

BAB IV

PENGEMBANGAN PERANGKAT LUNAK

4.1 Pendahuluan

Bab ini akan mendeskripsikan rangkaian proses pengerjaan pada pengembangan perangkat lunak menggunakan metode RUP (*Rational Unified Process*) secara rinci. Adapun proses pengembangan perangkat lunak menggunakan metode RUP dalam penelitian ini terdiri dari beberapa fase yaitu fase insepisi, fase elaborasi, fase konstruksi, dan fase transisi.

4.2 Fase Insepisi

Fase insepisi merupakan tahapan pertama yang perlu dilakukan untuk mengidentifikasi kebutuhan dan fitur yang dibutuhkan dalam proses pengembangan perangkat lunak. Rangkaian proses yang terjadi dalam fase ini diawali dengan proses analisis system, mengidentifikasi fungsionalitas dari perangkat lunak, kebutuhan pengguna dan merancang model diagram *use case*.

4.2.1 Pemodelan Bisnis

Klasifikasi teks merupakan suatu proses dalam mengetahui apakah suatu dokumen teks termasuk dalam suatu kategori kelas yang telah ditentukan. Dokumen teks yang digunakan dalam penelitian ini berupa pertanyaan berbahasa Indonesia. Pertanyaan berbahasa Indonesia dapat terbagi menjadi 3 kategori yaitu kelas *factoid*, *non-factoid* dan *others*. Untuk mempermudah penentuan kategori suatu kelas pertanyaan berbahasa Indonesia maka diperlukan suatu proses klasifikasi. Pada proses klasifikasi teks dapat dilakukan metode seleksi fitur untuk mengurangi

jumlah dimensi data sebagai upaya dalam meningkatkan hasil kinerja dari proses klasifikasi. Dalam penelitian ini, proses klasifikasi pada pertanyaan berbahasa Indonesia dapat menggunakan metode SVM (*Support Vector Machine*) dan seleksi fitur yang bekerja dengan mempelajari data latih dalam melakukan klasifikasi pada suatu objek.

Perangkat lunak yang dikembangkan merupakan suatu perangkat lunak berbasis desktop. Untuk mengetahui luaran hasil klasifikasi terbaik maka akan dilakukan perbandingan hasil klasifikasi pada algoritma SVM dan metode seleksi fitur yang digunakan dalam penelitian ini.

4.2.2 Kebutuhan Sistem

Untuk memenuhi pemodelan suatu perangkat lunak maka perlu adanya suatu perancangan yang dapat memenuhi kebutuhan sistem, baik berupa kebutuhan fungsional maupun kebutuhan non-fungsional. Kebutuhan fungsional merupakan komponen utama yang harus dipenuhi pada suatu perangkat lunak. Sedangkan kebutuhan non-fungsional merupakan komponen tambahan pada suatu perangkat lunak. Kebutuhan fungsional ditampilkan pada tabel IV.1 dan kebutuhan non-fungsional ditampilkan pada table IV.2.

Tabel IV-1. Kebutuhan Fungsional

No.	Kebutuhan Fungsional
1.	Perangkat lunak dapat melakukan data <i>preprocessing</i>
2.	Perangkat lunak dapat melakukan proses klasifikasi menggunakan algortima SVM

3.	Perangkat lunak dapat melakukan proses seleksi fitur <i>Information Gain</i> dan proses klasifikasi menggunakan algoritma SVM
4.	Perangkat lunak dapat melakukan proses seleksi fitur <i>Chi-Square</i> dan proses klasifikasi menggunakan algoritma SVM
5.	Perangkat lunak dapat melakukan proses seleksi fitur <i>Mutual Information</i> dan proses klasifikasi menggunakan algoritma SVM
6.	Perangkat lunak dapat melakukan pengujian berdasarkan model klasifikasi yang digunakan dan data masukan dari pengguna berupa pertanyaan berbahasa Indonesia

Tabel IV-2. Kebutuhan Non-Fungsional

No.	Kebutuhan Non-Fungsional
1.	Perangkat lunak dapat menampilkan <i>user interface</i> yang mudah digunakan dan dipahami

4.2.3 Analisis dan Desain

Salah satu proses penting yang terjadi pada fase insepisi ialah aktivitas analisis dan desain. Aktivitas analisis dan desain bertujuan untuk menganalisis kebutuhan perangkat lunak dan membuat rancangan desain *use case* diagram perangkat lunak.

4.2.3.1 Analisis Kebutuhan Perangkat Lunak

Berdasarkan uraian yang ada pada pemodelan bisnis, akan dibuat suatu perangkat lunak yang dapat melakukan proses klasifikasi pada pertanyaan

berbahasa indonesia. Adapun kemampuan yang harus dimiliki pada perangkat lunak ialah sebagai berikut.

1. Melakukan proses *data preprocessing*
2. Melakukan proses klasifikasi menggunakan algoritma *Support Vector Machine*.
3. Melakukan proses klasifikasi menggunakan metode seleksi fitur *Information Gain* dan algoritma *Support Vector Machine*.
4. Melakukan proses klasifikasi menggunakan metode seleksi fitur *Chi-Square* dan algoritma *Support Vector Machine*.
5. Melakukan proses klasifikasi menggunakan metode seleksi fitur *Mutual Information* dan algoritma *Support Vector Machine*.
6. Melakukan pengujian berdasarkan model klasifikasi yang digunakan dan data masukan dari pengguna berupa pertanyaan berbahasa Indonesia.

Berdasarkan kemampuan perangkat lunak yang telah disebutkan, Dilakukan pengembangan perangkat lunak yang diawali dengan proses pengumpulan data yang bertujuan agar perangkat lunak dapat melakukan pengolahan data. Selanjutnya ialah melakukan data *preprocessing* yang terdiri dari *noise removal*, *case folding* dan *tokenizing*. Kemudian, data yang telah dilakukan *preprocessing* diseleksi untuk menghilangkan *term* yang kurang relevan menggunakan 3 skema seleksi fitur yaitu *Information Gain*, *Chi-Square* dan *Mutual Information*. Setelah itu, dilakukan pembobotan pada setiap *term* yang telah diseleksi menggunakan TF-IDF. Hasil dari data yang telah dilakukan seleksi fitur dan pembobotan akan

digunakan pada proses klasifikasi menggunakan algoritma *Support Vector Machine*.

4.2.3.2 Analisis Data

Penelitian ini menggunakan data berupa kumpulan teks pertanyaan dalam Bahasa Indonesia yang telah dilabeli. Data yang digunakan berjumlah 1195 data pertanyaan yang terbagi menjadi 43% data pertanyaan untuk label *factoid* dengan rincian yaitu 519 data, 41% data pertanyaan untuk label *non-factoid* dengan rincian yaitu 491 data dan 16% data pertanyaan untuk label *others* dengan rincian yaitu 185 data. Kumpulan data tersebut disimpan dalam file dengan format .xls.

4.2.3.3 Analisis *Text Preprocessing*

Text processing yang dilakukan pada penelitian ini bertujuan untuk membuat data menjadi terstruktur dan menghilangkan *noise* pada data yang nantinya dapat mempengaruhi hasil klasifikasi. *Text Preprocessing* terdiri dari beberapa proses diantaranya *noise removal*, *case folding* dan *tokenizing*. Gambaran proses *text processing* yang menggunakan sampel data yang terdiri dari 15 data *training* yang terbagi menjadi 5 data berlabel positif dan 5 data berlabel negatif dan 5 data berlabel *others*. Gambaran *text preprocessing* dapat dilihat pada tabel IV-3.

Tabel IV-3. Contoh Data Pertanyaan

Data	Pertanyaan	Label
D1	alat apa yang digunakan untuk mencampur adonan kue	<i>Factoid</i>
D2	ada dimana gelang pemberian dari nenekmu	<i>Factoid</i>
D3	katakan siapa yang menyuruhmu	<i>Factoid</i>

D4	kapan diperingatinya hari pahlawan	<i>Factoid</i>
D5	siapa penemu telepon	<i>Factoid</i>
D6	apa alasanmu melakukan perbuatan keji itu	<i>Non-factoid</i>
D7	apa kendala yang dihadapi dalam mengerjakan tugas akhir	<i>Non-factoid</i>
D8	apa kegunaan mixer dalam proses pembuatan kue	<i>Non-factoid</i>
D9	jelaskan padaku mengapa kau merahasiakan ini dariku	<i>Non-factoid</i>
D10	bagaimana dokter itu bisa terpapar virus Covid-19	<i>Non-factoid</i>
D11	bersediakah kamu mengajariku bagaimana cara bermain sepeda	<i>Others</i>
D12	bukankah kamu tau harus bagaimana sekarang	<i>Others</i>
D13	menurutmu siapa agen sepakbola terbaik	<i>Others</i>
D14	ingatkah kau kapan terakhir kita berkunjung kesini	<i>Others</i>
D15	siapa saja yang ikut ke Gunung Simeulu	<i>Others</i>

1. Noise Removal

Noise removal merupakan suatu proses yang digunakan untuk menghapus karakter *non-alphabet* seperti simbol, angka, karakter *unicode*, url, dan hastag. Tujuan dari *Noise Removal* ialah menghilangkan *term* yang dapat mengurangi hasil kinerja klasifikasi. Hasil dari *Noise removal* ditampilkan pada tabel IV-4.

Tabel IV-4. Hasil *Noise Removal*

Data	Pertanyaan	Label
D1	alat apa yang digunakan untuk mencampur adonan kue	<i>Factoid</i>
D2	ada dimana gelang pemberian dari nenekmu	<i>Factoid</i>
D3	katakan siapa yang menyuruhmu	<i>Factoid</i>
D4	kapan diperingatinya hari pahlawan	<i>Factoid</i>

D5	siapa penemu telepon	<i>Factoid</i>
D6	apa alasanmu melakukan perbuatan keji itu	<i>Non-factoid</i>
D7	apa kendala yang dihadapi dalam mengerjakan tugas akhir	<i>Non-factoid</i>
D8	apa kegunaan mixer dalam proses pembuatan kue	<i>Non-factoid</i>
D9	jelaskan padaku mengapa kau merahasiakan ini dariku	<i>Non-factoid</i>
D10	bagaimana dokter itu bisa terpapar virus Covid-19	<i>Non-factoid</i>
D11	bersediakah kamu mengajarku bagaimana cara bermain sepeda	<i>Others</i>
D12	bukankah kamu tau harus bagaimana sekarang	<i>Others</i>
D13	menurutmu siapa saja agen sepakbola terbaik	<i>Others</i>
D14	ingatkah kau kapan terakhir kita berkunjung kesini	<i>Others</i>
D15	siapa saja yang ikut ke Gunung Simeulu	<i>Others</i>

2. Case Folding

Case folding merupakan proses yang digunakan untuk mengubah seluruh karakter huruf kedalam bentuk huruf kecil atau *lowercase*. Hasil proses *case folding* ditampilkan pada tabel IV-5.

Tabel IV-5. Hasil Proses *Case Folding*

Data	<i>Tweet</i>	Label
D1	alat apa yang digunakan untuk mencampur adonan kue	<i>Factoid</i>
D2	ada dimana gelang pemberian dari nenekmu	<i>Factoid</i>
D3	katakan siapa yang menyuruhmu	<i>Factoid</i>
D4	kapan diperingatinya hari pahlawan	<i>Factoid</i>
D5	siapa penemu telepon	<i>Factoid</i>
D6	apa alasanmu melakukan perbuatan keji itu	<i>Non-factoid</i>

D7	apa kendala yang dihadapi dalam mengerjakan tugas akhir	<i>Non-factoid</i>
D8	apa kegunaan mixer dalam proses pembuatan kue	<i>Non-factoid</i>
D9	jelaskan padaku mengapa kau merahasiakan ini dariku	<i>Non-factoid</i>
D10	bagaimana dokter itu bisa terpapar virus covid-19	<i>Non-factoid</i>
D11	bersediakah kamu mengajarku bagaimana cara bermain sepeda	<i>Others</i>
D12	bukankah kamu tau harus bagaimana sekarang	<i>Others</i>
D13	menurutmu siapa saja agen sepakbola terbaik	<i>Others</i>
D14	ingatkah kau kapan terakhir kita berkunjung kesini	<i>Others</i>
D15	siapa saja yang ikut ke gunung simeulu	<i>Others</i>

3. Tokenizing

Tokenizing merupakan suatu proses yang digunakan untuk memecah suatu kalimat menjadi token atau kata tunggal berdasarkan susunan katanya. Hasil proses *tokenizing* ditampilkan pada tabel IV-6.

Tabel IV-6. Hasil Proses *Tokenizing*

Data	<i>Tweet</i>	Label
D1	“alat”, “apa”, “yang”, “guna”, “untuk”, “campur”, “adon”, “kue”	<i>Factoid</i>
D2	“ada”, “mana”, “gelang”, “beri”, “dari”, “nenek”	<i>Factoid</i>
D3	“kata”, “siapa”, “yang”, “suruh”	<i>Factoid</i>
D4	“kapan”, “ingat”, “hari”, “pahlawan”	<i>Factoid</i>
D5	“siapa”, “temu”, “telepon”	<i>Factoid</i>
D6	“apa”, “alas”, “laku”, “buat”, “keji”, “itu”	<i>Non-factoid</i>
D7	“apa”, “kendala”, “yang”, “hadap”, “dalam”, “kerja”, “tugas”, “akhir”	<i>Non-factoid</i>

D8	“apa”, “guna”, “mixer”, “dalam”, “proses”, “buat”, “kue”	<i>Non-factoid</i>
D9	“jelas”, “pada”, “mengapa”, “kau”, “rahasia”, “ini”, “dari”	<i>Non-factoid</i>
D10	“Bagaimana”, “dokter”, “itu”, “bisa”, “papar”, “virus”, “covid-19”	<i>Non-factoid</i>
D11	“sedia”, “kamu”, “ajar”, “bagaimana”, “cara”, “main”, “sepeda”	<i>Others</i>
D12	“bukankah”, “kamu”, “tau” “harus”, “bagaimana”, “sekarang”	<i>Others</i>
D13	“turut”, “siapa”, “agen”, “sepakbola”, “baik”	<i>Others</i>
D14	“ingat”, “kau”, “kapan”, “akhir”, “kita”, “kunjung”, “kesini”	<i>Others</i>
D15	“siapa”, “saja”, “yang”, “ikut”, “ke”, “gunung”, “simeulu”	<i>Others</i>

4.2.3.4 Analisis Proses Klasifikasi

Proses klasifikasi pada penelitian ini akan menggunakan algoritma *Support Vector Machine* untuk melakukan pelatihan dan pengujian pada data yang digunakan. Proses klasifikasi terbagi kedalam empat skema proses klasifikasi diantaranya proses klasifikasi yang menggunakan algoritma SVM tanpa metode seleksi fitur, proses klasifikasi yang menggunakan kombinasi antara algoritma SVM dengan metode seleksi fitur *Information Gain*, proses klasifikasi yang menggunakan kombinasi antara algoritma SVM dengan metode seleksi fitur *Chi-Square*, dan proses klasifikasi yang menggunakan kombinasi antara algoritma SVM dengan metode seleksi fitur *Mutual Information*.

Langkah-langkah untuk setiap skema proses klasifikasi diuraikan sebagai berikut.

1. Proses klasifikasi menggunakan metode Support Vector Machine tanpa seleksi fitur

Langkah 1 : Melakukan ekstraksi fitur untuk menghitung bobot tiap *term* setelah data *preprocessing* menggunakan TF-IDF yang ditampilkan pada tabel IV-7.

Tabel IV-7. Hasil Pembobotan TF- IDF

<i>Term</i>	TF															DF	N	IDF
	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15			
alat	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	15	2,176
apa	1	0	0	0	0	1	1	1	0	0	0	0	0	0	0	4	15	1,574
yang	1	0	1	0	0	0	1	0	0	0	0	0	0	0	1	4	15	1,574
guna	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	3	15	1,699
untuk	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	15	2,176
campur	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	15	2,176
adon	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	15	2,176
kue	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	15	2,176
ada	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	15	2,176
mana	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	15	2,176
gelang	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	15	2,176
beri	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	15	2,176
dari	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	2	15	1,875
nenek	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	15	2,176
kata	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	15	2,176
siapa	0	0	1	0	1	0	0	0	0	0	0	0	1	0	1	4	15	1,574
suruh	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	15	2,176

kapan	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	2	15	1,875
ingat	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	15	2,176
hari	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	15	2,176
pahlawan	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	15	2,176
temu	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	15	2,176
telepon	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	2	15	1,875
alas	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	15	2,176
laku	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	15	2,176
buat	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	2	15	1,875
keji	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	2	15	1,875
itu	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	2	15	1,875
kendala	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	15	2,176
hadap	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	15	2,176
dalam	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	3	15	1,699
kerja	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	15	2,176
tugas	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	15	2,176
akhir	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	2	15	1,875
mixer	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	15	2,176
proses	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	15	2,176
jelas	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	15	2,176
pada	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	15	2,176
mengapa	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	15	2,176
kau	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	2	15	1,875
rahasia	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	15	2,176
ini	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	15	2,176
bagaimana	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	3	15	1,699
dokter	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	15	2,176

bisa	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	15	2,176
papar	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	15	2,176
virus	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	15	2,176
covid-19	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	15	2,176
sedia	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	15	2,176
kamu	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	15	2,176
ajar	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	2	15	1,875
cara	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	15	2,176
main	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	15	2,176
sepeda	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	15	2,176
bukankah	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	15	2,176
tau	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	15	2,176
harus	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	15	2,176
sekarang	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	15	2,176
turut	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	15	2,176
agen	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	15	2,176
sepakbola	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	15	2,176
baik	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	15	2,176
kita	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	15	2,176
kunjung	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	15	2,176
kesini	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	15	2,176
saja	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	2	15	1,875
ikut	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	15	2,176
ke	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	15	2,176
gunung	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	15	2,176
simeulu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	15	2,176

Langkah 2 : Melakukan proses pelatihan dan pengujian pada data menggunakan *Support Vector Machine*.

Pada proses pelatihan dilakukan pencarian nilai *hyperlane* terbaik dengan melakukan perhitungan untuk mendapatkan nilai dari *lagrange multipliers* (α) menggunakan persamaan II-15 dan kemudian mencari nilai bias menggunakan persamaan II-16. Data latih yang digunakan sebagai contoh dalam perhitungan ialah D1 dan D2. Fungsi kernel yang digunakan merupakan kernel linear dengan nilai $C=1$. Tahapan proses klasifikasi diuraikan sebagai berikut.

a. Menghitung nilai *dot product* pada data latih.

$$\begin{aligned}
 x_1^T x_1 &= (2.176 \times 2.176) + (1.574 \times 1.574) + (1.574 \times 1.574) + (1.699 \times 1.699) + \\
 &\quad (2.176 \times 2.176) + (2.176 \times 2.176) + (2.176 \times 2.176) + (2.176 \times 2.176) + \\
 &\quad (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + \\
 &\quad (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + \\
 &\quad (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + \\
 &\quad (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + \\
 &\quad (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + \\
 &\quad (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + \\
 &\quad (0 \times 0) + (0 \times 0) + (0 \times 0) \\
 &= 31.519
 \end{aligned}$$

$$\begin{aligned}
 x_1^T x_2 &= (2.176 \times 0) + (1.574 \times 0) + (1.574 \times 0) + (1.699 \times 1.699) + (2.176 \times 0) + \\
 &\quad (2.176 \times 0) + (2.176 \times 0) + (2.176 \times 0) + (0 \times 2.176) + (0 \times 2.176) + \\
 &\quad (0 \times 2.176) + (0 \times 2.176) + (0 \times 1.875) + (0 \times 2.176) + (0 \times 0) + (0 \times 0)
 \end{aligned}$$

$$\begin{aligned}
& + (0x0) + (0x0) (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) \\
& (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) \\
& + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + \\
& (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + \\
& (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + \\
& (0x0) + (0x0) + (0x0) \\
& = 2.886
\end{aligned}$$

$$\begin{aligned}
x_2^T x_1 = & (0x2.176) + (0x1.574) + (0x1.574) + (1.699x1.699) + (0x2.176) + \\
& (0x2.176) + (0x2.176) + (0x2.176) + (2.176x0) + (2.176x0) + \\
& (2.176x0) + (2.176x0) + (1.875x0) + (2.176x0) + (0x0) + (0x0) \\
& + (0x0) + (0x0) (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) \\
& (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) \\
& + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + \\
& (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + \\
& (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + \\
& (0x0) + (0x0) + (0x0) \\
& = 2.886
\end{aligned}$$

$$\begin{aligned}
x_2^T x_2 = & (0x0) + (0x0) + (1.574x1.574) + (1.699x1.699) + (0x0) + (0x0) + (0x0) \\
& + (0x0) + (2.176x2.176) + (2.176x2.176) + (2.176x2.176) + \\
& (2.176x2.176) + (1.875x1.875) + (2.176x2.176) + (0x0) + (0x0) \\
& + (0x0) + (0x0) (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) \\
& (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + \\
& (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + \\
& (0x0) + (0x0) + (0x0)
\end{aligned}$$

$$\begin{aligned}
& + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + \\
& (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) \\
& + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + \\
& (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) \\
& + (0x0) + (0x0) + (0x0) \\
& = 30,78
\end{aligned}$$

- b. Menghitung nilai *hyperlane* terbaik (a) menggunakan persamaan II-15 dan fungsi kernel *linear* dengan nilai $C = 1$.

$$\begin{aligned}
L(a) &= \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^n a_i a_j y_i y_j K(x_i, x_j) \\
0 &= (a_1 + a_2) - \frac{1}{2} \left\{ (a_1 a_1 y_1 y_1 x_1^T x_1) + (a_1 a_2 y_1 y_2 x_1^T x_2) + \right. \\
& \left. (a_2 a_1 y_2 y_1 x_2^T x_1) + (a_2 a_2 y_2 y_2 x_2^T x_2) \right\} \\
0 &= (a_1 + a_2) - \frac{1}{2} \left\{ (a_1 a_1 (31.52)) + (a_1 a_2 (-2.89)) + \right. \\
& \left. (a_2 a_1 (-2.89)) + (a_2 a_2 (30.08)) \right\}
\end{aligned}$$

$$\text{Dengan } \sum_{i=1}^n a_i y_i = 0$$

$$a_1(+1) + a_2(-1) = 0$$

$$a_1 - a_2 = 0$$

$$a_1 = a_2$$

$$0 = (a_1 + a_2) - \frac{1}{2} \left\{ (a_1 a_1 (31.52)) + (a_1 a_2 (-2.89)) + \right. \\ \left. (a_2 a_1 (-2.89)) + (a_2 a_2 (30.08)) \right\}$$

$$0 = (2a_1) - (27.91a_1 a_1)$$

$$43.38a_1 a_1 = 2a_1$$

$$a_1 = 0.036$$

$$\text{Nilai } a_1 = a_2 = 0.0225$$

- c. Mencari nilai bias melalui persamaan II-16.

$$\begin{aligned}
b &= \frac{1}{NSV} \sum_{x_j \in SV} \left(\frac{1}{y_i} - \sum_{x_j \in SV} (a_j y_j K(x_j, x_i)) \right) \\
&= \frac{1}{NSV} \left\{ \left(\frac{1}{y_1} - (a_1 y_1 x_1^T x_1) + (a_2 y_2 x_2^T x_1) \right) + \left(\frac{1}{y_2} - (a_1 y_1 x_1^T x_2) + \right. \right. \\
&\quad \left. \left. (a_2 y_2 x_2^T x_2) \right) \right\} \\
&= \frac{1}{2} \left\{ \left(\frac{1}{1} - (0.036 \times 1 \times 31.519) + (0.036 \times (-1) \times 2.886) \right) + \right. \\
&\quad \left. \left(\frac{1}{-1} - (0.036 \times 1 \times 2.886) + (0.036 \times (-1) \times 30.078) \right) \right\} \\
&= -0.03
\end{aligned}$$

Pada proses pelatihan akan dibangun 3 model pelatihan menggunakan metode SVM, dimana untuk setiap model yang dibangun berdasarkan proses pelatihan untuk setiap 2 kelas. Model yang didapatkan pada setiap proses pelatihan akan digunakan sebagai nilai masukan pada proses klasifikasi. Selanjutnya, akan dilakukan proses *voting* untuk menentukan hasil klasifikasi berdasarkan 3 fungsi keputusan dari model yang dibangun menggunakan metode *multi class SVM one-against-one*. Fungsi keputusan ditampilkan pada Tabel IV-12.

