannya satu kelompok keilmuan dengan kelompok keilmuan yang lain. Untuk melakukan identifikasi laporan skripsi dapat dilakukan dengan memanfaatkan metode klasifikasi.

Penelitian yang berkaitan dengan klasifikasi dokumen sudah pernah dilakukan sebelumnya. Penelitian yang dilakukan oleh Yosep Amin mengenai "Implementasi Algoritma *Fuzzy* K-*Nearest Neighbour* (*Fuzzy* K-NN) Untuk Klasifikasi Proposal Skripsi Berdasarkan Kelompok Keilmuan Di Teknik Informatika UNIKOM"[12], menunjukkan bahwa berdasarkan hasil penelitian, analisis, pelatihan dan pengujian algoritma *Fuzzy* K-*nearest neighbour* untuk klasifikasi proposal skripsi berdasarkan kelompok keilmuan di Teknik Informatika unikom dengan data latih yang digunakan sebanyak 150 buah dan data uji sebanyak 67 buah, didapatkan nilai keakuratan yang terbesar yakni 51% pada nilai K=7 dan K=8. Dari penelitian tersebut didapatkan akurasi yang masih rendah, sehingga masih dibutuhkan solusi lain agar mendapatkan akurasi yang lebih baik.

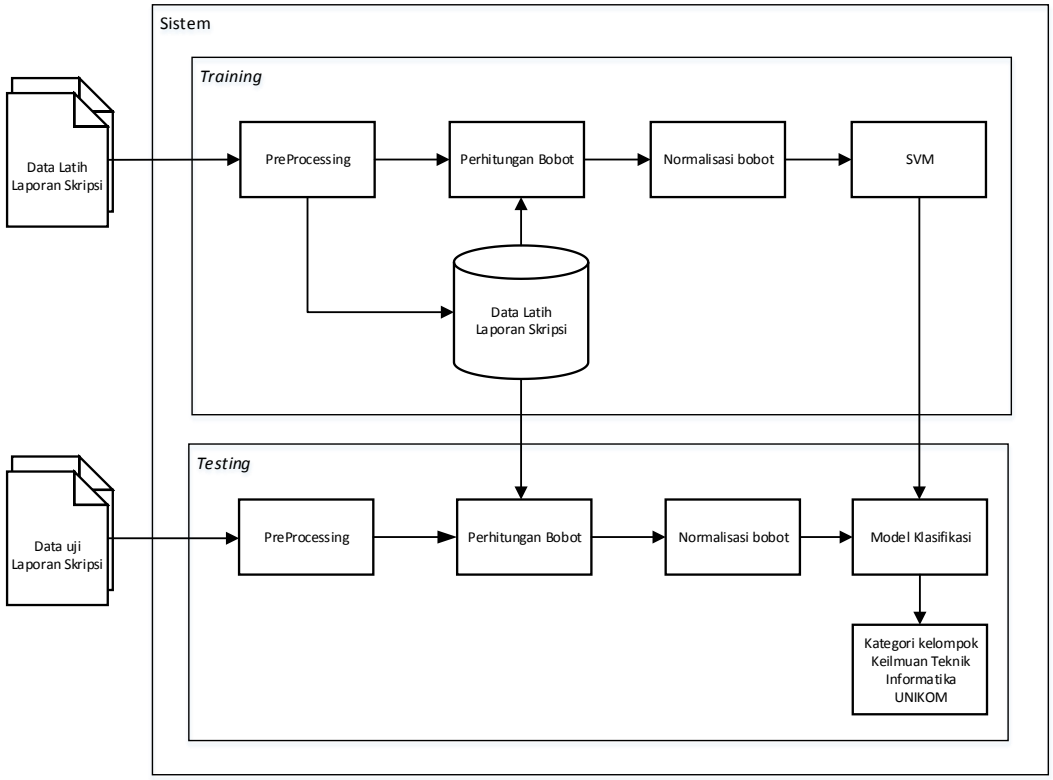
Pada penelitian ini dilakukan dengan menggunakan algoritma SVM dalam pengklasifikasiannya, dan penelitian yang terkait klasifikasi dokumen teks dengan algoritma SVM pernah dilakukan sebelumnya. Penelitian yang dilakukan oleh

Arini Daribti Putri dengan judul "Klasifikasi Dokumen Teks Menggunakan Metode Support Vector Machine Dengan Pemilihan Fitur Chi-Square"[?] menunjukkan bahwa pemilihan algortima SVM memberikan hasil akurasi yang baik untuk klasifikasi dokumen teks dengan dua kelas. Koleksi dokumen yang digunakan sebanyak 457 dokumen dibagi menjadi 70% dokumen latih (320 dokumen) dan 30% dokumen uji (137 dokumen) yang merupakan dokumen berbahasa Indonesia dengan format XML. Dalam penelitian tersebut pada tahap preprocessing tidak digunakan *stemming* dan *casefolding*, Pada pengujian dengan menggunakan kernel linear dan polinomial dihasilkan nilai akurasi sama baik, yaitu 96.35% dan pada kernel RBF dihasilkan akurasi sebesar 95.62%. Dalam penelitian lain oleh E. Dianawati dan A. Ardyanti dan R. N. Dawayanti dalam penelitiannya mengenai "Klasifikasi Dokumen Teks Menggunakan Svm-Dta (Support Vector Machines With Decission Tree Architecture)"[2], menjelaskan untuk hasil pengklasifikasian berita dinilai tepat, dengan melihat nilai error rate yang cukup rendah yakni 4% pada kernel polynomial dan kernel RBF. Dalam penelitian tersebut menggunakan klasifikasi 3 kelas dan pada tahap *preprocessing* tidak menggunakan *casefolding*. Dari hasil studi literature yang sudah diuraikan diatas, penggunaan SVM dalam klasifikasi dokumen teks memiliki keunggulan dengan tingkat keakurasian yang tinggi. Dalam penelitian tersebut masih diperlukan pengembangan dari batasan yang telah ada sebelumnya dalam mengklasifikasikan dokumen teks menggunakan SVM. Oleh karena itu, salah satu metode yang dapat diterapkan dalam kasus klasifikasi dokumen ini adalah dengan menggunakan *Support Vector Machine* (*SVM*).

3.2 Deskripsi Sistem

Klasifikasi dokumen menggunakan Support Vector Machine menjadi permasalahan yang akan dipecahkan dan dibahas pada penelitian ini, implementasi Support Vector Machine digunakan dalam pengklasifikasian laporan skripsi program studi teknik informatika, sistem pengklasifikasian laporan skripsi program studi teknik informatika berdasarkan kelompok keilmuan program studi teknik informatika. Sistem yang dirancang dapat dijalankan pada Perangkat Komputer (PC) dengan bahasa pemrograman *Java* berbasis *Dekstop*, yang dapat digunakan

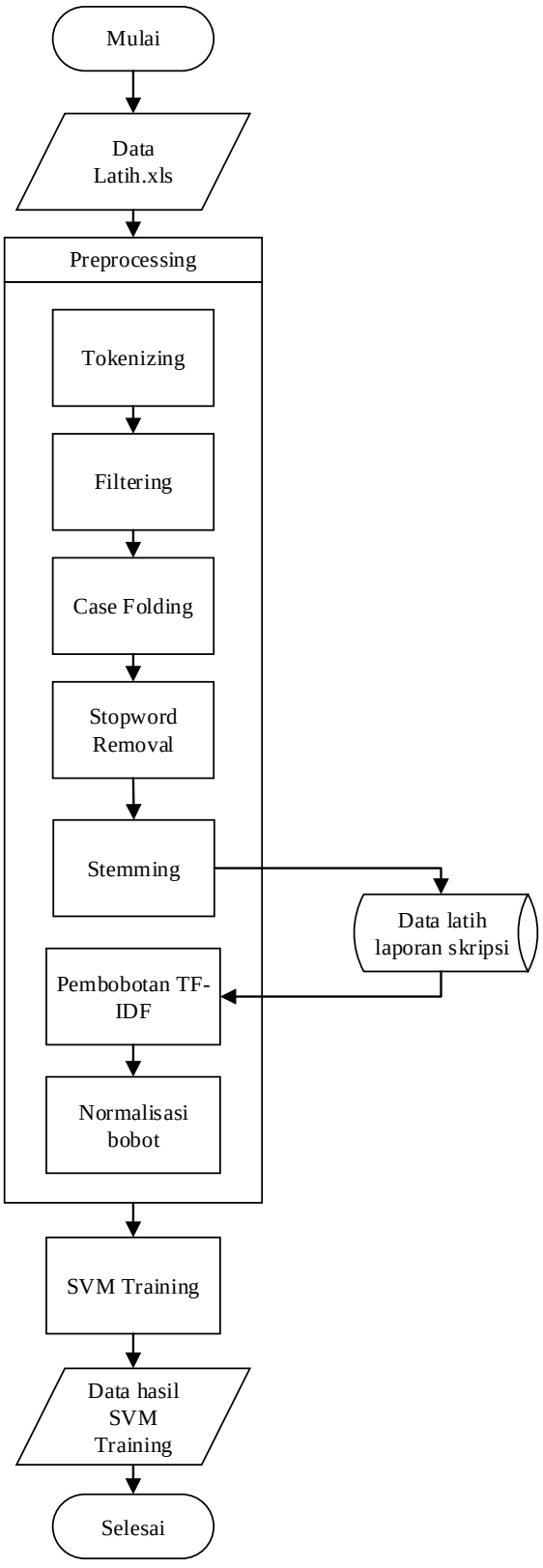
untuk melihat hasil klasifikasi dari laporan skripsi program studi teknik informatika.



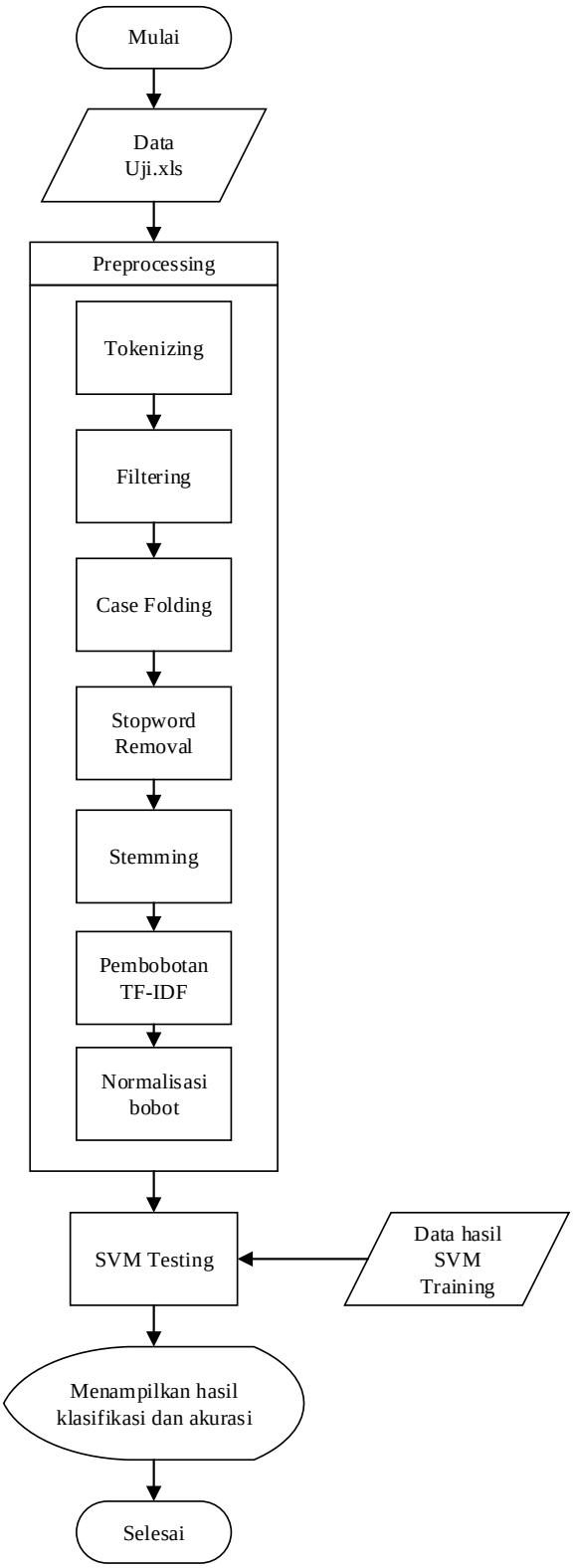
Gambar 3. 1 Deskripsi Sistem

3.3 Analisis Sistem

Dalam membangun sistem klasifikasi laporan skripsi, dilakukan beberapa tahapan analisis. Berikut tahapan yang dilakukan dapat dilihat pada gambar 3.2 dan 3.3.



Gambar 3. 2 *Flowchart* Sistem Latih



Gambar 3. 3 *Flowchart* Sistem Uji

3.4 Analisis Data Masukan

Data masukan didapatkan dari dokumen-dokumen laporan skripsi Prodi Teknik Informatika UNIKOM tahun 2015. Untuk mendapatkan dokumen tersebut dapat mengujungi perpustakaan digital UNIKOM dengan alamat situs elib.unikom.ac.id. Dokumen laporan skripsi yang diambil sebagai data masukan hanya pada bagian abstrak. Setelah mendapatkan dokumen yang di unduh dalam bentuk pdf maka dokumen tersebut dikonversi ke dalam bentuk .xls untuk data masukan ke dalam sistem dan database. Contoh data masukan pada .xls dalam tabel berikut.

Tabel 3. 1 Analisis Data Masukan

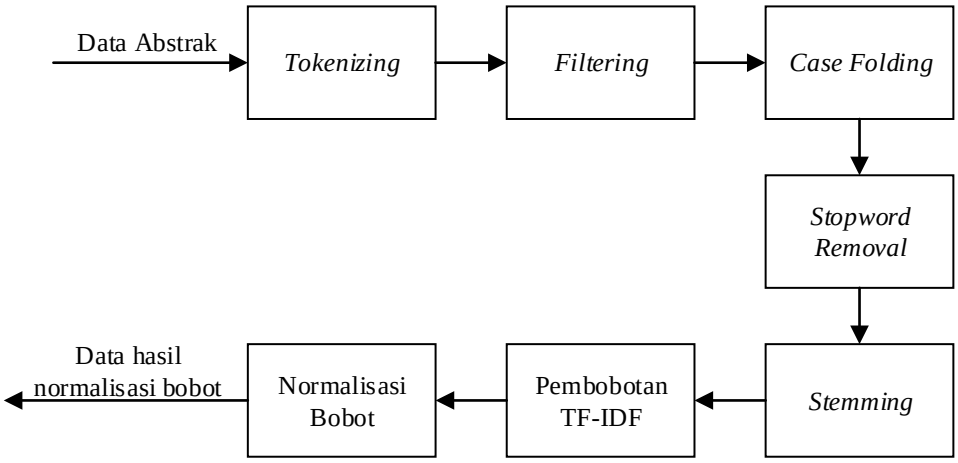
Abstrak	Kelas
ANALISIS PERFORMANSI METODE ADAPTIVE NEURO FUZZY INFERENCE SYSTEM (ANFIS) UNTUK MEMPREDIKSI TINGKAT POTENSI BANJIR Adaptive Neuro Fuzzy Inference System (ANFIS) merupakan suatu struktur yang menggabungkan system fuzzy dan system jaringan saraf tiruan. ANFIS yang menggunakan suatu prosedur learning dapat membangun suatu mapping input-output yang keduanya berdasarkan pada pengetahuan manusia (pada aturan fuzzy if-then). Kata Kunci : prediksi, potensi banjir, System Fuzzy, Jaringan Syaraf Tiruan, ANFIS	E
ANALISIS SENTIMEN PADA AKUN TWITTER PROVIDER TELEKOMUNIKASI. Pada penelitian ini metode klasifikasi yang digunakan untuk analisis sentimen adalah metode klasifikasi Support Vector Machine (SVM) dan metode klasifikasi Naive Bayes. Prinsip dasar metode SVM adalah pengklasifikasian linear (linear classifier). Kata kunci : Analisis Sentimen, Naive Bayes Classifier, Support Vector Machine, Preprocessing, Twitter	E
CUSTOMER RELATIONSHIP MANAGEMENT (CRM) DI PT. AZIZI AUDHINIA WISATA TOUR DAN TRAVEL. Dengan menerapkan konsep CRM (Customer Relationship Management). Tujuan dari proses identifikasi pelanggan adalah untuk mengetahui perilaku pelanggan dan menerapkan strategi pemasaran yang tepat, dengan media SMS Gateway untuk memberikan penawaran program-program perusahaan kepada pelanggan dengan tepat sasaran. Identifikasi pelanggan dapat dilakukan dengan cara mengelompokan pelanggan dari data transaksi pelanggan menggunakan metode RFM yaitu model berdasarkan atribut Recency, Frequency, dan Monetary. Kata kunci : Customer Relationship Management (CRM), Metode RFM	A

3.5 Analisis Proses

Proses pendekatan sistematis untuk mengidentifikasi permasalahan, peluang dan tujuan dalam merancang suatu sistem.

3.5.1 *Preprocessing*

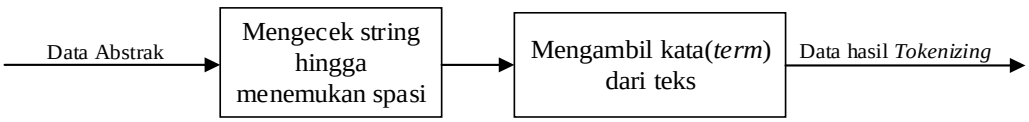
Tahapan dimana teks dipersiapkan menjadi data untuk diolah ke tahapan-tahapan selanjutnya. Input awal pada proses ini berupa dokumen dengan format .xls. Pada penelitian ini ada beberapa tahapan preprocessing yang dilakukan. Yaitu : proses *Tokenizing* , proses *Filtering*, proses *Case Folding*, proses *Stopword Removal*, proses *Stemming* dan Pembobotan TF-IDF serta Normalisasi Bobot. Berikut gambaran tahapan preprocessing dapat dilihat pada gambar 3.4.



Gambar 3. 4 *Preprocessing*

3.5.1.1 *Tokenizing*

Tahapan ini digunakan untuk memisahkan setiap kata yang terindentifikasi atau terpisahkan dengan kata yang lainnya oleh pemisah spasi yang akan dipecah dari abstrak menjadi kata-kata tunggal. Berikut tahapan yang dilakukan dapat dilihat pada gambar 3.5.



Gambar 3. 5 Proses *Tokenizing*

Contoh dokumen yang sudah melewati tahap *tokenizing* :

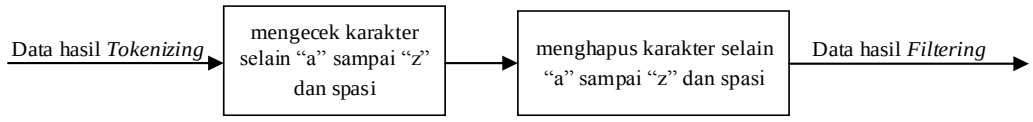
Tabel 3. 2 Contoh *Tokenizing*

Sebelum *tokenizing*	Sesudah *tokenizing*		
ANALISIS PERFORMANSI METODE ADAPTIVE NEURO FUZZY INFERENCE SYSTEM (ANFIS) UNTUK MEMPREDIKSI TINGKAT POTENSI BANJIR Adaptive Neuro Fuzzy Inference System (ANFIS) merupakan suatu struktur yang menggabungkan system fuzzy dan system jaringan saraf tiruan. ANFIS yang menggunakan suatu prosedur learning dapat membangun suatu mapping input-output yang keduanya berdasarkan pada pengetahuan manusia (pada aturan fuzzy if-then). Kata Kunci : prediksi, potensi banjir, System Fuzzy, Jaringan Syaraf Tiruan, ANFIS	ANALISIS	PERFORMANSI	METODE
	ADAPTIVE	NEURO	FUZZY
	INFERENCE	SYSTEM	(ANFIS)
	UNTUK	MEMPREDIKSI	TINGKAT
	POTENSI	BANJIR	Adaptive
	Neuro	Fuzzy	Inference
	System	(ANFIS)	merupakan
	suatu	struktur	yang
	menggabungkan	System	fuzzy
	dan	system	jaringan
	saraf	tiruan.	ANFIS
	yang	menggunakan	suatu
	prosedur	learning	dapat
	membangun	suatu	mapping
	input-output	yang	keduanya
	berdasarkan	pada	pengetahuan
	manusia	(pada	aturan
	fuzzy	if-then).	Kata
	Kunci	:	prediksi,
	potensi	banjir,	System
	Fuzzy,	Jaringan	Syaraf
	Tiruan,	ANFIS	
ANALISIS SENTIMEN PADA AKUN TWITTER PROVIDER TELEKOMUNIKASI. Pada penelitian ini metode klasifikasi yang digunakan untuk analisis sentimen adalah metode klasifikasi Support Vector Machine (SVM) dan metode klasifikasi Naive Bayes. Prinsip	ANALISIS	SENTIMEN	PADA
	AKUN	TWITTER	PROVIDER
	TELEKOMUNIKASI.	Pada	penelitian
	ini	metode	klasifikasi
	yang	digunakan	untuk
	analisis	sentimen	adalah
	metode	klasifikasi	Support
	Vector	Machine	(SVM)
	dan	metode	klasifikasi
	Naive	Bayes.	Prinsip
	dasar	metode	SVM

dasar metode SVM adalah pengklasifikasian linear (linear classifier). Kata kunci : Analisis Sentimen, Naive Bayes Classifier, Support Vector Machine, Preprocessing, Twitter	adalah	pengklasifikasian	linear
	(linear	classifier).	Kata
	kunci	:	Analisis
	Sentimen,	Naive	Bayes
	Classifier,	Support	Vector
	Machine,	Preprocessing,	Twitter
CUSTOMER RELATIONSHIP MANAGEMENT (CRM) DI PT. AZIZI AUDHINIA WISATA TOUR DAN TRAVEL. Dengan menerapkan konsep CRM (Customer Relationship Management). Tujuan dari proses identifikasi pelanggan adalah untuk mengetahui perilaku pelanggan dan menerapkan strategi pemasaran yang tepat, dengan media SMS Gateway untuk memberikan penawaran program-program perusahaan kepada pelanggan dengan tepat sasaran. Identifikasi pelanggan dapat dilakukan dengan cara mengelompokan pelanggan dari data transaksi pelanggan menggunakan metode RFM yaitu model berdasarkan atribut Recency, Frequency, dan Monetary. Kata kunci : Customer Relationship Management (CRM), Metode RFM	CUSTOMER	RELATIONSHIP	MANAGEMENT
	(CRM)	DI	PT.
	AZIZI	AUDHINIA	WISATA
	TOUR	DAN	TRAVEL.
	Dengan	menerapkan	konsep
	CRM	(Customer	Relationship
	Management).	Tujuan	dari
	proses	identifikasi	pelanggan
	adalah	untuk	mengetahui
	perilaku	pelanggan	dan
	menerapkan	strategi	pemasaran
	yang	tepat,	dengan
	media	SMS	Gateway
	untuk	memberikan	penawaran
	program-program	perusahaan	kepada
	pelanggan	dengan	tepat
	sasaran.	Identifikasi	pelanggan
	dapat	dilakukan	dengan
	cara	Mengelompokan	pelanggan
	dari	data	transaksi
	pelanggan	menggunakan	metode
	RFM	yaitu	model
	berdasarkan	atribut	Recency,
	Frequency,	dan	Monetary.
	Kata	kunci	:
	Customer	Relationship	Management
	(CRM),	Metode	RFM

3.5.1.2 *Filtering*

Tahap ini diperlukan untuk mendapatkan hasil *term* dengan tidak adanya karakter lain selain karakter “a” sampai “z” dan spasi. Contohnya karakter “(ANFIS)” akan menjadi “ANFIS” karena simbol “(“ dan “)” bukan termasuk karakter “a” sampai “z” dan spasi maka simbol tersebut akan dihapus. Berikut tahapan yang dilakukan dapat dilihat pada gambar 3.6.



Gambar 3. 6 Proses *Filtering*

Contoh dokumen yang sudah melewati tahap *filtering* :

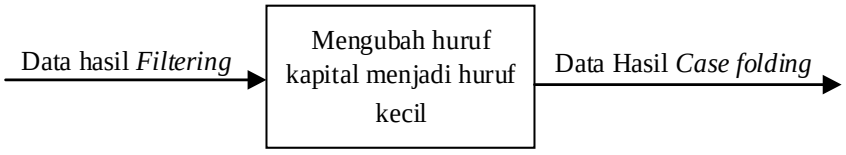
Tabel 3. 3 Contoh *Filtering*

Sebelum *Filtering*	Sesudah *Filtering*
ANALISIS PERFORMANSI METODE ADAPTIVE NEURO FUZZY INFERENCE SYSTEM (ANFIS) UNTUK MEMPREDIKSI TINGKAT POTENSI BANJIR Adaptive Neuro Fuzzy Inference System (ANFIS) merupakan suatu struktur yang menggabungkan system fuzzy dan system jaringan saraf tiruan. ANFIS yang menggunakan suatu prosedur learning dapat membangun suatu mapping input-output yang keduanya berdasarkan pada pengetahuan manusia (pada aturan fuzzy if-then). Kata Kunci : prediksi, potensi banjir, System Fuzzy, Jaringan Syaraf Tiruan, ANFIS	ANALISIS PERFORMANSI METODE ADAPTIVE NEURO FUZZY INFERENCE SYSTEM ANFIS UNTUK MEMPREDIKSI TINGKAT POTENSI BANJIR Adaptive Neuro Fuzzy Inference System ANFIS merupakan suatu struktur yang menggabungkan system fuzzy dan system jaringan saraf tiruan ANFIS yang menggunakan suatu prosedur learning dapat membangun suatu mapping input output yang keduanya berdasarkan pada pengetahuan manusia pada aturan fuzzy if then Kata Kunci prediksi potensi banjir System Fuzzy Jaringan Syaraf Tiruan ANFIS
ANALISIS SENTIMEN PADA AKUN TWITTER PROVIDER TELEKOMUNIKASI. Pada penelitian ini metode klasifikasi yang digunakan untuk analisis sentimen adalah metode klasifikasi Support Vector Machine (SVM) dan metode klasifikasi Naive Bayes. Prinsip dasar metode SVM	ANALISIS SENTIMEN PADA AKUN TWITTER PROVIDER TELEKOMUNIKASI Pada penelitian ini metode klasifikasi yang digunakan untuk analisis sentimen adalah metode klasifikasi Support Vector Machine SVM dan metode klasifikasi Naive Bayes Prinsip dasar metode SVM

adalah pengklasifikasian linear (linear classifier). Kata kunci : Analisis Sentimen, Naive Bayes Classifier, Support Vector Machine, Preprocessing, Twitter	adalah pengklasifikasian linear linear classifier Kata kunci Analisis Sentimen Naive Bayes Classifier Support Vector Machine Preprocessing Twitter
CUSTOMER RELATIONSHIP MANAGEMENT (CRM) DI PT. AZIZI AUDHINIA WISATA TOUR DAN TRAVEL. Dengan menerapkan konsep CRM (Customer Relationship Management). Tujuan dari proses identifikasi pelanggan adalah untuk mengetahui perilaku pelanggan dan menerapkan strategi pemasaran yang tepat, dengan media SMS Gateway untuk memberikan penawaran program-program perusahaan kepada pelanggan dengan tepat sasaran. Identifikasi pelanggan dapat dilakukan dengan cara mengelompokan pelanggan dari data transaksi pelanggan menggunakan metode RFM yaitu model berdasarkan atribut Recency, Frequency, dan Monetary. Kata kunci : Customer Relationship Management (CRM), Metode RFM	CUSTOMER RELATIONSHIP MANAGEMENT CRM DI PT AZIZI AUDHINIA WISATA TOUR DAN TRAVEL Dengan menerapkan konsep CRM Customer Relationship Management Tujuan dari proses identifikasi pelanggan adalah untuk mengetahui perilaku pelanggan dan menerapkan strategi pemasaran yang tepat dengan media SMS Gateway untuk memberikan penawaran program program perusahaan kepada pelanggan dengan tepat sasaran Identifikasi pelanggan dapat dilakukan dengan cara mengelompokan pelanggan dari data transaksi pelanggan menggunakan metode RFM yaitu model berdasarkan atribut Recency Frequency dan Monetary Kata kunci Customer Relationship Management CRM Metode RFM

3.5.1.3 *Case Folding*

Tahap ini diperlukan untuk menghasilkan *term* yang diseragamkan ke dalam huruf kecil. Contohnya pada kata "ANALISIS" akan berubah menjadi "analisis" karena kata tersebut mengandung huruf kapital. Berikut tahapan yang dilakukan dapat dilihat pada gambar 3.7.



