

**OPTIMASI METODE NAÏVE BAYES MENGGUNAKAN SELEKSI  
FITUR MUTUAL INFORMATION UNTUK KLASIFIKASI TEKS  
UJARAN KEBENCIAN**

**TUGAS AKHIR**

Sebagai syarat untuk memperoleh gelar sarjana S-1 di Program Studi Informatika, Jurusan  
Informatika, Fakultas Teknik Industri, Universitas Pembangunan Nasional “Veteran”  
Yogyakarta



Disusun oleh :

**PUTRA BAGASPATI ROHADI  
123180140**

**PROGRAM STUDI INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS TEKNIK INDUSTRI  
UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN”  
YOGYAKARTA  
2023**

**OPTIMASI METODE NAÏVE BAYES MENGGUNAKAN SELEKSI  
FITUR MUTUAL INFORMATION UNTUK KLASIFIKASI TEKS  
UJARAN KEBENCIAN**



Disusun oleh :

**PUTRA BAGASPATI ROHADI  
123180140**

**PROGRAM STUDI INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS TEKNIK INDUSTRI  
UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN"  
YOGYAKARTA  
2023**

**HALAMAN PENGESAHAN PEMBIMBING**


**OPTIMASI METODE NAÏVE BAYES MENGGUNAKAN SELEKSI  
FITUR MUTUAL INFORMATION UNTUK KLASIFIKASI TEKS  
UJARAN KEBENCIAN**


Disusun oleh :  
Putra Bagaspati Rohadi  
123180140

Telah diperiksa dan disetujui oleh pembimbing untuk disahkan  
pada tanggal : .....


Menyetujui,  
Pembimbing I

Pembimbing II

  
Mangaras Yanu F., S.T., M.T.  
NIDN. 0521018201

  
Juwairrah, S.Si., M.T.  
NIDN. 0527077601

Mengetahui,  
Koordinator Program Studi

  
Dr. Heriyanto, A.Md, S.Kom, M.Cs.  
NIDN. 0508067703



HALAMAN PENGESAHAN PENGUJI

OPTIMASI METODE NAÏVE BAYES MENGGUNAKAN SELEKSI  
FITUR MUTUAL INFORMATION UNTUK KLASIFIKASI TEKS  
UJARAN KEBENCIAN

Disusun oleh :

Putra Bagaspati Rohadi

123180140

Telah diuji dan dinyatakan lulus pada tanggal : .....

Menyetujui,  
Penguji I

Penguji II

Mangaras Yanu F., S.T., M.T.

NIDN. 0521018201

Juwairiah, S.Si., M.T.

NIDN. 0527077601

Penguji III

Penguji IV

Yuli Fauziah, S.T., M.T.

NIDN. 0508077102

Ahmad Taufiq Akbar, S.Si., M.Cs.

NIDN. 0518118701

## SURAT PERNYATAAN KARYA ASLI TUGAS AKHIR

Sebagai mahasiswa Program Studi Informatika, Fakultas Teknik Industri, Universitas Pembangunan Nasional “Veteran” Yogyakarta, yang bertanda tangan dibawah ini, saya:

Nama : Putra Bagaspati Rohadi  
NIM : 123180140  
Fakultas / Prodi : Teknik Industri / Informatika

Dengan ini saya menyatakan bahwa karya ilmiah saya yang berjudul :  
**OPTIMASI METODE NAÏVE BAYES MENGGUNAKAN SELEKSI FITUR MUTUAL INFORMATION UNTUK KLASIFIKASI TEKS UJARAN KEBENCIAN**  
Merupakan karya asli saya dan belum pernah dipublikasikan dimanapun. Apabila dikemudian hari, karya saya disinyalir bukan merupakan karya asli saya, maka saya bersedia menerima konsekuensi apapun yang diberikan Program Studi Informatika, Fakultas Teknik Industri, Universitas Pembangunan Nasional “Veteran” Yogyakarta kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Yogyakarta  
Pada Tanggal : 18 Agustus 2023

Yang Menyatakan



Putra Bagaspati Rohadi  
NIM. 123180140



## PERNYATAAN BEBAS PLAGIASI

Saya yang bertanda tangan di bawah ini :

Nama : Putra Bagaspati Rohadi  
NIM : 123180140  
Fakultas / Prodi : Teknik Industri / Informatika

Dengan ini saya menyatakan bahwa judul Tugas Akhir :

**OPTIMASI METODE NAÏVE BAYES MENGGUNAKAN SELEKSI FITUR  
MUTUAL INFORMATION UNTUK KLASIFIKASI TEKS UJARAN KEBENCIAN**

Adalah hasil kerja saya sendiri dan benar-benar bebas dari plagiasi kecuali cuplikan serta ringkasan yang terdapat di dalamnya telah saya jelaskan sumbernya (sitasi) dengan jelas. Apabila pernyataan ini terbukti tidak benar maka saya akan bersedia menerima sanksi sesuai peraturan Mendiknas RI No. 17 Tahun 2010 dan Peraturan Perundang-undangan yang berlaku

Demikian surat pernyataan ini saya dengan penuh tanggung jawab.

Yogyakarta, 18 Agustus 2023

Yang Membuat Pernyataan



Putra Bagaspati Rohadi

NIM. 123180140

## ABSTRAK

Ujaran kebencian merupakan sebuah bentuk ekspresi yang memuat kebencian terhadap kelompok atau individu yang menjadi sasaran untuk menghina atau mempermalukan. Ujaran kebencian dapat memberikan banyak dampak buruk bagi masyarakat seperti saling membenci, saling curiga, intoleran, diskriminatif, hingga dapat memunculkan kekerasan terhadap individu atau kelompok tertentu, sehingga dibutuhkan sebuah solusi untuk mendeteksi ujaran kebencian secara otomatis. Permasalahan tersebut dapat diatasi dengan melakukan klasifikasi teks menggunakan *machine learning*. Salah satu metode yang dapat digunakan dalam melakukan klasifikasi teks adalah *Naïve Bayes*. Namun metode tersebut memiliki kelemahan yang mengasumsikan setiap fitur bersifat independen satu dengan yang lain.

Kelemahan tersebut dapat diatasi dengan melakukan seleksi fitur untuk memilih fitur-fitur yang paling penting dalam melakukan klasifikasi. Penelitian ini menggunakan seleksi fitur *Mutual Information* yang merupakan salah satu metode paling efektif dalam memilih fitur-fitur yang paling relevan dalam menentukan suatu kelas pada klasifikasi teks. Sehingga metode ini dianggap mampu mengatasi kelemahan dari metode *Naïve Bayes*.

Pada penelitian ini dilakukan pengujian menggunakan *Confusion Matrix* untuk mengetahui nilai akurasi tertinggi yang didapatkan dari klasifikasi teks ujaran kebencian menggunakan metode *Naïve Bayes* dengan seleksi fitur *Mutual Information* maupun tanpa seleksi fitur serta split dataset 70:30, 80:20, dan 90:10. Berdasarkan pengujian yang telah dilakukan, skenario split dataset 90:10 dengan penggunaan seleksi fitur *Mutual Information* mampu meningkatkan akurasi *Naïve Bayes* dalam melakukan klasifikasi teks ujaran kebencian sebesar 1,4% dari 83% menjadi 84,4%. Hal ini menunjukkan bahwa seleksi fitur *Mutual Information* memiliki kemampuan untuk mengatasi kelemahan dari metode *Naïve Bayes* yang mengasumsikan setiap fitur bersifat independen.

**Kata Kunci :** ujaran kebencian, klasifikasi teks, seleksi fitur, *Naïve Bayes*, *Mutual Information*

## ABSTRACT

*Hate speech is a form of expression that contains hatred against a group or individual who is the target of humiliating or humiliating. Hate speech can have many negative impacts on society such as mutual hatred, mutual suspicion, intolerance, discrimination, so that it can lead to violence against certain individuals or groups, so a solution is needed to detect hate speech automatically. These problems can be overcome by classifying text using machine learning. One method that can be used to classify text is Naïve Bayes. However, this method has the disadvantage that it assumes that each feature is independent from one another.*

*These weaknesses can be overcome by performing feature selection to select the most important features in classifying. This study uses Mutual Information feature selection which is one of the most effective methods in selecting the most relevant features in determining a class in text classification. So this method is considered capable of overcoming the weaknesses of the Naïve Bayes method.*

*In this study, testing was carried out using the Confusion Matrix to determine the highest accuracy value obtained from the classification of hate speech texts using the Naïve Bayes method with Mutual Information feature selection or without feature selection and split datasets 70:30, 80:20, and 90:10. Based on the tests that have been carried out, the 90:10 split dataset scenario using the Mutual Information feature selection can increase the accuracy of Naïve Bayes in classifying hate speech texts by 1.4% from 83% to 84.4%. This shows that Mutual Information feature selection has the ability to overcome the weaknesses of the Naïve Bayes method which assumes each feature is independent.*

***Keywords : hate speech, text classification, feature selection, Naïve Bayes, Mutual Information***



## KATA PENGANTAR

Puji syukur atas kehadiran Allah SWT yang telah melimpahkan rahmat serta karunia-Nya sehingga penulis dapat menyelesaikan penelitian Tugas Akhir yang berjudul “Optimasi Metode *Naïve Bayes* Menggunakan Seleksi Fitur *Mutual Information* Untuk Klasifikasi Teks Ujaran Kebencian”. Tugas Akhir ini ditulis sebagai salah satu syarat dalam memperoleh gelar sarjana Program Studi Informatika, Fakultas Teknik Industri, Universitas Pembangunan Nasional “Veteran” Yogyakarta. Selama proses pengerjaan tugas akhir ini, penulis menyadari bahwa mendapatkan banyak bantuan dari berbagai pihak sehingga pada kesempatan ini penulis mengucapkan terimakasih kepada:

1. Allah SWT, yang senantiasa memberikan rahmat, petunjuk, kemudahan, dan kekuatan selama mengerjakan Tugas Akhir.
2. Ayah, Ibu, dan Kakak Saya yang selalu memberikan semangat, dukungan, motivasi, dan kepercayaan dalam menyelesaikan Tugas Akhir penulis.
3. Bapak Mangaras Yanu F., S.T., M.T., selaku dosen pembimbing I yang telah meluangkan waktu, pikiran, dan tenaga dalam membimbing serta memberikan saran kepada penulis.
4. Ibu Juwairiah, S.Si., M.T., selaku dosen pembimbing II yang telah meluangkan waktu, pikiran, dan tenaga dalam membimbing serta memberikan saran kepada penulis.
5. Ibu Yuli Fauziah, S.T., M.T., dan Bapak Ahmad Taufiq Akbar, S.Si., M.Cs., selaku dosen penguji yang telah memberikan saran, kritik, dan bimbingan kepada penulis.
6. Seluruh staff pengajar Jurusan Informatika, Fakultas Teknik Industri, Universitas Pembangunan Nasional “Veteran” Yogyakarta yang telah memberikan ilmu pengetahuan selama proses perkuliahan berlangsung.
7. Tahyaku Tania Titirinai selaku kekasih saya yang terus memberikan motivasi, semangat, dan dukungan dengan tulus untuk berjuang menyelesaikan Tugas Akhir.
8. Seluruh kawan-kawan Sahabat Gabut dan teman dekat penulis yang tidak bisa saya sebutkan satu persatu atas bantuan ilmu, buah pikiran, motivasi, dan semangat yang telah diberikan selama pengerjaan Tugas Akhir.
9. Seluruh kawan-kawan Mabar *Badminton* yang membantu penulis untuk menyalurkan hobi dan menjaga kesehatan selama mengerjakan Tugas Akhir.
10. Teman-teman mahasiswa Informatika 2018 yang telah memberikan dukungan serta bantuannya dalam pelaksanaan Tugas Akhir penulis.

Penulis menyadari bahwa penulisan Tugas Akhir ini masih jauh dari kata sempurna. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun agar penulis dapat lebih baik lagi kedepannya. Akhir kata, semoga Tugas Akhir ini dapat memberikan manfaat bagi penulis dan semua pihak yang membaca.

Yogyakarta, 18 Agustus 2023

Penulis

## DAFTAR ISI

HALAMAN PENGESAHAN PEMBIMBING .....	ii
HALAMAN PENGESAHAN PENGUJI.....	iii
SURAT PERNYATAAN KARYA ASLI TUGAS AKHIR.....	iv
PERNYATAAN BEBAS PLAGIASI.....	v
ABSTRAK .....	vi
ABSTRACT .....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI .....	ix
DAFTAR GAMBAR.....	xii
DAFTAR TABEL .....	xiii
DAFTAR MODUL PROGRAM.....	xiv
BAB I PENDAHULUAN .....	1
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah.....	2
1.3. Batasan Masalah .....	2
1.4. Tujuan Penelitian .....	3
1.5. Manfaat Penelitian .....	3
1.6. Tahapan Penelitian dan Metode Pengembangan Sistem .....	3
1.6.1. Tahapan Penelitian .....	3
1.6.2. Metode Pengembangan Sistem.....	4
1.7. Sistematika Penulisan .....	4
BAB II TINJAUAN LITERATUR .....	6
2.1. Ujaran Kebencian .....	6
2.2. Klasifikasi Teks .....	6
2.3. Text Preprocessing.....	7
2.3.1. Case Folding.....	7
2.3.2. Remove Emoji .....	7
2.3.3. Cleansing .....	7
2.3.4. Remove Repetition Character.....	8
2.3.5. Tokenization .....	8

2.3.6.	Normalisasi Kata Tidak Baku.....	8
2.3.7.	Negasi .....	8
2.3.8.	Stopwords Removal.....	9
2.3.9.	Stemming.....	9
2.4.	Term Frequency-Inverse Document Frequency (TF-IDF) .....	10
2.5.	Mutual Information (MI) .....	11
2.6.	Naïve Bayes (NB).....	12
2.7.	Confusion Matrix .....	13
2.8.	Penelitian Sebelumnya.....	14
<b>BAB III METODOLOGI PENELITIAN .....</b>		<b>17</b>
3.1.	Metodologi Penelitian.....	17
3.1.1.	Pengumpulan Data.....	17
3.1.2.	Data Preprocessing .....	18
3.1.3.	Pembobotan Kata.....	27
3.1.4.	Seleksi Fitur Mutual Information .....	30
3.1.5.	Naïve Bayes .....	32
3.2.	Pengembangan Sistem .....	34
3.2.1.	Analisis Kebutuhan .....	35
3.2.2.	Perancangan Sistem.....	36
3.2.3.	Implementasi .....	40
3.2.4.	Pengujian Sistem .....	40
<b>BAB IV HASIL DAN PEMBAHASAN.....</b>		<b>42</b>
4.1.	Hasil Penelitian .....	42
4.1.1.	Pengumpulan Data.....	42
4.1.2.	Data Preprocessing .....	42
4.1.3.	Pembobotan Kata Menggunakan TF-IDF .....	46
4.1.4.	Seleksi Fitur Menggunakan Mutual Information .....	46
4.1.5.	Pembuatan Model Naïve Bayes.....	48
4.1.6.	Pengujian Confusion Matrix.....	48
4.2.	Implementasi Perancangan Antarmuka .....	53
4.2.1.	Halaman Klasifikasi .....	53
4.2.2.	Halaman Dataset.....	54
4.2.3.	Halaman Overview .....	56

4.3. Pembahasan .....	57
BAB V KESIMPULAN DAN SARAN .....	60
5.1. Kesimpulan .....	60
5.2. Saran .....	60
DAFTAR PUSTAKA.....	61



## DAFTAR GAMBAR

Gambar 3.1 Metodologi penelitian .....	17
Gambar 3.2 Flowchart data preprocessing .....	18
Gambar 3.3 Flowchart case folding .....	19
Gambar 3.4 Flowchart remove emoji .....	20
Gambar 3.5 Flowchart cleansing .....	21
Gambar 3.6 Flowchart remove repetition character .....	22
Gambar 3.7 Flowchart tokenization .....	23
Gambar 3.8 Flowchart normalisasi kata .....	24
Gambar 3.9 Flowchart negasi .....	25
Gambar 3.10 Flowchart stopwords removal .....	26
Gambar 3.11 Flowchart stemming .....	27
Gambar 3.12 Flowchart pembobotan kata TF-IDF .....	28
Gambar 3.13 Flowchart seleksi fitur <i>Mutual Information</i> .....	31
Gambar 3.14 Flowchart <i>Naïve Bayes</i> .....	33
Gambar 3.15 Metode pengembangan sistem .....	34
Gambar 3.16 Arsitektur sistem .....	36
Gambar 3.17 <i>Flowchart</i> sistem .....	37
Gambar 3.18 Flowchart model <i>Naïve Bayes</i> .....	38
Gambar 3.19 Halaman input teks .....	39
Gambar 3.20 Halaman hasil klasifikasi .....	39
Gambar 3.21 Halaman dataset .....	40
Gambar 3.22 Halaman <i>overview</i> .....	40
Gambar 4.1 Implementasi Halaman klasifikasi .....	54
Gambar 4.2 Implementasi halaman hasil klasifikasi .....	54
Gambar 4.3 Implementasi halaman dataset .....	55
Gambar 4.4 Implementasi halaman <i>clean dataset</i> .....	55
Gambar 4.5 Implementasi halaman data <i>training</i> .....	56
Gambar 4.6 Implementasi halaman data <i>testing</i> .....	56
Gambar 4.7 Implementasi halaman <i>overview</i> .....	57
Gambar 4.8 Grafik perbandingan seleksi fitur pada <i>split dataset</i> 70:30 .....	57
Gambar 4.9 Grafik perbandingan seleksi fitur pada <i>split dataset</i> 80:20 .....	58
Gambar 4.10 Grafik perbandingan seleksi fitur pada <i>split dataset</i> 90:10 .....	58
Gambar 4.11 Grafik perbandingan akurasi model .....	59

## DAFTAR TABEL

Tabel 2.1 Contoh penggunaan <i>Case Folding</i> .....	7
Tabel 2.2 Contoh penggunaan Remove Emoji .....	7
Tabel 2.3 Contoh penggunaan Cleansing .....	7
Tabel 2.4 Contoh penggunaan Remove Repetition Character .....	8
Tabel 2.5 Contoh penggunaan Tokenization .....	8
Tabel 2.6 Contoh penggunaan normalisasi kata .....	8
Tabel 2.7 Contoh penggunaan negasi .....	9
Tabel 2.8 Contoh penggunaan <i>Stopwrods Removal</i> .....	9
Tabel 2.9 Kombinasi awalan-akhiran yang tidak diizinkan .....	10
Tabel 2.10 Contoh penggunaan <i>Stemming</i> .....	10
Tabel 2.11 <i>Contingency Table</i> .....	11
Tabel 2.12 <i>Confusion matrix</i> .....	13
Tabel 2.13 <i>State of the art</i> .....	14
Tabel 3.1 Contoh data .....	29
Tabel 3.2 Contoh perhitungan TF .....	29
Tabel 3.3 Contoh hasil TF-IDF .....	30
Tabel 3.5 <i>Contingency Table</i> .....	31
Tabel 3.6 Hasil seleksi fitur <i>Mutual Information</i> .....	32
Tabel 3.7 Spesifikasi perangkat keras .....	35
Tabel 3.8 Spesifikasi perangkat lunak .....	35
Tabel 3.9 Spesifikasi pengguna .....	35
Tabel 3.10 Pengujian <i>Confusion Matrix</i> .....	41
Tabel 3.11 Perbandingan skenario <i>split data training</i> dan data <i>testing</i> .....	41
Tabel 4.1 Hasil nilai <i>threshold</i> dengan akurasi tertinggi .....	47
Tabel 4.2 Pengujian <i>Confusion Matrix</i> pada skenario 1 .....	49
Tabel 4.3 Pengujian <i>Confusion Matrix</i> pada skenario 2 .....	50
Tabel 4.4 Pengujian <i>Confusion Matrix</i> pada skenario 3 .....	50
Tabel 4.5 Pengujian <i>Confusion Matrix</i> pada skenario 4 .....	51
Tabel 4.6 Pengujian <i>Confusion Matrix</i> pada skenario 5 .....	52
Tabel 4.7 Pengujian <i>Confusion Matrix</i> pada skenario 6 .....	52
Tabel 4.8 Hasil perbandingan skenario <i>split data training</i> dan data <i>testing</i> .....	53

## DAFTAR MODUL PROGRAM

Modul Program 4.1 Pengumpulan data .....	42
Modul Program 4.2 <i>Case folding</i> .....	43
Modul Program 4.3 <i>Remove emoji</i> .....	43
Modul Program 4.4 <i>Cleansing</i> .....	43
Modul Program 4.5 <i>Remove repetition character</i> .....	44
Modul Program 4.6 <i>Tokenization</i> .....	44
Modul Program 4.7 Normalisasi kata.....	44
Modul Program 4.8 Negasi.....	45
Modul Program 4.9 <i>Stopwords removal</i> .....	45
Modul Program 4.10 <i>Stemming</i> .....	46
Modul Program 4.11 Pembobotan kata TF-IDF.....	46
Modul Program 4.12 Perhitungan nilai MI .....	46
Modul Program 4.13 Perulangan nilai <i>threshold</i> .....	47
Modul Program 4.14 <i>Export</i> hasil seleksi fitur MI.....	48
Modul Program 4.15 Pembuatan model <i>Naïve Bayes</i> .....	48
Modul Program 4.16 Pengujian model menggunakan <i>Confusion Matrix</i> .....	49

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Klasifikasi teks merupakan sebuah proses untuk mengelompokkan dokumen-dokumen ke dalam suatu kategori atau lebih yang telah didefinisikan sebelumnya (Munitasri et al., 2018). Dalam melakukan sebuah proses klasifikasi terdapat dua tahapan, yaitu tahap pembelajaran dan tahap pengklasifikasian (Sartika & Sensuse, 2017). Proses membuat model klasifikasi merupakan tahapan pembelajaran, sedangkan proses menerapkan model klasifikasi untuk memprediksi kategori atau kelas dari suatu data merupakan tahapan pengklasifikasian (Sartika & Sensuse, 2017). Klasifikasi teks dapat digunakan untuk pengelompokan ujaran kebencian (Antariksa et al., 2019). Ujaran kebencian merupakan bentuk ekspresi kebencian terhadap kelompok yang menjadi sasaran atau dimaksudkan untuk menghina, mempermalukan, atau untuk menghina anggota kelompok (Davidson et al., 2017). Sehingga klasifikasi teks diperlukan dalam melacak ujaran kebencian agar dapat menentukan upaya preventif yang dapat dilakukan (Antariksa et al., 2019).

