Pebandingan Metode Seleksi Fitur untuk Klasifikasi Pertanyaan Berbahasa Indonesia Menggunakan Algoritma Support Vector Machine (SVM)

by 09021381823120 Syechky Al Qodrin Aruda

Submission date: 24-May-2022 01:12PM (UTC+0700)
Submission ID: 1843070826
File name: SYECHKY_AL_QODRIN_ARUDA_UNIVERSITAS_SRIWIJAYA_-_syechky_al.docx (2.02M)
Word count: 27545
Character count: 129729

BAB I
PENDAHULUAN

1.1 Pendahuluan

Pada bab ini akan membahas mengenai latar belakang, rumusan masalah, tujuan dan manfaat penelitian, batasan masalah serta sistematika penulisan skripsi. Bab ini akan memberikan penjelasan umum mengenai pokok pikiran dari keseluruhan penelitian.

1.2 Latar Belakang

Kalimat tanya merupakan serangkaian kata yang ditujukan untuk memperoleh informasi atau jawaban yang ditujukan kepada seseorang. Untuk mendapatkan suatu jawaban atau informasi yang tepat maka diperlukan suatu analisa dalam memahami kalimat tanya tersebut. Proses klasifikasi dokumen pertanyaan secara otomatis dapat dipergunakan untuk mempermudah melakukan analisa terkait kategori suatu konteks pertanyaan berbahasa Indonesia. Berdasarkan pembagian pada analisa pertanyaan, kalimat tanya sendiri terbagi menjadi 3 jenis pertanyaan yaitu pertanyaan *factoid, non-factoid* dan *other*. Untuk pertanyaan *factoid* didefinisikan sebagai pertanyaan yang hanya membutuhkan jawaban singkat berupa kata atau frasa, seperti nama orang, lokasi, organisasi, jumlah dan tanggal. Untuk pertanyaan *non-factoid* didefinisikan sebagai pertanyaan yang membutuhkan penjelasan seperti pertanyaan yang terkait dengan definisi, alasan, metode dan tatacara (Purwarianti & Yusliani, 2011). Sedangkan untuk pertanyaan diluar dari *factoid* dan *non-factoid* dibuat kategori tersendiri yaitu *other*.

Masalah utama pada klasifikasi teks adalah sebagian besar teks memiliki dimensi fitur yang banyak, sedangkan sebagian besar fitur ini tidak relevan dan bahkan mengandung *noise* yang dapat mengurangi tingkat akurasi klasifikasi (Chandra, 2019). Oleh karena itu *feature selection* umumnya digunakan dalam klasifikasi teks untuk mengurangi dimensi ruang fitur dan meningkatkan efisiensi dan akurasi pengklasifikasi. Terdapat beberapa metrik seleksi fitur yang terkenal seperti *Information Gain* (IG), *Chi-Square*, dan *Mutual Information* (Khan et al., 2010).

Tentunya masing-masing metode seleksi fitur tersebut memiliki kelebihan dan kekurangan tersendiri. Menurut (Rahmad & Pribadi, 2015), Metode *Chi-Square* dapat meningkatkan hasil kinerja klasifikasi seperti *Recall, Precision, F-measure*, dan *Accuracy*. Metode ini bekerja dengan melakukan pehitungan statistic dalam pemilihan seleksi fitur. Namun, pada metode *Chi-Square* penggunaan *treshold* sangat berpengaruh terhadap jumlah fitur yang didapat. (Maulida et al., 2016) dalam penelitiannya menjelaskan, metode *Information Gain* dapat mengurangi dimensi fitur pada dokumen berbahasa Indonesia dan metode ini menerapkan teknik *scoring* dalam melakukan pembobotan menggunakan maksimal *entropy*. Selain itu, (Irham et al., 2019) dalam penelitiannya menjelaskan metode *Mutual Information* membantu mengeliminasi fitur yang tidak menginterpretasikan sebuah kelas. Sehingga metode ini dapat meningkatkan kecepatan dan efektifitas hasil kinerja dalam klasifikasi.

Terdapat beberapa penelitian yang telah dilakukan sebelumnya berkaitan dengan komparasi fitur seleksi pada klasifkasi dokumen. Dalam penelitian

(Chandani, 2015) peneliti melakukan perbandingan algoritma *machine learning* (*Artificial Neural Network* (ANN), *Support Vector Machine* (SVM), *Naïve Bayes* (NB)) dan juga membandingkan seleksi fitur (*Information Gain, Chi-Square, Forward Selection dan Backward Selection*). Hasil dari penelitian tersebut didapati bahwa SVM sebagai algoritma *machine learning* terbaik dan *Information Gain* sebagai metode seleksi fitur terbaik. Selain itu, (Bahassine et al., 2020) juga melakukan perbandingan terhadap algoritma *machine learning* (*Decision Tree* dan SVM) dan metode seleksi fitur (ImpCHI, *Chi-Square*, IG, MI) untuk klasifikasi dokumen berbahasa arab. Hasil penelitian menunjukkan bahwa penggunaan seleksi fitur *Improvement Chi-Square* menunjukkan hasil kinerja terbaik diikuti *Information Gain* dan *Chi-Square*. Kemudian juga didapati bahwa algoritma SVM menunjukkan hasil kerja yang lebih baik dibandingkan dengan algoritma *Decision Tree*.

Berdasarkan uraian dan referensi penelitian sebelumnya, maka dari itu, akan dilakukan perbandingan terhadap 3 metode seleksi fitur yaitu *Information Gain, Chi-Square,* dan *Mutual Information* pada klasifikasi pertanyaan berbahasa Indonesia menggunakan algoritma *Support Vector Machine* untuk mengetahui perbandingan hasil kinerja tiga metode seleksi fitur tersebut. Hasil penelitian ini diharapkan dapat menjadi rujukan bagi pengembang sistem pertanyaan berbahasa Indonesia dan memberikan solusi dalam melakuan optimalisasi hasil kinerja pada klasifikasi pertanyaan berbahasa Indonesia menggunakan algoritma klasifikasi *Support Vector Machine*.

1.3 Rumusan Masalah

Berdasarkan latar belakang permasalahan yang telah dijabarkan maka rumusan masalah dari penelitian ini:

1. Bagaimana mengembangkan sistem klasifikasi pertanyaan berbahasa Indonesia menggunakan algoritma SVM ?
2. Bagaimana kinerja metode seleksi fitur *Information Gain* dalam melakukan pemilihan fitur pada sistem pengklasifikasian pertanyaan berbahasa Indonesia menggunakan seleksi fitur *Information Gain*?
3. Bagaimana kinerja metode seleksi fitur *Chi-Square* dalam melakukan pemilihan fitur pada sistem pengklasifikasian pertanyaan berbahasa Indonesia menggunakan seleksi fitur *Chi-Square*?
4. Bagaimana kinerja metode seleksi fitur *Mutual Information* dalam melakukan pemilihan fitur pada sistem pengklasifikasian pertanyaan berbahasa Indonesia menggunakan seleksi fitur *Mutual Information* ?

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut :

1. Menghasilkan perangkat lunak sistem klasifikasi pertanyaan Berbahasa Indonesia menggunakan algoritma SVM.
2. Mengetahui kinerja *Information Gain* dalam melakukan seleksi fitur pada sistem klasifikasi pertanyaan berbahasa Indonesia.
3. Mengetahui kinerja *Chi-Square* dalam melakukan seleksi fitur pada sistem klasifikasi pertanyaan berbahasa Indonesia.

4. Mengetahui kinerja *Mutual Information* dalam melakukan seleksi fitur pada sistem klasifikasi pertanyaan berbahasa Indonesia.
5. Mengetahui perbandingan dari kinerja *Information Gain*, *Chi-Square*, dan *Mutual Information* dalam melakukan seleksi fitur pada sistem klasifikasi pertanyaan berbahasa Indonesia.

1.5 Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah sebagai berikut :

1. Mendapatkan informasi mengenai kinerja dan hasil klasifikasi pertanyaan berbahasa Indonesia dengan membandingkan seleksi fitur *Information Gain, Chi-Square, Mutual Information* pada metode klasifikasi *Support Vector Machine*.
2. Mengetahui metode yang paling optimal dalam melakukan klasifikasi pertanyaan berbahasa Indonesia.
3. Hasil penelitan dapat menjadi rujukan penelitian yang relevan.

1.6 Batasan Masalah

Adapun batasan masalah pada penelitian ini adalah sebagai berikut :

1. Data yang digunakan merupakan data pertanyaan berbahasa Indonesia.
2. Data pertanyaan berbahasa Indonesia yang telah dikumpulkan berjumlah 1.195 pertanyaan berbahasa Indonesia yang terbagi menjadi 553 data berlabel '*factoid*', 451 data berlabel '*non-factoid*', dan 185 data berlabel '*others*'.
3. Klasifikasi terdiri dari 3 kelas, yaitu *factoid, non-factoid* dan *others*.

1.7 Sistematika Penulisan

Adapun sistematika penulisan pada penelitian ini adalah sebagai berikut:

BAB I. PENDAHULUAN

Bab ini akan membahas landasan dari penelitian, seperti latar belakang, rumusan masalah, tujuan dan manfaat penelitian, batasan masalah serta sistematika penulisan.

BAB II. KAJIAN LITERATUR

Bab ini membahas dasar-dasar teori yang berkaitan dengan penelitian ini, seperti pertanyaan Berbahasa Indonesia, jenis pertanyaan *factoid* dan *non-factoid*, seleksi fitur *Information gain*, seleksi fitur *Chi-square*, seleksi fitur *Mutual Information* dan algoritma *Support Vector Machine*, serta membahas beberapa penelitian yang relevan.

BAB III. METODOLOGI PENELITIAN

Bab ini membahas mengenai tahapan alur penelitian. Diantaranya pengumpulan data dan perancangan perangkat lunak yang akan dibangun. Selanjutnya tahapan penelitian dijelaskan secara detail merujuk pada kerangka kerja yang dibuat.

1.8 Kesimpulan

Pada Bab ini menguraikan dasar dan patokan pada penelitian, seperti latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah dan sistematika penulisan.

BAB II
KAJIAN LITERATUR

2.1 Pendahuluan

Bab ini akan menguraikan teori yang melandasi penelitian. Uraian teori tersebut terdiri dari deskripsi mengenai pertanyaan, klasifikasi teks, seleksi fitur, algoritma yang digunakan dalam penelitian yakni seleksi fitur *Information Gain*, *Chi-Square* dan *Mutual Information*, beberapa penelitian yang relevan, serta kesimpulan.

2.2 Landasan Teori

2.2.1 Pertanyaan

Pertanyaan merupakan suatu cara yang dilakukan seseorang dalam rangka memperoleh informasi atau mengkonfirmasi suatu hal melalui ekspresi yang disampaikan lewat suatu kalimat tanya baik secara lisan maupun tulisan (Sudin et al., 2019). Sebagai suatu frasa, kalimat tanya khususnya kalimat tanya berbahasa Indonesia tentu memiliki ciri tersendiri diantaranya, yakni :

1. Kalimat tanya selalu diakhiri dengan simbol berupa tanda baca tanya atau ‘?’.
2. Kalimat tanya biasanya diawali dengan kata tanya berupa 5W+1H diantaranya apa, kapan, dimana, mengapa, siapa dan bagaimana.
3. Kalimat tanya sering menggunakan imbuhan –kah pada kata tanya seperti siapakah, dimanakah dan sebagainya.
4. Kalimat tanya yang tidak diawali dengan kata tanya dapat menambahkan imbuhan –kan diakhir kalimat seperti “Kamu belum mandi, iya kan ?”.

5. Kalimat tanya tentunya membutuhkan tanggapan atau jawaban yang jelas terkait informasi yang diberikan.

Berdasarkan jawaban atau informasi yang diberikan, kalimat tanya juga dapat diklasifikasikan menjadi beberapa jenis pertanyaan, diantaranya: :

a) Pertanyaan *Factoid*

Pertanyaan *factoid* merupakan suatu pertanyaan yang hanya membutuhkan jawaban berupa fakta singkat dan ringkas seperti entitas, angka dan nama. Contohnya pertanyaan *factoid*, ialah:

1. "kucing jenis apa yang berbulu tebal ?"
2. "dimana kamu membeli baju tersebut ?"

b) Pertanyaan *Non-Factoid*

Pertanyaan *non-factoid* merupakan suatu pertanyaan yang membutuhkan jawaban berisi fakta yang cukup panjang disertai dengan penjelasan terkait suatu hal. Contoh dari pertanyaan *non-factoid*, seperti:

1. "bagaimana kau bisa tiba disini begitu cepat ?"
2. "lantas mengapa kau tak menyampaikan yang sebenarnya ?"

c) Pertanyaan *Others*

Pertanyaan *others* merupakan suatu pertanyaan yang membutuhkan jawaban diluar dari pertanyaan *factoid* dan *non-factoid* seperti pertanyaan dengan jawaban bertipe *yes-no*, list, dan opini. Contoh dari pertanyaan *others*, ialah:

1. "tahukah kamu dimana fakultas teknik ?"
2. "apa saja barang yang ingin kau jual kepadaku ?"

2.2.2 Klasifikasi Teks

Klasifikasi teks merupakan bagian penelitian dalam bidang *Information Retrieval* yang mengembangkan suatu cara dalam melakukan pengelompokkan dan penentuan suatu dokumen teks kedalam satu kelas atau lebih secara otomatis (Februariyanti & Zuliarso, 2012). Klasifikasi teks merupakan suatu pekerjaan yang ditujukan untuk menentukan apakah suatu teks dokumen termasuk dalam kategori tertentu (Suharno et al., 2017).

Pra-pengolahan (*preprocessing*) adalah rangkaian awal yang dilakukan dalam pemrosesan teks dengan tujuan untuk mengolah, mengatur dan menggali data yang digunakan agar data tersebut menjadi lebih tersetruktur sehingga *noise* yang ada pada dataset dapat berkurang. Hal ini penting karena akan mempengaruhi hasil akhir dari proses klasifikasi nantinya. Rangkaian tahapan *preprocessing* juga terdiri dari beberapa tahapan seperti *Case Folding* yang berfungsi untuk mengkonversi teks agar menjadi bentuk yang selaras, *Tokenizing* yang digunakan untuk memecah kalimat yang ada teks menjadi satu sukuan kata, *Filtering* yang dapat mereduksi variable yang tidak mempunyai korelasi dengan dokumen yang digunakan, dan *Stemming* yang ditujukan untuk memperkecil jumlah indeks yang berbeda pada suatu dokumen dengan mengembalikan suatu kata ke kata dasarnya (Rahman et al., 2021).

Perhitungan pembobotan Fitur TF- IDF dapat menggunakan rumus yang ada pada persamaan II-1 dan II-2.

$$W_{ij} = TF(i, j) \text{x } IDF \qquad \text{(II-1)}$$

Rumus mencari nilai *Inverse Document Frequency* (IDF) menggunakan persamaan II-2.

$$\text{IDF} = \log\left(\frac{N}{\text{DF}(j)}\right) + 1 \qquad \text{(II-2)}$$

Keterangan :

- N : Banyaknya dokumen
- W*ij* : Bobot nilai
- TF(*i*, *j*) : Jumlah kemunculan setiap kata *i* dalam sebuah dokumen *j*.
- DF(*i*) : Jumlah dokumen yang mengandung kata *i*.

Setelah dilakukan proses pembobotan fitur maka dapat dilakukan proses klasifikasi yang bertujuan untuk membagi setiap entitas kedalam kelas yang telah ditentukan. Pengklasifikasian sendiri terbagi menjadi dua jenis yaitu *supervised* dan *unsupervised*. Perbedaan keduanya terletak di target kelas, apabila target kelasnya telah ditentukan maka termasuk klasifikasi *supervised*. Jika target kelas belum ditentukan pada dataset maka termasuk klasifikasi *unsupervised* (Hanati & Sari, 2021).

Untuk mendapatkan hasil kinerja yang lebih maksimal, dapat ditambahkan proses seleksi fitur setelah dilakukan tahapan pra-pengolahan teks yang berguna untuk menemukan fitur terbaik sebelum dilakukan proses pembobotan fitur dan klasifikasi teks (Ridok & Latifah, 2015).

2.2.3 Seleksi Fitur

Seleksi Fitur ialah salah satu dari serangkaian tahap preprocessing ketika tahapan *stemming* dan penggunaan *stopword list* masih dirasa kurang dalam mereduksi dimensi (Suharno et al., 2017). Cara kerja seleksi fitur ialah dengan menentukan fitur yang relevan dan melakukan pembobotan pada setiap fitur. Seleksi fitur sendiri bertujuan untuk meningkatkan efisiensi dan efektifitas hasil kinerja dari algortima klasifikasi dengan mengurangi fitur yang tidak relevan dan banyaknya dimensi data (Buani, 2021).

Seleksi fitur terbagi menjadi dua jenis metode yaitu yaitu *filter* dan *wrapper*. Metode *wrapper* melakukan interaksi dengan algoritma klasifikasi dalam menentukan kegunaan suatu fitur sehingga menghasilkan *sub* fitur dengan kinerja yang lebih baik. Namun, cenderung lebih lambat jika dibandingkan dengan metode *filter* (Tanti et al., 2021). Metode filter sendiri memerlukan *confusion matrix* untuk melakukan evaluasi agar dapat mengukur kemampuan fitur yang membedakan setiap kelas. Contoh dari penggunaan metode filter yang umum digunakan dalam proses klasifikasi teks ialah *Document Frequency*, *Mutual Information, Information Gain*, dan C*hi-Square* (Khan et al., 2010).

2.2.4 *Information Gain*

Information Gain merupakan salah satu metode yang digunakan untuk proses seleksi fitur. Nilai suatu *Information Gain* didefinisikan menggunakan *entropy*. Suatu *entropy* menunjukkan berapa banyak informasi yang dibutuhkan saat akan melakukan kode terhadap suatu kelas. *Information Gain* bekerja dengan cara pemberian *scoring* pada tiap nominal atau dengan cara pemberian bobot berupa

atribut yang bersifat kontinu kemudian didiskretkan menggunakan nilai maksimal *entropy*. *Information Gain* pada suatu *term* didapat dengan cara menghitung jumlah bit informasi melalui prediksi kategori muncul atau tidaknya suatu term pada suatu dokumen (Maulida et al., 2016).

Information Gain ialah algoritma yang digunakan untuk menentukan batas yang digunakan pada atribut yang tersedia. *Information Gain* melambangkan kualitas atribut yang akan digunakan (Prakoso et al., 2019). Secara umum *Information Gain* terdiri dari tiga tahapan yakni menghitung nilai *Information Gain* setiap atribut yang ada di dataset, menghilangkan atribut yang tidak memenuhi ambang batas, dan memperbaiki atribut dataset yang memenuhi nilai ambang (Bramer, 2007).

Tahapan dalam menghitung bobot nilai IG diantaranya :

1. Hitung bobot nilai entropy dataset melalui persamaan II-20.

$$Entropy(D) = -\sum_{i=1}^{m} P_i \log 2\ (P_i) \qquad \text{(II-20)}$$

Keterangan:

- m : jumlah partisi D
- D : himpunan kasus
- pi : proposi dari Di terhadap D

2. Hitung entropy untuk tiap atribut yang didefinisikan dengan A melalui persamaan II-21.

$$Entropy_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} . Entropy\ (D_j) \qquad \text{(II-21)}$$

Keterangan:

- A : fitur
- D : himpunan kasus
- $|D_j|$: jumlah sampel data pada partisi ke j
- $|D|$: jumlah sampel data
- v : jumlah partisi fitur A
- Entropy(D_j) : total entropy dalam partisi

3. Kemudian langkah terakhir untuk menghitung nilai IG atribut A ialah menggunakan persamaan II-22.

$$Gain(A) = Entropy(D) - |Entropy_A(S)| \qquad \text{(II-22)}$$

Keterangan :

- $Gain(A)$: nilai informasi fitur A
- $Entropy_A(D)$: total *entropy*
- $Entropy_A(S)$: *entropy* A

Setelah mendapatkan bobot nilai akhir *Information Gain* pada tiap atribut maka langkah terakhir ialah dengan memfilter dan mengambil nilai bobot suatu atribut yang memenuhi ambang batas.

2.2.5 *Chi-Square*

Chi-Square merupakan suatu metode seleksi fitur yang digunakan untuk menghilangkan atribut yang kurang relevan pada proses klasifikasi. Seleksi fitur *Chi-Square* mengimplementasikan teori statiska dalam menguji tingkat ipendensi suatu kata berdasarkan kategori dari kata tersebut (Amrullah et al., 2020). Perhitungan *Chi-Square* sendiri terdiri dari beberapa tahapan. Tahapan pertama ialah dengan membuat suatu table kontingensi yang menunjukan relevansi antara suatu *term* dengan kelas yang ada seperti berikut ini.

Tabel II-1. Tabel Relevansi Antara Suatu Kata dan Kelas Kata

Kelas	**Term**	
	t	Not t
c	A	D
Not t	B	C

Kemudian menghitung fungsi *Chi-Square* menggunakan persamaan II-23.

$$x^2(t,c) = \frac{N(AD-CB)^2}{(A+C)(B+D)(A+B)(C+D)} \quad \text{(II-23)}$$

Keterangan :

x^2 : Score *Chi-Square*

t : *Term*

c : Kelas

N : Jumlah dokumen latih

A : Total dokumen pada kategori c yang terdapat *term* t

B : Total dokumen pada kategori selain c yang terdapat *term* t

C : Total dokumen pada kategori c yang tidak terdapat *term* t

D : Total dokumen pada kategori selain c yang tidak terdapat *term* t

Setelah mendapatkan nilai *Chi-Square* suatu term pada tiap kategori kelas kata maka diperlukan nilai *Chi-Square* tunggal sebagai skor akhir, dengan memilih satu nilai tunggal yang terbesar diantara nilai *Chi-Square* dikelas kata lainnya. Kemudian skor akhir tersebut akan diurutkan dari yang tertinggi untuk dilakukan proses seleksi fitur (Bahassine et al., 2020).

2.2.6 *Mutual Information*

Mutual Information merupakan salah satu metode yang digunakan untuk melakukan seleksi fitur. MI bekerja dengan cara menghitung banyaknya informasi yang ada pada suatu term, dan kontribusi yang diberikan oleh suatu term dalam menentukan hasil keputusan pada proses klasifikasi suatu term pada kelas kata secara benar (Hanafi et al., 2020).

Untuk menghitung nilai dari MI pada *term* untuk tiap kategori kelas kata maka dapat menggunakan persamaan II-24.

$$I(U,C) = \sum_{et\in\{1,0\}} \sum_{ec\in\{1,0\}} P(U = et,\ C = ec)\ log_2 \frac{P(U=et, C=ec)}{P(U=et)\ P(C=ec)} \qquad \text{(II-24)}$$

Persamaan II-24 dapat dijabarkan kembali menjadi persamaan II-25.

$$I(U,C) = \frac{N_{11}}{N} log_2 \frac{N.N_{11}}{N_1.N_1} + \frac{N_{01}}{N} log_2 \frac{N.N_{01}}{N_0.N_1} + \frac{N_{10}}{N} log_2 \frac{N.N_{10}}{N_1.N_0} + \frac{N_{00}}{N} log_2 \frac{N.N_{00}}{N_0.N_0} \quad \text{(II-25)}$$

Keterangan :

N : Jumlah dokumen yang memiliki et dan ec atau ($N = N_{00} + N_{01} + N_{10} + N_{11}$).