Tabel IV-9. Klasifikasi *Multiclass SVM one-against-one*

$y = 1$	$y = -1$	Fungsi Keputusan
Kelas Factoid	Kelas Non-factoid	$f_{12}(x) = (w_{12})x + b_{12}$
Kelas Factoid	Kelas Others	$f_{13}(x) = (w_{13})x + b_{13}$
Kelas Non-factoid	Kelas Others	$f_{23}(x) = (w_{23})x + b_{23}$

$$\begin{aligned}
& + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + \\
& (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) \\
& + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + \\
& (0x0) + (0x0) + (0x0) + (0x0) + (0x0) \\
& = 2.886
\end{aligned}$$

2. Menghitung fungsi pemisah optimal menggunakan persamaan II-15.

$$\begin{aligned}
f(x) &= \text{sgn}(\sum_{i=1}^{nsv} \alpha_i y_i K(x_i, x_d) + b) \\
&= \text{sgn}\left((a_1 y_1 K(x_1, x_d)) + (a_1 y_2 K(x_2, x_d)) + b\right) \\
&= \text{sgn}\left((0.036 \times 1 \times 31.519) + (0.036 \times (-1) \times 2.886) + \right. \\
&\quad \left. (-0,03)\right) \\
&= 1
\end{aligned}$$

Pada proses klasifikasi yang dilakukan pada persamaan diatas untuk data uji (x_d), menunjukkan hasil klasifikasi berupa +1 yaitu kelas factoid berdasarkan label yang diberikan pada tabel fungsi keputusan sebelumnya. Dikarenakan hasil klasifikasi menunjukkan kelas factoid maka *vote* untuk kelas factoid akan ditambah satu.

2. Proses klasifikasi menggunakan seleksi fitur *Information Gain* dan algoritma *Support Vector Machine* .

Langkah 1 : Melakukan proses seleksi fitur pada data yang telah di *preprocessing* menggunakan *Information Gain*. Hasil perhitungan bobot nilai ditampilkan pada tabel IV-10.

Tabel IV-10. Perhitungan Bobot Nilai *Information Gain*

Perhitungan Information Gain (Total Data = 15)															
<i>Term</i>	Data	Factoid	Non-factoid	Other	Tidak Hadir Factoid	Hadir Factoid	Tidak Hadir Non-factoid	Hadir Non-Factoid	Tidak Hadir Other	Hadir Other	Entropy Factoid	Entropy Non-Factoid	Entropy Others	Total Entropy	Information Gain
ada	1	1	0	0	4	1	5	0	5	0	0,722	0,000	0,000	0,241	0,759
adon	1	1	0	0	4	1	5	0	5	0	0,722	0,000	0,000	0,241	0,759
agen	1	0	0	1	5	0	5	0	4	1	0,000	0,000	0,722	0,241	0,759
ajar	1	0	0	1	5	0	5	0	4	1	0,000	0,000	0,722	0,241	0,759
akhir	2	0	1	1	5	0	4	1	4	1	0,000	0,722	0,722	0,481	0,519
alas	1	0	1	0	5	0	4	1	5	0	0,000	0,722	0,000	0,241	0,759
alat	1	1	0	0	4	1	5	0	5	0	0,722	0,000	0,000	0,241	0,759
apa	3	0	3	0	5	0	2	3	5	0	0,000	0,971	0,000	0,324	0,676
bagaimana	3	0	2	1	5	0	3	2	4	1	0,000	0,971	0,722	0,564	0,436
baik	1	0	0	1	5	0	5	0	4	1	0,000	0,000	0,722	0,241	0,759
beri	1	1	0	0	4	1	5	0	5	0	0,722	0,000	0,000	0,241	0,759
bisa	1	0	1	0	5	0	4	1	5	0	0,000	0,722	0,000	0,241	0,759
buat	2	0	2	0	5	0	3	2	5	0	0,000	0,971	0,000	0,324	0,676
bukankah	1	0	0	1	5	0	5	0	4	1	0,000	0,000	0,722	0,241	0,759
campur	1	1	0	0	4	1	5	0	5	0	0,722	0,000	0,000	0,241	0,759

cara	1	0	0	1	5	0	5	0	4	1	0,000	0,000	0,722	0,241	0,759
covid-19	1	0	1	0	5	0	4	1	5	0	0,000	0,722	0,000	0,241	0,759
dalam	2	0	2	0	5	0	3	2	5	0	0,000	0,971	0,000	0,324	0,676
dari	2	1	1	0	4	1	4	1	5	0	0,722	0,722	0,000	0,481	0,519
dokter	1	0	1	0	5	0	4	1	5	0	0,000	0,722	0,000	0,241	0,759
gelang	1	1	0	0	4	1	5	0	5	0	0,722	0,000	0,000	0,241	0,759
guna	2	1	1	0	4	1	4	1	5	0	0,722	0,722	0,000	0,481	0,519
gunung	1	0	0	1	5	0	5	0	4	1	0,000	0,000	0,722	0,241	0,759
hadap	1	0	1	0	5	0	4	1	5	0	0,000	0,722	0,000	0,241	0,759
hari	1	1	0	0	4	1	5	0	5	0	0,722	0,000	0,000	0,241	0,759
harus	1	0	0	1	5	0	5	0	4	1	0,000	0,000	0,722	0,241	0,759
ikut	1	0	0	1	5	0	5	0	4	1	0,000	0,000	0,722	0,241	0,759
ingat	2	1	0	1	4	1	5	0	4	1	0,722	0,000	0,722	0,481	0,519
ini	1	0	1	0	5	0	4	1	5	0	0,000	0,722	0,000	0,241	0,759
itu	2	0	2	0	5	0	3	2	5	0	0,000	0,971	0,000	0,324	0,676
jelas	1	0	1	0	5	0	4	1	5	0	0,000	0,722	0,000	0,241	0,759
kamu	2	0	0	2	5	0	5	0	3	2	0,000	0,000	0,971	0,324	0,676
kapan	2	1	0	1	4	1	5	0	4	1	0,722	0,000	0,722	0,481	0,519
kata	1	1	0	0	4	1	5	0	5	0	0,722	0,000	0,000	0,241	0,759
kau	2	0	1	1	5	0	4	1	4	1	0,000	0,722	0,722	0,481	0,519
ke	1	0	0	1	5	0	5	0	4	1	0,000	0,000	0,722	0,241	0,759
keji	1	0	1	0	5	0	4	1	5	0	0,000	0,722	0,000	0,241	0,759
kendala	1	0	1	0	5	0	4	1	5	0	0,000	0,722	0,000	0,241	0,759
kerja	1	0	1	0	5	0	4	1	5	0	0,000	0,722	0,000	0,241	0,759
kesini	1	0	0	1	5	0	5	0	4	1	0,000	0,000	0,722	0,241	0,759
kita	1	0	0	1	5	0	5	0	4	1	0,000	0,000	0,722	0,241	0,759
kue	2	1	1	0	4	1	4	1	5	0	0,722	0,722	0,000	0,481	0,519
kunjung	1	0	0	1	5	0	5	0	4	1	0,000	0,000	0,722	0,241	0,759

laku	1	0	1	0	5	0	4	1	5	0	0,000	0,722	0,000	0,241	0,759
main	1	0	0	1	5	0	5	0	4	1	0,000	0,000	0,722	0,241	0,759
mana	1	1	0	0	4	1	5	0	5	0	0,722	0,000	0,000	0,241	0,759
mengapa	1	0	1	0	5	0	4	1	5	0	0,000	0,722	0,000	0,241	0,759
mixer	1	0	1	0	5	0	4	1	5	0	0,000	0,722	0,000	0,241	0,759
nenek	1	1	0	0	4	1	5	0	5	0	0,722	0,000	0,000	0,241	0,759
pada	1	0	1	0	5	0	4	1	5	0	0,000	0,722	0,000	0,241	0,759
pahlawan	1	1	0	0	4	1	5	0	5	0	0,722	0,000	0,000	0,241	0,759
papar	1	0	1	0	5	0	4	1	5	0	0,000	0,722	0,000	0,241	0,759
proses	1	0	1	0	5	0	4	1	5	0	0,000	0,722	0,000	0,241	0,759
rahasia	1	0	1	0	5	0	4	1	5	0	0,000	0,722	0,000	0,241	0,759
saja	2	0	0	2	5	0	5	0	4	1	0,000	0,000	0,722	0,241	0,759
sedia	1	0	0	1	5	0	5	0	4	1	0,000	0,000	0,722	0,241	0,759
sekarang	1	0	0	1	5	0	5	0	4	1	0,000	0,000	0,722	0,241	0,759
sepakbola	1	0	0	1	5	0	5	0	4	1	0,000	0,000	0,722	0,241	0,759
sepeda	1	0	0	1	5	0	5	0	4	1	0,000	0,000	0,722	0,241	0,759
siapa	4	2	0	2	3	2	5	0	3	2	0,971	0,000	0,971	0,647	0,353
simeulu	1	0	0	1	5	0	5	0	4	1	0,000	0,000	0,722	0,241	0,759
suruh	1	1	0	0	4	1	5	0	5	0	0,722	0,000	0,000	0,241	0,759
tau	1	0	0	1	5	0	5	0	4	1	0,000	0,000	0,722	0,241	0,759
telepon	1	1	0	0	4	1	5	0	5	0	0,722	0,000	0,000	0,241	0,759
temu	1	0	1	0	5	1	5	0	5	0	0,722	0,000	0,000	0,241	0,759
tugas	1	0	0	1	5	0	4	1	5	0	0,000	0,722	0,000	0,241	0,759
turut	1	1	0	0	4	0	5	0	4	1	0,000	0,000	0,722	0,241	0,759
untuk	1	0	1	0	5	1	5	0	5	0	0,722	0,000	0,000	0,241	0,759
virus	4	2	1	1	3	0	4	1	5	0	0,000	0,722	0,000	0,241	0,759
yang	1	0	1	0	5	2	4	1	4	1	0,971	0,722	0,722	0,805	0,195

Langkah 2 : Melakukan melakukan pengurutan bobot nilai pada kata, kemudian melakukan ekstraksi fitur pada bobot kata yang memiliki nilai *threshold* (K) $\geq 0,759$ (diambil 25 fitur kata) menggunakan metode TF-IDF. Hasil dari proses seleksi fitur menggunakan *information gain* ditampilkan pada tabel IV-11.

Tabel IV-11. Hasil Seleksi Fitur *Information Gain*

Hasil Seleksi Fitur		
Data	Hasil	Kelas
D1	alat	Factoid
D2	ada mana	Factoid
D3	suruh	Factoid
D4	hari	Factoid
D5	temu	Factoid
D6	laku	Non-factoid
D7	kendala hadap	Non-factoid
D8	proses kue	Non-factoid
D9	jelas pada mengapa ini	Non-factoid
D10	bisa	Non-factoid
D11	cara	Others
D12	bukankah harus tau	Others

D13	saja agen	Others
D14	kita	Others
D15	saja ke	Others

Kemudian akan dilakukan ekstraksi fitur untuk mengubah teks kedalam bentuk matriks berupa *vector* (nilai) dengan menggunakan metode TF-IDF. Hasil ekstraksi fitur ditampilkan pada tabel IV-12 dan tabel IV-13.

Tabel IV-12. Hasil Ekstraksi Fitur TF-IDF

<i>Term</i>	TF															DF	IDF
	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15		
ada	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2,176091259
agen	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	2,176091259
alat	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2,176091259
baik	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	2,176091259
bisa	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	2,176091259
bukankah	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	2,176091259
laku	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	2,176091259
cara	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	2,176091259
temu	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	2,176091259
hadap	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	2,176091259
hari	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	2,176091259
harus	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	2	1,875061263
ini	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	2,176091259

jelas	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	2,176091259
kita	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	2,176091259
ke	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	2,176091259
kendala	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	2,176091259
mana	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	2,176091259
mengapa	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	2,176091259
pada	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	2,176091259
kue	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	2,176091259
proses	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	2,176091259
saja	0	0	0	0	0	0	0	0	0	0	0	1	0	1	2	1,875061263
suruh	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	2,176091259
tau	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	2,176091259

Tabel IV-16. Hasil Ekstraksi Fitur TF-IDF

TTF-IDF (TF*IDF)																
Term	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	
ada	0	2,176	0	0	0	0	0	0	0	0	0	0	0	0	0	0
agen	0	0	0	0	0	0	0	0	0	0	0	0	2,176	0	0	0
alat	2,176	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
baik	0	0	0	0	0	0	0	0	0	0	0	0	2,176	0	0	0
bisa	0	0	0	0	0	0	0	0	0	2,176	0	0	0	0	0	0
bukankah	0	0	0	0	0	0	0	0	0	0	0	2,176	0	0	0	0
laku	0	0	0	0	0	2,176	0	0	0	0	0	0	0	0	0	0
cara	0	0	0	0	0	0	0	0	0	0	2,176	0	0	0	0	0
temu	0	0	0	0	2,176	0	0	0	0	0	0	0	0	0	0	0
hadap	0	0	0	0	0	0	2,176	0	0	0	0	0	0	0	0	0
hari	0	0	0	2,176	0	0	0	0	0	0	0	0	0	0	0	0
harus	0	0	0	0	0	0	0	0	0	0	0	1,875	0	0	0	1,875

ini	0	0	0	0	0	0	0	0	2,176	0	0	0	0	0	0
jelas	0	0	0	0	0	0	0	0	2,176	0	0	0	0	0	0
kita	0	0	0	0	0	0	0	0	0	0	0	0	0	2,176	0
ke	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2,176
kendala	0	0	0	0	0	0	2,176	0	0	0	0	0	0	0	0
mana	0	2,176	0	0	0	0	0	0	0	0	0	0	0	0	0
mengapa	0	0	0	0	0	0	0	0	2,176	0	0	0	0	0	0
pada	0	0	0	0	0	0	0	0	2,176	0	0	0	0	0	0
kue	0	0	0	0	0	0	0	2,176	0	0	0	0	0	0	0
proses	0	0	0	0	0	0	0	2,176	0	0	0	0	0	0	0
saja	0	0	0	0	0	0	0	0	0	0	0	0	1,875	0	1,875
suruh	0	0	2,176	0	0	0	0	0	0	0	0	0	0	0	0
tau	0	0	0	0	0	0	0	0	0	0	0	2,176	0	0	0

Langkah 3: Melakukan proses pelatihan dan pengujian menggunakan algoritma *Support Vector Machine*.

Pada proses pelatihan dilakukan pencarian nilai *hyperlane* terbaik dengan melakukan perhitungan untuk mendapatkan nilai dari *lagrange multipliers* (α) menggunakan persamaan II-15 dan kemudian mencari nilai bias menggunakan persamaan II-16. Data latih yang digunakan sebagai contoh dalam perhitungan ialah D1 dan D2. Fungsi kernel yang digunakan merupakan kernel *linear* dengan nilai $C=1$. Tahapan proses klasifikasi diuraikan sebagai berikut.

a. Menghitung nilai *dot product* pada data latih.

$$\begin{aligned} x_1^T x_1 &= (0x0) + (0x0) + (2.176x2.176) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) \\ &\quad + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) \\ &\quad + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) \\ &\quad (0x0) \\ &= 4.735 \end{aligned}$$

$$\begin{aligned} x_1^T x_2 &= (0x2.176) + (0x0) + (2.176x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) \\ &\quad + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) \\ &\quad + (0x0) + (0x2.176) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + \\ &\quad (0x0) (0x0) \\ &= 0 \end{aligned}$$

$$\begin{aligned} x_2^T x_1 &= (2.176x0) + (0x0) + (0x2.176) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) \\ &\quad + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) \end{aligned}$$

$$\begin{aligned}
& + (0x0) + (2.176x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + \\
& (0x0) (0x0) \\
& = 0
\end{aligned}$$

$$\begin{aligned}
x_2^T x_2 & = (2.176x2.176) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) \\
& + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) \\
& + (0x0) + (2.176x2.176) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + \\
& (0x0) (0x0) \\
& = 9.47
\end{aligned}$$

b. Menghitung nilai *hyperlane* terbaik (a) menggunakan persamaan II-15 dan fungsi kernel *linear* dengan nilai $C = 1$.

$$L(a) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^n a_i a_j y_i y_j K(x_i, x_j)$$

$$0 = (a_1 + a_2) - \frac{1}{2} \left\{ (a_1 a_1 y_1 y_1 x_1^T x_1) + (a_1 a_2 y_1 y_2 x_1^T x_2) + \right. \\ \left. (a_2 a_1 y_2 y_1 x_2^T x_1) + (a_2 a_2 y_2 y_2 x_2^T x_2) \right\}$$

$$0 = (a_1 + a_2) - \frac{1}{2} \left\{ (a_1 a_1 (4.375)) + (a_1 a_2 (0)) + \right. \\ \left. (a_2 a_1 (0)) + (a_2 a_2 (9.47)) \right\}$$

$$\text{Dengan } \sum_{i=1}^n a_i y_i = 0$$

$$a_1(+1) + a_2(-1) = 0$$

$$a_1 - a_2 = 0$$

$$a_1 = a_2$$

$$0 = (a_1 + a_2) - \frac{1}{2} \left\{ (a_1 a_1 (4.375)) + (a_1 a_2 (0)) + \right. \\ \left. (a_2 a_1 (0)) + (a_2 a_2 (9.47)) \right\}$$

$$0 = (2a_1) - (14.21a_1 a_1)$$

$$14.21a_1 a_1 = 2a_1$$

$$a_1 = 0.141$$

$$\text{Nilai } a_1 = a_2 = 0.141$$

c. Mencari nilai bias melalui persamaan II-16.

$$\begin{aligned} b &= \frac{1}{NSV} \sum_{x_j \in SV} \left(\frac{1}{y_i} - \sum_{x_j \in SV} (\alpha_j y_j K(x_j, x_i)) \right) \\ &= \frac{1}{NSV} \left\{ \left(\frac{1}{y_1} - (a_1 y_1 x_1^T x_1) + (a_2 y_2 x_2^T x_1) \right) + \left(\frac{1}{y_2} - (a_1 y_1 x_1^T x_2) + \right. \right. \\ &\quad \left. \left. (a_2 y_2 x_2^T x_2) \right) \right\} \\ &= \frac{1}{2} \left\{ \left(\frac{1}{1} - (0.141 \times 1 \times 4.735) + (0.141 \times (-1) \times 0) \right) + \left(\frac{1}{-1} - \right. \right. \\ &\quad \left. \left. (0.141 \times 1 \times 0) + (0.141 \times (-1) \times 9.47) \right) \right\} \\ &= 0.334 \end{aligned}$$

Pada proses pelatihan akan dibangun 3 model pelatihan menggunakan metode SVM, dimana untuk setiap model yang dibangun berdasarkan proses pelatihan untuk setiap 2 kelas. Model yang didapatkan pada setiap proses pelatihan akan digunakan sebagai nilai masukan pada proses klasifikasi. Selanjutnya, akan dilakukan proses *voting* untuk menentukan hasil klasifikasi berdasarkan 3 fungsi keputusan dari model yang dibangun menggunakan metode *multi class SVM one-against-one*. Fungsi keputusan ditampilkan pada Tabel IV-14.

Tabel IV-14. Klasifikasi *Multiclass SVM one-against-one*

y = 1	y = -1	Fungsi Keputusan
Kelas Factoid	Kelas Non-factoid	$f_{12}(x) = (w_{12})x + b_{12}$
Kelas Factoid	Kelas Others	$f_{13}(x) = (w_{13})x + b_{13}$
Kelas Non-factoid	Kelas Others	$f_{23}(x) = (w_{23})x + b_{23}$

Setelah mendapatkan nilai a dan b pada proses pelatihan. Maka selanjutnya mengecek kembali x_1 sebagai data uji apakah akan sesuai dengan hasil proses klasifikasi.