Pada umumnya terdapat banyak algoritma klasifikasi yang sering digunakan, yaitu *Naïve Bayes*, *Support Vector Machine*, *K-Nearest Neighbor*, *Random Forest*, dan *Decision Tree* (Wibisono & Fahrurrozi, 2019). Penelitian yang dilakukan oleh (Sharma et al., 2018) melakukan klasifikasi teks ujaran kebencian menggunakan tiga metode yaitu, *Support Vector Machine*, *Naïve Bayes*, dan *Random Forest* dimana klasifikasi dengan menggunakan metode *random forest* mendapatkan nilai akurasi tertinggi dibandingkan dengan metode lainnya. Selain itu terdapat metode klasifikasi teks lainnya, seperti *Logistic Regression* (LR) dan *K-Nearest Neighbor* (KNN) (Thangaraj & Sivakami, 2018). *Logistic Regression* memiliki kelebihan tidak membutuhkan terlalu banyak *computational resources* tetapi tidak bisa menyelesaikan masalah *non-linear* dan membutuhkan data sampel yang besar (Kowsari et al., 2019). KNN memiliki kelebihan secara alami menangani kumpulan data multi-kelas tetapi metode ini memiliki kekurangan dalam susahnya mencari *nearest neighbors* dan menentukan nilai *k* yang optimal (Thangaraj & Sivakami, 2018).

*Naïve Bayes* dapat memecahkan masalah *Logistic Regression* dan KNN, yaitu membutuhkan data sampel yang besar dan waktu komputasi yang lama tetapi *Naïve Bayes* memiliki kelemahan yang mengasumsikan bahwa setiap fitur adalah independen satu sama lain (Kowsari et al., 2019)(Thangaraj & Sivakami, 2018). Penggunaan seleksi fitur dapat menjadi solusi untuk mengatasi kelemahan dari metode *Naïve Bayes* untuk mengurangi jumlah fitur pada dataset. Penelitian sebelumnya yang menggabungkan metode *Naïve Bayes* dan seleksi fitur pada klasifikasi pesan SMS dilakukan oleh (Munitasri et al., 2018) yang menggunakan seleksi fitur *Genetic Algorithm* dan dapat meningkatkan akurasi sebesar 0.34%, namun penggunaan seleksi fitur tersebut dapat menghasilkan solusi yang berlebihan dalam menentukan fitur yang relevan.



Penerapan yang dilakukan oleh (Prakoso et al., 2019) dengan menggunakan seleksi fitur *Information Gain* dan penerapan *Naïve Bayes* dengan *Bayesian Boosting* untuk klasifikasi berita mendapatkan hasil bahwa penggunaan seleksi fitur *Information Gain* tidak berpengaruh atas kenaikan akurasi dari metode *Naïve Bayes* hal ini terjadi dikarenakan *Information Gain* hanya mempertimbangkan informasi pada variabel target atau kelas, tanpa mempertimbangkan hubungan antara fitur-fitur yang ada.

Penelitian lainnya juga dilakukan oleh (Irham et al., 2019) dalam melakukan klasifikasi berita yang menggabungkan metode *Support Vector Machine* dengan seleksi fitur *Mutual Information*, dimana dalam penelitian ini seleksi fitur *Mutual Information* terbukti dapat meningkatkan akurasi dari 93.63% menjadi 94.24%. Seleksi fitur *Mutual Information* juga terbukti efektif membuang fitur-fitur yang tidak dibutuhkan pada sebuah kelas sehingga klasifikasi hanya menggunakan fitur yang dibutuhkan dan fokus pada kuantitas informasi yang relevan antara fitur dan variabel target (Irham et al., 2019).

Berdasarkan uraian yang sudah dipaparkan diatas, penerapan seleksi fitur mampu mengurangi jumlah fitur untuk mengatasi kekurangan metode *Naïve Bayes* dan meningkatkan akurasi dari metode *Naïve Bayes*, sehingga pada penelitian ini akan menerapkan seleksi fitur *Mutual Information* pada metode *Naïve Bayes* untuk melakukan klasifikasi teks ujaran kebencian. Dengan penerapan metode tersebut diharapkan dapat menghasilkan sebuah model yang dapat melakukan klasifikasi teks ujaran kebencian ke dalam dua kelas yaitu ujaran kebencian dan bukan ujaran kebencian.

## 1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka permasalahan yang dapat dirumuskan adalah apakah dengan seleksi fitur *Mutual Information* dapat mengatasi kelemahan *Naïve Bayes* yang mengasumsikan setiap fitur bersifat independen untuk klasifikasi teks ujaran kebencian.

## 1.3. Batasan Masalah

Adapun batasan masalah yang ditentukan untuk menghindari perluasan pembahasan dalam penelitian ini sebagai berikut:

1. Data yang digunakan adalah data sekunder dari kaggle *Indonesian Hate Speech Twitter Text*.
2. Dataset berformat .csv.
3. Jumlah dataset yang digunakan yaitu 10.000
4. Data yang digunakan adalah data teks berbahasa Indonesia.
5. Kategori klasifikasi yang dihasilkan ada dua, yaitu ujaran kebencian dan bukan ujaran kebencian.
6. Seleksi fitur yang digunakan adalah *Mutual Information*.
7. Penelitian ini tidak mempertimbangkan simbol *emoji*.

#### 1.4. Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk mengetahui penerapan seleksi fitur *Mutual Information* pada metode *Naïve Bayes* dalam melakukan klasifikasi teks ujaran kebencian serta untuk mengetahui peningkatan akurasinya.

#### 1.5. Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat kepada *user* untuk mengetahui informasi dan pengetahuan mengenai implementasi dari metode *Naïve Bayes* dengan seleksi fitur *Mutual Information* dalam melakukan klasifikasi teks ujaran kebencian yang dapat bermanfaat untuk masyarakat umum sehingga dapat menjadi acuan dalam membuat pendeteksian otomatis pada teks ujaran kebencian.

#### 1.6. Tahapan Penelitian dan Metode Pengembangan Sistem

Tahapan tahapan penelitian dan metode pengembangan sistem yang akan dilakukan pada penelitian ini sebagai berikut:

##### 1.6.1. Tahapan Penelitian

- a. Pengumpulan data  
Pada tahap pengumpulan data, data yang digunakan bersumber dari *website* [www.kaggle.com](http://www.kaggle.com). Pada *website* ini dipilih *dataset* yang berisikan teks ujaran kebencian dan bukan ujaran kebencian dari *Indonesian Hate Speech Twitter Text*.
- b. Data *preprocessing*  
Pada tahapan ini, data yang telah didapatkan akan diolah untuk mengubah data yang tidak terstruktur menjadi data terstruktur dan memiliki format yang baik sehingga dapat digunakan untuk proses klasifikasi teks. Tahapan *preprocessing* yang dilakukan adalah *case folding*, *remove emoji*, *cleansing*, *remove repetition character*, *tokenization*, normalisasi kata, negasi, *stopwords removal*, dan *stemming*.
- c. Pembobotan kata menggunakan TF-IDF  
Pada tahapan ini, akan dilakukan pembobotan kata menggunakan TF-IDF (*Term Frequency-Inverse Document Frequency*). Pembobotan kata digunakan untuk mengetahui nilai frekuensi sebuah kata pada dokumen atau teks.
- d. Seleksi fitur menggunakan *Mutual Information*  
Pada bagian ini, akan dilakukan seleksi fitur menggunakan *Mutual Information* yang dimana merupakan proses pemilihan fitur dari *dataset* dengan mengurangi jumlah fitur yang tidak relevan.
- e. Klasifikasi menggunakan metode *Naïve Bayes*  
Pada bagian ini dilakukan klasifikasi teks dengan metode *Naïve Bayes*. *Naïve Bayes* merupakan sebuah algoritma klasifikasi yang berdasarkan perhitungan probabilitas dari *dataset* yang digunakan.
- f. Pengujian

Pada tahap ini, akan dilakukan pengujian berdasarkan hasil dari model yang telah dibuat dengan menggunakan *Confusion Matrix*.

### 1.6.2. Metode Pengembangan Sistem

Dalam penelitian ini pengembangan sistem menggunakan metode *Waterfall* (Pressman, 2014). Metode *Waterfall* memiliki 5 tahapan yaitu, analisis kebutuhan, perancangan sistem, implementasi, pengujian sistem, dan pemeliharaan sistem. Namun pada penelitian ini tahapan dari metode *Waterfall* hanya dilakukan sampai pengujian sistem, tahapan pemeliharaan sistem tidak dilakukan. Berikut ini merupakan uraian dari tahapan metode *Waterfall* :

#### 1. Analisis Kebutuhan

Analisis kebutuhan meliputi penentuan apa saja yang dibutuhkan untuk pengembangan sistem yang terdiri dari kebutuhan fungsional dan non fungsional seperti kebutuhan *input* data, proses data, dan *output* data.

#### 2. Perancangan Sistem

Pada tahap ini dilakukan perancangan terhadap sistem yang akan dibangun, mulai dari perancangan setiap proses pada sistem, perancangan alur sistem, analisis terhadap algoritma yang digunakan.

#### 3. Implementasi

Tahap implementasi merupakan tahap pembuatan sistem berdasarkan tahap analisis kebutuhan dan alur pada analisis perancangan.

#### 4. Pengujian Sistem

Pada tahap ini dilakukan pengujian terhadap sistem yang telah dibangun untuk mengetahui akurasi yang dihasilkan oleh metode *Naïve Bayes* dengan seleksi fitur *Mutual Information* pada klasifikasi teks ujaran kebencian.

### 1.7. Sistematika Penulisan

Sistematika penulisan disusun untuk membantu pembaca memahami penelitian dan memberikan gambaran umum tentang penelitian yang dikerjakan. Sistematika penulisan pada penelitian ini sebagai berikut :

#### BAB I PENDAHULUAN

Pada bab ini membahas latar belakang masalah, merumuskan masalah, menentukan batasan masalah, tujuan dari penelitian, manfaat penelitian, tahapan penelitian serta sistematika penulisan.

#### BAB II TINJAUAN LITERATUR

Pada bab ini berisi kajian dan konsep dasar yang berkaitan dengan objek penelitian, masalah penelitian dan metode penelitian dari pustaka ilmiah atau penelitian sebelumnya sebagai referensi yang berkaitan dan bersifat fundamental terhadap masalah yang akan diselesaikan.

#### BAB III METODOLOGI PENELITIAN

Pada bab ini membahas tentang pemodelan sistem, analisis data, analisis sistem dan perancangan sistem klasifikasi teks.

#### **BAB IV HASIL DAN PEMBAHASAN**

Pada bab ini membahas tentang implementasi sistem dan pengujian sistem untuk mengetahui hasil dari implementasi tersebut.

#### **BAB V KESIMPULAN DAN SARAN**

Pada bab ini berisi kesimpulan dari penelitian yang telah dilakukan dan saran yang dapat digunakan sebagai referensi untuk penelitian selanjutnya.



## **BAB II**

### **TINJAUAN LITERATUR**

#### **2.1. Ujaran Kebencian**

Ujaran kebencian merupakan sebuah ujaran yang memuat kebencian, menyerang, dan berkobar-kobar yang memiliki tujuan untuk memberikan orang lain inspirasi untuk menyakiti atau melakukan kekerasan terhadap orang atau kelompok tertentu (Widayati, 2018). Ujaran kebencian juga dimaksudkan untuk menghina dan mempermalukan sebuah individu atau kelompok yang dalam penyampaiannya ujaran kebencian tersebut disertai dengan bahasa kasar (Davidson et al., 2017). Ujaran kebencian dapat memberikan banyak dampak buruk bagi masyarakat seperti saling membenci, saling curiga, intoleran, diskriminatif hingga dapat memunculkan kekerasan terhadap individu atau kelompok tertentu yang lebih lemah (Sya'bani et al., 2021).

#### **2.2. Klasifikasi Teks**

Klasifikasi teks merupakan sebuah proses untuk menempatkan kalimat, teks, atau dokumen ke dalam suatu kategori yang sesuai dengan isinya (Munitasri et al., 2018). Klasifikasi memiliki tujuan untuk memperkirakan kategori dari suatu dokumen yang belum diketahui kategorinya (VijayGaikwad et al., 2014). Dalam melakukan sebuah proses klasifikasi terdapat dua tahapan, yaitu tahap pembelajaran dan tahap pengklasifikasian. Proses membuat model klasifikasi merupakan tahapan pembelajaran, sedangkan proses menerapkan model klasifikasi untuk memprediksi kategori atau kelas dari suatu data merupakan tahapan pengklasifikasian (Sartika & Sensuse, 2017). Klasifikasi merupakan salah satu teknik pembelajaran supervised learning, dimana hasil dari proses klasifikasi berupa nilai tertentu yang dihasilkan dari pembelajaran data yang digunakan (Zulfikar & Lukman, 2016). Pada umumnya terdapat banyak algoritma klasifikasi yang sering digunakan, yaitu *Naïve Bayes*, *Support Vector Machine*, *K-Nearest Neighbor*, *Random Forest*, dan *Decision Tree* (Wibisono & Fahrurozi, 2019).

Metode tersebut juga dapat menerapkan seleksi fitur, seperti penelitan yang dilakukan oleh (Munitasri et al., 2018) yang menggabungkan metode *Naïve Bayes* dengan seleksi fitur *Genetic Algorithm* dalam melakukan klasifikasi SMS yang menghasilkan peningkatan akurasi sebesar 0.34%, penelitian lain juga dilakukan oleh (Prakoso et al., 2019) yang menggunakan metode *Naïve Bayes* dan seleksi fitur *Information Gain* untuk melakukan klasifikasi berita, dimana hasil penelitian tersebut mendapatkan tidak adanya peningkatan akurasi setelah penggunaan seleksi fitur. Penelitian yang dilakukan (Ulfa et al., 2018) yang menggunakan metode *Naïve Bayes* dan seleksi fitur *Mutual Information* dalam melakukan analisis sentimen mendapatkan ada peningkatan akurasi dan pengurangan waktu komputasi yang dibutuhkan, yaitu masing-masing secara berurutan sebesar 1.7% dan 51.52%.

## 2.3. Text Preprocessing

*Text preprocessing* merupakan suatu proses untuk membersihkan dan menyiapkan sebuah dokumen teks sebelum dokumen tersebut digunakan (Hegazi et al., 2021). Tahapan *preprocessing* yang akan dilakukan pada penelitian ini adalah *case folding*, *remove emoji*, *cleansing*, *remove repetition character*, *tokenization*, normalisasi kata, negasi, *stopwords removal*, dan *stemming*.

### 2.3.1. Case Folding

*Case folding* merupakan suatu proses untuk mengubah seluruh huruf atau karakter yang kapital pada dokumen menjadi huruf kecil atau *lowercase* (Maylawati et al., 2019). Huruf kapital yang diubah adalah huruf yang terdapat didalam alfabet. *Case folding* dibutuhkan pada penelitian ini karena pada umumnya dokumen yang berisi teks tidak konsisten terhadap penggunaan huruf kapital. Berikut ini merupakan contoh penggunaan *case folding* :

**Tabel 2.1** Contoh penggunaan *Case Folding*

Sebelum	Sesudah
Gila gw seneng banget kemarin Adek gw juara 1 lomba matematikaa, kerenn banget!!	gila gw seneng banget kemarin adek gw juara 1 lomba matematikaa, kerenn banget!!

### 2.3.2. Remove Emoji

Proses menghapus emoji adalah langkah untuk menghilangkan simbol-simbol emoji dari suatu data (Alam and Yao, 2019). Emoji seringkali digunakan oleh pengguna untuk mengungkapkan perasaan mereka, seperti kegembiraan, kebahagiaan, kemarahan, dan kesedihan. Namun, dalam beberapa situasi, pengguna juga menggunakan emoji secara sarkastik, yang dimana merupakan bentuk penggunaan bahasa yang rumit dan kompleks (Khan et al. 2017). Berikut ini merupakan contoh penggunaan *remove emoji*:

**Tabel 2.2** Contoh penggunaan *Remove Emoji*

Sebelum	Sesudah
gila gw seneng banget kemarin adek gw juara 1 lomba matematikaa, kerenn banget!! ☺ ☺	gila gw seneng banget kemarin adek gw juara 1 lomba matematikaa, kerenn banget!!

### 2.3.3. Cleansing

*Cleansing* merupakan proses untuk membersihkan teks dari karakter yang tidak diperlukan seperti URL, simbol, angka, emoji, spasi yang berlebihan dan tanda baca yang berupa titik, koma, tanda seru, tanda tanya, tanda kutip, dan lain sebagainya (Septian et al., 2019). Proses pembersihan tersebut digunakan untuk mengurangi *noise* pada *dataset* yang dapat mengganggu proses klasifikasi. Berikut ini merupakan contoh penggunaan *cleasing* :

**Tabel 2.3** Contoh penggunaan *Cleansing*

Sebelum	Sesudah
gila gw seneng banget kemarin adek gw juara 1 lomba matematikaa, kerenn banget!!	gila gw seneng banget kemarin adek gw juara lomba matematikaa kerenn banget

### 2.3.4. *Remove Repetition Character*

*Remove repetition character* merupakan sebuah proses untuk menghapus huruf atau karakter dalam suatu kata yang berulang-ulang yang bertujuan untuk memudahkan dalam menemukan kata sesungguhnya (Anugerah & Djunaidy, 2017). Berikut ini merupakan contoh penggunaan *remove repetition character* :

**Tabel 2.4** Contoh penggunaan Remove Repetition Character

Sebelum	Sesudah
gila gw seneng bangeeet kemarin adek gw juara lomba matematikaa keren banget	gila gw seneng banget kemarin adek gw juara lomba matematika keren banget

### 2.3.5. *Tokenization*

*Tokenization* merupakan proses mengubah teks menjadi bagian yang lebih kecil atau dari teks menjadi kalimat dan dari kalimat menjadi kata (Rahman & Hossen, 2019). Kata atau token yang telah didapatkan akan digunakan dalam menghitung bobot kata.

**Tabel 2.5** Contoh penggunaan Tokenization

Sebelum	Sesudah
mobil sangat bagus	mobil, sangat, bagus

### 2.3.6. *Normalisasi Kata Tidak Baku*

Proses normalisasi kata tidak baku merupakan sebuah proses yang bertujuan untuk mengubah kata-kata yang tidak baku menjadi kata baku yang sesuai dengan Kamus Besar Bahasa Indonesia (KBBI) (Buntoro, et al., 2014). Terdapat berbagai jenis perbaikan kata yang tidak baku, seperti perbaikan penulisan dengan bahasa slang atau modern (lu = kamu), perbaikan penulisan dengan penyingkatan kata (blm = belum), perbaikan penulisan dengan kata yang salah eja (aps = apa) , dan lain sebagainya.

Pada penelitian ini, normalisasi kata tidak baku yang digunakan adalah perbaikan penulisan dengan bahasa slang atau modern, penyingkatan kata, dan kata yang salah eja. Berikut ini merupakna contoh penggunaan normalisasi kata tidak baku :

**Tabel 2.6** Contoh penggunaan normalisasi kata

Sebelum	Sesudah
gila gw seneng banget kemarin adek gw juara lomba matematika keren banget	gila saya senang banget kemarin adek saya juara lomba matematika keren banget

### 2.3.7. *Negasi*

Negasi adalah sebuah bentuk penyangkalan atau ingkaran dari sebuah kata atau pernyataan tertentu (Tarecha et al., 2022). Pada umumnya dalam Bahasa Indonesia terdapat empat kata negasi yang sering digunakan yaitu tidak, bukan, jangan, dan belum (Syafar, 2016). Sebagai contoh, jika terdapat teks “tidak pintar” maka jika tidak dilakukan proses pengubahan kata negasi akan mengakibatkan teks tersebut dianggap sebagai dua kata yang berbeda, dimana ini akan mempengaruhi makna dari teks tersebut. Sehingga perlu dilakukan sebuah proses untuk mengubah teks tersebut menjadi satu kata tanpa mengubah makna teks tersebut. Masalah tersebut dapat diatas dengan mengubah satu kata yang terdapat setelah

kata negasi menjadi lawan kata atau antonim, misalnya teks “tidak pintar” diubah menjadi kata “bodoh”.