$N_{1.}$: Jumlah dokumen yang memiliki et atau ($N_{1.} = N_{10} + N_{11}$).

N_1 : Jumlah dokumen yang memiliki ec atau ($N_1 = N_{01} + N_{11}$).

$N_{0.}$: Jumlah dokumen yang tidak memiliki et atau ($N_{0.} = N_{01} + N_{00}$).

N_0 : Jumlah dokumen yang tidak memiliki ec atau ($N_0 = N_{10} + N_{00}$).

Setelah dilakukan proses perhitungan nilai MI pada suatu kelas, maka selanjutnya ialah menghitung nilai MI pada kelas lain dengan cara yang sama. Kemudian hasil nilai MI pada tiap kelas dibandingkan, nilai MI yang terbesar akan disimpan sebagai hasil akhir. Nilai akhir MI yang didapat pada masing-masing fitur akan diurutkan dari yang terbesar hingga terkecil (Irham et al., 2019).

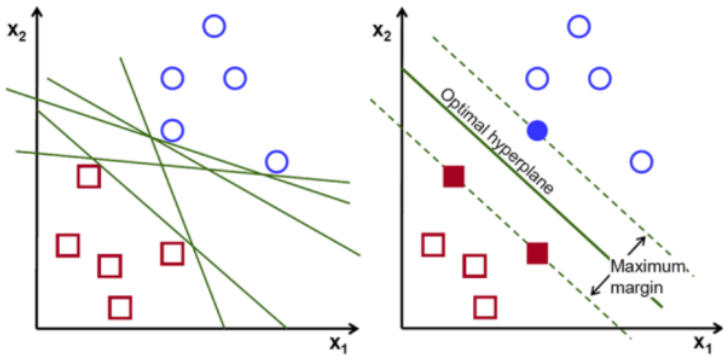
2.2.7 *Support Vector Machine* (SVM)

Support Vector Machine (SVM) merupakan suatu algortima *supervised learning* yang cukup populer digunakan untuk pengklasifikasian. Algoritma SVM bekerja dengan cara mencari *hyperlane* dengan *margin* yang terbesar. Margin merupakan jarak diantara *support vector* dan *hyperlane*. Dengan memaksimalkan nilai margin, dapat membuat *hyperlane* menjadi lebih baik (Yulietha et al., 2017). Sebagai algoritma klasifikator, SVM termasuk algoritma yang tergolong biner,

linier dan probabilistik. SVM menentukan hasil klasifikasi dari data *training* menggunakan *decision boundary* atau batas keputusan. *Decision boundary* bertujuan untuk mencari sebuah model linear atau *hyperlane* teroptimal dalam proses klasifikasi (Mutawalli et al., 2019).

SVM membagi data yang telah diketahui sebelumnya berdasarkan klasifikasi untuk menguji tingkat keakuratan data di sebuah sistem. Dalam penerapannya algoritma SVM digunakan untuk membagi data yang bersifat *non-linear*. Kemudian SVM akan membaginya kedalam *hyperline* yang memisahkan titik *vector*. (Tuhenay, 2021).

Secara umum cara kerja dari SVM dapat tergambarkan pada Gambar II-1 dibawah ini.



Gambar II-1. Ilustrasi Pola SVM

Untuk menghitung persamaan garis *hyperlane* dapat menggunakan persamaan II-3.

$$wi.\ x + b = 0 \qquad \text{(II-3)}$$

Apabila data termasuk kelas positif maka dapat digunakan rumus persamaan II-4.

$$wi.\ x + b \geq +1 \qquad \text{(II-4)}$$

Untuk data yang termasuk kelas negative dapat menggunakan rumus persamaan II-5.

$$wi.\ x + b \leq -1 \qquad \text{(II-5)}$$

Dengan adanya dua garis pemisah tersebut maka dapat menghasilkan persamaan II-6.

$$y_i(\vec{x_i}.\vec{w} + b - 1) \geq 0 \qquad \text{(II-6)}$$

Margin yang terletak diantara *hyperlane* dan garis pemisah dapat dirumuskan dengan $\frac{1}{||w||} + \frac{1}{||w||} = \frac{2}{||w||}$. Nilai margin dapat dimaksimalkan dengan cara meminimalkan nilai dari ||w|| sebagai penyebut. *Quadratic Programming* merupakan suatu cara untuk mencari titik minimal dari ||w||, dengan persamaan II-7.

$$min\frac{1}{2}||w||^2 \qquad \text{(II-7)}$$

Salah satu teknik permasalahan *Quadratic Programming* yang dapat digunakan adalah *Lagrange Multiplier*. *Lagrange Multiplier* dapat didefinisikan dengan persamaan II-8.

$$L(\vec{w}, b, a) = \frac{1}{2}||w||^2 - \sum_{i=1}^{n} a_i(y_i((\vec{x_i}.\vec{w} + b) - 1) \qquad \text{(II-8)}$$

$$i = 1,2,\ldots,n$$

Lagrange Multipliers memiliki koefisien dengan nilai nol atau positif ($a_i \geq 0$). Nilai maksimal dari *Lagrange Multipliers* dapat diketahui dengan

meminimalkan nilai variable $\overrightarrow{w}$ dan b kemudian memaksimalkan nilai variable a_i melaui persamaan II-9 dan II-10.

$$L(a) = \sum_{i=1}^{l} a_i - \frac{1}{2} \sum_{ij=1}^{n} a_i\, a_j y_i y_j \overrightarrow{x_i}\, \overrightarrow{x_j} \qquad \text{(II-9)}$$

Dengan syarat :

$$a_i \geq 0, (i = 1,2, \dots, n) \sum_{i=1}^{n} a_i y_i = 0$$

Keterangan :

n : Jumlah data

x : Data

y : Kelas data

Penggunaan persamaan diatas hanya dapat digunakan ketika diperkirakan *hyperlane* dapat memisahkan dua kelas yang terpisah secara *linear* dengan sempurna. Akan tetapi, nyatanya tidak semua kelas dapat dipisahkan oleh *hyperlane* secara linear (*non linear separable)*. Sehingga proses optimasi pada persamaan (II-6) tidak dapat dilakukan karna batasan tidak berlaku.

Permasalahan tersebut dapat diatasi dengan teknik *softmargin*. Teknik *softmargin* dapat bekerja dengan cara memodifikasi persamaan yang ada pada persamaan (II-6) dan (II-7) dengan menambah variable *slack* ξi $(\xi i > 0)$ sebagai berikut :

$$y_i(\overrightarrow{x_i}.\, \overrightarrow{w} + b - 1) \geq 1 - \xi_i, \forall_i \qquad \text{(II-10)}$$

$$\frac{1}{2}||w||^2 + C \sum_{i=1}^{n} \xi_i \qquad \text{(II-11)}$$

Variable C digunakan untuk mengontrol penggunaan *trade off* antara *error* klasifikasi ξ dan *margin*. Penggunaan variable C yang semakin besar akan memberikan penalti untuk kesalahan menjadi semakin besar. Dengan menambahkan variable C dan *slack* ξ_i akan memodifikasi persamaan (II-9) menjadi bentuk persamaan II-12.

$$0 \leq \alpha_i \leq C, (i = 1,2,..,n) \qquad \text{(II-12)}$$

Proses algoritma pelatihan dapat dilakukan dengan persamaan II-13 sebagai berikut.

$$L(a) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} a_i a_j y_i y_j \mathrm{K}(x_i, x_j) \qquad \text{(II-13)}$$

Dengan syarat.

$$0 \leq a_i \leq C, \ (i = 1,2,..,n), \ \sum_{i=1}^{n} \alpha_i y_i = 0 \qquad \text{(II-14)}$$

Fungsi f(x) sebagai fungsi pemisah optimal didefinisikan melalui persamaan II-15 dan II-16.

$$f(x) = sgn(\sum_{i=1}^{nsv} \alpha_i \, y_i K(x_i, x_d) + b) \qquad \text{(II-15)}$$

$$b = \frac{1}{NSV} \sum_{x_j \in SV} (\frac{1}{y_i} - \sum_{x_j \in SV} (\alpha_j y_j K \, (x_j, x_i))) \qquad \text{(II-16)}$$

Keterangan :

a : Nilai *Lagrange Multiplier* data *support vector*

nsv : Jumlah *support vector*

x : Data *support vector*

b : Bias

y : Kelas data

x_d : Data uji

$K(x_i, x_j)$: Fungsi kernel

sgn ( ) : Fungsi untuk menentukan tanda bilangan riil

Fungsi *kernel* SVM yang sering digunakan diantaranya :

- Linear

$$K(x_i, x) = x_i{}^T . x \quad \text{(II-17)}$$

- Polinomial

$$K(x_i, x) = (\gamma(x_i{}^T . x) + 1)^d \quad \text{(II-18)}$$

- *Radial Basis Function* (RBF)

$$K(x_i, x) = \exp(-\gamma||x_i - x||^2) \quad \text{(II-19)}$$

Keterangan :

γ : Nilai gamma

d : Derajat Polinom

2.2.8 *Multiclass* SVM

Support Vector Machine diketahui hanya dapat melakukan klasifikasi data kedalam dua kelas. Untuk dapat melakukan klasifikasi menjadi lebih dari dua kelas maka dapat dilakukan dengan dua metode pendekatan yaitu *one againts one* vs *one againts all*. Metode *one against one* terdiri dari $\left(\frac{n(n-1)}{2}\right)$ buah model SVM. Sedangkan metode *one againts all* terdiri dari *n* buah model, yang mana *n* merupakan banyaknya kelas (Saniyah, 2019).

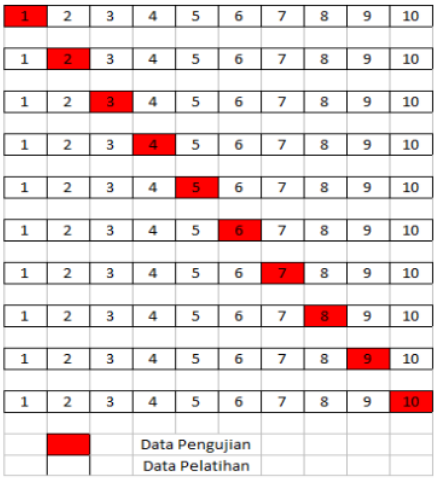
Penelitian ini akan menggunakan metode *one against one* untuk mengklasifikasi kelas pada dokumen pertanyaan berbahasa Indonesia. Setiap model

klasifikasi didapatkan melalui proses pelatihan terhadap dua kelas. Kemudian, data uji dimasukkan ke dalam fungsi keputusan $f(x)$. Setelah model klasifikasi telah selesai dibangun maka dapat dilakukan metode voting (Alita et al., 2020). Ketika hasil voting menunjukkan data suatu dokumen termasuk kedalam kelas *i* maka jumlah *vote* terhadap kelas i ditambahkan satu. Hasil klasifikasi ditentukan berdasarkan jumlah hasil voting terbanyak terhadap semua model yang telah dibangun.

2.2.9 *K-Fold Cross Validation*

Salah satu metode yang digunakan untuk mengecek terjadinya *overfitting* pada model yang digunakan ialah dengan *k-fold cross validation*. *Overfitting* sendiri terjadi apabila terdapat penyimpangan yang cukup jauh pada prediksi suatu data. Data yang dibagi menjadi *k* bagian mengizinkan setiap bagian data berhenti untuk memprediksi data lebih cepat dibandingkan jika tidak dibagi terlebih dahulu (Nasution & Hayaty, 2019).

Pada penggunaan metode *k-fold cross validation,* kumpulan data dibagi sebanyak *k* buah bagian secara acak. Kemudian, dilakukan sebanyak *k*-kali percobaan dengan menggunakan data bagian ke-*k* sebagai data pengujian dan menggunakan sisa bagian lain sebagai data pelatihan. Percobaan yang dilakukan menyesuaikan dengan jumlah bagian yang dilakukan (Supartini et al., 2017)



Gambar II-2. Ilustrasi *k-fold cross validation*

2.2.10 *Confusion Matrix*

Dalam mengukur hasil dan mengevaluasi hasil kinerja metode klasifikasi ada suatu metode yang dikenal dengan *Confusion Matrix* atau sering juga disebut table *confusion*. *Confution Matrix* dapat merepresentasikan kelas aktual dan hasil prediksi dari model klasifikasi. Pada penelitian ini menggunakan *Confusion Matrix* yang terdiri dari tiga kelas yaitu kelas *factoid*, *non-factoid* dan *others*. Tiga kelas yang terdapat pada *confusion matrix* dapat membentuk *matrix* dengan ukuran 3x3 dan menghasilkan sembilan sel *matrix* (Irmanda & Astriratma, 2020). Model *Confusion Matrix* 3x3 dapat di ilustrasikan seperti pada tabel II-2.

Tabel II-2. Model *Confusion Matrix*

Fakta	**Prediksi**		
	A	**B**	**C**
A	TA	FB1	FC1
B	FA1	TB	FC2
C	FA2	FB2	TC

Keterangan *confusion matrix* :

TA : Kelas A yang diklasifikasikan dengan benar

TB : Kelas B yang diklasifikasikan dengan benar

TC : Kelas C yang diklasifikasikan dengan benar

FA1 : Kelas B yang diklasifikasikan kedalam kelas A

FA2 : Kelas C yang diklasifikasikan kedalam kelas A

FB1 : Kelas A yang diklasifikasikan kedalam kelas B

FB2 : Kelas C yang diklasifikasikan kedalam kelas B

FC1 : Kelas A yang diklasifikasikan kedalam kelas C

FC2 : Kelas B yang diklasifikasikan kedalam kelas C

Setelah matrix 3x3 dibentuk, ada beberapa kriteria performa yang dapat diukur dalam *confusion matrix* yaitu *Accuracy, Recall, Precision*, dan *F-measure*. Berikut beberapa persamaan untuk kriteria performa tersebut.

a. *Accuracy* bertujuan untuk menghitung tingkat ketepatan suatu model dalam melakukan klasifikasi data. Perhitungan *accuracy* ditunjukkan pada persamaan II-26.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad \text{(II-26)}$$

b. *Recall* bertujuan untuk menghitung kemampuan model dalam menemukan informasi yang berhubungan. Perhitungan nilai *recall* ditunjukkan pada persamaan II-27.

$$Recall = \frac{TP}{TP+FN} \quad \text{(II-27)}$$

c. *Precision* merupakan perbandingan antara total dokumen yang saling berhubungan dengan total dokumen keseluruhan yang digunakan dalam model klasifikasi. Perhitungan nilai *precision* ditunjukkan pada persamaan II-28.

$$Precision = \frac{TP}{TP+FP} \quad \text{(II-28)}$$

d. *F-Measure* bertujuan untuk menunjukkan keseimbangan yang terjadi antara nilai *recall* dan nilai *precision*. Perhitungan nilai *F-measure* ditunjukkan pada persamaan II-29.

$$F-measure = \frac{2\ X\ (Precision\ X\ Recall)}{Precision+Recall} \quad \text{(II-29)}$$

2.2.11 *Rational Unified Process*

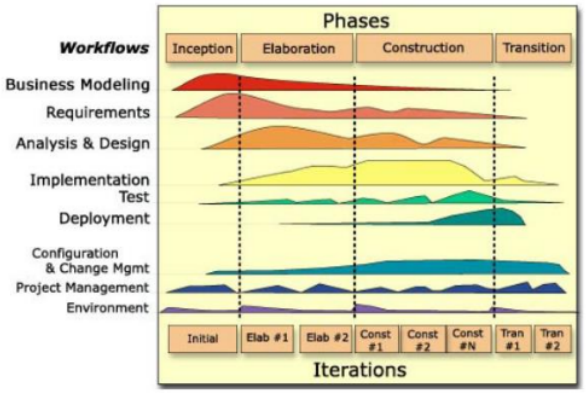
Rational Unified Process (RUP) merupakan suatu proses yang bersifat iteratif dalam pengembangan rekayasa perangkat lunak. Proses ini dilakukan melalui pendekatan secara disiplin dalam menetapkan suatu tugas dan tanggung

jawab dalam suatu organisasi pengembang perangkat lunak. Tujuan dari proses ini ialah untuk memastikan bahwasanya produksi dari perangkat luank yang telah dikembangakn memiliki kulaitas yang tinggi dan sesuai dengan harapan dari *end-user*. Karakteristik dari metode RUP ialah menggunakan menggunakan metodology OOP (*Object Oriented Programming*), menggunakan pendekatan bersifat iteratif untuk *lifecycle* dalam pengembangan perangkat lunak, serta model pengembangan menggunakan UML (*Unified Model Language*). RUP dalam satu siklus, RUP terdiri dari 4 fase diantaranya *Inception*, *Elaboration, Construction*, dan *Transition* (Pessoa & Anwar, 2014).

1. *Inception* merupakan fase awal pada proses RUP yang bertujuan untuk mendapatkan dan menganalisa kebutuhan dari semua pihak yang terkait. Pada fase ini dilakukan beberapa hal seperti membuat suatu *business case,* dan menetapkan ruang lingkup dari system yang akan dikembangkan.
2. *Elaboration* merupakan fase yang dikerjakan setelah melalui fase *inception*. Pada tahap ini dilakukan analisa manajemen resiko, menetapkan *base line* atau aturan kesepakatan, dan dilakukan analisa kebutuhan sistem.
3. *Construction* merupakan fase dimana proses pengembangan sistem yang telah dianalisa sebelumnya, di implemtasi di fase ini. Proses implementasi yang dilakukan melibatkan proses diantaranya analisa, desain, *prototype* dan *testing*.
4. *Transition* merupakan tahapan akhir dari fase yang ada di RUP. Fase ini berfokus pada peluncuran (*launching)* dari suatu produk, pembuatan

dokumentasi dari produk yang dikembangkan dan juga pemeliharaan (*maintenance*) dari perangkat lunak.

Rangkaian fase RUP dapat dilihat pada gambar II-3.



Gambar II-3. Fase *Rational Unified Process* (RUP)

2.3 Penelitian Lain yang Relevan

Pada penelitian ini terdapat uraian mengenai beberapa penelitian relevan yang bersumber dari prosiding dan jurnal ilmiah dari peneliti lain. Dalam rangka menambah referensi bagi peneliti dalam memperkuat landasan berpikir dalam menyelesaikan masalah pada topik yang diangkat.

Pada penelitian yang dilakukan oleh (Lucia & Londo, 2019), mempelajari beberapa algoritma klasifikasi yang digunakan untuk mengklasifikasi artikel berbahasa Indonesia. Ada banyak model *Machine Learning* yang dapat digunakan untuk melakukan tugas tersebut. Untuk itu, peneliti membandingkan penggunaan beberapa model *Machine Learning* yang cukup popular diantaranya *Multinomial Naïve Bayes, Decision Tree* dan *Support Vector Machine*. Hasil penelitian

menunujukkan SVM mendapatkan total akurasi terbaik sebesar 93%. Kemudian, diikuti MNB dengan akurasi berkisar 85-88%. *Decision Tree* memperoleh hasil terkecil karna hanya mendapatkan hasil akurasi sebesar 80-81%.

(Made et al., 2018) melakukan pengklasifikasian untuk mengidentifikasi teks tang mengandung konten *bullying*. SVM digunakan sebagai metode klasifikasi untuk mencari *hyperlane* pada kelas positif dan negative. Peneliti juga menggunakan metode seleksi fitur *Information Gain* untuk menyeleksi fitur dengan tingkat relevansi yang rendah. Hasil dari penelitian menggunakan metode SVM dengan nilai percobaan *iterMax* = 20, $\lambda = 0.5$, $\gamma = 0.001$, $\varepsilon = 0.000001$, dan $C = 1$. Untuk penggunaan *threshold* terbaik pada seleksi fitur *Information Gain* ialah 90%. *Threshold* ini mendapatkan hasil kinerja berupa 76.66% *accuracy,* 72,22% *precision*, *recall* 86.66% dan *f-measure* 78.78%.

(Luthfiana et al., 2020) dalam penelitiannya, melakukan klasifikasi untuk analisis sentiment berupa *user feedback* pada aplikasi. Penlitian ini membagi dataset menjadi 3 kelas yaitu positif, negatif dan netral. Berdasarkan hasil uji coba yang telah dilakukan dengan rasio pembagian data 80:20 dan penggunaan *threshold Chis-Square* sebesar 6,63 mendapatkan hasil kinerja terbaik berupa *accuracy* 77%, *precision* 50%, *recall* dan *F1-Score* 73%. Untuk pengklasifikasian dengan SVM dengan parameter terbaik yaitu C 100 dan gamma 0,001. Untuk proses klasifikasi tanpa menggunakan *Chi-Square* hanya mendapatkan hasil kinerja terbaik dengan parameter C 10 dan gamma 0,01 berupa *accuracy* sebesar 69%, *precision* 48%, *recall* 53% dan *F-1 Score* 50%.

(Hanafi et al., 2020) meneliti klasifikaisi multilabel pada dokumen hadis bukhari dalam terjemahan Bahasa Indonesia. Proses klasifikasi menggunakan model unigram/bigram, seleksi fitur menggunakan *Mutual Information* (MI) dan algoritma klasifikasi menggunakan *Support Vector Machine* (SVM). Pada penelitian ini metode seleksi fitur *Mutual Information* mampu meningkatkan *accuracy* dari 91,86% menjadi 93,4%.

Perbandingan metode seleksi fitur dan algoritma dalam rangka optimasi kinerja teks sebelumnya telah dilakukan oleh (Chandra, 2019) dengan membandingkan beberapa metode seleksi yang cukup terkenal diantaranya *Chi-Square, Term Frequency* dan *Mutual Information*. Selain itu peneliti juga melakukan perbandingan terhadap algortima klasifikasi yaitu SVM dan MNB. Dataset yang digunakan ialah dokumen tidak seimbang pada text Indonesia. Hasil pengujian mendapatkan skor akhir mulai dari 85 hingga 90 di tiap F1-Score. Metode ini diuji pada 2 data set yaitu pada *standard benchmarking dataset* dan dataset berupa teks berita Indonesia. Pada pengujian ini, metode seleksi fitur *Chi-Square* mendapatkan hasil yang paling konsisten dengan hasil rata-rata yaitu 89.76% saat dikombinasikan menggunakan algoritma SVM.

(Sahuri, 2018) melakukan studi perbandingan untuk memperoleh fitur terbaik dan metode seleksi yang cocok dengan metode klasifikasi tertentu. Metode seleksi fitur yang dibandingkan dalam penelitian ini ialah *Information Gain, Gini Index*, dan *Chi-Square*. Selanjutnya, untuk metode pengklasifikasian peneliti membandingkan metode klasifikasi *Neural Network*, *K-Nearest Neighbor*, *Naïve Bayes* dan *Support Vector Machine*. Hasil menunjukkan metode klasifikasi *K*-

Nearest Neighbor memperoleh hasil maksimal ketika dikombinasikan dengan metode seleksi fitur *Information Gain* dengan nilai k sebesar 4. *Neural Network* juga menghasilkan nilai optimal jika dikombinasikan dengan *Information Gain*. Sedangkan klasifikasi *Support Vector Machine* memperoleh hasil maksimal yakni sebesar 82% ketika diuji menggunakan seleksi fitur *Chi-Square* dan *Information Gain*.