Gambar 3. 7 Proses *Case Folding*

Contoh dokumen yang sudah melewati tahap *case folding* :

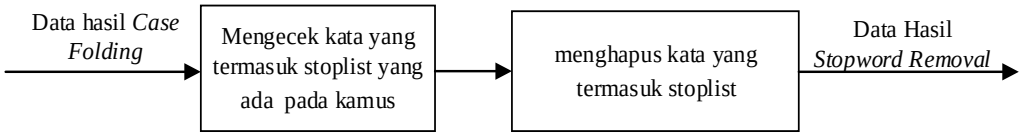
Tabel 3. 4 Contoh *Case Folding*

Sebelum *Case Folding*	Sesudah *Case Folding*
ANALISIS PERFORMANSI METODE ADAPTIVE NEURO FUZZY INFERENCE SYSTEM ANFIS UNTUK MEMPREDIKSI TINGKAT POTENSI BANJIR Adaptive Neuro Fuzzy Inference System ANFIS merupakan suatu struktur yang menggabungkan system fuzzy dan system jaringan saraf tiruan ANFIS yang menggunakan suatu prosedur learning dapat membangun suatu mapping input output yang keduanya berdasarkan pada pengetahuan manusia pada aturan fuzzy if then Kata Kunci prediksi potensi banjir System Fuzzy Jaringan Syaraf Tiruan ANFIS	analisis performansi metode adaptive neuro fuzzy inference system anfis untuk memprediksi tingkat potensi banjir adaptive neuro fuzzy inference system anfis merupakan suatu struktur yang menggabungkan system fuzzy dan system jaringan saraf tiruan anfis yang menggunakan suatu prosedur learning dapat membangun suatu mapping input output yang keduanya berdasarkan pada pengetahuan manusia pada aturan fuzzy if then kata kunci prediksi potensi banjir system fuzzy jaringan syaraf tiruan anfis
ANALISIS SENTIMEN PADA AKUN TWITTER PROVIDER TELEKOMUNIKASI Pada penelitian ini metode klasifikasi yang digunakan untuk analisis sentimen adalah metode klasifikasi Support Vector Machine SVM dan metode klasifikasi Naive Bayes Prinsip dasar metode SVM adalah pengklasifikasian linear linear classifier Kata kunci Analisis Sentimen Naive Bayes Classifier Support Vector Machine Preprocessing Twitter	analisis sentimen pada akun twitter provider telekomunikasi pada penelitian ini metode klasifikasi yang digunakan untuk analisis sentimen adalah metode klasifikasi support vector machine svm dan metode klasifikasi naive bayes prinsip dasar metode svm adalah pengklasifikasian linear linear classifier kata kunci analisis sentimen naive bayes classifier support vector machine preprocessing twitter
CUSTOMER RELATIONSHIP MANAGEMENT CRM DI PT AZIZI AUDHINIA WISATA TOUR DAN TRAVEL Dengan menerapkan konsep CRM Customer Relationship Management Tujuan dari proses identifikasi pelanggan adalah untuk mengetahui perilaku pelanggan dan menerapkan strategi pemasaran yang tepat dengan media SMS Gateway untuk memberikan penawaran program program perusahaan kepada pelanggan dengan tepat sasaran Identifikasi	customer relationship management crm di pt azizi audhinia wisata tour dan travel dengan menerapkan konsep crm customer relationship management tujuan dari proses identifikasi pelanggan adalah untuk mengetahui perilaku pelanggan dan menerapkan strategi pemasaran yang tepat dengan media sms gateway untuk memberikan penawaran program program perusahaan kepada pelanggan dengan tepat sasaran identifikasi pelanggan dapat dilakukan dengan cara

pelanggan dapat dilakukan dengan cara mengelompokan pelanggan dari data transaksi pelanggan menggunakan metode RFM yaitu model berdasarkan atribut Recency Frequency dan Monetary Kata kunci Customer Relationship Management CRM Metode RFM	mengelompokan pelanggan dari data transaksi pelanggan menggunakan metode rfm yaitu model berdasarkan atribut recency frequency dan monetary kata kunci customer relationship management crm metode rfm

3.5.1.4 *Stopword Removal*

Tahap ini diperlukan untuk menghasilkan *term* yang hanya memiliki keterkaitan dengan topik tertentu. Untuk mendapatkan *term* tersebut maka dilakukan pembuangan pada kata-kata yang tidak terkait. kata-kata yang tidak terkait tersebut sudah tersedia didalam kamus *stopword* yang telah dibuat. Biasanya kata-kata yang sering muncul atau kata-kata tersebut tidak memiliki arti atau tidak relevan, Seperti kata sambung, kata depan, kata ganti, kata penghubung, dll. Contohnya pada kata "untuk" akan dibandingkan dengan kata yang sudah ada pada kamus *stopword* yang telah dibuat. Karena kata tersebut ada dalam kamus *stopword* maka kata tersebut akan dihapus. Berikut tahapan yang dilakukan dapat dilihat pada gambar 3.8.



Gambar 3. 8 Proses *Stopword Removal*

Contoh dokumen yang sudah melewati tahap *stopword removal* :

Tabel 3. 5 Contoh *Stopword Removal*

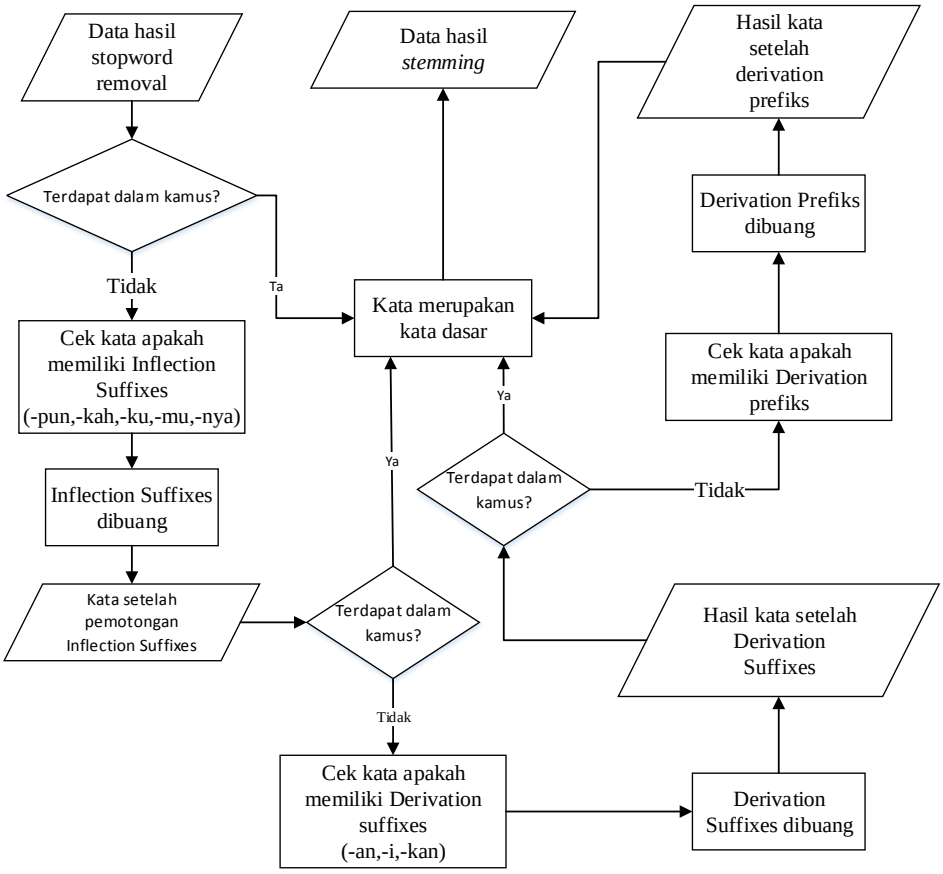
Sebelum *Stopword Removal*	Sesudah *Stopword Removal*
analisis performansi metode adaptive neuro fuzzy inference system anfis untuk memprediksi tingkat potensi banjir adaptive neuro fuzzy inference system anfis merupakan suatu struktur yang menggabungkan system fuzzy dan system jaringan saraf tiruan anfis yang	analisis performansi metode adaptive neuro fuzzy inference system anfis memprediksi potensi banjir adaptive neuro fuzzy inference system anfis struktur menggabungkan system fuzzy system jaringan saraf tiruan anfis

menggunakan suatu prosedur learning dapat membangun suatu mapping input output yang keduanya berdasarkan pada pengetahuan manusia pada aturan fuzzy if then kata kunci prediksi potensi banjir system fuzzy jaringan syaraf tiruan anfis	prosedur learning membangun mapping input output keduanya pengetahuan manusia pada aturan fuzzy if then kunci prediksi potensi banjir system fuzzy jaringan syaraf tiruan anfis
analisis sentimen pada akun twitter provider telekomunikasi pada penelitian ini metode klasifikasi yang digunakan untuk analisis sentimen adalah metode klasifikasi support vector machine svm dan metode klasifikasi naive bayes prinsip dasar metode svm adalah pengklasifikasian linear linear classifier kata kunci analisis sentimen naive bayes classifier support vector machine preprocessing twitter	analisis sentimen akun twitter provider telekomunikasi pada penelitian metode klasifikasi analisis sentimen metode klasifikasi support vector machine svm metode klasifikasi naive bayes prinsip dasar metode svm pengklasifikasian linear linear classifier kunci analisis sentimen naive bayes classifier support vector machine preprocessing twitter
customer relationship management crm di pt azizi audhinia wisata tour dan travel dengan menerapkan konsep crm customer relationship management tujuan dari proses identifikasi pelanggan adalah untuk mengetahui perilaku pelanggan dan menerapkan strategi pemasaran yang tepat dengan media sms gateway untuk memberikan penawaran program program perusahaan kepada pelanggan dengan tepat sasaran identifikasi pelanggan dapat dilakukan dengan cara mengelompokan pelanggan dari data transaksi pelanggan menggunakan metode rfm yaitu model berdasarkan atribut recency frequency dan monetary kata kunci customer relationship management crm metode rfm	customer relationship management crm pt azizi audhinia wisata tour travel menerapkan konsep crm customer relationship management tujuan identifikasi pelanggan perilaku pelanggan menerapkan strategi pemasaran tepat media sms gateway memberikan penawaran program-program perusahaan pelanggan tepat sasaran identifikasi pelanggan mengelompokan pelanggan data transaksi pelanggan metode rfm model atribut recency frequency monetary kunci customer relationship management crm metode rfm

3.5.1.5 *Stemming*

Tahap ini diperlukan untuk menghasilkan *term* yang dengan kata dasar tanpa imbuhan Kata-kata yang mempunyai imbuhan akan ditransformasikakn menjadi kata-kata dasar (*root word*) dengan menggunakan aturan-aturan tertentu. Contohnya pada kata “memprediksi”, kata tersebut akan melewati tahap pengecekan *inflection suffixes, derivation suffixes* dan *derivation prefiks* kemudian

menghasilkan kata “perdiksi” seperti dalam kamus dan akan disimpan di dalam database. Berikut tahapan yang dilakukan dapat dilihat pada gambar 3.9.



Gambar 3. 9 Proses *Stemming*

Contoh dokumen yang sudah melewati tahap *stemming* :

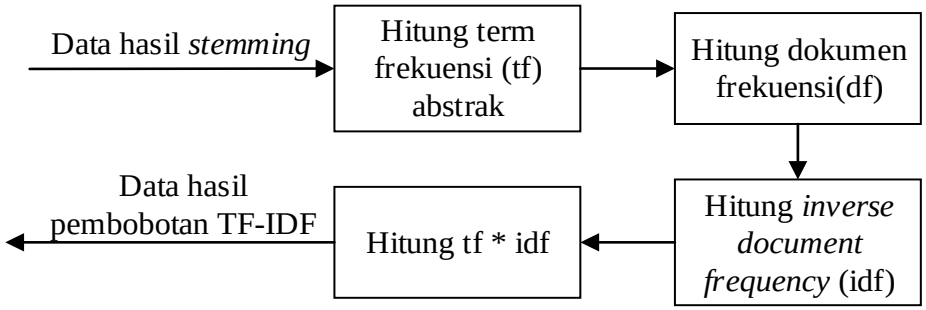
Tabel 3. 6 Contoh *Stemming*

Sebelum *Stemming*	Sesudah *Stemming*
analisis performansi metode adaptive neuro fuzzy inference system anfis memprediksi potensi banjir adaptive neuro fuzzy inference system anfis struktur menggabungkan system fuzzy system jaringan saraf tiruan anfis prosedur learning membangun mapping input output keduanya pengetahuan manusia pada aturan fuzzy if then kunci prediksi potensi banjir system fuzzy jaringan syaraf tiruan anfis	analisis performansi metode adaptive neuro fuzzy inference system anfis prediksi potensi banjir adaptive neuro fuzzy inference system anfis struktur gabung system fuzzy system jaring saraf tiru anfis prosedur learning bangun mapping input output dua pengetahuan manusia pada atur fuzzy if then kunci prediksi potensi banjir system fuzzy jaring syaraf tiru anfis

analisis sentimen akun twitter provider telekomunikasi pada penelitian metode klasifikasi analisis sentimen metode klasifikasi support vector machine svm metode klasifikasi naive bayes prinsip dasar metode svm pengklasifikasian linear linear classifier kunci analisis sentimen naive bayes classifier support vector machine preprocessing twitter	analisis sentimen akun twitter provider telekomunikasi pada teliti metode klasifikasi analisis sentimen metode klasifikasi support vector machine svm metode klasifikasi naive bayes prinsip dasar metode svm klasifikasi linear linear classifier kunci analisis sentimen naive bayes classifier support vector machine preprocessing twitter
customer relationship management crm pt azizi audhinia wisata tour travel menerapkan konsep crm customer relationship management tujuan identifikasi pelanggan perilaku pelanggan menerapkan strategi pemasaran tepat media sms gateway memberikan penawaran program program perusahaan pelanggan tepat sasaran identifikasi pelanggan mengelompokan pelanggan data transaksi pelanggan metode rfm model atribut recency frequency monetary kunci customer relationship management crm metode rfm	customer relationship management crm pt aziz audhinia wisata tour travel terap konsep crm customer relationship management tuju identifikasi langgan perilaku langgan terap strategi pasar tepat media sms gateway beri tawaran program usaha langgan tepat sasar identifikasi langgan kelompok langgan data transaksi langgan metode rfm model atribut recency frequency monetary kunci customer relationship management crm metode rfm

3.5.1.6 Pembobotan TF-IDF

Pada penelitian ini dilakukan perhitungan bobot diperoleh dari jumlah kemunculan *term* dalam satu dokumen (*tf*) dan jumlah kemunculan *term* dalam kumpulan dokumen (*idf*). Untuk mendapatkan nilai IDF dilakukan perhitungan dengan persamaan (2.1). Setelah mendapatkan nilai *tf* dan *idf* maka selanjutnya menghitung pembobotan pada *term*, untuk mendapatkan bobot (W) masing-masing dokumen pada setiap *term* dilakukan persamaan (2.2). Berikut alur proses dapat dilihat pada gambar 3.10.



Gambar 3. 10 Proses Pembobotan Tf-Idf

Berikut adalah contoh pembobotan TF-IDF dengan dokumen abstrak yang sudah melalui tahap *stemming* dapat dilihat pada tabel 3.5.

Tabel 3. 5 *Term* dari Abstrak Laporan

Dokumen Abstrak	*Term*
S1	analisis performansi metode adaptive neuro fuzzy inference system anfis prediksi potensi banjir adaptive neuro fuzzy inference system anfis struktur gabung system fuzzy system jaring saraf tiru anfis prosedur learning bangun mapping input output dua pengetahuan manusia pada atur fuzzy if then kunci prediksi potensi banjir system fuzzy jaring syaraf tiru anfis
S2	analisis sentimen akun twitter provider telekomunikasi pada teliti metode klasifikasi analisis sentimen metode klasifikasi support vector machine svm metode klasifikasi naive bayes prinsip dasar metode svm klasifikasi linear linear classifier kunci analisis sentimen naive bayes classifier support vector machine preprocessing twitter
S3	customer relationship management crm pt aziz audhinia wisata tour travel terap konsep crm customer relationship management tuju identifikasi langgan perilaku langgan terap strategi pasar tepat media sms gateway beri tawaran program usaha langgan tepat sasar identifikasi langgan kelompok langgan data transaksi langgan metode rfm model atribut recency frequency monetary kunci customer relationship management crm metode rfm

Proses awal dilakukan perhitungan kata (*term*) pada tiap dokumen, sehingga mendapatkan frekuensi *term.* Selanjutnya adalah menghitung *df,* karena *df* merupakan banyaknya dokumen dimana suatu *term* muncul. Hasil frekuensi *term* dan *df* tersebut dapat dilihat pada tabel 3.6.

Setelah mendapatkan nilai *df,* maka dilakukan perhitungan *idf* dengan persamaan (2.1).

$$IDF_t = \log(N/df) \qquad (2.1)$$

Diambil contoh pada kata “tepat”. Didapatkan banyak dokumen (N) = 3, dan df = 1. Maka, dihitung seperti berikut.

$IDF_t = \log(3/1) = 0{,}4771$

Selanjutnya untuk mendapatkan bobot term maka dilakukan pehitungan *tf* dan *idf* dengan persamaan (2.2).

$$W_{dt} = tf_{dt} * IDF_t \qquad (2.2)$$

Diperoleh *tf* =2 , dan *IDF* = 0,4771. Maka, perhitungan yang diperoleh sebagai berikut.

$W_{dt} = 2 * 0{,}4771 = 0{,}9542$

Sehingga, kata “tepat” memiliki bobot 0,9542

Hasil dari perhitungan pembobotan TF-IDF dapat dilihat pada tabel 3.6.

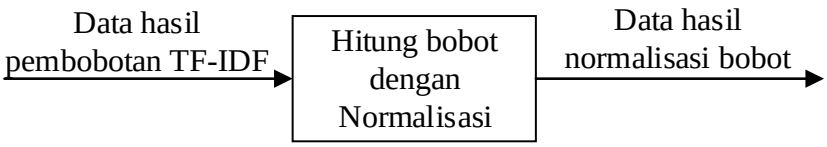
Tabel 3. 6 Frekuensi *term* dan df

Kata	Tf			Df	N/df	Idf	W		
	S1	S2	S3				S1	S2	S3
tepat	0	0	2	1	3/1	0,4771	0	0	0,9542
terap	0	0	2	1	3/1	0,4771	0	0	0,9542
wisata	0	0	1	1	3/1	0,4771	0	0	0,4771
analisis	1	3	0	2	3/2	0,1761	0,1761	0,5283	0
pt	0	0	1	1	3/1	0,4771	0	0	0,4771
data	0	0	1	1	3/1	0,4771	0	0	0,4771
pengetahuan	1	0	0	1	3/1	0,4771	0,4771	0	0
aziz	0	0	1	1	3/1	0,4771	0	0	0,4771
jaring	2	0	0	1	3/1	0,4771	0,9542	0	0
saraf	1	0	0	1	3/1	0,4771	0,4771	0	0
program	0	0	1	1	3/1	0,4771	0	0	0,4771
struktur	1	0	0	1	3/1	0,4771	0,4771	0	0
prediksi	2	0	0	1	3/1	0,4771	0,9542	0	0
frequency	0	0	1	1	3/1	0,4771	0	0	0,4771
metode	1	4	2	3	3/3	0	0	0	0
kelompok	0	0	1	1	3/1	0,4771	0	0	0,4771

twitter	0	2	0	1	3/1	0,4771	0	0,9542	0
bangun	1	0	0	1	3/1	0,4771	0,4771	0	0
transaksi	0	0	1	1	3/1	0,4771	0	0	0,4771
prosedur	1	0	0	1	3/1	0,4771	0,4771	0	0
classifier	0	2	0	1	3/1	0,4771	0	0,9542	0
pada	1	1	0	2	3/2	0,1761	0,1761	0,1761	0
usaha	0	0	1	1	3/1	0,4771	0	0	0,4771
model	0	0	1	1	3/1	0,4771	0	0	0,4771
if	1	0	0	1	3/1	0,4771	0,4771	0	0
telekomunikasi	0	1	0	1	3/1	0,4771	0	0,4771	0
sentimen	0	3	0	1	3/1	0,4771	0	1,4314	0
mapping	1	0	0	1	3/1	0,4771	0,4771	0	0
potensi	2	0	0	1	3/1	0,4771	0,9542	0	0
preprocessing	0	1	0	1	3/1	0,4771	0	0,4771	0
audhinia	0	0	1	1	3/1	0,4771	0	0	0,4771
kunci	1	1	1	3	3/3	0	0	0	0
input	1	0	0	1	3/1	0,4771	0,4771	0	0
atribut	0	0	1	1	3/1	0,4771	0	0	0,4771
system	5	0	0	1	3/1	0,4771	2,3856	0	0
rfm	0	0	2	1	3/1	0,4771	0	0	0,9542
tuju	0	0	1	1	3/1	0,4771	0	0	0,4771
identifikasi	0	0	2	1	3/1	0,4771	0	0	0,9542
atur	1	0	0	1	3/1	0,4771	0,4771	0	0
support	0	2	0	1	3/1	0,4771	0	0,9542	0
pemasaran	0	0	1	1	3/1	0,4771	0	0	0,4771
manusia	1	0	0	1	3/1	0,4771	0,4771	0	0
langgan	0	0	6	1	3/1	0,4771	0	0	2,8627
inference	2	0	0	1	3/1	0,4771	0,9542	0	0
anfis	4	0	0	1	3/1	0,4771	1,9085	0	0
learning	1	0	0	1	3/1	0,4771	0,4771	0	0
penelitian	0	1	0	1	3/1	0,4771	0	0,4771	0
media	0	0	1	1	3/1	0,4771	0	0	0,4771
tiru	2	0	0	1	3/1	0,4771	0,9542	0	0
perilaku	0	0	1	1	3/1	0,4771	0	0	0,4771
adaptive	2	0	0	1	3/1	0,4771	0,9542	0	0
sasar	0	0	1	1	3/1	0,4771	0	0	0,4771
provider	0	1	0	1	3/1	0,4771	0	0,4771	0
sms	0	0	1	1	3/1	0,4771	0	0	0,4771
penawaran	0	0	1	1	3/1	0,4771	0	0	0,4771
bayes	0	2	0	1	3/1	0,4771	0	0,9542	0
vector	0	2	0	1	3/1	0,4771	0	0,9542	0
relationship	0	0	3	1	3/1	0,4771	0	0	1,4314
fuzzy	5	0	0	1	3/1	0,4771	2,3856	0	0
crm	0	0	3	1	3/1	0,4771	0	0	1,4314
banjir	2	0	0	1	3/1	0,4771	0,9542	0	0
linear	0	2	0	1	3/1	0,4771	0	0,9542	0
monetary	0	0	1	1	3/1	0,4771	0	0	0,4771
neuro	2	0	0	1	3/1	0,4771	0,9542	0	0
performansi	1	0	0	1	3/1	0,4771	0,4771	0	0
syaraf	1	0	0	1	3/1	0,4771	0,4771	0	0
dasar	0	1	0	1	3/1	0,4771	0	0,4771	0
svm	0	2	0	1	3/1	0,4771	0	0,9542	0
strategi	0	0	1	1	3/1	0,4771	0	0	0,4771

recency	0	0	1	1	3/1	0,4771	0	0	0,4771
gabung	1	0	0	1	3/1	0,4771	0,4771	0	0
akun	0	1	0	1	3/1	0,4771	0	0,4771	0
prinsip	0	1	0	1	3/1	0,4771	0	0,4771	0
tour	0	0	1	1	3/1	0,4771	0	0	0,4771
dua	1	0	0	1	3/1	0,4771	0,4771	0	0
konsep	0	0	1	1	3/1	0,4771	0	0	0,4771
management	0	0	3	1	3/1	0,4771	0	0	1,4314
machine	0	2	0	1	3/1	0,4771	0	0,9542	0
klasifikasi	0	4	0	1	3/1	0,4771	0	1,9085	0
travel	0	0	1	1	3/1	0,4771	0	0	0,4771
Naive	0	2	0	1	3/1	0,4771	0	0,9542	0
Gateway	0	0	1	1	3/1	0,4771	0	0	0,4771
Customer	0	0	3	1	3/1	0,4771	0	0	1,4314

3.5.1.7 Normalisasi Bobot



Gambar 3. 11 Proses Normalisasi Bobot

Tahapan ini dilakukan agar menjaga nilai tetap pada rentang 0 – 1. Contoh bobot normalisasi dari S3 dengan kata "tepat" dengan hasil bobot dari TF-IDF adalah 0,9542. Normalisasi yang dilakukan menggunakan rumus :

$$\boldsymbol{w(word_i)} = \frac{\boldsymbol{w(word_i)}}{\sqrt{\boldsymbol{w^2(word_1)} + \boldsymbol{w^2(word_2)} + \cdots + \boldsymbol{w^2(word_n)}}} \qquad (2.3)$$

$$w(tepat) = \frac{0{,}9542}{\sqrt{\begin{array}{c} 0{,}9542^2 + 0{,}9542^2 + 0{,}4771^2 + 0{,}4771^2 + 0{,}4771^2 + 0{,}4771^2 + 0{,}4771^2 + 0{,}4771^2 + \\ 0{,}4771^2 + 0{,}4771^2 + 0{,}4771^2 + 0{,}4771^2 + 0{,}4771^2 + 0{,}4771^2 + 0{,}9542^2 + 0{,}4771^2 + \\ 0{,}9542^2 + 0{,}4771^2 + 2{,}8627^2 + 0{,}4771^2 + 0{,}4771^2 + 0{,}4771^2 + 0{,}4771^2 + 0{,}4771^2 + \\ 1{,}4314^2 + 1{,}4314^2 + 0{,}4771^2 + 0{,}4771^2 + 0{,}4771^2 + 0{,}4771^2 + 0{,}4771^2 + 1{,}4314^2 + \\ 0{,}4771^2 + 0{,}4771^2 + 1{,}4314^2 \end{array}}} = 0.1890$$

Maka didapatlah bobot yang baru tiap dokumen menjadi :

Tabel 3. 7 Bobot Normalisasi

Kata	Tf			Df	N/df	Idf	W normalisasi		
	S1	S2	S3				S1	S2	S3
tepat	0	0	2	1	3/1	0,4771	0	0	0.1890
terap	0	0	2	1	3/1	0,4771	0	0	0.1890
wisata	0	0	1	1	3/1	0,4771	0	0	0.1453
analisis	1	3	0	2	3/2	0,1761	0	0	0
pt	0	0	1	1	3/1	0,4771	0	0	0.1453
data	0	0	1	1	3/1	0,4771	0	0	0.1453
pengetahuan	1	0	0	1	3/1	0,4771	0.1646	0	0
aziz	0	0	1	1	3/1	0,4771	0	0	0.1453
jaring	2	0	0	1	3/1	0,4771	0.2142	0	0
saraf	1	0	0	1	3/1	0,4771	0.1646	0	0
program	0	0	1	1	3/1	0,4771	0	0	0.1453
struktur	1	0	0	1	3/1	0,4771	0.1646	0	0
prediksi	2	0	0	1	3/1	0,4771	0.2142	0	0
frequency	0	0	1	1	3/1	0,4771	0	0	0.1453
metode	1	4	2	3	3/3	0	0	0	0.1890
kelompok	0	0	1	1	3/1	0,4771	0	0	0.1453
twitter	0	2	0	1	3/1	0,4771	0	0.2504	0
bangun	1	0	0	1	3/1	0,4771	0.1646	0	0
transaksi	0	0	1	1	3/1	0,4771	0	0	0.1453
prosedur	1	0	0	1	3/1	0,4771	0.1646	0	0
classifier	0	2	0	1	3/1	0,4771	0	0.2504	0
pada	1	1	0	2	3/2	0,1761	0	0	0
usaha	0	0	1	1	3/1	0,4771	0	0	0.1453
model	0	0	1	1	3/1	0,4771	0	0	0.1453
if	1	0	0	1	3/1	0,4771	0.1646	0	0
telekomunikasi	0	1	0	1	3/1	0,4771	0	0.1925	0
sentimen	0	3	0	1	3/1	0,4771	0	0.2843	0
mapping	1	0	0	1	3/1	0,4771	0.1646	0	0
potensi	2	0	0	1	3/1	0,4771	0.2142	0	0
preprocessing	0	1	0	1	3/1	0,4771	0	0.1925	0
audhinia	0	0	1	1	3/1	0,4771	0	0	0.1453
kunci	1	1	1	3	3/3	0	0	0	0.1453
input	1	0	0	1	3/1	0,4771	0.1646	0	0
atribut	0	0	1	1	3/1	0,4771	0	0	0.1453
system	5	0	0	1	3/1	0,4771	0.2797	0	0
rfm	0	0	2	1	3/1	0,4771	0	0	0.1890
tuju	0	0	1	1	3/1	0,4771	0	0	0.1453
identifikasi	0	0	2	1	3/1	0,4771	0	0	0.1890
atur	1	0	0	1	3/1	0,4771	0.1646	0	0
support	0	2	0	1	3/1	0,4771	0	0.2504	0
pemasaran	0	0	1	1	3/1	0,4771	0	0	0.1453
manusia	1	0	0	1	3/1	0,4771	0.1646	0	0
langgan	0	0	6	1	3/1	0,4771	0	0	0.2584
inference	2	0	0	1	3/1	0,4771	0.2142	0	0
anfis	4	0	0	1	3/1	0,4771	0.2638	0	0
learning	1	0	0	1	3/1	0,4771	0.1646	0	0