**Tabel 2.7** Contoh penggunaan negasi

Sebelum	Sesudah
kemarin aku beli baju dan bahannya tidak jelek	kemarin aku beli baju dan bahannya bagus

### 2.3.8. *Stopwords Removal*

*Stopword* merupakan kata yang sering muncul dalam kumpulan teks atau dokumen dan tidak memiliki makna atau informasi yang bisa digunakan dalam proses klasifikasi (Kurnia, 2020). Sehingga *stopwords removal* merupakan sebuah proses untuk menghilangkan kata yang tidak bermakna. Kata-kata yang merupakan *stopword* adalah ini, itu, dan, atau, ada, lagi, dengan, mungkin, dan lain-lain.

**Tabel 2.8** Contoh penggunaan *Stopwrods Removal*

Sebelum	Sesudah
aku beli baju dan bahannya bagus	aku beli baju bahannya bagus

### 2.3.9. *Stemming*

*Stemming* merupakan sebuah proses mengubah kata menjadi kata dasar dengan menghilangkan imbuhan. Sebagai contoh, kata memakan, dimakan, makanan, akan diubah ke kata dasarnya yaitu makan. Proses *stemming* dilakukan karena dapat meningkatkan efisiensi klasifikasi dalam memilah dokumen (Nugroho, 2018).

Secara umum terdapat dua algoritma yang sering digunakan dalam proses *stemming* yaitu algoritma Porter dan Nazief & Adriani, dimana algoritma Nazief & Adriani memiliki presentase keakuratan lebih tinggi dibandingkan dengan algoritma Porter (Agusta, 2009). Pada algoritma Nazief & Adriani terdapat beberapa tahapan, yaitu :

1. Periksa kata tersebut dalam kamus. Jika ditemukan, maka diasumsikan bahwa kata tersebut merupakan kata dasar, maka proses berhenti.
2. Membuang *Inflection Suffixes* (“-lah”, “-kah”, “-ku”, “-mu”, atau “-nya”). Jika kata merupakan *particles* (“-lah”, “-kah”, “-tah” atau “-pun”), maka tahap ini diulangi lagi dengan membuang *Possesive Pronouns* (“-ku”, “-mu”, atau “-nya”).
3. Membuang *Derivation Suffixes* (“-i”, “-an” atau “-kan”). Jika kata ditemukan di kamus, maka proses berhenti. Jika tidak maka lanjut ke tahap 3a.
  - a. Jika “-an” telah dihapus dan huruf terakhir dari kata tersebut adalah “-k”, maka “-k” juga ikut di hapus. Jika kata tersebut ditemukan dalam kamus maka proses berhenti. Jika tidak maka lanjut ke tahap 3b.
  - b. Akhiran yang dihapus (“-i”, “-an” atau “-kan”) dikembalikan dan lanjut ke tahap ke 4.
4. Membuang *Derivation Prefix*. Jika pada tahap 3 ada sufiks yang dihapus maka lanjut ke tahap 4a, jika tidak maka lanjut ke tahap 4b.
  - a. Periksa tabel kombinasi awalan – akhiran yang tidak diizinkan. Jika ditemukan maka proses berhenti, jika tidak maka lanjut ke tahap 4b.

**Tabel 2.9** Kombinasi awalan-akhiran yang tidak diizinkan

Awalan	Akhiran yang tidak diizinkan
be-	-i
di-	-an
ke-	-i, -kan
me-	-an
se-	-i, -kan
te-	-an

- b. Menentukan tipe awalan kemudian hapus awalan. Jika kata dasar belum juga ditemukan, maka lanjut ke tahap 5. Jika sudah, maka proses berhenti jika awalan kedua sama dengan awalan pertama algoritma.
5. Melakukan *Recoding* (penyusunan kembali kata-kata yang mengalami proses *Stemming* berlebih).
6. Jika semua langkah telah selesai tetapi tidak juga berhasil, maka kata awal diasumsikan sebagai kata dasar.

Berikut contoh kalimat sebelum dan sesudah proses *stemming* :

**Tabel 2.10** Contoh penggunaan *Stemming*

Sebelum	Sesudah
mobilnya sangatlah bagus	mobil sangat bagus

#### 2.4. Term Frequency-Inverse Document Frequency (TF-IDF)

*Term Frequency-Inverse Document Frequency* (TF-IDF) adalah sebuah metode untuk mengetahui seberapa relevan sebuah kata atau *term* di dalam dokumen dengan cara menentukan nilai frekuensi sebuah kata di dalam sebuah dokumen serta frekuensi sebuah kata di dalam banyak dokumen (Hendy Evan & Sigit Purnomo, 2014). TF-IDF bertujuan untuk memberikan bobot pada setiap kata dengan melakukan dua perhitungan, yaitu *Term Frequency* (TF) yang menghitung frekuensi kemunculan sebuah *term* pada sebuah dokumen dan *Document Frequency* (DF) dengan menghitung frekuensi kemunculan sebuah *term* pada banyak atau seluruh dokumen. Rumus dari pembobotan kata menggunakan TF-IDF (Septian et al., 2019) dapat dilihat sebagai berikut :

$$W_{t,d} = tf_{t,d} \times idf_t = tf_{t,d} \times \log \frac{N}{df_t} \dots\dots\dots(2.1)$$

Keterangan :

$W_{t,d}$  : Bobot TF-IDF

$tf_{t,d}$  : Jumlah frekuensi kata

$idf_t$  : Jumlah *inverse* frekuensi dokumen tiap kata

$df_t$  : Jumlah frekuensi dokumen tiap kata

N : Jumlah total dokumen

**2.5. Mutual Information (MI)**

*Mutual Information* merupakan sebuah metode untuk menghitung seberapa banyak informasi yang terkandung dalam *term*, dan kontribusinya dalam menentukan suatu kelas yang tepat pada klasifikasi (Nurfikri et al., 2018). Seleksi fitur menggunakan *Mutual Information* bertujuan untuk mengurai jumlah fitur yang banyak menjadi lebih pendek dan efektif (Zareapoor & K. R, 2015). Sebagai contoh *term* “bodoh” merupakan *term* dengan nilai MI tertinggi untuk kelas ujaran kebencian, maka jika terdapat sebuah dokumen yang memiliki *term* tersebut, besar kemungkinan untuk dokumen tersebut masuk ke dalam kelas ujaran kebencian. Rumus perhitungan nilai MI (Ulfa et al., 2018) dapat dilihat pada persamaan (2.2).

$$I(T, C) = \sum_{ec \in C} \sum_{et \in T} P(et, ec) \log_2 \frac{P(et, ec)}{P(et)P(ec)} \dots\dots\dots(2.2)$$

Dimana variabel T merupakan *term* acak dengan nilai et = 1 (dokumen mengandung *term* T) dan et = 0 (dokumen tidak mengandung *term* T), sedangkan C merupakan kelas acak dengan nilai ec = 1 (dokumen berada dikelas C) dan ec = 0 (dokumen tidak berada dikelas C). Untuk lebih mudah memahami persamaan (2.2) dapat dilakukan dengan membuat tabel *contingency* yang pada dilihat pada tabel 2.11.

**Tabel 2.11 Contingency Table**

	ec = 1	ec = 0
et = 1	BB	BS
et = 0	SB	SS

Keterangan:

BB = jumlah dokumen dari term yang dicari dan terdapat pada kelas yang dipilih

SB = jumlah dokumen dari term yang tidak dicari dan terdapat pada kelas yang dipilih

BS = jumlah dokumen dari term yang dicari dan tidak terdapat pada kelas yang dipilih

SS = jumlah dokumen dari term yang tidak dicari dan tidak terdapat pada kelas yang dipilih

Sehingga dari persamaan 2.2 dan tabel 2.11 dapat dijabarkan menjadi seperti persamaan 2.3.

$$I(T, C) = \frac{BB}{N} \log_2 \frac{N \cdot BB}{N_{1*} N_{*1}} + \frac{SB}{N} \log_2 \frac{N \cdot SB}{N_{0*} N_{*1}} + \frac{BS}{N} \log_2 \frac{N \cdot BS}{N_{1*} N_{*0}} + \frac{SS}{N} \log_2 \frac{N \cdot SS}{N_{0*} N_{*0}} \dots\dots\dots(2.3)$$

Keterangan:

N = Jumlah dokumen yang memiliki *et* dan *ec* atau (N = BB + SB + BS + SS).

N<sub>1\*</sub> = Jumlah dokumen yang memiliki *et* = 1 atau (N<sub>1\*</sub> = BB + BS).

N<sub>\*1</sub> = Jumlah dokumen yang memiliki *ec* = 1 atau (N<sub>\*1</sub> = BB + SB).

N<sub>0\*</sub> = Jumlah dokumen yang tidak memiliki *et* = 0 atau (N<sub>0\*</sub> = SB + SS).

$N_{*0}$  = Jumlah dokumen yang tidak memiliki  $ec = 0$  atau ( $N_{*0} = BS + SS$ ).

## 2.6. Naïve Bayes (NB)

*Naïve Bayes* merupakan sebuah metode atau algoritma untuk melakukan klasifikasi berdasarkan probabilitas dengan menggunakan *Teorema Bayes* (Rini et al., 2016). *Naïve Bayes* melakukan prediksi peluang terjadinya kejadian di masa depan berdasarkan data atau pengalaman yang ada sebelumnya. *Naïve Bayes* juga memiliki banyak keunggulan yaitu mudah untuk diimplementasikan, sederhana, tidak membutuhkan sumber daya komputasi yang besar, tidak memerlukan waktu yang banyak (Birjali et al., 2021), dan sangat baik untuk digunakan dalam analisis data teks (Rahat et al., 2020).

Bentuk umum atau rumus *Naïve Bayes* (Irsad et al., 2019) adalah sebagai berikut :

$$P(c_j|w_i) = \frac{P(c_j) P(w_i|c_j)}{P(w_i)} \dots\dots\dots(2.4)$$

Keterangan :

$P(c_j|w_i)$  : Peluang kategori  $j$  ketika terdapat kemunculan kata  $I$  atau *posterior*

$P(w_i|c_j)$  : Peluang sebuah kata  $i$  masuk ke dalam kategori  $j$  atau *conditional probability*

$P(c_j)$  : Peluang kemunculan sebuah kategori  $j$  atau *prior*

$P(w_i)$  : Peluang kemunculan sebuah kata atau *evidence*

Pada persamaan 2.4 peluang kemunculan sebuah kata atau *evidence* tidak memiliki pengaruh pada perbandingan hasil klasifikasi setiap kategori sehingga peluang tersebut dapat dihilangkan dan bentuk persamaan dapat dilihat sebagai berikut :

$$P(c_j|w_i) = P(c_j) P(w_i|c_j) \dots\dots\dots(2.5)$$

Untuk mencari nilai *prior* pada persamaan 2.5 dapat menggunakan persamaan sebagai berikut :

$$P(c_j) = \frac{N_c}{N} \dots\dots\dots(2.6)$$

Keterangan :

$N_c$  : Banyak dokumen kategori  $c_j$  pada data latih

$N$  : Jumlah keseluruhan data latih yang digunakan

Sedangkan untuk mencari nilai *posterior* dapat dilakukan dengan mengalikan nilai *prior* dengan total *conditional probability* dan didapatkan persamaan sebagai berikut :

$$P(c_j|w_i) = P(c_j) \times P(w_i|c_j) \times \dots \times P(w_n|c_j) \dots\dots\dots(2.7)$$

## 2.7. Confusion Matrix

*Confusion Matrix* merupakan tabel yang menyatakan klasifikasi jumlah data uji yang benar dan jumlah data uji yang salah (Ali, 2017). *Confusion Matrix* dapat digunakan sebagai metode pengujian untuk mengetahui sejauh mana performa dari kinerja sistem yang dibangun. Berikut merupakan tabel dari *Confusion Matrix* (Andono et al., 2017):

**Tabel 2.12** *Confusion matrix*

		Prediksi	
		Positive	Negative
Aktual	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Keterangan:

*True Positive* (TP) : Dokumen diprediksi bernilai positif dan diklasifikasikan sebagai kelas positif (true)

*True Negative* (TN) : Dokumen diprediksi bernilai negatif dan diklasifikasikan sebagai kelas negatif (true)

*False Positive* (FP) : Dokumen diprediksi bernilai positif namun diklasifikasikan sebagai kelas negatif (false)

*False Negative* (FN) : Dokumen diprediksi bernilai negatif namun diklasifikasikan sebagai kelas positif (false)

Dari tabel *Confusion Matrix* diatas, dapat menghitung beberapa ukuran evaluasi pada sebuah sistem klasifikasi yaitu akurasi, *precision* dan *recall* dengan rumus sebagai berikut (Ali, 2017) :

a) Akurasi

Akurasi merupakan nilai untuk mengukur sejauh mana model klasifikasi mampu membuat prediksi yang benar secara keseluruhan. Perhitungan akurasi dapat dilihat pada persamaan 2.8 berikut ini :

$$\text{Akurasi} = \frac{TP + TN}{TP + TN + FP + FN} \dots\dots\dots(2.8)$$

b) *Recall*

*Recall* digunakan untuk mengetahui sejauh mana model klasifikasi mampu mendeteksi secara benar semua data positif yang ada ataupun data negatif. Perhitungan *recall* dapat dilihat pada persamaan 2.9 berikut ini :

$$\text{Recall Positif} = \frac{TP}{TP + FN} \dots\dots\dots(2.9)$$

$$\text{Recall Negatif} = \frac{TN}{TN + FP} \dots\dots\dots(2.10)$$

$$\text{Recall Rata-Rata} = \frac{\text{Recall Positif} + \text{Recall Negatif}}{2} \dots\dots\dots(2.11)$$



c) *Precision*

*Precision* digunakan untuk mengukur seberapa banyak prediksi positif yang bernilai benar atau seberapa banyak prediksi negatif yang bernilai benar. Perhitungan *precision* dapat dilihat pada persamaan 2.10 berikut ini :

$$Precision \text{ Positif} = \frac{TP}{TP + FP} \dots\dots\dots(2.12)$$

$$Precision \text{ Negatif} = \frac{TN}{TN + FN} \dots\dots\dots(2.13)$$

$$Precision \text{ Rata-Rata} = \frac{Precision \text{ Positif} + Precision \text{ Negatif}}{2} \dots\dots\dots(2.14)$$

## 2.8. Penelitian Sebelumnya

Penelitian sebelumnya dapat dilihat pada tabel dibawah ini :

**Tabel 2.13** *State of the art*

No	Penulis	Metode	Dataset	Hasil
1	(Munitasri et al., 2018)	<i>Naïve bayes</i> dengan Seleksi Fitur <i>Genetic Algorithm</i>	dataset sms dari UCI machine learning <a href="http://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection">http://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection</a> berjumlah 5572 data	Penggunaan seleksi fitur <i>Genetic Algorithm</i> mampu meningkatkan hasil akurasi sebesar 0.34% dari 89.39% ke 89.73%
2	(Prakoso et al., 2019)	<i>Naive Bayes Classifier</i> dengan <i>Bayesian Boosting</i> dan seleksi fitur <i>Information Gain</i>	Situs website detik.com	Penggunaan Bayesian boosting dan seleksi fitur <i>Information Gain</i> tidak berpengaruh terhadap kenaikan hasil akurasi dari NB
3	(Ali Fauzi et al., 2017)	<i>Naïve Bayes</i> dengan seleksi fitur <i>Information Gain</i> dan <i>Maximal Marginal Relevance for Feature Selection</i> (MMR-FS)	Situs website kompas.com berjumlah 500 data	Penggunaan seleksi fitur <i>Information Gain</i> dan <i>Maximal Marginal Relevance for Feature Selection</i> mendapatkan akurasi yaitu 80% dengan jumlah fitur 200
4	(Hermanto et al., 2020)	<i>Naive Bayes</i> dan <i>Support Vector Machine</i>	Email dari mahasiswa pada sisfo akademik mahasiswa (students.bsi.ac.id) berjumlah 5954 data	Metode SVM mendapatkan akurasi sebesar 84.45% dan metode NBC mendapatkan akurasi sebesar 69.75%

5	(Devita et al., 2018)	<i>Naive Bayes</i> dan <i>K-Nearest Neighbor</i> .	Data yang digunakan adalah abstrak dari artikel jurnal berbahasa Indonesia yang ada di Universitas Negeri Malang berjumlah 40 data	Dengan jumlah dataset 40 didapatkan akurasi dari metode NB sebesar 70% dan metode KNN sebesar 40%
6	(Wati, 2020)	Metode <i>Naive Bayes</i> dan <i>Particel Swarm Optimization</i> (PSO)	Berita dari website <a href="http://turnbackhoax.id">turnbackhoax.id</a> berjumlah 150 data	PSO dapat meningkatkan nilai akurasi pada metode NBC sebesar 10.52% dari 74.67% menjadi 85.19%
7	(Irham et al., 2019)	<i>Support Vector Machine</i> dengan seleksi fitur <i>Mutual Information</i>	Tidak disebutkan sumber dataset, namun menggunakan 360 data teks berita Bahasa Indonesia	Seleksi fitur <i>Mutual Information</i> dapat meningkatkan akurasi dari 93.63% menjadi 94.24%.
8	(Santoso et al., 2018)	<i>Self-Training Naive Bayes</i> Berbasis <i>Word2Vec</i>	Tidak disebutkan sumber dataset, namun menggunakan 29.587 dokumen	Metode <i>Self-Training Naive Bayes</i> menghasilkan nilai akurasi yang lebih rendah jika dibandingkan dengan penelitian sebelumnya dengan menggunakan metode <i>Multinomial Naive Bayes</i>
9	<b>Penelitian yang akan dilakukan</b>	<i>Naive Bayes</i> dengan seleksi fitur <i>Mutual Information</i>	<b>Situs website <a href="http://www.kaggle.com">www.kaggle.com</a></b>	<b>Mengetahui peningkatan akurasi dari metode <i>naive bayes</i> dengan penerapan seleksi fitur <i>mutual information</i></b>

Berdasarkan penelitian sebelumnya, penelitian yang akan dilakukan memiliki perbedaan pada metode, objek dan jumlah *dataset* yang digunakan. Berikut beberapa perbandingan pada penelitian sebelumnya dan penelitian yang akan dilakukan :

1. Penelitian yang dilakukan oleh (Munitasri et al., 2018) menggunakan metode *Naive Bayes* dengan seleksi fitur *Genetic Algorithm* dan (Prakoso et al., 2019) menggunakan metode *Naive Bayes Classifier* dengan *Bayesian Boosting* dan seleksi fitur *Information Gain*, sedangkan pada penelitian ini menggunakan metode *Naive Bayes* dengan seleksi fitur *Mutual Information*.
2. Pada penelitian (Munitasri et al., 2018), (Prakoso et al., 2019), (Ali Fauzi et al., 2017), (Hermanto et al., 2020), dan (Devita et al., 2018) memiliki perbedaan pada objek yang akan dilakukan klasifikasi, yaitu sms, email, berita, dan jurnal. Penelitian ini akan melakukan klasifikasi teks pada ujaran kebencian.

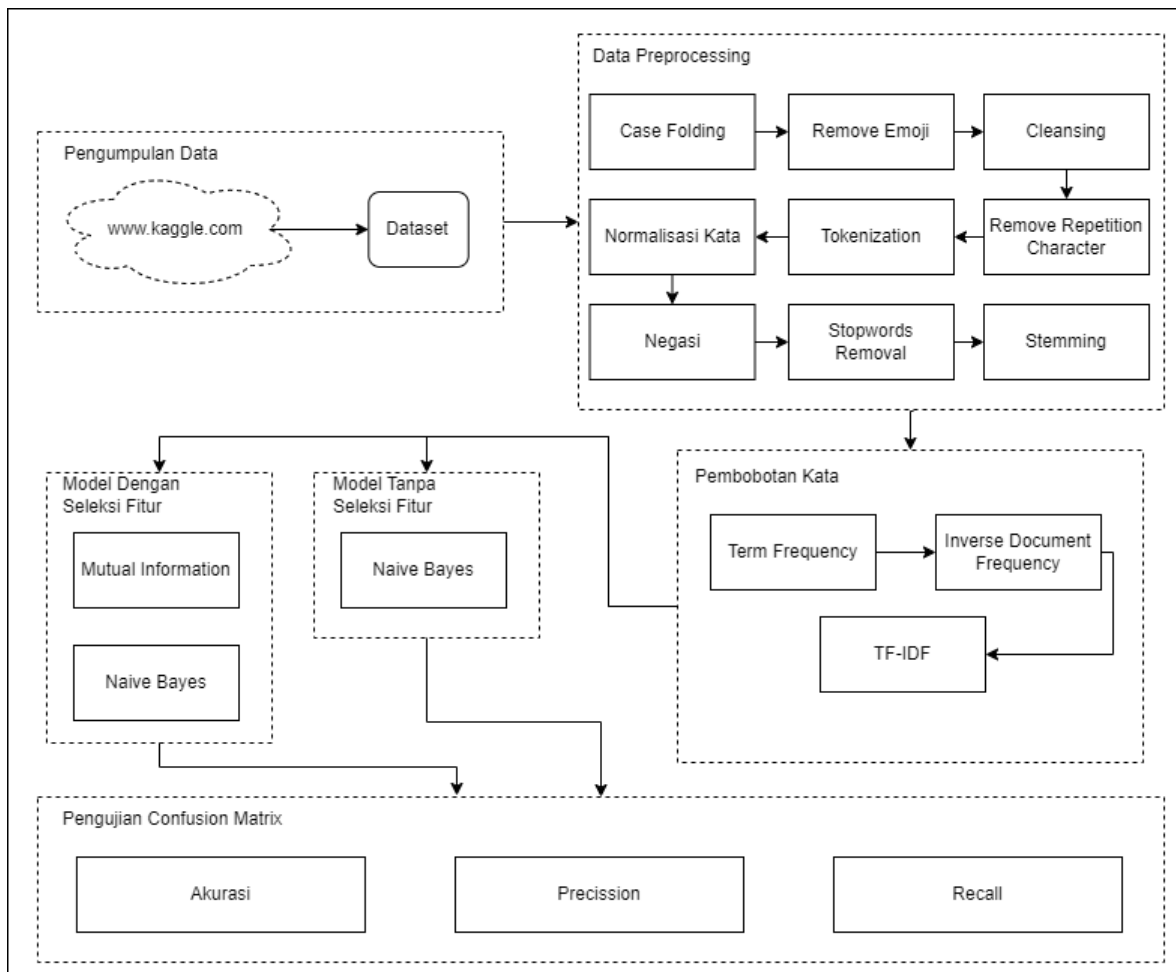
3. Penelitian ini menggunakan *dataset* sebanyak 10.000 lebih banyak dari penelitian yang dilakukan oleh (Hermanto et al., 2020) menggunakan *dataset* sebanyak 8699, namun lebih sedikit dari penelitian yang dilakukan oleh (Santoso et al., 2018) menggunakan *dataset* sebanyak 29.587.