2.4 Kesimpulan

Bab ini telah menguraikan landasan teori yang digunakan dan juga menguraikan hasil penelitian-penelitian sebelumnya yang berhubungan dengan konsep penelitian yang dilakukan yaitu klasifikasi pertanyaan berbahasa indonesia menggunakan seleksi fitur IG, *Chi-Square* dan MI. Kemudian, menggunakan SVM sebagai algortima klasifkasi. Landasan teori yang dicantumkan dalam bab ini akan menjadi referensi pada bab selanjutnya.

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 insepsi, fase elaborasi, fase konstruksi, dan fase transisi.

4.2 Fase Insepsi

Fase insepsi 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 *preprocessing*
2.	Perangkat lunak dapat melakukan proses klasifikasi menggunakan algortima SVM

3.	Perangkat lunak dapat melakukan proses seleksi fitur *Information Gain* dan proses klasifikasi menggunakan algortima SVM
4.	Perangkat lunak dapat melakukan proses seleksi fitur *Chi-Square* dan proses klasifikasi menggunakan algortima SVM
5.	Perangkat lunak dapat melakukan proses seleksi fitur *Mutual Information* dan proses klasifikasi menggunakan algortima 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 *user interface* yang mudah digunakan dan dipahami

4.2.3 Analisis dan Desain

Salah satu proses penting yang terjadi pada fase insepsi 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 dimililiki 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 algortima *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 masukkan 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 algortima *Support Vector Machine*.

4.2.3.2 Analisis Data

Penelitian ini menggunakan data berupa kumpulan teks pertanyaan dalam Bahasa Indonesia yang telah dilabeli. Data tersebut bersumber dari penelitian (Yusliani et al., 2021). Data yang digunakan berjumlah 1195 data pertanyaan yang terbagi menjadi 519 data pertanyaan dengan label *factoid*, 491 data pertanyaan dengan label *non-factoid* dan 185 data pertanyaan dengan label others. Kumpulan data tersebut disimpan dalam file dengan formal .xls.

4.2.3.3 Analisis *Text Preprocessing*

Text processing yang dilakukan pada penelitian ini bertujuan untuk membuat data menjadi terstruktur dan menghilngkan *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	*Factoid*
D2	ada dimana gelang pemberian dari nenekmu	*Factoid*
D3	katakan siapa yang menyuruhmu	*Factoid*

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

1. Noise Removal

Noise removal merupkan 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	*Factoid*
D2	ada dimana gelang pemberian dari nenekmu	*Factoid*
D3	katakan siapa yang menyuruhmu	*Factoid*
D4	kapan diperingatinya hari pahlawan	*Factoid*
D5	siapa penemu telepon	*Factoid*

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

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	***Tweet***	**Label**
D1	alat apa yang digunakan untuk mencampur adonan kue	*Factoid*
D2	ada dimana gelang pemberian dari nenekmu	*Factoid*
D3	katakan siapa yang menyuruhmu	*Factoid*
D4	kapan diperingatinya hari pahlawan	*Factoid*
D5	siapa penemu telepon	*Factoid*
D6	apa alasanmu melakukan perbuatan keji itu	*Non-factoid*
D7	apa kendala yang dihadapi dalam mengerjakan tugas akhir	*Non-factoid*

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

3. Tokenizing

Tokenizing merupakan sutau 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	***Tweet***	**Label**
D1	“alat”, “apa”, “yang”, “guna”, “untuk”, “campur”, “adon”, “kue”	*Factoid*
D2	“ada”, “mana” ,”gelang”, “beri”, “dari”, “nenek”	*Factoid*
D3	“kata”, “siapa” ,”yang”, “suruh”	*Factoid*
D4	“kapan”, “ingat”, “hari”, “pahlawan”	*Factoid*
D5	“siapa”, “temu”, “telepon”	*Factoid*
D6	“apa”, “alas”, “laku”, “buat”, “keji”, “itu”	*Non-factoid*
D7	“apa”, “kendala”, “yang”, “hadap”, “dalam”, “kerja”, “tugas”, “akhir”	*Non-factoid*
D8	“apa”, “guna”, “mixer”, “dalam”, “proses”, “buat”, “kue”	*Non-factoid*

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

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*.

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

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

Langkah 1 : Melakukan ekstrasi 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

	TF															DF	N	IDF
Term	**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

Tabel IV-8. Hasil Pembobotan Kata TF-IDF

TF-IDF (TF*IDF)															
Term	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15
alat	2,176	0	0	0	0	0	0	0	0,000	0	0	0	0	0	0
apa	1,574	0	0	0	0	1,574	1	1,574	0,000	0	0	0	0	0	0
yang	1,574	0	1,574	0	0	0	1	0	0	0	0	0	0	0	1,574
guna	1,699	1,699	0	0	0	0	0	1,699	0	0	0	0	0	0	0
untuk	2,176	0	0	0	0	0	0	0	0	0	0	0	0	0	0
campur	2,176	0	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
kue	2,176	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ada	0	2,176	0	0	0	0	0	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
gelang	0	2,176	0	0	0	0	0	0	0	0	0	0	0	0	0
beri	0	2,176	0	0	0	0	0	0	0	0	0	0	0	0	0
dari	0	1,875	0	0	0	0	0	0	1,875	0	0	0	0	0	0
nenek	0	2,176	0	0	0	0	0	0	0	0	0	0	0	0	0
kata	0	0	2,176	0	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
suruh	0	0	2,176	0	0	0	0	0	0	0	0	0	0	0	0
kapan	0	0	0	1,875	0	0	0	0	0	0	0	0	0	1,875	0
ingat	0	0	0	2,176	0	0	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
pahlawan	0	0	0	2,176	0	0	0	0	0	0	0	0	0	0	0
temu	0	0	0	0	2,176	0	0	0	0	0	0	0	0	0	0
telepon	0	0	0	0	1,875	0	1	0	0	0	0	0	0	0	0
alas	0	0	0	0	0	2,176	0	0	0	0	0	0	0	0	0
laku	0	0	0	0	0	2,176	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
keji	0	0	0	0	0	1,875	0	0	1,875	0	0	0	0	0	0
itu	0	0	0	0	0	1,875	0	0	0	0	0	0	0	0	0
kendala	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
hadap	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

dalam	0	0	0	0	0	0	1	1,699	0	0	0	0	0	0	1,699
kerja	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
tugas	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
akhir	0	0	0	0	0	0	1	0	0	0	0	0	0	1,875	0
mixer	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
jelas	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
mengapa	0	0	0	0	0	0	0	0	2,176	0	0	0	0	0	0
kau	0	0	0	0	0	0	0	0	1,875	0	0	0	0	1,875	0
rahasia	0	0	0	0	0	0	0	0	2,176	0	0	0	0	0	0
ini	0	0	0	0	0	0	0	0	2,176	0	0	0	0	0	0
bagaimana	0	0	0	0	0	0	0	0	0	1	1,699	1,699	0	0	0
dokter	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
bisa	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
papar	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
virus	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
covid-19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
sedia	0	0	0	0	0	0	0	0	0	0	2,176	0	0	0	0
kamu	0	0	0	0	0	0	0	0	0	0	2,176	0	0	0	0
ajar	0	0	0	0	0	0	0	0	0	0	1,875	1,875	0	0	0
cara	0	0	0	0	0	0	0	0	0	0	2,176	0	0	0	0
main	0	0	0	0	0	0	0	0	0	0	2,176	0	0	0	0
sepeda	0	0	0	0	0	0	0	0	0	0	2,176	0	0	0	0
bukankah	0	0	0	0	0	0	0	0	0	0	0	2,176	0	0	0
tau	0	0	0	0	0	0	0	0	0	0	0	2,176	0	0	0
harus	0	0	0	0	0	0	0	0	0	0	0	2,176	0	0	0
sekarang	0	0	0	0	0	0	0	0	0	0	0	2,176	0	0	0
turut	0	0	0	0	0	0	0	0	0	0	0	0	2,176	0	0
agen	0	0	0	0	0	0	0	0	0	0	0	0	2,176	0	0
sepakbola	0	0	0	0	0	0	0	0	0	0	0	0	2,176	0	0
baik	0	0	0	0	0	0	0	0	0	0	0	0	2,176	0	0
kita	0	0	0	0	0	0	0	0	0	0	0	0	0	2,176	0
kunjung	0	0	0	0	0	0	0	0	0	0	0	0	0	2,176	0

kesini	0	0	0	0	0	0	0	0	0	0	0	0	0	2,176	0
saja	0	0	0	0	0	0	0	0	0	0	0	0	1,875	0	1,875
ikut	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2,176
ke	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2,176
gunung	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2,176
simeulu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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 medapatkan 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.

$x_1{}^T x_1$ = (2.176x2.176) + (1.574x1.574) + (1.574x1.574) + (1.699x1.699) +
(2.176x2.176) + (2.176x2.176)+ (2.176x2.176) + (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) + (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)
= 31.519

$x_1{}^T x_2$ = (2.176x0) + (1.574x0) + (1.574x0) + (1.699x1.699) + (2.176x0) +
(2.176x0)+ (2.176x0) + (2.176x0) + (0x2.176) + (0x2.176) +
(0x2.176) + (0x2.176) + (0x1.875) + (0x2.176) + (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) + (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

$x_2{}^T x_1$ = (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) + (0x0) + (0x0) + (0x0)
+ (0x0) + (0x0) + (0x0)

= 2.886

$x_2{}^T x_2$ = (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) + (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

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

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

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

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

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

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

$$a_1 - a_2 = 0$$

$$a_1 = a_2$$

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

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

$$43.38 a_1 a_1 = 2a_1$$

$$a_1 = 0.036$$

Nilai $a_1 = a_2 = 0.0225$

c. Mencari nilai bias melalui persamaan II-16.

$$b = \frac{1}{NSV} \sum_{x_j \in SV} \left(\frac{1}{y_i} - \sum_{x_j \in SV} (\alpha_j y_j K(x_j, x_i))\right)$$

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

$$= \frac{1}{2} \left\{ \left(\frac{1}{1} - (0.036 \times 1 \times 31.519) + (0.036 \times (-1) \times 2.886)\right) + \left(\frac{1}{-1} - (0.036 \times 1 \times 2.886) + (0.036 \times (-1) \times 30.078)\right)\right\}$$

$$= -0.03$$

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	$f_{23(x)} = (w_{23})x + b_{23}$

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.

$x_1{}^T x_d =$ (2.176x2.176) + (1.574x1.574) + (1.574x1.574) + (1.699x1.699) + (2.176x2.176) + (2.176x2.176)+ (2.176x2.176) + (2.176x2.176) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) (0x0) + (0x0)+ (0x0) + (0x0)

= 31.519

$x_2{}^T x_d =$ (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) + (0x0) +
(0x0) + (0x0) + (0x0) + (0x0) + (0x0)
= 2.886

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

$$f(x) = sgn(\sum_{i=1}^{nsv} \alpha_i \, y_i K(x_i, x_d) + b)$$

$$= sgn\left(\left(a_1 y_1 K(x_1, x_d)\right) + \left(a_1 y_2 K(x_2, x_d)\right) + b\right)$$

$$= sgn\big((0.036 \times 1 \times 31.519) + (0.036 \times (-1) \times 2.886) + (-0{,}03)\big)$$

$$= 1$$

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 menunjukan 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)															
Term	Data	Factoid	Non-factoid	Other	Tidak Hadir Factoid	Hadir Factoid	Tidak Hadir Non-factoid	Hadir Non-Factoid	Tidak Hadir Other	Hadir Other	Entorpy 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 ektraksi fitur pada bobot kata yang memiliki nilai *threshold* (K) ≥ 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	K=15
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 ektraksi fitur untuk mengubah teks kedalam bentuk matriks berupa *vector* (nilai) dengan menggunakan metode TF-IDF. Hasil ekraksi fitur ditampilkan pada tabel IV-12 dan tabel IV-13.

Tabel IV-12. Hasil Ektrasksi Fitur TF-IDF

	TF															DF	IDF
Term	**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	0	1	2,176091259
kita	0	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	0	1	1	2,176091259
kendala	0	0	0	0	0	0	1	0	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	0	1	2,176091259
mengapa	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	2,176091259
pada	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	2,176091259
kue	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	2,176091259
proses	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	2,176091259
saja	0	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	0	1	2,176091259
tau	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	2,176091259

Tabel IV-16. Hasil Ektrasksi 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
agen	0	0	0	0	0	0	0	0	0	0	0	0	2,176	0	0
alat	2,176	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
bisa	0	0	0	0	0	0	0	0	0	2,176	0	0	0	0	0
bukankah	0	0	0	0	0	0	0	0	0	0	0	2,176	0	0	0
laku	0	0	0	0	0	2,176	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
temu	0	0	0	0	2,176	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
hari	0	0	0	2,176	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	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 medapatkan 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.

$x_1{}^T x_1$ = (0x0) + (0x0) + (2.176x2.176) + (0x0) + (0x0) + (0x0)+ (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0)

= 4.735

$x_1{}^T x_2$ = (0x2.176) + (0x0) + (2.176x0) + (0x0) + (0x0) + (0x0)+ (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x2.176) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) (0x0)

= 0

$x_2{}^T x_1$ = (2.176x0) + (0x0) + (0x2.176) + (0x0) + (0x0) + (0x0)+ (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0)

+ (0x0) + (2.176x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) (0x0)

= 0

$x_2{}^T x_2$ = (2.176x2.176) + (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

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

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

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

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

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

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

$$a_1 - a_2 = 0$$

$$a_1 = a_2$$

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

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

$$14.21 a_1 a_1 = 2a_1$$

$a_1 = 0.141$

Nilai $a_1 = a_2 = 0.141$

c. Mencari nilai bias melalui persamaan II-16.

$$b = \frac{1}{NSV} \sum_{x_j \in SV} \left(\frac{1}{y_i} - \sum_{x_j \in SV} (\alpha_j y_j K(x_j, x_i))\right)$$

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

$$= \frac{1}{2} \left\{ \left(\frac{1}{1} - (0.141 \times 1 \times 4.735) + (0.141 \times (-1) \times 0) \right) + \left(\frac{1}{-1} - (0.141 \times 1 \times 0) + (0.141 \times (-1) \times 9.47) \right) \right\}$$

$$= 0.334$$

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.

$x_1{}^T x_d$ = (0x0) + (0x0) + (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)

= 4.735

$x_2{}^T x_d$ = (2.176x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0)+ (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (2.176x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0)

= 0

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

$$f(x) = sgn(\sum_{i=1}^{nsv} \alpha_i\, y_i K(x_i, x_d) + b)$$

$$= sgn\Big(\big(a_1 y_1 K(x_1, x_d)\big) + \big(a_1 y_2 K(x_2, x_d)\big) + b\Big)$$

$$= sgn\big((0.141 \times 1 \times 0) + (0.141 \times (-1) \times 0) + 0.334\big)$$

$= 1$

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.

1. 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	Non-factoid	Other	Max Term
	15	A	B	C	D	A	B	C	D	A	B	C	D	$X^2(t,R)$	$X^2(t,R)$	$X^2(t,R)$	
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 ektraksi fitur pada bobot kata yang memiliki nilai *threshold* (K) ≥ 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 ektraksi fitur untuk mengubah teks kedalam bentuk matriks berupa vector (nilai) dengan menggunakan metode TF-IDF. Hasil ekraksi fitur ditampilkan pada tabel IV-17 dan tabel IV-18.

Tabel IV-17. Hasil Ektrasksi Fitur TF-IDF

TF																DF	IDF
Term	**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	1	1	0	0	0	0	0	0	0	2	1,875061263
dari	0	1	0	0	0	0	0	0	1	0	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	1	0	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	1	0	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	1	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
laku	0	0	0	0	0	1	0	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	1	0	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	1	0	1	0	0	0	0	0	0	0	2	1,875061263
temu	0	0	0	0	1	0	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

untuk	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2,176091259

Tabel IV-21. Hasil Ektraksi 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
alat	2,176	0	0	0	0	0	0	0	0	0	0	0	0	0	0
bagaimana	0	0	0	0	0	0	0	0	0	1,875	0	1,875	0	0	0
baik	0	0	0	0	0	0	0	0	0	0	0	0	2,176	0	0
bisa	0	0	0	0	0	0	0	0	0	2,176	0	0	0	0	0
dalam	0	0	0	0	0	0	1,875	1,875	0	0	0	0	0	0	0
dari	0	1,875	0	0	0	0	0	0	1,875	0	0	0	0	0	0
hari	0	0	0	2,176	0	0	0	0	0	0	0	0	0	0	0
ikut	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2,176
itu	0	0	0	0	0	0	0	0	0	1,875	1,875	0	0	0	0
jelas	0	0	0	0	0	0	0	0	2,176	0	0	0	0	0	0
kata	0	0	2,176	0	0	0	0	0	0	0	0	0	0	0	0
kau	0	0	0	0	0	0	0	0	1,875	0	0	0	0	1,875	0
kesini	0	0	0	0	0	0	0	0	0	0	0	0	0	2,176	0
kendala	0	0	0	0	0	0	2,176	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
laku	0	0	0	0	0	2,176	0	0	0	0	0	0	0	0	0
main	0	1,875	0	0	0	0	0	0	0	0	1,875	0	0	0	0
rahasia	0	0	0	0	0	0	0	0	2,176	0	0	0	0	0	0
sedia	0	0	0	0	0	0	0	0	0	0	2,176	0	0	0	0
siapa	0	0	1,574	0	1,574	0	0	0	0	0	0	0	1,574	0	1,000
buat	0	0	0	0	0	1,875	0	1,875	0	0	0	0	0	0	0
temu	0	0	0	0	2,176	0	0	0	0	0	0	0	0	0	0
turut	0	0	0	0	0	0	0	0	0	0	0	0	2,176	0	0
untu	2,176	0	0	0	0	0	0	0	0	0	0	0	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 medapatkan 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.

$x_1{}^T x_1$ = (0x0) + (2.176x2.176) + (0x0) + (0x0) + (0x0) + (0x0)+ (0x0) + (2.176x2.176)

= 9.47

$x_1{}^T x_2$ = (0x2.176) + (2.176x0) + (0x0) + (0x0) + (0x0) + (0x0)+ (0x1.875) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x1.875) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (2.176x0)

= 0

$x_2{}^T x_1$ = (2.176x0) + (0x2.176) + (0x0) + (0x0) + (0x0) + (0x0)+ (1.875x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (1.875x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x2.176)

$= 0$

$x_2{}^T x_2 = (2.176\text{x}2.176) + (0\text{x}0) + (0\text{x}0) + (0\text{x}0) + (0\text{x}0) + (0\text{x}0) + (1.875\text{x}1.875)$

$+ (0\text{x}0) + (0\text{x}0) + (0\text{x}0) + (0\text{x}0) + (0\text{x}0) + (0\text{x}0) + (0\text{x}0) + (0\text{x}0)$

$+ (0\text{x}0) + (0\text{x}0) + (1.875\text{x}1.875) + (0\text{x}0) + (0\text{x}0) + (0\text{x}0) + (0\text{x}0)$

$+ (0\text{x}0) + (0\text{x}0) + (0\text{x}0)$

$= 11.76$

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

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

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

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

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

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

$$a_1 - a_2 = 0$$

$$a_1 = a_2$$

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

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

$$21.24 a_1 a_1 = 2a_1$$

$$a_1 = 0.094$$

Nilai $a_1 = a_2 = 0.094$

c. Mencari nilai bias melalui persamaan II-16.

$$b = \frac{1}{NSV} \sum_{x_j \in SV} \left(\frac{1}{y_i} - \sum_{x_j \in SV} (\alpha_j y_j K(x_j, x_i))\right)$$

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

$$= \frac{1}{2} \left\{ \left(\frac{1}{1} - (0.094 \times 1 \times 9.47) + (0.094 \times (-1) \times 0)\right) + \left(\frac{1}{-1} - (0.094 \times 1 \times 0) + (0.094 \times (-1) \times 11.77)\right)\right\}$$

$$= 0.107$$

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.

$x_1{}^T x_d$ = (0x0) + (2.176x2.176) + (0x0) + (0x0) + (0x0) + (0x0)+ (0x0) + (2.176x2.176)

= 4.735

$x_2{}^T x_d$ = (2.176x0) + (0x2.176) + (0x0) + (0x0) + (0x0) + (0x0)+ (1.875x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (1.875x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x2.176)

= 0

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

$$f(x) = sgn(\sum_{i=1}^{nsv} \alpha_i\, y_i K(x_i, x_d) + b)$$

$$= sgn\left(\left(a_1 y_1 K(x_1, x_d)\right) + \left(a_1 y_2 K(x_2, x_d)\right) + b\right)$$

$$= sgn\big((0.107 \times 1 \times 4.735) + (0.107 \times (-1) \times 0) + 0.017\big)$$

$$= 1$$

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 menunjukan kelas factoid maka *vote* untuk kelas factoid akan ditambah satu.

2. 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)																	
		Menentukan Panjang Kelas												Perhitungan Manual MI			Nilai MI
Term	Data	Factoid				Non-factoid				Other				Factoid	Non-factoid	Other	Max Term
	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 ektraksi fitur pada bobot kata yang memiliki nilai *threshold* (K) ≥ 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 ektraksi fitur untuk mengubah teks kedalam bentuk matriks berupa vector (nilai) dengan menggunakan metode TF-IDF. Hasil ekraksi fitur ditampilkan pada tabel IV-22 dan tabel IV-23.

Tabel IV-22. Hasil Ektrasksi Fitur TF-IDF

	TF															DF	IDF
Term	**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	1	0	1	4	1,574031268
Suruh	0	0	1	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	1	2,176091259
Temu	0	0	0	0	1	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	1	2,176091259

Tabel IV-23. Hasil Ektraksi 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

Suruh	0	0	2,176	0	0	0	0	0	0	0	0	0	0	0	0
Telepon	0	0	0	0	2,176	0	0	0	0	0	0	0	0	0	0
Temu	0	0	0	0	2,176	0	0	0	0	0	0	0	0	0	0
Untuk	2,176	0	0	0	0	0	0	0	0	0	0	0	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 medapatkan 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.