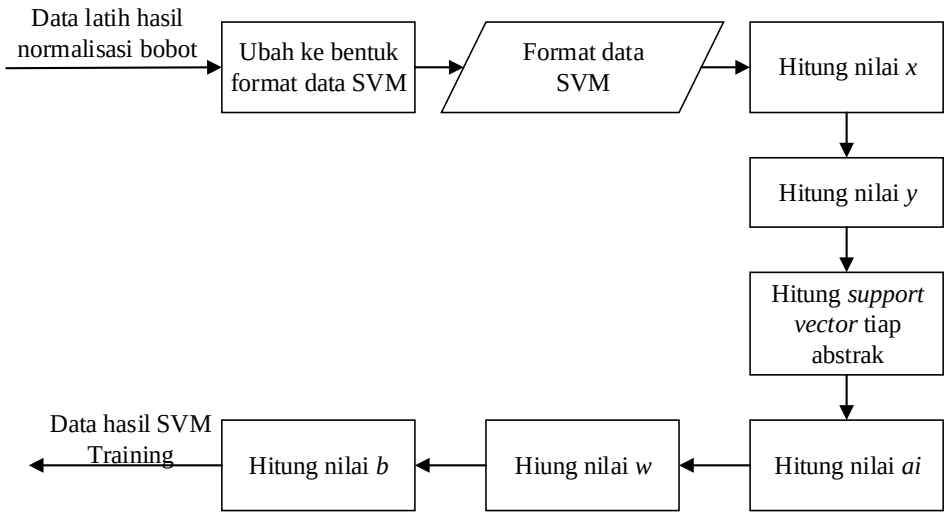
penelitian	0	1	0	1	3/1	0,4771	0	0.1925	0
media	0	0	1	1	3/1	0,4771	0	0	0.1453
tiru	2	0	0	1	3/1	0,4771	0.2142	0	0
perilaku	0	0	1	1	3/1	0,4771	0	0	0.1453
adaptive	2	0	0	1	3/1	0,4771	0.2142	0	0
sasar	0	0	1	1	3/1	0,4771	0	0	0.1453
provider	0	1	0	1	3/1	0,4771	0	0.1925	0
sms	0	0	1	1	3/1	0,4771	0	0	0.1453
penawaran	0	0	1	1	3/1	0,4771	0	0	0.1453
bayes	0	2	0	1	3/1	0,4771	0	0.2504	0
vector	0	2	0	1	3/1	0,4771	0	0.2504	0
relationship	0	0	3	1	3/1	0,4771	0	0	0.2146
fuzzy	5	0	0	1	3/1	0,4771	0.2797	0	0
crm	0	0	3	1	3/1	0,4771	0	0	0.2146
banjir	2	0	0	1	3/1	0,4771	0.2142	0	0
linear	0	2	0	1	3/1	0,4771	0	0.2504	0
monetary	0	0	1	1	3/1	0,4771	0	0	0.1453
neuro	2	0	0	1	3/1	0,4771	0.2142	0	0
performansi	1	0	0	1	3/1	0,4771	0.1646	0	0
syaraf	1	0	0	1	3/1	0,4771	0.1646	0	0
dasar	0	1	0	1	3/1	0,4771	0	0.1925	0
svm	0	2	0	1	3/1	0,4771	0	0.2504	0
strategi	0	0	1	1	3/1	0,4771	0	0	0.1453
recency	0	0	1	1	3/1	0,4771	0	0	0.1453
gabung	1	0	0	1	3/1	0,4771	0.1646	0	0
akun	0	1	0	1	3/1	0,4771	0	0.1925	0
prinsip	0	1	0	1	3/1	0,4771	0	0.1925	0
tour	0	0	1	1	3/1	0,4771	0	0	0.1453
dua	1	0	0	1	3/1	0,4771	0.1646	0	0
konsep	0	0	1	1	3/1	0,4771	0	0	0.1453
management	0	0	3	1	3/1	0,4771	0	0	0.2146
machine	0	2	0	1	3/1	0,4771	0	0.2504	0
klasifikasi	0	4	0	1	3/1	0,4771	0	0.3084	0
travel	0	0	1	1	3/1	0,4771	0	0	0.1453
naive	0	2	0	1	3/1	0,4771	0	0.2504	0
gateway	0	0	1	1	3/1	0,4771	0	0	0.1453
customer	0	0	3	1	3/1	0,4771	0	0	0.2146

Pembobotan dan normalisasi yang telah dilakukan untuk memberikan nilai bobot pada setiap *term*. Hasil dari nilai bobot tersebut kemudian akan digunakan dalam pembentukan vektor pada klasifikasi SVM.

3.5.2 Klasifikasi Menggunakan Metode SVM

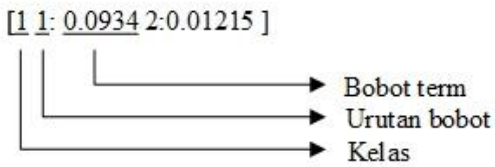
3.5.2.1 SVM Training

Pada proses pelatihan SVM bertujuan untuk menemukan vektor α, nilai w dan konstanta b untuk mendapatkan hyperplane terbaik. Dalam proses pelatihan dibutuhkan satu set *input-output* data atau dalam kasus ini dibutuhkan dokumen abstrak keilmuan A, B, C, D dan E dimana kategori abstrak keilmuan suatu dokumen telah diketahui. Berikut alur proses dapat dilihat pada gambar 3.12.



Gambar 3. 12 Proses SVM Training

Data hasil pembobotan dan normalisasi diubah ke dalam format data SVM. Dalam penelitian ini untuk format representasi data menggunakan format [*kelas urutan_bobot*$_1$*:bobot*$_1$ *urutan_bobot*$_n$*:bobot*$_n$].



Masukan yang pertama pada *kelas* adalah +1 atau -1 menyatakan dua label awalan yang diberikan, dimana angka 1 menyatakan data abstrak masuk dalam kelas positif dan label -1 menyatakan data abstrak masuk dalam kelas negatif.

Setelah kelas, *urutan_bobot* berikutnya menyatakan dimensi data ke-n dalam vektor. Selanjutnya *bobot* angka 0.0934 menyatakan nilai bobot dari *term* tersebut.

Sebagai contoh dalam pengubahan data teks menjadi data vektor dari hasil bobot normalisasi yang diambil dari contoh kasus yang telah melalui tahap pembobotan TF-IDF. Dokumen yang digunakan adalah S1, S2 dan S3 dengan sudah diberikan label, Sebagai inisialisasi dokumen abstrak keilmuan A diberi label 1 selain itu diberi label -1 maka S3 diberikan label 1 sedangkan S1 dan S2 diberi label -1. seperti pada Tabel 3.8

Tabel 3. 8 Abstrak menjadi Format Vektor

S1	0.0 0.0 0.0 0.0 0.0 0.0 0.1646 0.0 0.2142 0.1646 0.0 0.1646 0.2142 0.0 0.0 0.0 0.0 0.1646 0.0 0.1646 0.0 0.0 0.0 0.0 0.1646 0.0 0.0 0.1646 0.2142 0.0 0.0 0.0 0.1646 0.0 0.2797 0.0 0.0 0.0 0.1646 0.0 0.0 0.1646 0.0 0.2142 0.2638 0.1646 0.0 0.0 0.2142 0.0 0.2142 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.2797 0.0 0.2142 0.0 0.0 0.2142 0.1646 0.1646 0.0 0.0 0.0 0.0 0.1646 0.0 0.0 0.0 0.1646 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
Kelas	E
Vektor	[-1 1:0.0 2:0.0 3:0.0 4:0.0 5:0.0 6:0.0 7:0.1646 8:0.0 9:0.2142 10:0.1646 11:0.0 12:0.1646 13:0.2142 14:0.0 15:0.0 16:0.0 17:0.0 18:0.1646 19:0.0 20:0.1646 21:0.0 22:0.0 23:0.0 24:0.0 25:0.1646 26:0.0 27:0.0 28:0.1646 29:0.2142 30:0.0 31:0.0 32:0.0 33:0.1646 34:0.0 35:0.2797 36:0.0 37:0.0 38:0.0 39:0.1646 40:0.0 41:0.0 42:0.1646 43:0.0 44:0.2142 45:0.2638 46:0.1646 47:0.0 48:0.0 49:0.2142 50:0.0 51:0.2142 52:0.0 53:0.0 54:0.0 55:0.0 56:0.0 57:0.0 58:0.0 59:0.2797 60:0.0 61:0.2142 62:0.0 63:0.0 64:0.2142 65:0.1646 66:0.1646 67:0.0 68:0.0 69:0.0 70:0.0 71:0.1646 72:0.0 73:0.0 74:0.0 75:0.1646 76:0.0 77:0.0 78:0.0 79:0.0 80:0.0 81:0.0 82:0.0 83:0.0]
S2	0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.2504 0.0 0.0 0.0 0.2504 0.0 0.0 0.0 0.0 0.0 0.1925 0.2843 0.0 0.0 0.1925 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.2504 0.0 0.0 0.0 0.0 0.0 0.0 0.1925 0.0 0.0 0.0 0.0 0.0 0.1925 0.0 0.0 0.2504 0.2504 0.0 0.0 0.0 0.0 0.0 0.2504 0.0 0.0 0.0 0.0 0.1925 0.2504 0.0 0.0 0.0 0.1925 0.1925 0.0 0.0 0.0 0.0 0.2504 0.3084 0.0 0.2504 0.0 0.0
Kelas	E
Vektor	[-1 1:0.0 2:0.0 3:0.0 4:0.0 5:0.0 6:0.0 7:0.0 8:0.0 9:0.0 10:0.0 11:0.0 12:0.0 13:0.0 14:0.0 15:0.0 16:0.0 17:0.2504 18:0.0 19:0.0 20:0.0 21:0.2504 22:0.0 23:0.0 24:0.0 25:0.0 26:0.1925 27:0.2843 28:0.0 29:0.0 30:0.1925 31:0.0 32:0.0 33:0.0 34:0.0 35:0.0 36:0.0 37:0.0 38:0.0 39:0.0 40:0.2504 41:0.0 42:0.0 43:0.0 44:0.0 45:0.0 46:0.0 47:0.1925 48:0.0 49:0.0 50:0.0 51:0.0 52:0.0 53:0.1925 54:0.0 55:0.0 56:0.2504 57:0.2504 58:0.0 59:0.0 60:0.0 61:0.0 62:0.2504 63:0.0 64:0.0 65:0.0 66:0.0 67:0.1925 68:0.2504 69:0.0 70:0.0 71:0.0 72:0.1925 73:0.1925 74:0.0 75:0.0 76:0.0 77:0.0 78:0.2504 79:0.3084 80:0.0 81:0.2504 82:0.0 83:0.0]
S3	0.1890 0.1890 0.1453 0.0 0.1453 0.1453 0.0 0.1453 0.0 0.0 0.1453 0.0 0.0 0.1453 0.1890 0.1453 0.0 0.0 0.1453 0.0 0.0 0.0 0.1453 0.1453 0.0 0.0 0.0 0.0 0.0 0.0 0.1453 0.1453 0.0 0.1453 0.0 0.1890 0.1453 0.1890 0.0 0.0 0.1453 0.0 0.2584 0.0 0.0 0.0 0.0 0.0 0.1453 0.0 0.1453 0.0 0.1453 0.0 0.1453 0.1453 0.0 0.0 0.2146 0.0 0.2146 0.0 0.0 0.0 0.1453 0.0 0.0 0.0 0.0 0.0 0.1453 0.1453 0.0 0.0 0.0 0.1453 0.0 0.1453 0.2146 0.0 0.0 0.1453 0.0 0.1453 0.2146
Kelas	A
Vektor	[1 1:0.1890 2:0.1890 3:0.1453 4:0.0 5:0.1453 6:0.1453 7:0.0 8:0.1453 9:0.0 10:0.0 11:0.1453 12:0.0 13:0.0 14:0.1453 15:0.1890 16:0.1453 17:0.0 18:0.0 19:0.1453 20:0.0 21:0.0 22:0.0 23:0.1453 24:0.1453 25:0.0 26:0.0 27:0.0 28:0.0 29:0.0 30:0.0

	31:0.1453 32:0.1453 33:0.0 34:0.1453 35:0.0 36:0.1890 37:0.1453 38:0.1890 39:0.0 40:0.0 41:0.1453 42:0.0 43:0.2584 44:0.0 45:0.0 46:0.0 47:0.0 48:0.1453 49:0.0 50:0.1453 51:0.0 52:0.1453 53:0.0 54:0.1453 55:0.1453 56:0.0 57:0.0 58:0.2146 59:0.0 60:0.2146 61:0.0 62:0.0 63:0.1453 64:0.0 65:0.0 66:0.0 67:0.0 68:0.0 69:0.1453 70:0.1453 71: 0.0 72:0.0 73:0.0 74:0.1453 75:0.0 76:0.1453 77:0.2146 78:0.0 79:0.0 80:0.1453 81:0.0 82:0.1453 83:0.2146]

Tahap ini akan terus berulang hingga semua *term* pada abstrak terwakili oleh format data vektor. Untuk *term* yang sama muncul lebih dari sekali dalam sebuah abstrak akan diwakili sebuah data vektor saja dengan nilai bobot yang bersesuaian. Berikut gambaran vector Abstrak yang disajikan dalam bentuk tabel 3.9.

Tabel 3. 9 Abstrak yang sudah memiliki label

	S1	S2	S3		S1	S2	S3
$term_1$	0.1997	0.0	0.0	$term_{31}$	0.1997	0.0	0.0
$term_2$	0.1997	0.0	0.0	$term_{32}$	0.1997	0.0	0.0
$term_3$	0.0	0.0	0.1822	$term_{33}$	0.0	0.2912	0.0
$term_4$	0.1997	0.0	0.0	$term_{34}$	0.0	0.0	0.1822
$term_5$	0.0	0.2238	0.0	$term_{35}$	0.0	0.2238	0.0
$term_6$	0.1997	0.0	0.0	$term_{36}$	0.0	0.0	0.2371
$term_7$	0.1997	0.0	0.0	$term_{37}$	0.1997	0.0	0.0
$term_8$	0.0	0.0	0.1822	$term_{38}$	0.1997	0.0	0.0
$term_9$	0.0	0.2912	0.0	$term_{39}$	0.0	0.0	0.2371
$term_{10}$	0.1997	0.0	0.0	$term_{40}$	0.0	0.2912	0.0
$term_{11}$	0.1997	0.0	0.0	$term_{41}$	0.0	0.0	0.2371
$term_{12}$	0.0	0.2238	0.0	$term_{42}$	0.0	0.0	0.2371
$term_{13}$	0.0	0.2238	0.0	$term_{43}$	0.2950	0.0	0.0
$term_{14}$	0.0	0.0	0.1822	$term_{44}$	0.1997	0.0	0.0
$term_{15}$	0.0	0.0	0.2371	$term_{45}$	0.0	0.0	0.2371
$term_{16}$	0.1997	0.0	0.0	$term_{46}$	0.0	0.2238	0.0
$term_{17}$	0.2599	0.0	0.0	$term_{47}$	0.1997	0.0	0.0
$term_{18}$	0.0	0.2238	0.0	$term_{48}$	0.0	0.2238	0.0
$term_{19}$	0.0	0.2238	0.0	$term_{49}$	0.0	0.0	0.2919
$term_{20}$	0.0	0.0	0.1822	$term_{50}$	0.0	0.0	0.1822
$term_{21}$	0.2950	0.0	0.0	$term_{51}$	0.1997	0.0	0.0
$term_{22}$	0.1997	0.0	0.0	$term_{52}$	0.0	0.0	0.1822
$term_{23}$	0.0	0.0	0.1822	$term_{53}$	0.0	0.2912	0.0
$term_{24}$	0.0	0.0	0.1822	$term_{54}$	0.1997	0.0	0.0
$term_{25}$	0.0	0.0	0.1822	$term_{55}$	0.0	0.0	0.1822
$term_{26}$	0.0	0.0	0.1822	$term_{56}$	0.1997	0.0	0.0
$term_{27}$	0.0	0.0	0.1822	$term_{57}$	0.0	0.3306	0.2371
$term_{28}$	0.0	0.0	0.1822	$term_{58}$	0.0	0.2238	0.0
$term_{29}$	0.1997	0.0	0.0	$term_{59}$	0.0	0.0	0.2371
$term_{30}$	0.0	0.2238	0.0	$term_{60}$	0.0	0.2238	0.0
Y	-1	-1	1	*Y*	-1	-1	1

Tahap selanjutnya menghitung nilai x. nilai *X* pada tabel akan digunakan untuk perhitungan kernel. Untuk $x_1 = \{term_1, term_2, \ldots\ldots term_i\}$ adalah seluruh nilai yang diambil dari nilai x pada kolom S1, x_2 =S2 dan x_3 =S3. Sehingga setiap abstrak untuk nilai x_1, x_2, x_3 sesuai hasil pembobotan tf-idf dapat dilihat pada tabel 3.10.

Tabel 3. 10 Nilai x1, x2, x3

x_1			x_2			x_3		
[0.1997	0.1997	0.0	[0.0	0.0	0.0	[0.0	0.0	0.1822
0.1997	0.0	0.1997	0.0	0.2238	0.0	0.0	0.0	0.0
0.1997	0.0	0.0	0.0	0.0	0.2912	0.0	0.1822	0.0
0.1997	0.1997	0.0	0.0	0.0	0.2238	0.0	0.0	0.0
0.0	0.0	0.0	0.2238	0.0	0.0	0.0	0.1822	0.2371
0.1997	0.2599	0.0	0.0	0.0	0.2238	0.0	0.1822	0.0
0.0	0.0	0.2950	0.2238	0.0	0.0	0.0	0.1822	0.1822
0.1997	0.0	0.0	0.0	0.0	0.0	0.1822	0.1822	0.1822
0.0	0.0	0.0	0.0	0.0	0.0	0.1822	0.0	0.0
0.0	0.1997	0.0	0.0	0.0	0.2238	0.0	0.0	0.0
0.1997	0.1997	0.0	0.	0.0	0.2912	0.1822	0.0	0.2371
0.0	0.0	0.0	0.0	0.2238	0.0	0.0	0.0	0.2371
0.1997	0.1997	0.0	0.0	0.0	0.0	0.0	0.2371	0.2371
0.0	0.0	0.0	0.2912	0.0	0.0	0.0	0.0	0.2371
0.2950	0.1997	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.1997	0.0	0.2238	0.0	0.2238	0.0	0.0	0.0
0.0	0.0	0.1997	0.0	0.0	0.0	0.2919	0.1822	0.0
0.0	0.0	0.1997	0.0	0.2912	0.0	0.1822	0.0	0.0
0.0	0.1997	0.0	0.0	0.0	0.3306	0.1822	0.0	0.2371
0.0	0.0	0.0]	0.2238	0.0	0.2238]	0.0	0.2371	0.0]

Selanjutnya yaitu melakukan kernelisasi menggunakan fungsi Kernel linier $K(x_i, x_j) = x_i x_j^T$. Untuk data yang pertama $x_i x_j^T$, maka dilakukan perhitungan matriks yang dapat dilihat pada tabel 3.11.

Tabel 3. 11 Perhitungan Nilai x_1 dengan Kernel

x_1			x_1^T			$x_1 x_1^T = x_1 * x_1^T$
[0.1997	0.1997	0.0	[0.1997;	0.1997;	0.0;	0.9993
0.1997	0.0	0.1997	0.1997;	0.0;	0.1997;	
0.1997	0.0	0.0	0.1997;	0.0;	0.0;	
0.1997	0.1997	0.0	0.1997;	0.1997;	0.0;	
0.0	0.0	0.0	0.0;	0.0;	0.0;	
0.1997	0.2599	0.0	0.1997;	0.2599;	0.0;	
0.0	0.0	0.2950	0.0;	0.0;	0.2950;	
0.1997	0.0	0.0	0.1997;	0.0;	0.0;	
0.0	0.0	0.0	0.0;	0.0;	0.0;	
0.0	0.1997	0.0	0.0;	0.1997;	0.0;	
0.1997	0.1997	0.0	0.1997;	0.1997;	0.0;	
0.0	0.0	0.0	0.0;	0.0;	0.0;	
0.1997	0.1997	0.0	0.1997;	0.1997;	0.0;	

0.0	0.0	0.0	0.0;	0.0;	0.0;	
0.2950	0.1997	0.0	0.2950;	0.1997;	0.0;	
0.0	0.1997	0.0	0.0;	0.1997;	0.0;	
0.0	0.0	0.1997	0.0;	0.0;	0.1997;	
0.0	0.0	0.1997	0.0;	0.0;	0.1997;	
0.0	0.1997	0.0	0.0;	0.1997;	0.0;	
0.0	0.0	0.0]	0.0;	0.0;	0.0]	

Maka untuk nilai $x_i x_j^T$ selanjutnya didapatkan nilai pada tabel 3.12.

Tabel 3. 12 Hasil Perhitungan Nilai x dengan Kernel

$x_1x_1^T$	$x_1x_2^T$	$x_1x_3^T$	$x_2x_1^T$	$x_2x_2^T$	$x_2x_3^T$	$x_3x_1^T$	$x_3x_2^T$	$x_3x_3^T$
0.9993	0	0	0	0.9994	0.0784	0	0.0784	0.9997

Setelah dilakukan perhitungan pada seluruh nilai X pada data abstrak, maka matriks yang terbentuk dari hasil perhitungan $x_i x_j^T$ adalah sebagai berikut.

$$x_i x_j^T = \begin{bmatrix} x1x1 & x2x1 & x3x1 \\ x1x2 & x2x2 & x3x2 \\ x1x3 & x2x3 & x3x3 \end{bmatrix}$$

$$x_i x_j^T = \begin{bmatrix} 0.9993 & 0 & 0 \\ 0 & 0.9994 & 0.0784 \\ 0 & 0.0784 & 0.9997 \end{bmatrix}$$

Kemudian tahap selanjutnya adalah melakukan perhitungan terhadap *y*. Untuk nilai y adalah nilai dari label yang diberikan. Nilai y tersebut dapat dilihat pada tabel 3.13.

Tabel 3. 13 Nilai Label pada y

y_1	y_2	y_3
-1	-1	1

Setelahnya nilai *y* melakukan perhitungan dengan kernel seperti yang dilakukan pada nilai *x.* Hasil dari perhitungan tersebut dapat dilihat pada tabel 3.14.

Tabel 3. 14 Perhitungan Nilai y_1 dengan Kernel

$y_1y_1^T$	$y_1y_2^T$	$y_1y_3^T$	$y_2y_1^T$	$y_2y_2^T$	$y_2y_3^T$	$y_3y_1^T$	$y_3y_2^T$	$y_3y_3^T$
1	1	-1	1	1	-1	-1	-1	1

Sehingga matriks yang terbentuk dari hasil perhitungan $y_i y_j^T$ adalah hasilnya sebagai berikut.

$$y_i y_j^T = \begin{bmatrix} 1 & 1 & -1 \\ 1 & 1 & -1 \\ -1 & -1 & 1 \end{bmatrix}$$

Tahap selanjutnya mengubah setiap abstrak menjadi nilai vektor (*support vector*) = $\binom{x}{y}$ agar mendapatkan nilai a_i. Nilai x didapatkan menggunakan persamaan (2.22) kernel linear untuk x berikut.

$$\sum_{i=1,j=1}^{n} \boldsymbol{x_i x_j^T}, (\boldsymbol{i, j = 1, .., n}) \qquad (2.22)$$

Hasil perhitungan $x_i x_j^T$ yang telah dilakukan akan masuk ke dalam matriks.

$$x_i x_j^T = \begin{bmatrix} 0.9993 & 0 & 0 \\ 0 & 0.9994 & 0.0784 \\ 0 & 0.0784 & 0.9997 \end{bmatrix}$$

$X_{S1} = x_1 x_1^T + x_1 x_2^T + x_1 x_3^T = 0{,}9993+0+0 = 0{,}9993$

$X_{S2} = x_2 x_1^T + x_2 x_2^T + x_2 x_3^T = 0+0{,}9994+0{,}0784 = 1{,}0778$

$X_{S3} = x_3 x_1^T + x_3 x_2^T + x_3 x_3^T = 0+0{,}0784+0{,}9997 = 1{,}0781$

Sehingga didapatkan untuk nilai *x* pada setiap abstrak pada Tabel 3.15.

Tabel 3. 15 Nilai x pada setip Abstrak

Abstrak	S1	S2	S3
X	0,9993	1,0778	1,0781

Nilai y didapatkan menggunakan persamaan (2.23) kernel linear untuk y berikut.

$$\sum_{i=1,j=1}^{1} \boldsymbol{y_i y_j^T}, (\boldsymbol{i, j = 1, .., n}) \qquad (2.23)$$

Dengan perhitungan sebagai berikut.

$$y_i y_j^T = \begin{bmatrix} 1 & 1 & -1 \\ 1 & 1 & -1 \\ -1 & -1 & 1 \end{bmatrix}$$

$Y_{S1} = y_1 y_1^T + y_1 y_2^T + y_1 y_3^T = 1 + 1 + -1 = -1$

$Y_{S2} = y_2 y_1^T + y_2 y_2^T + y_2 y_3^T = 1 + 1 + -1 = -1$

$Y_{S3} = y_3 y_1^T + y_3 y_2^T + y_3 y_3^T = -1 + -1 + 1 = 1$

sehingga didapatkan untuk nilai *y* pada setiap abstrak pada Tabel 3.16**.**

Tabel 3. 16 Hasil Nilai y pada setiap Abstrak

Absrak	S1	S2	S3
Y	-1	-1	1

Setelah nilai x dan y didapatkan, substitusikan nilai tersebut ke persamaan

$$\boldsymbol{\phi}\begin{bmatrix}\boldsymbol{x}\\ \boldsymbol{y}\end{bmatrix} = \begin{cases} \sqrt{\boldsymbol{x_n^2 + y_n^2}} > \boldsymbol{2\ maka} \begin{bmatrix} \sqrt{\boldsymbol{x_n^2 + y_n^2}} - \boldsymbol{x} + |\boldsymbol{x} - \boldsymbol{y}| \\ \sqrt{\boldsymbol{x_n^2 + y_n^2}} - \boldsymbol{y} + |\boldsymbol{x} - \boldsymbol{y}| \end{bmatrix} \\ \sqrt{\boldsymbol{x_n^2 + y_n^2}} \leq \boldsymbol{2\ maka} \begin{bmatrix}\boldsymbol{x}\\ \boldsymbol{y}\end{bmatrix} \end{cases} \quad (2.21)$$

Nilai x_n yang didapat dari x_{s3} dan y_n dari y_{s3} yang disubtitusikan ke dalam persamaan $\sqrt{x_n^2 + y_n^2} = \sqrt{1{,}0781^2 + 1^2} = 1{,}470475981 \leq 2$. Karena hasil yang didapatkan $\sqrt{x_n^2 + y_n^2} \leq 2$. Maka,

$$\phi\begin{bmatrix}x\\ y\end{bmatrix} = \begin{bmatrix}x\\ y\end{bmatrix}$$

$$\phi(\text{S1}) = \begin{bmatrix}0{,}9993\\ (-1)\end{bmatrix},\ \phi(\text{S2}) = \begin{bmatrix}1{,}0778\\ (-1)\end{bmatrix},\ \phi(\text{S3}) = \begin{bmatrix}1{,}0781\\ 1\end{bmatrix}$$

Setelah dilakukan perhitungan terhadap seluruh abstrak, maka didapatkan hasilnya pada Tabel 3.17.

Tabel 3. 17 Nilai $\boldsymbol{\phi}$ pada tiap Abstrak

Absrak	$\boldsymbol{\phi}$(S1)	$\boldsymbol{\phi}$(S2)	$\boldsymbol{\phi}$(S3)
Support Vector	$\begin{bmatrix}\boldsymbol{0,9993}\\ \boldsymbol{(-1)}\end{bmatrix}$	$\begin{bmatrix}\boldsymbol{1,0778}\\ \boldsymbol{(-1)}\end{bmatrix}$	$\begin{bmatrix}\boldsymbol{1,0781}\\ \boldsymbol{1}\end{bmatrix}$

Setelah itu masing-masing *support vector* diberi nilai bias 1. Untuk mendapatkan jarak tegak lurus yang optimal dengan mempertimbangkan vektor positif, serta membantu mendapatkan nilai b atau nilai *hyperplane*, hasilnya dapat dilihat pada Tabel 3.18.