# BAB III

## METODOLOGI PENELITIAN

### 3.1. Metodologi Penelitian

Metode penelitian yang digunakan dalam penelitian ini adalah metode kuantitatif yang diterapkan pada model matematis dari algoritma *Naïve Bayes* dengan seleksi fitur *Mutual Information*. Adapun tahapan metodologi penelitian yang digunakan terdiri dari 6 tahap seperti pada Gambar 3.1.



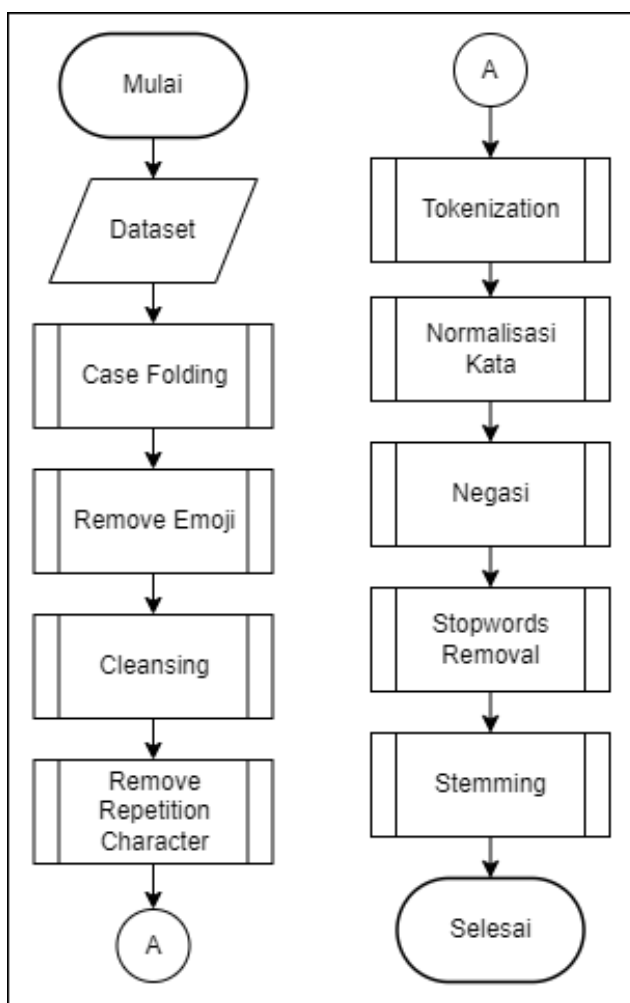
Gambar 3.1 Metodologi penelitian

#### 3.1.1. Pengumpulan Data

Pengumpulan data merupakan sebuah proses untuk mendapatkan data yang akan digunakan dalam penelitian ini. Pada penelitian ini akan menggunakan *dataset* sekunder yang didapatkan dari *website kaggle* yaitu *dataset* ujaran kebencian dalam bahasa Indonesia pada media sosial *Twitter* yang dipublikasikan oleh Muhammad Okky Ibrohim and Indra Budi pada tahun 2019 (Ibrohim & Budi, 2019). *Dataset* tersebut berjumlah 10.000 data teks yang terbagi kedalam dua kelas yaitu 5.000 data ujaran kebencian dan 5.000 data bukan ujaran kebencian.

### 3.1.2. Data Preprocessing

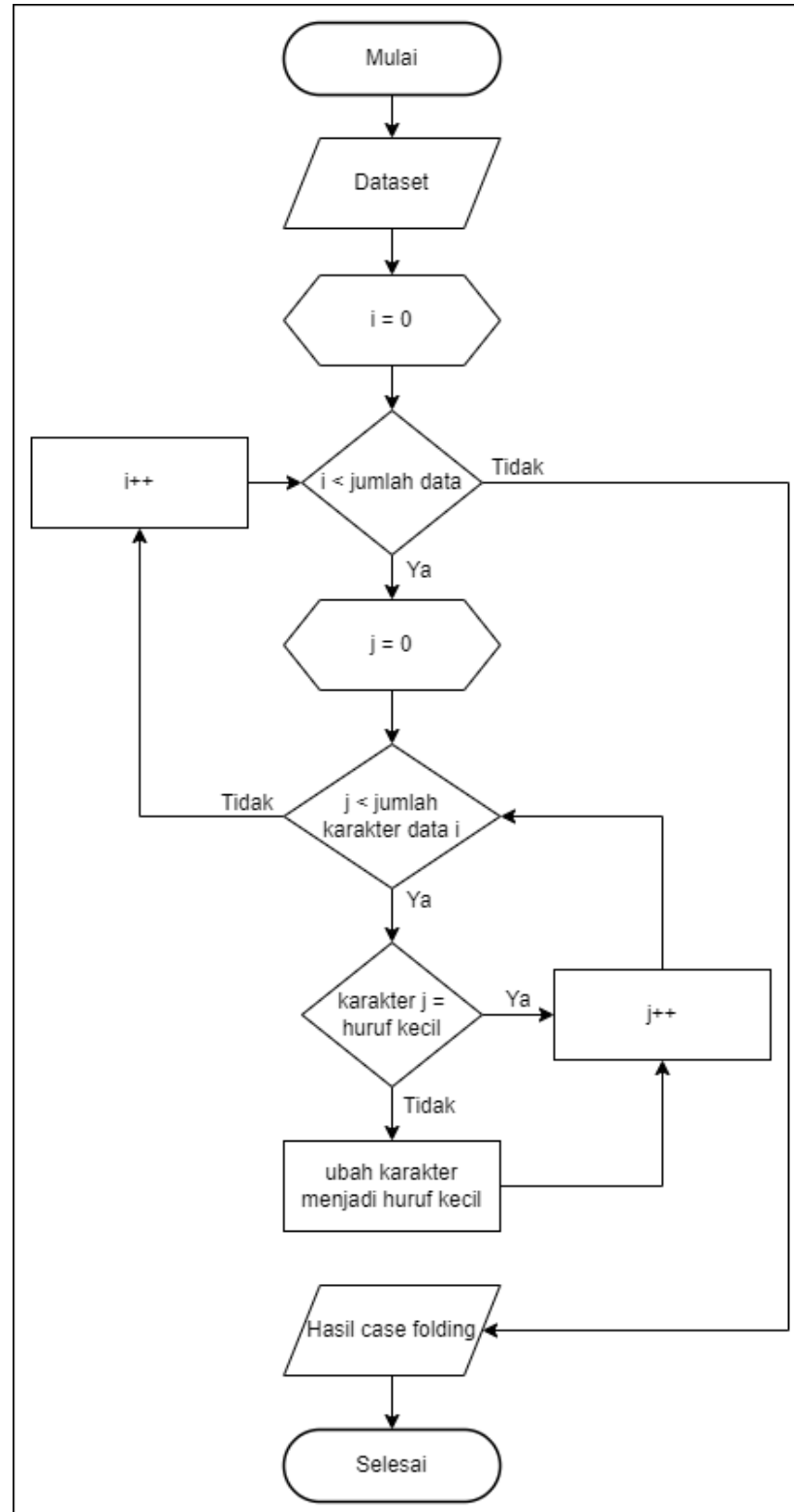
Tahap data *preprocessing* dilakukan dengan tujuan untuk membersihkan dan mempersiapkan *dataset* yang akan digunakan dalam proses analisis. Pada tahap data *preprocessing* terdiri dari beberapa proses yaitu *case folding*, *remove emoji*, *cleansing*, *remove repetition character*, *tokenization*, normalisasi kata, negasi, *stopwords removal*, dan *stemming*. Proses data *preprocessing* dapat dilihat pada Gambar 3.2.



**Gambar 3.2** Flowchart data preprocessing

**a. Case Folding**

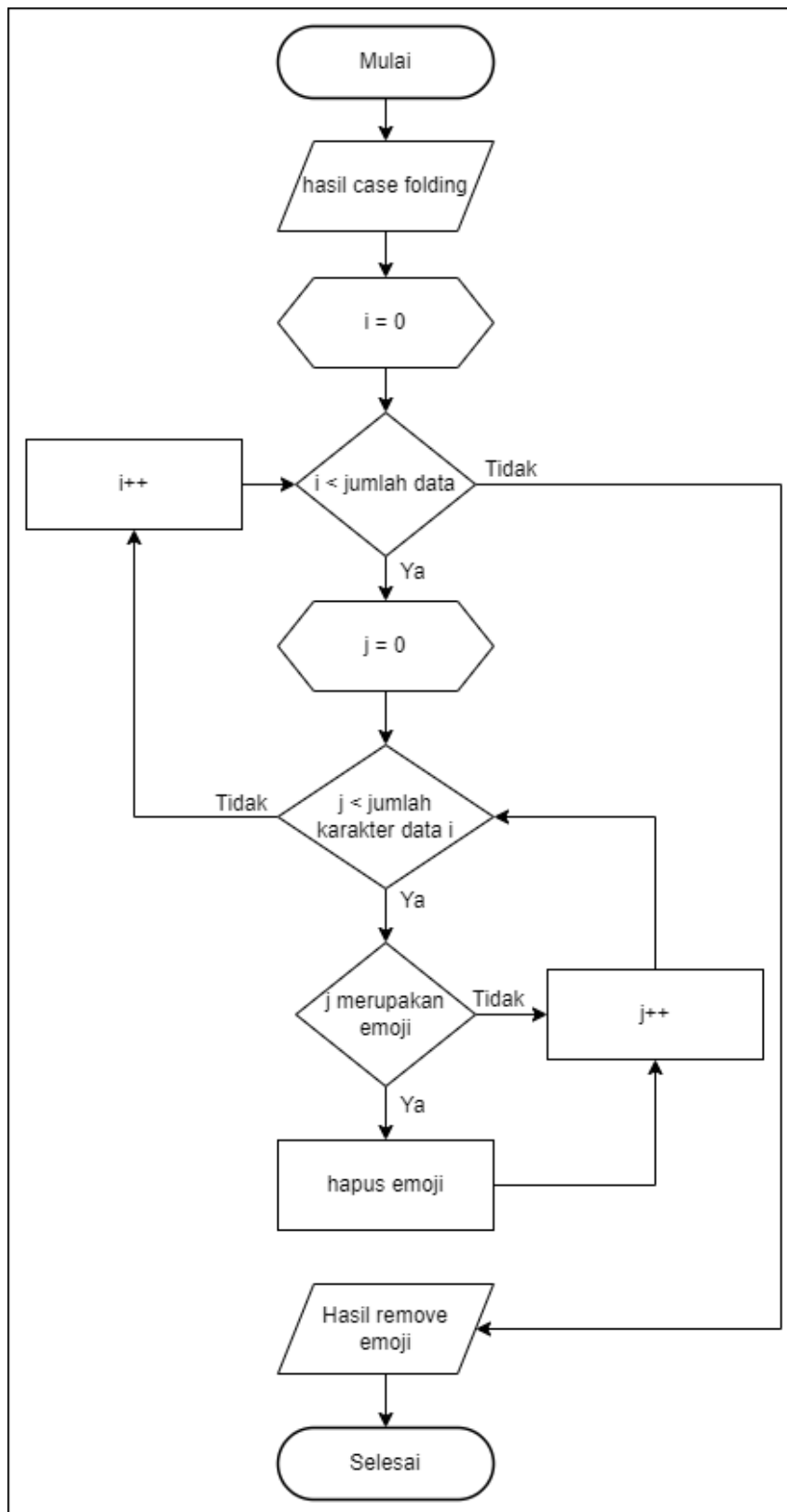
*Case folding* merupakan proses untuk mengubah semua huruf atau teks dalam dokumen menjadi huruf kecil atau *lowercase*. Berikut *flowchart* dari proses *case folding* dapat dilihat pada Gambar 3.3.



**Gambar 3.3** *Flowchart case folding*

### b. Remove Emoji

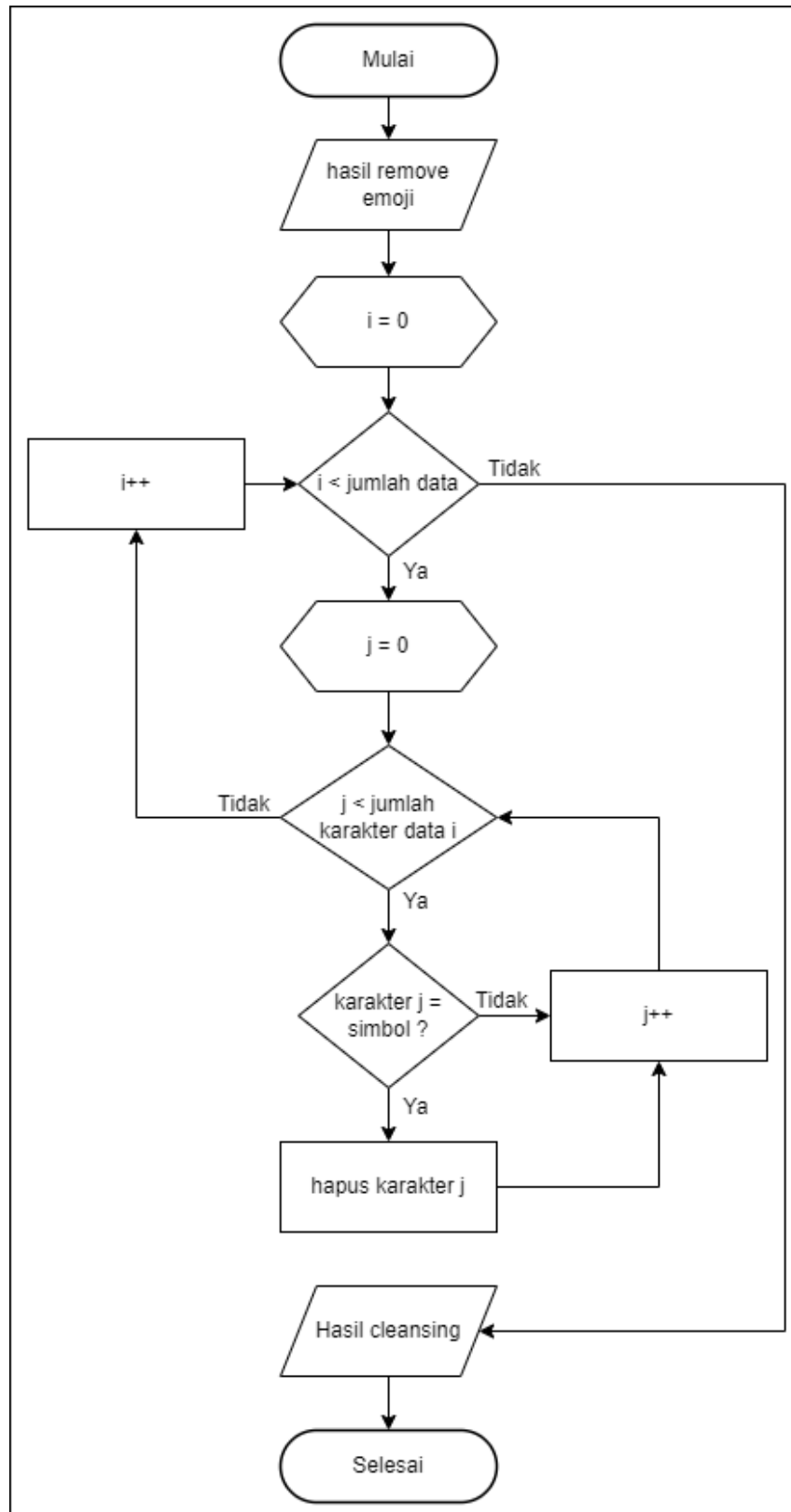
*Remove Emoji* merupakan proses untuk menghilangkan emoji pada *dataset* yang digunakan. Berikut *flowchart* dari proses *remove emoji* dapat dilihat pada Gambar 3.4



**Gambar 3.4** *Flowchart remove emoji*

### c. *Cleansing*

*Cleansing* merupakan proses untuk membersihkan data dari angka, tanda baca, url, hastag, dan lain-lain. Berikut *flowchart* dari proses *cleansing* dapat dilihat pada Gambar 3.5.

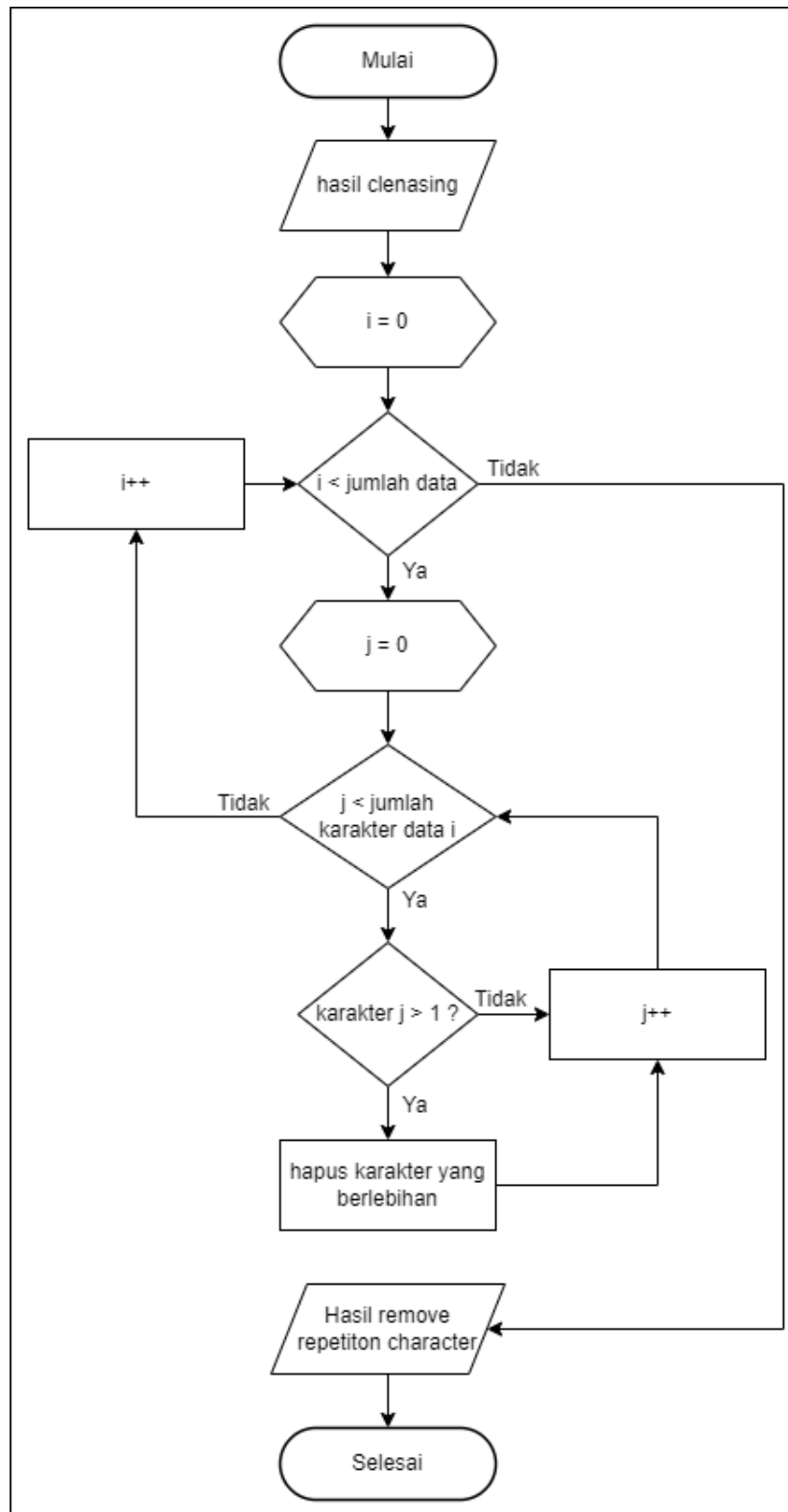


**Gambar 3.5** *Flowchart cleansing*



#### d. *Remove Repetition Character*

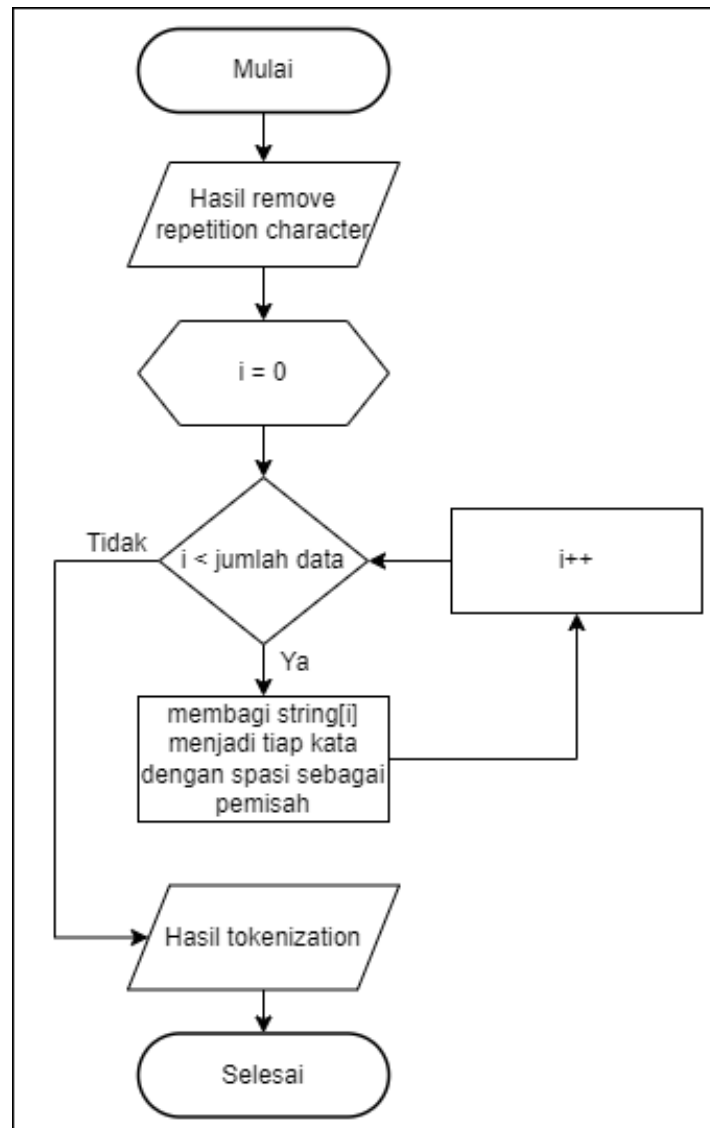
*Remove Repetition Character* merupakan proses untuk membersihkan data dari karakter yang berulang. Berikut *flowchart* dari proses *remove repetition character* dapat dilihat pada Gambar 3.6.