$x_1{}^T x_1$ = (0x0) + (2.176x2.176) + (0x0) + (0x0) + (2.176x2.176) + (0x0)+ (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (2.176x2.176)

= 14.205

$x_1{}^T x_2$ = (0x2.176) + (2.176x0) + (0x0) + (0x0) + (2.176x0) + (0x0)+ (0x1.875) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x1.875) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (2.176x0)

= 0

$x_2{}^T x_1$ = (2.176x0) + (0x2.176) + (0x0) + (0x0) + (0x2.176) + (0x0)+ (1.875x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (1.875x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x2.176)

= 0

$x_2{}^T x_2$ = (2.176x2.176) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0)+ (1.875x1.875) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (1.875x1.875) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0)

= 8.251

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

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

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

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

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

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

$$a_1 - a_2 = 0$$

$$a_1 = a_2$$

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

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

$$27.1 a_1 a_1 = 2a_1$$

$$a_1 = 0.074$$

Nilai $a_1 = a_2 = 0.074$

c. Mencari nilai bias melalui persamaan II-16.

$$b = \frac{1}{NSV} \sum_{x_j \in SV} \left(\frac{1}{y_i} - \sum_{x_j \in SV} (\alpha_j y_j K(x_j, x_i))\right)$$

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

$$= \frac{1}{2} \left\{ \left(\frac{1}{1} - (0.074 \times 1 \times 14.052) + (0.074 \times (-1) \times 0)\right) + \left(\frac{1}{-1} - (0.074 \times 1 \times 0) + (0.074 \times (-1) \times 8.251)\right)\right\}$$

$$= -0{,}22$$

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.

$x_1{}^T x_d$ = (0x0) + (2.176x2.176) + (0x0) + (0x0) + (2.176x2.176) + (0x0)+ (0x0) + (2.176x2.176)

= 14,205

$x_2{}^T x_d$ = (2.176x0) + (0x2.176) + (0x0) + (0x0) + (0x2.176) + (0x0)+ (1.875x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (1.875x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x0) + (0x2.176)

= 0

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

$$f(x) = sgn(\sum_{i=1}^{nsv} \alpha_i\, y_i K(x_i, x_d) + b)$$

$$= sgn\left(\left(a_1 y_1 K(x_1, x_d)\right) + \left(a_1 y_2 K(x_2, x_d)\right) + b\right)$$

$$= sgn\big((0.074 \times 1 \times 14{,}205) + (0.074 \times (-1) \times 0) + (-0{,}22)\big)$$

$$= 1$$

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 menunjukan kelas factoid maka *vote* untuk kelas factoid akan ditambah satu.

4.2.3.5 Analisis Hasil Klasifikasi

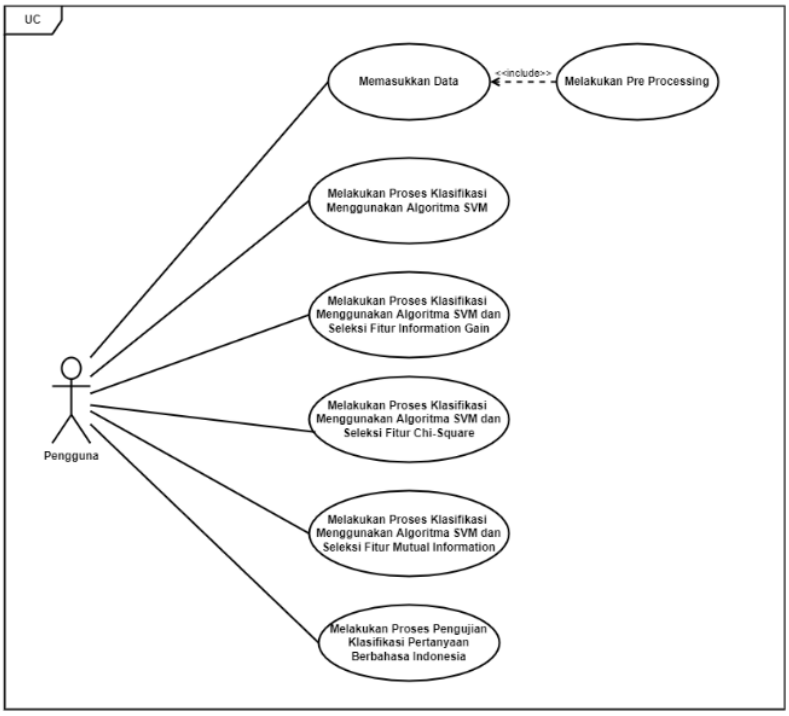
Untuk mengetahui kualitas dari hasil klasifikasi yang didapatkan, maka diperlukan suatu metode evaluasi. *Confusion Matrix* merupakan suatu metode yang digunakan untuk mengevaluasi hasil setiap proses klasifikasi dari perangkat lunak yang akan dikembangkan. Perhitungan waktu komputasi dan jumlah fitur 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 parangkat 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 ditujukkan untuk melakukan proses memasukkan *dataset* yang akan digunakan pada proses klasifikasi.
2.	Melakukan preprocessing	Proses ini ditujukkan untuk melakukan proses pengolahan data pada *dataset* masukkan yang terdiri dari *noise removal* dan *case folding*.
3.	Melakukan proses klasifikasi menggunakan algoritma SVM	Proses ini ditujukkan untuk melakukan proses pengklasifikasian pada data masukkan menggunakan algoritma SVM.
4.	Melakukan proses klasifikasi menggunakan algoritma SVM dan seleksi fitur *information gain*	Proses ini ditujukkan untuk melakukan proses pengklasifikasian pada data masukkan menggunakan algoritma SVM dan seleksi fitur *information gain*.
5.	Melakukan proses klasifikasi menggunakan algoritma SVM dan seleksi fitur *chi-square*	Proses ini ditujukkan untuk melakukan proses pengklasifikasian pada data masukkan menggunakan algoritma SVM dan seleksi fitur *chi-square*.
6.	Melakukan proses klasifikasi menggunakan algoritma SVM dan seleksi fitur *mutual information*	Proses ini ditujukkan untuk melakukan proses pengklasifikasian pada data masukkan menggunakan algoritma SVM dan seleksi fitur *mutual information*.
7.	Melakukan proses pengujian klasifikasi	Proses ini ditujukkan untuk melakukan proses pengujian pada data uji berupa pertanyaan

	pertanyaan berbahasa Indonesia	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 Praproses 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 masukkan.

Tabel IV-28. Skenario Melakukan Praproses Data

No. *Use Case*	002
Nama *Use Case*	Melakukan *Pre Processing*
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 *noise removal* dan *case folding*.
Kondisi Awal	Belum terdapat data masukkan
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 (*case folding*, *noise removal* dan *tokenization*) 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 belum memilih kernel dan/atau nilai C	
1. Pengguna tidak memilih metode kernel dan/atau nilai C	
	2. Menampilkan *pop-up box warning"*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 *Information Gain*.
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 *Information Gain*.
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 *Information Gain* pada tiap *term*
	7. Melakukan seleksi fitur berdasarkan nilai *Information Gain*
	8. Melakukan proses pembobotan kata menggunakan TF-IDF
	9. Membagi data latih dan data uji menggunakan metode *k-fold cross validation*
	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. Pengguna tidak memilih metode kernel dan/atau nilai C	
	2. Menampilkan *pop-up box warning"*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 *Chi-Square*
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 *Chi-Square*
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 *Chi-Square* pada tiap *term*
	7. Melakukan seleksi fitur berdasarkan nilai *Chi-Square*
	8. Melakukan proses pembobotan kata menggunakan TF-IDF
	9. Membagi data latih dan data uji menggunakan metode *k-fold cross validation*
	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. Pengguna tidak memilih metode kernel dan/atau nilai C	
	2. Menampilkan *pop-up box warning"*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 *Mutual Information*
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 *Mutual Information*
Kondisi Awal	Data masukan hasil pra-pengolahan
Kondisi Akhir	Sistem menampilkan hasil pengujian
Skenario Normal	
Actor	**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 *Mutual Information* pada tiap *term*
	7. Melakukan seleksi fitur berdasarkan nilai *Mutual Information*
	8. Melakukan proses pembobotan kata menggunakan TF-IDF
	9. Membagi data latih dan data uji menggunakan metode *k-fold cross validation*
	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. Pengguna tidak memilih metode kernel dan/atau nilai C	
	2. Menampilkan *pop-up box warning"*Masukkan parameter ulang"

Tabel IV-33. Skenario Melakukan Proses Pengujian Klasifikasi PertanyaanBerbahasa 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 mengenail pemodelan bisnis, perancangan data, perancangan tampilan antarmuka, diagram *activity*, diagram *sequence* dan dokumentasi.

4.3.1 Pemodelan Bisnis

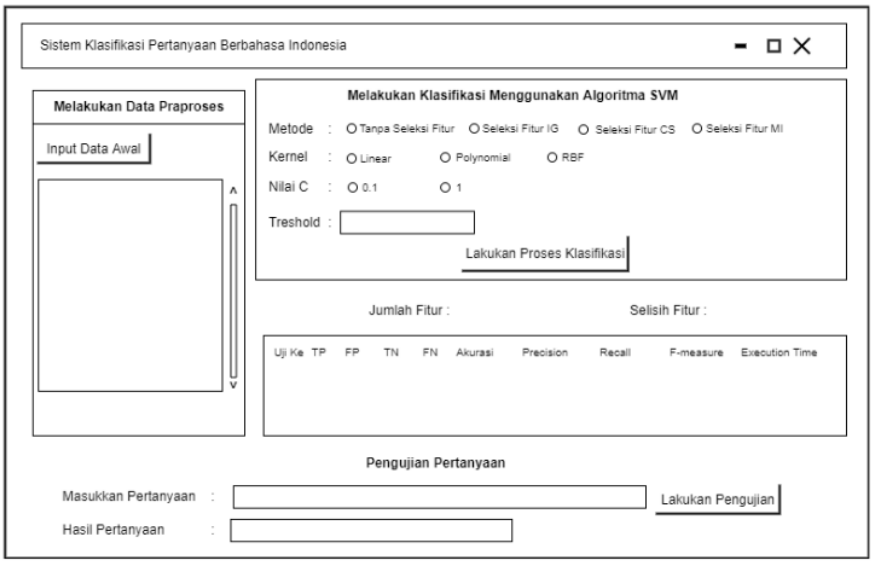
Subbab ini akan menguraikan mengenai penjelasan tentang perancangan perangkat lunak yang nantinya akan dikembangkan. Perancangan tersebut akan dibuat berdasarkan pemodelan bisnis yang telah diuraikan pada fase insepsi. Pada tahap ini akan menguraikan mengenai perancangan data yang akan digunakan dan 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.xlxs.

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:

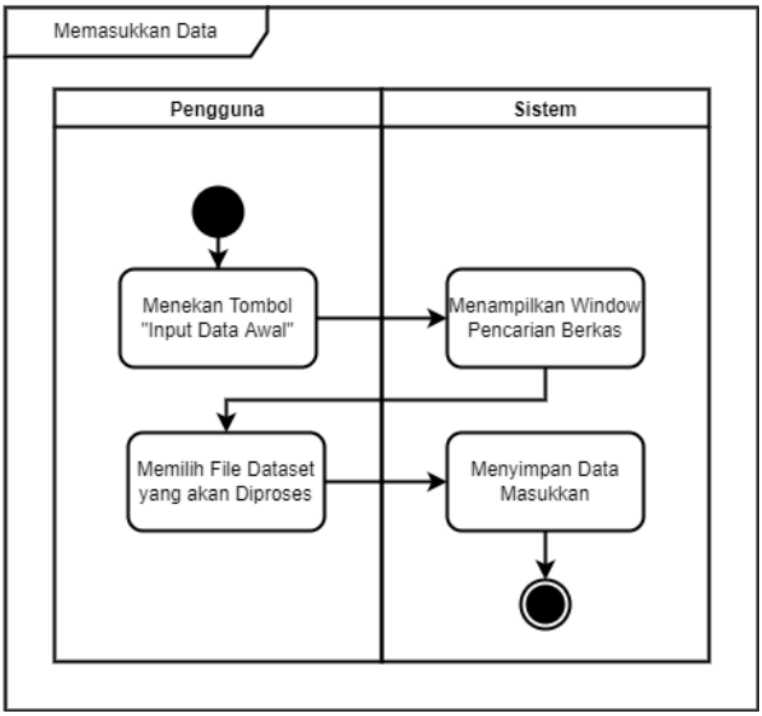
1. *Processor* Intel Core i5-5005U
2. RAM 8 GB
3. *Harddisk* 500 GB

Sedangkan perangkat lunak yang digunakan sebagai berikut:

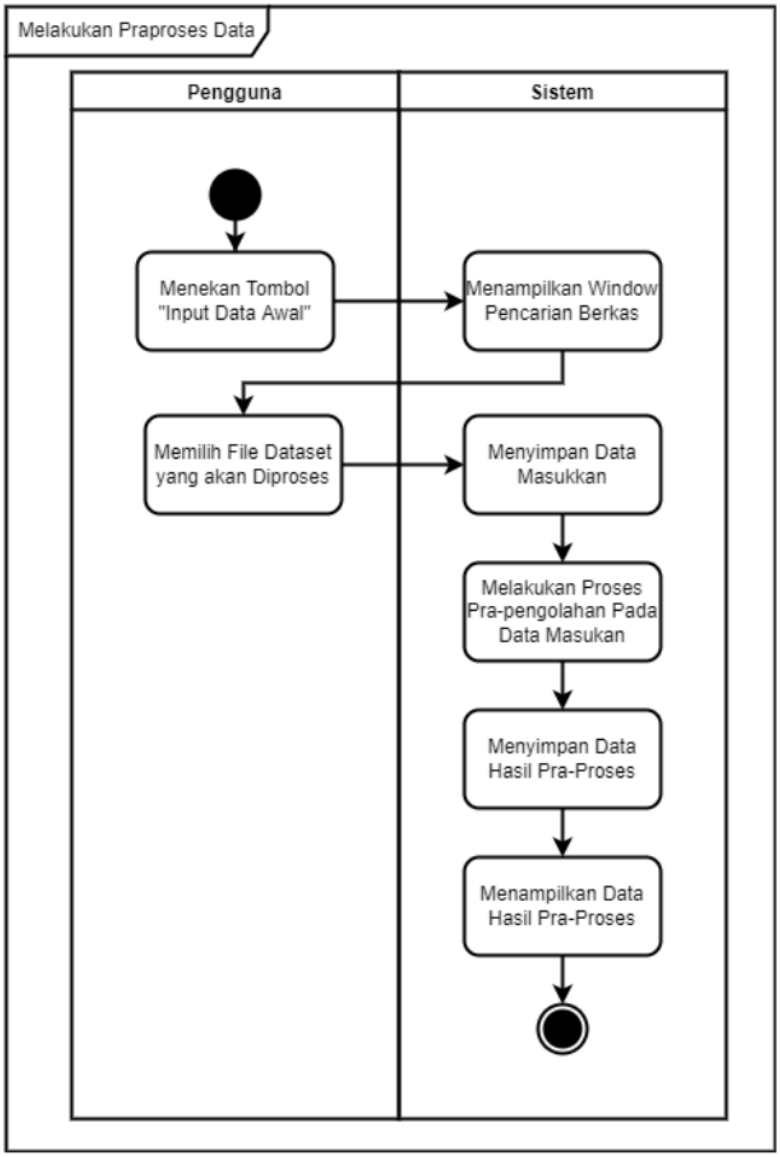
1. Sistem Operasi Windows 10 64-bit
2. Text Editor Spyder

4.3.5 Diagram Aktivitas

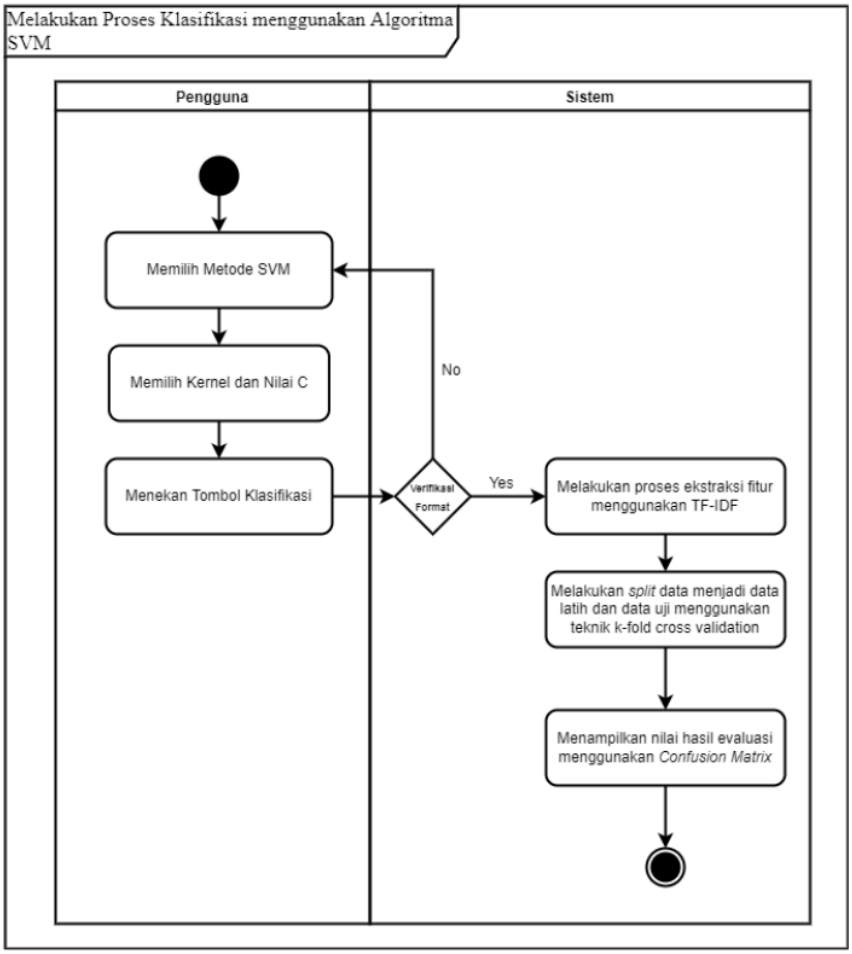
Diagram aktivitas merupakan suatu diagram yang menggambarkan aktivitas yang dilakukan oleh *actor*. Pada penelitian ini terdiri dari tujuh rancangan diagram aktivitas yang mengacu pada diagram *use case* untuk pengembang perangkat lunak yang dilakukan.



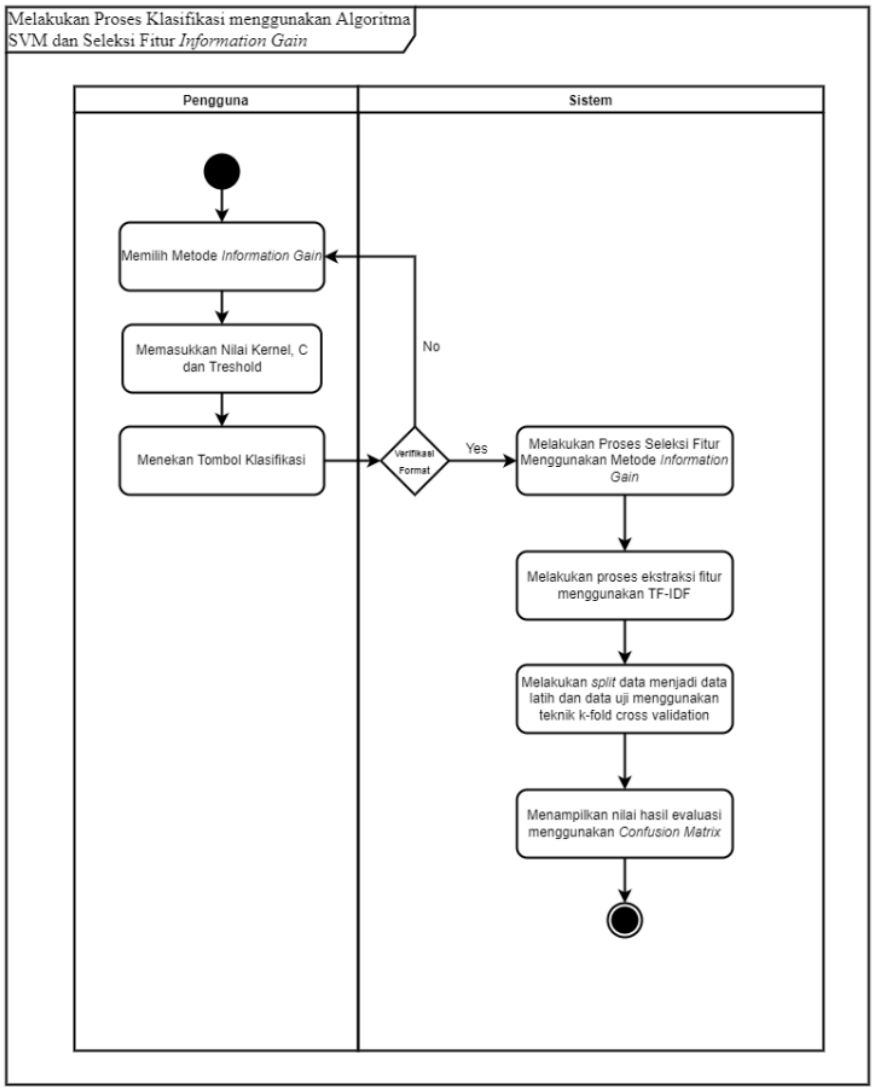
Gambar IV-3. Diagram Aktivitas Memasukkan Data



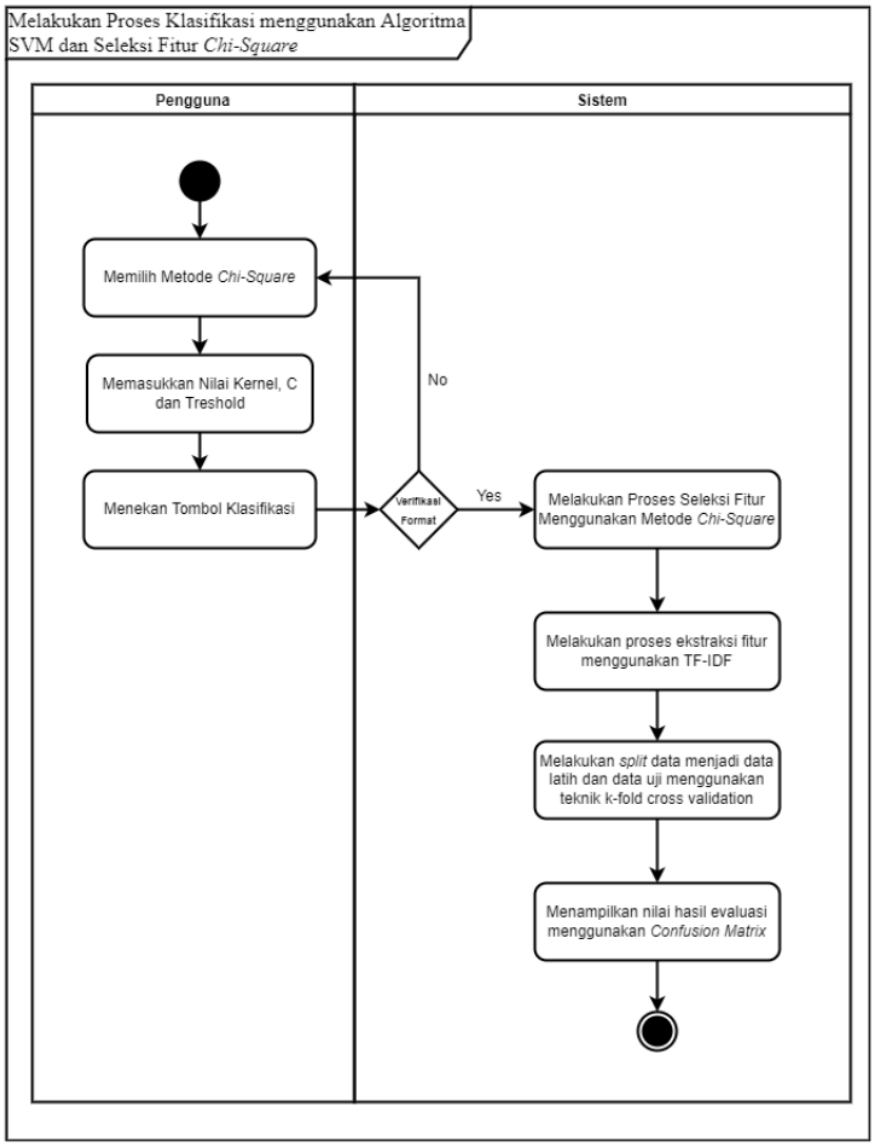
Gambar IV-4. Diagram Aktivitas Melakukan *Pre Processing*