Tabel 3. 18 Support Vector Bias

Absrak	S1	S2	S3
Support Vector Bias	$\begin{bmatrix}\mathbf{0,9993}\\(-\mathbf{1})\\\mathbf{1}\end{bmatrix}$	$\begin{bmatrix}\mathbf{1,0778}\\(-\mathbf{1})\\\mathbf{1}\end{bmatrix}$	$\begin{bmatrix}\mathbf{1,0781}\\\mathbf{1}\\\mathbf{1}\end{bmatrix}$

Setelah mendapat nilai support Vector, Kemudian tahap selanjutnya adalah mencari nilai a_i, didapatkan dengan mengalikan setiap data abstrak menggunakan persamaan sebagai berikut.

$$\sum_{i=1,j=1}^{n} \boldsymbol{a_i S_i^T S_j} \qquad (2.24)$$

Dengan perhitungan pada S1 sebagai berikut.

$$a_1 \begin{bmatrix}0,9993\\(-1)\\1\end{bmatrix}^T * \begin{bmatrix}0,9993\\(-1)\\1\end{bmatrix} = 2.9986a_1$$

$$a_2 \begin{bmatrix}0,9993\\(-1)\\1\end{bmatrix}^T * \begin{bmatrix}1,0778\\(-1)\\1\end{bmatrix} = 3.0770a_2$$

$$a_3 \begin{bmatrix}0,9993\\(-1)\\1\end{bmatrix}^T * \begin{bmatrix}1,0781\\1\\1\end{bmatrix} = 1.0773a_3$$

Dengan perhitungan pada S2 sebagai berikut.

$$a_1 \begin{bmatrix}1,0778\\(-1)\\1\end{bmatrix}^T * \begin{bmatrix}0,9993\\(-1)\\1\end{bmatrix} = 3.0770a_1$$

$$a_2 \begin{bmatrix}1,0778\\(-1)\\1\end{bmatrix}^T * \begin{bmatrix}1,0778\\(-1)\\1\end{bmatrix} = 3.1617a_2$$

$$a_3 \begin{bmatrix}1,0778\\(-1)\\1\end{bmatrix}^T * \begin{bmatrix}1,0781\\1\\1\end{bmatrix} = 1.1620a_3$$

Dengan perhitungan pada S3 sebagai berikut.

$$a_1 \begin{bmatrix}1,0781\\1\\1\end{bmatrix}^T * \begin{bmatrix}0,9993\\(-1)\\1\end{bmatrix} = 1.0773a_1$$

$$a_2 \begin{bmatrix} 1{,}0781 \\ 1 \\ 1 \end{bmatrix}^T * \begin{bmatrix} 1{,}0778 \\ (-1) \\ 1 \end{bmatrix} = 1.1620a_2$$

$$a_3 \begin{bmatrix} 1{,}0781 \\ 1 \\ 1 \end{bmatrix}^T * \begin{bmatrix} 1{,}0781 \\ 1 \\ 1 \end{bmatrix} = 3.1623a_3$$

Setelah dilakukan perhitungan pada seluruh abstrak Kemudian cari parameter a_i menggunakan persamaan (2.25), dengan cara substitusikan nilai hasil dari perhitungan menggunakan persamaan (2.24), sehingga bentuknya dapat dilihat sebagai berikut.

$$\sum_{i=1,j=1}^{n} \boldsymbol{a_i S_i^T S_j = y_i} \qquad (2.25)$$

2.9986a_1+3.0770a_2+1.0773a_3 = -1

3.0770a_1+3.1617a_2+1.1620a_3 = -1

1.0773a_1+1.1620a_2+3.1623a_3 = 1

Dari persamaan diatas, untuk mendapatkan nilai a_1, a_2, a_3 maka di gunakan metode *gauss jordan* dan subtitusi. Yang di hitung menggunakan fungsi matlab. Sehingga didapatlah nilai untuk a_1 sampai a_3 berikut.

a_1 = -0.0020, a_2 = -0.4981, a_3 = 0.4999

Karena nilai a_i adalah bernilai nol atau positif, maka nilai a_1 dan nilai a_2 tidak dihitung ketahap selanjutnya. Setelah didapatkan nilai a_i masukkan ke persamaan (2.25) sesuai Tabel 3.19 untuk mendapatkan nilai w dan b.

$$\widetilde{W} = \sum_{i=1}^{n} a_i S_i \qquad (2.26)$$

$$\widetilde{W} = 0.4999 \begin{bmatrix} 1{,}0781 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.5389 \\ 0.4999 \\ 0.4999 \end{bmatrix} = \begin{bmatrix} 0.5389 \\ 0.4999 \\ 0.4999 \end{bmatrix}$$

Kemudian hasil yang didapatkan melalui perhitungan dengan menggunakan persamaan (2.26), digunakan persamaan (2.27) dengan hasil yang didapatkan :

$w = \begin{bmatrix} 0.5389 \\ 0.4999 \end{bmatrix}$, dan dengan $b = 0.4999$

Sedemikian sehingga didapatkanlah nilai *hyperplane* untuk mengklasifikasikan kedua kelas, yaitu 0.4999.

3.5.2.1.1 *Multiclass*

Proses klasifikasi menggunkan metode SVM membagi 2 kelas abstrak menjadi kelas positif dan negatif. Ada 2 proses yang dilakukan dalam mengklasifikasi , yaitu proses *training* dan *classifier*. Proses *training* dilakukan untuk menghasilkan model fitur yang akan digunakan dalam menentukan klasifikasi abstrak kedalam kelas positif atau negatif.

Seperti yang telah dijelaskan sebelumnya, SVM sebenarnya digunakan untuk klasifikasi biner (dua kelas), dalam kasus multiclass, SVM yang digunakan adalah dengan mengkombinasikan beberapa SVM biner tersebut.

Metode "One-against-all" salah satunya. Dibangun sejumlah *k* SVM biner, dengan *k* adalah jumlah kelas. Untuk setiap model klasifikasi ke-i dilatih dengan menggunakan keseluaruhan data, untuk mencari solusi permasalahan Contohnya, untuk persoalan klasifikasi dengan 4 buah jumlah kelas, digunakan 4 buah SVM biner pada tabel 2.4 dan penggunaannya pada pengklasifikasian data baru.

Dari Tabel 2.4 dan gambar 2.1 Untuk penggunaannya dapat digambarkan dengan perulangan sebagai berikut.

Perulangan 1 : SVM label positif (A) dengan label negatif (B,C,D,E)
Jika hasil adalah label positif, dokumen = A
Jika hasil adalah label negatif = lanjut perulangan 2

Perulangan 2 : SVM label positif (B) dengan label negatif (C,D,E)
Jika hasil adalah label positif, dokumen = B
Jika hasil adalah label negatif = lanjut perulangan 3

Perulangan 3 : SVM label positif (C) dengan label negatif (D,E)
Jika hasil adalah label positif, dokumen = C
Jika hasil adalah label negatif = lanjut perulangan 4

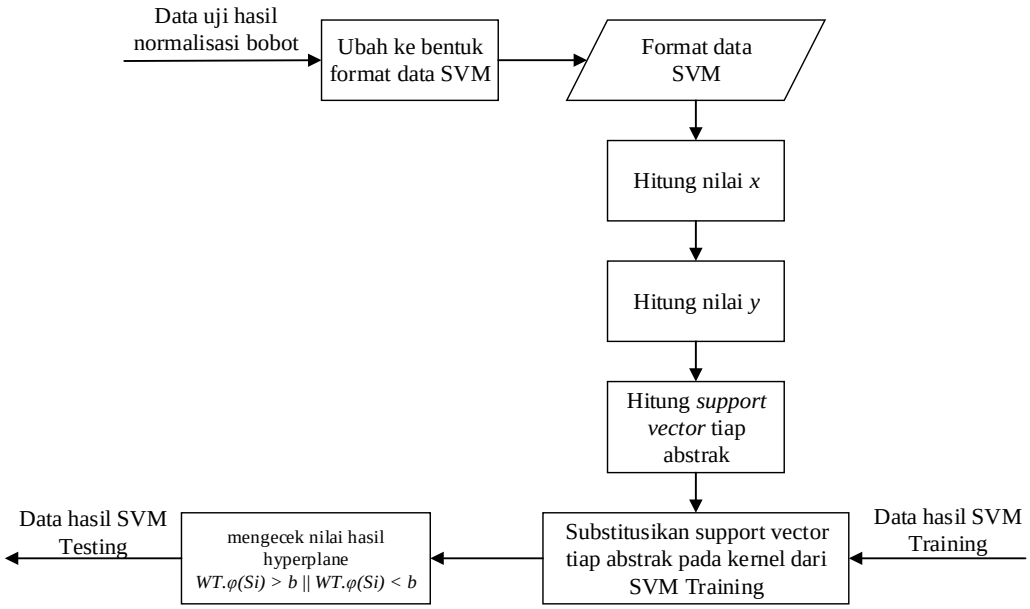
Perulangan 3 : SVM label positif (D) dengan label negatif (E)

Jika hasil adalah label positif, dokumen = D

Jika hasil adalah label negatif, dokumen = E

3.5.2.2 SVM Testing

Setelah mendapatkan nilai w dan b atau *hyperplane* dari hasil SVM Training, selanjutnya dapat menentukan data abstrak masuk dalam kelas positif atau negatif dengan nilai w dan *hyperplane* tersebut. Jika nilai hasil uji lebih besar dari nilai *hyperplane* maka abstrak tersebut masuk dalam kelas positif, jika lebih kecil dari nilai *hyperplane* maka abstrak tersebut masuk dalam kelas negatif. Berikut alur proses dapat dilihat pada gambar 3.13.



Gambar 3. 13 Proses SVM Testing

Sebagai contoh, data uji dilakukan pada Sebuah Abstrak Testing yang kemudian akan diolah melalui Preprocessing.

Term $S_{Testing}$	mendiagnosa sakit akurasi data sangat hasil solusi butuh akurasi metode diagnosa sakit jadi timbang metode metode naive bayes metode backpropagation neural network metode sama akurasi data

$S_{Testing}$ yang diuji akan dihitung berdasarkan Abstrak pelatihan dalam kasus ini adalah S1, S2 dan S3 yang ada pada Analisis Proses SVM Training. Berikut gambaran vector Abstrak yang disajikan dalam bentuk tabel 3.19

Tabel 3. 19 Vektor Abstrak

	S1	S2	S3	$S_{Testing}$		S1	S2	S3	$S_{Testing}$
$term_1$	0.1997	0.0	0.0	0.0	$term_{31}$	0.0	0.0	0.1822	0.0
$term_2$	0.1997	0.0	0.0	0.0	$term_{32}$	0.0	0.0	0.1822	0.0
$term_3$	0.0	0.0	0.0	0.2745	$term_{33}$	0.0	0.0	0.1822	0.0
$term_4$	0.0	0.0	0.0	0.2110	$term_{34}$	0.0	0.0	0.1822	0.0
$term_5$	0.0	0.0	0.1822	0.0	$term_{35}$	0.0	0.2238	0.0	0.0
$term_6$	0.0	0.0	0.0	0.2110	$term_{36}$	0.1997	0.0	0.0	0.0
$term_7$	0.1997	0.0	0.0	0.0	$term_{37}$	0.1997	0.0	0.0	0.0
$term_8$	0.0	0.2238	0.0	0.0	$term_{38}$	0.1997	0.0	0.0	0.0
$term_9$	0.1997	0.0	0.0	0.0	$term_{39}$	0.0	0.2912	0.0	0.0
$term_{10}$	0.0	0.0	0.1822	0.0	$term_{40}$	0.0	0.0	0.1822	0.0
$term_{11}$	0.1997	0.0	0.0	0.0	$term_{41}$	0.0	0.0	0.2371	0.0
$term_{12}$	0.0	0.2912	0.0	0.0	$term_{42}$	0.0	0.2238	0.0	0.0
$term_{13}$	0.0	0.0	0.0	0.2110	$term_{43}$	0.1997	0.0	0.0	0.0
$term_{14}$	0.1997	0.0	0.0	0.0	$term_{44}$	0.0	0.0	0.0	0.2745
$term_{15}$	0.1997	0.0	0.0	0.0	$term_{45}$	0.1997	0.0	0.0	0.0
$term_{16}$	0.0	0.2238	0.0	0.0	$term_{46}$	0.0	0.0	0.0	0.2110
$term_{17}$	0.0	0.0	0.0	0.2110	$term_{47}$	0.0	0.0	0.0	0.2110
$term_{18}$	0.0	0.0	0.1822	0.0	$term_{48}$	0.0	0.0	0.2371	0.0
$term_{19}$	0.0	0.2238	0.0	0.0	$term_{49}$	0.0	0.2912	0.0	0.0
$term_{20}$	0.0	0.0	0.2371	0.0	$term_{50}$	0.0	0.0	0.2371	0.0
$term_{21}$	0.1997	0.0	0.0	0.0	$term_{51}$	0.0	0.0	0.2371	0.0
$term_{22}$	0.2599	0.0	0.0	0.0	$term_{52}$	0.1997	0.0	0.0	0.0
$term_{23}$	0.0	0.2238	0.0	0.0	$term_{53}$	0.2950	0.0	0.0	0.0
$term_{24}$	0.0	0.0	0.0	0.3585	$term_{54}$	0.0	0.0	0.0	0.2110
$term_{25}$	0.0	0.0	0.1822	0.0	$term_{55}$	0.0	0.0	0.2371	0.0
$term_{26}$	0.0	0.2238	0.0	0.0	$term_{56}$	0.0	0.2238	0.0	0.0
$term_{27}$	0.1997	0.0	0.0	0.0	$term_{57}$	0.1997	0.0	0.0	0.0
$term_{28}$	0.2950	0.0	0.0	0.0	$term_{58}$	0.0	0.0	0.0	0.2110
$term_{29}$	0.0	0.0	0.1822	0.0	$term_{59}$	0.0	0.2238	0.0	0.0
$term_{30}$	0.0	0.0	0.1822	0.0	$term_{60}$	0.0	0.0	0.1822	0.0
Y	-1	-1	1	0	Y	-1	-1	1	0

Tahap selanjutnya menghitung nilai x. Nilai X pada tabel akan di gunakan untuk perhitungan dot product. Untuk $x_1 = \{term_1, term_2, \ldots\ldots term_i\}$ adalah seluruh nilai yang diambil dari nilai x pada kolom S1 sampai $S_{testing}$. Sehingga setiap abstrak untuk nilai $x_{1,}$ sesuai hasil pembobotan tf-idf dapat dilihat pada tabel 3.20

Tabel 3. 20 Nilai yang diambil dari Hasil Pembobotan

x_1		x_2		x_3		$x_{testing}$	
[0.1997	0.1997	[0.0	0.0	[0.0	0.0	[0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.2745	0.2110
0.0	0.0	0.0	0.0	0.1822	0.0	0.0	0.2110
0.1997	0.0	0.0	0.2238	0.0	0.0	0.0	0.0
0.1997	0.0	0.0	0.0	0.0	0.1822	0.0	0.0
0.1997	0.0	0.0	0.2912	0.0	0.0	0.0	0.0
0.0	0.1997	0.0	0.0	0.0	0.0	0.2110	0.0
0.1997	0.0	0.0	0.2238	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.1822	0.2110	0.0
0.0	0.0	0.2238	0.0	0.0	0.2371	0.0	0.0
0.1997	0.2599	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.2238	0.0	0.0	0.0	0.0	0.3585
0.0	0.0	0.0	0.2238	0.1822	0.0	0.0	0.0
0.1997	0.2950	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.1822	0.1822	0.0	0.0
0.0	0.0	0.0	0.0	0.1822	0.1822	0.0	0.0
0.0	0.0	0.0	0.0	0.1822	0.1822	0.0	0.0
0.0	0.1997	0.2238	0.0	0.0	0.0	0.0	0.0
0.1997	0.1997	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.2919	0.0	0.0	0.1822	0.0	0.0
0.0	0.0	0.0	0.2238	0.2371	0.0	0.0	0.0
0.1997	0.0	0.0	0.0	0.0	0.0	0.0	0.2745
0.1997	0.0	0.0	0.0	0.0	0.0	0.0	0.2110
0.0	0.0	0.0	0.0	0.0	0.2371	0.2110	0.0
0.0	0.0	0.2919	0.0	0.0	0.2371	0.0	0.0
0.0	0.1997	0.0	0.0	0.2371	0.0	0.0	0.0
0.2950	0.0	0.0	0.0	0.0	0.0	0.0	0.2110
0.0	0.0	0.0	0.2238	0.2371	0.0	0.0	0.0
0.1997	0.0	0.0	0.0	0.0	0.0	0.0	0.2110
0.0	0.0]	0.2238	0.0]	0.0	0.1822]	0.0	0.0]

Selanjutnya yaitu melakukan kernelisasi menggunakan fungsi Kernel linier $K(x_i,x_j) = x_i x_j^T$. Untuk data yang pertama $x_i x_j^T$, maka dilakukan perhitungan yang dapat dilihat pada tabel 3.21

Tabel 3. 21 Perhitungan $x_i x_j^T$

x_1			x_1^T			$x_1 x_1^T = x_1 * x_1^T$
0.1997	0.1997	0.0	0.1997;	0.1997;	0.0;	0.8797
0.0	0.0	0.0	0.0;	0.0;	0.0;	
0.1997	0.0	0.1997	0.1997;	0.0;	0.1997;	
0.0	0.1997	0.0	0.0;	0.1997;	0.0;	
0.0	0.1997	0.1997	0.0;	0.1997;	0.1997;	
0.0	0.0	0.0	0.0;	0.0;	0.0;	
0.0	0.0	0.1997	0.0;	0.0;	0.1997;	
0.2599	0.0	0.0	0.2599;	0.0;	0.0;	
0.0	0.0	0.1997	0.0;	0.0;	0.1997;	
0.2950	0.0	0.0	0.2950;	0.0;	0.0;	
0.0	0.0	0.0	0.0;	0.0;	0.0;	
0.0	0.0	0.1997	0.0;	0.0;	0.1997;	

0.1997	0.1997	0.0	0.1997;	0.1997;	0.0;	
0.0	0.0	0.0	0.0;	0.0;	0.0;	
0.1997	0.0	0.1997	0.1997;	0.0;	0.1997;	
0.0	0.0	0.0	0.0;	0.0;	0.0;	
0.0	0.0	0.0	0.0;	0.0;	0.0;	
0.1997	0.2950	0.0	0.1997;	0.2950;	0.0;	
0.0	0.0	0.1997	0.0;	0.0;	0.1997;	
0.0	0.0	0.0	0.0;	0.0;	0.0;	

Maka untuk nilai $x_i x_j^T$ selanjutnya didapatkan nilai pada tabel 3.22

Tabel 3. 22 Hasil Perhitungan $x_i x_j^T$

$x_1x_1^T$	$x_1x_2^T$	$x_1x_3^T$	$x_1x_4^T$	$x_2x_1^T$	$x_2x_2^T$	$x_2x_3^T$	$x_2x_4^T$
0.8797	0	0	0	0	0.7052	0	0
$x_3x_1^T$	$x_3x_2^T$	$x_3x_3^T$	$x_3x_4^T$	$x_4x_1^T$	$x_4x_2^T$	$x_4x_3^T$	$x_4x_4^T$
0	0	0.7357	0	0	0	0	0.6354

Setelah dilakukan perhitungan pada seluruh nilai X pada data abstrak, maka matriks yang terbentuk dari hasil perhitungan $x_i x_j^T$ adalah sebagai berikut:

$$x_i x_j^T = \begin{bmatrix} x1x1 & x2x1 & x3x1 & x4x1 \\ x1x2 & x2x2 & x3x2 & x4x2 \\ x1x3 & x2x3 & x3x3 & x3x4 \\ x1x4 & x2x4 & x3x4 & x4x4 \end{bmatrix}$$

$$x_i x_j^T = \begin{bmatrix} 0.8797 & 0 & 0 & 0 \\ 0 & 0.7052 & 0 & 0 \\ 0 & 0 & 0.7357 & 0 \\ 0 & 0 & 0 & 0.6354 \end{bmatrix}$$

Kemudian tahap selanjutnya adalah melakukan perhitungan terhadap *y*. Untuk nilai y adalah nilai dari label yang diberikan. Nilai y tersebut dapat dilihat pada tabel 3.23.

Tabel 3. 23 Nilai Label pada y

y_1	y_2	y_3	y_4
-1	-1	1	0

Setelahnya nilai *y* melakukan perhitungan dengan kernel seperti yang dilakukan pada nilai *x*. Hasil dari perhitungan tersebut dapat dilihat pada tabel 3.24.

Tabel 3. 24 Perhitungan Nilai y_1 dengan Kernel

$y_1y_1^T$	$y_1y_2^T$	$y_1y_3^T$	$y_1y_4^T$	$y_2y_1^T$	$y_2y_2^T$	$y_2y_3^T$	$y_2y_4^T$
1	1	-1	0	1	1	-1	0
$y_3y_1^T$	$y_3y_2^T$	$y_3y_3^T$	$y_3y_4^T$	$y_4y_1^T$	$y_4y_2^T$	$y_4y_3^T$	$y_4y_4^T$
-1	-1	1	0	0	0	0	0

Sehingga matriks yang terbentuk dari hasil perhitungan $y_iy_j^T$ adalah hasilnya sebagai berikut:

$$y_iy_j^T = \begin{bmatrix} 1 & 1 & -1 & 0 \\ 1 & 1 & -1 & 0 \\ -1 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Tahap selanjutnya mengubah setiap abstrak menjadi nilai vektor (*support vector*) = $\begin{pmatrix} x \\ y \end{pmatrix}$ agar mendapatkan nilai a_i. Nilai x didapatkan menggunakan persamaan (2.22) kernel linear untuk x berikut:

$$\sum_{i=1,j=1}^{n} x_i x_j^T , (i,j = 1,..,n) \qquad (2.22)$$

Hasil perhitungan $x_ix_j^T$ yang telah dilakukan akan masuk ke dalam matriks.

$$x_ix_j^T = \begin{bmatrix} 0.8797 & 0 & 0 & 0 \\ 0 & 0.7052 & 0 & 0 \\ 0 & 0 & 0.7357 & 0 \\ 0 & 0 & 0 & 0.6354 \end{bmatrix}$$

$X_{S1} = x_1x_1^T + x_1x_2^T + x_1x_3^T + x_1x_4^T$ = 0,8797+0+0+0 = 0,8797

$X_{S2} = x_2x_1^T + x_2x_2^T + x_2x_3^T + x_2x_4^T$ = 0+0,7052+0+0 = 0,7052

$X_{S3} = x_3x_1^T + x_3x_2^T + x_3x_3^T + x_3x_4^T$ = 0+0+0,7357+0 = 0,7357

$X_{STesting} = x_4x_1^T + x_4x_2^T + x_4x_3^T + x_4x_4^T$ = = 0+0+0+0.6354 = 0.6354

Sehingga didapatkan untuk nilai *x* pada setiap abstrak pada Tabel 3.25

Tabel 3. 25 Nilai x pada setiap Abstrak

Abstrak	S1	S2	S3	Stesting
X	0,8797	0,7052	0,7357	0.6354

Nilai y didapatkan menggunakan persamaan (2.22) kernel linear untuk y berikut:

$$\sum_{i=1,j=1}^{1} \boldsymbol{y_i y_j^T}, (\boldsymbol{i,j = 1,..,n}) \qquad (2.22)$$

Dengan perhitungan sebagai berikut.

$$y_i y_j^T = \begin{bmatrix} 1 & 1 & -1 & 0 \\ 1 & 1 & -1 & 0 \\ -1 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$Y_{S1} = y_1 y_1^T + y_1 y_2^T + y_1 y_3^T + y_1 y_4^T = 1 + 1 + -1 + 0 = -1$

$Y_{S2} = y_2 y_1^T + y_2 y_2^T + y_2 y_3^T + y_1 y_4^T = 1 + 1 + -1 + 0 = -1$

$Y_{S2} = y_3 y_1^T + y_3 y_2^T + y_3 y_3^T + y_1 y_4^T = -1 + -1 + 1 + 0 = 1$

$Y_{STesting} = y_4 y_1^T + y_4 y_2^T + y_4 y_3^T + y_4 y_4^T = 0 + 0 + 0 + 0 = 0$

sehingga didapatkan untuk nilai *y* pada setiap abstrak pada Tabel 3.26**.**

Tabel 3. 26 Hasil nilai y pada setiap Abstrak

Absrak	S1	S2	S3	STesting
Y	-1	-1	1	0

Seteleh nilai x dan y didapatkan, maka untuk nilai x dan y dari abstrak testing di subtitusikan ke dalam persamaan sebagai berikut.

$$\boldsymbol{\phi}\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix} = \begin{cases} \sqrt{\boldsymbol{x_n^2 + y_n^2}} > \boldsymbol{2\ maka} \begin{bmatrix} \sqrt{\boldsymbol{x_n^2 + y_n^2}} - \boldsymbol{x} + |\boldsymbol{x - y}| \\ \sqrt{\boldsymbol{x_n^2 + y_n^2}} - \boldsymbol{y} + |\boldsymbol{x - y}| \end{bmatrix} \\ \sqrt{\boldsymbol{x_n^2 + y_n^2}} \leq \boldsymbol{2\ maka} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix} \end{cases} \qquad (2.21)$$

Nilai x_n yang didapat dari x_{s3} dan y_n dari y_{s3} yang disubtitusikan kedalam persamaan $\sqrt{x_n^2 + y_n^2} = \sqrt{0.6354^2 + 0^2} = 0.6354 \leq 2$. Karena hasil yang didapatkan $\sqrt{x_n^2 + y_n^2} \leq 2$. Maka, $\phi \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix}$

$\boldsymbol{\phi}$ (STesting)
$\begin{bmatrix} \mathbf{0,6354} \\ \mathbf{0} \end{bmatrix}$

Selanjutnya untuk mendapatkan nilai kelas dari $S_{Testing}$. Lakukan Perhitungan tersebut ke persamaan berikut

$$w^T . \phi(S_{Testing}) = [0.5389 \quad 0.4999] * \begin{bmatrix} 0{,}6354 \\ 0 \end{bmatrix} = 0.3424 < 0.4999$$

Hasil uji data abstrak adalah 0.3424 sehingga $S_{Testing}$ lebih kecil dari pada hasil nilai *b* SVM training yaitu 0.4999 yang berarti $S_{Testing}$ termasuk dalam kelas negatif dan diklasifikasi sebagai kelas E.

3.6 Analisis Kebutuhan Non Fungsional

3.6.1 Analisis Kebutuhan Perangkat Keras

Spesifikasi perangkat keras yang digunakan untuk menjalankan *system* klasifikasi dokumen pada laporan skripsi ini harus memenuhi perangkat keras dengan spesifikasi minimal sebagai berikut.

Tabel 3. 27 Spesifikasi Kebutuhan Perangkat Keras Minimal

No	Perangkat Keras	Spesifikasi
1	Prosesor	Intel Pentium Dual Core
2	Memori	1 GB
3	Hardisk	250 GB

Dengan menganalisis perangkat keras tersebut, maka dapat disimpulkan bahwa perangkat keras yang digunakan telah sesuai untuk menjalankan *system* klasifikasi dokumen pada laporan skripsi.

3.6.2 Analisis Kebutuhan Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan Untuk menjalankan *system* klasifikasi dokumen pada laporan skripsi harus dipenuhi dahulu spesifikasi minimal perangkat lunak yang diantaranya.

Tabel 3. 28 Spesifikasi Kebutuhan Perangkat Lunak Minimal

No	Perangkat Lunak
1	*System* operasi Windows 7 atau windows dengan versi yang lebih baru
2	Web server XAMPP
3	JAVA
4	Netbeans IDE 6

Dengan menganalisis perangkat lunak tersebut, dapat disimpulkan bahwa perangkat lunak yang digunakan telah memenuhi kebutuhan untuk dapat menjalankan *system* klasifikasi dokumen pada laporan skripsi.

3.6.3 Analisis Pengguna

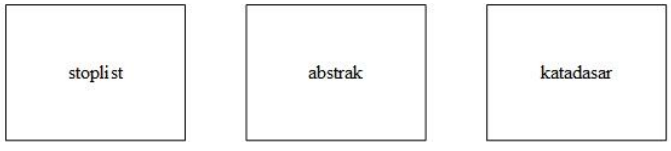
Dalam *system* yang akan dibangun nantinya hanya akan melibatkan 1 entitas, yaitu penguji dengan karateristik sebagai berikut.

Tabel 3. 29 Analisis Pengguna

Pengguna	Penguji
Hak Akses	a. Olah Data latih Laporan Skripsi b. Olah Data uji Laporan Skripsi c. Melakukan Uji Coba

3.6.4 Analisis Basis Data

ERD atau Entity Relationship Diagrama merupakan pemodelan data yang membantu mengorganisasikan data dalam suatu proyek ke dalam entitas-entitas dan menentukan hubungan antar entitas. ERD yang dibangun dalam penilitian ini dapat dilihat pada gambar



Gambar 3. 14 ERD Sistem Klasifikasi

Kamus data dari ERD tersebut dapat dilihat pada tabel 3.30.