1. Menghitung nilai dot product pada data latih dan data uji.

$$\begin{aligned}
 x_1^T x_d &= (0 \times 0) + (0 \times 0) + (2.176 \times 2.176) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) \\
 &\quad + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + \\
 &\quad (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + \\
 &\quad (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) \\
 &= 4.735
 \end{aligned}$$

$$\begin{aligned}
 x_2^T x_d &= (2.176 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + \\
 &\quad (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + \\
 &\quad (0 \times 0) + (0 \times 0) + (0 \times 0) + (2.176 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) \\
 &\quad + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) \\
 &= 0
 \end{aligned}$$

2. Menghitung fungsi pemisah optimal menggunakan persamaan II-15.

$$\begin{aligned}
 f(x) &= \text{sgn}(\sum_{i=1}^{nsv} \alpha_i y_i K(x_i, x_d) + b) \\
 &= \text{sgn}((a_1 y_1 K(x_1, x_d)) + (a_1 y_2 K(x_2, x_d)) + b) \\
 &= \text{sgn}((0.141 \times 1 \times 0) + (0.141 \times (-1) \times 0) + 0.334)
 \end{aligned}$$

$$= 1$$

Proses klasifikasi yang dilakukan pada persamaan diatas untuk data uji (x_d), menunjukkan hasil klasifikasi berupa +1 yaitu kelas factoid berdasarkan label yang diberikan pada tabel fungsi keputusan sebelumnya. Dikarenakan hasil klasifikasi menunjukan kelas factoid maka *vote* untuk kelas factoid akan ditambah satu.

3. Proses klasifikasi menggunakan seleksi fitur *Chi-square* dan algoritma *Support Vector Machine* .

Langkah 1 : Melakukan proses seleksi fitur pada data yang telah di *preprocessing* menggunakan metode *Chi-square*. Hasil pembobotan fitur menggunakan *Chi-square* ditampilkan pada tabel IV-15.

Tabel IV-15. Perhitungan Bobot Nilai *Chi-square*

Perhitungan Chi-square (Total Data = 15)																	
Term	Data	Factoid				Non-factoid				Other				Factoid $X^2(t,R)$	Non-factoid $X^2(t,R)$	Other $X^2(t,R)$	Max Term
		A	B	C	D	A	B	C	D	A	B	C	D				
	15																
ada	1	1	0	4	10	0	1	5	9	0	1	5	9	2,143	0,536	0,536	2,143
adon	1	1	0	4	10	0	1	5	9	0	1	5	9	2,143	0,536	0,536	2,143
agen	1	0	1	5	9	0	1	5	9	1	0	4	10	0,536	0,536	2,143	2,143
ajar	1	0	1	5	9	0	1	5	9	1	0	4	10	0,536	0,536	2,143	2,143
akhir	2	0	2	5	8	1	1	4	9	1	1	4	9	1,154	0,288	0,288	0,288
alas	1	0	1	5	9	1	0	4	10	0	1	5	9	0,536	2,143	0,536	2,143
alat	1	0	1	5	10	0	1	5	9	0	1	5	9	0,455	0,536	0,536	0,536
apa	3	0	3	5	7	3	0	2	10	0	3	5	7	1,875	7,500	1,875	7,500
bagaimana	3	0	3	5	7	2	1	3	9	1	2	4	8	1,875	1,875	0,000	1,875
baik	1	0	1	5	9	0	1	5	9	1	0	4	10	0,536	0,536	0,000	0,000
beri	1	1	0	4	10	0	1	5	9	0	1	5	9	2,143	0,536	0,536	2,143
bisa	1	0	1	5	9	1	0	4	10	0	1	5	9	0,536	2,143	0,536	2,143
buat	2	0	2	5	8	2	0	3	10	0	2	5	8	1,154	4,615	1,154	4,615
bukankah	1	0	1	5	9	0	1	5	9	1	0	4	10	0,536	0,536	2,143	2,143
campur	1	1	0	4	10	0	1	5	9	0	1	5	9	2,143	0,536	0,536	2,143
cara	1	0	1	5	9	0	1	5	9	1	0	4	10	0,536	0,536	2,143	2,143
covid-19	1	0	1	5	9	1	0	4	10	0	1	5	9	0,536	2,143	0,536	2,143
dalam	2	0	2	5	8	2	0	3	10	0	2	5	8	1,154	4,615	1,154	4,615
dari	2	1	1	4	9	1	1	4	9	0	2	5	8	0,288	0,288	1,154	1,154
dokter	1	0	1	5	9	1	0	4	10	0	1	5	9	0,536	2,143	0,536	2,143

gelang	1	1	0	4	10	0	1	5	9	0	1	5	9	2,143	0,536	0,536	2,143
guna	2	1	1	4	9	1	1	4	9	0	2	5	8	0,288	0,288	1,154	1,154
gunung	1	0	1	5	9	0	1	5	9	1	0	4	10	0,536	0,536	2,143	2,143
hadap	1	0	1	5	9	1	0	4	10	0	1	5	9	0,536	2,143	0,536	2,143
hari	1	1	0	4	10	0	1	5	9	0	1	5	9	2,143	0,536	0,536	2,143
harus	1	0	1	5	9	0	1	5	9	1	0	4	10	0,536	0,536	2,143	2,143
ikut	1	0	1	5	9	0	1	5	9	1	0	4	10	0,536	0,536	2,143	2,143
ingat	2	1	1	4	9	0	2	5	8	1	1	4	9	0,288	1,154	0,288	1,154
ini	1	0	1	5	9	1	0	4	10	0	1	5	9	0,536	2,143	0,536	2,143
itu	2	0	2	5	8	2	0	3	10	0	2	5	8	1,154	4,615	1,154	4,615
jelas	1	0	1	5	9	1	0	4	10	0	1	5	9	0,536	2,143	0,536	2,143
kamu	2	0	2	5	8	0	2	5	8	2	0	3	10	1,154	1,154	4,615	4,615
kapan	2	1	1	4	9	0	2	5	8	1	1	4	9	0,288	1,154	0,288	1,154
kata	1	1	0	4	10	0	1	5	9	0	1	5	9	2,143	0,536	0,536	2,143
kau	2	0	2	5	8	1	1	4	9	1	1	4	9	1,154	0,288	0,288	1,154
ke	1	0	1	5	9	0	1	5	9	1	0	4	10	0,536	0,536	2,143	2,143
keji	1	0	1	5	9	1	0	4	10	0	1	5	9	0,536	2,143	0,536	2,143
kendala	1	0	1	5	9	1	0	4	10	0	1	5	9	0,536	2,143	0,536	2,143
kerja	1	0	1	5	9	1	0	4	10	0	1	5	9	0,536	2,143	0,536	2,143
kesini	1	0	1	5	9	0	1	5	9	1	0	4	10	0,536	0,536	2,143	2,143
kita	1	0	1	5	9	0	1	5	9	1	0	4	10	0,536	0,536	2,143	2,143
kue	2	1	1	4	9	1	1	4	9	0	2	5	8	0,288	0,288	1,154	1,154
kunjung	1	0	1	5	9	0	1	5	9	1	0	4	10	0,536	0,536	2,143	2,143
laku	1	0	1	5	9	1	0	4	10	0	1	5	9	0,536	2,143	0,536	2,143
main	1	0	1	5	9	0	1	5	9	1	0	4	10	0,536	0,536	2,143	2,143
mana	1	1	0	4	10	0	1	5	9	0	1	5	9	2,143	0,536	0,536	2,143
mengapa	1	0	1	5	9	1	0	4	10	0	1	5	9	0,536	2,143	0,536	2,143
mixer	1	0	1	5	9	1	0	4	10	0	1	5	9	0,536	2,143	0,536	0,536
nenek	1	1	0	4	10	0	1	5	9	0	1	5	9	2,143	0,536	0,536	2,143
pada	1	0	1	5	9	1	0	4	10	0	1	5	9	0,536	2,143	0,536	2,143
pahlawan	1	1	0	4	10	0	1	5	9	0	1	5	9	2,143	0,536	0,536	2,143
papar	1	0	1	5	9	1	0	4	10	0	1	5	9	0,536	2,143	0,536	2,143
proses	1	0	1	5	9	1	0	4	10	0	1	5	9	0,536	2,143	0,536	0,536
rahasia	1	0	1	5	9	1	0	4	10	0	1	5	9	0,536	2,143	0,536	2,143

saja	1	0	1	5	9	0	1	5	9	1	0	4	10	0,536	0,536	2,143	2,143
sedia	1	0	1	5	9	0	1	5	9	1	0	4	10	0,536	0,536	2,143	2,143
sekarang	1	0	1	5	9	0	1	5	9	1	0	4	10	0,536	0,536	2,143	2,143
sepakbola	1	0	1	5	9	0	1	5	9	1	0	4	10	0,536	0,536	2,143	2,143
sepeda	1	0	1	5	9	0	1	5	9	1	0	4	10	0,536	0,536	2,143	2,143
siapa	4	2	2	3	8	0	4	5	6	2	2	3	8	0,682	2,727	0,682	2,727
simeulu	1	0	1	5	9	0	1	5	9	1	0	4	10	0,536	0,536	2,143	2,143
suruh	1	1	0	4	10	0	1	5	9	0	1	5	9	2,143	0,536	0,536	2,143
tau	1	0	1	5	9	0	1	5	9	1	0	4	10	0,536	0,536	2,143	2,143
telepon	1	1	0	4	10	0	1	5	9	0	1	5	9	2,143	0,536	0,536	2,143
temu	1	1	0	4	10	0	1	5	9	0	1	5	9	2,143	0,536	0,536	2,143
tugas	1	0	1	5	9	1	0	4	10	0	1	5	9	0,536	2,143	0,536	2,143
turut	1	0	1	5	9	0	1	5	9	1	0	4	10	0,536	0,536	2,143	2,143
untuk	1	1	0	4	10	0	1	5	9	0	1	5	9	2,143	0,536	0,536	2,143
virus	1	0	1	5	9	1	0	4	10	0	1	5	9	0,536	2,143	0,536	2,143
yang	4	2	2	3	8	1	2	4	8	1	2	3	8	0,682	0,000	0,045	0,682

Langkah 2 : Melakukan melakukan pengurutan bobot nilai pada kata, kemudian melakukan ekstraksi fitur pada bobot kata yang memiliki nilai *threshold* (K) $\geq 2,143$ (diambil 25 fitur kata) menggunakan metode TF-IDF. Hasil dari proses seleksi fitur menggunakan *Chi-square* ditampilkan pada tabel IV-16.

Tabel IV-16. Hasil Seleksi Fitur *Chi-square*.

Hasil Seleksi Fitur		K=15
Data	Hasil	Kelas
D1	“alat”, “untuk”	Factoid
D2	“ada”, “mana”, “dari”	Factoid
D3	“kata”, “siapa”	Factoid
D4	“hari”	Factoid
D5	“siapa”, “temu”	Factoid
D6	“laku”, “buat”	Non-factoid
D7	“kendala”, “dalam”	Non-factoid
D8	“dalam”, “buat”	Non-factoid
D9	“jelas”, “kau”, “rahasia”, “dari”	Non-factoid
D10	“Bagaimana”, “itu”, “bisa”	Non-factoid
D11	“sedia”, “bagaimana”, “main”	Others
D12	“bagaimana”	Others
D13	“turut”, “siapa”, “baik”	Others
D14	“kau”, “kita”, “kesini”	Others
D15	“siapa”, “ikut”	Others

Kemudian akan dilakukan ekstraksi fitur untuk mengubah teks kedalam bentuk matriks berupa vector (nilai) dengan menggunakan metode TF-IDF. Hasil ekstraksi fitur ditampilkan pada tabel IV-17 dan tabel IV-18.

Tabel IV-17. Hasil Ekstraksi Fitur TF-IDF

<i>Term</i>	TF															DF	IDF
	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15		
ada	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2,176091259
alat	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2,176091259
bagaimana	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	2	1,875061263
baik	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	2,176091259
bisa	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	2,176091259
dalam	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	2	1,875061263
dari	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	2	1,875061263
hari	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	2,176091259
ikut	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	2,176091259
itu	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	2	1,875061263
jelas	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	2,176091259
kata	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	2,176091259
kau	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	2	1,875061263
kesini	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	2,176091259
kendala	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	2,176091259
kita	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	2,176091259
laku	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	2,176091259
main	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	2	1,875061263
rahasia	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	2,176091259
sedia	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	2,176091259
siapa	0	0	1	0	1	0	0	0	0	0	0	0	1	0	1	4	1,574031268
buat	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	2	1,875061263
temu	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	2,176091259
turut	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	2,176091259

Langkah 3: Melakukan proses pelatihan dan pengujian menggunakan algoritma *Support Vector Machine*.

Pada proses pelatihan dilakukan pencarian nilai *hyperlane* terbaik dengan melakukan perhitungan untuk mendapatkan nilai dari *lagrange multipliers* (α) menggunakan persamaan II-15 dan kemudian mencari nilai bias menggunakan persamaan II-16. Data latih yang digunakan sebagai contoh dalam perhitungan ialah D1 dan D2. Fungsi kernel yang digunakan merupakan kernel *linear* dengan nilai $C=1$. Tahapan proses klasifikasi diuraikan sebagai berikut.

a. Menghitung nilai *dot product* pada data latih.

$$\begin{aligned} x_1^T x_1 &= (0 \times 0) + (2.176 \times 2.176) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) \\ &\quad + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) \\ &\quad + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + \\ &\quad (2.176 \times 2.176) \\ &= 9.47 \end{aligned}$$

$$\begin{aligned} x_1^T x_2 &= (0 \times 2.176) + (2.176 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 1.875) + \\ &\quad (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + \\ &\quad (0 \times 0) + (0 \times 0) + (0 \times 1.875) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) \\ &\quad + (0 \times 0) + (2.176 \times 0) \\ &= 0 \end{aligned}$$

$$\begin{aligned} x_2^T x_1 &= (2.176 \times 0) + (0 \times 2.176) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (1.875 \times 0) + \\ &\quad (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + \\ &\quad (0 \times 0) + (0 \times 0) + (1.875 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) \\ &\quad + (0 \times 0) + (0 \times 2.176) \end{aligned}$$

$$\begin{aligned}
&= 0 \\
x_2^T x_2 &= (2.176 \times 2.176) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (1.875 \times 1.875) \\
&\quad + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) \\
&\quad + (0 \times 0) + (0 \times 0) + (1.875 \times 1.875) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) \\
&\quad + (0 \times 0) + (0 \times 0) + (0 \times 0) \\
&= 11.76
\end{aligned}$$

b. Menghitung nilai *hyperlane* terbaik (a) menggunakan persamaan II-15 dan fungsi kernel *linear* dengan nilai $C = 1$.

$$\begin{aligned}
L(a) &= \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j K(x_i, x_j) \\
0 &= (a_1 + a_2) - \frac{1}{2} \left\{ (a_1 a_1 y_1 y_1 x_1^T x_1) + (a_1 a_2 y_1 y_2 x_1^T x_2) + \right. \\
&\quad \left. (a_2 a_1 y_2 y_1 x_2^T x_1) + (a_2 a_2 y_2 y_2 x_2^T x_2) \right\} \\
0 &= (a_1 + a_2) - \frac{1}{2} \left\{ (a_1 a_1 (9.47)) + (a_1 a_2 (0)) + \right. \\
&\quad \left. (a_2 a_1 (0)) + (a_2 a_2 (11.77)) \right\}
\end{aligned}$$

$$\text{Dengan } \sum_{i=1}^n \alpha_i y_i = 0$$

$$a_1(+1) + a_2(-1) = 0$$

$$a_1 - a_2 = 0$$

$$a_1 = a_2$$

$$0 = (a_1 + a_2) - \frac{1}{2} \left\{ (a_1 a_1 (9.47)) + (a_1 a_2 (0)) + \right. \\ \left. (a_2 a_1 (0)) + (a_2 a_2 (11.77)) \right\}$$

$$0 = (2a_1) - (21.24a_1 a_1)$$

$$21.24a_1 a_1 = 2a_1$$

$$a_1 = 0.094$$

$$\text{Nilai } a_1 = a_2 = 0.094$$

c. Mencari nilai bias melalui persamaan II-16.

$$\begin{aligned}
b &= \frac{1}{NSV} \sum_{x_j \in SV} \left(\frac{1}{y_i} - \sum_{x_j \in SV} (a_j y_j K(x_j, x_i)) \right) \\
&= \frac{1}{NSV} \left\{ \left(\frac{1}{y_1} - (a_1 y_1 x_1^T x_1) + (a_2 y_2 x_2^T x_1) \right) + \left(\frac{1}{y_2} - (a_1 y_1 x_1^T x_2) + \right. \right. \\
&\quad \left. \left. (a_2 y_2 x_2^T x_2) \right) \right\} \\
&= \frac{1}{2} \left\{ \left(\frac{1}{1} - (0.094 \times 1 \times 9.47) + (0.094 \times (-1) \times 0) \right) + \left(\frac{1}{-1} - \right. \right. \\
&\quad \left. \left. (0.094 \times 1 \times 0) + (0.094 \times (-1) \times 11.77) \right) \right\} \\
&= 0.107
\end{aligned}$$

Pada proses pelatihan akan dibangun 3 model pelatihan menggunakan metode SVM, dimana untuk setiap model yang dibangun berdasarkan proses pelatihan untuk setiap 2 kelas. Model yang didapatkan pada setiap proses pelatihan akan digunakan sebagai nilai masukan pada proses klasifikasi. Selanjutnya, akan dilakukan proses *voting* untuk menentukan hasil klasifikasi berdasarkan 3 fungsi keputusan yang dibangun menggunakan metode *multi class SVM one-against-one*. Fungsi keputusan ditampilkan pada Tabel IV-19.

Tabel IV-19. Klasifikasi *Multiclass SVM one-against-one*

$y = 1$	$y = -1$	Fungsi Keputusan
Kelas Factoid	Kelas Non-factoid	$f_{12}(x) = (w_{12})x + b_{12}$
Kelas Factoid	Kelas Others	$f_{13}(x) = (w_{13})x + b_{13}$
Kelas Non-factoid	Kelas Others	$f_{23}(x) = (w_{23})x + b_{23}$

Setelah mendapatkan nilai a dan b pada proses pelatihan. Maka selanjutnya mengecek kembali x_1 sebagai data uji apakah akan sesuai dengan hasil proses klasifikasi.

1. Menghitung nilai dot product pada data latih dan data uji.

$$\begin{aligned}
 x_1^T x_d &= (0 \times 0) + (2.176 \times 2.176) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) \\
 &\quad + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + \\
 &\quad (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + \\
 &\quad (0 \times 0) + (0 \times 0) + (0 \times 0) + (2.176 \times 2.176) \\
 &= 4.735
 \end{aligned}$$

$$\begin{aligned}
 x_2^T x_d &= (2.176 \times 0) + (0 \times 2.176) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + \\
 &\quad (1.875 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) \\
 &\quad + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (1.875 \times 0) + (0 \times 0) + \\
 &\quad (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 2.176) \\
 &= 0
 \end{aligned}$$

2. Menghitung fungsi pemisah optimal menggunakan persamaan II-15.

$$\begin{aligned}
 f(x) &= \text{sgn}(\sum_{i=1}^{n_{sv}} \alpha_i y_i K(x_i, x_d) + b) \\
 &= \text{sgn}((a_1 y_1 K(x_1, x_d)) + (a_1 y_2 K(x_2, x_d)) + b) \\
 &= \text{sgn}((0.107 \times 1 \times 4.735) + (0.107 \times (-1) \times 0) + \\
 &\quad 0.017) \\
 &= 1
 \end{aligned}$$

Pada proses klasifikasi yang dilakukan pada persamaan diatas untuk data uji (x_d), menunjukkan hasil klasifikasi berupa +1 yaitu kelas factoid berdasarkan label yang diberikan pada tabel fungsi keputusan sebelumnya. Dikarenakan hasil klasifikasi menunjukkan kelas factoid maka *vote* untuk kelas factoid akan ditambah satu.

4. Proses klasifikasi menggunakan seleksi fitur *Mutual Information* dan algoritma *Support Vector Machine* .

Langkah 1 : Melakukan proses seleksi fitur pada data yang telah di *preprocessing* menggunakan metode *Mutual Information*.

Hasil pembobotan menggunakan *Mutual Information* ditampilkan pada tabel IV-20.