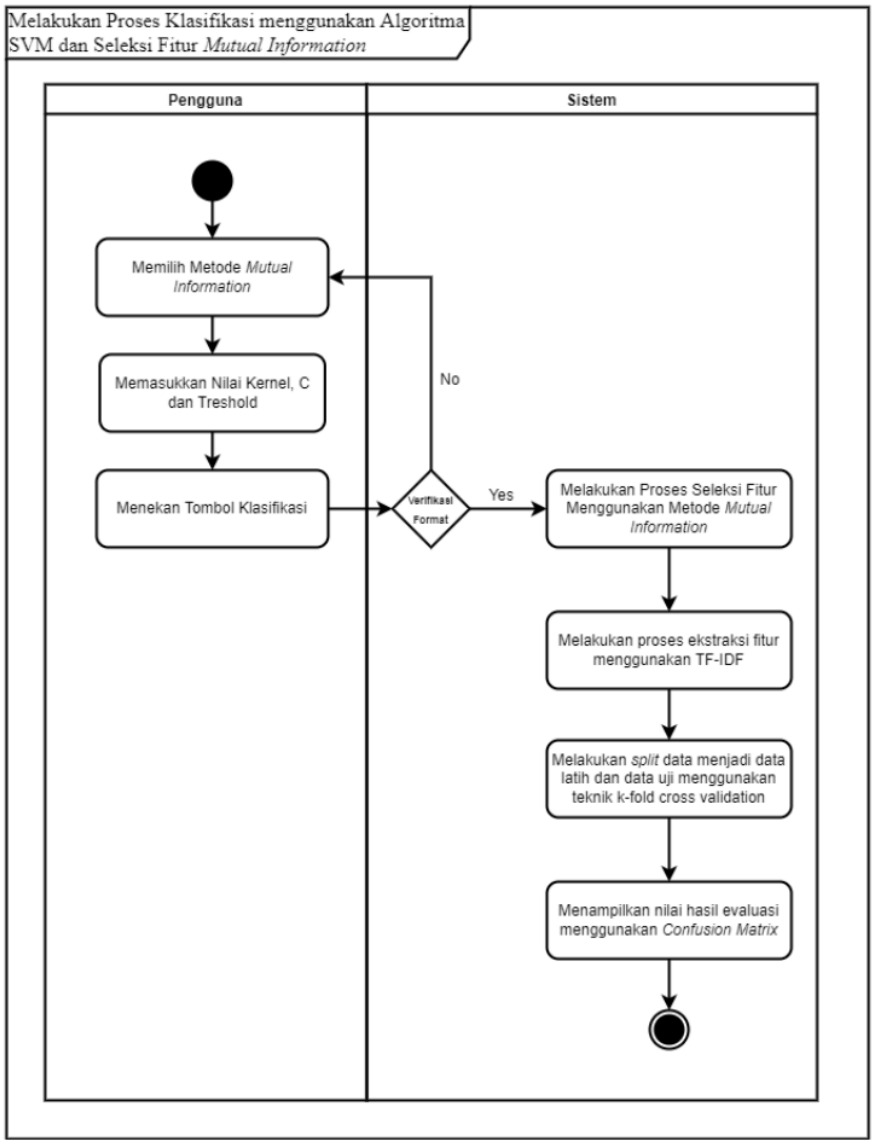
Gambar IV-5. Diagram Aktivitas Melakukan Proses Klasifikasi menggunakan Algoritma SVM



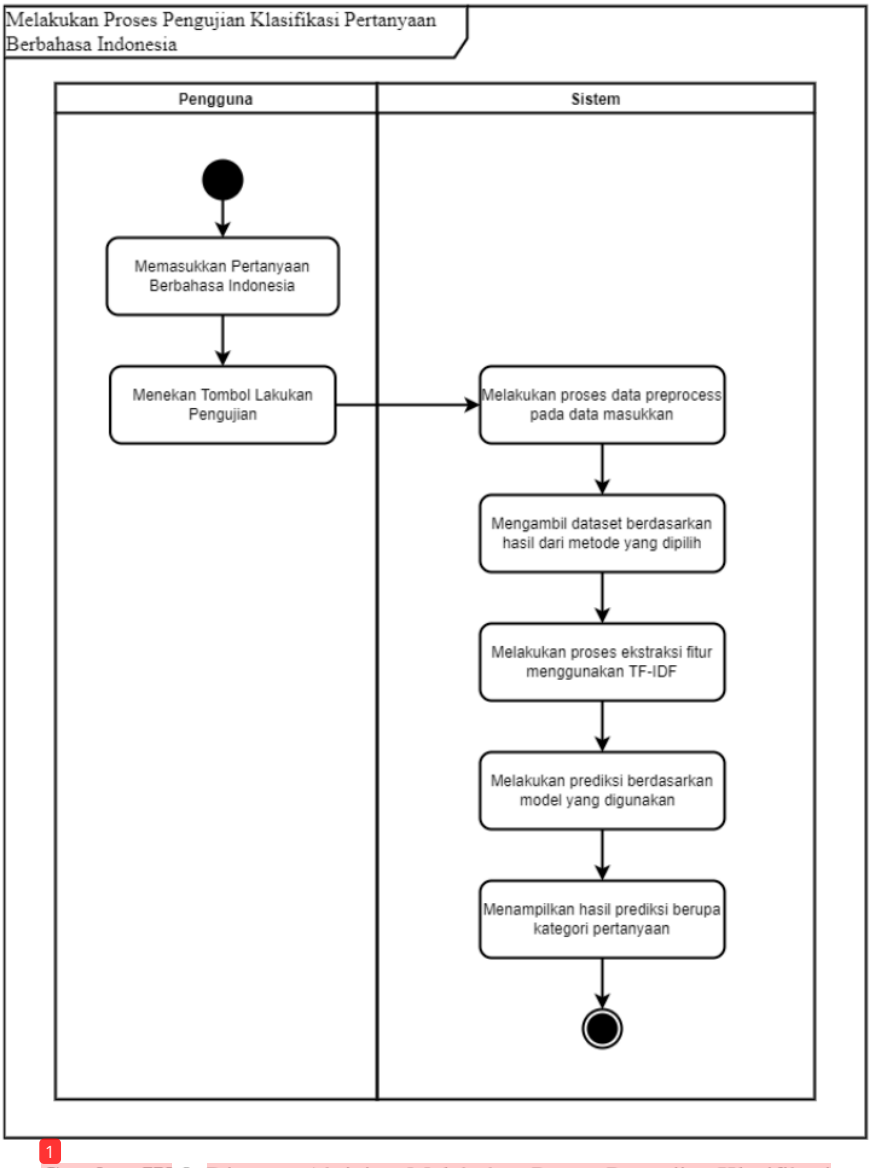
Gambar IV-6. Diagram Aktivitas Melakukan Proses Klasifikasi menggunakan Algoritma SVM dan Seleksi Fitur *Information Gain*



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



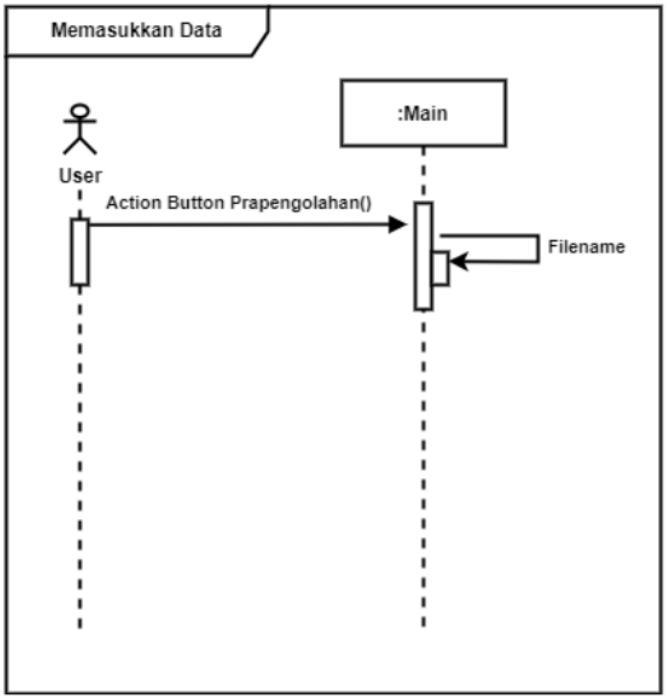
Gambar IV-8. Diagram Aktivitas Melakukan Proses Klasifikasi menggunakan Algoritma SVM dan Seleksi Fitur *Mutual Information*



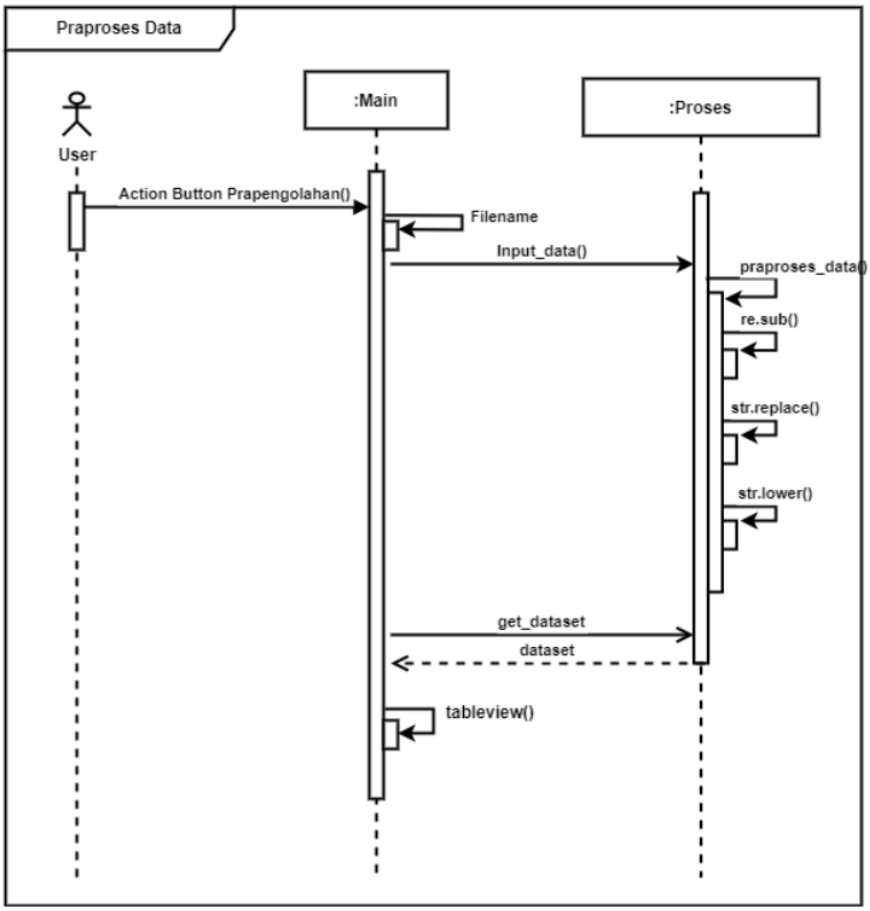
Gambar IV-9. Diagram Aktivitas Melakukan Proses Pengujian Klasifikasi Pertanyaan Berbahasa Indonesia

4.3.6 Diagram *Sequence*

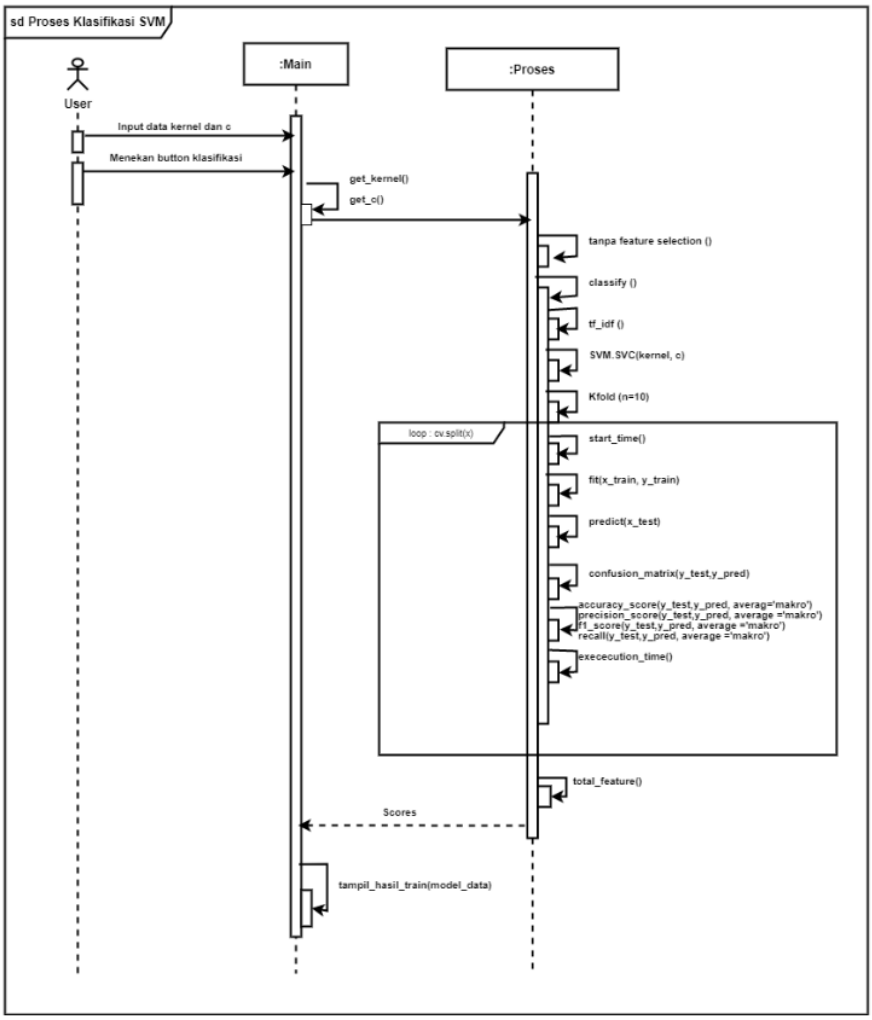
Diagram *sequence* atau yang biasa disebut diagram alur merupakan suatu diagram yang menggambarkan interaksi antar objek secara berurut yang ada di dalam dan disekitar sistem. Berdasarkan *use case* yang telah dirancang, terdapat tujuh diagram *sequence* yang digunakan pada penelitian ini. Diagram *sequence*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



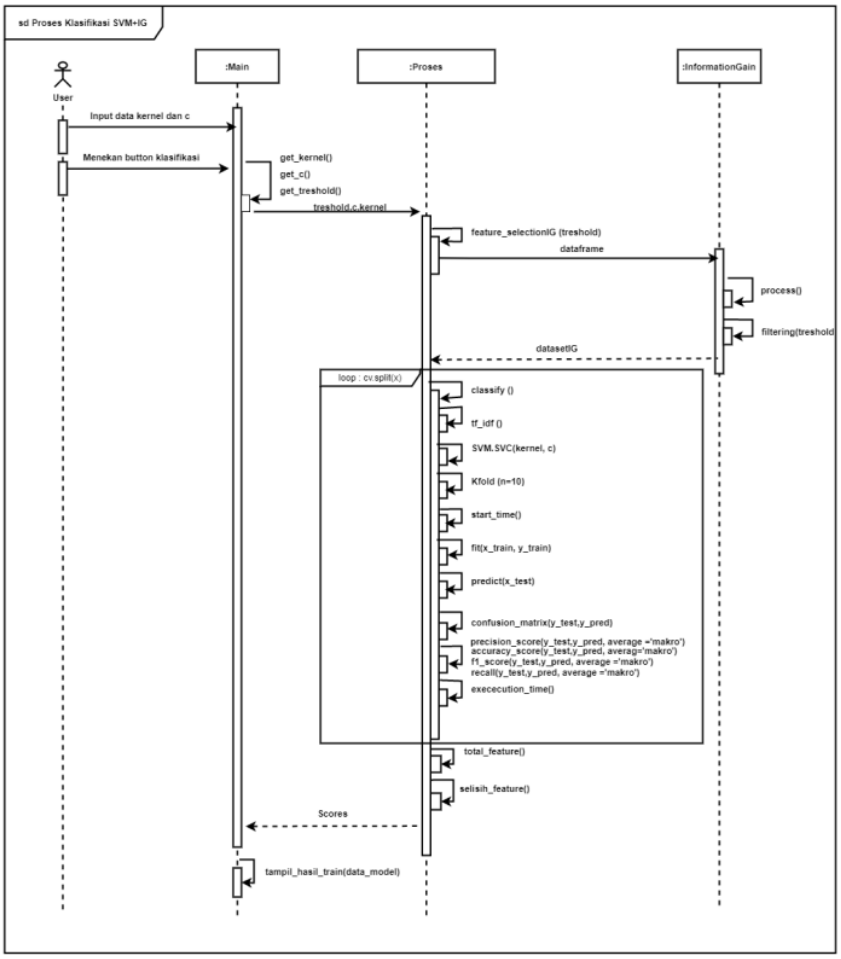
Gambar IV-10. Diagram *Sequence* Memasukkan Data



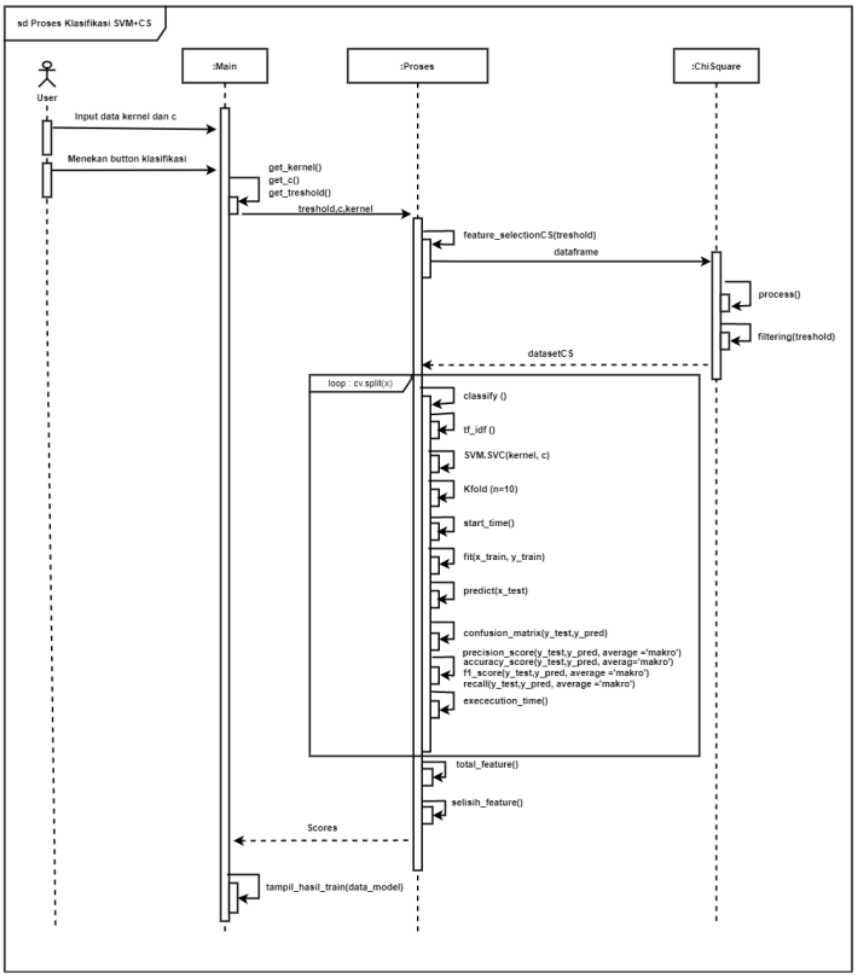
Gambar IV-11. Diagram *Sequence* Memilih *Dataset*



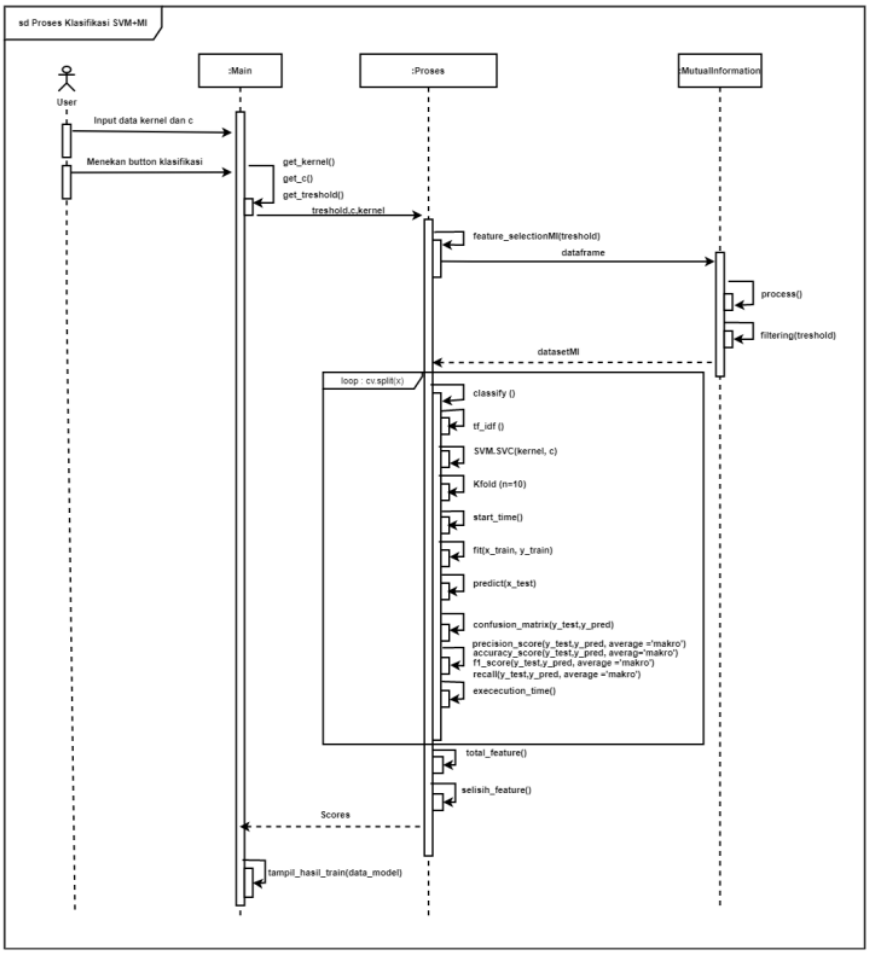
Gambar IV-12. Diagram *Sequence* Melakukan Proses Klasifikasi Menggunakan SVM