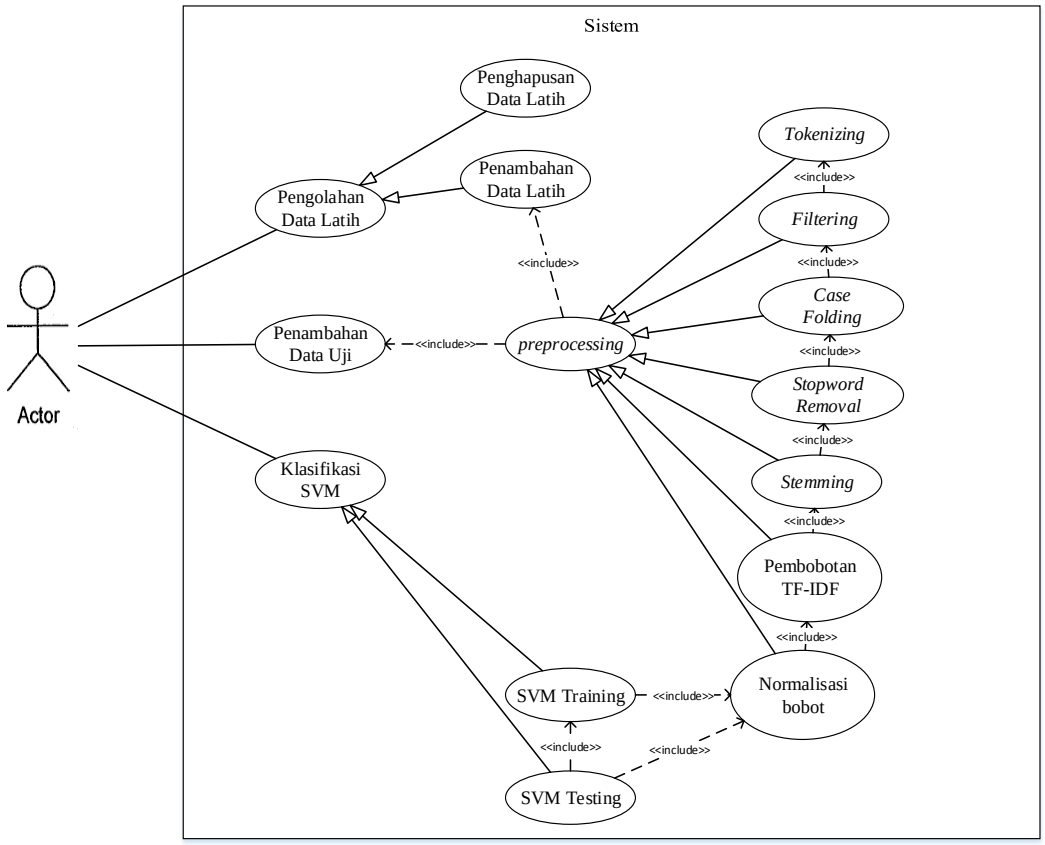
Tabel 3. 30 Analisis Basis Data

Entitas	Atribut
Stoplist	{id_stoplist, kata}
Abstrak	{id_abstrak, kelas, Abstrak, preProsesing}
Katadasar	{id_kata, kata, type }

3.7 Analisis Kebutuhan Fungsional

Fungsi utama dari perangkat lunak yang akan dikembangkan dapat dilihat pada :

3.7.1 *Use Case Diagram*



Gambar 3. 15 *Use Case Diagram*

Berikut merupakan deskripsi aktor yang dijelaskan pada tabel 3.31.

Tabel 3. 31 Deskripsi Aktor

Aktor	Deskripsi
Penguji	Penguji dapat melakukan pemilihan data masukan berupa dokumen, melihat data hasil proses *Preprocessing*, jumlah *term*, pembobotan tf-idf, hasil klasifikasi SVM dan akurasi.

Berikut merupakan deskripsi *use case* yang dijelaskan pada tabel 3.32.

Tabel 3. 32 Deskripsi *Use Case*

No	*Use Case*	Deskripsi
1	Pengolahan Data Latih	Tahap ini digunakan untuk memilih data masukan berupa dokumen dengan format file .xls dan menghapus data latih yang sudah ada.
2	Tambah Data Uji	Tahap ini digunakan untuk memilih data masukan yang akan diuji berupa dokumen dengan format file .xls
3	*Tokenizing*	Tahap ini digunakan untuk memecahkan data kalimat dan memisahkannya menjadi data perkata (token).
4	*Filtering*	Tahap ini digunakan untuk menghilangkan karakter selain "a" sampai "z" dan hanya menerima spasi.
5	*Case folding*	Tahap ini digunakan untuk mengubah data kalimat yang memiliki huruf kapital menjadi huruf kecil.
6	*Stopword Removal*	Tahap ini digunakan untuk menghilangkan kata-kata yang masuk ke dalam daftar *stopword*.
7	*Stemming*	Tahap ini digunakan untuk mengubah kata yang memiliki imbuhan kedalam kata dasarnya.
8	Pembobotan tf-idf	Tahap ini digunakan untuk memberi nilai bobot pada tiap *term* (kata).
9	Normalisasi bobot	Tahap ini digunakan untuk menormalkan nilai bobot pada tiap *term* (kata).
10	SVM Training	Tahap ini digunakan untuk membentuk model fitur dari data latih yang sudah diproses.
11	SVM Testing	Tahap ini digunakan untuk mengklasifikasikan abstrak ke dalam kelas.

3.7.2 Use Case Skenario

Dari use case diagram yang telah dibuat. Berikut penjelasan skenario dari masing-masing use case.

Tabel 3. 33 *Use Case* Skenario Pengolahan Data Latih

Nama *Use Case*	Pengolahan Data Latih
Tujuan	Memasukkan data latih
Deskripsi	Tahap ini digunakan untuk memilih data masukan berupa dokumen dengan format file .xls
Aktor	Penguji
Related Case	-
Kondisi Sukses	Data masukan tersimpan pada database dan dapt dilihat oleh pengguna..

Kondisi Gagal	Data tidak dapat diproses.
Include Case	-
Skenario Utama	
Aksi aktor	Reaksi Sistem
1. Pengguna menekan tombol tambah 2. Pengguna memilih data masukan sesuai dengan format *file* yang dipilih (*Choose File*).	
	3. Sistem melakukan pembacaan file 4. Sistem melakukan preprocessing dan menyimpannya pada database 5. Sistem menampilkan data masukan =.
Skenario Alternatif	
Aksi aktor	Reaksi Sistem
	1. Sistem mengambil data pada database 2. Sistem menampilkan data yang ada pada database

Tabel 3. 34 *Use Case* Skenario *Tokenizing*

Nama *Use Case*	*Tokenizing*
Tujuan	Memisahkan data kalimat menjadi data perkata (term)
Deskripsi	Tahap ini digunakan untuk memecahkan data kalimat dan memisahkannya menjadi data perkata yang akan membentuk sebuat term.
Aktor	Penguji
Related Case	Tambah data latih , tambah data uji
Kondisi Sukses	Data masukan berhasil melalui proses *tokenizing.*
Kondisi Gagal	Data masukan gagal melalui proses *tokenizing.*
Include Case	*Filtering*
Skenario Utama	
Aksi aktor	Reaksi Sistem
	1. Sistem mengambil abstrak yang ada pada database atau data masukan 2. Sistem membaca teks dari awal abstrak 3. Sistem mngambil tiap kata dengan pemisah spasi (“ “) 4. Sistem menyimpan hasilnya dengan bentuk *term* dalam array
Skenario Alternatif	
Aksi aktor	Reaksi Sistem
	1. Sistem akan terus mengambil *term* hingga menemukan spasi (“ ”) atau sampai sistem membaca hingga akhir teks abstrak

Tabel 3. 35 *Use Case* Skenario *Filtering*

Nama *Use Case*	*Filtering*
Tujuan	Menghilangkan kata selain “a” sampai “z”. dan hanya menerima spasi
Deskripsi	Tahap ini digunakan untuk menghilangkan karakter selain “a” sampai “z” dan hanya menerima spasi.
Aktor	Penguji
Related Case	*Tokenizing*

Kondisi Sukses	Data masukan berhasil melalui proses *filtering.*
Kondisi Gagal	Data masukan gagal melalui proses *filtering.*
Include Case	*Case Folding*
Skenario Utama	
Aksi aktor	Reaksi Sistem
	1. Sistem membaca hasil *tokenizing* 2. Sistem menghilangkan setiap *term* hasil tokenizing selain "a" sampai "z". dan hanya menerima spasi. 3. Sistem menyimpan hasilnya dengan bentuk *term* dalam array
Skenario Alternatif	
Aksi aktor	Reaksi Sistem
	1. Sistem akan mengabaikan *term* dengan tetap menyimpannya.

Tabel 3. 36 *Use Case* Skenario *Case Folding*

Nama *Use Case*	*Case Folding*
Tujuan	Mengubah huruf kapital menjadi huruf kecil.
Deskripsi	Tahap ini digunakan penguji untuk mengubah data kalimat yang memiliki huruf kapital menjadi huruf kecil.
Aktor	Penguji
Related Case	*Filtering*
Kondisi Sukses	Data masukan berhasil melalui proses *case folding.*
Kondisi Gagal	Data masukan gagal melalui proses *case folding.*
Include Case	*stopword removal*
Skenario Utama	
Aksi aktor	Reaksi Sistem
	1. Sistem membaca hasil *filtering* 2. Sistem melakukan *string lowercase* pada setiap *term.* 3. Sistem menyimpan hasilnya dengan bentuk *term* dalam array
Skenario Alternatif	
Aksi aktor	Reaksi Sistem
	1. Sistem akan mengabaikan *term* dengan tetap menyimpannya.

Tabel 3. 37 *Use Case* Skenario *Stopword Removal*

Nama *Use Case*	*Stopword Removal*
Tujuan	Menghilangkan kata yang terdapat dalam kamus stopword.
Deskripsi	Tahap ini digunakan untuk menghilangkan kata-kata yang masuk ke dalam daftar *stoplist.*
Aktor	Penguji
Related Case	*Case Folding*
Kondisi Sukses	Data masukan berhasil melalui proses *stopword removal.*
Kondisi Gagal	Data masukan gagal melalui proses *stopword removal.*
Include Case	*Stemming*
Skenario Utama	
Aksi aktor	Reaksi Sistem

	1. Sistem membaca hasil *casefolding* 2. Sistem melakukan pengecekan setiap *term*, dengan membandingkannya dengan stoplist. 3. Sistem menghapus *term* yang sama dengan stoplist. 4. Sistem menyimpan hasilnya dengan bentuk *term* dalam array
Skenario Alternatif	
Aksi aktor	Reaksi Sistem
	1. Sistem akan mengabaikan *term* dengan tetap menyimpannya.

Tabel 3. 38 *Use Case* Skenario *Stemming*

Nama *Use Case*	*Stemming*
Tujuan	Mengubah kata berimbuhan menjadi kata dasar.
Deskripsi	Tahap ini digunakan penguji untuk mengubah kata yang memiliki imbuhan kedalam kata dasarnya.
Aktor	Penguji
Related Case	*stopword removal*
Kondisi Sukses	Data masukan berhasil melalui proses *stemming.*
Kondisi Gagal	Data masukan gagal melalui proses *stemming.*
Include Case	Pembobotan TF-IDF
Skenario Utama	
Aksi aktor	Reaksi Sistem
	1. Sistem membaca hasil *stopword removal* 2. Sistem melakukan pengecekan pada tiap *term* yang masih mengandung imbuhan 3. Sistem melakukan penghapusan setiap imbuhan yang ditemukan. 4. Sistem menyimpan hasilnya dengan bentuk *text* yang tiap *term* nya dipisahkan dengan spasi (“ “)
Skenario Alternatif	
Aksi aktor	Reaksi Sistem
	1. Sistem akan mengabaikan *term* dengan tetap menyimpannya.

Tabel 3. 39 *Use Case* Skenario Pembobotan TF-IDF

Nama *Use Case*	Pembobotan TF-IDF
Tujuan	Memberi nilai bobot pada tiap kata.
Deskripsi	Fungsionalitas ini digunakan penguji untuk memberi nilai bobot pada tiap *term* (kata).
Aktor	Penguji
Related Case	*Stemming*
Kondisi Sukses	Data hasil *stemming* berhasil diberi nilai bobot
Kondisi Gagal	Data hasil *stemming* gagal diberi nilai bobot
Include Case	Normalisasi bobot
Skenario Utama	
Aksi aktor	Reaksi Sistem
	1. Sistem membaca hasil pembobotan TF-IDF 2. Sistem melakukan perhitungan jumlah dari tiap *term* (*TF).*

	3. Sistem melakukan perhitungan terhadap suatu *term* terdapat di berapa dokumen. (*DF*) 4. Sistem melakukan pembobotan TF_IDF
Skenario Alternatif	
Aksi aktor	Reaksi Sistem
	1. Sistem akan mengabaikan *term* dengan tetap membobotkannya.

Tabel 3. 40 *Use Case* Skenario Normalisasi Bobot

Nama *Use Case*	Normalisasi bobot
Tujuan	Memberi nilai normalisasi bobot pada tiap kata.
Deskripsi	Fungsionalitas ini digunakan penguji untuk memberi nilai normalisasi bobot pada tiap *term* (kata).
Aktor	Penguji
Related Case	Pembobotan TF-IDF
Kondisi Sukses	Data hasil pembobotan TF-IDF berhasil dinormalisasi
Kondisi Gagal	Data hasil *stemming* gagal dinormalisasi
Include Case	SVM *training*, SVM *testing*
Skenario Utama	
Aksi aktor	Reaksi Sistem
	1. Sistem membaca hasil pembobotan TF-IDF 2. Sistem melakukan normalisasi bobot 3. Sistem menampilkan hasil normalisasi bobot
Skenario Alternatif	
Aksi aktor	Reaksi Sistem
	2. Sistem akan mengabaikan *term* dengan tetap membobotkannya.

Tabel 3. 41 *Use Case* Skenario SVM Training

Nama *Use Case*	SVM Training
Tujuan	Melakukan training pada sistem
Deskripsi	Fungsionalitas ini digunakan penguji untuk memetakan data abstrak dan mendapatkan hyperplane
Aktor	Penguji
Related Case	Normalisasi bobot
Kondisi Sukses	Data Abstrak berhasil dipetakan dengan mendapatkan hyperplane
Kondisi Gagal	Data Abstrak gagal dipetakan dan tidak terdapat hyperplane
Include Case	-
Skenario Utama	
Aksi aktor	Reaksi Sistem
	1. Sistem membaca data pada database atau data masukan yang telah melalui *Preprocessing* 2. Sistem mengambil normalisasi bobot dari *term* untuk melakukan perhitungan 3. Sistem melakukan pemodelan SVM 4. Sistem membentuk hyperplane sebagai model training.
Skenario Alternatif	

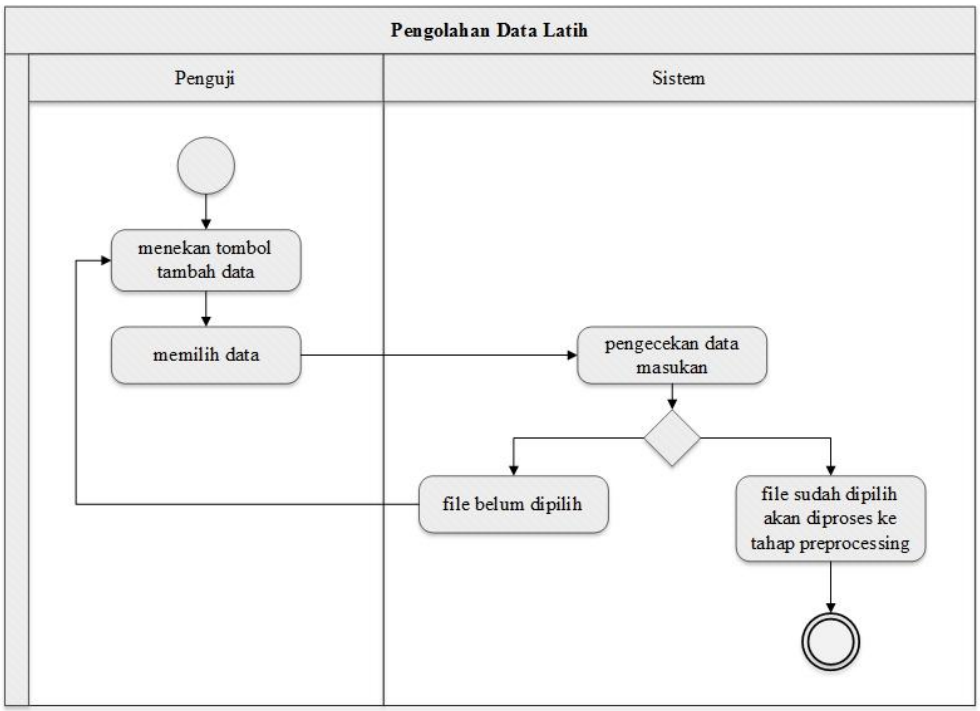
Aksi aktor	Reaksi Sistem
	1. Gagal mendapatkan hyperplane

Tabel 3. 42 *Use Case* Skenario SVM Testing

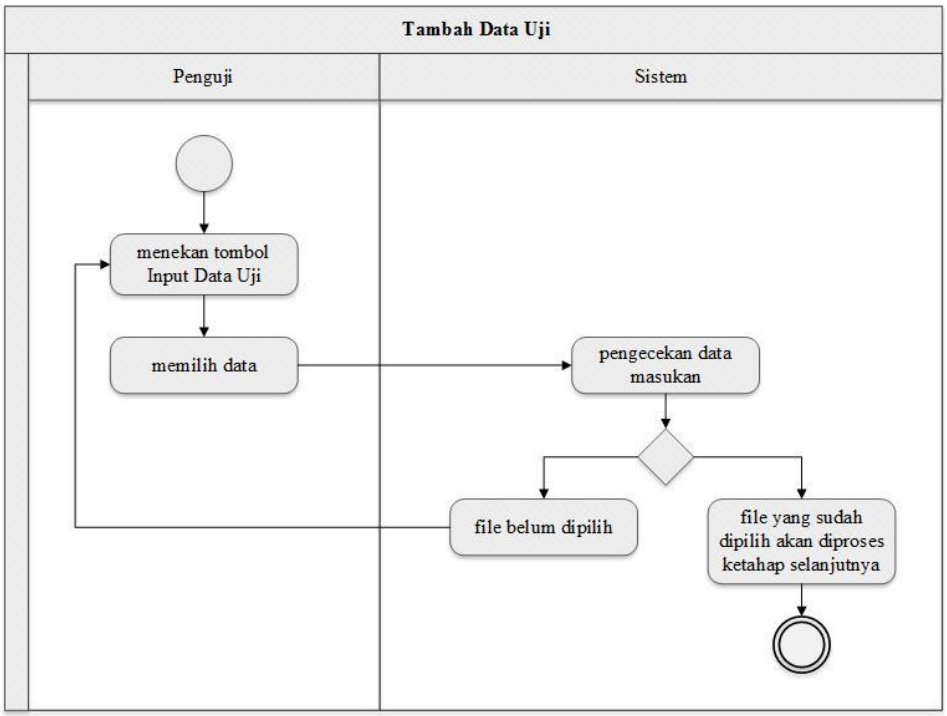
Nama *Use Case*	SVM Testing
Tujuan	Menggunakan hyperplane untuk menguji data abstrak
Deskripsi	Fungsionalitas ini digunakan penguji untuk menguji data abstrak menggunakan hyperplane dari SVM Training
Aktor	Penguji
Related Case	Normalisasi bobot, SVM *training*
Kondisi Sukses	Data Uji berhasil diklasiikasi
Kondisi Gagal	Data Uji gagal diklasifikasi
Include Case	-
Skenario Utama	
Aksi aktor	Reaksi Sistem
	1. Sistem membaca data masuan untuk pengujian 2. Sistem melakukan *Preprocessing* 3. Sistem mengambil normaliasi bobot dari *term* untuk melakukan perhitungan 4. Sistem mengklasifikasikan data uji dengan menggunakan model SVM Training 5. Sistem menampilkan hasil klasifikasi
Skenario Alternatif	
Aksi aktor	Reaksi Sistem
	1. Sistem mengklasifikaasi dengan label -1

3.7.3 *Activity Diagram*

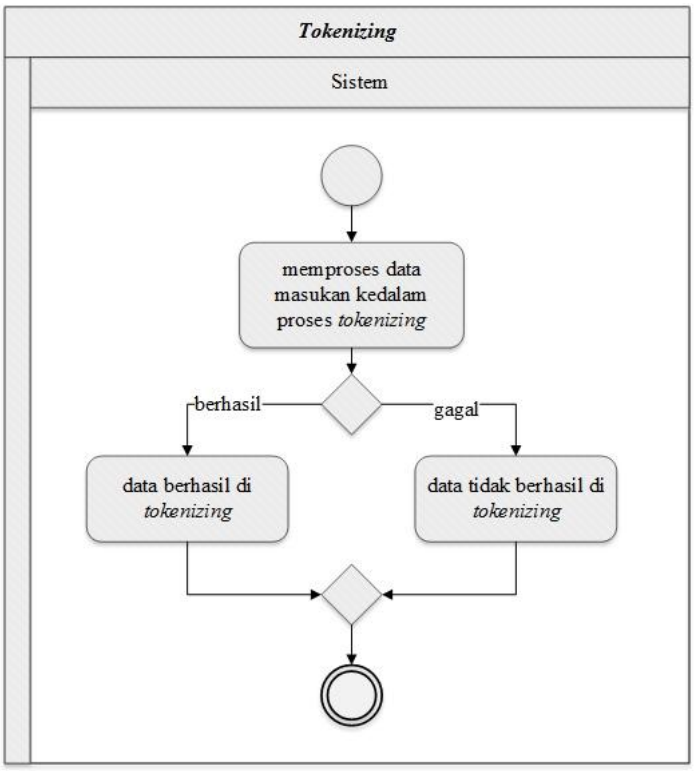
Adapun *activity diagram* dari masing-masing *use case.*



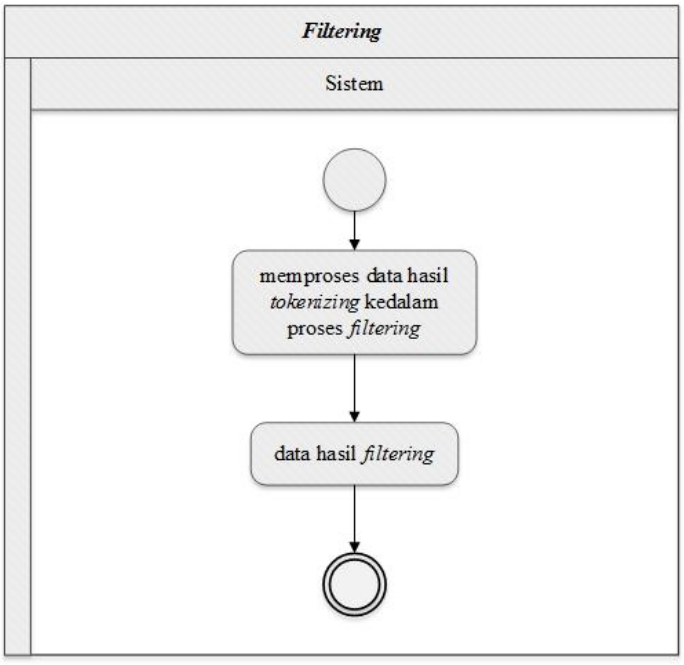
Gambar 3. 16 *Activity Diagram* Pengolahan Data Latih



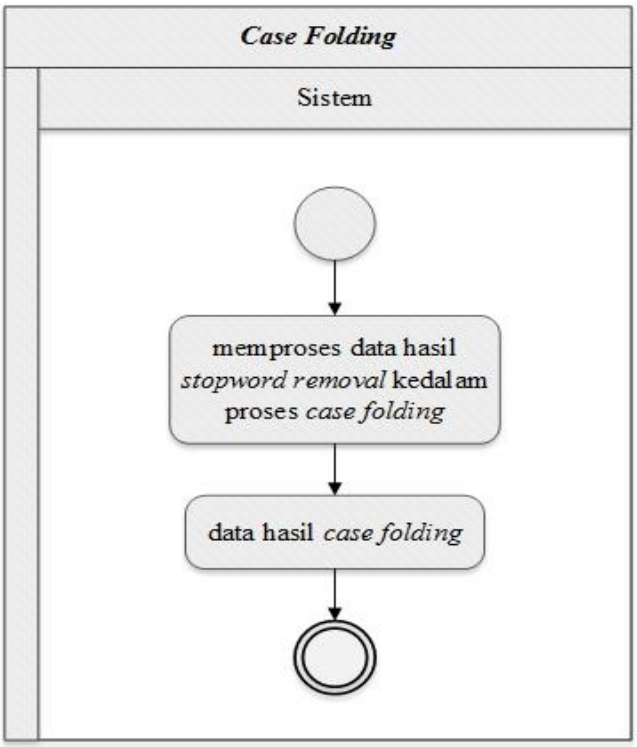
Gambar 3. 17 *Activity Diagram* Tambah Data Uji



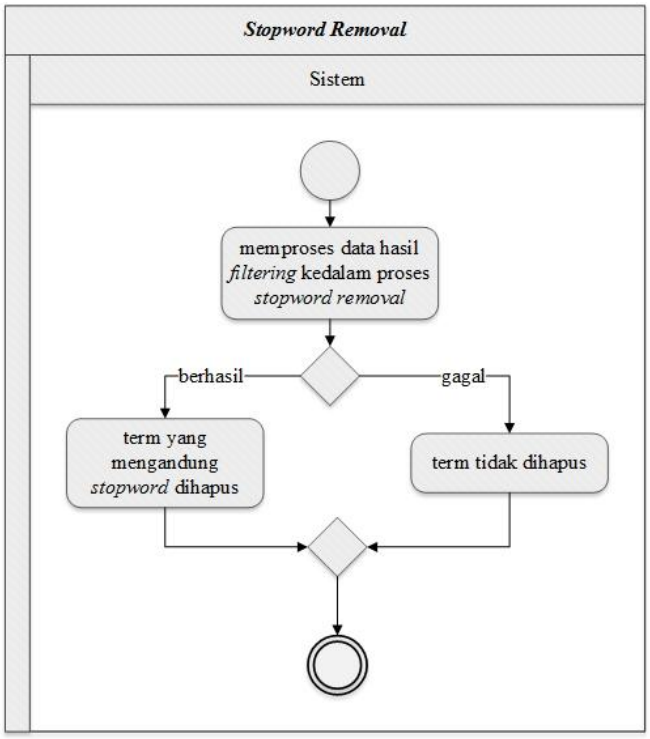
Gambar 3. 18 ***Activity Diagram Tokenizing***



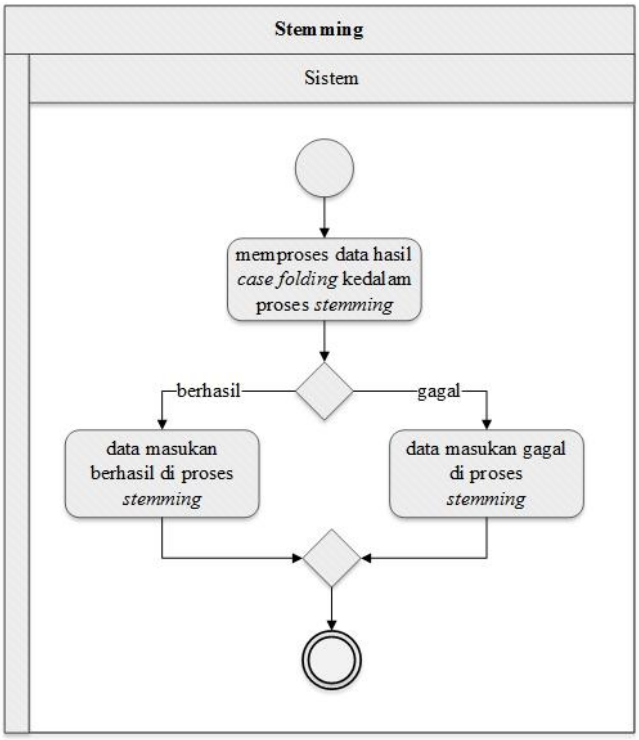
Gambar 3. 19 ***Activity Diagram Filtering***



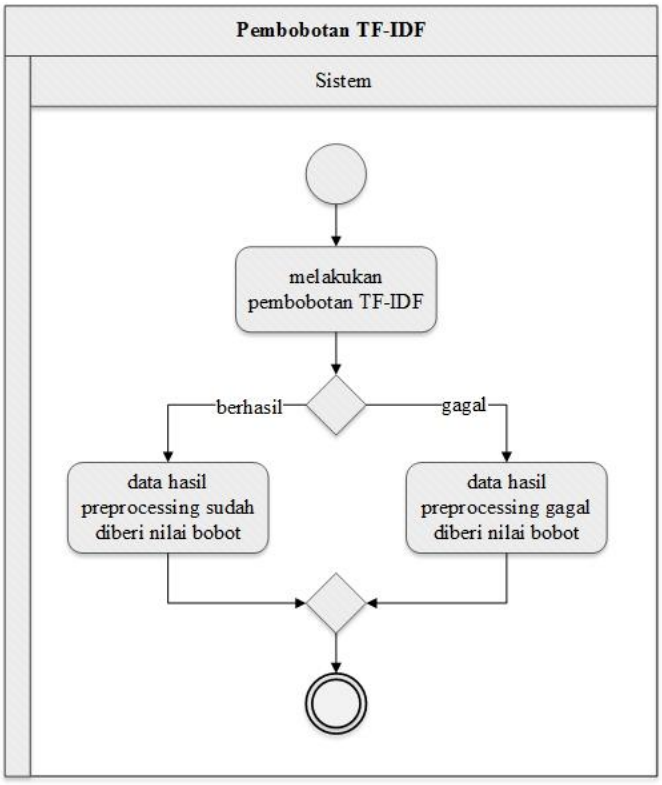
Gambar 3. 20 ***Activity Diagram Case Folding***