**Gambar 3.6** *Flowchart remove repetition character*

### e. Tokenization

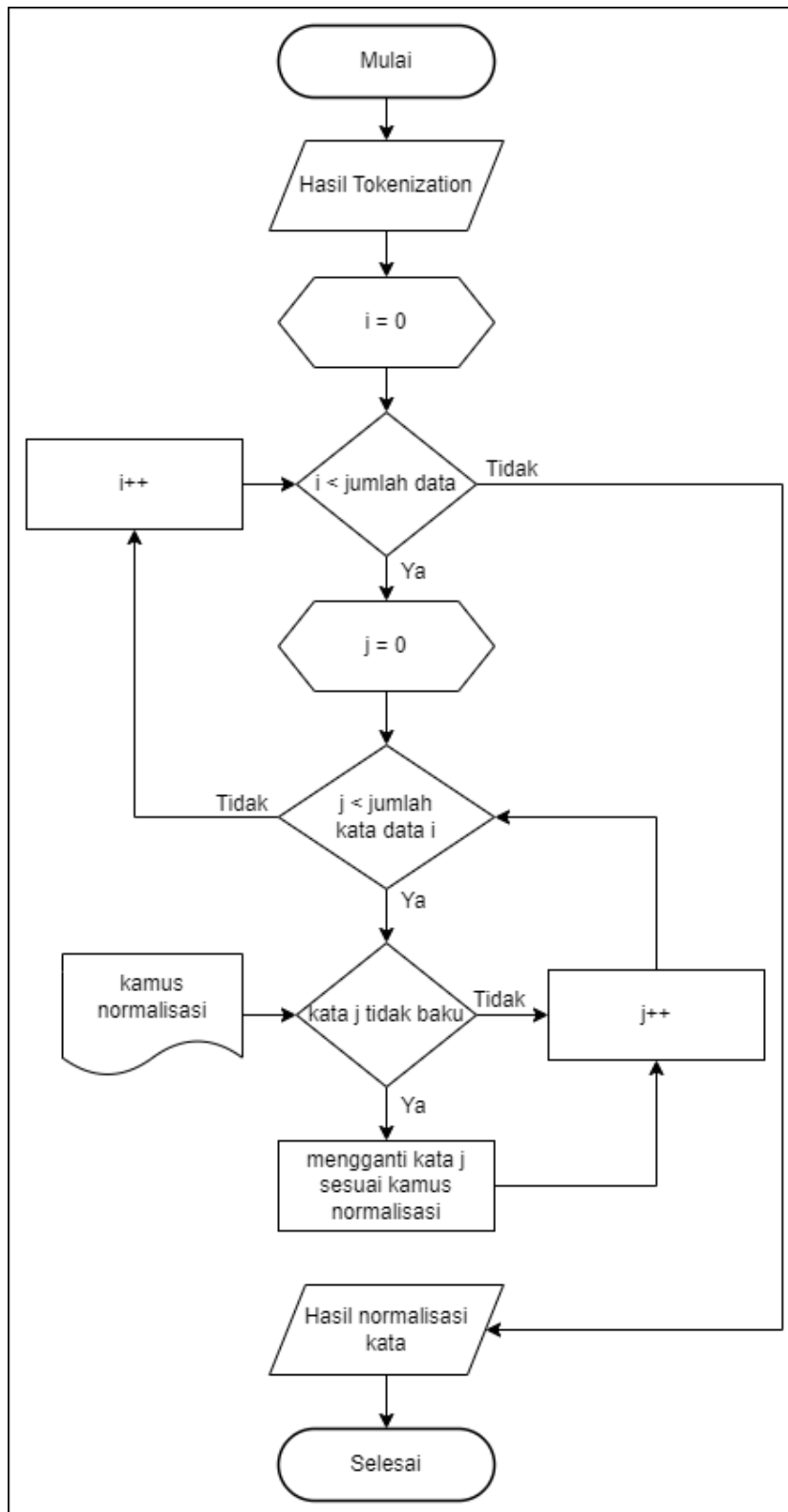
*Tokenization* merupakan proses untuk mengubah data yang berbentuk kalimat menjadi kumpulan tiap kata atau *token*. Berikut *flowchart* dari proses *tokenization* dapat dilihat pada Gambar 3.7.



**Gambar 3.7** *Flowchart tokenization*

### f. Normalisasi Kata

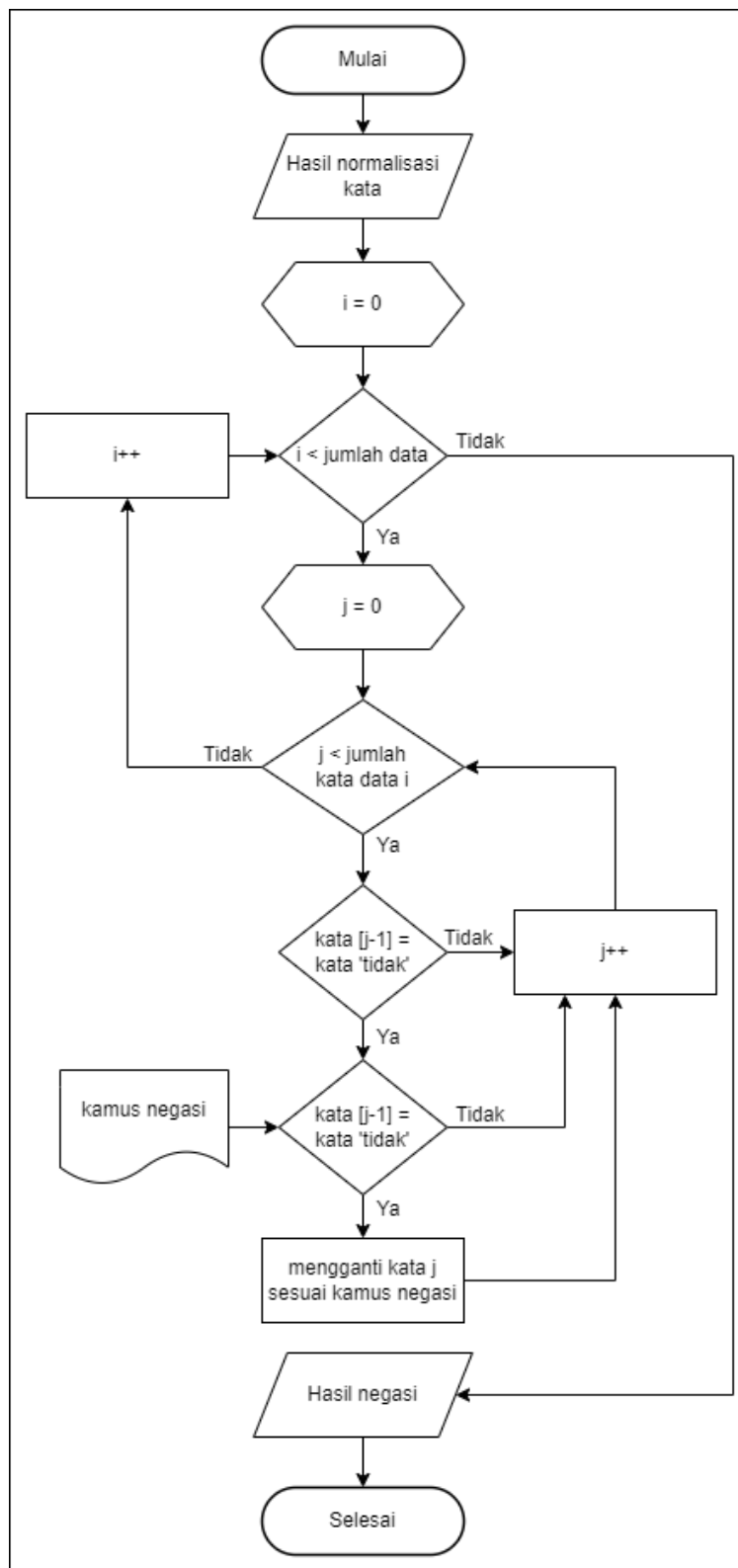
Normalisasi kata merupakan proses untuk mengubah kata yang tidak baku atau bahasa slang, penyingkatan kata, dan kata yang salah eja menjadi kata Bahasa Indonesia yang baku. Berikut *flowchart* dari proses normalisasi kata dapat dilihat pada Gambar 3.8.



**Gambar 3.8** Flowchart normalisasi kata

### g. Negasi

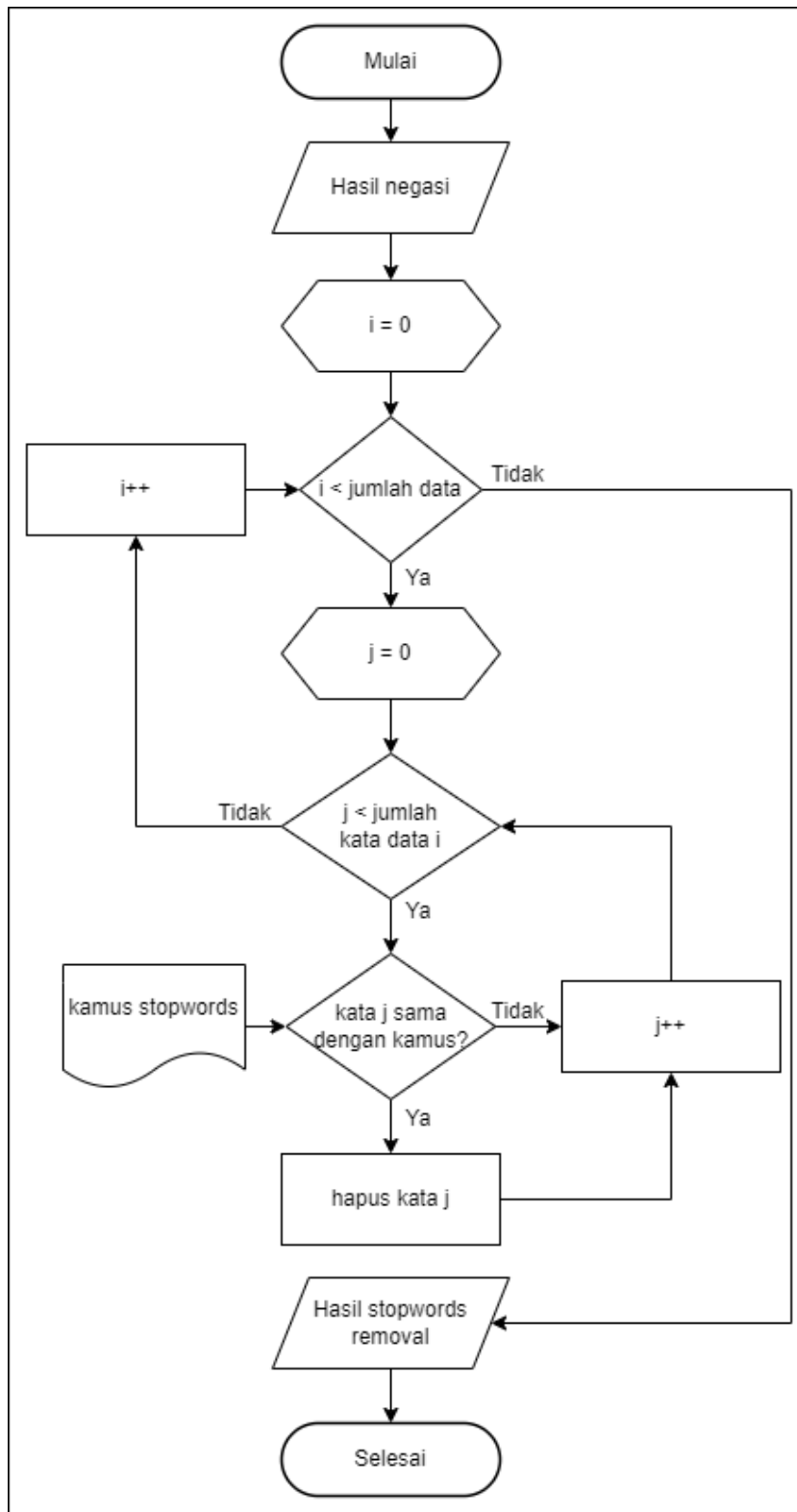
Negasi merupakan proses untuk mengubah semua kata setelah kata “tidak” menjadi kata yang sesuai dengan kamus negasi. Berikut *flowchart* dari proses negasi dapat dilihat pada Gambar 3.9.



Gambar 3.9 Flowchart negasi

### h. Stopwords Removal

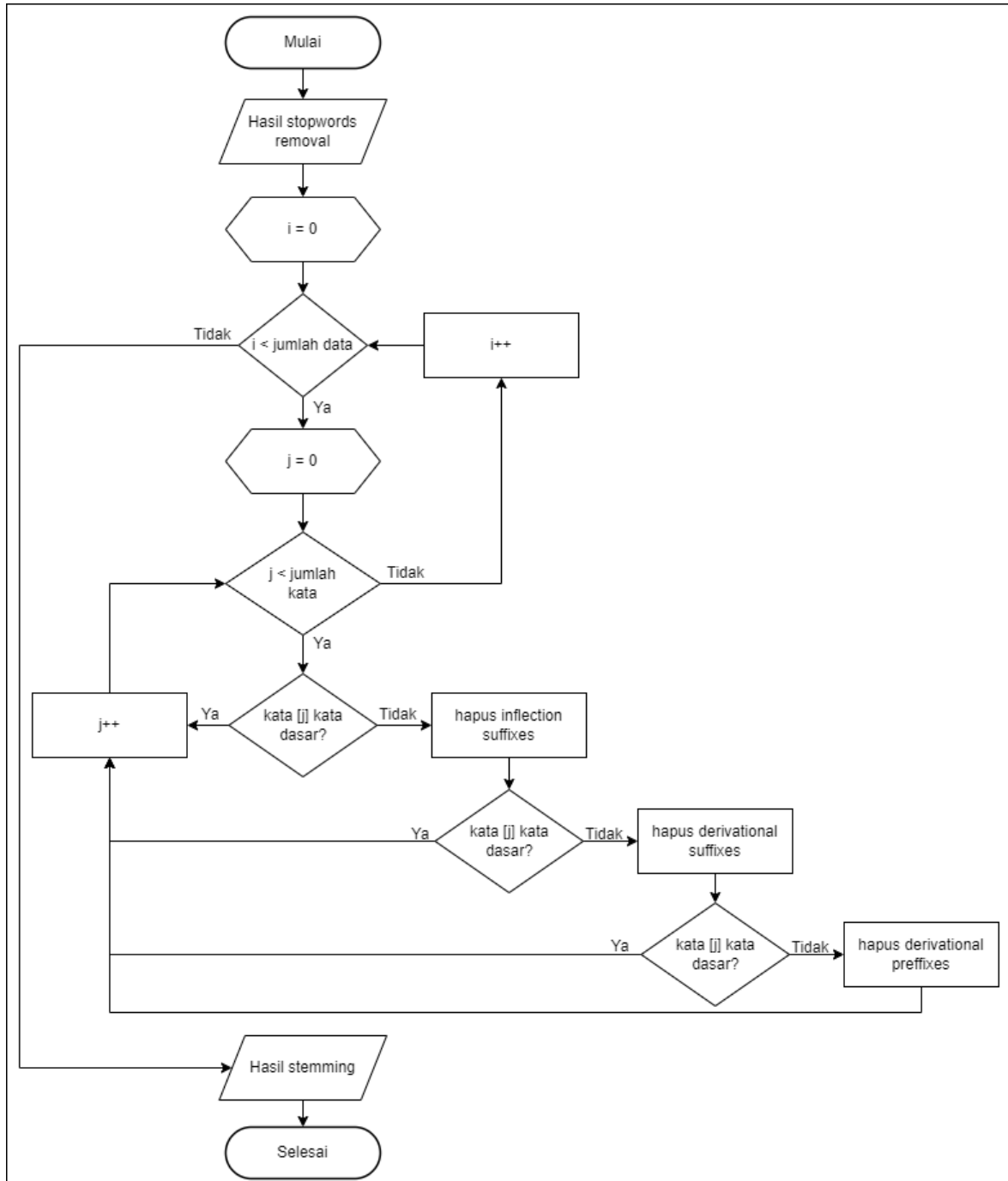
*Stopwords removal* merupakan proses untuk menghapus kata-kata yang tidak memiliki arti atau makna tetapi tidak mengubah arti atau makna dari sebuah kalimat. Berikut *flowchart* dari proses *stopwords removal* dapat dilihat pada Gambar 3.10.



**Gambar 3.10** Flowchart stopwords removal

### i. Stemming

*Stemming* merupakan proses untuk mengubah kata yang berhimpunan menjadi kata dasar dengan menghapus awalan, akhiran atau sisipan. Berikut *flowchart* dari proses *stemming* dapat dilihat pada Gambar 3.11.

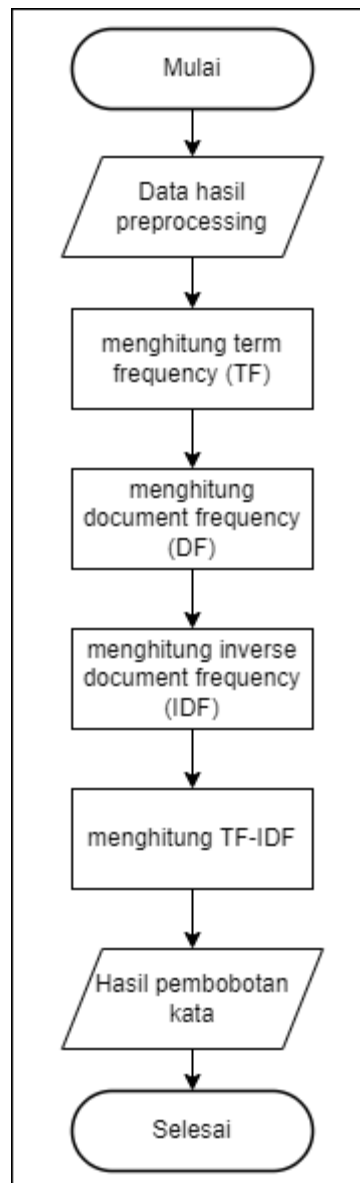


**Gambar 3.11** *Flowchart stemming*

### 3.1.3. Pembobotan Kata

Pembobotan kata menggunakan metode *Term Frequency-Inverse Document Frequency* (TF-IDF) merupakan sebuah metode untuk mengevaluasi pentingnya sebuah kata dalam sebuah dokumen. Dalam melakukan pembobotan kata, metode TF-IDF

mengkombinasikan dua faktor yaitu frekuensi kata dalam dokumen (*Term Frequency*) serta keberadaan kata tersebut di seluruh korpus dokumen (*Inverse Document Frequency*). Berikut *flowchart* dari proses TF-IDF dapat dilihat pada Gambar 3.12.