Tabel IV-20. Perhitungan Bobot Nilai *Mutual Information*

Perhitungan Mutual Information (Total Data = 15)																	
Term	Data	Menentukan Panjang Kelas												Perhitungan Manual MI			Nilai MI Max Term
		Factoid				Non-factoid				Other				Factoid	Non-factoid	Other	
	15	N11	N01	N10	N00	N11	N01	N10	N00	N11	N01	N10	N00				
Ada	1	1	24	0	67	0	35	1	56	0	32	1	59	0,021	-0,003	0,007	0,021
Adon	1	1	24	0	67	0	35	1	56	0	32	1	59	0,021	-0,003	0,007	0,021
Agen	1	0	25	1	66	0	35	1	56	1	31	0	60	0,005	-0,003	0,017	0,017
Ajar	1	0	25	1	66	0	35	1	56	1	31	0	60	0,005	-0,003	0,017	0,017
Akhir	2	0	25	2	65	1	34	1	56	1	31	1	59	0,010	0,001	0,002	0,010
Alas	1	0	25	1	66	1	34	0	57	0	32	1	59	0,005	0,015	0,007	0,015
Alat	1	1	24	0	67	0	35	1	56	0	32	1	59	0,021	-0,003	0,007	0,021
Apa	3	1	24	2	65	2	33	1	56	0	32	3	57	0,000	0,015	0,021	0,021
Bagaimana	3	0	25	3	64	1	34	2	55	2	30	1	59	0,015	-0,009	0,010	0,015
Baik	1	0	25	1	66	0	35	1	56	1	31	0	60	0,005	-0,003	0,017	0,017
Beri	1	1	24	0	67	0	35	1	56	0	32	1	59	0,021	-0,003	0,007	0,021
Bisa	1	0	25	1	66	1	34	0	57	0	32	1	59	0,005	0,015	0,007	0,015
Buat	2	0	25	2	65	2	33	0	57	0	32	2	58	0,010	0,031	0,014	0,031
Bukankah	1	0	25	0	67	0	35	0	57	0	32	0	60	0,000	0,000	0,000	0,000
Campur	1	1	24	0	67	0	35	1	56	0	32	1	59	0,021	-0,003	0,007	0,021
Cara	1	0	25	1	66	0	35	1	56	1	31	0	60	0,005	-0,003	0,017	0,017
Covid-19	1	0	25	1	66	1	34	0	57	0	32	1	59	0,005	0,015	0,007	0,015
Dalam	2	0	25	2	65	2	33	0	57	0	32	2	58	0,010	0,031	0,014	0,031
Dari	2	1	24	1	66	1	34	1	56	0	32	2	58	0,004	0,001	0,014	0,014

Dokter	1	0	25	1	66	1	34	0	57	0	32	1	59	0,005	0,015	0,007	0,015
Gelang	1	1	24	0	67	0	35	1	56	0	32	1	59	0,021	-0,003	0,007	0,021
Guna	2	1	24	0	67	0	35	1	56	0	32	1	59	0,021	-0,003	0,007	0,021
Gunung	1	0	25	0	66	0	35	0	57	0	32	0	60	0,016	0,000	0,000	0,016
Hadap	1	0	25	1	66	1	34	0	57	0	32	1	59	0,005	0,015	0,007	0,015
Hari	1	0	25	0	67	0	35	0	57	0	32	0	60	0,000	0,000	0,000	0,000
Harus	1	0	25	0	67	0	35	0	57	0	32	0	60	0,000	0,000	0,000	0,000
Ikut	1	0	25	0	67	0	35	0	57	0	32	0	60	0,000	0,000	0,000	0,000
Ingat	2	1	24	1	66	0	35	2	55	1	31	1	59	0,004	-0,007	0,002	0,004
Ini	1	0	25	0	67	0	35	0	57	0	32	0	60	0,000	0,000	0,000	0,000
Itu	2	0	25	2	65	2	33	0	57	0	32	2	58	0,010	0,031	0,014	0,031
Jelas	1	0	25	1	66	1	34	0	57	0	32	1	59	0,005	0,015	0,007	0,015
Kamu	2	0	25	2	65	0	35	2	55	2	30	0	60	0,010	-0,007	0,034	0,034
Kapan	2	1	24	1	66	0	35	2	55	1	31	1	59	0,004	-0,007	0,002	0,004
Kata	1	1	24	0	67	0	35	1	56	0	32	1	59	0,021	-0,003	0,007	0,021
Kau	2	0	25	2	65	1	34	1	56	1	31	1	59	0,010	0,001	0,002	0,010
Ke	1	0	25	0	66	0	35	0	57	0	32	0	60	0,016	0,000	0,000	0,016
Keji	1	0	25	1	66	1	34	0	57	0	32	1	59	0,005	0,015	0,007	0,015
Kendala	1	0	25	1	66	1	34	0	57	0	32	1	59	0,005	0,015	0,007	0,015
Kerja	1	0	25	1	66	1	34	0	57	0	32	1	59	0,005	0,015	0,007	0,015
Kesini	1	0	25	1	66	0	35	1	56	1	31	0	60	0,005	-0,003	0,017	0,017
Kita	1	0	25	1	66	0	35	1	56	1	31	0	60	0,005	-0,003	0,017	0,017
Kue	2	1	24	1	66	1	34	1	56	0	32	2	58	0,004	0,001	0,014	0,014
Kunjung	1	0	25	1	66	0	35	1	56	1	31	0	60	0,005	-0,003	0,017	0,017
Laku	1	0	25	1	66	1	34	0	57	0	32	1	59	0,005	0,015	0,007	0,015
Main	1	0	25	1	66	0	35	1	56	1	31	0	60	0,005	-0,003	0,017	0,017
Mana	1	1	24	0	67	0	35	1	56	0	32	1	59	0,021	-0,003	0,007	0,021
Mengapa	1	0	25	1	66	1	34	0	57	0	32	1	59	0,005	0,015	0,007	0,015
Mixer	1	0	25	1	66	1	34	0	57	0	32	1	59	0,005	0,015	0,007	0,015
Nenek	1	1	24	0	67	0	35	1	56	0	32	1	59	0,021	-0,003	0,007	0,021
Pada	1	0	25	1	66	1	34	0	57	0	32	1	59	0,005	0,015	0,007	0,015
Pahlawan	1	1	24	0	67	0	35	1	56	0	32	1	59	0,021	-0,003	0,007	0,021
Papar	1	0	25	0	67	0	35	0	57	0	32	0	60	0,000	0,000	0,000	0,000
Proses	1	0	25	1	66	1	34	0	57	0	32	1	59	0,005	0,015	0,007	0,015

Rahasia	1	0	25	1	66	1	34	0	57	0	32	1	59	0,005	0,015	0,007	0,015
Saja	1	0	25	0	66	0	35	0	57	0	32	0	60	0,016	0,000	0,000	0,016
Sedia	1	0	25	1	66	0	35	1	56	1	31	0	60	0,005	-0,003	0,017	0,017
Sekarang	1	0	25	1	66	0	35	1	56	1	31	0	60	0,005	-0,003	0,017	0,017
Sepakbola	1	0	25	1	66	0	35	1	56	1	31	0	60	0,005	-0,003	0,017	0,017
Sepeda	1	0	25	1	66	0	35	1	56	1	31	0	60	0,005	-0,003	0,017	0,017
Siapa	4	2	23	1	65	0	35	2	55	1	31	1	59	0,032	-0,007	0,002	0,032
Simeulu	1	0	25	0	66	0	35	0	57	0	32	0	60	0,016	0,000	0,000	0,016
Suruh	1	1	24	0	67	0	35	1	56	0	32	1	59	0,021	-0,003	0,007	0,021
Tau	1	0	25	1	66	0	35	1	56	1	31	0	60	0,005	-0,003	0,017	0,017
Telepon	1	1	24	0	67	0	35	1	56	0	32	1	59	0,021	-0,003	0,007	0,021
Temu	1	1	24	0	67	0	35	1	56	0	32	1	59	0,021	-0,003	0,007	0,021
Tugas	1	0	25	0	67	0	35	0	57	0	32	0	60	0,000	0,000	0,000	0,000
Turut	1	0	25	1	66	0	35	1	56	1	31	0	60	0,005	-0,003	0,017	0,017
Untuk	1	1	24	0	67	0	35	1	56	0	32	1	59	0,021	-0,003	0,007	0,021
Virus	1	0	25	1	66	1	34	0	57	0	32	1	59	0,005	0,015	0,007	0,015
Yang	4	2	23	2	65	0	35	1	56	0	32	1	59	0,008	-0,003	0,007	0,008

Langkah 2 : Melakukan melakukan pengurutan bobot nilai pada kata, kemudian melakukan ekstraksi fitur pada bobot kata yang memiliki nilai *threshold* (K) $\geq 0,017$ (diambil 25 fitur kata) menggunakan metode TF-IDF. Hasil dari proses seleksi fitur menggunakan *Mutual Information* ditampilkan pada tabel IV-21.

Tabel IV-21. Hasil Seleksi Fitur *Mutual Information*

Hasil Seleksi Fitur		K=15
Data	Hasil	Kelas
D1	“alat”, “untuk”, “campur”, “adon”	Factoid
D2	“ada”, “mana”, “gelang”, “beri”, “nenek”	Factoid
D3	“kata”, “siapa”, “suruh”	Factoid
D4	, “pahlawan”	Factoid
D5	“siapa”, “temu”, “telepon”	Factoid
D6	“apa”, “buat”, “itu”	Non-factoid
D7	“apa”, “dalam”	Non-factoid
D8	“apa”, “guna”, “dalam”, “buat”	Non-factoid
D9	“jelas”	Non-factoid
D10	“itu”,	Non-factoid
D11	“kamu”, “ajar”	Others
D12	“kamu”	Others
D13	“siapa”, “agen”	Others
D14	“kita”	Others
D15	“siapa”	Others

Kemudian akan dilakukan ekstraksi fitur untuk mengubah teks kedalam bentuk matriks berupa vector (nilai) dengan menggunakan metode TF-IDF. Hasil ekstraksi fitur ditampilkan pada tabel IV-22 dan tabel IV-23.

Tabel IV-22. Hasil Ekstraksi Fitur TF-IDF

<i>Term</i>	TF															DF	IDF
	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15		
Ada	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2,176091259
Adon	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2,176091259
Agen	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	2,176091259
Ajar	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	2,176091259
Alat	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2,176091259
Apa	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	3	1,698970004
Beri	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2,176091259
Buat	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	2	1,875061263
Campur	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2,176091259
Dalam	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	2,176091259
Gelang	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2,176091259
Guna	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	2,176091259
Itu	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	2	1,875061263
Jelas	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	2,176091259
Kamu	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	2	1,875061263
Kata	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	2,176091259
Kita	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	2,176091259
Mana	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2,176091259
Nenek	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2,176091259
Pahlawan	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	2,176091259

Siapa	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	1	4	1,574031268
Suruh	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2,176091259
Telepon	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	2,176091259
Temu	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	2,176091259
Untuk	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2,176091259

Tabel IV-23. Hasil Ekstraksi Fitur TF-IDF

TTF-IDF (TF*IDF)																
Term	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	
Ada	0	2,176	0	0	0	0	0	0	0	0	0	0	0	0	0	
Adon	2,176	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Agen	0	0	0	0	0	0	0	0	0	0	0	0	2,176	0	0	
Ajar	0	0	0	0	0	0	0	0	0	0	2,176	0	0	0	0	
Alat	2,176	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Apa	0	0	0	0	0	1,699	1,699	1,699	0	0	0	0	0	0	0	
Beri	0	2,176	0	0	0	0	0	0	0	0	0	0	0	0	0	
Buat	0	0	0	0	0	1,875	0	1,875	0	0	0	0	0	0	0	
Campur	2,176	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Dalam	0	0	0	0	0	0	2,176	0	0	0	0	0	0	0	0	
Gelang	0	2,176	0	0	0	0	0	0	0	0	0	0	0	0	0	
Guna	0	0	0	0	0	0	0	2,176	0	0	0	0	0	0	0	
Itu	0	0	0	0	0	1,875	0	0	0	1,875	0	0	0	0	0	
Jelas	0	0	0	0	0	0	0	0	2,176	0	0	0	0	0	0	
Kamu	0	0	0	0	0	0	0	0	0	0	1,875	1,875	0	0	0	
Kata	0	0	2,176	0	0	0	0	0	0	0	0	0	0	0	0	
Kita	0	0	0	0	0	0	0	0	0	0	0	0	0	2,176	0	
Mana	0	2,176	0	0	0	0	0	0	0	0	0	0	0	0	0	
Nenek	0	2,176	0	0	0	0	0	0	0	0	0	0	0	0	0	
Pahlawan	0	0	0	2,176	0	0	0	0	0	0	0	0	0	0	0	
Siapa	0	0	1,574	0	1,574	0	0	0	0	0	0	0	1,574	0	1,574	

Langkah 3: Melakukan proses pelatihan dan pengujian menggunakan algoritma *Support Vector Machine*.

Pada proses pelatihan dilakukan pencarian nilai *hyperlane* terbaik dengan melakukan perhitungan untuk mendapatkan nilai dari *lagrange multipliers* (α) menggunakan persamaan II-15 dan kemudian mencari nilai bias menggunakan persamaan II-16. Data latih yang digunakan sebagai contoh dalam perhitungan ialah D1 dan D2. Fungsi kernel yang digunakan merupakan kernel linear dengan nilai $C=1$. Tahapan proses klasifikasi diuraikan sebagai berikut.

a. Menghitung nilai *dot product* pada data latih.

$$\begin{aligned} x_1^T x_1 &= (0 \times 0) + (2.176 \times 2.176) + (0 \times 0) + (0 \times 0) + (2.176 \times 2.176) + (0 \times 0) + (0 \times 0) \\ &\quad + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) \\ &\quad + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) \\ &\quad + (0 \times 0) + (2.176 \times 2.176) \\ &= 14.205 \end{aligned}$$

$$\begin{aligned} x_1^T x_2 &= (0 \times 2.176) + (2.176 \times 0) + (0 \times 0) + (0 \times 0) + (2.176 \times 0) + (0 \times 0) + (0 \times 1.875) \\ &\quad + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) \\ &\quad + (0 \times 0) + (0 \times 0) + (0 \times 1.875) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) \\ &\quad + (0 \times 0) + (2.176 \times 0) \\ &= 0 \end{aligned}$$

$$\begin{aligned} x_2^T x_1 &= (2.176 \times 0) + (0 \times 2.176) + (0 \times 0) + (0 \times 0) + (0 \times 2.176) + (0 \times 0) + (1.875 \times 0) \\ &\quad + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) \\ &\quad + (0 \times 0) + (0 \times 0) + (1.875 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) \\ &\quad + (0 \times 0) + (0 \times 2.176) \end{aligned}$$

$$= 0$$

$$\begin{aligned} x_2^T x_2 &= (2.176 \times 2.176) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (1.875 \times 1.875) \\ &+ (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + \\ &(0 \times 0) + (0 \times 0) + (1.875 \times 1.875) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + \\ &(0 \times 0) + (0 \times 0) + (0 \times 0) \\ &= 8.251 \end{aligned}$$

b. Menghitung nilai *hyperlane* terbaik (a) menggunakan persamaan II-15 dan fungsi kernel *linear* dengan nilai $C = 1$.

$$L(a) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

$$0 = (a_1 + a_2) - \frac{1}{2} \left\{ (a_1 a_1 y_1 y_1 x_1^T x_1) + (a_1 a_2 y_1 y_2 x_1^T x_2) + \right. \\ \left. (a_2 a_1 y_2 y_1 x_2^T x_1) + (a_2 a_2 y_2 y_2 x_2^T x_2) \right\}$$

$$0 = (a_1 + a_2) - \frac{1}{2} \left\{ (a_1 a_1 (14.21)) + (a_1 a_2 (0)) + \right. \\ \left. (a_2 a_1 (0)) + (a_2 a_2 (8.251)) \right\}$$

Dengan $\sum_{i=1}^n \alpha_i y_i = 0$

$$a_1(+1) + a_2(-1) = 0$$

$$a_1 - a_2 = 0$$

$$a_1 = a_2$$

$$0 = (a_1 + a_2) - \frac{1}{2} \left\{ (a_1 a_1 (14.21)) + (a_1 a_2 (0)) + \right. \\ \left. (a_2 a_1 (0)) + (a_2 a_2 (8.251)) \right\}$$

$$0 = (2a_1) - (27.1a_1 a_1)$$

$$27.1a_1 a_1 = 2a_1$$

$$a_1 = 0.074$$

Nilai $a_1 = a_2 = 0.074$

c. Mencari nilai bias melalui persamaan II-16.

$$\begin{aligned}
b &= \frac{1}{NSV} \sum_{x_j \in SV} \left(\frac{1}{y_i} - \sum_{x_j \in SV} (a_j y_j K(x_j, x_i)) \right) \\
&= \frac{1}{NSV} \left\{ \left(\frac{1}{y_1} - (a_1 y_1 x_1^T x_1) + (a_2 y_2 x_2^T x_1) \right) + \left(\frac{1}{y_2} - (a_1 y_1 x_1^T x_2) + \right. \right. \\
&\quad \left. \left. (a_2 y_2 x_2^T x_2) \right) \right\} \\
&= \frac{1}{2} \left\{ \left(\frac{1}{1} - (0.074 \times 1 \times 14.052) + (0.074 \times (-1) \times 0) \right) + \left(\frac{1}{-1} - \right. \right. \\
&\quad \left. \left. (0.074 \times 1 \times 0) + (0.074 \times (-1) \times 8.251) \right) \right\} \\
&= -0,22
\end{aligned}$$

Pada proses pelatihan akan dibangun 3 model pelatihan menggunakan metode SVM, dimana untuk setiap model yang dibangun berdasarkan proses pelatihan untuk setiap 2 kelas. Model yang didapatkan pada setiap proses pelatihan akan digunakan sebagai nilai masukan pada proses klasifikasi. Selanjutnya, akan dilakukan proses *voting* untuk menentukan hasil klasifikasi berdasarkan 3 fungsi keputusan yang dibangun menggunakan metode *multi class SVM one-against-one*. Fungsi keputusan ditampilkan pada Tabel IV-24.

Tabel IV-24. Klasifikasi *Multiclass SVM one-against-one*

$y = 1$	$y = -1$	Fungsi Keputusan
Kelas Factoid	Kelas Non-factoid	$f_{12}(x) = (w_{12})x + b_{12}$
Kelas Factoid	Kelas Others	$f_{13}(x) = (w_{13})x + b_{13}$
Kelas Non-factoid	Kelas Others	$f_{23}(x) = (w_{23})x + b_{23}$

Setelah mendapatkan nilai a dan b pada proses pelatihan. Maka selanjutnya mengecek kembali x_1 sebagai data uji apakah akan sesuai dengan hasil proses klasifikasi.

1. Menghitung nilai dot product pada data latih dan data uji.

$$\begin{aligned}
 x_1^T x_d &= (0 \times 0) + (2.176 \times 2.176) + (0 \times 0) + (0 \times 0) + (2.176 \times 2.176) + \\
 &\quad (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + \\
 &\quad (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + \\
 &\quad (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (2.176 \times 2.176) \\
 &= 14,205
 \end{aligned}$$

$$\begin{aligned}
 x_2^T x_d &= (2.176 \times 0) + (0 \times 2.176) + (0 \times 0) + (0 \times 0) + (0 \times 2.176) + (0 \times 0) + \\
 &\quad (1.875 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + \\
 &\quad (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (1.875 \times 0) + (0 \times 0) + (0 \times 0) \\
 &\quad + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 2.176) \\
 &= 0
 \end{aligned}$$

2. Menghitung fungsi pemisah optimal menggunakan persamaan II-15.

$$\begin{aligned}
 f(x) &= \text{sgn}(\sum_{i=1}^{n_{sv}} \alpha_i y_i K(x_i, x_d) + b) \\
 &= \text{sgn}((a_1 y_1 K(x_1, x_d)) + (a_1 y_2 K(x_2, x_d)) + b) \\
 &= \text{sgn}((0.074 \times 1 \times 14,205) + (0.074 \times (-1) \times \\
 &\quad 0) + (-0,22)) \\
 &= 1
 \end{aligned}$$

Pada proses klasifikasi yang dilakukan pada persamaan diatas untuk data uji (x_d), menunjukkan hasil klasifikasi berupa +1 yaitu kelas factoid berdasarkan label yang diberikan pada tabel fungsi keputusan sebelumnya. Dikarenakan hasil klasifikasi menunjukkan kelas factoid maka *vote* untuk kelas factoid akan ditambah satu.

4.2.3.5 Analisis Hasil Klasifikasi

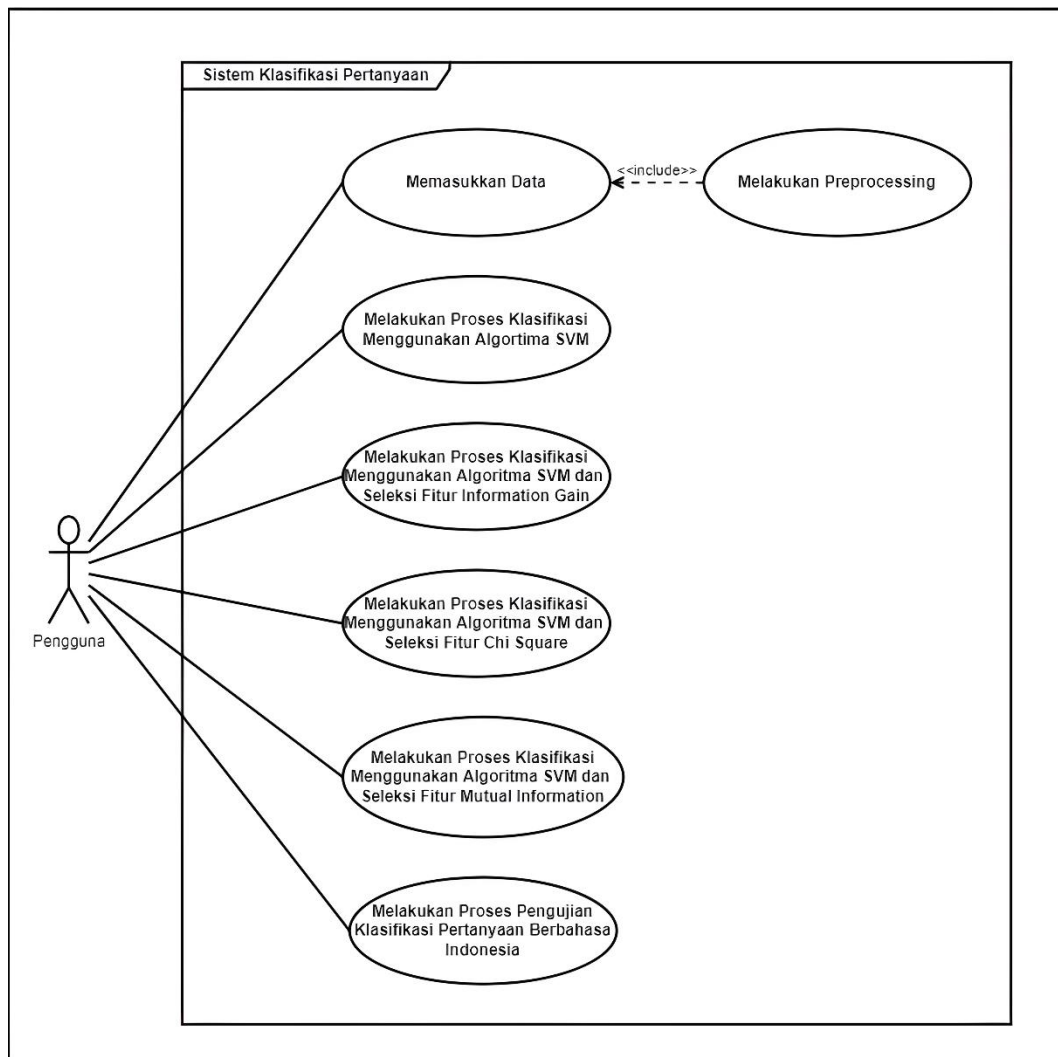
Untuk mengetahui kualitas dari hasil klasifikasi yang didapatkan, maka diperlukan suatu metode evaluasi. *Confusion Matrix* merupakan suatu metode yang digunakan untuk mengevaluasi hasil setiap proses klasifikasi dari perangkat lunak yang akan dikembangkan. Perhitungan waktu komputasi dan jumlah fitur data masukan juga dilakukan untuk mengetahui kecepatan dan hasil akhir dari jumlah fitur masukan untuk setiap proses klasifikasi.