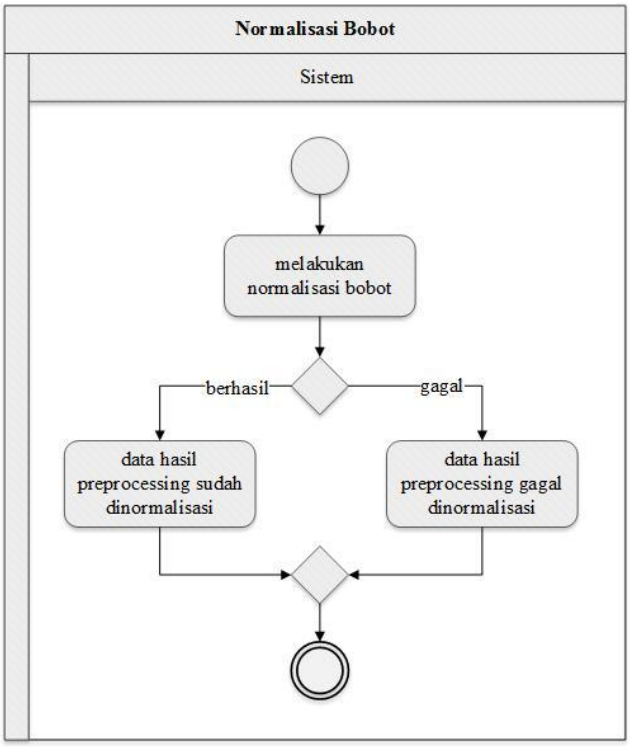
Gambar 3. 21 ***Activity Diagram Stopword Removal***



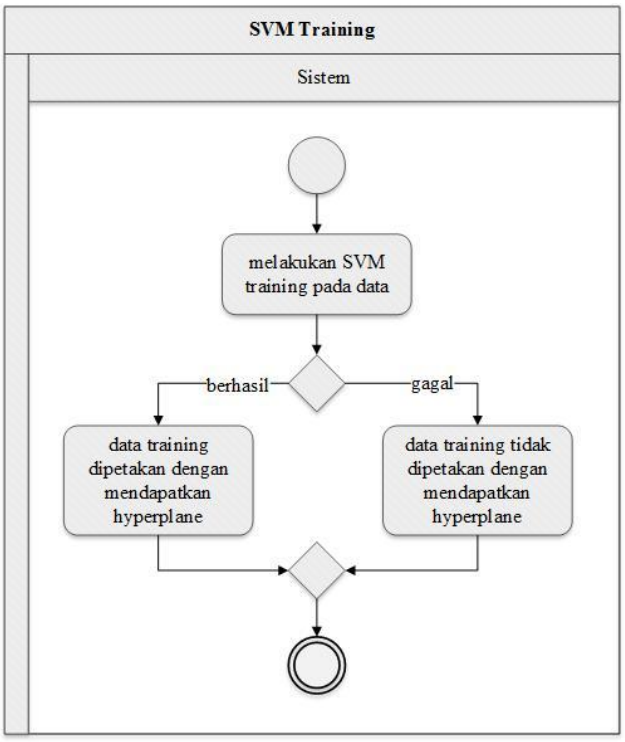
Gambar 3. 22 *Activity Diagram* Stemming



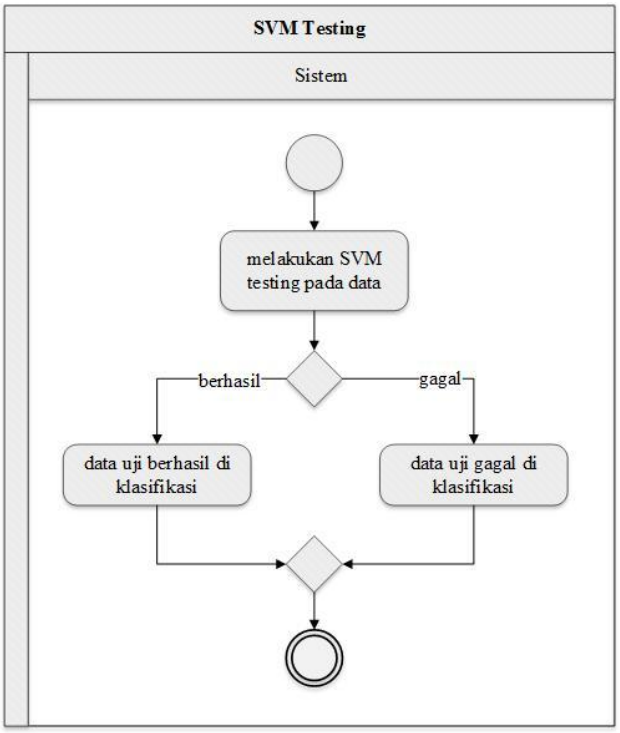
Gambar 3. 23 *Activity Diagram* Pembobotan TF-IDF



Gambar 3. 24 *Activity Diagram* Normalisasi Bobot



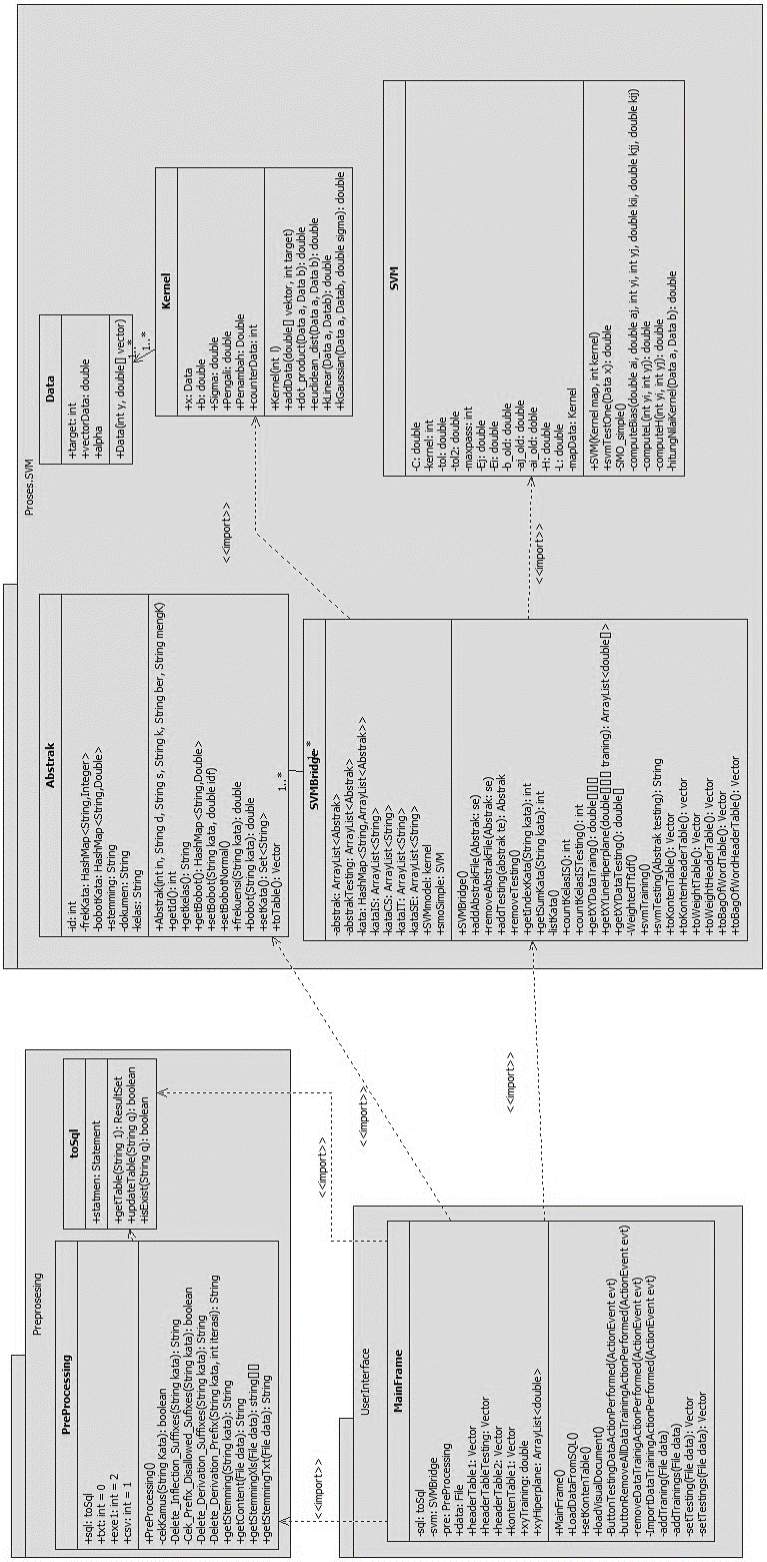
Gambar 3. 25 *Activity Diagram* SVM Training



Gambar 3. 26 *Activity Diagram* SVM Testing

3.7.4 *Class Diagram*

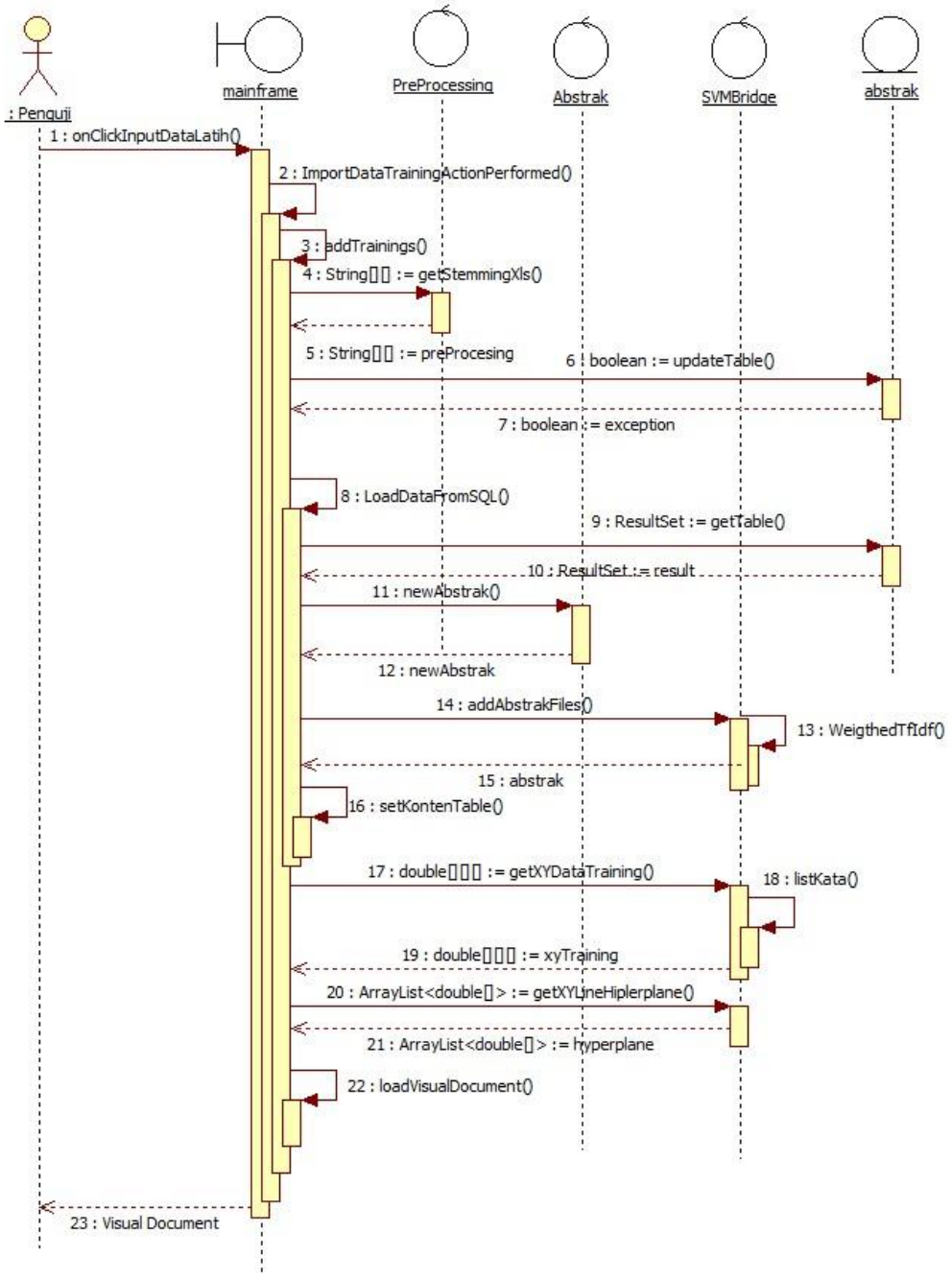
Class diagram adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari desain berorientasi objek. Berikut adalah bentuk *class diagram* pada Gambar 3.26.



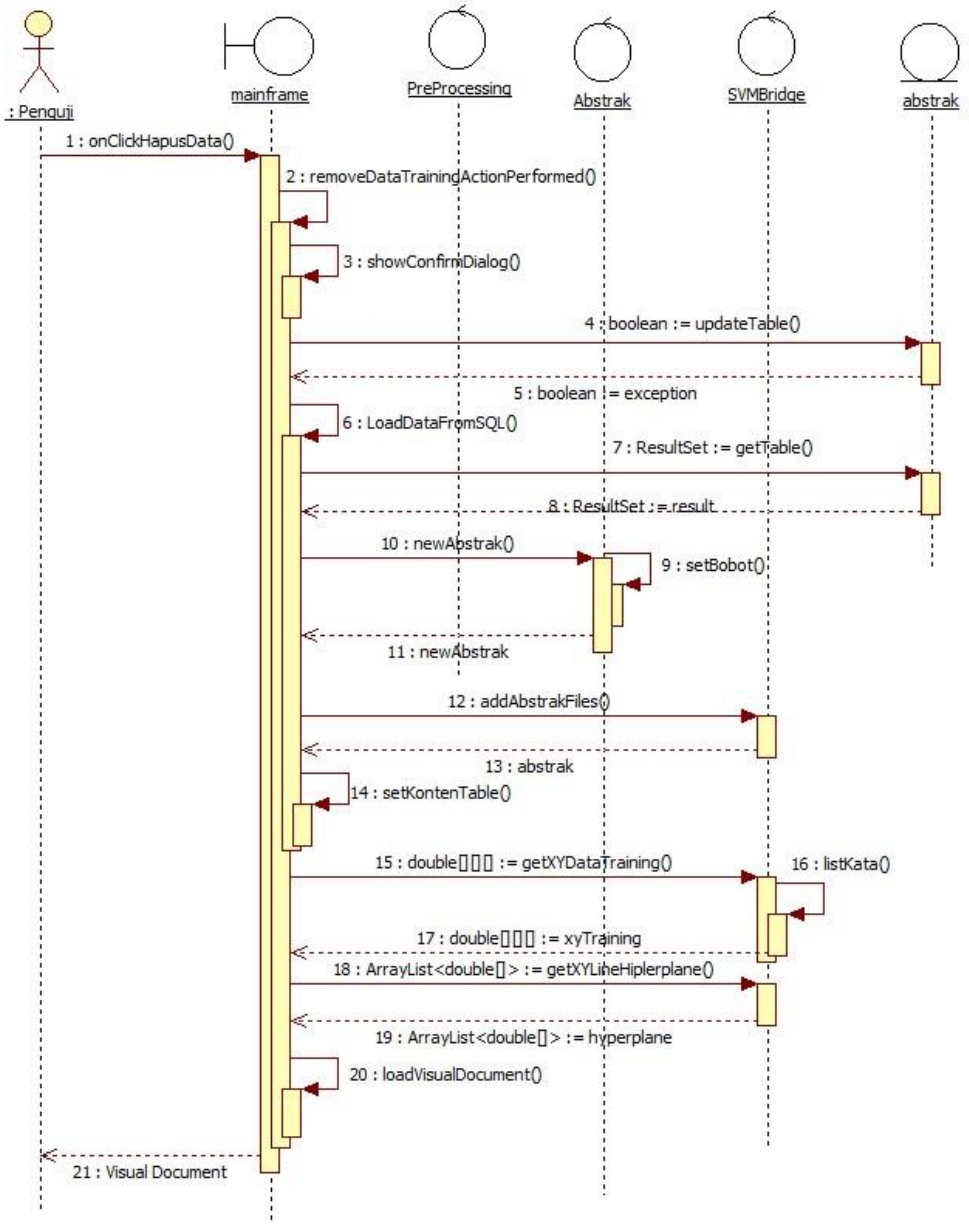
Gambar 3. 27 *Class Diagram*

3.7.5 *Sequence Diagram*

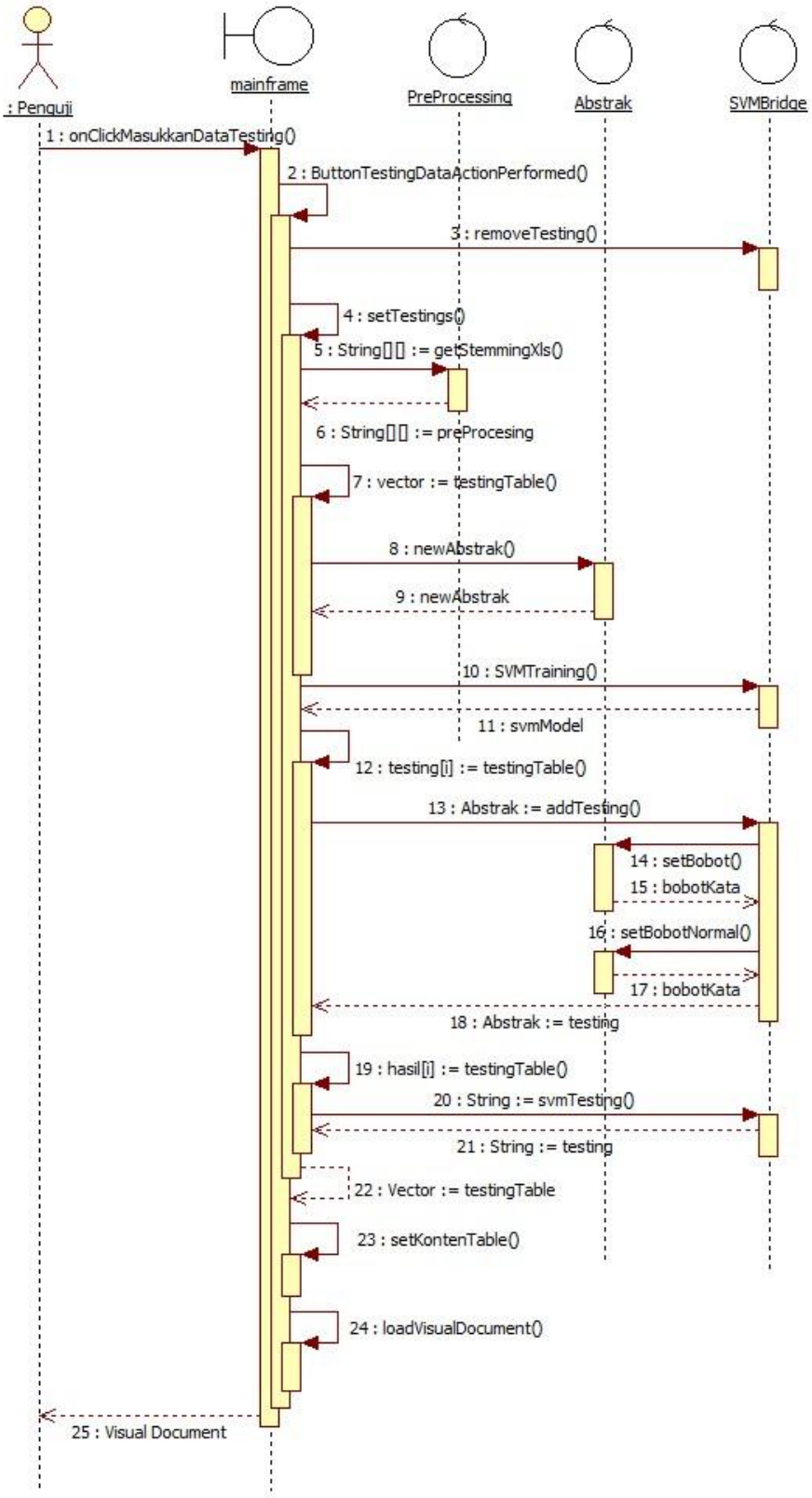
Berikut ini adalah *sequence diagram* yang terdapat pada pembangunan aplikasi yang dibangun.



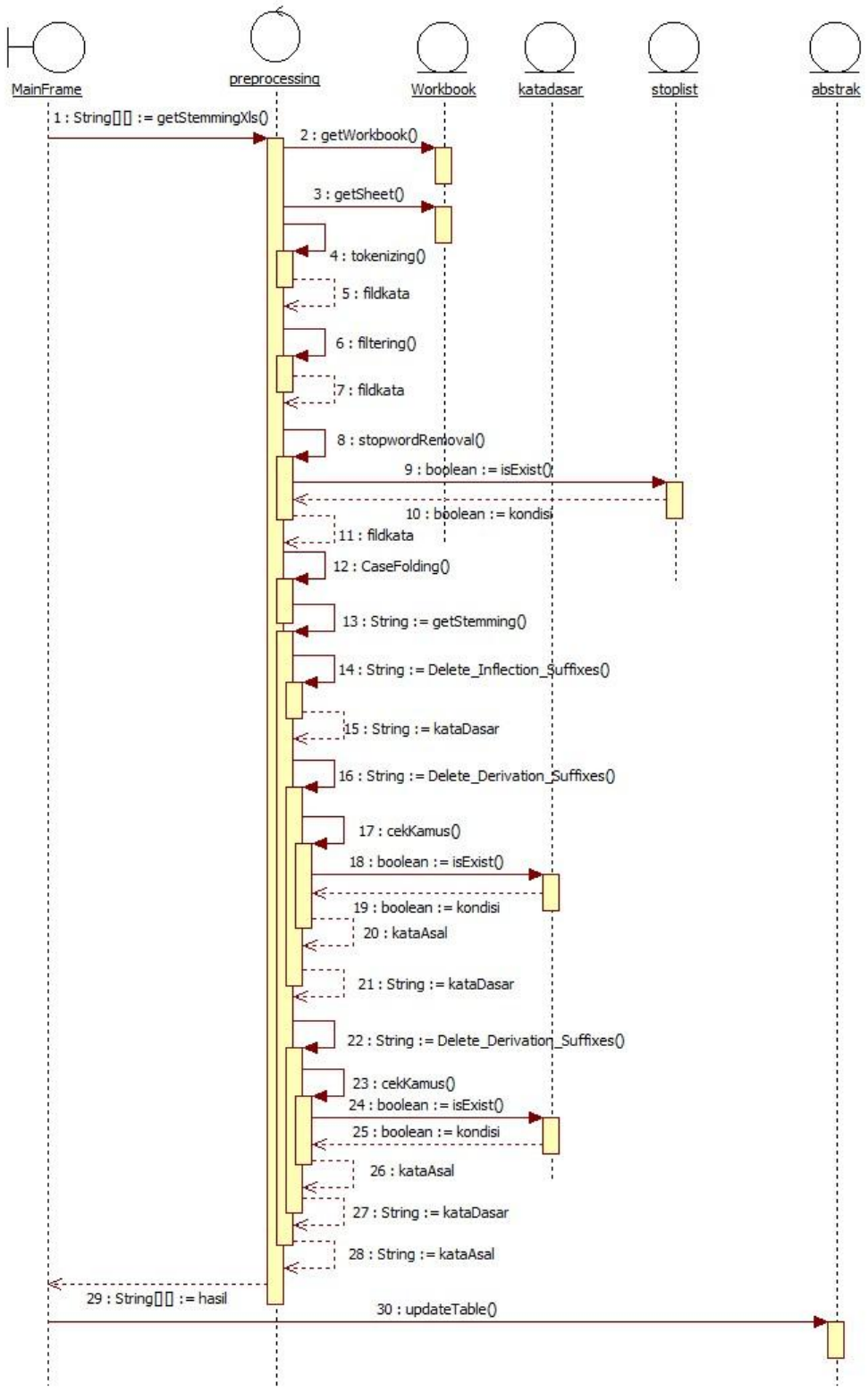
Gambar 3. 28 *Sequence Diagram* Tambah Data Latih



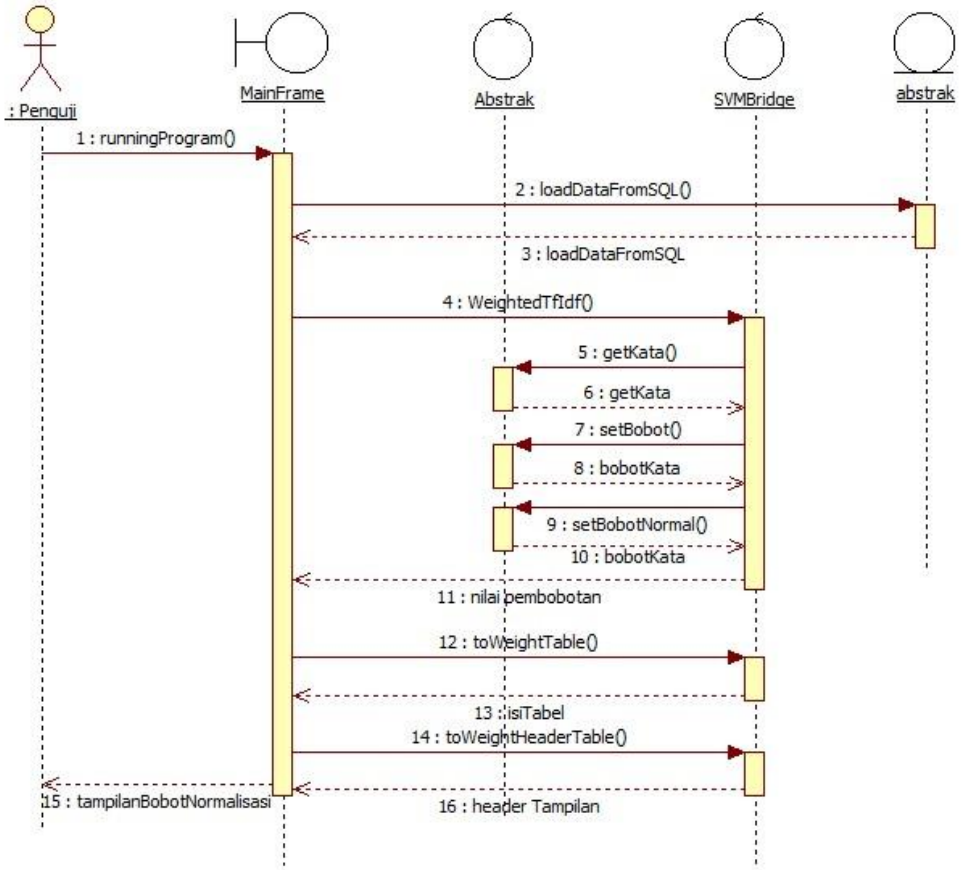
Gambar 3. 29 *Sequence Diagram* Hapus Data Latih