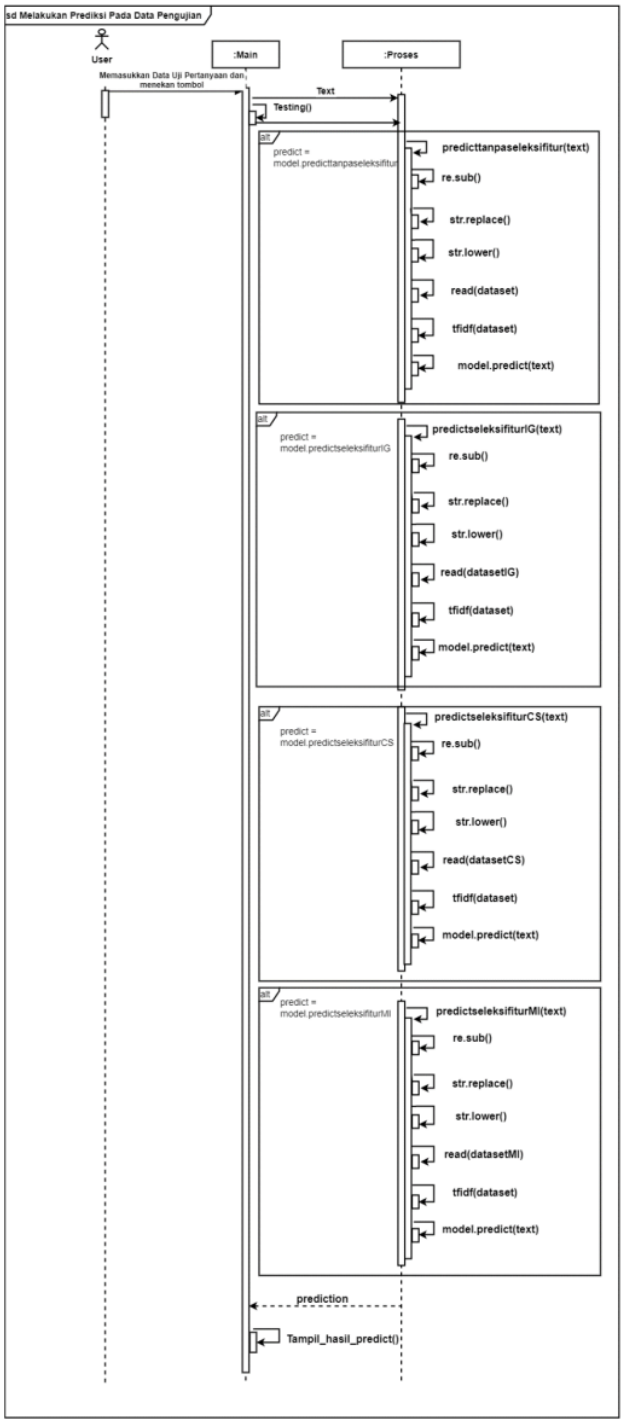
Gambar IV-13. Diagram *Sequence* Melakukan Proses Klasifikasi Menggunakan SVM dan Seleksi Fitur *Information Gain*



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*


sd Melakukan Prediksi Pada Data Pengujian
User
:Main
:Proses
Memasukkan Data Uji Pertanyaan dan menekan tombol
Text
Testing()
alt
predict = model.predicttanpaseleksifitur
predicttanpaseleksifitur(text)
re.sub()
str.replace()
str.lower()
read(dataset)
tfidf(dataset)
model.predict(text)
alt
predict = model.predictseleksifiturIG
predictseleksifiturIG(text)
re.sub()
str.replace()
str.lower()
read(datasetIG)
tfidf(dataset)
model.predict(text)
alt
predict = model.predictseleksifiturCS
predictseleksifiturCS(text)
re.sub()
str.replace()
str.lower()
read(datasetCS)
tfidf(dataset)
model.predict(text)
alt
predict = model.predictseleksifiturMI
predictseleksifiturMI(text)
re.sub()
str.replace()
str.lower()
read(datasetMI)
tfidf(dataset)
model.predict(text)
prediction
Tampil_hasil_predict()

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

4.4 Fase Konstruksi

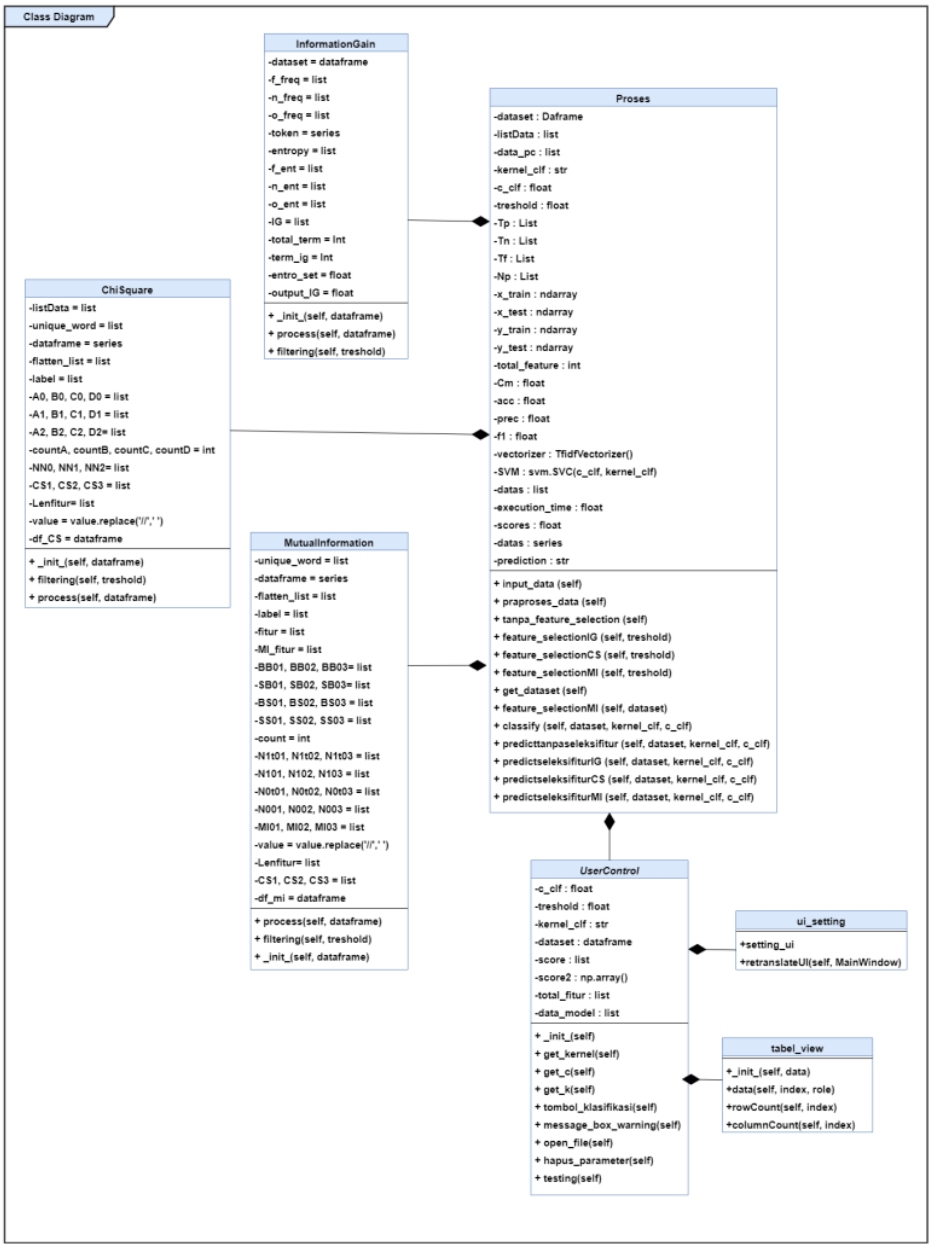
Fase konstruksi merupakan fase yang membahas perihal bagian inti dan fitur lain pada perangkat lunak yang akan dikembangkan. Pada tahapan ini, akan dilakukan beberapa rangkaian proses iterasi berupa proses analisis, desain, implementasi dan pengujian yang mana nantinya akan menghasilkan suatu perangkat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

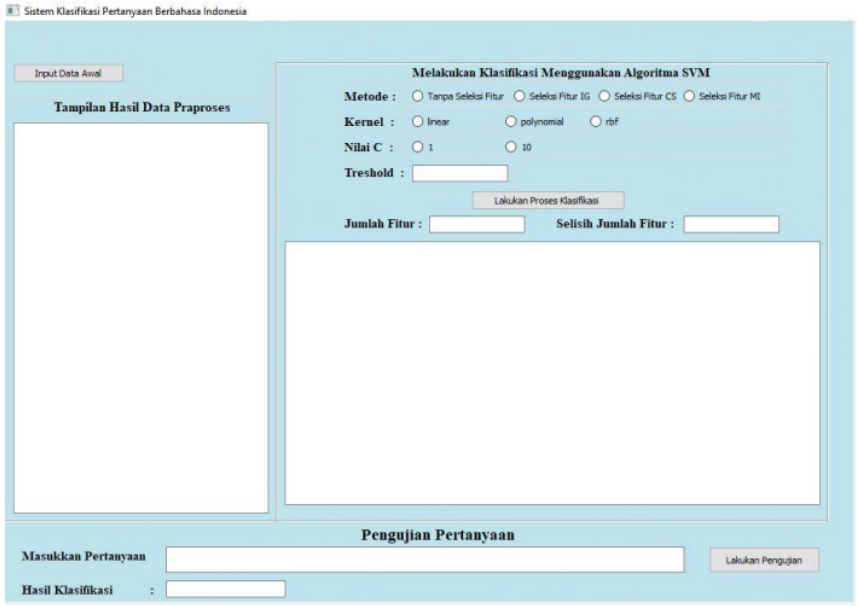
Tabel IV-34. Implementasi Kelas

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

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 *Information Gain* untuk proses penyeleksian fitur.
6.	ChiSquare	ChiSquare.py	Kelas ChiSquare adalah kelas yang berisikan rangkaian perhitungan *Chi Square* untuk proses penyeleksian fitur.
7.	MutualInformation	MutualInformation.py	Kelas Mutual Information adalah kelas yang berisikan rangkaian perhitungan *Mutual Information* 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.



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 parangkat 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 *dataset* yang telah dipilih	*Black box*	Pengujian Unit

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

NO.	ID	Pengujian	Jenis Pengujian	Tingkat Pengujian
1.	UC-2-1	Melakukan proses klasifikasi data dengan memilih metode tanpa seleksi fitur dan memasukkan parameter yang akan digunakan.	*Black box*	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
1.	UC-3-1	Melakukan proses klasifikasi data dengan memilih metode seleksi fitur *Information Gain* dan memasukkan parameter yang akan digunakan.	*Black box*	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
1.	UC-4-1	Melakukan proses klasifikasi data dengan memilih metode seleksi fitur *ChiSquare* dan memasukkan parameter yang akan digunakan.	*Black box*	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
1.	UC-5-1	Melakukan proses klasifikasi data dengan memilih metode seleksi fitur *Mutual Information*	*Black box*	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
1.	UC-6-1	Melakukan proses pengujian dengan memasukkan data uji berupa kalimat tanya berbahasa Indonesia yang natinya akan dilakukan proses klasifikasi.	*Black box*	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 Praprses Data

ID	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Hasil yang Didapat	Kesimpulan
UC-1-1	Melakukan data *preprocessing*	Menekan tombol "input data awal"	dataset	Hasil praproses data pada data masukkan	Perangkat lunak berhasil memuat data dan berhasil menampilkan data yang telah di lakukan data *preprocessing*	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 Algortima SVM

ID	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Hasil yang Didapat	Kesimpulan
UC-2-1	Melakukan proses klasifikasi menggunakan metode SVM tanpa seleksi fitur	Menekan tombol "Lakukan Proses Klasifikasi"	Pengguna memilih Nilai C dan Kermel	Menampilkan hasil nilai evasluasi 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 Sekeksi Fitur *Information Gain*

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

	information Gain			fitur dan selisih fitur	fitur dan selisih fitur	

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 Sekeksi Fitur *Chi Square*

ID	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Hasil yang Didapat	Kesimpulan
UC-3-1	Melakukan proses klasifikasi menggunakan metode SVM dan seleksi fitur *Chi Square*	Menekan tombol "Lakukan Proses Klasifikasi"	Pengguna memilih Nilai C, Kernel dan memasukkan nilai *treshold*	Menampilkan hasil nilai evasluasi 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 Sekeksi Fitur *Mutual Information*

ID	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Hasil yang Didapat	Kesimpulan
UC-4-1	Melakukan proses klasifikasi menggunakan metode SVM dan seleksi fitur *Mutual Information*	Menekan tombol "Lakukan Proses Klasifikasi"	Pengguna memilih Nilai C, Kernel dan memasukkan nilai *treshold*	Menampilkan hasil nilai evasluasi 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

4.5.3.7 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-6-1	Melakukan proses pengujian pada data uji berupa pertanyaan berbahasa Indonesia	Menekan tombol "Lakukan Pengujian"	Pengguna memilih Nilai C, Kernel dan memasukkan nilai *treshold*	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.

BAB V
HASIL DAN ANALISIS PENELITIAN

5.1 Pendahuluan

Bab ini akan menguraikan hasil penelitian dari perangkat lunak yang digunakan untuk melakukan proses klasifikasi teks berupa pertanyaan berbahasa Indonesia menggunakan algoritma *Support Vector Machine* (SVM) dan metode seleksi fitur berupa *Information Gain*, *Chi Square* dan *Mutual Information*. Hasil penelitian akan ditampilkan dalam bentuk *matrix* berupa nilai *accuracy, precision, recall, f-measure*, waktu komputasi dan juga jumlah fitur. Akan dilakukan analisis terkait hasil yang didapatkan.

5.2 Data Hasil Penelitian

5.2.1 Konfigurasi Percobaan

Pengujian perangkat lunak dilakukan dengan menggunakan data masukan berupa data pertanyaan yang sebelumnya telah dijelaskan pada subbab 3.3.3. Proses pengujian yang dilakukan telah disesuaikan dengan perancangan yang dibuat pada bab sebelumnya. Pengujian terbagi menjadi empat konfigurasi diantaranya konfigurasi pertama yaitu pengujian menggunakan algoritma SVM tanpa metode seleksi fitur, konfigurasi kedua yaitu pengujian menggunakan algoritma SVM dan metode seleksi fitur *Information Gain*, konfigurasi ketiga yaitu pengujian menggunakan algoritma SVM dan metode seleksi fitur *Chi Square*, dan konfigurasi keempat yaitu pengujian menggunakan algoritma SVM dan metode seleksi fitur *Mutual Information*. Model yang digunakan akan diuji menggunakan *cross*

validation sebanyak 10-fold dengan parameter kernel, nilai C dan *treshold* yang berbeda. Kemudian ditampilkan hasil evaluasi berupa nilai rata-rata yang didapatkan pada hasil pengujian dari 10-fold tersebut. Selanjutnya dilakukan analisa pada perbandingan hasil evaluasi yang telah disajikan dalam bentuk tabel. Hasil evaluasi yang akan ditampilkan pada tabel diantaranya hasil akhir berupa nilai rata-rata dari *accuracy*, *precision, recall*, *f-measure*, waktu komputasi dan jumlah fitur.

5.2.1.1 Data Hasil Konfigurasi 1

Pada data konfigurasi 1 dilakukan pengujian menggunakan algoritma SVM tanpa seleksi fitur yang terdiri dari 3 jenis kernel diantaranya kernel Linear, Polynomial dan Rbf. Serta nilai C yang terdiri dari nilai C: 1 dan nilai C: 10. Hasil Konfigurasi ditampilkan sebagai berikut :

Tabel V-1. Hasil Evaluasi Metode Klasifikasi SVM Tanpa Seleksi Fitur pada Kernel Linear

Nilai C	Nilai *Treshold*	Kernel : Linear					
		Jumlah Fitur	*Accuracy*	*Precision*	*Recall*	*F-Measure*	Waktu Komputasi
0.1	-	2004	0.75	0.51	0.59	0.54	0.14
1	-	2004	0.91	0.92	0.86	0.88	0.12
10	-	2004	0.89	0.9	0.86	0.87	0.14

Tabel V-2. Hasil Evaluasi Metode Klasifikasi SVM Tanpa Seleksi Fitur pada Kernel Polynomial

Nilai C	Nilai *Treshold*	Kernel : Polynomial					
		Jumlah Fitur	*Accuracy*	*Precision*	*Recall*	*F-Measure*	Waktu Komputasi
0.1	-	2004	0.43	0.15	0.33	0.2	0.18
1	-	2004	0.75	0.84	0.61	0.59	0.19
10	-	2004	0.76	0.77	0.63	0.62	0.18

Tabel V-3. Hasil Evaluasi Pada Klasifikasi SVM untuk Kernel Rbf

Nilai C	Nilai *Treshold*	Kernel : Rbf					
		Jumlah Fitur	*Accuracy*	*Precision*	*Recall*	*F-Measure*	Waktu Komputasi
0.1	-	2004	0.44	0.35	0.34	0.21	0.16
1	-	2004	0.87	0.9	0.84	0.81	0.18
10	-	2004	0.88	0.9	0.82	0.84	0.19

Bedasarkan tabel V-1, V-2, dan V-3 dapat dilihat hasil evaluasi yang terjadi pada klasifikasi menggunakan algoritma SVM tanpa seleksi fitur mendapatkan hasil yang berbeda pada setiap parameter, yakni penggunaan parameter kernel dan nilai c. Hasil evaluasi yang telah didapat akan dibandingkan dan dianalisa lebih lanjut.

5.2.1.2 Data Hasil Konfigurasi II

Pada data konfigurasi 2 dilakukan pengujian menggunakan algoritma SVM dengan seleksi fitur *Information Gain* yang terdiri dari 3 jenis kernel diantaranya kernel Linear, Polynomial dan Rbf. Serta nilai C yang terdiri dari nilai C: 0.1 ,C: 1

dan nilai C: 10. Kemudian akan dilakukan juga pengujian pada 3 jenis *treshold* diantaranya 1.66, 1.67 dan 1.68. Hasil konfigurasi ditampilkan sebagai berikut :

Tabel V-4. Hasil Evaluasi Metode Klasifikasi SVM+IG pada Kernel Linear

Nilai C	Nilai *Treshold*	Kernel : Linear					
		Jumlah Fitur	*Accuracy*	*Precision*	*Recall*	*F-Measure*	Waktu Komputasi
0.1	1.66	801	0.76	0.52	0.6	0.55	0.11
	1.67	511	0.44	0.41	0.34	0.22	0.14
	1.68	405	0.72	0.5	0.57	0.52	0.12
1	1.66	630	0.91	0.92	0.85	0.87	0.08
	1.67	432	0.9	0.91	0.85	0.88	0.08
	1.68	350	0.88	0.9	0.83	0.85	0.08
10	1.66	671	0.88	0.87	0.85	0.86	0.08
	1.67	608	0.88	0.88	0.85	0.86	0.07
	1.68	240	0.85	0.83	0.82	0.82	0.07

Tabel V-5. Hasil Evaluasi Metode Klasifikasi SVM+IG pada Kernel Polynomial

Nilai C	Nilai *Treshold*	Kernel : Polynomial					
		Jumlah Fitur	*Accuracy*	*Precision*	*Recall*	*F-Measure*	Waktu Komputasi
0.1	1.66	801	0.43	0.15	0.33	0.2	0.14
	1.67	511	0.45	0.38	0.34	0.22	0.14
	1.68	405	0.5	0.49	0.38	0.29	0.13
1	1.66	630	0.79	0.85	0.69	0.71	0.16
	1.67	432	0.82	0.87	0.74	0.76	0.15
	1.68	350	0.81	0.85	0.73	0.75	0.13
10	1.66	671	0.81	0.84	0.72	0.73	0.15
	1.67	608	0.82	0.87	0.72	0.74	0.13
	1.68	240	0.82	0.85	0.73	0.76	0.13

Tabel V-6. Hasil Evaluasi Pada Klasifikasi SVM+IG untuk Kernel Rbf

Nilai C	Nilai *Treshold*	Kernel : Rbf					
		Jumlah Fitur	*Accuracy*	*Precision*	*Recall*	*F-Measure*	Waktu Komputasi
0.1	1.66	801	0.64	0.47	0.5	0.45	0.14
	1.67	511	0.71	0.49	0.56	0.51	0.12
	1.68	405	0.74	0.51	0.58	0.53	0.12
1	1.66	630	0.88	0.91	0.82	0.84	0.14
	1.67	432	0.89	0.92	0.83	0.86	0.12
	1.68	350	0.88	0.9	0.82	0.84	0.12
10	1.66	671	0.89	0.91	0.84	0.86	0.16
	1.67	608	0.9	0.91	0.86	0.88	0.14
	1.68	240	0.88	0.88	0.84	0.85	0.13

Bedasarkan tabel V-4, V-5, dan V-6 dapat dilihat hasil evaluasi yang terjadi pada klasifikasi menggunakan algoritma SVM menggunakan metode seleksi fitur *Information Gain* mendapatkan hasil yang berbeda pada setiap parameter, yakni penggunaan parameter kernel, nilai C dan *treshold*. Nilai *treshold* yang memiliki hasil evaluasi terbaik dan relatif stabil pada tiap parameternya didapatkan oleh *treshold* 1.67 dengan jumlah fitur sebesar 511. Hal ini dikarenakan *treshold* sebesar 1.67 mampu mengurangi kesalahan model dalam melakukan klasifikasi fitur dengan cara menghilangkan fitur yang kurang relevan. Berdasarkan hasil tersebut, nilai *treshold* yang akan digunakan pada metode seleksi fitur *Information Gain* ialah 1.67. Hasil evaluasi dengan parameter *treshold* terbaik akan dibandingkan dan dianalisa lebih lanjut.