4.2.4. Implementasi

Berdasarkan kebutuhan dan hasil analisis yang telah dilakukan pada fase sebelumnya, desain pada perangkat lunak akan dirancang menggunakan diagram *use case*. Rancangan desain diagram *use case* ditampilkan pada gambar IV-

1. Use Case

Diagram *use case* digunakan untuk menggambarkan interaksi antara aktor dengan perangkat lunak. Aktivitas pada perangkat lunak yang dapat dilakukan oleh aktor dapat dilihat pada diagram *usecase* pada gambar IV-1.



Gambar IV-1. Diagram *usecase*

2. Tabel Definisi *Actor*

Actor merupakan satu orang atau lebih yang berinteraksi dengan perangkat lunak sistem pengklasifikasi pertanyaan berbahasa Indonesia. Tabel IV-25 menampilkan tabel yang berisi definisi *actor* pada sistem perangkat lunak yang dibuat.

Tabel IV-25. Definisi *Actor*

No	Actor	Definisi
1.	Pengguna	Pengguna merupakan seseorang yang dapat berinteraksi secara langsung dengan sistem dan mendapatkan akses seluruh fitur yang ada pada perangkat lunak

3. Tabel Definisi Use Case

Tabel definisi *use case* pada sistem ini dapat dilihat pada tabel IV-26.

Tabel IV-26. Definisi *Use Case*

No	Use Case	Definisi
1.	Memasukkan data	Proses ini ditujukan untuk melakukan proses memasukkan <i>dataset</i> yang akan digunakan pada proses klasifikasi.
2.	Melakukan preprocessing	Proses ini ditujukan untuk melakukan proses pengolahan data pada <i>dataset</i> masukkan yang terdiri dari <i>noise removal</i> dan <i>case folding</i> .
3.	Melakukan proses klasifikasi menggunakan algoritma SVM	Proses ini ditujukan untuk melakukan proses pengklasifikasian pada data masukkan menggunakan algoritma SVM.
4.	Melakukan proses klasifikasi menggunakan algoritma SVM dan seleksi fitur <i>information gain</i>	Proses ini ditujukan untuk melakukan proses pengklasifikasian pada data masukkan menggunakan algoritma SVM dan seleksi fitur <i>information gain</i> .
5.	Melakukan proses klasifikasi menggunakan algoritma SVM dan seleksi fitur <i>chi-square</i>	Proses ini ditujukan untuk melakukan proses pengklasifikasian pada data masukkan menggunakan algoritma SVM dan seleksi fitur <i>chi-square</i> .
6.	Melakukan proses klasifikasi menggunakan algoritma SVM dan	Proses ini ditujukan untuk melakukan proses pengklasifikasian pada data masukkan

	seleksi fitur <i>mutual information</i>	menggunakan algoritma SVM dan seleksi fitur <i>mutual information</i> .
7.	Melakukan proses pengujian pertanyaan berbahasa Indonesia	Proses ini ditunjukkan untuk melakukan proses pengujian pada data uji berupa pertanyaan berbahasa Indonesia berdasarkan model klasifikasi yang digunakan.

4. Skenario Use Case

Skenario *use case* merupakan uraian dari setiap tahapan yang menggambarkan urutan interaksi yang terjadi antara *actor* dan sistem yang dibangun. Berdasarkan tabel diatas, tabel berikut merupakan scenario untuk setiap *use case*.

Tabel IV-27. Skenario Melakukan Memasukkan Data

No. Use Case	001
Nama Use Case	Memasukkan Data
Actor	Pengguna
Tujuan	Pengguna memasukkan dataset yang akan digunakan sebagai data masukkan
Deskripsi	Proses ini digunakan oleh pengguna untuk melakukan proses pra-pengolahan pada data masukan sebelum data tersebut dilakukan proses klasifikasi
Kondisi Awal	Belum terdapat data masukkan
Kondisi Akhir	Sistem menyimpan file data masukkan
Skenario Normal	
Actor	Sistem
1. Menekan tombol “Input Data Awal”.	
	2. Menampilkan jendela pencarian berkas.

3. Memilih file dataset yang akan diproses.	
Kondisi Akhir Skenario Normal	4. Menyimpan file data masukan.

Tabel IV-28. Skenario Melakukan Praproses Data

No. Use Case	002
Nama Use Case	Melakukan <i>Pre Processing</i>
Actor	Pengguna
Tujuan	Melakukan proses pra-pengolahan pada data masukan
Deskripsi	Proses ini digunakan untuk melakukan proses pra-pengolahan pada data masukan yang terdiri dari proses <i>noise removal</i> dan <i>case folding</i> .
Kondisi Awal	Belum terdapat data masukan
Kondisi Akhir	Sistem menampilkan data hasil pra-pengolahan
Skenario Normal	
Actor	Sistem
1. Menekan tombol "Input Data Awal".	
	2. Menampilkan jendela pencarian berkas.
3. Memilih file dataset yang akan diproses.	
	4. Menyimpan file data masukan.
	5. Melakukan proses pra pengolahan (<i>case folding</i> , <i>noise removal</i> dan <i>tokenization</i>) pada data masukan.
	6. Menyimpan data hasil pra-pengolahan.
Kondisi Akhir Skenario Normal	7. Menampilkan data hasil pra-pengolahan.

Tabel IV-29. Skenario Melakukan Proses Klasifikasi Menggunakan Algoritma SVM

No. Use Case	003
Nama Use Case	Melakukan proses klasifikasi menggunakan algoritma SVM.
Actor	Pengguna
Tujuan	Melakukan proses klasifikasi pada data masukan, kemudian menampilkan hasil evaluasi dari proses klasifikasi yang dilakukan
Deskripsi	Proses digunakan oleh pengguna untuk melakukan hasil klafisikasi menggunakan algortima SVM.
Kondisi Awal	Data masukan hasil pra-pengolahan
Kondisi Akhir	Sistem menampilkan hasil pengujian
Skenario Normal	
Actor	Sistem
1. Memilih metode SVM	
2. Memilih Kernel	
3. Memilih nilai C	
4. Menekan tombol klasifikasi	
	6. Melakukan proses pembobotan kata menggunakan TF-IDF
	7. Membagi data latih dan data uji menggunakan metode k-fold cross validation
	8. Melakukan proses klasifikasi menggunakan SVM
Kondisi Akhir Skenario Normal	9. Menampilkan nilai evaluasi klasifikasi (TP, FP, TN, FN, Akurasi, Precision, Recall, FMeasure, waktu

	komputasi, jumlah fitur dan selisih fitur)
Skenario alternatif : Pengguna tidak memilih kernel dan/atau nilai C	
1. Memilih metode SVM	
2. Tidak memilih Kernel	
3. Tidak memilih nilai C	
4. Menekan tombol klasifikasi	
	5. Menampilkan <i>pop-up box warning</i> "Masukkan ulang parameter"

Tabel IV-30. Skenario Melakukan Proses Klasifikasi Menggunakan Algoritma SVM dan Seleksi Fitur *Information Gain*

No. Use Case	004
Nama Use Case	Melakukan proses klasifikasi menggunakan algoritma SVM dan seleksi fitur <i>Information Gain</i> .
Actor	Pengguna
Tujuan	Melakukan proses klasifikasi pada data masukan, kemudian menampilkan hasil evaluasi dari proses klasifikasi yang dilakukan.
Deskripsi	Proses digunakan oleh pengguna untuk melakukan hasil klafisikasi menggunakan algortima SVM dan melakukan proses seleksi fitur menggunakan metode <i>Information Gain</i> .
Kondisi Awal	Data masukan hasil pra-pengolahan.
Kondisi Akhir	Sistem menampilkan hasil pengujian.
Skenario Normal	
Actor	Sistem
1. Memilih metode SVM + IG	
2. Memilih Kernel	
3. Memilih nilai C	

4. Masukkan nilai Treshold	
5. Menekan tombol Lakukan Proses Klasifikasi	
	6. Menghitung bobot nilai <i>Information Gain</i> pada tiap <i>term</i>
	7. Melakukan seleksi fitur berdasarkan nilai <i>Information Gain</i>
	8. Melakukan proses pembobotan kata menggunakan TF-IDF
	9. Membagi data latih dan data uji menggunakan metode <i>k-fold cross validation</i>
	10. Melakukan proses klasifikasi menggunakan SVM
Kondisi Akhir Skenario Normal	11. Menampilkan nilai evaluasi klasifikasi (TP, FP, TN, FN, Akurasi, Precision, Recall, FMeasure, waktu komputasi, jumlah fitur dan selisih fitur)
Skenario alternatif : Pengguna belum memilih kernel dan/atau nilai C	
1. Memilih metode seleksi fitur <i>Information Gain</i>	
2. Tidak memilih Kernel	
3. Tidak memilih nilai C	
4. Tidak memasukkan nilai treshold	
5. Menekan tombol klasifikasi	
	6. Menampilkan <i>pop-up box warning</i> "Masukkan parameter ulang"

Tabel IV-31. Skenario Melakukan Proses Klasifikasi Menggunakan Algoritma SVM dan Seleksi Fitur Chi-Square

No. Use Case	005
Nama Use Case	Melakukan proses klasifikasi menggunakan algoritma SVM dan seleksi fitur <i>Chi-Square</i>
Actor	Pengguna
Tujuan	Melakukan proses klasifikasi pada data masukan, kemudian menampilkan hasil evaluasi dari proses klasifikasi yang dilakukan
Deskripsi	Proses digunakan oleh pengguna untuk melakukan hasil klasifikasi menggunakan algoritma SVM dan melakukan proses seleksi fitur menggunakan metode <i>Chi-Square</i>
Kondisi Awal	Data masukan hasil pra-pengolahan
Kondisi Akhir	Sistem menampilkan hasil pengujian
Skenario Normal	
Actor	Sistem
1. Memilih metode SVM + CS	
2. Memilih Kernel	
3. Memilih nilai C	
4. Memasukkan nilai Treshold	
5. Menekan tombol Lakukan Proses Klasifikasi	
	6. Menghitung bobot nilai <i>Chi-Square</i> pada tiap <i>term</i>
	7. Melakukan seleksi fitur berdasarkan nilai <i>Chi-Square</i>
	8. Melakukan proses pembobotan kata menggunakan TF-IDF
	9. Membagi data latih dan data uji menggunakan metode <i>k-fold cross validation</i>

	10. Melakukan proses klasifikasi menggunakan SVM
Kondisi Akhir Skenario Normal	11. Menampilkan nilai evaluasi klasifikasi (TP, FP, TN, FN, Akurasi, Precision, Recall, FMeasure, waktu komputasi, jumlah fitur dan selisih fitur)
Skenario alternatif : Pengguna tidak memilih kernel dan/atau nilai C	
1. Memilih metode seleksi fitur <i>Chi Square</i>	
2. Tidak memilih Kernel	
3. Tidak memilih nilai C	
4. Tidak memasukkan nilai treshold	
5. Menekan tombol klasifikasi	
	6. Menampilkan <i>pop-up box warning</i> "Masukkan parameter ulang"

Tabel IV-32. Skenario Melakukan Proses Klasifikasi Menggunakan Algoritma SVM dan Seleksi Fitur *Mutual Information*

No. Use Case	006
Nama Use Case	Melakukan proses klasifikasi menggunakan algoritma SVM dan seleksi fitur <i>Mutual Information</i>
Actor	Pengguna
Tujuan	Melakukan proses klasifikasi pada data masukan, kemudian menampilkan hasil evaluasi dari proses klasifikasi yang dilakukan
Deskripsi	Proses digunakan oleh pengguna untuk melakukan hasil klafisikasi menggunakan algortima SVM dan melakukan proses seleksi fitur menggunakan metode <i>Mutual Information</i>

Kondisi Awal	Data masukan hasil pra-pengolahan
Kondisi Akhir	Sistem menampilkan hasil pengujian
Skenario Normal	
<i>Actor</i>	Sistem
1. Memilih metode SVM + MI	
2. Memilih Kernel	
3. Memilih nilai C	
4. Memasukkan Nilai Treshold	
5. Menekan tombol Lakukan Proses Klasifikasi	
	6. Menghitung bobot nilai <i>Mutual Information</i> pada tiap <i>term</i>
	7. Melakukan seleksi fitur berdasarkan nilai <i>Mutual Information</i>
	8. Melakukan proses pembobotan kata menggunakan TF-IDF
	9. Membagi data latih dan data uji menggunakan metode <i>k-fold cross validation</i>
	10. Melakukan proses klasifikasi menggunakan SVM
Kondisi Akhir Skenario Normal	11. Menampilkan nilai evaluasi klasifikasi (TP, FP, TN, FN, Akurasi, Precision, Recall, FMeasure, waktu komputasi, jumlah fitur dan selisih fitur)
Skenario alternatif : Pengguna tidak memilih kernel dan/atau nilai C	
1. Memilih metode seleksi fitur <i>Mutual Information</i>	
2. Tidak memilih kernel	

3. Tidak memilih nilai C	
4. Tidak memasukkan nilai treshold	
5. Menekan tombol klasifikasi	
	6. Menampilkan <i>pop-up box warning</i> "Masukkan parameter ulang"

Tabel IV-33. Skenario Melakukan Proses Pengujian Klasifikasi
Pertanyaan Berbahasa Indonesia

No. Use Case	007
Nama Use Case	Melakukan proses pengujian klasifikasi pertanyaan berbahasa indonesia
Actor	Pengguna
Tujuan	Melakukan proses klasifikasi untuk memprediksi label pada data masukan pengguna
Deskripsi	Proses digunakan untuk memprediksi label pada data masukan pengguna berdasarkan model yang digunakan pada proses klasifikasi.
Kondisi Awal	Data masukan berupa pertanyaan berbahasa indonesia
Kondisi Akhir	Sistem menampilkan hasil prediksi label pertanyaan
Skenario Normal	
Actor	Sistem
1. Memasukkan data pertanyaan berbahasa indonesia	
2. Menekan tombol "Lakukan Pengujian"	
	3. Melakukan pra-pengolahan
	4. Melakukan ekstraksi fitur menggunakan metode seleksi fitur dan TF-IDF

	5. Melakukan proses prediksi pertanyaan berdasarkan model yang telah digunakan
Kondisi Akhir Skenario Normal	6. Menampilkan hasil prediksi data masukan untuk mengklasifikasi kategori pertanyaan factoid, non-factoid dan others.

4.3 Fase Elaborasi

Fase elaborasi merupakan fase kedua yang perlu dilakukan dalam proses pengembangan perangkat lunak menggunakan RUP. Fase ini akan mencakup uraian mengenai pemodelan bisnis, perancangan data, perancangan tampilan antarmuka, diagram *activity*, diagram *sequence* dan dokumentasi.

4.3.1 Pemodelan Bisnis

Subbab ini akan menguraikan mengenai penjelasan tentang perancangan perangkat lunak yang nantinya akan dikembangkan. Perancangan tersebut akan dibuat berdasarkan pemodelan bisnis yang telah diuraikan pada fase insepisi. Pada tahap ini akan menguraikan mengenai perancangan data yang akan digunakan dan perancangan desain antarmuka pengguna pada sistem yang akan dibangun.

4.3.2 Perancangan Data

Perangkat lunak pada penelitian ini menggunakan data masukan sebagai data yang akan dilakukan proses klasifikasi. Data masukan yang digunakan berupa dataset pertanyaan Berbahasa Indonesia yang telah diberi label. Data tersebut akan disimpan dalam format.xlsx.

4.3.3 Perancangan Antarmuka

Subbab ini akan menampilkan gambaran desain *interface* perangkat lunak yang akan dibangun. Gambar IV-3, IV-4, dan IV-5 menunjukkan rancangan desain *interface* pada penelitian ini.

Gambar IV-2. Rancangan Antarmuka Perangkat Lunak

4.3.4 Kebutuhan Sistem

Pada perangkat lunak yang dibangun memerlukan perangkat keras (*hardware*), perangkat lunak (*Software*), dan bahasa pemrograman. Bahasa pemrograman yang dipakai dalam penelitian ini ialah Python. Perangkat keras yang dibutuhkan untuk melakukan proses pembangunan pada sistem ini sebagai berikut:

1. *Processor* Intel Core i5-5005U
2. RAM 8 GB

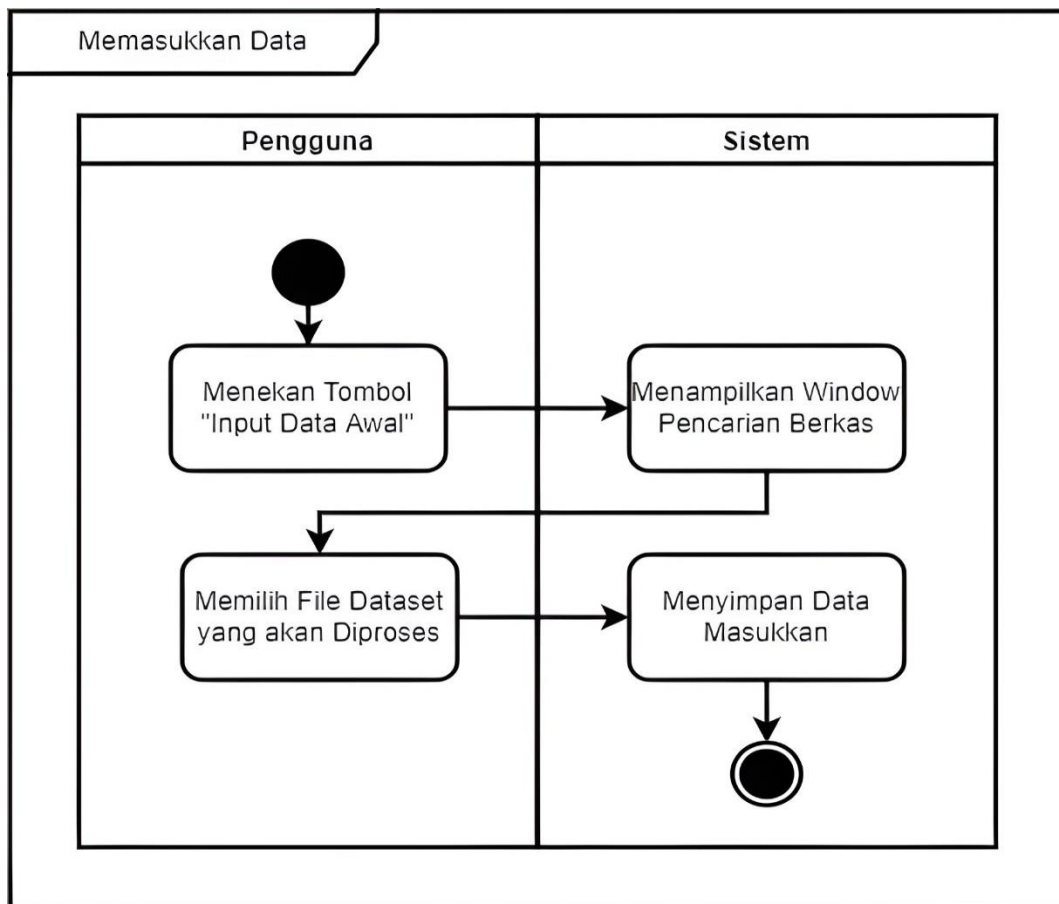
3. Harddisk 500 GB

Sedangkan perangkat lunak yang digunakan sebagai berikut:

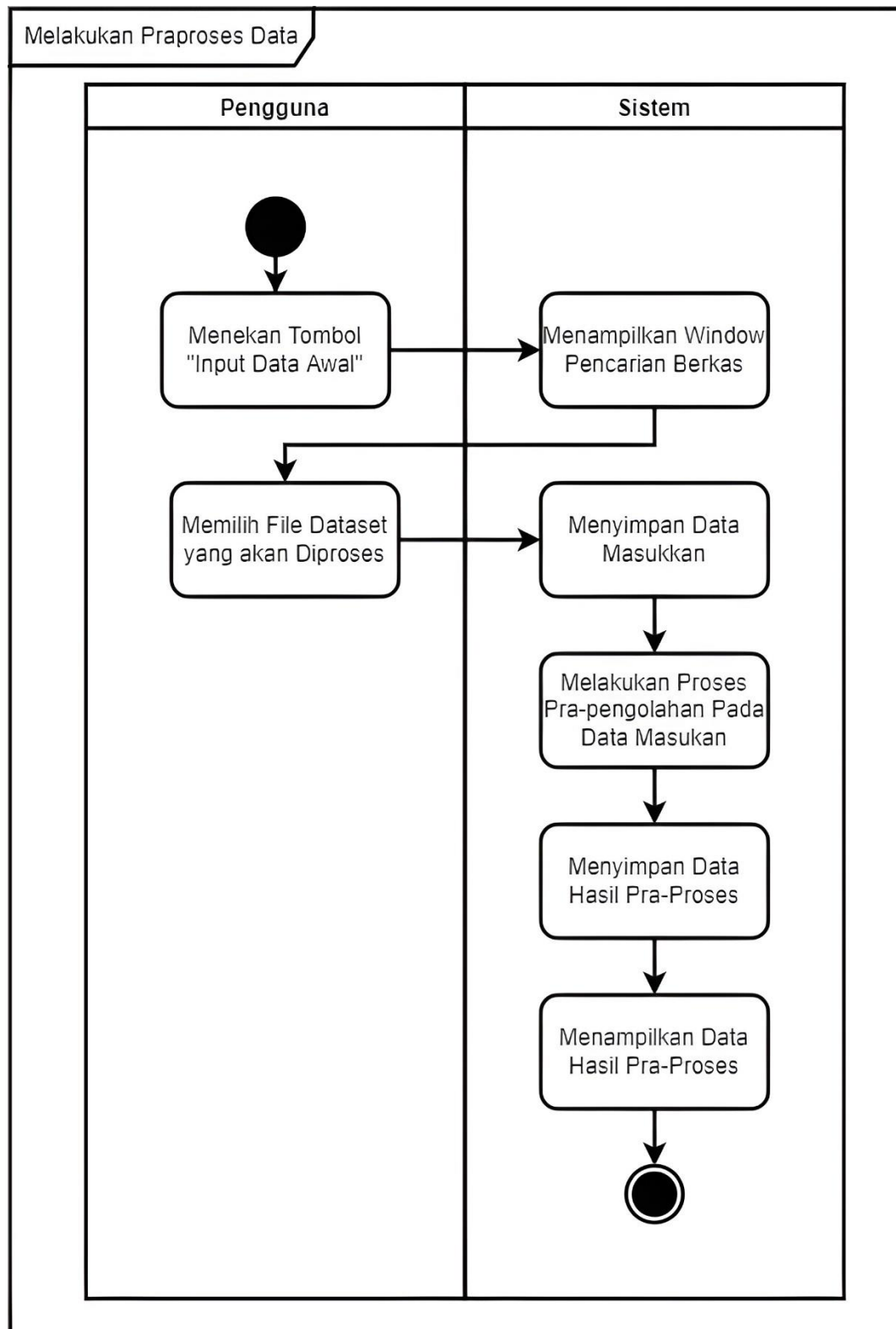
1. Sistem Operasi Windows 10 64-bit
2. Text Editor Spyder