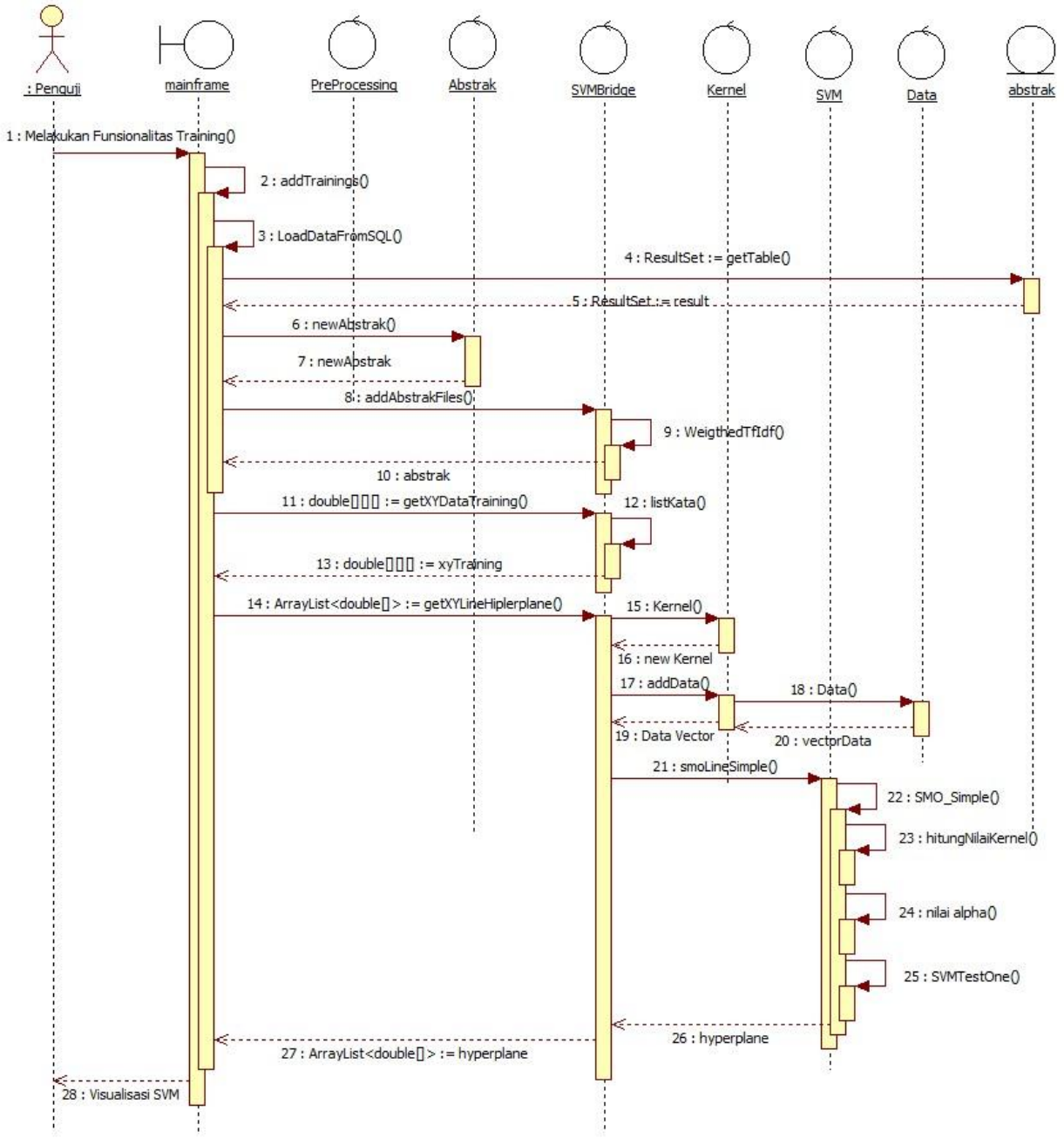
Gambar 3. 30 *Sequence Diagram* Tambah Data Uji



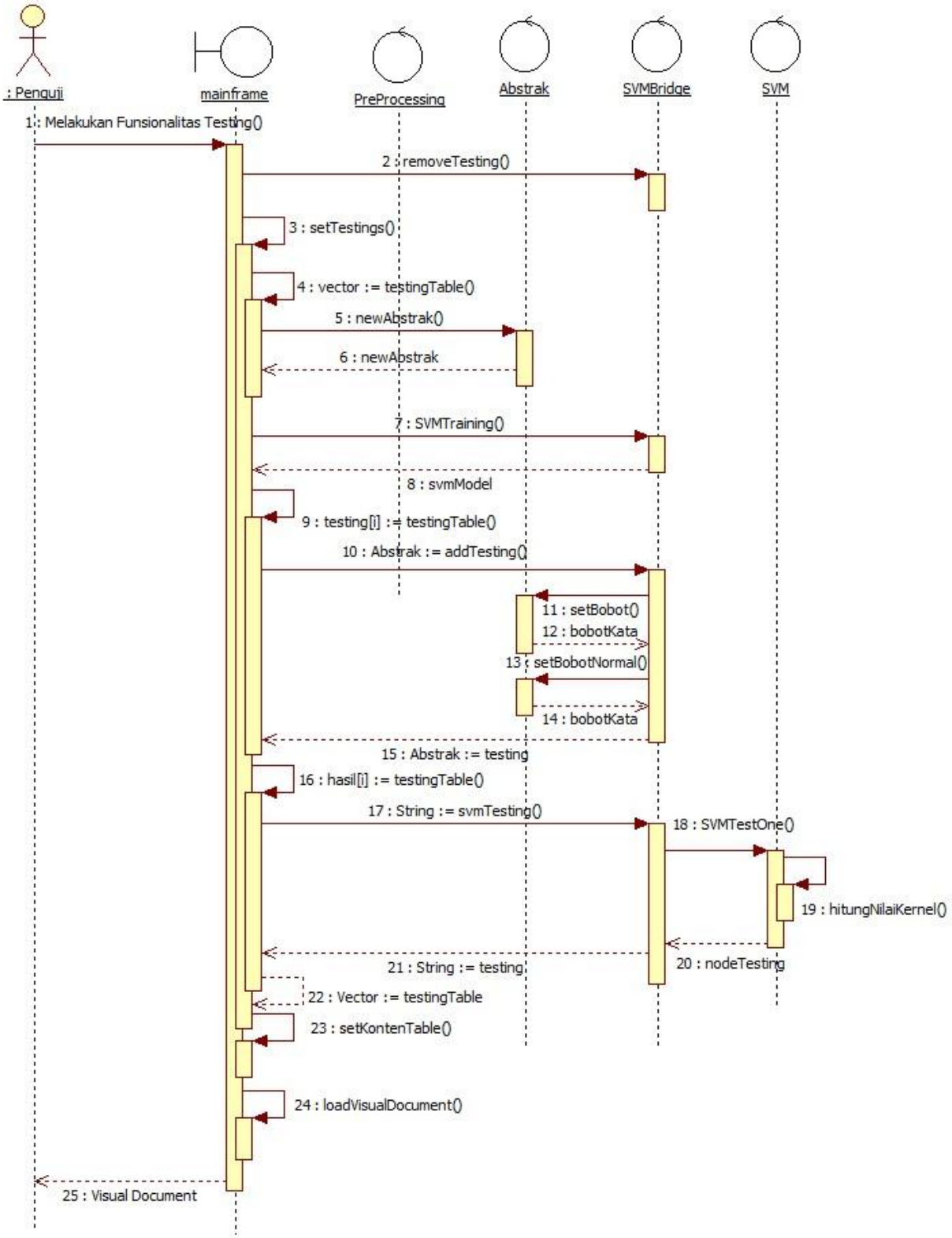
Gambar 3. 31 ***Sequence Diagram Tokenizing, Filtering, Case Folding, Stopword Removal, Stemming***



Gambar 3. 32 *Sequence Diagram* TF-IDF



Gambar 3. 33 *Sequence Diagram* SVM Training



Gambar 3. 34 *Sequence Diagram* SVM Testing

3.8 Perancangan Sistem

Dari hasil analisis yang telah dilakukan, maka dibuatlah perancangan sistem. Perancangan sistem adalah penggambaran serta perencanaan dari analisis yang telah dilakukan. Perencanaan sistem yang dibangun terdiri dari basis data dan perancangan antarmuka.

3.8.1 Struktur Tabel

Struktur tabel dari sistem yang dibangun adalah sebagai berikut :

1. Tabel Abstrak
   Tabel yang berfungsi untuk menyimpan data abstrak.

Tabel 3. 43 Tabel Abstrak

Struktur Tabel				
Nama Field	Tipe	Panjang	Kunci	Keterangan
No	Int	11	*Primary Key*	*Not Null, Auto Increament*
Kelas	varchar	12		
Abstrak	Text			
preProsesing	Text	3		

2. Tabel Kata Dasar
   Tabel yang berfungsi untuk menyimpan data kata dasar yang akan digunakan pada saat proses stemming.

Tabel 3. 44 Tabel Kata Dasar

Struktur Tabel				
Nama Field	Tipe	Panjang	Kunci	Keterangan
No	Int	11	*Primary Key*	*Not Null, Auto Increament*
Kata	varchar	50		
Type	varchar	3		

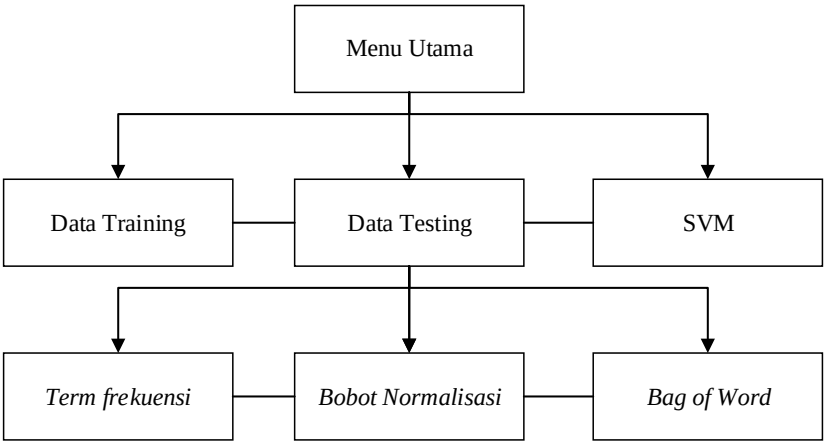
3. Tabel *Stoplist*
   Tabel yang berfungsi untuk menyimpan data stoplist yang akan digunakan pada saat proses *stopword removal.*

Tabel 3. 45 Tabel *Stoplist*

Struktur Tabel				
Nama Field	Tipe	Panjang	Kunci	Keterangan
No	Int	11	*Primary Key*	*Not Null, Auto Increament*
Kata	varchar	50		

3.8.2 Perancangan Struktur Menu

Fungsi-fungsi yang dirancang pada tahap perancangan ini dibagi kedalam beberapa menu yang bertujuan untuk memudahkan pengoperasian program. Struktur menu ini dapat dilihat pada gambar 3.34.



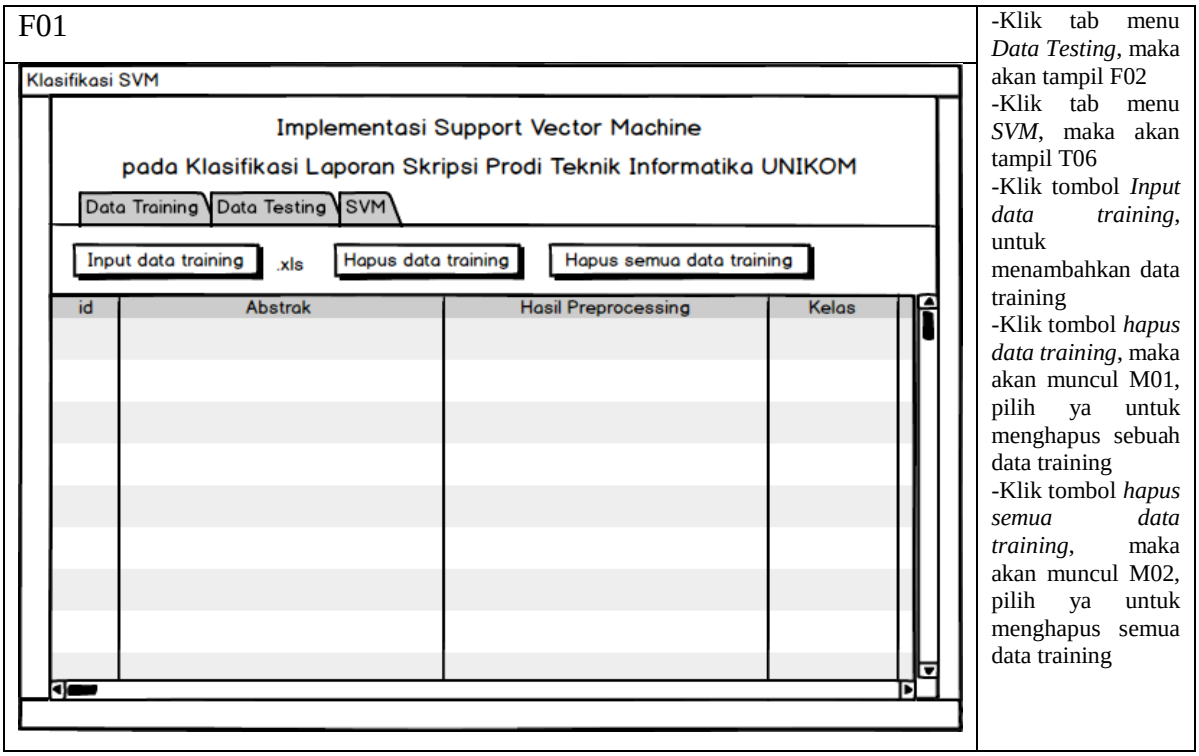
Gambar 3. 35 Struktur Menu

3.8.3 Perancangan Antarmuka

Perancangan antarmuka merupakan rancangan untuk menentukan tampilan dari sistem yang akan dibuat. Berikut adalah perancangan pada aplikasi yang akan dibangun.

1. F01 – *Form* Halaman Utama dan Pngolahan data latih

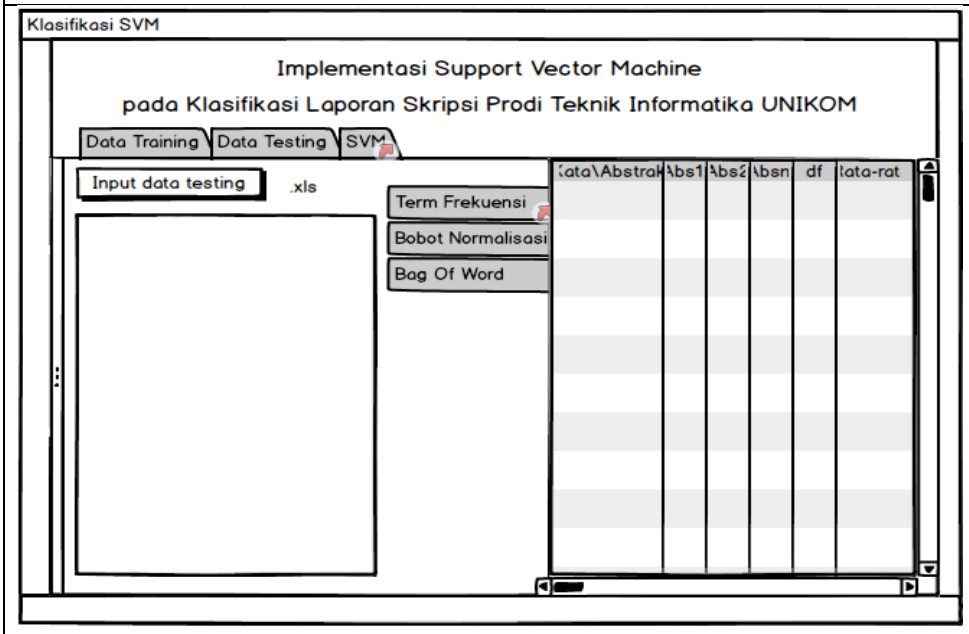
   Pada perancangan F01 adalah perancangan halaman utama pada saat pertama kali menjalankan aplikasi. Pada perancangan halaman ini juga ditampilkan Input Data Uji, namun button data uji tersebut bisa dipakai setelah menginputkan data latih.



Gambar 3. 36 Perancangan Antarmuka Halaman Utama dan Input Data Uji

2. F02 – *Form* Input data uji

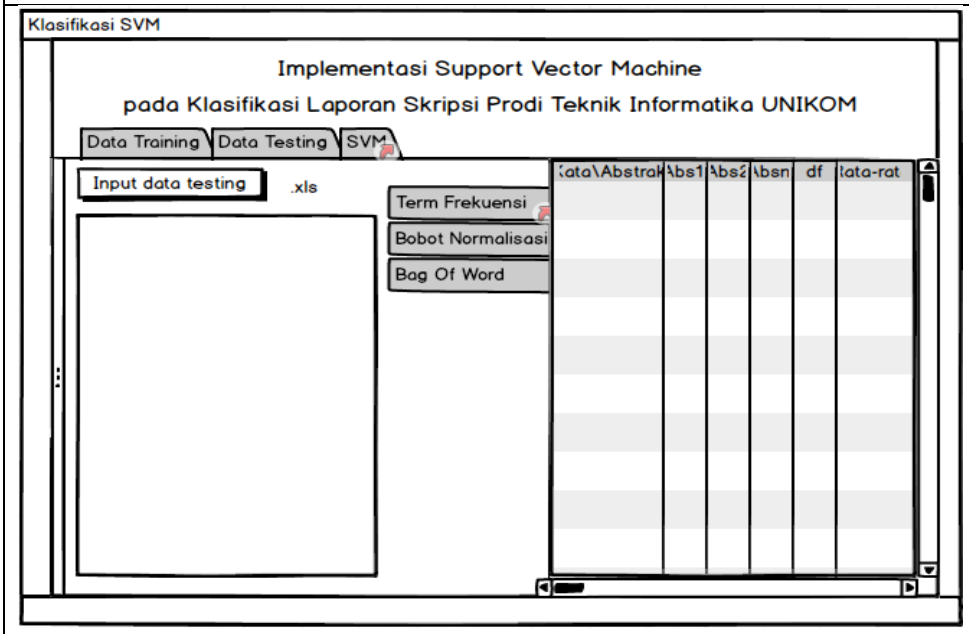
Pada perancangan F02 adalah perancangan untuk pengolahan data latih, yaitu tambah/input data , hapus data, hapus semua data. Setelah data diinputkan maka tabel id, isi dokumen dan yang lainnya akan terisi.

F02	
	-Klik tab menu *Data Training*, maka akan tampil F01 -Klik tab menu *SVM*, maka akan tampil T06 -Klik tombol *Input data testing*, untuk memasukkan data testing

Gambar 3. 37 Perancangan Antarmuka Pengolahan Data Latih

3. T03 – Tabmenu *Term Frekuensi*

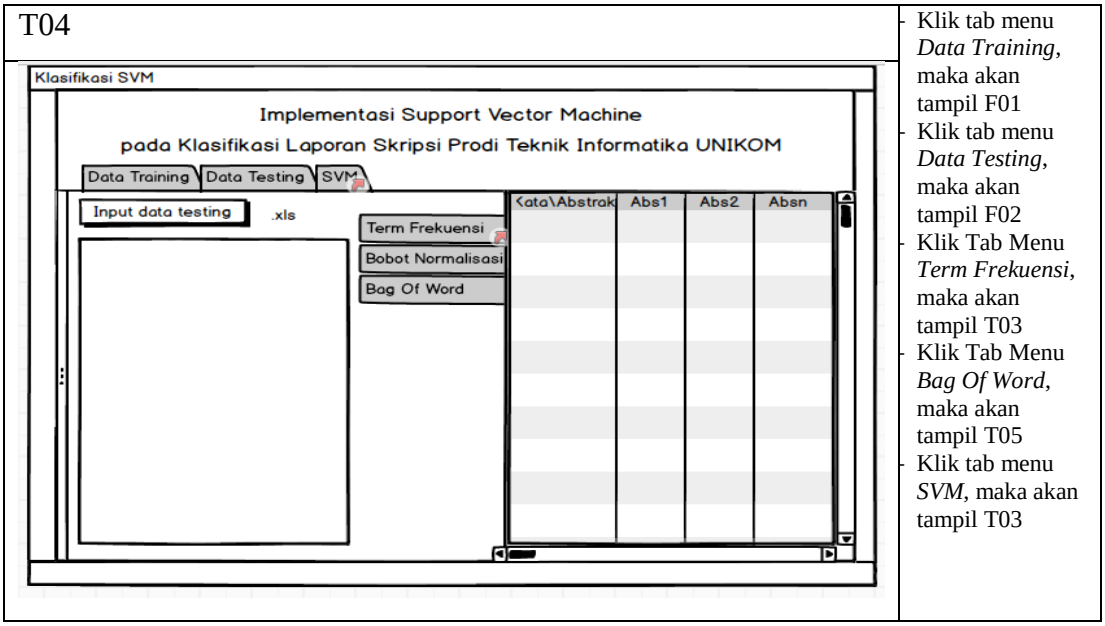
Pada perancangan T03 adalah perancangan untuk menampilkan hasil *term* yang sudah dihitung frekuensi kemunculannya / nilai df nya.

T03	
	- Klik tab menu *Data Training*, maka akan tampil F01 - Klik tab menu *Data Testing*, maka akan tampil F02 - Klik tab menu *SVM*, maka akan tampil T06 - Klik Tab Menu *Bobot Normalisasi*, maka akan tampil T04 - Klik Tab Menu *Bag Of Word*, maka akan tampil T05

Gambar 3. 38 Perancangan Antarmuka *Term* Frekuensi

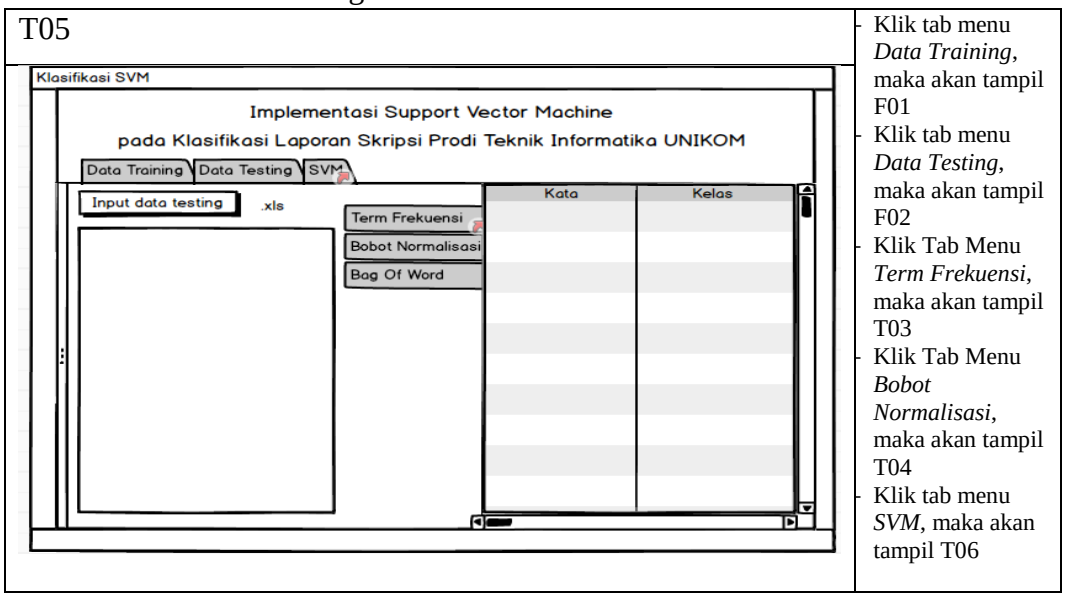
4. T04 – Tabmenu Bobot Normalisasi

Pada perancangan T04 adalah perancangan unutk menampilkan bobot Tf-Idf .



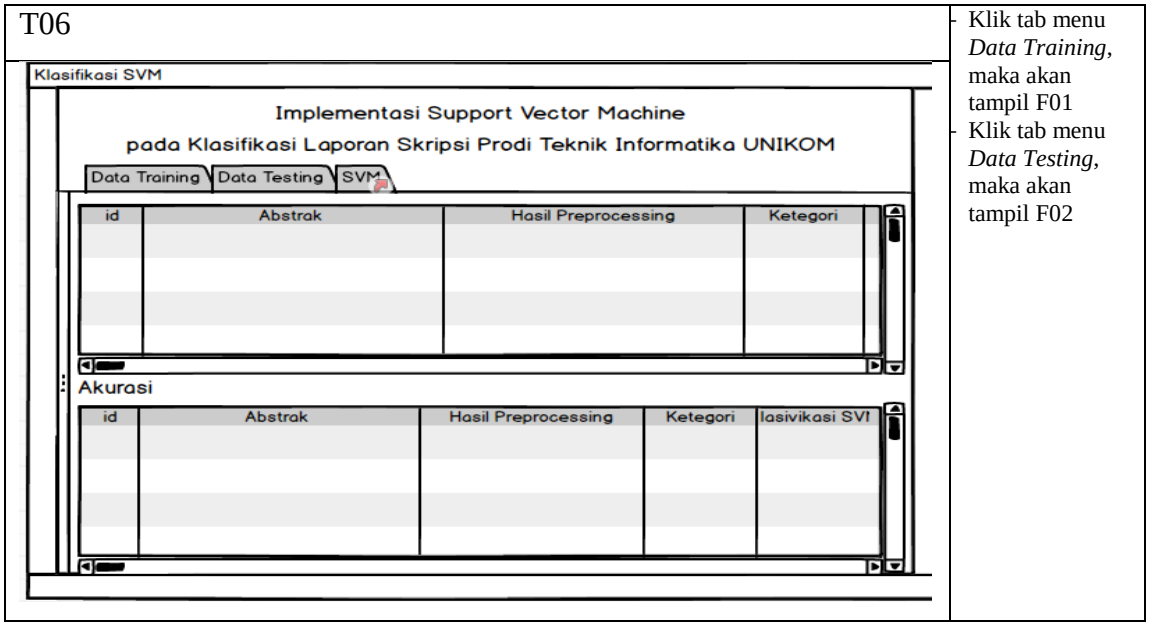
Gambar 3. 39 Perancangan Antarmuka Bobot Normalisasi

5. T05 – Tabmenu Bag of Word



Gambar 3. 40 Perancangan Antarmuka *Bag Of Word*

6. T06 – Tabmenu SVM



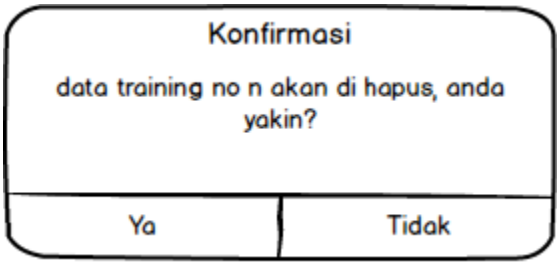
Gambar 3. 41 Perancangan Antarmuka SVM

3.8.4 Perancangan Pesan

Tabel 3. 46 Perancangan Pesan

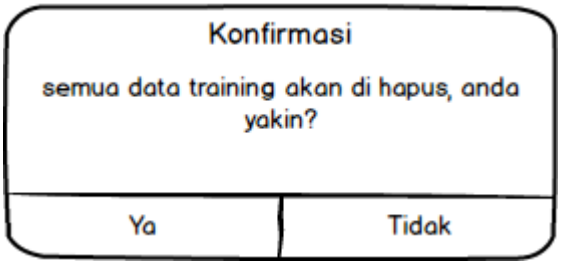
Kode Pesan	Keterangan Pesan
M01	Hapus data training
M02	Hapus semua data training

M01



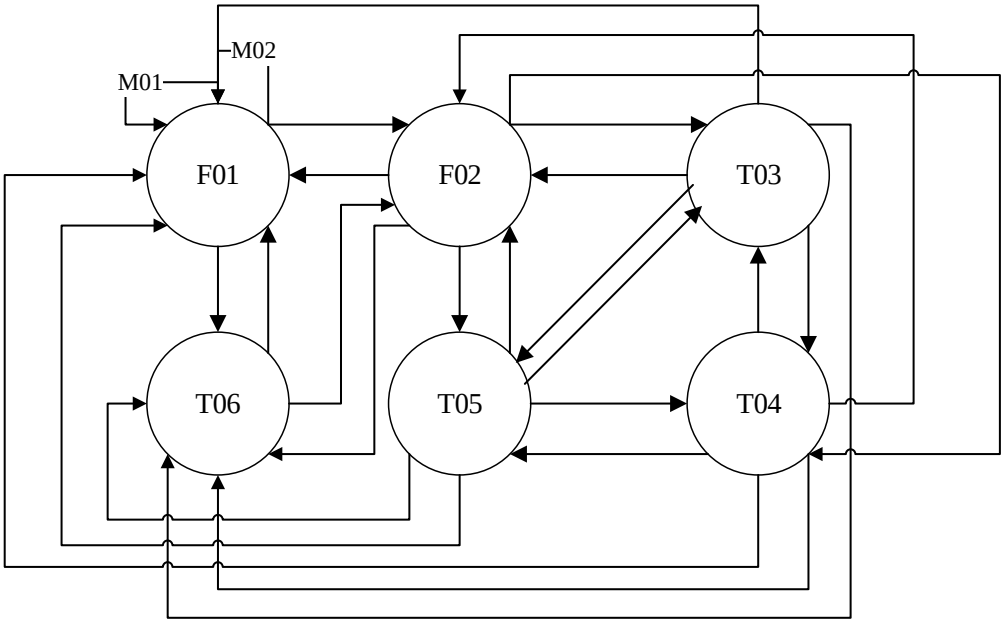
Gambar 3. 42 Perancangan Pesan M01

M02



Gambar 3. 43 Pesan Perancangan Pesan M02

3.8.5 Jaringan Semantik

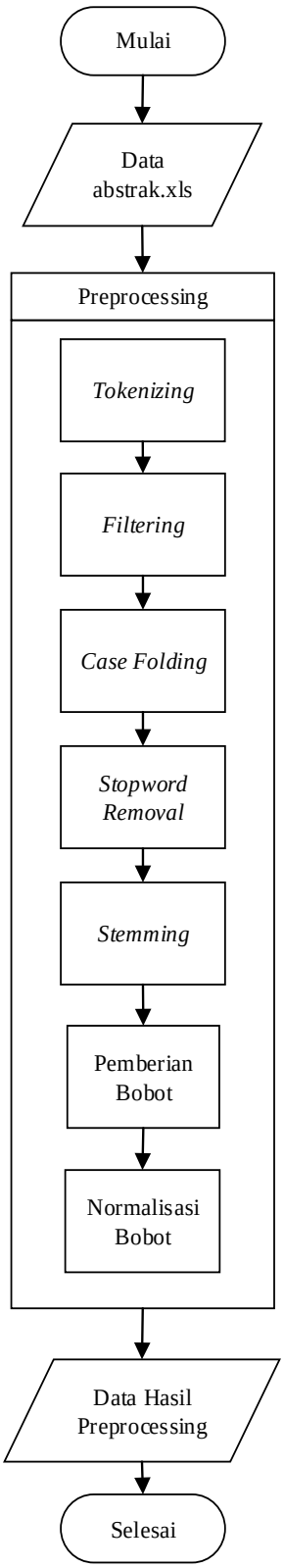


Gambar 3. 44 Jaringan semantik

3.8.6 Perancangan Prosedural

3.8.6.1 *Preprocessing*

Proses dimana teks dipersiapkan menjadi data untuk diolah. proses preprocessing yang dilakukan akan mempengaruhi hasil klasifikasi karena dari preprocessing ini akan diambil nilai bobotnya dengan menggunakan perhitungan bobot TF-IDF. Berikut gambaran tahapan preprocessing dapat dilihat pada gambar 3.44.