**Gambar 3.12** *Flowchart* pembobotan kata TF-IDF

Berikut merupakan langkah-langkah yang dilakukan dalam pembobotan kata menggunakan TF-IDF :

1. Menghitung *Term Frequency*

Menghitung *Term Frequency* (TF) dilakukan dengan menghitung frekuensi kemunculan kata di dalam dokumen tersebut. Berikut merupakan contoh perhitungan TF dari data yang telah melalui tahap *preprocessing* :

**Tabel 3.1** Contoh data

No	Dokumen	Label
1	aku benci orang sipit	Ujaran Kebecnian
2	dasar kaum cebong dongok	Ujaran Kebencian
3	kamu punya mata sipit	Bukan Ujaran Kebecnian
4	kata orang suara aku bagus	Bukan Ujaran Kebecnian
5	aku suka suara kamu	Bukan Ujaran Kebecnian

Dari tabel 3.1 dapat dilakukan perhitungan *Term Frequency* dengan menghitung jumlah kemunculan *term* atau kata pada tiap dokumen. Berikut merupakan hasil dari perhitungan *Term Frequency* :

**Tabel 3.2** Contoh perhitungan TF

No	Term	Term Frequency					DF
		D1	D2	D3	D4	D5	
1	aku	1	0	0	1	1	3
2	benci	1	0	0	0	0	1
3	orang	1	0	0	1	0	2
4	sipit	1	0	1	0	0	2
5	dasar	0	1	0	0	0	1
6	kaum	0	1	0	0	0	1
7	cebong	0	1	0	0	0	1
8	dongok	0	1	0	0	0	1
9	kamu	0	0	1	0	1	2
10	punya	0	0	1	0	0	1
11	mata	0	0	1	0	0	1
12	kata	0	0	0	1	0	1
13	suara	0	0	0	1	1	2
14	bagus	0	0	0	1	0	1
15	suka	0	0	0	0	1	1

2. Menghitung *Inverse Document Frequency*

*Inverse Document Frequency* (IDF) melakukan perhitungan seberapa sering sebuah kata muncul di dalam seluruh korpus dokumen. Untuk melakukan perhitungan IDF dapat menggunakan persamaan 2.1 dan sebagai contoh, pada *term* “aku” dengan jumlah total dokumen adalah 5 dan jumlah frekuensi dokumen *term* “aku” adalah 3, maka perhitungan IDF dilakukan sebagai berikut :

$$IDF = \log \frac{5}{3} = 0,2218$$

3. Mengkombinasikan nilai TF dan IDF

Setelah mendapatkan nilai dari TF dan IDF selanjutnya dilakukan perkalian untuk menentukan bobot dari sebuah kata. Hasil dari perkalian TF dan IDF dapat dilihat pada tabel 3.3

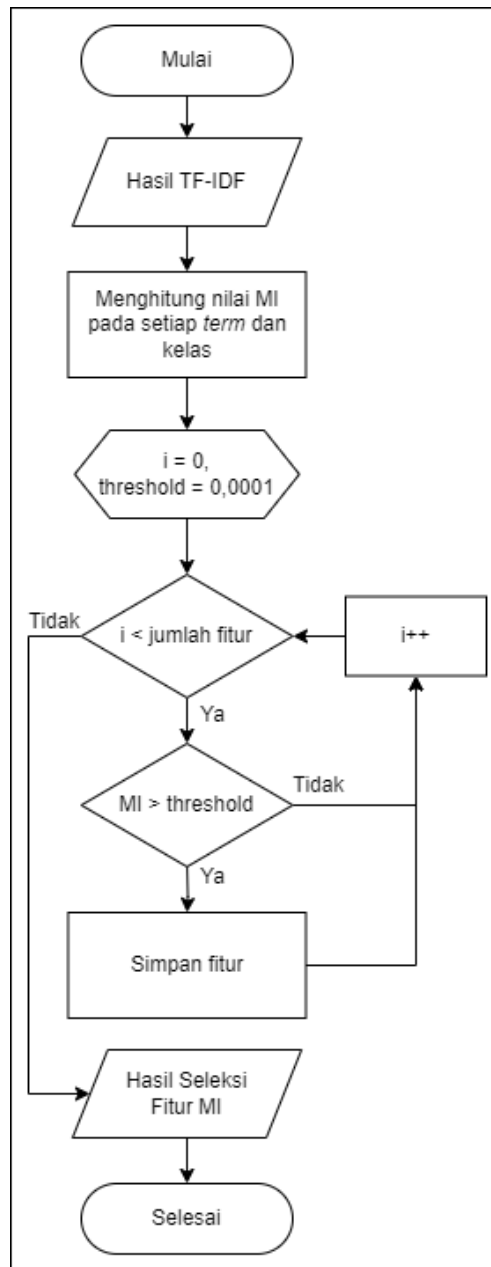


Tabel 3.3 Contoh hasil TF-IDF

No	Term	IDF	TF-IDF				
			D1	D2	D3	D4	D5
1	aku	0,2218	0,2218	0	0	0,2218	0,2218
2	benci	0,6989	0,6989	0	0	0	0
3	orang	0,3979	0,3979	0	0	0,3979	0
4	sipit	0,3979	0,3979	0	0,3979	0	0
5	dasar	0,6989	0	0,6989	0	0	0
6	kaum	0,6989	0	0,6989	0	0	0
7	cebong	0,6989	0	0,6989	0	0	0
8	dongok	0,6989	0	0,6989	0	0	0
9	kamu	0,3979	0	0	0,3979	0	0,3979
10	punya	0,6989	0	0	0,6989	0	0
11	mata	0,6989	0	0	0,6989	0	0
12	kata	0,6989	0	0	0	0,6989	0
13	suara	0,3979	0	0	0	0,3979	0,3979
14	bagus	0,6989	0	0	0	0,6989	0
15	suka	0,6989	0	0	0	0	0,6989

#### 3.1.4. Seleksi Fitur Mutual Information

Seleksi fitur adalah tahap dimana setiap *term* akan diseleksi dan dipilih untuk mengambil *term* yang paling relevan terhadap masing-masing kelas dan membuang *term* yang tidak relevan sehingga pada proses ini jumlah fitur akan berkurang dan menjadi informatif serta efektif. Pada penelitian ini akan menggunakan metode seleksi fitur *Mutual Information*. Metode *Mutual Information* atau MI digunakan untuk mengetahui seberapa penting atau seberapa banyak informasi yang terkandung dalam sebuah *term* untuk menentukan suatu kelas yang tepat pada klasifikasi. Setelah menghitung nilai MI selanjutnya akan dilakukan seleksi fitur dengan mengambil fitur yang memiliki nilai lebih besar dari nilai *threshold* yang ditentukan. Penentuan nilai *threshold* akan dijelaskan lebih lanjut pada bab selanjutnya. Berikut merupakan *flowchart* dari penerapan seleksi fitur *Mutual Information*:



**Gambar 3.13** Flowchart seleksi fitur *Mutual Information*

Berikut merupakan contoh perhitungan MI dengan menggunakan *term* “benci” dan kelas 1 atau ujaran kebencian.

**Tabel 3.4** Contingency Table

	ec = e”ujaran kebencian” = 1	ec = e”bukan ujaran kebencian” = 0
et = e”benci” = 1	1	0
et = e”benci” = 0	1	3

Dari tabel 3.5 didapat persamaan sebagai berikut :

$$I(\text{benci, ujaran kebencian}) = \frac{1}{5} \log_2 \frac{5 \times 1}{(1 + 0) \times (1 + 1)} + \frac{1}{5} \log_2 \frac{5 \times 1}{(1 + 3) \times (1 + 1)}$$

$$+ \frac{0}{5} \log_2 \frac{5 \times 0}{(1 + 0) \times (0 + 3)} + \frac{3}{5} \log_2 \frac{5 \times 3}{(1 + 3) \times (0 + 3)}$$

$$\approx 0,3219$$

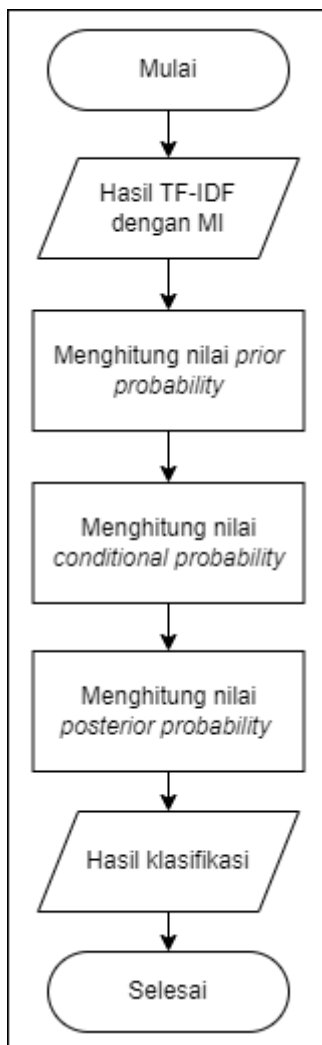
Setelah semua nilai MI telah didapatkan, selanjutnya akan dipilih fitur dengan nilai MI lebih besar dari pada nilai *threshold* yang sudah ditentukan. Hasil dari seleksi fitur dapat dilihat pada tabel 3.6.

**Tabel 3.5** Hasil seleksi fitur *Mutual Information*

No	Term	Nilai MI
1	aku	0.0199
2	benci	0.3219
3	orang	0.0199
4	sipit	0.0199
5	dasar	0.3219
6	kaum	0.3219
7	cebong	0.3219
8	dongok	0.3219
9	kamu	0.4199
10	punya	0.1709
11	mata	0.1709
12	kata	0.1709
13	suara	0.4199
14	bagus	0.1709
15	suka	0.1709

### 3.1.5. Naïve Bayes

*Naïve Bayes* merupakan metode yang melakukan prediksi peluang terjadinya kejadian di masa depan berdasarkan data atau pengalaman yang ada sebelumnya. Penelitian ini akan menerapkan metode tersebut untuk melakukan klasifikasi teks ujaran kebencian. Berikut merupakan *flowchart* dari penerapan metode *Naïve Bayes* pada klasifikasi teks :



**Gambar 3.14** Flowchart *Naïve Bayes*

Berikut merupakan langkah dari perhitungan *Naïve Bayes* dengan menggunakan data “dasar dongok banget” :

- a. Menghitung *prior probability* (probabilitas prior)

**Kelas Ujaran Kebencian (HS)**

$$P(\text{“HS”}) = \frac{\text{jumlah dokumen pada kelas ujaran kebencian}}{\text{jumlah seluruh dokumen}}$$

$$P(\text{“HS”}) = \frac{2}{5}$$

$$P(\text{“HS”}) = 0,4$$

**Kelas Bukan Ujaran Kebencian (Non\_HS)**

$$P(\text{“Non_HS”}) = \frac{\text{jumlah dokumen pada kelas bukan ujaran kebencian}}{\text{jumlah seluruh dokumen}}$$

$$P(\text{“Non_HS”}) = \frac{3}{5}$$

$$P(\text{“Non_HS”}) = 0,6$$

- b. Menghitung *conditional probability* (probabilitas setiap *term*)

**Kelas Ujaran Kebencian (HS)**

$$P(\text{“dasar”}|\text{“HS”}) = \frac{(0,6989 + 1)}{(1 + 15)} = 0,1061$$

$$P(\text{"dongok"}|\text{"HS"}) = \frac{(0,6989 + 1)}{(1 + 15)} = 0,1061$$

$$P(\text{"banget"}|\text{"HS"}) = \frac{(0 + 1)}{(0 + 15)} = 0,0667$$

**Kelas Bukan Ujaran Kebencian (Non\_HS)**

$$P(\text{"dasar"}|\text{"Non_HS"}) = \frac{(0 + 1)}{(0 + 15)} = 0,0667$$

$$P(\text{"dongok"}|\text{"Non_HS"}) = \frac{(0 + 1)}{(0 + 15)} = 0,0667$$

$$P(\text{"banget"}|\text{"Non_HS"}) = \frac{(0 + 1)}{(0 + 15)} = 0,0667$$

- c. Menghitung nilai *posterior probability* (maksimal probabilitas)

**Kelas Ujaran Kebencian (HS)**

$$\begin{aligned} \text{Probabilitas} &= 0,4 \times 0,1061 \times 0,1061 \times 0,0667 \\ &= 0,0003003 \end{aligned}$$

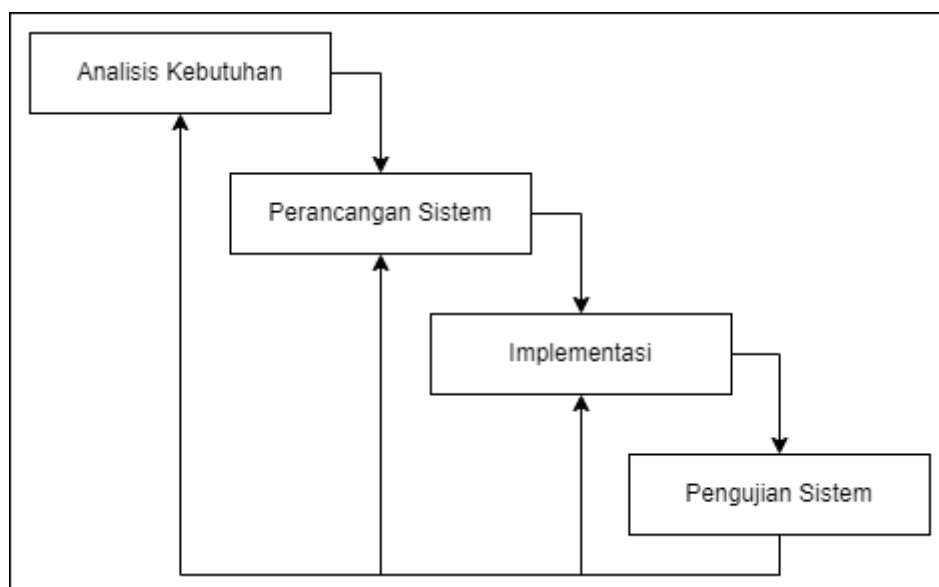
**Kelas Bukan Ujaran Kebencian (Non\_HS)**

$$\begin{aligned} \text{Probabilitas} &= 0,6 \times 0,0667 \times 0,0667 \times 0,0667 \\ &= 0,0001780 \end{aligned}$$

Dari hasil *posterior probability* pada setiap kelas, maka didapatkan nilai tertinggi pada kelas ujaran kebencian (HS) yaitu **0,0003003** sehingga kalimat “dasar dongok banget” diklasifikasikan ke dalam kelas ujaran kebencian.

### 3.2. Pengembangan Sistem

Dalam penelitian ini metode pengembangan sistem yang digunakan adalah metode *Waterfall* (Pressman, 2014). Metode *Waterfall* memiliki 5 tahapan yaitu, analisis kebutuhan, perancangan sistem, implementasi, pengujian sistem, dan pemeliharaan sistem. Namun pada penelitian ini tahapan dari metode waterfall hanya dilakukan sampai pengujian sistem, tahapan pemeliharaan sistem tidak dilakukan. Tahapan dalam metode pengembangan sistem dengan *Waterfall* dapat dilihat pada gambar 3.15



**Gambar 3.15** Metode pengembangan sistem

Berdasarkan pada gambar 3.15 metode *Waterfall* memiliki 4 tahapan dalam melakukan pengembangan sistem yaitu sebagai berikut :

### 3.2.1. Analisis Kebutuhan

Tahapan analisis kebutuhan merupakan proses untuk melakukan indentifikasi atau mengumpulkan kebutuhan dari sebuah sistem yang akan dibangun. Analisis kebutuhan terdiri dari dua bagian, yaitu kebutuhan fungsional dan non fungsional.

#### 1) Kebutuhan Fungsional

Kebutuhan fungsional merupakan semua proses yang dapat dilakukan oleh sistem yang akan dibangun. Kebutuhan fungsional yang diperlukan pada penelitian ini adalah sebagai berikut :

- a) Sistem dapat melakukan *input* data teks
- b) Sistem dapat melakukan *preprocessing* pada data *input* teks
- c) Sistem dapat melakukan proses klasifikasi pada data teks
- d) Sistem dapat menampilkan akurasi dari pengujian

#### 2) Kebutuhan Non Fungsional

Kebutuhan non fungsional merupakan spesifikasi minimum perangkat keras, perangkat lunak, dan pengguna yang digunakan dalam melakukan penelitian ini. Berikut kebutuhan non fungsional pada penelitian ini :

- a) Kebutuhan Perangkat Keras (*Hardware*)

**Tabel 3.6** Spesifikasi perangkat keras

No	Perangkat Keras	Keterangan
1	<i>Processor</i>	Intel Core i3-4030U 1.90GHz
2	RAM	8 GB
3	<i>Storage</i>	1TB
4	Perangkat <i>input</i> dan <i>output</i>	<i>Mouse, keyboard, monitor</i>
5	Koneksi	WiFi

- b) Kebutuhan Perangkat Lunak (*Software*)

**Tabel 3.7** Spesifikasi perangkat lunak

No	Perangkat Lunak	Keterangan
1	Windows 10 Pro 64 bit	Sistem operasi
2	Python 3	Bahasa pemrograman
3	Jupyter Notebook atau Google colab	Web open source
4	Draw.io	Desain sistem dan flowchart
5	Figma	Desain antarmuka
6	Google Chrome	Web browser
7	Visual Studio Code	Kode editor

- c) Kebutuhan Pengguna (*User*)

**Tabel 3.8** Spesifikasi pengguna

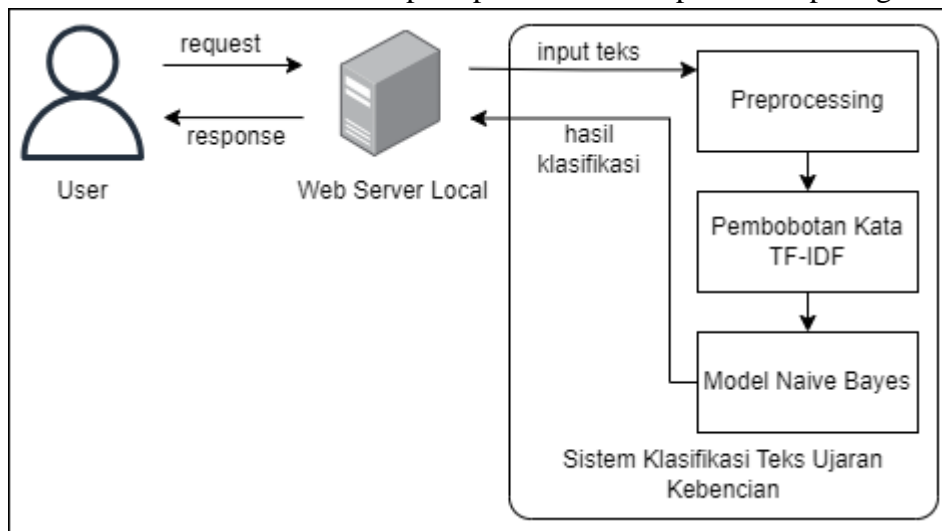
No	Keterangan
1	Memahami cara menggunakan komputer
2	Memahami cara menggunakan web browser
3	Memahami cara pengoperasian sistem klasifikasi teks dalam mengklasifikasikan ujaran kebencian

### 3.2.2. Perancangan Sistem

Tahapan perancangan sistem akan membahas tentang perancangan untuk membangun sistem klasifikasi teks ujaran kebencian serta untuk mencapai tujuan yang diinginkan. Perancangan yang akan dibuat meliputi perancangan arsitektur sistem, perancangan sistem, dan perancangan antarmuka.

#### a) Perancangan Arsitektur Sistem

Proses perancangan arsitektur sistem merupakan gambaran dari jalannya sistem secara keseluruhan. Arsitektur sistem pada penelitian ini dapat dilihat pada gambar 3.16.

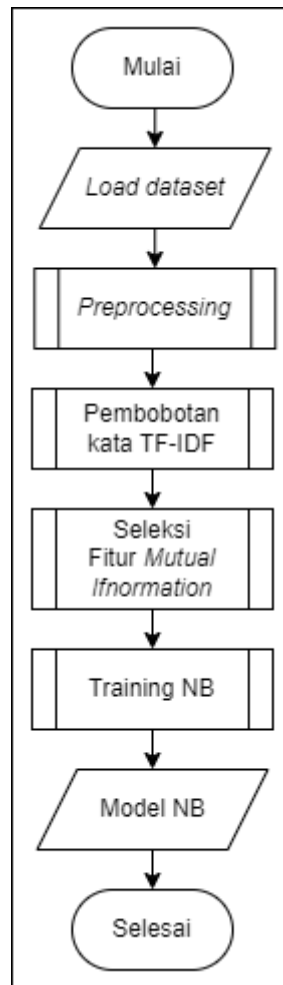


**Gambar 3.16** Arsitektur sistem

Pada gambar 3.16 *user* melakukan *input* teks terhadap aplikasi klasifikasi teks ujaran kebencian. Setelah itu, sistem akan melakukan tahapan *preprocessing* pada data teks tersebut, kemudian data tersebut dilakukan proses klasifikasi menggunakan metode *Naïve Bayes* dengan seleksi fitur *Mutual Information*. Data dari hasil klasifikasi tersebut akan ditampilkan pada aplikasi klasifikasi teks kepada *user*.