5.2.1.3 Data Hasil Konfigurasi III

Pada data konfigurasi 3 dilakukan pengujian menggunakan algoritma SVM dengan seleksi fitur *Chi Square* yang terdiri dari 3 jenis kernel diantaranya kernel Linear, Polynomial dan Rbf. Serta nilai C yang terdiri dari nilai C: 1 dan nilai C: 10. Kemudian akan dilakukan juga pengujian pada 3 jenis *treshold* diantaranya *treshold* dengan nilai 2.5, 3.5 dan 4.5. Hasil Konfigurasi ditampilkan sebagai berikut :

Tabel V-7. Hasil Evaluasi Metode Klasifikasi SVM+CS pada Kernel Linear

Nilai C	Nilai *Treshold*	Kernel : Linear					
		Jumlah Fitur	*Accuracy*	*Precision*	*Recall*	*F-Measure*	Waktu Komputasi
0.1	2.5	630	0.78	0.53	0.62	0.56	0.17
	3.5	432	0.79	0.53	0.62	0.57	0.13
	4.5	350	0.79	0.53	0.62	0.57	0.06
1	2.5	630	0.92	0.92	0.93	0.87	0.08
	3.5	432	0.92	0.93	0.89	0.91	0.08
	4.5	350	0.91	0.92	0.92	0.9	0.07
10	2.5	630	0.91	0.91	0.88	0.89	0.07
	3.5	432	0.9	0.91	0.88	0.89	0.07
	4.5	350	0.9	0.9	0.87	0.88	0.05

Tabel V-8. Hasil Evaluasi Metode Klasifikasi SVM+CS pada Kernel Polynomial

Nilai C	Nilai *Treshold*	Kernel : Polynomial					
		Jumlah Fitur	*Accuracy*	*Precision*	*Recall*	*F-Measure*	Waktu Komputasi
0.1	2.5	630	0.59	0.5	0.47	0.4	0.11
	3.5	432	0.74	0.52	0.58	0.53	0.11
	4.5	350	0.78	0.53	0.62	0.57	0.08
	2.5	630	0.85	0.89	0.75	0.77	0.12

1	3.5	432	0.92	0.9	0.8	0.77	0.12
	4.5	350	0.91	0.91	0.88	0.89	0.16
10	2.5	630	0.85	0.88	0.76	0.78	0.12
	3.5	432	0.87	0.89	0.81	0.83	0.1
	4.5	350	0.9	0.9	0.88	0.89	0.09

Tabel V-9. Hasil Evaluasi Pada Klasifikasi SVM+CS untuk Kernel Rbf

Nilai C	Nilai *Treshold*	Kernel : Rbf					
		Jumlah Fitur	*Accuracy*	*Precision*	*Recall*	*F-Measure*	Waktu Komputasi
0.1	2.5	630	0.75	0.51	0.59	0.54	0.11
	3.5	432	0.77	0.52	0.61	0.56	0.09
	4.5	350	0.79	0.53	0.62	0.57	0.08
1	2.5	630	0.91	0.93	0.85	0.87	0.11
	3.5	432	0.91	0.93	0.89	0.9	0.08
	4.5	350	0.92	0.92	0.9	0.91	0.07
10	2.5	630	0.92	0.93	0.88	0.89	0.11
	3.5	432	0.92	0.92	0.9	0.91	0.09
	4.5	350	0.91	0.91	0.9	0.9	0.07

Bedasarkan tabel V-7, V-8, dan V-9 dapat dilihat hasil evaluasi yang terjadi pada klasifikasi menggunakan algoritma SVM menggunakan metode seleksi fitur *Chi-Square* mendapatkan hasil yang berbeda pada setiap parameter, yakni penggunaan parameter kernel, nilai C dan *treshold*. Nilai *treshold* yang memiliki hasil evaluasi terbaik dan relatif stabil didapatkan oleh *treshold* 3.5 dengan jumlah fitur sebesar 432. Hal ini dikarenakan *treshold* sebesar 3.5 mampu mengurangi kesalahan model dalam melakukan klasifikasi fitur dengan cara menghilangkan fitur yang kurang relevan. Berdasarkan hasil tersebut, nilai *treshold* yang akan digunakan pada metode seleksi fitur *Chi Square* ialah 3.5. Hasil evaluasi dengan

parameter *treshold* terbaik akan dibandingkan dan dianalisa lebih lanjut.

5.2.1.4 Data Hasil Konfigurasi IV

Pada data konfigurasi 4 dilakukan pengujian menggunakan algoritma SVM dengan seleksi fitur M*utual Information* yang terdiri dari 3 jenis kernel diantaranya kernel Linear, Polynomial dan Rbf. Serta nilai C yang terdiri dari nilai C: 1 dan nilai C: 10. Kemudian akan dilakukan juga pengujian pada 3 jenis *treshold* diantaranya *treshold* dengan nilai 0.0002, 0.0003 dan 0.0004. Hasil Konfigurasi ditampilkan sebagai berikut :

Tabel V-10. Hasil Evaluasi Metode Klasifikasi SVM+MI pada Kernel Linear

Nilai C	Nilai *Treshold*	Kernel : Linear					
		Jumlah Fitur	*Accuracy*	*Precision*	*Recall*	*F-Measure*	Waktu Komputasi
0.1	0.0002	671	0.75	0.51	0.59	0.54	0.11
	0.0003	608	0.77	0.52	0.61	0.56	0.09
	0.0004	240	0.79	0.53	0.62	0.57	0.08
1	0.0002	671	0.91	0.93	0.85	0.87	0.11
	0.0003	608	0.91	0.93	0.89	0.9	0.08
	0.0004	240	0.92	0.92	0.9	0.91	0.07
10	0.0002	671	0.92	0.93	0.88	0.89	0.11
	0.0003	608	0.92	0.92	0.9	0.91	0.09
	0.0004	240	0.91	0.91	0.9	0.9	0.07

Tabel V-11. Hasil Evaluasi Metode Klasifikasi SVM+MI pada Kernel Polynomial

Nilai C	Nilai *Treshold*	Kernel : Polynomial					
		Jumlah Fitur	*Accuracy*	*Precision*	*Recall*	*F-Measure*	Waktu Komputasi
	0.0002	671	0.55	0.49	0.44	0.36	0.13

0.1	0.0003	608	0.66	0.5	0.53	0.47	0.11
	0.0004	240	0.76	0.51	0.6	0.55	0.09
1	0.0002	671	0.83	0.87	0.72	0.74	0.12
	0.0003	608	0.86	0.88	0.76	0.78	0.13
	0.0004	240	0.89	0.9	0.84	0.86	0.1
10	0.0002	671	0.84	0.88	0.75	0.77	0.13
	0.0003	608	0.86	0.88	0.78	0.8	0.11
	0.0004	240	0.89	0.88	0.85	0.86	0.11

Tabel V-12. Hasil Evaluasi Pada Klasifikasi SVM+MI untuk Kernel Rbf

Nilai C	Nilai *Treshold*	Kernel : Polynomial					
		Jumlah Fitur	*Accuracy*	*Precision*	*Recall*	*F-Measure*	Waktu Komputasi
0.1	0.0002	671	0.74	0.51	0.58	0.53	0.11
	0.0003	608	0.75	0.52	0.6	0.55	0.1
	0.0004	240	0.78	0.52	0.62	0.56	0.08
1	0.0002	671	0.9	0.92	0.85	0.87	0.12
	0.0003	608	0.91	0.93	0.87	0.89	0.12
	0.0004	240	0.91	0.92	0.87	0.89	0.09
10	0.0002	671	0.91	0.92	0.87	0.89	0.14
	0.0003	608	0.91	0.93	0.88	0.89	0.13
	0.0004	240	0.91	0.9	0.88	0.89	0.09

Bedasarkan tabel V-10, V-11, dan V-12 dapat dilihat hasil evaluasi yang terjadi pada klasifikasi menggunakan algoritma SVM menggunakan metode seleksi fitur *Mutual Information* mendapatkan hasil yang berbeda pada setiap parameter, yakni penggunaan parameter kernel, nilai C dan *treshold*. Nilai *treshold* yang memiliki hasil evaluasi terbaik dan relatif stabil pada tiap parameternya didapatkan oleh *treshold* 0.0004 dengan jumlah fitur sebesar 240. Hal ini dikarenakan *treshold* sebesar 0.0004 mampu mengurangi kesalahan model dalam melakukan klasifikasi

fitur dengan cara menghilangkan fitur yang kurang relevan. Berdasarkan hasil tersebut, maka nilai k yang akan digunakan pada metode seleksi fitur *Mutual Information* ialah 0.0004. Hasil evaluasi dengan parameter *treshold* terbaik akan dibandingkan dan dianalisa lebih lanjut.

5.2.1.5 Perbandingan Data Hasil Konfigurasi

Berikut data perbandingan hasil klasifikasi menggunakan SVM tanpa seleksi fitur, SVM dengan seleksi fitur *Information Gain*, SVM dengan seleksi fitur *Chi Square* dan SVM dengan seleksi fitur *Mutual Information*.

Tabel V-13. Hasil Evaluasi Metode Klasifikasi Model SVM pada Kernel Linear

Algoritma	Nilai C	Nilai *Treshold*	Kernel : Linear					
			Accuracy	*Precision*	*Recall*	*F-Measure*	Waktu Komputasi	Jumlah Fitur
SVM	0.1	-	0.75	0.51	0.59	0.54	0.14	2004
	1		0.91	0.92	0.86	0.88	0.12	2004
	10		0.89	0.9	0.86	0.87	0.14	2004
SVM+IG	0.1	1.67	0.44	0.41	0.34	0.22	0.14	511
	1		0.9	0.91	0.85	0.88	0.08	511
	10		0.88	0.88	0.85	0.86	0.07	511
SVM+CS	0.1	3.5	0.79	0.53	0.62	0.57	0.13	432
	1		0.92	0.93	0.89	0.91	0.08	432
	10		0.9	0.91	0.88	0.89	0.07	432
SCM+MI	0.1	0.0004	0.79	0.53	0.62	0.57	0.07	240
	1		0.92	0.92	0.89	0.9	0.07	240
	10		0.9	0.9	0.88	0.89	0.05	240

Tabel V-14. Hasil Evaluasi Metode Klasifikasi Model SVM pada Kernel Polynomial

Algoritma	Nilai C	Nilai *Treshold*	Kernel : Polynomial					
			Accuracy	*Precision*	*Recall*	*F-Measure*	Waktu Komputasi	Jumlah Fitur
SVM	0.1		0.43	0.15	0.33	0.2	0.18	2004
	1		0.75	0.84	0.61	0.59	0.19	2004
	10		0.76	0.77	0.63	0.62	0.18	2004
SVM+IG	0.1	1.67	0.45	0.38	0.34	0.22	0.14	511
	1		0.82	0.87	0.74	0.76	0.15	511
	10		0.82	0.87	0.72	0.74	0.13	511
SVM+CS	0.1	3.5	0.74	0.52	0.58	0.53	0.11	432
	1		0.92	0.9	0.8	0.77	0.12	432
	10		0.87	0.89	0.81	0.83	0.1	432
SCM+MI	0.1	0.0004	0.76	0.51	0.6	0.55	0.09	240
	1		0.89	0.9	0.84	0.86	0.1	240
	10		0.89	0.88	0.85	0.86	0.11	240

Tabel V-15. Hasil Evaluasi Pada Klasifikasi Model SVM untuk Kernel Rbf

Algoritma	Nilai C	Nilai *Treshold*	Kernel : Rbf					
			Accuracy	*Precision*	*Recall*	*F-Measure*	Waktu Komputasi	Jumlah Fitur
SVM	0.1	-	0.44	0.35	0.34	0.21	0.16	2004
	1		0.87	0.9	0.84	0.81	0.18	2004
	10		0.88	0.9	0.82	0.84	0.19	2004
SVM+IG	0.1	1.67	0.71	0.49	0.56	0.51	0.12	511
	1		0.89	0.92	0.83	0.86	0.12	511
	10		0.9	0.91	0.86	0.88	0.14	511
SVM+CS	0.1	3.5	0.77	0.52	0.61	0.56	0.09	432
	1		0.91	0.93	0.89	0.9	0.08	432
	10		0.92	0.92	0.9	0.91	0.09	432
SCM+MI	0.1	0.0004	0.78	0.52	0.62	0.56	0.08	240
	1		0.91	0.92	0.87	0.89	0.09	240
	10		0.91	0.9	0.88	0.89	0.09	240

Bedasarkan tabel V-13, V-14, dan V-15 dapat dilihat perbandingan dari

hasil evaluasi yang terbagi menjadi 3 kernel diantaranya kernel Linear, Polynomial dan Rbf. Hasil dari tabel perbandingan selanjutnya akan diubah kedalam bentuk grafik. Grafik tersebut ditujukan untuk menjelaskan pengaruh parameter yang digunakan di SVM dan metode seleksi fitur dalam bentuk visualisasi. Selanjutnya, pada analisis hasil penelitian akan dibahas berbagai fenomena yang ditemukan pada hasil klasifikasi. Hasil dari analisis diharapkan dapat melihat pengaruh dari penggunaan metode seleksi fitur dan juga dapat menjawab rumusan masalah yang menjadi landasan pada penelitian ini.

5.2.1.6 Data Konfigurasi Hasil Pengujian Klasifikasi Pertanyaan

Berikut data perbandingan hasil klasifikasi pada pengujian menggunakan berbagai model yang sebelumnya telah dilatih. Masukan berupa pertanyaan berbahasa Indonesia pada pengujian ini terbagi 3, pertanyaan untuk label *factoid* yaitu "Siapa nama presiden rusia", pertanyaan untuk label *non-factoid* yaitu "Bagaimana cara menggunakan sumpit" dan pertanyaan untuk label *others* yaitu "Apa saja yang menjadi syarat melakukan ibadah puasa". Status factoid, non-factoid dan others diberi label berhasil jika pertanyaan diprediksi dengan benar berdasarkan data aktual.

Tabel V-13. Data Hasil Pengujian Prediksi Klasifikasi Pertanyaan menggunakan Kernel Linear

Algoritma	Nilai C	Nilai *Treshold*	Kernel : Linear		
			Status *Factoid*	Status *Non-Factoid*	Status *Others*
SVM	0.1	-	Berhasil	Berhasil	Tidak Berhasil
	1		Berhasil	Berhasil	Berhasil

	10		Berhasil	Berhasil	Berhasil
SVM+IG	0.1	1.67	Berhasil	Berhasil	Tidak Berhasil
	1		Berhasil	Berhasil	Berhasil
	10		Berhasil	Berhasil	Berhasil
SVM+CS	0.1	3.5	Berhasil	Berhasil	Tidak Berhasil
	1		Berhasil	Berhasil	Berhasil
	10		Berhasil	Berhasil	Tidak Berhasil
SVM+MI	0.1	0.0004	Berhasil	Berhasil	Tidak Berhasil
	1		Berhasil	Berhasil	Berhasil
	10		Berhasil	Berhasil	Berhasil

Tabel V-14. Data Hasil Pengujian Prediksi Klasifikasi Pertanyaan menggunakan Kernel Polynomial

Algoritma	Nilai C	Nilai *Treshold*	Kernel : Polynomial		
			Status *Factoid*	Status *Non-Factoid*	Status *Others*
SVM	0.1	-	Berhasil	Tidak Berhasil	Tidak Berhasil
	1		Berhasil	Berhasil	Tidak Berhasil
	10		Berhasil	Berhasil	Tidak Berhasil
SVM+IG	0.1	1.67	Berhasil	Berhasil	Tidak Berhasil
	1		Berhasil	Berhasil	Tidak Berhasil
	10		Berhasil	Berhasil	Tidak Berhasil
SVM+CS	0.1	3.5	Berhasil	Berhasil	Tidak Berhasil
	1		Berhasil	Berhasil	Berhasil
	10		Berhasil	Berhasil	Berhasil
SVM+MI	0.1	0.0004	Berhasil	Berhasil	Tidak Berhasil
	1		Berhasil	Berhasil	Berhasil
	10		Berhasil	Berhasil	Berhasil

Tabel V-15. Data Hasil Pengujian Prediksi Klasifikasi Pertanyaan menggunakan Kernel Rbf

Algoritma	Nilai C	Nilai *Treshold*	Kernel : Rbf		
			Status *Factoid*	Status *Non-Factoid*	Status *Others*

SVM	0.1	-	Berhasil	Tidak Berhasil	Tidak Berhasil
	1		Berhasil	Berhasil	Tidak Berhasil
	10		Berhasil	Berhasil	Berhasil
SVM+IG	0.1	1.67	Berhasil	Berhasil	Tidak Berhasil
	1		Berhasil	Berhasil	Tidak Berhasil
	10		Berhasil	Berhasil	Berhasil
SVM+CS	0.1	3.5	Berhasil	Berhasil	Tidak Berhasil
	1		Berhasil	Berhasil	Berhasil
	10		Berhasil	Berhasil	Berhasil
SVM+MI	0.1	0.0004	Berhasil	Berhasil	Tidak Berhasil
	1		Berhasil	Berhasil	Berhasil
	10		Berhasil	Berhasil	Berhasil

5.3 Analisis Hasil Penelitian

5.3.1 Analisis Kernel, Nilai C dan *Treshold*

Penggunaan algoritma SVM sebagai model klasifikasi menggunakan masukan parameter yakni parameter kernel dan juga nilai c. Data hasil klasifikasi yang telah ditampilkan pada subbab sebelumnya menunjukkan hasil evaluasi yang berbeda pada tiap penggunaan parameternya. Penggunan parameter masukan pada model klasifikasi yang digunakan akan dianalisis sebagai berikut.

5.3.1.1 Kernel Linear

Gambar V-1. Grafik Data Perbandingan Hasil Model Klasifikasi Kernel Linear & C: 0.1

Gambar V-2. Grafik Data Perbandingan Hasil Model Klasifikasi Kernel Linear & C: 1

Gambar V-3. Grafik Data Perbandingan Hasil Model Klasifikasi Kernel Linear & C: 10

Grafik yang ditampilkan pada gambar diatas menunjukkan bahwa penggunaan nilai C : 1 dan C : 10 bekerja dengan baik dan cukup stabil pada kernel linear. Hasil akurasi terbaik sebesar 0.92 diperoleh oleh penggunaan metode seleksi fitur yakni *Chi Square* dan *Mutual Information* dengan penggunaan nilai C:1. Sedangkan, pada penggunaan nilai C : 0,1 memberikan hasil kinerja yang buruk pada setiap model klasifikasi di penggunaan kernel Linear. Hal ini memberikan fenomena bahwa pemilihan nilai C yang tepat diperlukan untuk mempengaruhi hasil akhir model klasifikasi.

Nilai C sendiri digunakan untuk mengontrol trade off antara margin dan error klasifikasi. Hal ini bertujuan untuk menginformasikan optimasi SVM seberapa banyak kesalahan klasifikasi yang ingin dihindari pada proses pelatihan. Untuk penggunaan nilai C yang sangat kecil menyebabkan pengoptimal mencari

hyperlane dengan fungsi pemisah dengan margin yang lebih besar. Hal ini menyebabkan kemungkinan adanya dot.product atau fitur yang diabaikan sehingga rasio kesalahan dalam prediksi menjadi lebih besar.

Pada kernel Linear ditampilkan juga hasil dari data evaluasi berupa waktu komputasi yang diperlukan untuk kinerja SVM menunjukkan hasil yang cukup baik untuk setiap model klasifikasi yang digunakan. Kernel linear bekerja dengan membagi data secara *linear* yang mana cara ini lebih sederhana dibandingkan dengan penggunaan kernel non-linear. Sehingga pada penggunaan kernel ini membutuhkan waktu pemrosesan yang lebih sedikit dibanding penggunaan kernel lain. Dan juga penggunaan metode seleksi fitur berupa *Information Gain, Chi Square* dan *Mutual Information* menunjukkan fenomena bahwa penggunaan metode seleksi fitur terbukti efektif mampu mengurangi waktu komputasi pada kinerja algoritma SVM. Metode seleksi fitur *Mutual Information* menunjukkan kinerja terbaik dalam mengurangi waktu komputasi pada kinerja SVM dengan memberikan waktu komputasi sebesar 4 s. Hal ini disebabkan penggunaan treshold pada metode seleksi fitur mampu mengurangi jumlah fitur secara signifikan yaitu menjadi sebesar 240 fitur. Semakin sedikit jumlah fitur maka waktu komputasi yang diperlukan pada kinerja SVM juga semakin berkurang.

Secara keseluruhan penggunaan model SVM baik tanpa seleksi fitur maupun dengan seleksi fitur bekerja cukup efektif dan stabil pada kernel linear terutama dengan nilai C:1 dan C:10. Penggunaan metode seleksi fitur *Mutual Information* dan *Chi Square* mampu memberikan peningkatan pada hasil akhir data evaluasi namun peningkatan hasil yang diberikan tidak terlalu signifikan.

Sedangkan, pada penggunaan seleksi fitur *Information Gain* memberikan hasil evaluasi yang sedikit menurun jika dibandingkan dengan model SVM tanpa seleksi fitur.

5.3.1.2 Kernel Polynomial

Gambar V-4. Grafik Data Perbandingan Hasil Model Klasifikasi Kernel Polynomial & C: 0.1

Gambar V-5. Grafik Data Perbandingan Hasil Model Klasifikasi Kernel Polynomial & C: 1

Gambar V-6. Grafik Data Perbandingan Hasil Model Klasifikasi Kernel Polynomial & C: 10

Grafik yang ditampilkan pada gambar diatas menunjukkan bahwa penggunaan kernel polynomial dengan nilai C : 0.1 , C : 1 dan C : 10 menunjukkan hasil evaluasi yang kurang baik dan cenderung tidak stabil pada tiap model klasifikasi yang digunakan. Berdasarkan grafik tersebut didapatkan fenomena bahwa penggunaan kernel polynomial menghasilkan hasil evaluasi yang kurang baik jika dibandingkan dengan penggunaan kernel lain. Sehingga didapatkan fenomena bahwa pemilihan kernel yang tepat diperlukan untuk melakukan optimasi pada algoritma SVM.

Berdasarkan grafik diatas didapatkan informasi bahwa penggunaan Nilai C : 0,1 menghasilkan hasil evaluasi yang sangat buruk bila dibandingkan dengan penggunaan nilai C : 1 dan nilai C : 10. Bahkan, penggunaan nilai C : 0,1 pada model SVM tanpa seleksi fitur hanya mendapatkan *accuracy* sebesar 0.44. Hal ini menunjukkan bahwa diperlukan pemilihan nilai C yang tepat untuk mendapatkan hasil evaluasi yang baik pada algortima SVM. Parameter nilai C bertujuan untuk memberikan informasi pada optimasi SVM seberapa banyak kesalahan klasifikasi yang ingin dihindari pada proses pelatihan. Untuk penggunaan nilai C yang sangat kecil menyebabkan fungsi pengoptimal mencari hyperlane pada fungsi pemisah dengan margin yang lebih besar. Hal ini menyebabkan kemungkinan adanya dot.product atau fitur yang diabaikan sehingga rasio kesalahan dalam prediksi menjadi lebih besar.