4.3.5 Diagram Aktivitas

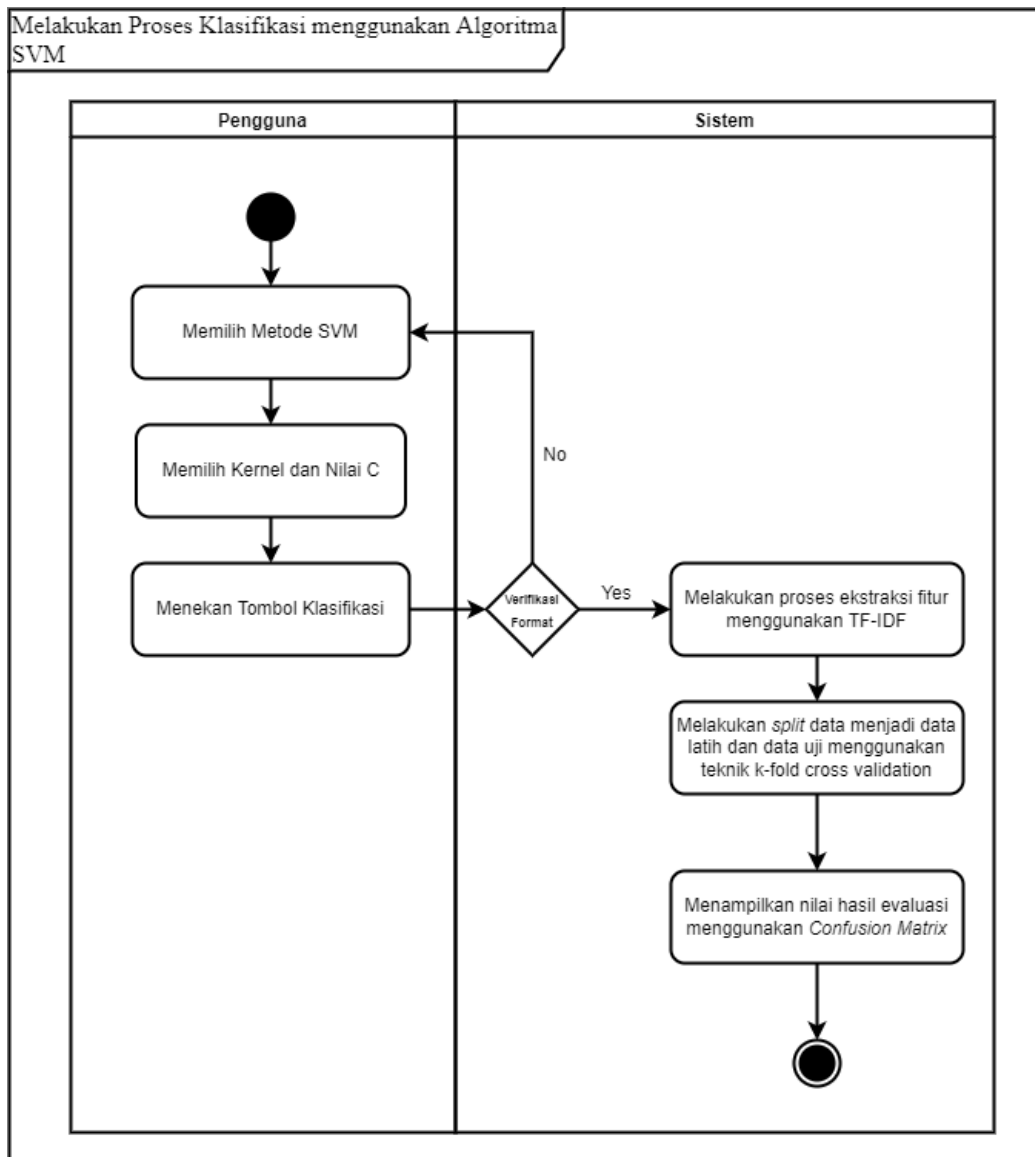
Diagram aktivitas merupakan suatu diagram yang menggambarkan aktivitas yang dilakukan oleh *actor*. Pada penelitian ini terdiri dari tujuh rancangan diagram aktivitas yang mengacu pada diagram *use case* untuk pengembang perangkat lunak yang dilakukan.



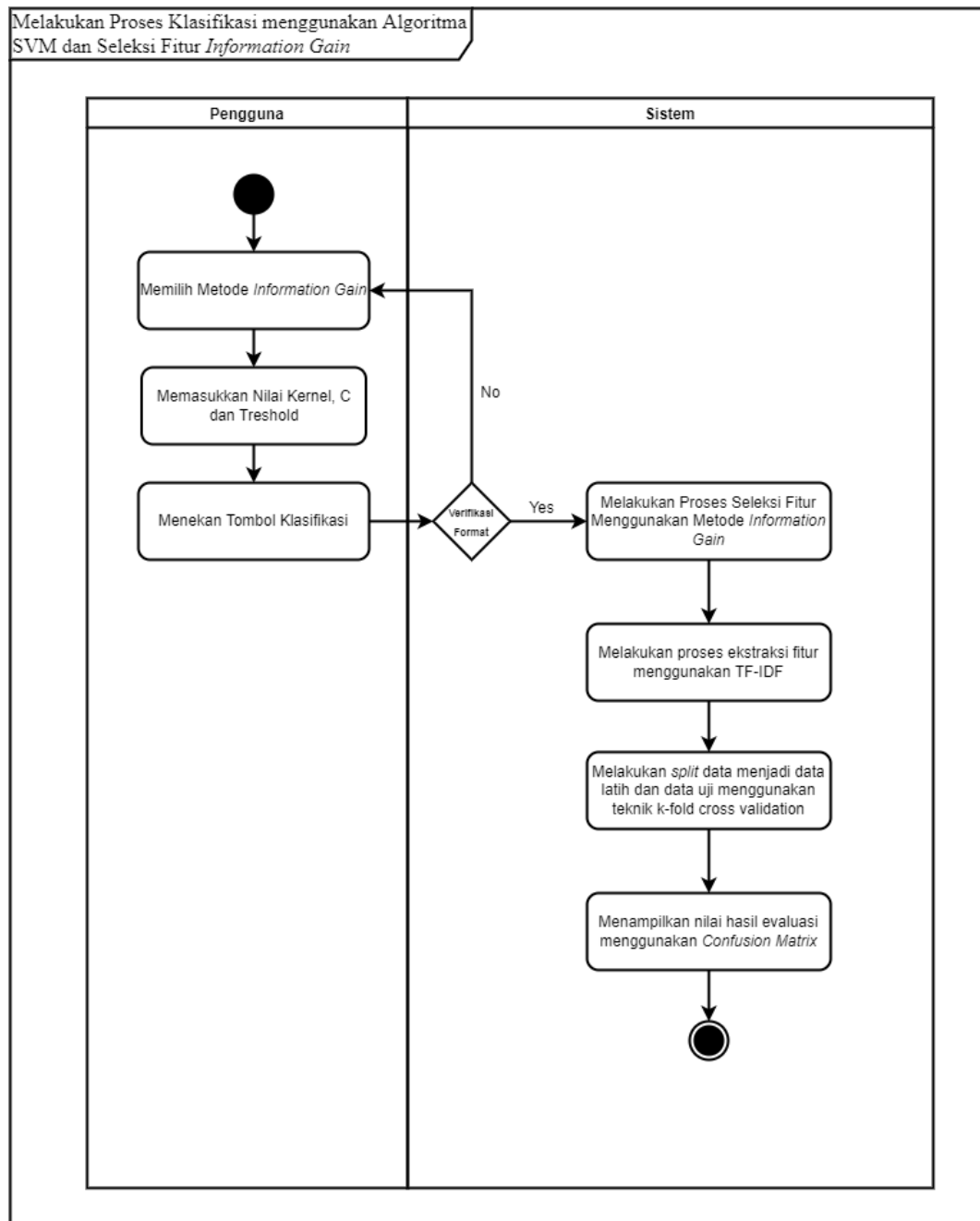
Gambar IV-3. Diagram Aktivitas Memasukkan Data



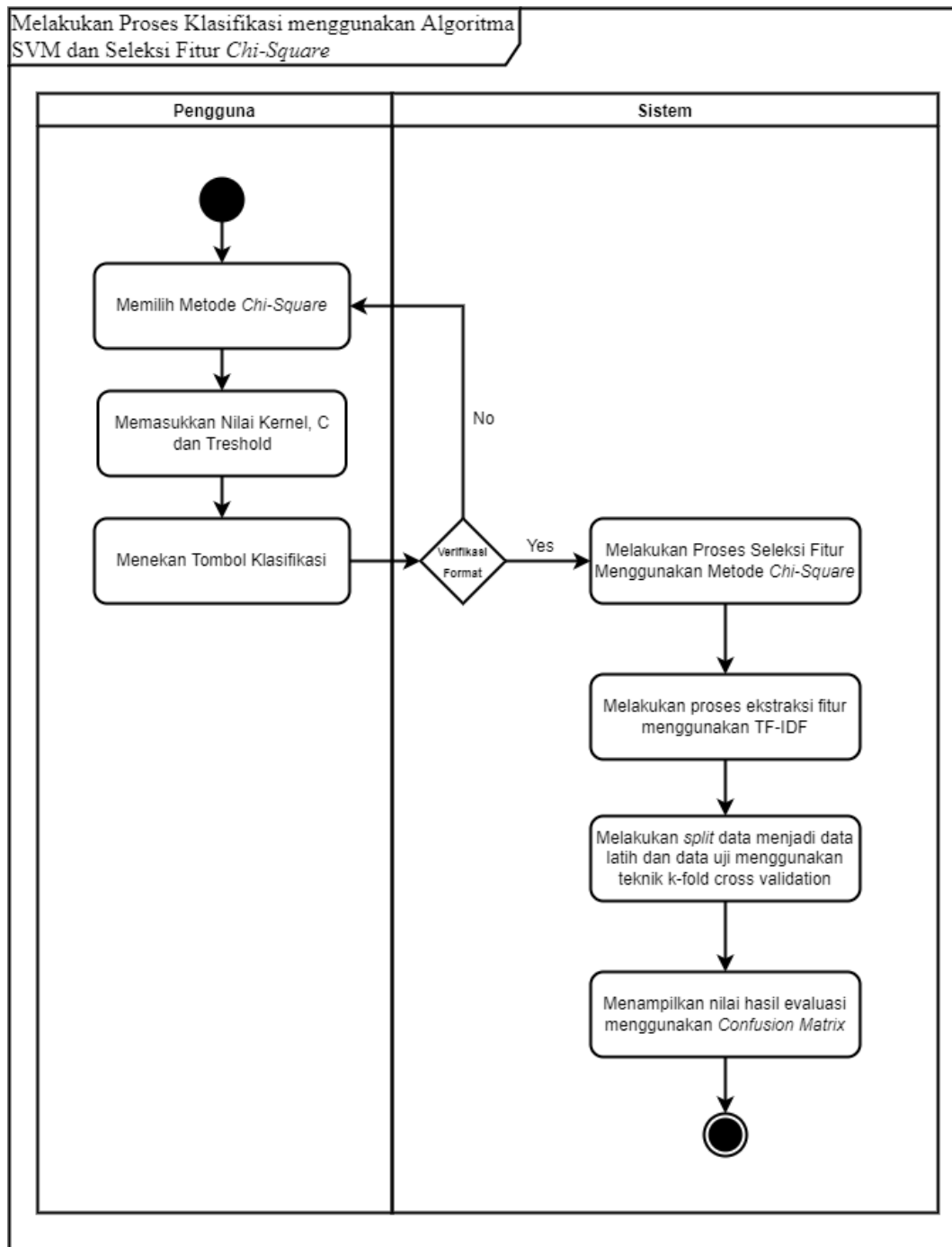
Gambar IV-4. Diagram Aktivitas Melakukan *Pre Processing*



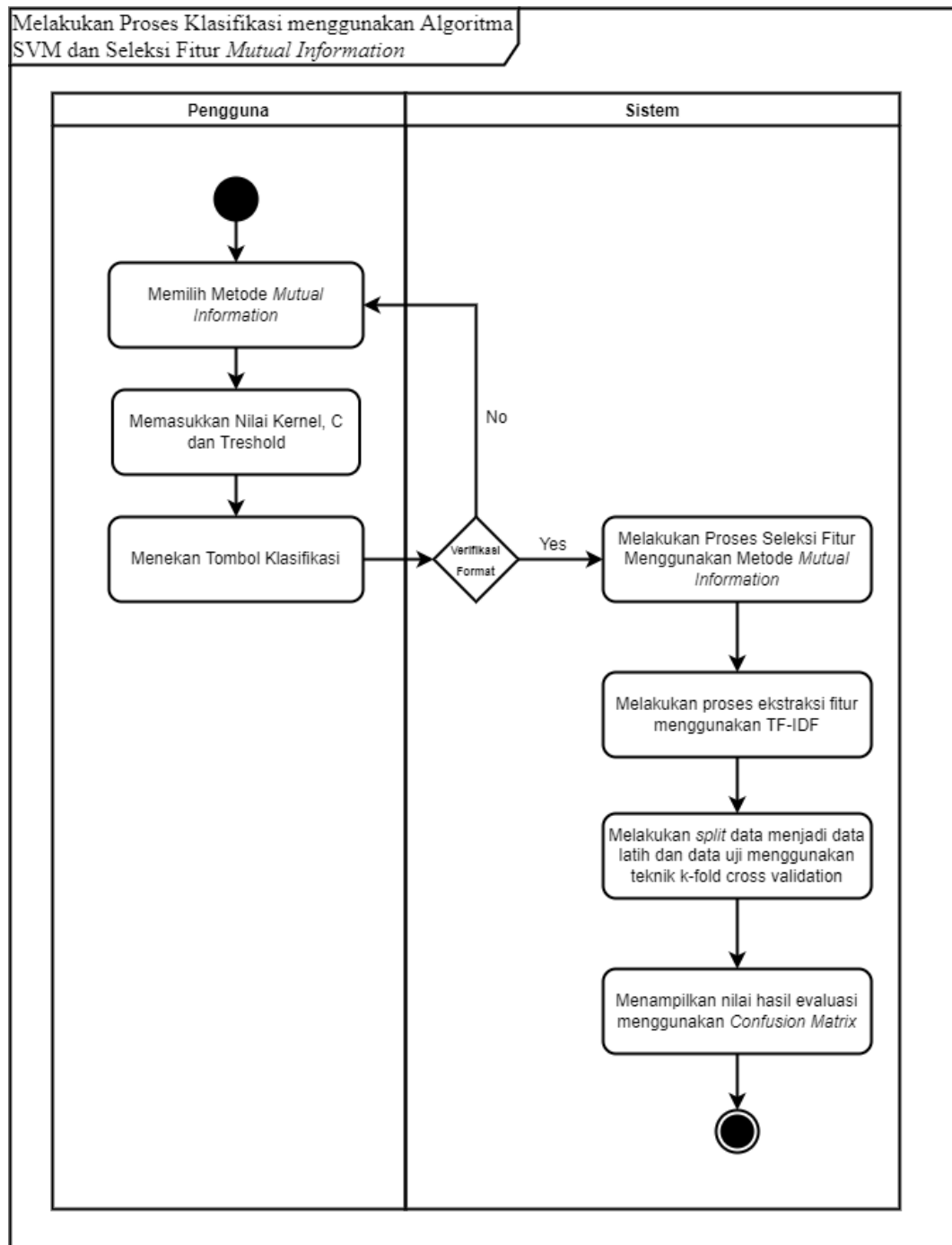
Gambar IV-5. Diagram Aktivitas Melakukan Proses Klasifikasi menggunakan Algoritma SVM



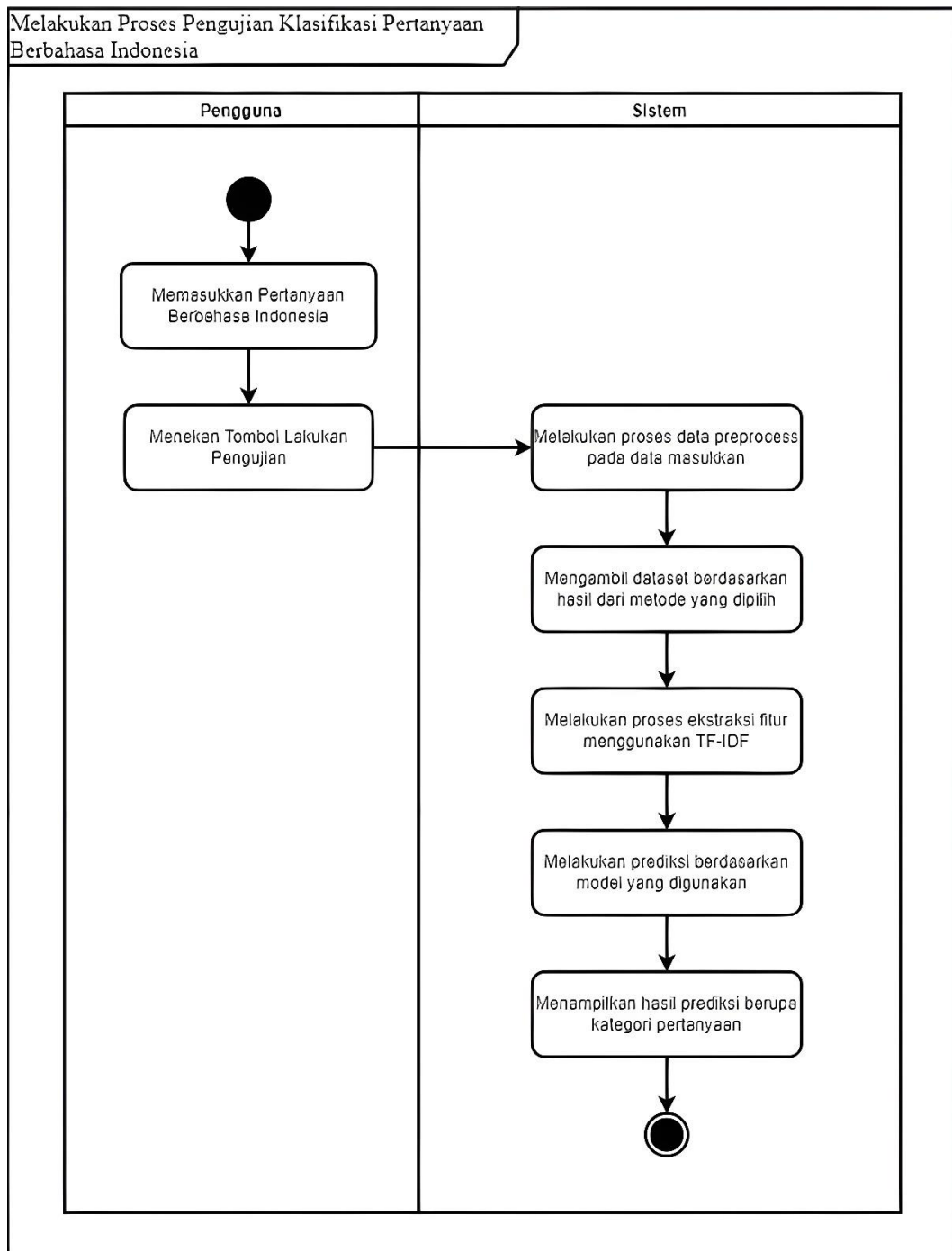
Gambar IV-6. Diagram Aktivitas Melakukan Proses Klasifikasi menggunakan Algoritma SVM dan Seleksi Fitur *Information Gain*



Gambar IV-7. Diagram Aktivitas Melakukan Proses Klasifikasi menggunakan Algoritma SVM dan Seleksi Fitur *Chi-Square*