#### b) Perancangan Sistem

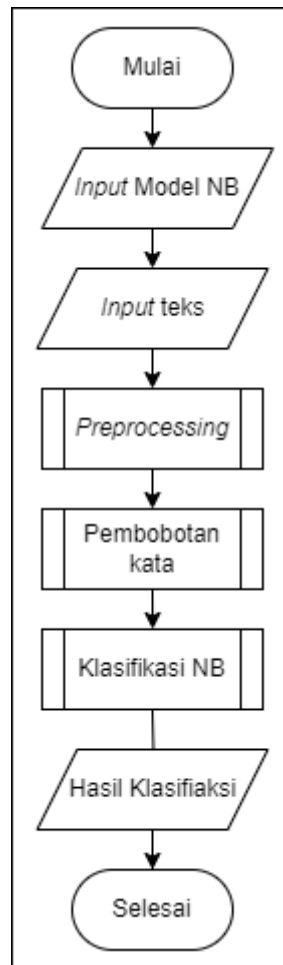
Proses perancangan sistem bertujuan untuk memahami alur sistem yang akan dibangun. Pada penelitian ini sistem akan memuat model NB yang telah dibuat, Proses dari pembuatan model tersebut dapat dilihat pada *flowchart* model NB yang ditunjukkan pada gambar 3.17.



**Gambar 3.17** Flowchart pembuatan model *Naïve Bayes*

Pada gambar 3.17 pembuatan model NB dimulai dengan melakukan *load dataset* yang selanjutnya dilakukan proses *preprocessing* yang terdiri dari *case folding*, *remove emoji*, *cleansing*, *remove repetition character*, *tokenization*, normalisasi kata, negasi, *stopwords removal*, dan *stemming*. Kemudian dilakukan proses pembobotan kata menggunakan TF-IDF dan dilanjutkan dengan seleksi fitur menggunakan *Mutual Information*. Hasil dari proses sebelumnya akan digunakan untuk melakukan *training* pembuatan model *Naïve Bayes* yang akan digunakan dalam proses klasifikasi.





**Gambar 3.18** Flowchart klasifikasi

Pada gambar 3.18 menampilkan proses klasifikasi pada penelitian ini. Pertama model yang telah dibuat akan dimuat untuk digunakan pada proses klasifikasi, kemudian *input* data teks atau kalimat yang akan diklasifikasikan, selanjutnya data tersebut akan melalui proses *preprocessing* dan pembobotan kata sebelum dilakukannya proses klasifikasi menggunakan model yang telah dibangun. Setelah itu, maka akan didapatkan hasil dari klasifikasi teks ujaran kebencian.

c) Perancangan Antarmuka

Pada proses perancangan antarmuka akan dilakukan pembuatan desain antarmuka dari sistem klasifikasi teks yang akan dibangun. Desain antarmuka tersebut sebagai berikut :

1) Halaman klasifikasi

Halaman klasifikasi digunakan untuk melakukan input teks dan akan menampilkan hasil dari klasifikasi teks ujaran kebencian. Pada halaman ini terdapat *text box* untuk menuliskan teks dan tombol *submit* untuk memproses teks yang diinputkan dan mengarahkan ke halaman hasil klasifikasi.



**Gambar 3.19** Halaman *input* teks

Setelah melakukan input teks dan menekan tombol *submit*, maka akan menampilkan halaman yang berisikan hasil klasifikasi teks beserta hasil *preprocessing*, probabilitas, dan kelas klasifikasinya.



**Gambar 3.20** Halaman hasil klasifikasi

## 2) Halaman dataset

Pada halaman ini akan menampilkan dataset yang digunakan dan memiliki 4 sub menu, yaitu halaman dataset untuk menampilkan semua data, halaman *clean* data untuk menampilkan semua data hasil *preprocessing*, halaman data *train* untuk menampilkan data *train* yang digunakan, dan halaman data *test* untuk menampilkan data *test* yang digunakan.

No	Teks	Label

**Gambar 3.21** Halaman dataset

### 3) Halaman *overview*

Halaman *overview* merupakan halaman untuk menampilkan hasil dari *confusion matrix* dan nilai akurasi, *precision*, serta *recall* dari model yang telah dibangun.

Evaluasi Model		
Akurasi	Precision	Recall

**Gambar 3.22** Halaman *overview*

### 3.2.3. Implementasi

Tahapan implementasi merupakan tahapan dimana hasil dari tahapan perancangan sistem diterapkan pada aplikasi menggunakan bahasa pemrograman python dan *library* nya. Implementasi pada tahapan ini akan menghasilkan aplikasi yang berbasis *website* yang dimana pengguna atau *user* dapat melakukan klasifikasi teks ujaran kebencian dengan menginputkan teks atau sebuah kalimat, kemudian teks tersebut akan diklasifikasikan oleh aplikasi yang telah dibangun.

### 3.2.4. Pengujian Sistem

Pada tahapan pengujian sistem bertujuan untuk mengetahui seberapa baik sistem yang telah dibangun dalam melakukan klasifikasi teks ujaran kebencian. Pengujian sistem akan menggunakan *Confusional Matrix* untuk menghitung nilai akurasi, presisi, dan *recall*. *Confusion Matrix* akan digunakan untuk mengetahui perbandingan nilai akurasi, presisi, dan *recall* pada setiap skenario *split data training* dan *data testing*. Rancangan pengujian sistem

dengan *Confusion Matrix* dan skenario split dataset masing-masing dapat dilihat pada tabel 3.10 dan tabel 3.11.

**Tabel 3.9** Pengujian *Confusion Matrix*

		Prediksi	
		Ujaran Kebencian	Bukan Ujaran Kebencian
Aktual	Ujaran Kebencian		
	Bukan Ujaran Kebencian		

**Tabel 3.10** Perbandingan skenario *split data training* dan data *testing*

Skenario	Training : Testing	Model Naïve Bayes	Akurasi	Presisi	Recall
1	70:30	Tanpa <i>Mutual Information</i>			
2	70:30	Dengan <i>Mutual Information</i>			
3	80:20	Tanpa <i>Mutual Information</i>			
4	80:20	Dengan <i>Mutual Information</i>			
5	90:10	Tanpa <i>Mutual Information</i>			
6	90:10	Dengan <i>Mutual Information</i>			

Pada tabel 3.11 terdapat beberapa skenario *split dataset* yang menggunakan model *Naïve Bayes* dengan seleksi fitur *Mutual Information* dimana pada setiap skenario tersebut akan dilakukan pengujian untuk mencari nilai *threshold* yang memiliki akurasi tertinggi.

## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1. Hasil Penelitian

Pada bagian ini akan menjelaskan hasil penelitian yang berupa implementasi dari rancangan yang telah dibuat pada bab sebelumnya. Implementasi pada penelitian ini dimulai dari pengumpulan data, data *preprocessing*, pembobotan kata menggunakan *Term Frequency-Inverse Document Frequency* (TF-IDF), seleksi fitur menggunakan *Mutual Information*, pembuatan model menggunakan *Naïve Bayes*, dan pengujian model yang telah dibangun menggunakan *Confusion Matrix*.

##### 4.1.1. Pengumpulan Data

Pengumpulan data dengan mengambil dataset yang didapatkan melalui situs resmi *kaggle*. Dataset tersebut berisi kumpulan data teks yang masing-masing data memiliki sebuah label yaitu ujaran kebencian atau bukan ujaran kebencian. Selanjutnya dataset tersebut akan dimuat kedalam program untuk dilakukan proses selanjutnya. Proses ini dapat dilihat pada modul program 4.1.

---

**Algoritma 1:** Pengumpulan Data

---

```
Input      :  
            dataset with .csv format  
  
Output     :  
            dataset with data type dataframe  
  
Procedure  :  
            data = pd.read_csv('dataset.csv', encoding='latin-1')  
            data = data[['Tweet', 'HS']]
```

---

##### Modul Program 4.1 Pengumpulan data

##### 4.1.2. Data Preprocessing

Proses selanjutnya yaitu melakukan data *preprocessing*. Proses ini memiliki beberapa tahapan, dimulai dari *case folding*, *remove emoji*, *cleansing*, *remove repetition character*, *tokenization*, normalisasi kata, negasi, *stopwords removal*, dan *stemming*. Implementasi dari proses tersebut dapat dilihat sebagai berikut.

##### a) *Case Folding*

Pada tahap *case folding* akan mengubah semua huruf kapital pada dataset menjadi huruf kecil. Proses ini dapat dilihat pada modul program 4.2.

---

**Algoritma 2:** *Case Folding*

---

```
Input      :  
            Text  
  
Output     :  
            Text tanpa huruf kapital  
  
Procedure  :  
            def case_folding(text):  
                text = text.lower()
```

---

---

```
return text
```

---

### Modul Program 4.2 Case folding

#### b) *Remove Emoji*

Setelah melakukan *case folding*, tahap selanjutnya adalah *remove emoji*. Pada tahap ini akan dilakukan penghapusan simbol-simbol yang berupa *emoji*. Proses ini dapat dilihat pada modul program 4.3.

---

#### Algoritma 3: *Remove Emoji*

---

```
Input      :
           : Text tanpa huruf kapital
Output     :
           : Text tanpa simbol emoji
Procedure  :
           :
           : def emoji(text):
           :     text = re.sub(r'^\x00-\x7f', r'', text)
           :     text = re.sub(r'(\u[0-9A-Fa-f]+)', r'', text)
           :     return text
```

---

### Modul Program 4.3 Remove emoji

#### c) *Cleansing*

Setelah melakukan *remove emoji*, tahap selanjutnya adalah *cleansing*. Pada tahap ini akan dilakukan penghapusan angka, tanda baca, *mention*, url, *hashtag*, dan simbol-simbol yang tidak berguna dalam klasifikasi teks. Proses ini dapat dilihat pada modul program 4.4.

---

#### Algoritma 4: *Cleansing*

---

```
Input      :
           : Text hasil remove emoji
Output     :
           : Text hasil cleansing
Procedure  :
           :
           : def cleaning_text(text):
           :     text = re.sub(r'@[w]*', ' ', text)
           :     text = re.sub(r'\\[w]*', r'', text)
           :     text = re.sub(r"^[A-Za-z0-9^!.\/'+-=]", " ", text)
           :     text = re.sub(r'\\u[w]w[w]w', '', text)
           :     text = re.sub(r'http\S+', '', text)
           :     text = re.sub(r'user', '', text)
           :     text = re.sub(r'rt', '', text)
           :     text = re.sub(r'#([\s]+)', '', text)
           :     text = re.sub(r"[.,;+!\\_<^/?\"'\\(\\)\\d]*", " ", text)
           :     return text
```

---

### Modul Program 4.4 Cleansing

#### d) *Remove Repetition Character*

Setelah melakukan *cleansing*, tahap selanjutnya adalah *remove repetition character*. Pada tahap ini akan dilakukan penghapusan huruf atau karakter yang berulang dalam suatu kata. Proses ini dapat dilihat pada modul program 4.5.

---

**Algoritma 5: Remove repetition character**

---

Input : Text hasil cleansing

Output : Text hasil remove repetition character

Procedure :

```
def replaceThreeOrMore(text):
    pattern = re.compile(r"(\.){2,}", re.DOTALL)
    return pattern.sub(r"\1", text)
```

---

**Modul Program 4.5 Remove repetition character****e) Tokenization**

Setelah melakukan *remove repetition character*, tahap selanjutnya adalah *tokenization*. Pada tahap ini akan dilakukan pemisahan kalimat menjadi kumpulan kata. Proses ini dapat dilihat pada modul program 4.6.

---

**Algoritma 6: Tokenization**

---

Input : Text hasil remove repetition character

Output : Text hasil tokenization

Procedure :

```
def tokenize(text):
    return word_tokenize(text)
```

---

**Modul Program 4.6 Tokenization****f) Normalisasi Kata**

Setelah melakukan *tokenization*, tahap selanjutnya adalah normalisasi kata. Pada tahap ini akan dilakukan pengubahan kata tidak baku atau *slang words* serta singkatan menjadi kata baku sesuai dengan kamus normalisasi kata yang tersedia. Proses ini dapat dilihat pada modul program 4.7.

---

**Algoritma 7: Normalisasi kata**

---

Input : Text hasil tokenization

Output : Text hasil normalisasi kata

Procedure :

```
def convertToSlangword(text):
    kamus_slangword = eval(open("kamus_slang.txt").read())
    pattern = re.compile(r'\b(' + '|'.join(
        (kamus_slangword.keys()) + r')\b')
    content = []
    for kata in text:
        filteredSlang = pattern.sub(lambda x:
            kamus_slangword[x.group()], kata)
        content.append(filteredSlang.lower())
    text = content
    return text
```

---

**Modul Program 4.7 Normalisasi kata**

### g) Negasi

Setelah melakukan normalisasi kata, tahap selanjutnya adalah negasi. Pada tahap ini akan dilakukan perubahan semua kata setelah kata “tidak” menjadi kata yang sesuai dengan kamus negasi. Proses ini dapat dilihat pada modul program 4.8.

---

#### Algoritma 8: Negasi

---

```

Input      :
            Text hasil normalisasi kata

Output     :
            Text hasil negasi

Procedure  :
            def ganti_negasi(w):
                kamus_negasi = eval(open("negasi.txt").read())
                w_splited = w.split(' ')
                if 'tidak' in w_splited:
                    index_negasi = w_splited.index('tidak')
                    for i,k in enumerate(w_splited):
                        if k in kamus_negasi and w_splited[i-1] ==
                            'tidak':
                                w_splited[i] = kamus_negasi[k]
                return ' '.join(w_splited)

```

---

#### Modul Program 4.8 Negasi

### h) Stopwords Removal

Setelah melakukan negasi, tahap selanjutnya adalah *stopwords removal*. Pada tahap ini akan dilakukan penghapusan semua kata yang tidak memiliki arti atau makna dalam melakukan klasifikasi. Pada penelitian ini menggunakan kamus *stopwords* yang berasal dari *library Sastrawi*. Proses ini dapat dilihat pada modul program 4.9.

---

#### Algoritma 9: Stopwords removal

---

```

Input      :
            Text hasil negasi

Output     :
            Text hasil stopwords removal

Procedure  :
            def remove_stopword(text, stop_words=stop_words):
                word_tokens = word_tokenize(text)
                filtered_sentence = [w for w in word_tokens if not w in
                                    stop_words]
                return ' '.join(filtered_sentence)

```

---

#### Modul Program 4.9 Stopwords removal

### i) Stemming

Setelah melakukan *stopwords removal*, tahap selanjutnya adalah *stemming*. Pada tahap ini akan dilakukan perubahan semua kata menjadi kata dasar. Pada penelitian ini proses *stemming* akan menggunakan *library* dari *Sastrawi* yang dimana *library* ini menggunakan algoritma Nazief dan Adriani. Proses ini dapat dilihat pada modul program 4.10.



---

**Algoritma 10: Stemming**

---

Input : Text hasil stopwords removal

Output : Text hasil stemming

Procedure :

```
def stemming_and_lemmatization(text):
    factory = StemmerFactory()
    stemmer = factory.create_stemmer()
    return stemmer.stem(text)
```

---

**Modul Program 4.10 Stemming****4.1.3. Pembobotan Kata Menggunakan TF-IDF**

Proses selanjutnya yaitu pembobotan kata menggunakan TF-IDF. Proses pembobotan kata akan dilakukan menggunakan *library* dari *Scikit-learn* yaitu *TfidfVectorizer*. Setelah hasil dari pembobotan kata TF-IDF didapatkan, maka hasil tersebut akan disimpan dengan *file* berformat *pickle*. Proses ini dapat dilihat pada modul program 4.11.

---

**Algoritma 11: Pembobotan kata TF-IDF**

---

Input : Text hasil preprocessing

Output : Hasil pembobotan kata TF-IDF

Procedure :

```
tfidf_vectorizer = TfidfVectorizer()
X_train_tfidf = tfidf_vectorizer.fit_transform(
    X_train.values.astype('U'))
X_test_tfidf = tfidf_vectorizer.transform(
    X_test.values.astype('U'))

with open('model_tfidf.pkl', 'wb') as f:
    pickle.dump(tfidf_vectorizer, f)
```

---

**Modul Program 4.11 Pembobotan kata TF-IDF****4.1.4. Seleksi Fitur Menggunakan Mutual Information**

Tahap selanjutnya adalah melakukan seleksi fitur menggunakan *Mutual Information*. Pada tahap ini fitur-fitur yang telah dilakukan pembobotan kata akan dihitung nilai *Mutual Information* untuk setiap fiturnya. Proses seleksi fitur akan dilakukan menggunakan *library* dari *Scikit-learn* yaitu *Mutual\_info\_classif*. Proses ini dapat dilihat pada modul program 4.12.

---

**Algoritma 12: Perhitungan nilai Mutual Information**

---

Input : Hasil pembobotan kata TF-IDF

Output : Hasil perhitungan nilai MI

Procedure :

```
mi = mutual_info_classif(X_train_tfidf, y_train)
```

---

**Modul Program 4.12 Perhitungan nilai MI**

Setelah mendapatkan nilai *Mutual Information*, maka akan dilakukan seleksi fitur atau memilih fitur-fitur yang lebih besar dari nilai *threshold*. Untuk menentukan nilai *threshold* yang akan digunakan, maka akan dilakukan perulangan untuk mencari nilai *threshold* yang memiliki akurasi tertinggi. Pada perulangan yang dilakukan akan menggunakan nilai *threshold* dari 0,001 hingga 0,00001 dengan *step* atau *decrement* sebesar 0.0000001. Proses ini dapat dilihat pada modul program 4.13.

---

**Algoritma 13:** Pencarian nilai *threshold*

---

```

Input      :
            Hasil perhitungan nilai MI

Output     :
            Hasil seleksi fitur MI

Procedure  :
            mi_lower = 0.00001
            mi_upper = 0.001
            mi_step  = -0.0000001
            mi_values = []
            mi_features = []
            prev_num_features = -1

            for threshold in np.arange(mi_upper, mi_lower, mi_step):
                selected_indices = np.where(mi > threshold)[0]
                num_features = len(selected_indices)
                if num_features != prev_num_features:
                    mi_values.append((threshold, num_features))
                    mi_features.append((threshold, selected_indices))
                    prev_num_features = num_features

            clf = ComplementNB()
            accuracy_each_model = []
            for threshold, features in mi_features:
                formatted_threshold = "{:.8f}".format(threshold)
                X_train_tfidf_selected = X_train_tfidf[:, features]
                X_test_tfidf_selected = X_test_tfidf[:, features]
                NB = clf.fit(X_train_tfidf_selected, y_train)
                y_pred = clf.predict(X_test_tfidf_selected)
                accuracy = accuracy_score(y_test, y_pred)
                accuracy_each_model.append((accuracy,
                    formatted_threshold, len(features)))

```

---

**Modul Program 4.13** Perulangan nilai *threshold*

Modul program 4.13 akan digunakan pada skenario *split dataset* 70:30, 80:20, dan 90:10 sehingga mendapatkan nilai *threshold* yang memiliki akurasi tertinggi. Hasil dengan akurasi tertinggi dari masing-masing skenario *split dataset* dapat dilihat sebagai berikut.

**Tabel 4.1** Hasil nilai *threshold* dengan akurasi tertinggi

No	Skenario Split Dataset	Threshold	Jumlah Fitur	Akurasi
1	70:30	0,0003017	3064	82,47%
2	80:20	0,0001725	5783	83,4%
3	90:10	0,0003076	3231	<b>84,4 %</b>

Setelah hasil dari seleksi fitur MI pada setiap skenario *split dataset* didapatkan, maka hasil tersebut akan disimpan dengan *file* berformat *pickle*. Proses ini dapat dilihat pada modul program 4.14.

---

**Algoritma 14:** *Export* hasil seleksi fitur MI

---

```

Input      :
            Hasil seleksi fitur MI

Output     :
            Export hasil seleksi fitur MI

Procedure  :
            with open('model_selected_feature.pkl', 'wb') as f:
                pickle.dump(selected_feature_indices, f)

```

---

**Modul Program 4.14** *Export* hasil seleksi fitur MI

#### 4.1.5. Pembuatan Model Naïve Bayes

Tahapan selanjutnya adalah pembuatan model *Naïve Bayes* yang menggunakan fitur-fitur hasil dari seleksi fitur dengan *Mutual Information*. Proses dari pembuatan model *Naïve Bayes* akan dilakukan menggunakan *library* dari *Scikit-learn* yaitu *ComplementNB* kemudian model tersebut akan disimpan dengan *file* berformat *pickle*. Proses ini dapat dilihat pada modul program 4.15.