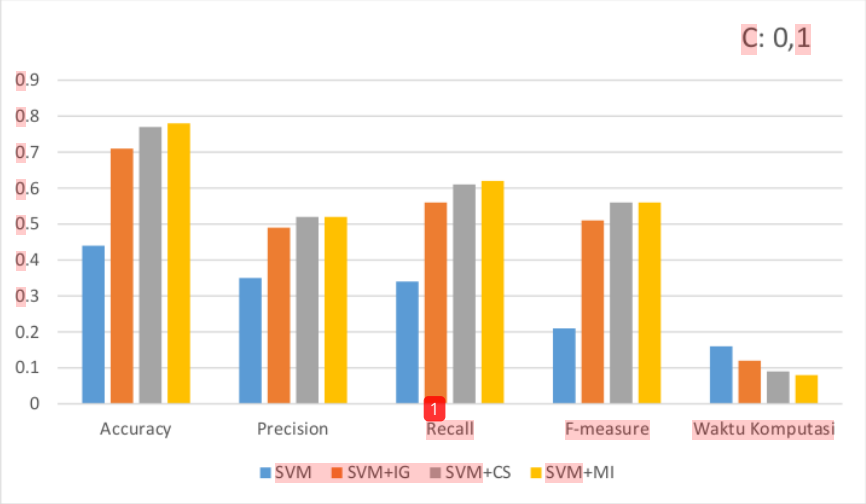
Pada kernel polynomial juga ditampilkan hasil dari data evaluasi berupa waktu komputasi yang diperlukan untuk kinerja SVM pada tiap model menunjukkan hasil yang lebih lama dibandingkan dengan waktu komputasi pada

penggunaan kernel linear. Kernel Polynomial termasuk kedalam *non-linear* kernel model yang bekerja dengan cara mengubah data ke dalam bentuk ruang fitur (*feature space*) berdimensi tinggi. Oleh karena itu, data dapat dipisah secara linear pada ruang fitur. Ruang fitur memiliki dimensi yang lebih tinggi dari vector *input*. Sehingga komputasi pada ruang fitur menjadi sangat besar yang mengakibatkan ruang fitur akan memiliki jumlah fitur yang tidak terhigga. Penggunaan metode seleksi fitur berupa *Information Gain, Chi Square* dan *Mutual Information* menunjukkan fenomena bahwa penggunaan metode seleksi fitur tersebut mampu mengurangi waktu komputasi pada kinerja algoritma klasifikasi SVM. Metode seleksi fitur *Mutual Information* menunjukkan kinerja terbaik dalam mengurangi waktu komputasi pada kinerja SVM, dengan memberikan waktu komputasi sebesar 9 s pada penggunaan parameter nilai C : 0.1. Penggunaan *treshold* 0.0004 pada metode seleksi fitur *Mutual Information* mampu mengurangi jumlah fitur secara signifikan yaitu menjadi sebesar 240 fitur. Semakin sedikit jumlah fitur maka waktu komputasi yang diperlukan pada kinerja SVM juga semakin berkurang.

Penggunaan metode seleksi fitur berupa *Information Gain, Chi Square* dan *Mutual Information* menunjukkan kinerja yang baik pada kernel polynomial. Walaupun secara keseluruhan hasil evaluasi yang didapat pada kernel ini kurang baik dan cenderung tidak stabil, penggunaan metode seleksi fitur tersebut mampu memberikan hasil yang cukup signifikan dalam meningkatkan hasil evaluasi pada proses klasifikasi SVM menggunakan kernel polynomial, terutama pada metode seleksi fitur *Chi Square* dan *Mutual Information*. Pada penggunaan parameter nilai C : 1 SVM tanpa seleksi fitur hanya mendapatkan hasil akurasi sebesar 0.75.

Dengan menggunakan metode *Information Gain* hasil akurasi mampu meningkat sebesar 0.82, metode *Chi Square* mampu meningkatkan hasil akurasi menjadi 0.92 dan Metode *Mutual Information* mampu meningkatkan hasil akurasi sebesar 0.89. Hal ini memberikan fenomena bahwa metode *Chi Square* mampu memberikan fenomena terbaik dalam meningkatkan hasil evaluasi pada penggunaan kernel polynomial.

5.3.1.3 Kernel Rbf



Gambar V-7. Grafik Data Perbandingan Hasil Model Klasifikasi Kernel Rbf & C: 0.1

Gambar V-8. Grafik Data Perbandingan Hasil Model Klasifikasi Kernel Rbf & C: 1

Gambar V-9. Grafik Data Perbandingan Hasil Model Klasifikasi Kernel Rbf & C: 10

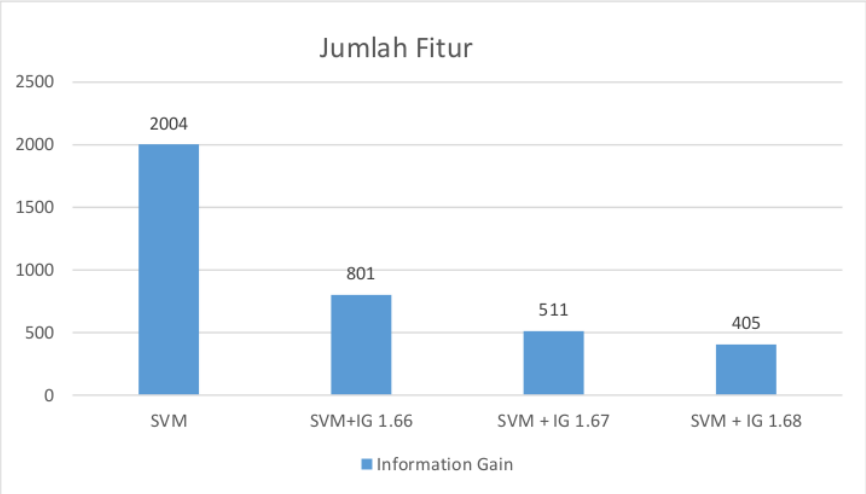
Grafik yang ditampilkan pada gambar diatas menunjukkan bahwa penggunaan kernel Rbf terutama dengan parameter nilai C : 1 dan C : 10 menunjukkan hasil evaluasi yang sangat baik dan cenderung stabil pada tiap model klasifikasi. Sedangkan kinerja Rbf cenderung kurang baik dan tidak stabil pada penggunaan parameter C : 0,1. Hal ini menunjukkan fenomena bahwa pemilihan nilai C yang tepat dapat memberikan hasil evaluasi terbaik. Penggunaan nilai C yang sangat kecil menyebabkan pengoptimal mencari hyperlane dengan fungsi pemisah dengan margin yang lebih besar. Hal ini menyebabkan kemungkinan adanya dot.product atau fitur yang diabaikan sehingga rasio kesalahan dalam prediksi menjadi lebih besar.

Pada kernel Rbf juga ditampilkan hasil dari data evaluasi berupa waktu komputasi yang diperlukan untuk kinerja SVM pada tiap model menunjukkan hasil yang lebih lama dibandingkan penggunaan kernel linear. Kernel Rbf termasuk kedalam *non-linear* kernel model yang bekerja dengan cara mengubah data ke dalam bentuk ruang fitur (*feature space*) berdimensi tinggi. Oleh karena itu, data dapat dipisah secara linear pada ruang fitur. Ruang fitur memiliki dimensi yang lebih tinggi dari vector input. Sehingga komputasi pada ruang fitur menjadi sangat besar yang mengakibatkan ruang fitur akan memiliki jumlah fitur yang tidak terhigga. Hal ini menyebabkan waktu komputasi pada kernel Rbf menjadi lebih besar. Penggunaan metode seleksi fitur berupa *Information Gain, Chi Square* dan *Mutual Information* menunjukkan fenomena bahwa penggunaan metode seleksi fitur terbukti efektif mampu mengurangi waktu komputasi pada kinerja algoritma SVM. Metode seleksi fitur *Mutual Information* menunjukkan kinerja terbaik dalam

mengurangi waktu komputasi pada kinerja SVM dengan memberikan waktu komputasi sebesar 8 s pada parameter nilai C : 0.1 . Hal ini disebabkan penggunaan treshold pada metode seleksi fitur mampu mengurangi jumlah fitur secara signifikan yaitu menjadi sebesar 240 fitur. Semakin sedikit jumlah fitur maka waktu komputasi yang diperlukan pada kinerja SVM juga semakin berkurang.

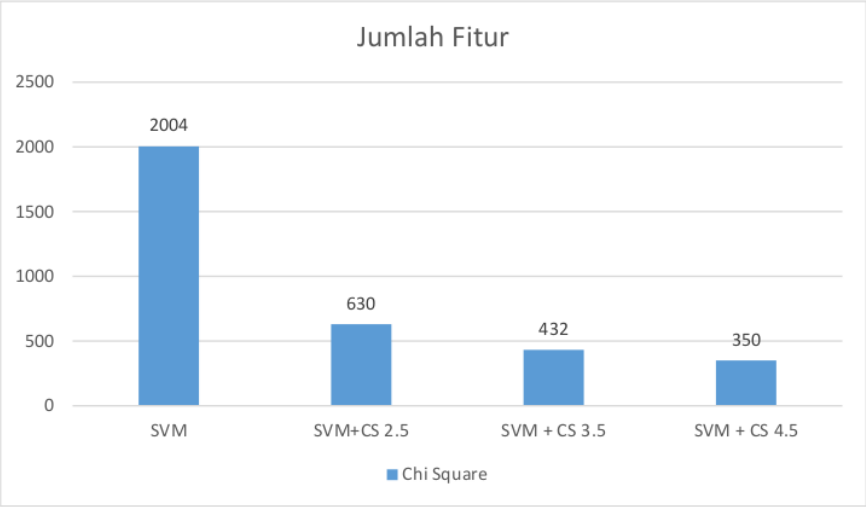
Penggunaan metode seleksi fitur berupa *Information Gain, Chi Square* dan *Mutual Information* menunjukkan kinerja yang baik pada kernel Rbf. Penggunaan metode seleksi fitur tersebut mampu meningkatkan hasil evaluasi pada proses klasifikasi SVM menggunakan kernel Rbf pada metode seleksi fitur *Information Gain*, *Chi Square* dan *Mutual Information*. Pada penggunaan parameter nilai C : 10 SVM tanpa seleksi fitur mendapatkan hasil akurasi yaitu 0.88. Dengan menggunakan metode *Information Gain* mampu meningkatkan hasil akurasi menjadi 0.9, metode *Chi Square* mampu meningkatkan hasil akurasi menjadi 0.92 dan metode *Mutual Information* mampu meningkatkan hasil akurasi sebesar 0.91. Hal ini memberikan fenomena bahwa metode *Chi Square* mampu memberikan kinerja terbaik dalam meningkatkan hasil evaluasi pada penggunaan kernel Rbf.

5.3.2 Analisis Jumlah Fitur



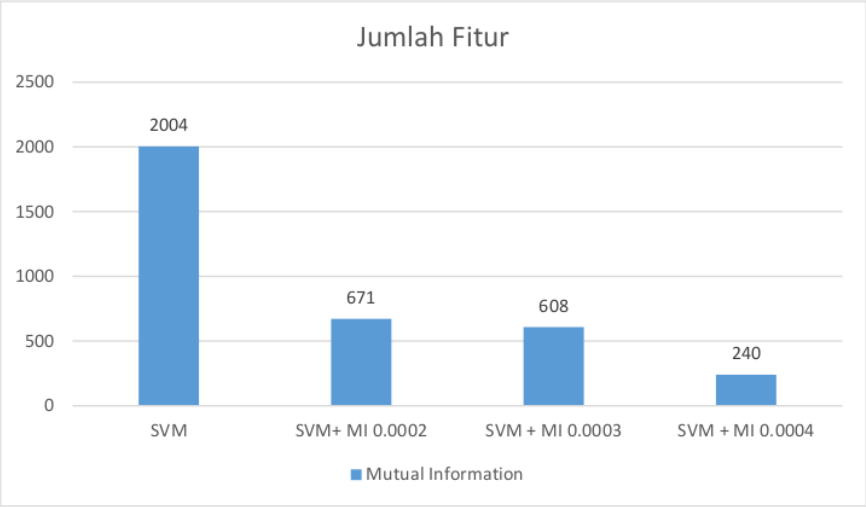
Gambar V-10. Grafik Data Perbandingan Jumlah Fitur Metode Information Gain

Berdasarkan grafik yang ditampilkan pada gambar V-10 penggunaan metode seleksi fitur *Information Gain* dengan *treshold* 1.66, 1.67 dam 1.68 memberikan hasil berupa pengurangan pada jumlah fitur. Terlihat pada grafik bahwa semakin tinggi *treshold* yang digunakan maka jumlah fitur akan semakin berkurang, dan berlaku pula sebaliknya. Hal ini dikarenakan semakin tinggi nilai *treshold* yang diberikan maka nilai ambang batas untuk fitur dapat terseleksi juga semakin besar.



Gambar V-11. Grafik Data Perbandingan Jumlah Fitur Metode Chi Square

Berdasarkan grafik yang ditampilkan pada gambar V-11 penggunaan metode seleksi fitur *Chi Square* dengan *treshold* 2.5, 3.5 dam 4.5 memberikan hasil berupa pengurangan pada jumlah fitur. Terlihat pada grafik bahwa semakin tinggi *treshold* yang digunakan maka jumlah fitur akan semakin berkurang, dan berlaku pula sebaliknya. Hal ini dikarenakan semakin tinggi nilai *treshold* yang diberikan maka nilai ambang batas untuk fitur dapat terseleksi juga semakin besar.



Gambar V-12. Grafik Data Perbandingan Jumlah Fitur Metode Mutual Information

Berdasarkan grafik yang ditampilkan pada gambar V-12 penggunaan metode seleksi fitur *Mutual Information* dengan *treshold* 0.0002, 0.0003 dam 0.0004 memberikan hasil berupa pengurangan pada jumlah fitur. Terlihat pada grafik bahwa semakin tinggi *treshold* yang digunakan maka jumlah fitur akan semakin berkurang, dan berlaku pula sebaliknya. Hal ini dikarenakan semakin tinggi nilai *treshold* yang diberikan maka nilai ambang batas untuk fitur dapat terseleksi juga semakin besar.

5.3.3 Analisis Hasil Kinerja Metode Seleksi Fitur

Penggunaan metode seleksi fitur berpengaruh dalam mereduksi data yang kurang relevan sehingga dapat mempercepat waktu komputasi dan meningkatkan hasil evaluasi. Berdasarkan data hasil kinerja berupa grafik yang ditampilkan pada subbab 5.3.1 ditunjukkan bahwasanya penggunaan metode seleksi fitur berupa

Information Gain, Chi Square, dan *Mutual Information* mampu mempercepat waktu komputasi dan memberikan peningkatan hasil kinerja pada setiap parameter yang ada pada Algoritma *Support Vector Machine*. Metode seleksi fitur *Chi Square* dan *Mutual Information* memberikan hasil yang sangat baik dan relatif stabil untuk setiap parameter di SVM. Perbedaan hasil kinerja yang ditunjukkan oleh dua metode seleksi fitur tersebut tidak terlalu signifikan, hal ini disebabkan karna cara kerja perhitungan bobot pada metode seleksi fitur *Chi Square* dan *Mutual Information* memiliki kesamaan yaitu dengan cara mencari informasi pada setiap term untuk menghitung ketergantungan kelas pada suatu fitur. Metode seleksi fitur Chi Square mendapatkan nilai akurasi terbaik sebesar 0.92 dengan jumlah fitur sebesar 432 dan waktu komputasi sebesar 9 s pada kernel linear dengan nilai C :1 dan *treshold* : 3.5. Kemudian, Metode seleksi fitur *Mutual Information* mendapatkan nilai akurasi terbaik sebesar 0.92 dengan jumlah fitur sebesar 240 dan waktu komputasi sebesar 7 s pada kernel linear dengan nilai C :1 dan *treshold* : 0.0004.

Sedangkan, penggunaan metode seleksi fitur *Information Gain* menunjukkan hasil kinerja yang cukup baik namun relatif kurang stabil dikarenakan terjadi penurunan hasil kinerja pada beberapa parameter yang ada di algoritma SVM. Metode seleksi fitur *Information Gain* bekerja dengan menghitung informasi pada setiap term kemudian melakukan teknik *scoring* untuk menentukan bobot menggunakan maksimal entropy. Metode seleksi fitur Information Gain mendapatkan nilai akurasi terbaik sebesar 0.9 dengan jumlah fitur sebesar 511 dan waktu komputasi sebesar 8 s pada kernel linear dengan nilai C :1 dan *treshold* :

1.67. Hal ini memberikan fenomena bahwa perhitungan bobot pada setiap metode seleksi fitur sangat berpengaruh dalam mereduksi term yang kurang relevan sehingga dapat memberikan peningkatan pada hasil kinerja.

5.3.4 Analisis Hasil Prediksi Pengujian Klasifikasi

Berdasarkan data hasil prediksi pengujian klasifikasi yang ditampilkan pada tabel V-13, V-14 dan V-15 dengan data masukkan berupa pertanyaan berbahasa Indonesia menunjukkan hasil yang cukup baik terutama saat melakukan prediksi untuk data masukkan berupa pertanyaan dengan label *factoid* dan *non-factoid*. Sedangkan untuk data masukkan berupa pertanyaan dengan label *other*, model cukup banyak melakukan kesalahan saat melakukan prediksi pada data masukkan dengan label tersebut. Hal ini dikarenakan dataset yang digunakan pada penelitian ini belum dapat dikatakan seimbang dimana data untuk pertanyaan dengan label factoid berjumlah 519, data dengan label non-factoid berjumlah 491, dan data dengan label others berjumlah 185. Jumlah data yang terlalu sedikit pada data pertanyaan dengan label others menyebabkan terjadinya permasalahan *imbalance* data. Sehingga model cukup banyak melakukan kesalahan pada saat melakukan prediksi pengujian klasifikasi pada data uji dengan label *other*.

5.4 Kesimpulan

Pada bab ini diuraikan analisa yang dilakukan berdasarkan hasil pengujian yang didapatkan pada penelitian perbandingan metode seleksi fitur pada klasifikasi pertanyaan berbahasa Indonesia menggunakan algoritma *Support Vector Machine*. Berdasarkan analisa yang telah diuraikan sebelumnya, dapat disimpulkan bahwa penggunaan metode seleksi fitur berupa *Information Gain, Chi Square* dan *Mutual*

Information memberikan pengaruh dalam meningkatkan hasil akurasi, mengurangi jumlah fitur , dan mengurangi waktu komputasi pada kinerja algoritma SVM. Penggunaan metode seleksi fitur *Chi Square* pada algortima SVM dengan kernel linear dan parameter C: 1 menghasilkan kinerja terbaik dengan rata-rata *accuracy* : 0.92 , *precision* : 0.93, *recall* : 0.89, *f-measure* : 0.91 dan waktu komputasi : 8 detik. Pemilihan parameter serta nilai *treshold* yang tepat diperlukan untuk mendapatkan hasil evaluasi terbaik pada setiap model klasifikasi yang digunakan pada penelitian ini.

BAB VI
KESIMPULAN DAN SARAN

6.1 Pendahuluan

Bab ini akan membahas perihal kesimpulan dan saran berdasarkan uraian yang telah dibahas pada bab sebelumnya. Kesimpulan serta saran yang diberikan pada bab ini diharapkan dapat menjadi acuan bagi penliti untuk melanjutkan penelitian ini.

6.2 Kesimpulan

Berdasarkan hasil uraian yang telah didapatkan maka penulis dapat menarik beberapa kesimpulan sebagai berikut:

1. Proses pada system pengklasifikasian pertanyaan berbahasa Indonesia menggunakan algoritma Support Vector Machine (SVM) dan metode seleksi fitur *Information Gain*, *Chi Square* dan *Mutual Information* berhasil diimplementasikan.
2. Penggunaan metode seleksi fitur *Information Gain* yang dikombinasikan dengan algoritma klasifikasi SVM pada kernel Rbf dengan parameter berupa nilai C :10 dan treshold : 1.67 berhasil mendapatkan hasil kinerja terbaik dengan nilai *accuracy* : 0.9, *precision* : 0.91, *recall* : 0.86, *f-measure* : 0.88, waktu komputasi : 14 s dan jumlah fitur : 511.

3. Penggunaan metode seleksi fitur *Chi Square* yang dikombinasikan dengan algoritma klasifikasi SVM pada kernel *linear* dengan parameter berupa nilai C :1 dan treshold : 3.5 berhasil mendapatkan hasil kinerja terbaik dengan nilai *accuracy* : 0.92, *precision* : 0.93, *recall* : 0.89, *f-measure* : 0.91,waktu komputasi : 8 s dan jumlah fitur : 432.
4. Penggunaan metode seleksi fitur *Mutual Information* yang dikombinasikan dengan algoritma klasifikasi SVM pada kernel *linear* dengan parameter berupa nilai C :1 dan treshold : 3.5 berhasil mendapatkan hasil kinerja terbaik dengan nilai *accuracy* : 0.92, *precision* : 0.93, *recall* : 0.89, *f-measure* : 0.9,waktu komputasi : 7 s dan jumlah fitur : 240.

6.2 Saran

Adapun saran yang dapat digunakan pada penelitian selanjutnya, yaitu:

1. Melakukan penambahan data pada dataset penelitian serta membuat jumlah fitur yang ada pada data penelitian menjadi seimbang untuk tiap label agar mencegah terjadi *imbalance data*.
2. Melakukan kombinasi pada penelitian menggunakan metode *resampling* untuk mengatasi terjadinya *imbalance data*.
3. Melakukan perbandingan terhadap metode seleksi fitur dengan tipe lain seperti metode seleksi fitur bertipe *wrapper* dan *embedded selector*.

Pebandingan Metode Seleksi Fitur untuk Klasifikasi Pertanyaan Berbahasa Indonesia Menggunakan Algoritma Support Vector Machine (SVM)

ORIGINALITY REPORT

15%	3%	2%	15%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Sriwijaya University Student Paper	13%
2	www.yonkerspublicschools.org Internet Source	1%
3	repository.ub.ac.id Internet Source	1%

Exclude quotes	On	Exclude matches	< 1%
Exclude bibliography	On		

SURAT KETERANGAN PENGECEKAN SIMILARITY

Saya yang bertanda tangan di bawah ini

Nama : Syechky Al Qodrin Aruda

Nim : 09021381823120

Prodi : Teknik Informatika Bilingual

Fakultas : Ilmu Komputer

Menyatakan bahwa benar hasil pengecekan similarity Skripsi~~/Tesis/Disertasi/Lap~~. Penelitian yang berjudul "Perbandingan Metode Seleksi Fitur Untuk Klasifikasi Pertanyaan Berbahasa Indonesia Menggunakan Algoritma *Support Vector Machine* (SVM)" adalah 15%. Dicek oleh operator *:

1. Dosen Pembimbing
2. UPT Perpustakaan
3. Operatur Fakultas Ilmu Komputer

Demikianlah surat keterangan ini saya buat dengan sebenarnya dan dapat saya pertanggung jawabkan.

Indralaya, 03 Juli 2022

Menyetujui

Dosen pembimbing I,

Novi Yusliani, M.T.
NIP.198211082012122001

Dosen Pembibing II,

Alvi Syahrini Utami, M.Kom
NIP.11978122220060420003

Yang menyatakan,

Syechky Al Qodrin Aruda
09021381823120