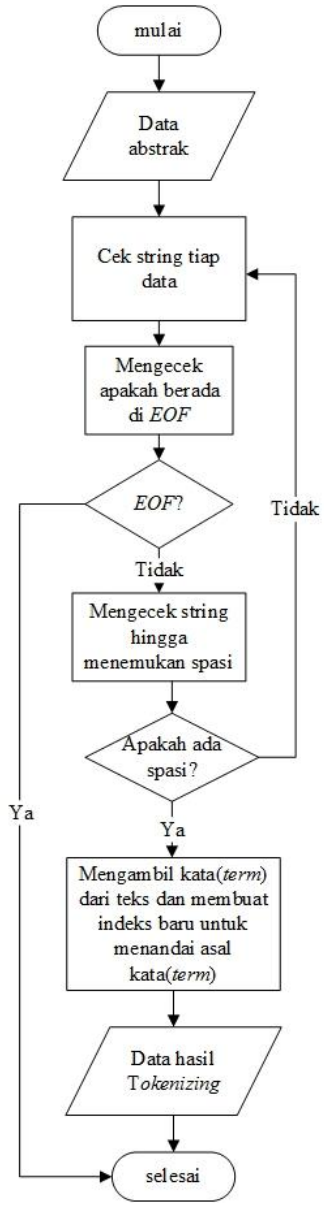
Gambar 3. 45 *Flowchart Preprocessing*

Adapun tahap *preprocessing* yang dilakukan adalah sebagai berikut :

4. Membaca data abstrak dalam format .xls
5. Melakukan proses *Tokenizing* dengan membaca kata(*term*) yang dipisahkan dengan spasi(‘ ‘)
6. Melakukan proses *Filtering* untuk menghilangkan *symbol*
7. Melakukan proses *Case Folding* untuk menyamakan *case* dari string
8. Melakukan proses *Stopword Removal* untuk menghilangkan kata(*term*) yang tidak digunakan
9. Melakukan proses *Stemming* untuk menghilangkan imbuhan pada kata(*term*)
10. Melakukan perhitungan bobot TF-IDF untuk memberikan nilai pada kata(*term*)
11. Melakukan Normalisasi bobot untuk menjaga nilai bobot pada rentang 0 sampai 1.
12. Proses selesai.

3.8.6.1.1 *Tokenizing*

Tahap awal yang dilakukan setelah data diinputkan yaitu tahap *tokenizing.* tahap ini diperlukan untuk mendapatkan hasil *term* yang akan menjadi data masukan pada tahap berikutnya yaitu filtering. Berikut tahapan yang dilakukan dapat dilihat pada gambar 3.45.



Gambar 3. 46 *Flowchart Tokenizing*

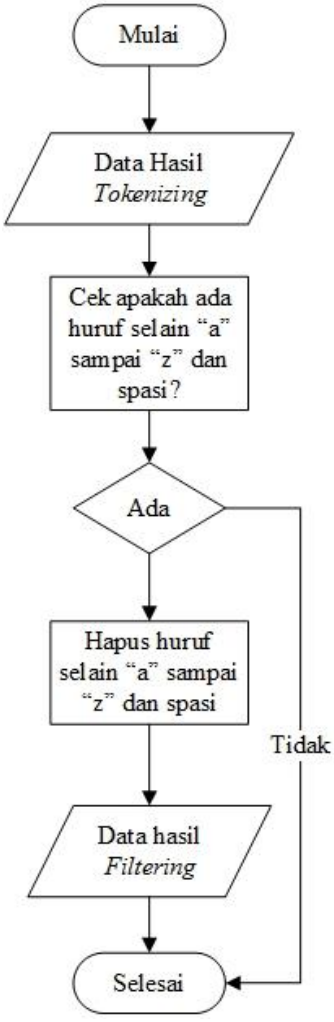
Adapun tahap *tokenizing* yang dilakukan adalah sebagai berikut :

1. Data masukan abstrak dalam format .xls
2. Memeriksa apakah sudah EOF. Jika sudah maka proses selesai, jika belum maka ke proses 3.
3. Memeriksa teks dengan memecehnya menjadi kata (*term*). Mengecek apakah menemukan spasi. Jike bertemu spasi maka ke prosse 4. Jika tidak maka scan kembali.

4. Mengambil kata (*term*) dari teks dan membuat indeks baru untuk menandai asal kata (*term*).
5. Proses selesai.

3.8.6.1.2 *Filtering*

Tahap selanjutnya yang dilakukan yaitu tahap *filtering.* Tahap ini diperlukan untuk mendapatkan hasil *term* dengan tidak adanya karakter lain selain karakter “a” sampai “z” dan spasi yang akan menjadi data masukan pada tahap berikutnya yaitu filtering. Penghilangan karakter-karakter tersebut dikarenakan tidak digunakan pada tahap selanjutnya. Berikut tahapan yang dilakukan dapat dilihat pada gambar 3.46.



Gambar 3. 47 *Flowchart Filtering*

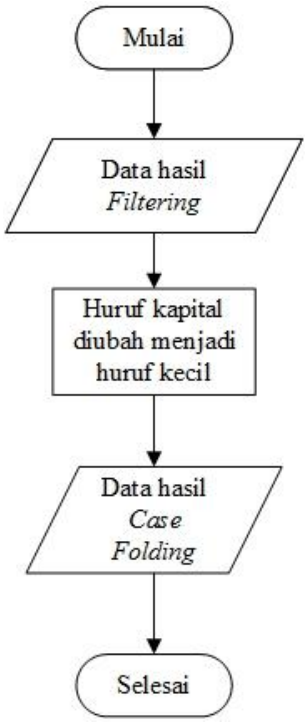
Adapun tahap *filtering* yang dilakukan adalah sebagai berikut:

1. Membaca data hasil *tokenizing*
2. Mengecek apakah ada huruf selain “a” sampai “z” dan “ “ (spasi). Jika tidak ditemukan, maka proses berakhir. Jika ada.
3. Hapus karakter yang bukan termasuk “a” sampai “z” dan “ ” (Spasi).
4. Proses selesai

3.8.6.1.3 *Case Folding*

Setelah mendapatkan data hasil *stopword removal* tahap selanjutnya dilakukan proses *case folding.* Tahap ini diperlukan untuk menghasilkan *term* dalam huruf kecil, yang akan dijadikan data masukan pada tahap berikutnya yaitu *stopword removal.*

Tahap ini juga mempengaruhi hasil perhitungan frakuensi *term* karena tidak semua dokumen teks konsisten dalam penggunaan huruf kapital, sehingga memungkinkan redudansi pada *term*. Berikut tahapan yang dilakukan dapat dilihat pada gambar 3.47.



Gambar 3. 48 *Flowchart Case Folding*

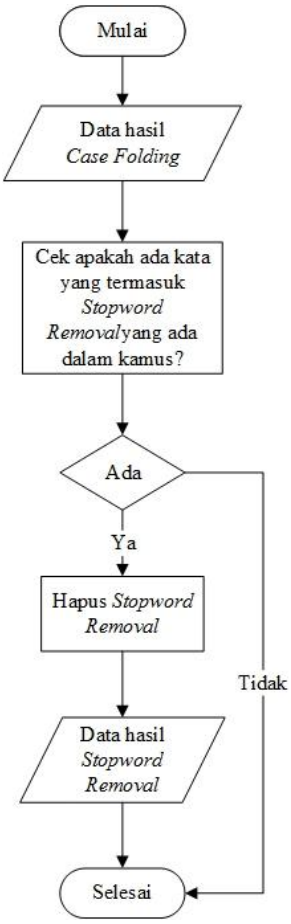
Adapun tahap *case folding* yang dilakukan adalah sebagai berikut:

1. Membaca data hasil *filtering*
2. Lakukan perubahan string lower pada setiap string.
3. Proses selesai

3.8.6.1.4 *Stopword Removal*

Setelah mendapatkan data hasil *filtering* tahap selanjutnya dilakukan proses *stopword removal*. tahap ini diperlukan untuk menghasilkan *term* yang hanya memiliki keterkaitan dengan topik tertentu yang akan dijadikan data masukan pada tahap berikutnya yaitu *stemming*

Untuk mendapatkan *term* tersebut maka dilakukan pembuangan pada kata-kata yang tidak terkait. Penentuan kata tersebut dibuang atau tidak dilakukan dengan membandingkannya dengan kata yang sudah ada pada kamus *stopword* yang telah dibuat. Penentuan kata yang *term*asuk *stopword removal* yang dibuat yaitu kata-kata yang sering muncul atau kata-kata tersebut tidak memiliki arti atau tidak relevan, Seperti kata sambung, kata depan, kata ganti, kata penghubung, dll. Namun artinya tidak deskriptif dan tidak memiliki keterkaitan dengan topik tertentu.Pembuangan kata-kata tersebut bertujuan meminimalkan kata yang muncul yang mepengaruhi pada perhitungan bobot. Berikut tahapan yang dilakukan dapat dilihat pada gambar 3.48.



Gambar 3. 49 *Flowchart Stopword Removal*

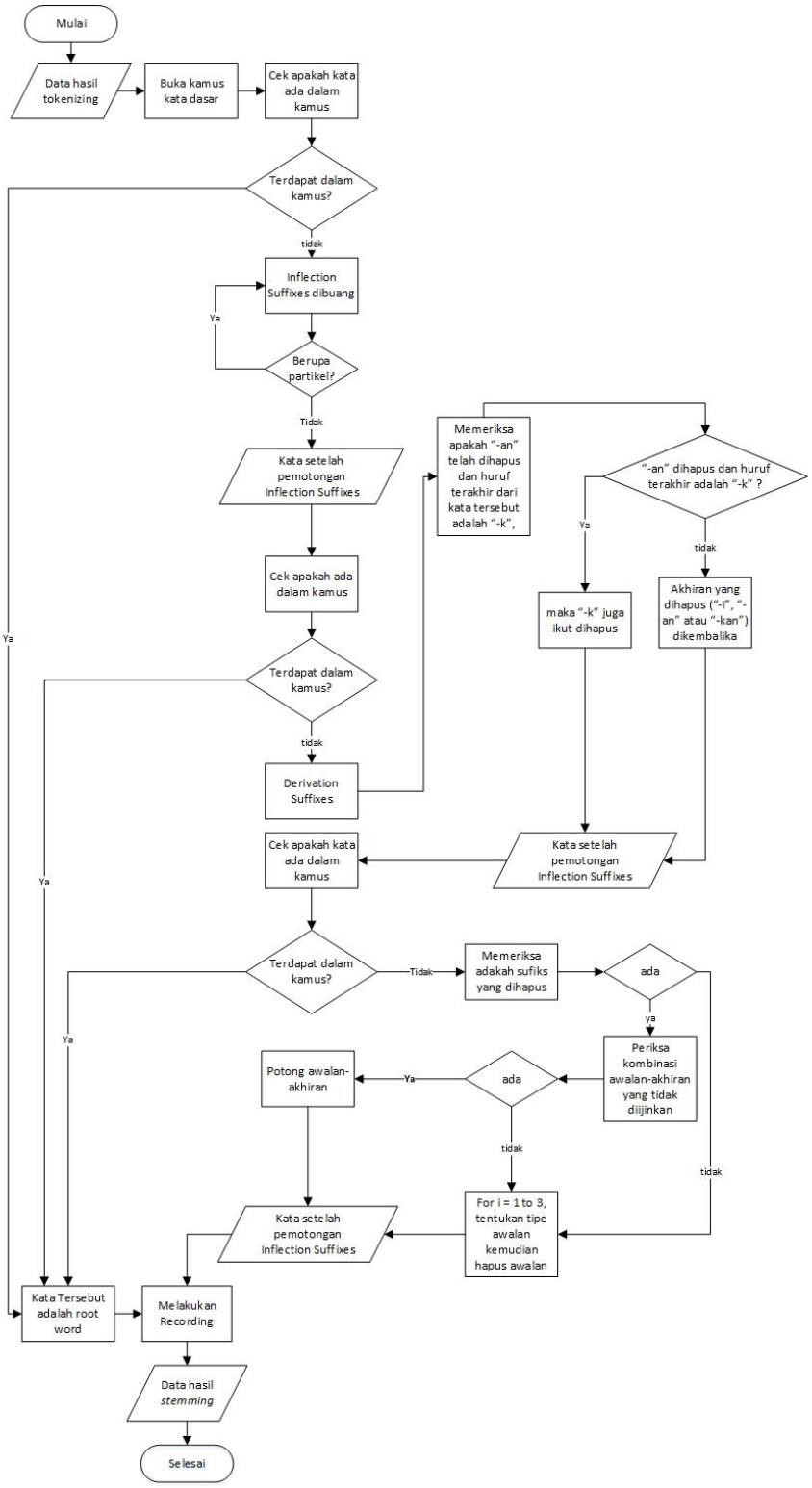
Adapun tahap *stopword removal* yang dilakukan adalah sebagai berikut:

1. Membaca data hasil *case folding*
2. Mengecek apakah ada kata yang termasuk *Stopword Removal* yang ada dalam kamus, jika tidak ada maka algoritma selesai, jika ada
3. Menghapus semua kata yang termasuk *stopword removal*
4. Proses selesai

3.8.6.1.5 *Stemming*

Setelah mendapatkan data hasil *stopword removal* tahap selanjutnya dilakukan proses *stemming.* Tahap ini diperlukan untuk menghasilkan *term* yang dengan kata dasar tanpa imbuhan yang akan dijadikan data masukan pada proses berikutnya yaitu pembobotan TF-IDF.

Kata-kata yang mempunyai imbuhan akan ditransformasikakn menjadi kata-kata dasar (*root word*) dengan menggunakan aturan-aturan tertentu. Karena dokumen yang di klasifikasi adalah teks berbahasa indonesia, maka aturan yang dipakai pada teks berbahasa Indonesia, tidak hanya sufiks melainkan prefiks, dan konfiks juga dihilangkan. Dalam penelitian ini digunakan algoritma Bobby Nazief dan Mirna Adriani dalam proses *stemming.* Berikut tahapan yang dilakukan dapat dilihat pada gambar 3.49.



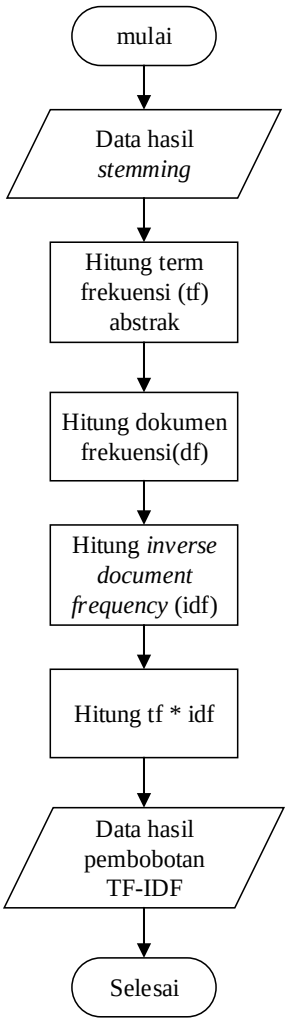
Gambar 3. 50 *Flowchart Stemming*

Adapun tahap *stemming* yang dilakukan adalah sebagai berikut:

1. Kata yang hendak di *stemming* dicari terlebih dahulu pada kamus. Jika ditemukan, maka kata tersebut adalah kata dasar, jika tidak maka langkah 2 yang dilakukan.
2. Pada kata hasil *stopword removal* akan dicek apakah memiliki *inflectional suffixes*, yaitu akhiran(“-lah”,“-kah”,“-tah”,“-pun”) dan kata ganti kepunyaan atau *possessive pronoun* PP (“-ku”, “-mu”, “-nya”) ternyata pada kata dalam data uji tidak terdapat *inflectional suffixes* dan *possessive pronoun* kemudian sistem melakukan proses selanjutnya *Derivation prefiks*.
2. kemudian akan dicek apakah memiliki *Derivation suffixes*, yaitu akhiran(“-an”, ”-i”, “-kan”) ternyata pada kata dalam data uji terdapat *Derivation suffixes* maka sistem menghapus akhiran *derivation suffixes* lalu sistem mengecek ke kamus kata dasar jika kata hasil *derivation suffixes* ada di dalam kamus kata dasar maka sistem berhenti jika tidak ditemukan akan dilakukan proses selanjutnya.
3. Kemudian akan dicek, apakah memiliki *derivation prefixes*, ternyata mengandung *derivation prefiixes* kemudian sistem mencari kata ke kamus kata dasar jika ditemukan maka kata adalah kata dasar dan proses berhenti, jika tidak maka kata dikembalikan dan proses berhenti.

3.8.6.1.6 Pembobotan TF-IDF

Setelah mendapatkan data hasil *stemming* tahap selanjutnya dilakukan proses pembobotan TF-IDF. Tahap ini diperlukan untuk menghasilkan bobot masing masing dokumen pada setiap *term*. Berikut alur proses dapat dilihat pada gambar 3.50.



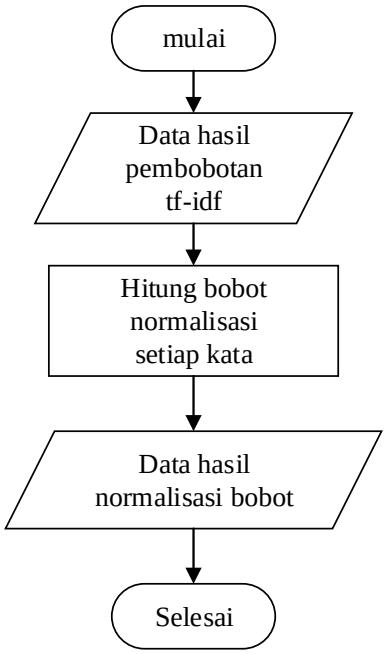
Gambar 3. 51 *Flowchart* Pembobotan TF-IDF

Adapun tahap pembobotan TF-IDF yang dilakukan adalah sebagai berikut:

1. Membaca data hasil *stemming*
2. Menghitung jumlah kemunculan *term* (*tf*)
3. Menghitung jumlah kemunculan *term* dalam kumpulan dokumen (*df*)
4. Menghitung *inverse document frequency* (IDF)
5. Menghitung bobot, dengan mengalikan hasil *tf* dengan IDF
6. Proses selesai

3.8.6.1.7 Normalisasi Bobot

Setelah mendapatkan data hasil pembobotan TF-IDF tahap selanjutnya dilakukan proses normalisasi pembobotan TF-IDF. Tahap ini diperlukan untuk menghasilkan bobot yang menjaga nilai tetap pada rentang 0 – 1. Berikut alur proses dapat dilihat pada gambar 3.51.



Gambar 3. 52 *Flowchart* Normalisasi Bobot

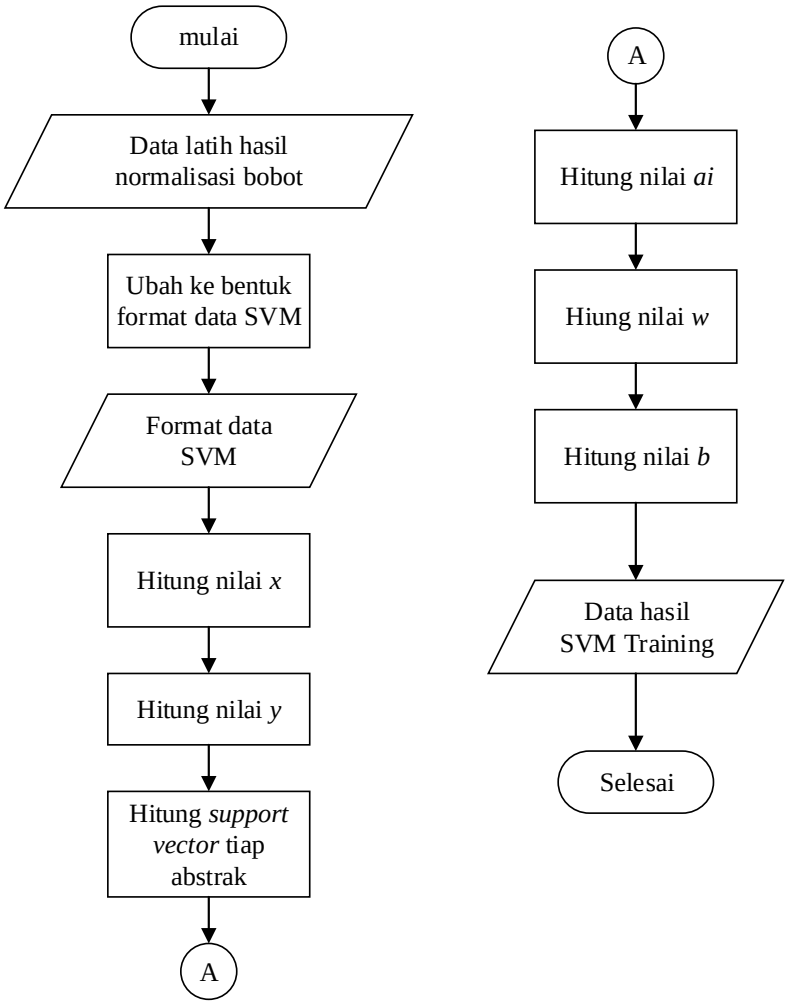
Adapun tahap normalisasi bobot yang dilakukan adalah sebagai berikut:

1. Membaca data hasil TF-IDF
2. Menghitung bobot normalisasi setiap kata(*term*)
3. Proses selesai

3.8.6.2 Klasifikasi *Support Vector Machine* (SVM)

3.8.6.2.1 SVM Training

Pada proses pelatihan SVM bertujuan untuk menemukan vektor α, nilai w dan konstanta b untuk mendapatkan hyperplane terbaik. Dalam proses pelatihan dibutuhkan satu set *input-output* data. Sebagai inisialisasi awal dokumen abstrak keilmuan A diberi label 1 selain itu diberi label -1. Berikut alur proses dapat dilihat pada gambar 3.52.



Gambar 3. 53 *Flowchart* SVM Training

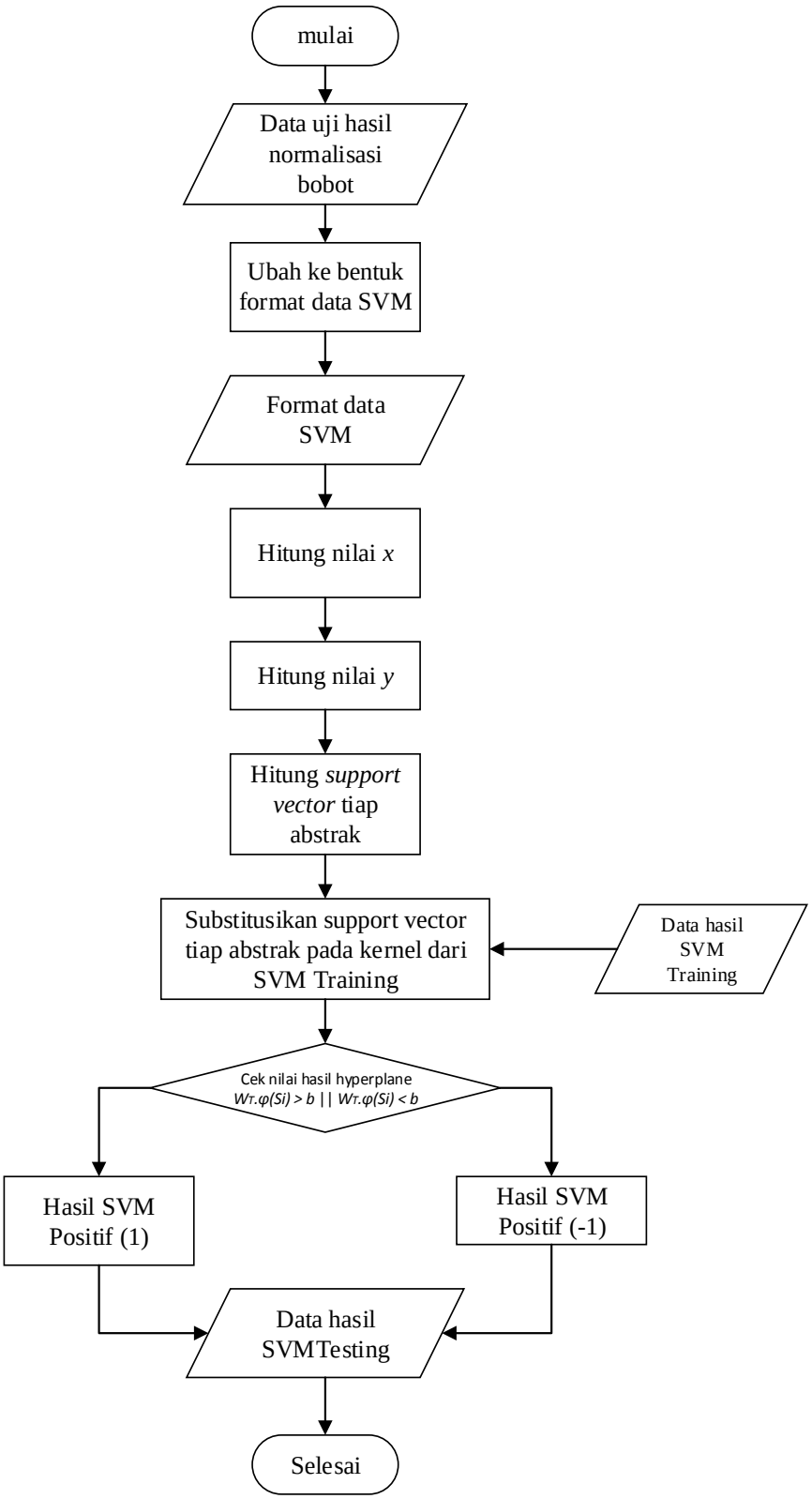
Adapun tahap SVM Training yang dilakukan adalah sebagai berikut:

1. Data yang sudah dibobotkan kemudian dibuat menjadi bentuk format SVM.
2. Format data SVM yang sudah dibentuk kemudian dihitung *dot product* nilai *x* dari bobot begitu pula nilai *y*.
3. Setelah didapatkan nilai *x* dan *y* dari tiap abstrak, hitung *support vector* dari tiap abstrak.
4. Setelah didapatkan nilai *support vector*, gunakan nilai *support vrctor* untuk mendapatkan nilai alpha(*a*) dengan metode substitusi.

5. Setelah didapatkan nilai *support vector* dan *a,* gunakan untuk mendapatkan nilai *w* dan *b* sebagai pemisah atau *hyperplane* dari model SVM.

3.8.6.2.2 SVM Testing

Nilai w dan b atau *hyperplane* yang dihaasilkan dari SVM Training dapat menentukan data abstrak masuk dalam kelas positif atau negatif. Jika nilai hasil uji lebih besar dari nilai *hyperplane* maka abstrak tersebut masuk dalam kelas positif, jika lebih kecil dari nilai *hyperplane* maka abstrak tersebut masuk dalam kelas negatif. Berikut alur proses dapat dilihat pada gambar 3.52.



Gambar 3. 54 *Flowchart* SVM Testing

Adapun tahap SVM Testing yang dilakukan adalah sebagai berikut:

1. Membaca model SVM dari SVM Training dan membaca hasil pembobotan data testing.
2. Data yang sudah dibobotkan kemudian dibuat menjadi bentuk format SVM.
3. Format data SVM yang sudah dibentuk kemudian dihitung *dot product* nilai *x* dari bobot begitu pula nilai *y*.
4. Setelah didapatkan nilai *x* dan *y* dari tiap abstrak, hitung *support vector* dari tiap abstrak.
5. Setelah didapatkan nilai *support vector*, gunakan nilai *support vrctor* untuk disubstitusikan pada nilai w dan b dari hasil SVM Training.
6. Jika nilai hasil uji lebih besar dari nilai *hyperplane* maka data testing tersebut masuk dalam kelas positif, jika lebih kecil dari nilai *hyperplane* maka data testing tersebut masuk dalam kelas negatif