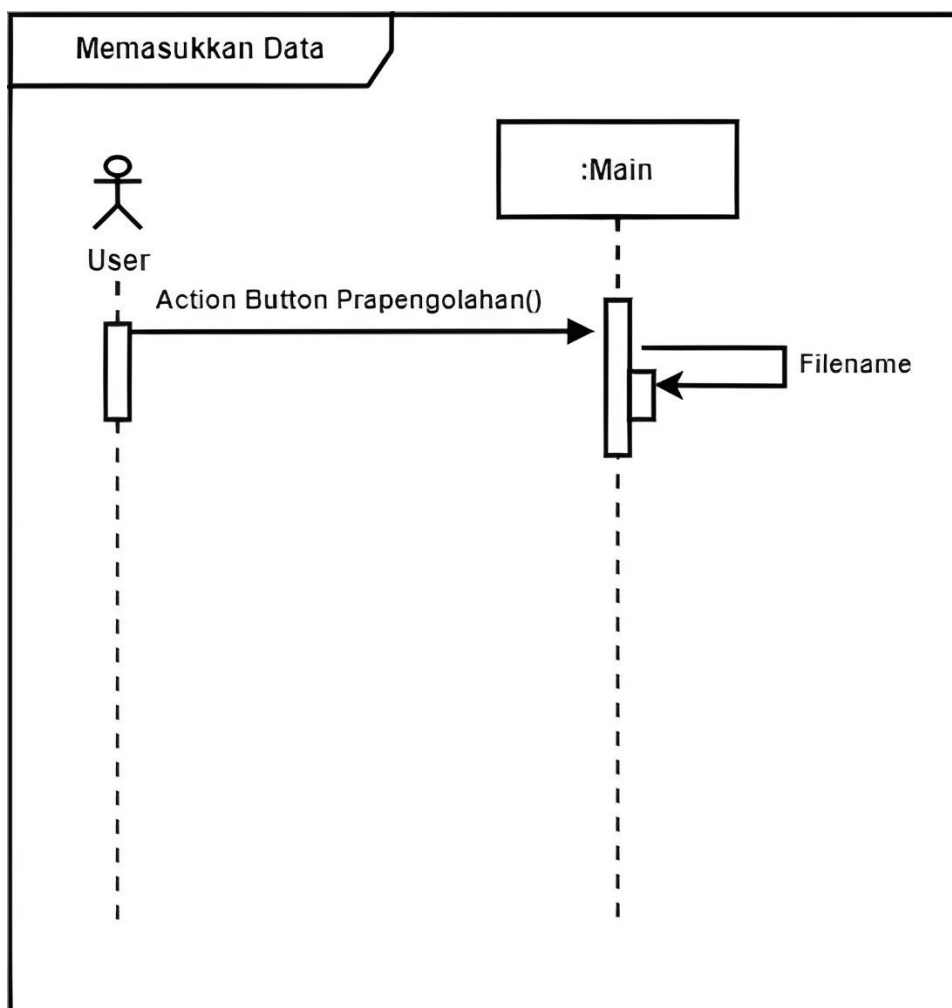
Gambar IV-8. Diagram Aktivitas Melakukan Proses Klasifikasi menggunakan Algoritma SVM dan Seleksi Fitur *Mutual Information*



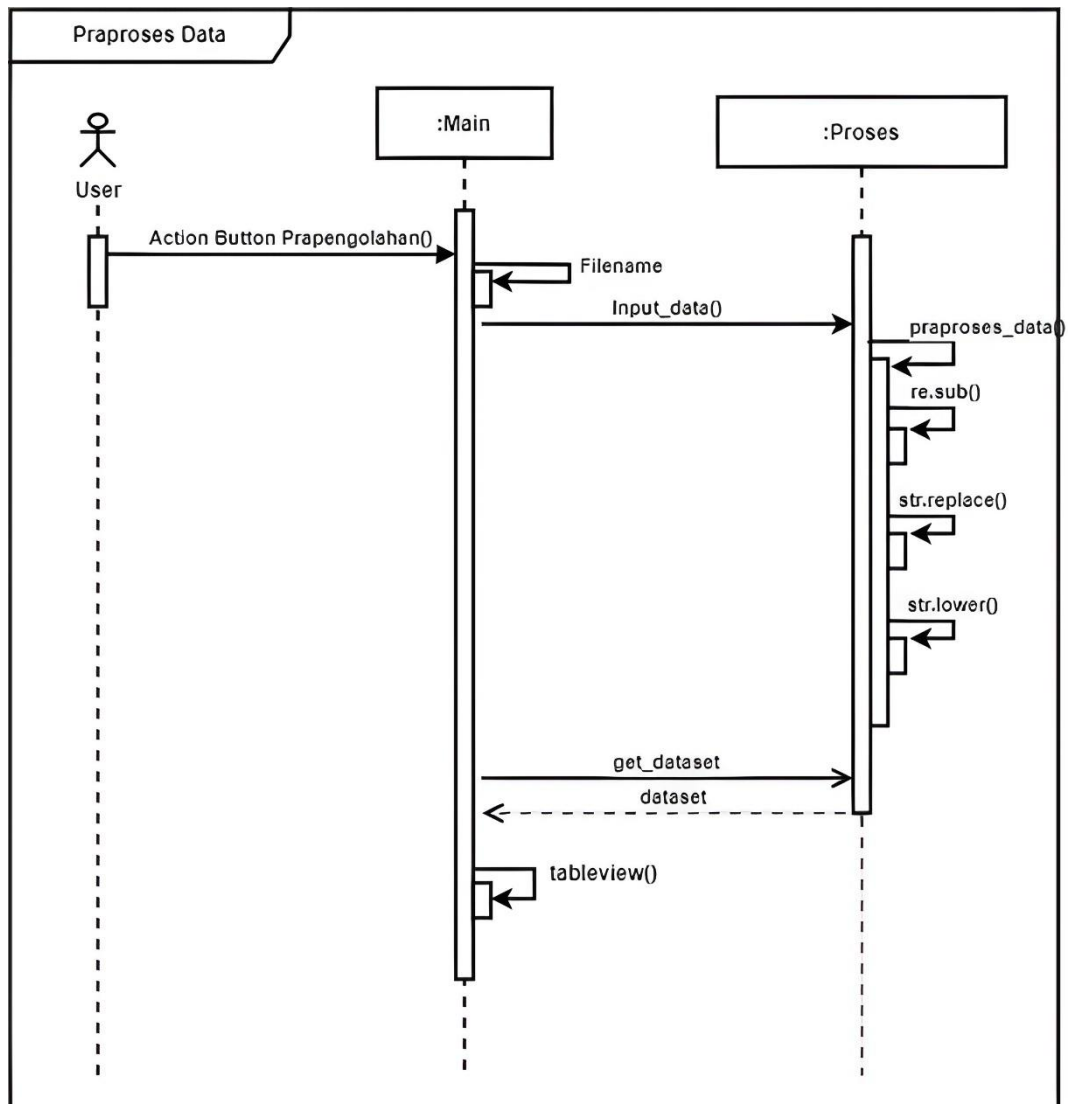
Gambar IV-9. Diagram Aktivitas Melakukan Proses Pengujian Klasifikasi Pertanyaan Berbahasa Indonesia

4.3.6 Diagram Sequence

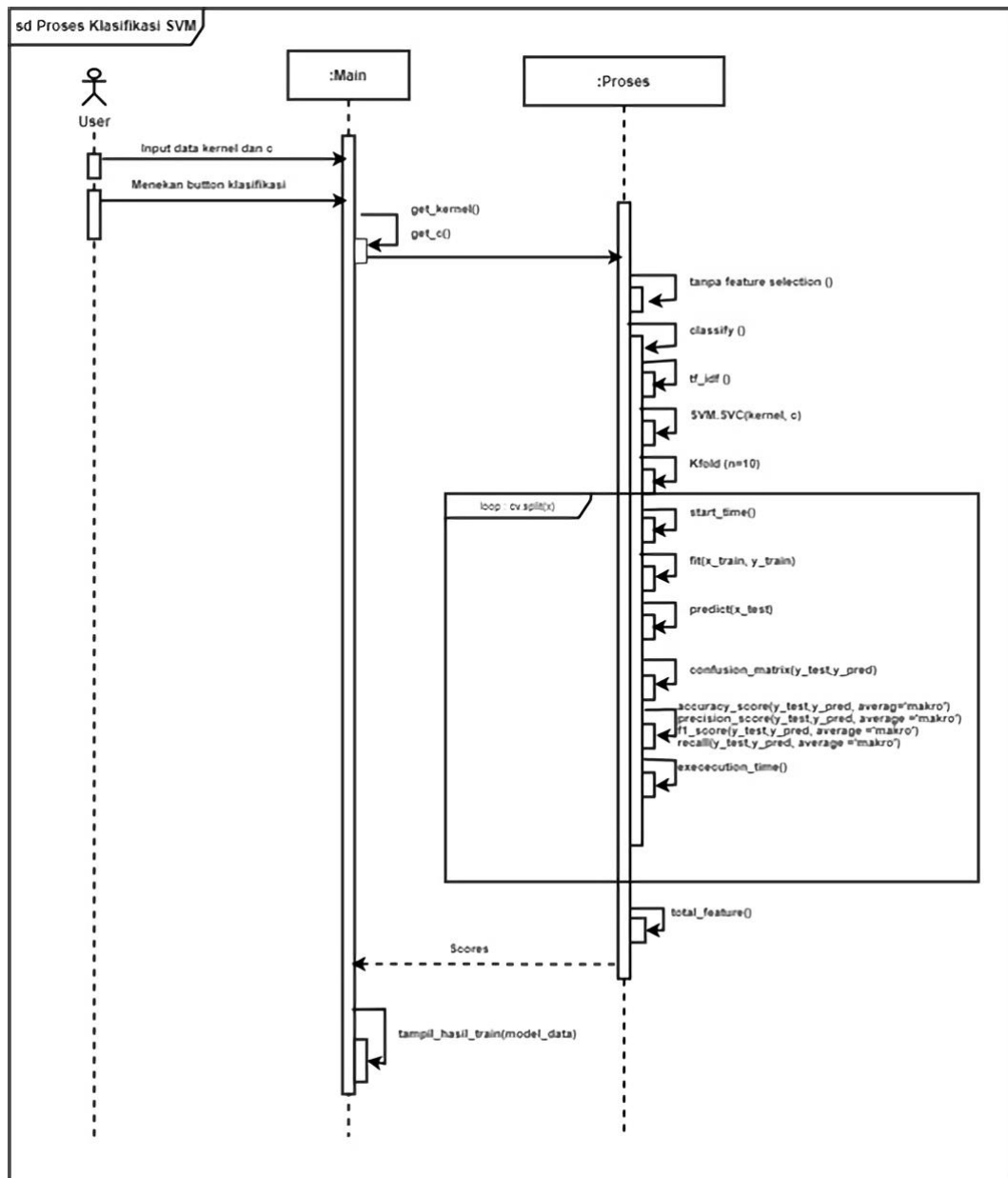
Diagram *sequence* atau yang biasa disebut diagram alur merupakan suatu diagram yang menggambarkan interaksi antar objek secara berurut yang ada di dalam dan disekitar sistem. Berdasarkan *use case* yang telah dirancang, terdapat tujuh diagram *sequence* yang digunakan pada penelitian ini. Diagram *sequence* tersebut ditampilkan pada gambar berikut.



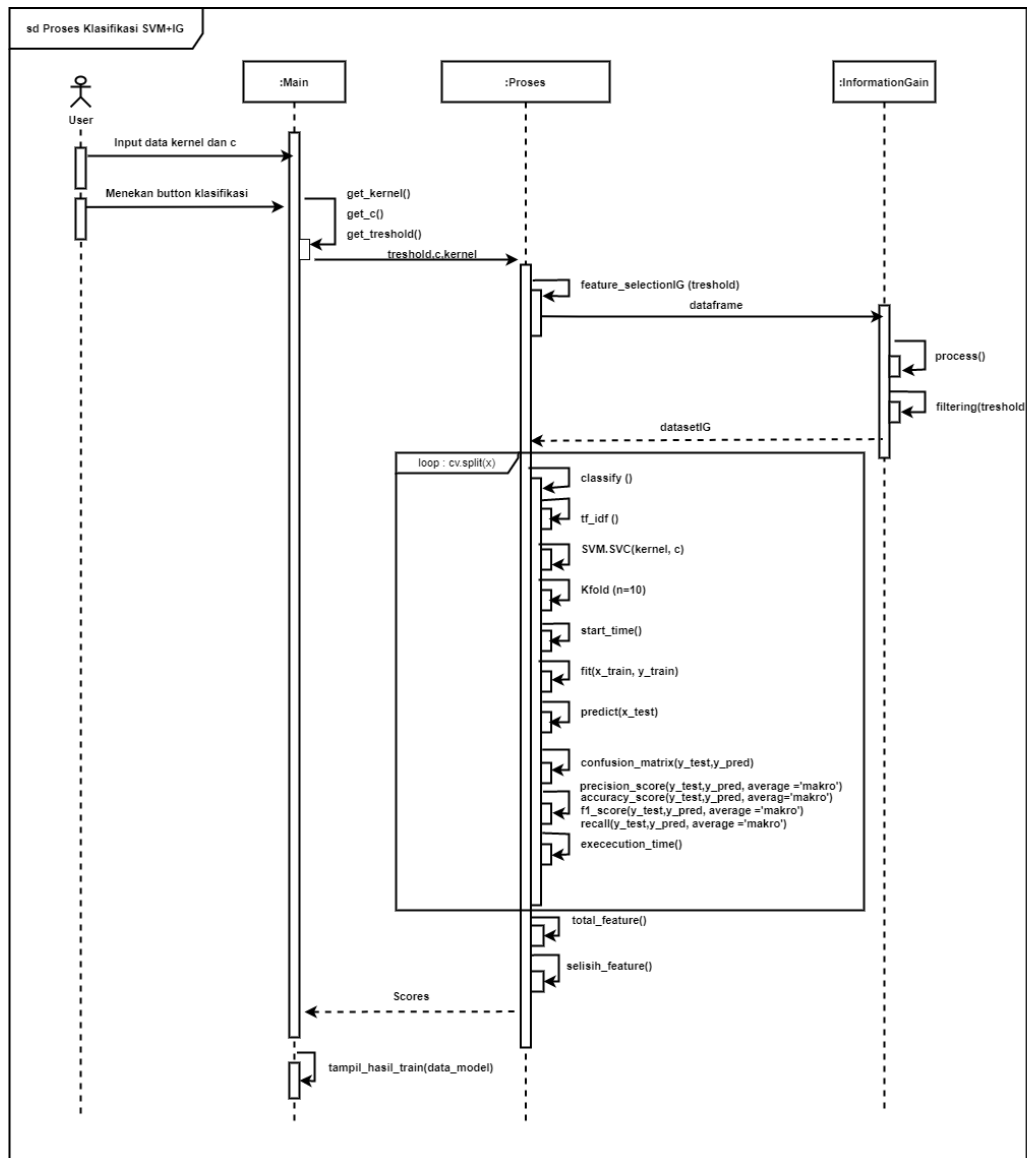
Gambar IV-10. Diagram Sequence Memasukkan Data



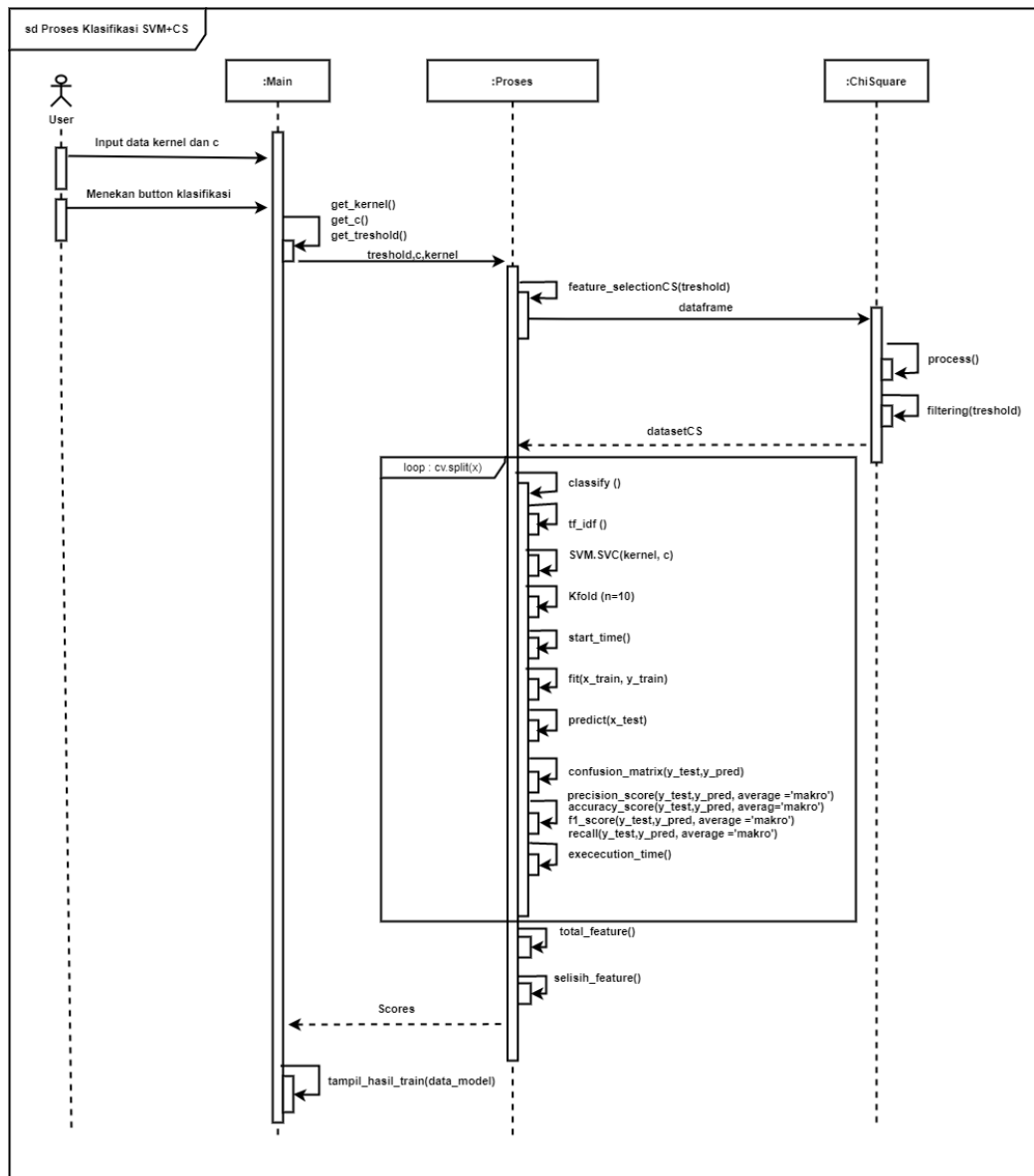
Gambar IV-11. Diagram *Sequence* Memilih *Dataset*



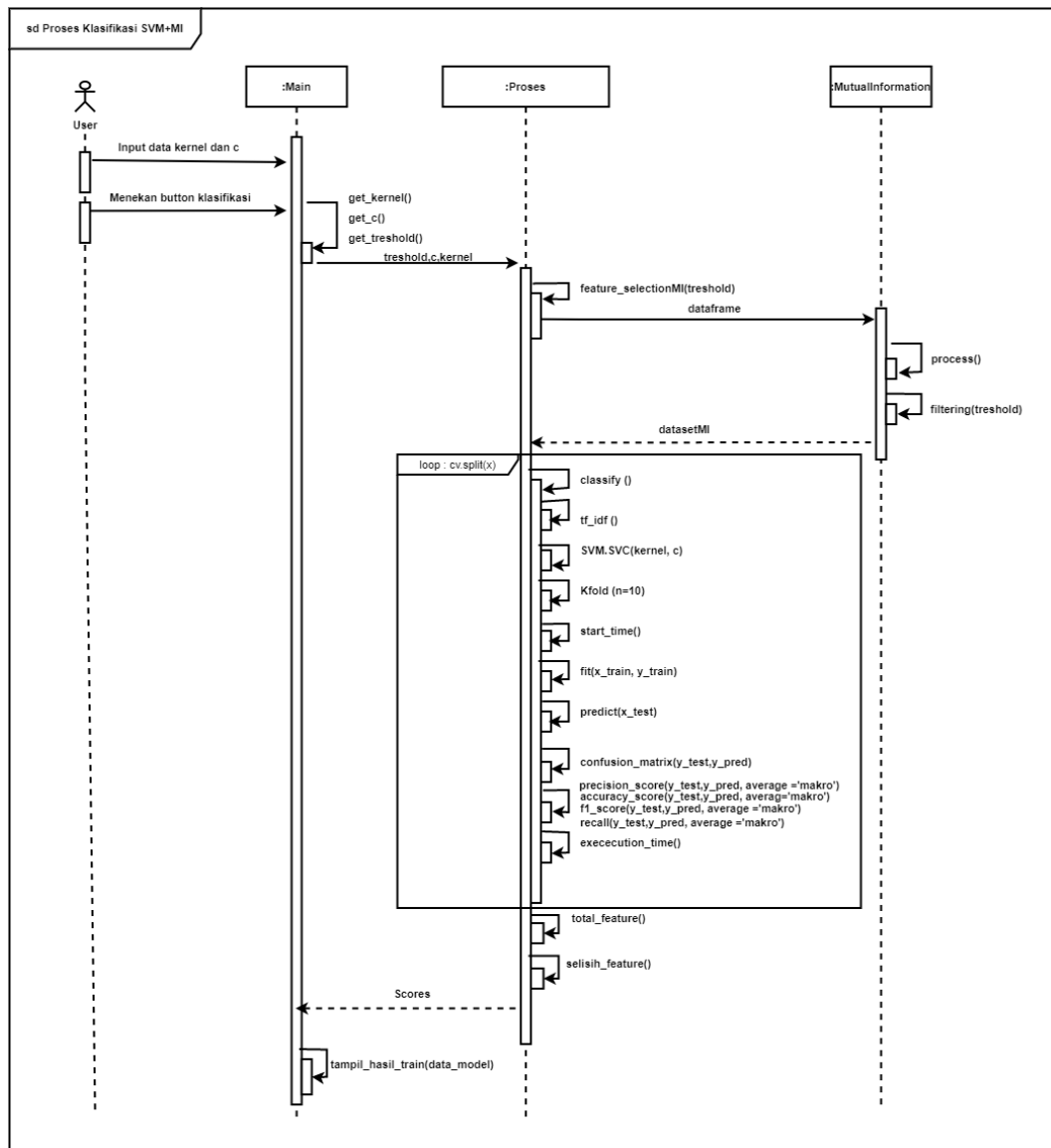
Gambar IV-12. Diagram *Sequence* Melakukan Proses Klasifikasi Menggunakan SVM



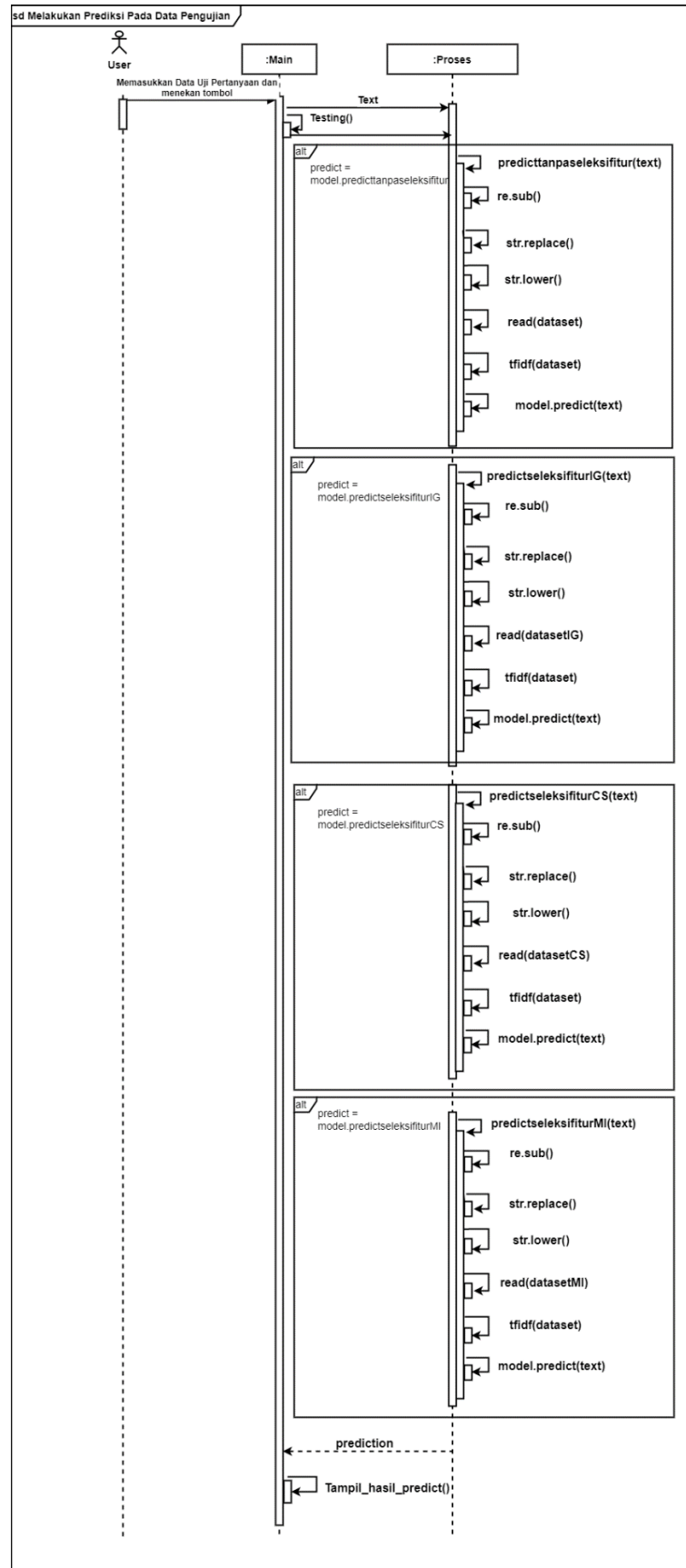
Gambar IV-13. Diagram *Sequence* Melakukan Proses Klasifikasi Menggunakan SVM dan Seleksi Fitur *Information Gain*



Gambar IV-14. Diagram *Sequence* Melakukan Proses Klasifikasi Menggunakan SVM dan Seleksi Fitur *Chi-Square*



Gambar IV-15. Diagram *Sequence* Melakukan Proses Klasifikasi Menggunakan SVM dan Seleksi Fitur *Mutual Information*



Gambar IV-16. Diagram *Sequence* Melakukan Proses Pengujian Klasifikasi Pertanyaan Berbahasa Indonesia

4.4 Fase Konstruksi

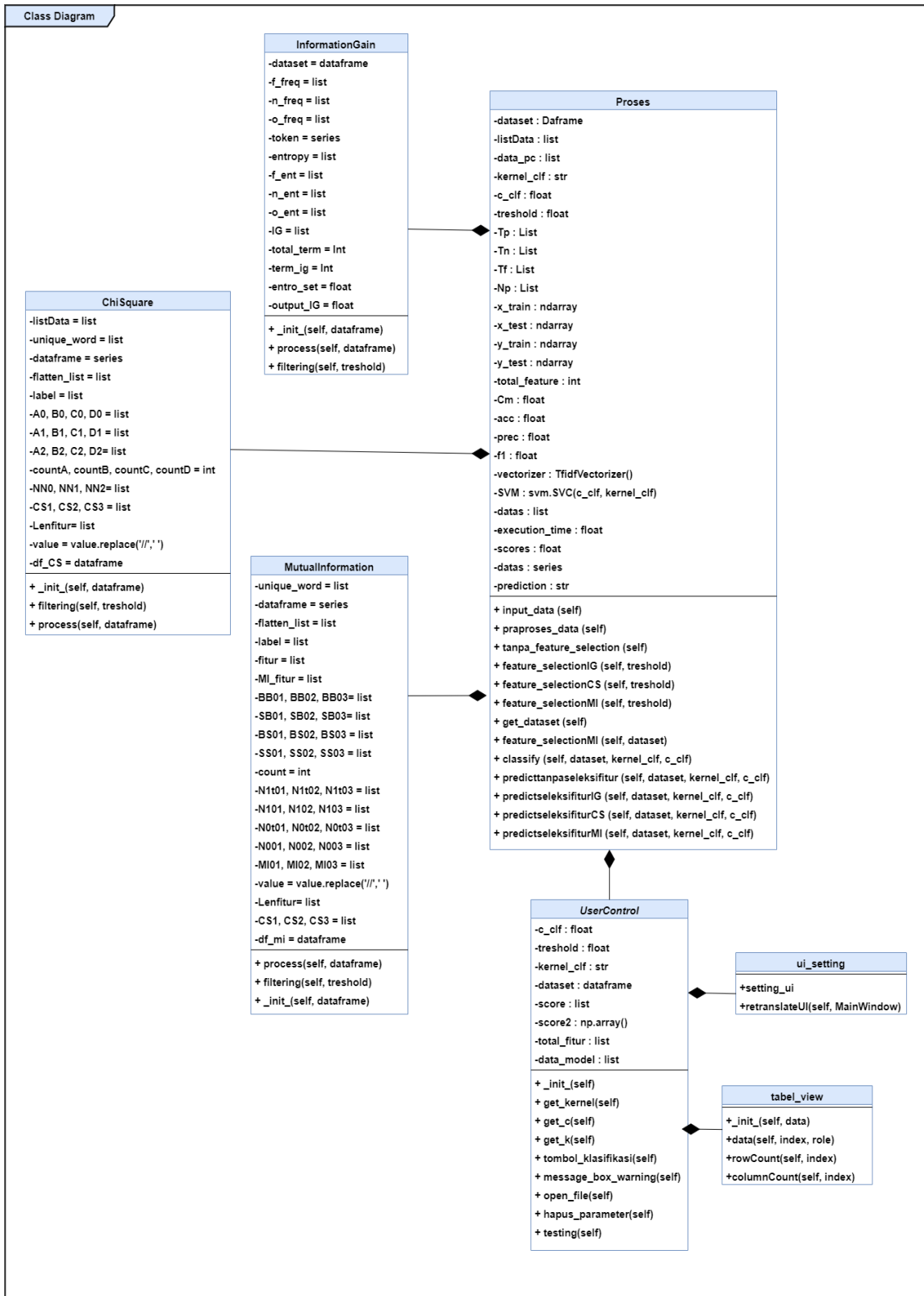
Fase konstruksi merupakan fase yang membahas perihal bagian inti dan fitur lain pada perangkat lunak yang akan dikembangkan. Pada tahapan ini, akan dilakukan beberapa rangkaian proses iterasi berupa proses analisis, desain, implementasi dan pengujian yang mana nantinya akan menghasilkan suatu perangkat lunak yang bisa dipakai sebagai alat penelitian.

4.4.1 Kebutuhan Sistem

Dalam proses pengembangan perangkat lunak pada penelitian ini menggunakan beberapa *library* yang terdapat pada python. *Library* pyqt versi 5 yang dibutuhkan untuk membuat tampilan antarmuka dari perangkat lunak. *Library* nltk yang dibutuhkan untuk proses *tokenizing* dan *untokenized* pada tahapan data *preprocessing*. *Library* math yang digunakan pada proses perhitungan matematika. Kemudian, *library* sklearn yang digunakan pada proses klasifikasi data latih dan data uji.

4.4.2 Diagram Kelas

Diagram kelas menggambarkan struktur dan deskripsi yang ada di tiap kelas, *method*, dan atribut. Diagram kelas untuk pengembangan perangkat lunak pada penelitian ini dapat dilihat pada gambar IV-17.



Gambar IV-17. Diagram Kelas

4.4.3 Implementasi

Tahapan implementasi pada perangkat lunak dalam penelitian ini akan dikembangkan berdasarkan perancangan dan diagram kelas yang telah dibuat pada pembahasan sebelumnya.

4.4.3.1 Implementasi Kelas

Perangkat lunak pada penelitian ini akan dikembangkan menggunakan bahasa pemrograman python berdasarkan diagram kelas yang telah dibuat sebelumnya. Tabel IV-34 menunjukkan penjelasan mengenai detail program yang dibuat berdasarkan setiap kelas yang terdapat di dalam program pada penelitian ini.

Tabel IV-34. Implementasi Kelas

No.	Nama Kelas	Nama File	Keterangan
1.	Ui_Setting	Main.py	Kelas Ui_Setting adalah kelas yang digunakan sebagai kelas <i>parent</i> untuk membuat objek pada <i>interface</i> .
2.	UserControl	Main.py	Kelas UserControl adalah kelas yang bertujuan untuk mengkoneksikan antara objek yang ada pada program dan aktivitas <i>user</i> pada UI.
3.	TabelView	Main.py	Kelas TabelView adalah kelas yang mengatur objek dan fungsi dari tampilan tabel pada <i>interface</i> .

4.	Proses	Proses.py	Kelas proses adalah kelas yang berisikan proses yang dibutuhkan untuk klasifikasi dan pengujian sistem.
5.	InformationGain	InformationGain.py	Kelas InformationGain adalah kelas yang berisikan rangkaian perhitungan <i>Information Gain</i> untuk proses penyeleksian fitur.
6.	ChiSquare	ChiSquare.py	Kelas ChiSquare adalah kelas yang berisikan rangkaian perhitungan <i>Chi Square</i> untuk proses penyeleksian fitur.
7.	MutualInformation	MutualInformation.py	Kelas Mutual Information adalah kelas yang berisikan rangkaian perhitungan <i>Mutual Information</i> untuk proses penyeleksian fitur.

4.4.3.1 Implementasi Antarmuka

Proses implementasi antarmuka dibuat berdasarkan perancangan yang sebelumnya telah dibuat pada tahapan elaborasi. Adapun tampilan pada perangkat lunak ditampilkan pada Gambar IV-18.

Sistem Klasifikasi Pertanyaan Berbahasa Indonesia

Input Data Awal

Tampilan Hasil Data Praproses

Melakukan Klasifikasi Menggunakan Algoritma SVM

Metode : Tanpa Seleksi Fitur Seleksi Fitur IG Seleksi Fitur CS Seleksi Fitur MI

Kernel : linear polynomial rbf

Nilai C : 1 10

Treshold :

Lakukan Proses Klasifikasi

Jumlah Fitur : Selisih Jumlah Fitur :

Pengujian Pertanyaan

Masukkan Pertanyaan

Lakukan Pengujian

Hasil Klasifikasi :

Gambar IV-18. Implementasi Tampilan Antarmuka Perangkat Lunak

4.5 Fase Transisi

Fase Transisi adalah fase akhir dalam proses pengembangan perangkat lunak menggunakan RUP. Dalam fase ini akan dilakukan pengujian terhadap perangkat lunak yang telah dibuat.

4.5.1 Pemodelan Bisnis

Pada tahapan pemodelan bisnis perangkat lunak yang dibangun akan diuji berdasarkan skenario *black box testing* unruk setiap *use case* yang telah dirancang. Skenario pengujian *black box* pada perngkat lunak akan ditampilkan pada tabel IV-38 hingga IV-43.

4.5.2 Rencana Pengujian

Tahapan ini akan membahas mengenai rencana pengujian perangkat lunak yang telah dibangun. Rencana pengujian *black box* pada perangkat lunak akan ditampilkan pada tabel IV-35 hingga IV-40.

1. Rencana Pengujian *Use Case* Melakukan Praproses Data

Tabel IV-35. Rencana Pengujian *Use Case* Melakukan Praproses Data

NO.	ID	Pengujian	Jenis Pengujian	Tingkat Pengujian
1.	UC-1-1	Menekan tombol “Input Data Awal” untuk menampilkan hasil data praproses pada <i>dataset</i> yang telah dipilih	<i>Black box</i>	Pengujian Unit

Tabel IV-36. Rencana Pengujian *Use Case* Melakukan Klasifikasi Menggunakan Algoritma SVM

NO.	ID	Pengujian	Jenis Pengujian	Tingkat Pengujian
2.	UC-2-1	Melakukan proses klasifikasi data dengan memilih metode tanpa seleksi fitur dan memasukkan parameter yang akan digunakan.	<i>Black box</i>	Pengujian Unit

Tabel IV-37. Rencana Pengujian *Use Case* Melakukan Klasifikasi Menggunakan Algoritma SVM dan Seleksi Fitur *Information Gain*

NO.	ID	Pengujian	Jenis Pengujian	Tingkat Pengujian
3.	UC-3-1	Melakukan proses klasifikasi data dengan memilih metode seleksi fitur <i>Information Gain</i> dan memasukkan parameter yang akan digunakan.	<i>Black box</i>	Pengujian Unit

Tabel IV-38. Rencana Pengujian *Use Case* Melakukan Klasifikasi Menggunakan Algoritma SVM dan Seleksi Fitur *Chi-Square*

NO.	ID	Pengujian	Jenis Pengujian	Tingkat Pengujian
4.	UC-4-1	Melakukan proses klasifikasi data dengan memilih metode seleksi fitur <i>ChiSquare</i> dan memasukkan parameter yang akan digunakan.	<i>Black box</i>	Pengujian Unit

Tabel IV-39. Rencana Pengujian *Use Case* Melakukan Klasifikasi Menggunakan Algoritma SVM dan Seleksi Fitur *Mutual Information*

NO.	ID	Pengujian	Jenis Pengujian	Tingkat Pengujian
5.	UC-5-1	Melakukan proses klasifikasi data dengan memilih metode seleksi fitur <i>Mutual Information</i>	<i>Black box</i>	Pengujian Unit

		dan memasukkan parameter yang akan digunakan.		
--	--	---	--	--

Tabel IV-40. Rencana Pengujian *Use Case* Melakukan Proses Pengujian Klasifikasi Pertanyaan Berbahasa Indonesia

NO.	ID	Pengujian	Jenis Pengujian	Tingkat Pengujian
6.	UC-6-1	Melakukan proses pengujian dengan memasukkan data uji berupa kalimat tanya berbahasa Indonesia yang nantinya akan dilakukan proses klasifikasi.	<i>Black box</i>	Pengujian Unit

4.5.3 Implementasi

Pada tahapan ini akan menguraikan mengenai skenario uji yang akan dilakukan mengacu pada rencana pengujian sebelumnya.

4.5.3.1 Pengujian *Use Case* Memasukkan Data

Hasil pengujian terhadap *use case* memasukkan data ditampilkan pada tabel IV-41.

Tabel IV-41. Pengujian *Use Case* Memasukkan Data

ID	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Hasil yang Didapat	Kesimpulan
UC-1-1	Memasukkan Data	Menekan tombol “input data awal”	dataset	Data tersimpan pada system	Perangkat lunak berhasil membaca data masukkan	Terpenuhi

4.5.3.2 Pengujian *Use Case* Melakukan Praproses Data

Hasil pengujian terhadap *use case* melakukan praproses data ditampilkan pada tabel IV-42.

Tabel IV-42. Pengujian *Use Case* Melakukan Praproses Data

ID	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Hasil yang Didapat	Kesimpulan
UC-2-1	Melakukan data <i>preprocessing</i>	Menekan tombol “input data awal”	dataset	Hasil praproses data pada data masukkan	Perangkat lunak berhasil memuat data dan berhasil menampilkan data yang telah dilakukan data <i>preprocessing</i>	Terpenuhi

4.5.3.3 Pengujian *Use Case* Melakukan Proses Klasifikasi Menggunakan Algoritma SVM

Hasil pengujian terhadap *use case* melakukan praproses data ditampilkan pada tabel IV-43.

Tabel IV-43. Pengujian Proses Klasifikasi Menggunakan Algoritma SVM

ID	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Hasil yang Didapat	Kesimpulan
UC-3-1	Melakukan proses klasifikasi menggunakan metode SVM tanpa seleksi fitur	Menekan tombol “Lakukan Proses Klasifikasi”	Pengguna memilih Nilai C dan Kernal	Menampilkan hasil nilai evaluasi pada proses klasifikasi, menampilkan hasil jumlah fitur dan selisih fitur	Menampilkan nilai dari hasil evaluasi yang digunakan pada tabel, menampilkan hasil jumlah fitur dan selisih fitur	Terpenuhi

4.5.3.4 Pengujian *Use Case* Melakukan Proses Klasifikasi Menggunakan Algoritma SVM dan Seleksi Fitur *Information Gain*

Hasil pengujian terhadap *use case* melakukan proses klasifikasi menggunakan algoritma svm dan seleksi fitur *information gain* ditampilkan pada tabel IV-44.

Tabel IV-44. Pengujian Proses Klasifikasi Menggunakan Algoritma SVM dan Seleksi Fitur *Information Gain*

ID	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Hasil yang Didapat	Kesimpulan
UC-4-1	Melakukan proses klasifikasi menggunakan metode SVM dan seleksi fitur <i>information Gain</i>	Menekan tombol “Lakukan Proses Klasifikasi”	Pengguna memilih Nilai C, Kernal dan memasukkan nilai <i>threshold</i>	Menampilkan hasil nilai evaluasi pada proses klasifikasi, menampilkan hasil jumlah fitur dan selisih fitur	Menampilkan nilai dari hasil evaluasi yang digunakan pada tabel, menampilkan hasil jumlah fitur dan selisih fitur	Terpenuhi

4.5.3.5 Pengujian *Use Case* Melakukan Proses Klasifikasi Menggunakan Algoritma SVM dan Seleksi Fitur *Chi Square*

Hasil pengujian terhadap *use case* melakukan proses klasifikasi menggunakan algoritma svm dan seleksi fitur *Chi Square* ditampilkan pada tabel IV-45.

Tabel IV-45. Pengujian Proses Klasifikasi Menggunakan Algoritma SVM dan Seleksi Fitur *Chi Square*

ID	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Hasil yang Didapat	Kesimpulan
UC-5-1	Melakukan proses klasifikasi menggunakan metode SVM dan seleksi fitur <i>Chi Square</i>	Menekan tombol “Lakukan Proses Klasifikasi”	Pengguna memilih Nilai C, Kernel dan memasukkan nilai <i>threshold</i>	Menampilkan hasil nilai evaluasi pada proses klasifikasi, menampilkan hasil jumlah fitur dan selisih fitur	Menampilkan nilai dari hasil evaluasi yang pada tabel, menampilkan hasil jumlah fitur dan selisih fitur	Terpenuhi

4.5.3.6 Pengujian *Use Case* Melakukan Proses Klasifikasi Menggunakan Algoritma SVM dan Seleksi Fitur *Mutual Information*

Hasil pengujian terhadap *use case* melakukan proses klasifikasi menggunakan algoritma svm dan seleksi fitur *Mutual Information* ditampilkan pada tabel IV-46.

Tabel IV-46. Pengujian Proses Klasifikasi Menggunakan Algoritma SVM dan Seleksi Fitur *Mutual Information*

ID	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Hasil yang Didapat	Kesimpulan
UC-6-1	Melakukan proses klasifikasi menggunakan	Menekan tombol “Lakukan	Pengguna memilih Nilai C, Kernel dan	Menampilkan hasil nilai evaluasi pada proses	Menampilkan nilai dari hasil evaluasi yang pada	Terpenuhi

	metode SVM dan seleksi fitur <i>Mutual Information</i>	Proses Klasifikasi”	memasukkan nilai <i>threshold</i>	klasifikasi, menampilkan hasil jumlah fitur dan selisih fitur	tabel, menampilkan hasil jumlah fitur dan selisih fitur	
--	--	---------------------	-----------------------------------	---	---	--

4.5.3.7 Pengujian *Use Case* Melakukan Proses Pengujian Klasifikasi Pertanyaan Berbahasa Indonesia

Hasil pengujian terhadap *use case* melakukan proses pengujian klasifikasi pertanyaan berbahasa Indonesia ditampilkan pada tabel IV-50.

Tabel IV-47. Melakukan Proses Pengujian Klasifikasi Pertanyaan Berbahasa Indonesia

ID	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Hasil yang Didapat	Kesimpulan
UC-7-1	Melakukan proses pengujian pada data uji berupa pertanyaan berbahasa Indonesia	Menekan tombol “Lakukan Pengujian”	Pengguna memilih Nilai C, Kernel dan memasukkan nilai <i>threshold</i>	Menampilkan hasil prediksi pada data uji	Menampilkan hasil prediksi pada sistem	Terpenuhi

4.6 Kesimpulan

Bab ini telah menjelaskan mengenai rangkaian proses pengembangan

perangkat lunak klasifikasi pertanyaan berbahasa Indonesia menggunakan algoritma *Support Vector Machine*. Metode seleksi fitur seperti *Information Gain*, *Chi Square*, *Mutal Information* dan algoritma *Support Vector Machine*. Alur hasil pengembangan perangkat lunak dijelaskan dan diuraikan sehingga menghasilkan perangkat lunak yang sesuai dengan kebutuhan penelitian.