---

**Algoritma 15:** Pembuatan model *Naïve Bayes*

---

```

Input      :
            Hasil seleksi fitur MI

Output     :
            Export model naïve bayes

Procedure  :
            X_train_tfidf_selected = X_train_tfidf[:,
                selected_feature_indices]
            X_test_tfidf_selected = X_test_tfidf[:,
                selected_feature_indices]

            clf = ComplementNB()
            NB = clf.fit(X_train_tfidf_selected, y_train)
            y_pred = clf.predict(X_test_tfidf_selected)
            accuracy = accuracy_score(y_test, y_pred)
            print('Accuracy:', accuracy)

```

---

**Modul Program 4.15** Pembuatan model *Naïve Bayes*

#### 4.1.6. Pengujian Confusion Matrix

Setelah pembuatan model *Naïve Bayes*, selanjutnya akan dilakukan pengujian menggunakan *Confusion Matrix*. Pengujian ini bertujuan untuk mengetahui seberapa baik model yang telah dibangun. Proses dari pengujian model *Naïve Bayes* menggunakan *Confusion Matrix* akan dilakukan menggunakan *library* dari *Scikit-learn* yaitu *Confusion\_matrix*. Proses ini dapat dilihat pada modul program 4.16.

---

**Algoritma 16:** Pengujian model menggunakan *Confusion Matrix*

---

```

Input      :
            Model naïve bayes

Output     :
            Hasil confusion matrix

```

---

---

```

Procedure :
confusion_mat = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(8, 6))
sns.heatmap(confusion_mat, annot=True, fmt='d',
            cbar=False)
plt.xlabel('Predicted Labels')
plt.ylabel('Actual Labels')
plt.title('Confusion Matrix NB dengan MI (split dataset
          80:20)')
plt.show()

```

---

#### **Modul Program 4.16** Pengujian model menggunakan *Confusion Matrix*

Pada penelitian ini terdapat 6 skenario pengujian menggunakan *Confusion Matrix* yang dapat dilihat pada tabel 3.11. Hasil dari pengujian setiap skenario sebagai berikut :

- 1) Skenario *split* data *training* dan data *testing* 70:30 dan model tanpa seleksi fitur.

Pada skenario ini, akan dilakukan *split dataset* yang bertujuan untuk membagi dataset menjadi data *training* dan data *testing* menggunakan perbandingan 70:30. Selanjutnya hasil dari *split dataset* tersebut akan digunakan untuk membangun model *Naïve Bayes* tanpa seleksi fitur *Mutual Information*. Kemudian dilakukan pengujian menggunakan *Confusion Matrix* untuk mengetahui nilai akurasi, nilai *precision*, dan nilai *recall*. Hasil dari pengujian tersebut dapat dilihat pada tabel berikut.

**Tabel 4.2** Pengujian *Confusion Matrix* pada skenario 1

		Prediksi	
		Ujaran Kebencian	Bukan Ujaran Kebencian
Aktual	Ujaran Kebencian	1307	156
	Bukan Ujaran Kebencian	385	1152

Dari hasil pengujian menggunakan *Confusion Matrix*, maka didapatkan nilai dari akurasi, *precision*, dan *recall* sebagai berikut.

$$\text{Akurasi} = \frac{1307 + 1152}{1307 + 1152 + 385 + 156} = 0,8197 \times 100 = 81,97\%$$

$$\text{Recall Positif} = \frac{1307}{1307 + 156} = 0,8934 \times 100 = 89,34\%$$

$$\text{Recall Negatif} = \frac{1152}{1152 + 385} = 0,7495 \times 100 = 74,95\%$$

$$\text{Recall Rata-Rata} = \frac{89,34 + 74,95}{2} = 0,8214 \times 100 = 82,14\%$$

$$\text{Precision Positif} = \frac{1307}{1307 + 385} = 0,7725 \times 100 = 77,25\%$$

$$\text{Precision Negatif} = \frac{1152}{1152 + 156} = 0,8807 \times 100 = 88,07\%$$

$$\text{Precision Rata-Rata} = \frac{77,25 + 88,07}{2} = 0,8266 \times 100 = 82,66\%$$

- 2) Skenario *split* data *training* dan data *testing* 70:30 dan model dengan seleksi fitur.

Pada skenario ini, akan dilakukan *split dataset* yang bertujuan untuk membagi dataset menjadi data *training* dan data *testing* menggunakan perbandingan 70:30. Selanjutnya hasil dari *split dataset* tersebut akan digunakan untuk membangun model *Naïve Bayes* dengan

seleksi fitur *Mutual Information* yang menggunakan nilai *threshold* berdasarkan tabel 4.1 yaitu sebesar 0,0003017. Kemudian dilakukan pengujian menggunakan *Confusion Matrix* untuk mengetahui nilai akurasi, nilai *precision*, dan nilai *recall*. Hasil dari pengujian tersebut dapat dilihat pada tabel berikut.

**Tabel 4.3** Pengujian *Confusion Matrix* pada skenario 2

		Prediksi	
		Ujaran Kebencian	Bukan Ujaran Kebencian
Aktual	Ujaran Kebencian	1285	178
	Bukan Ujaran Kebencian	348	1189

Dari hasil pengujian menggunakan *Confusion Matrix*, maka didapatkan nilai dari akurasi, *precision*, dan *recall* sebagai berikut.

$$\text{Akurasi} = \frac{1285 + 1189}{1285 + 1189 + 348 + 178} = 0,8247 \times 100 = 82,47\%$$

$$\text{Recall Positif} = \frac{1285}{1285 + 178} = 0,8783 \times 100 = 87,83\%$$

$$\text{Recall Negatif} = \frac{1189}{1189 + 348} = 0,7736 \times 100 = 77,36\%$$

$$\text{Recall Rata-Rata} = \frac{87,83 + 77,36}{2} = 0,8260 \times 100 = 82,60\%$$

$$\text{Precision Positif} = \frac{1285}{1285 + 348} = 0,7869 \times 100 = 78,69\%$$

$$\text{Precision Negatif} = \frac{1189}{1189 + 178} = 0,8698 \times 100 = 86,98\%$$

$$\text{Precision Rata-Rata} = \frac{78,69 + 86,98}{2} = 0,8283 \times 100 = 82,83\%$$

3) Skenario *split data training* dan data *testing* 80:20 dan model tanpa seleksi fitur.

Pada skenario ini, akan dilakukan *split dataset* yang bertujuan untuk membagi dataset menjadi data *training* dan data *testing* menggunakan perbandingan 80:20. Selanjutnya hasil dari *split dataset* tersebut akan digunakan untuk membangun model *Naïve Bayes* tanpa seleksi fitur *Mutual Information*. Kemudian dilakukan pengujian menggunakan *Confusion Matrix* untuk mengetahui nilai akurasi, nilai *precision*, dan nilai *recall*. Hasil dari pengujian tersebut dapat dilihat pada tabel berikut.

**Tabel 4.4** Pengujian *Confusion Matrix* pada skenario 3

		Prediksi	
		Ujaran Kebencian	Bukan Ujaran Kebencian
Aktual	Ujaran Kebencian	888	100
	Bukan Ujaran Kebencian	245	767

Dari hasil pengujian menggunakan *Confusion Matrix*, maka didapatkan nilai dari akurasi, *precision*, dan *recall* sebagai berikut.

$$\text{Akurasi} = \frac{888 + 767}{888 + 767 + 245 + 100} = 0,8275 \times 100 = 82,25\%$$

$$\text{Recall Positif} = \frac{888}{888 + 100} = 0,8988 \times 100 = 89,88\%$$

$$\text{Recall Negatif} = \frac{767}{767 + 245} = 0,7579 \times 100 = 75,79\%$$

$$\text{Recall Rata-Rata} = \frac{89,88 + 75,79}{2} = 0,8283 \times 100 = 82,83\%$$

$$\text{Precision Positif} = \frac{888}{888 + 245} = 0,7838 \times 100 = 78,38\%$$

$$\text{Precision Negatif} = \frac{767}{767 + 100} = 0,8847 \times 100 = 88,47\%$$

$$\text{Precision Rata-Rata} = \frac{78,83 + 88,47}{2} = 0,8342 \times 100 = 83,42\%$$

- 4) Skenario *split* data *training* dan data *testing* 80:20 dan model dengan seleksi fitur.

Pada skenario ini, akan dilakukan *split dataset* yang bertujuan untuk membagi dataset menjadi data *training* dan data *testing* menggunakan perbandingan 80:20. Selanjutnya hasil dari *split dataset* tersebut akan digunakan untuk membangun model *Naïve Bayes* dengan seleksi fitur *Mutual Information* yang menggunakan nilai *threshold* berdasarkan tabel 4.1 yaitu sebesar 0,0001725. Kemudian dilakukan pengujian menggunakan *Confusion Matrix* untuk mengetahui nilai akurasi, nilai *precision*, dan nilai *recall*. Hasil dari pengujian tersebut dapat dilihat pada tabel berikut.

**Tabel 4.5** Pengujian *Confusion Matrix* pada skenario 4

		Prediksi	
		Ujaran Kebencian	Bukan Ujaran Kebencian
Aktual	Ujaran Kebencian	882	106
	Bukan Ujaran Kebencian	226	786

Dari hasil pengujian menggunakan *Confusion Matrix*, maka didapatkan nilai dari akurasi, *precision*, dan *recall* sebagai berikut.

$$\text{Akurasi} = \frac{882 + 786}{882 + 786 + 226 + 106} = 0,834 \times 100 = 83,4\%$$

$$\text{Recall Positif} = \frac{882}{882 + 106} = 0,8927 \times 100 = 89,27\%$$

$$\text{Recall Negatif} = \frac{786}{786 + 226} = 0,7767 \times 100 = 77,67\%$$

$$\text{Recall Rata-Rata} = \frac{89,27 + 77,67}{2} = 0,8347 \times 100 = 83,47\%$$

$$\text{Precision Positif} = \frac{882}{882 + 226} = 0,7960 \times 100 = 79,60\%$$

$$\text{Precision Negatif} = \frac{786}{786 + 106} = 0,8812 \times 100 = 88,12\%$$

$$\text{Precision Rata-Rata} = \frac{79,60 + 88,12}{2} = 0,8386 \times 100 = 83,86\%$$

- 5) Skenario *split* data *training* dan data *testing* 90:10 dan model tanpa seleksi fitur.

Pada skenario ini, akan dilakukan *split dataset* yang bertujuan untuk membagi dataset menjadi data *training* dan data *testing* menggunakan perbandingan 90:10. Selanjutnya hasil dari *split dataset* tersebut akan digunakan untuk membangun model *Naïve Bayes* tanpa seleksi fitur *Mutual Information*. Kemudian dilakukan pengujian menggunakan *Confusion Matrix* untuk mengetahui nilai akurasi, nilai *precision*, dan nilai *recall*. Hasil dari pengujian tersebut dapat dilihat pada tabel berikut.

**Tabel 4.6** Pengujian *Confusion Matrix* pada skenario 5

		Prediksi	
		Ujaran Kebencian	Bukan Ujaran Kebencian
Aktual	Ujaran Kebencian	453	55
	Bukan Ujaran Kebencian	115	377

Dari hasil pengujian menggunakan *Confusion Matrix*, maka didapatkan nilai dari akurasi, *precision*, dan *recall* sebagai berikut.

$$\text{Akurasi} = \frac{453 + 377}{453 + 377 + 115 + 55} = 0,83 \times 100 = 83\%$$

$$\text{Recall Positif} = \frac{453}{453 + 55} = 0,8917 \times 100 = 89,17\%$$

$$\text{Recall Negatif} = \frac{337}{337 + 115} = 0,7663 \times 100 = 76,63\%$$

$$\text{Recall Rata-Rata} = \frac{89,17 + 76,63}{2} = 0,8290 \times 100 = 82,90\%$$

$$\text{Precision Positif} = \frac{453}{453 + 115} = 0,7975 \times 100 = 79,75\%$$

$$\text{Precision Negatif} = \frac{377}{377 + 55} = 0,8727 \times 100 = 87,27\%$$

$$\text{Precision Rata-Rata} = \frac{79,75 + 87,27}{2} = 0,8351 \times 100 = 83,51\%$$

- 6) Skenario *split* data *training* dan data *testing* 90:10 dan model dengan seleksi fitur.

Pada skenario ini, akan dilakukan *split dataset* yang bertujuan untuk membagi dataset menjadi data *training* dan data *testing* menggunakan perbandingan 90:10. Selanjutnya hasil dari *split dataset* tersebut akan digunakan untuk membangun model *Naive Bayes* dengan seleksi fitur *Mutual Information* yang menggunakan nilai *threshold* berdasarkan tabel 4.1 yaitu sebesar 0,0003076. Kemudian dilakukan pengujian menggunakan *Confusion Matrix* untuk mengetahui nilai akurasi, nilai *precision*, dan nilai *recall*. Hasil dari pengujian tersebut dapat dilihat pada tabel berikut.

**Tabel 4.7** Pengujian *Confusion Matrix* pada skenario 6

		Prediksi	
		Ujaran Kebencian	Bukan Ujaran Kebencian
Aktual	Ujaran Kebencian	442	66
	Bukan Ujaran Kebencian	90	402

Dari hasil pengujian menggunakan *Confusion Matrix*, maka didapatkan nilai dari akurasi, *precision*, dan *recall* sebagai berikut.

$$\text{Akurasi} = \frac{442 + 402}{442 + 402 + 90 + 66} = 0,844 \times 100 = 84,4\%$$

$$\text{Recall Positif} = \frac{442}{442 + 66} = 0,8701 \times 100 = 87,01\%$$

$$\text{Recall Negatif} = \frac{402}{402 + 90} = 0,8171 \times 100 = 81,71\%$$

$$\text{Recall Rata-Rata} = \frac{87,01 + 81,71}{2} = 0,8436 \times 100 = 84,36\%$$

$$\text{Precision Positif} = \frac{442}{442 + 90} = 0,8308 \times 100 = 83,08\%$$

$$\textit{Precision Negatif} = \frac{402}{402 + 66} = 0,8590 \times 100 = 85,90\%$$

$$\textit{Precision Rata-Rata} = \frac{84,36 + 83,08}{2} = 0,8449 \times 100 = 84,49\%$$

Berdasarkan hasil pengujian menggunakan *Confusion Matrix* dari setiap skenario, maka didapatkan nilai akurasi, *precision*, dan *recall* sebagai berikut.

**Tabel 4.8** Hasil perbandingan skenario *split data training* dan *data testing*

Skenario	Training : Testing	Model Naïve Bayes	Akurasi	Presisi	Recall
1	70:30	Tanpa <i>Mutual Information</i>	81,97%	82,66%	82,14%
2	70:30	Dengan <i>Mutual Information</i>	82,47%	82,60%	82,93%
3	80:20	Tanpa <i>Mutual Information</i>	82,25%	82,83%	83,42%
4	80:20	Dengan <i>Mutual Information</i>	83,4%	83,47%	83,86%
5	90:10	Tanpa <i>Mutual Information</i>	83%	82,90%	83,51%
6	90:10	Dengan <i>Mutual Information</i>	<b>84,4%</b>	84,36%	84,49%

Dari tabel 4.8 didapatkan nilai akurasi tertinggi sebesar 84,4% pada skenario split dataset 90:10 dan model *Naïve Bayes* dengan seleksi fitur *Mutual Information*, sehingga model dari skenario tersebut akan digunakan untuk melakukan klasifikasi teks.

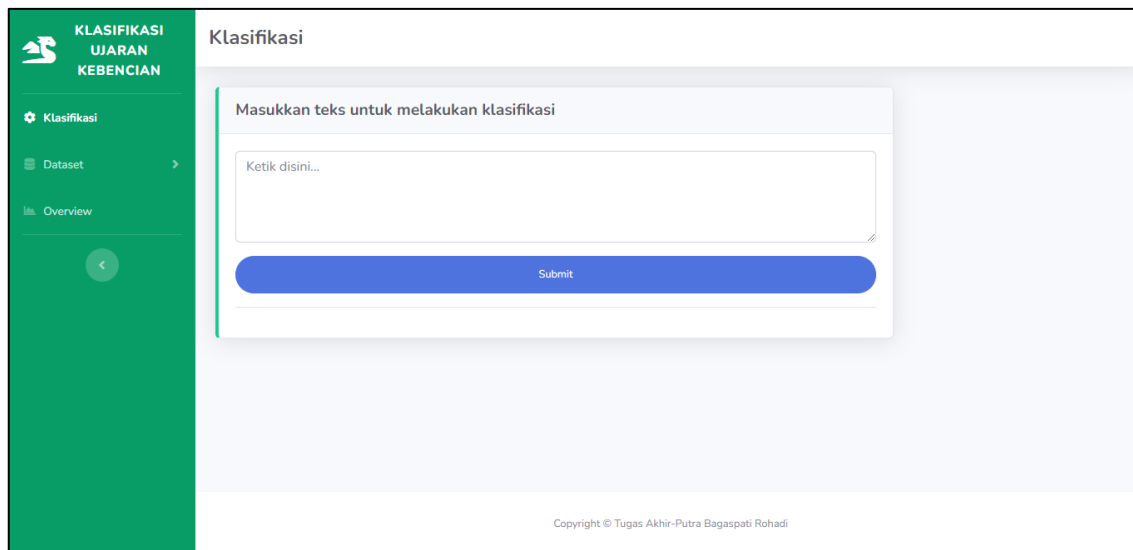
## 4.2. Implementasi Perancangan Antarmuka

Pada bagian ini akan membahas hasil implementasi dari perancangan antarmuka yang sudah dibuat. Perancangan antarmuka yang telah dibuat terdiri dari 3 halaman utama yaitu halaman klasifikasi, halaman *dataset*, dan halaman *overview*. Berikut merupakan tampilan dari setiap halaman yang telah dibuat.

### 4.2.1. Halaman Klasifikasi

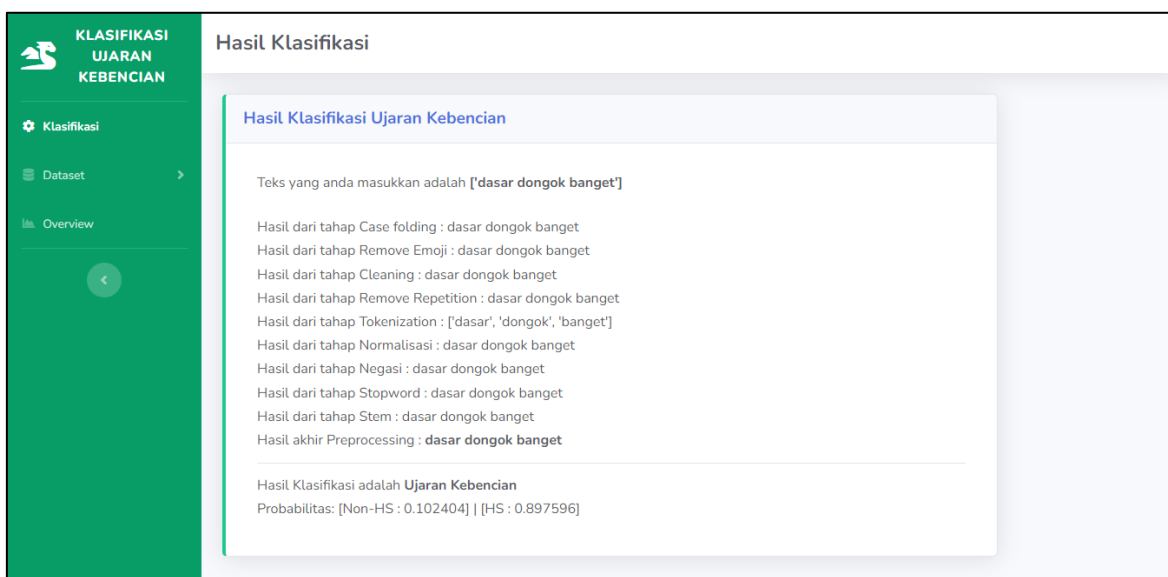
Halaman klasifikasi merupakan tampilan awal ketika aplikasi dijalankan. Halaman ini berguna untuk memasukkan *input* teks yang akan dilakukan klasifikasi. Pada halaman ini terdapat *text box* untuk menuliskan teks dan tombol *submit* untuk memproses teks yang dimasukkan dan mengarahkan ke halaman hasil klasifikasi. Halaman klasifikasi dapat dilihat pada gambar berikut.





**Gambar 4.1** Implementasi Halaman klasifikasi

Setelah memasukkan teks dan menekan tombol *submit*, maka akan menampilkan halaman yang berisikan hasil klasifikasi teks beserta hasil *preprocessing*, nilai probabilitas, dan kelas klasifikasinya. Halaman hasil klasifikasi dapat dilihat pada gambar berikut.



**Gambar 4.2** Implementasi halaman hasil klasifikasi

#### 4.2.2. Halaman Dataset

Halaman *dataset* akan menampilkan *dataset* yang digunakan dan memiliki 4 *sub menu*, yaitu halaman *dataset* untuk menampilkan semua data, halaman *clean dataset* untuk menampilkan semua data hasil *preprocessing*, halaman data *train* untuk menampilkan data *train* yang digunakan, dan halaman data *test* untuk menampilkan data *test* yang digunakan.

No	Teks	Label
1	- disaat semua cowok berusaha melacak perhatian gue. loe lantas remehkan perhatian yg gue kasih khusus ke elo. basic elo cowok bego !!!	1
2	RT USER: USER siapa yang telat ngasih tau elu?edan sarap gue bergaul dengan cigax jifla calis sama siapa noh licew juga'	0
3	41. Kadang aku berfikir, kenapa aku tetap percaya pada Tuhan padahal aku selalu jatuh berkali-kali. Kadang aku merasa Tuhan itu ninggalkan aku sendirian. Ketika orangtuaku berencana berpisah, ketika kakakku lebih memilih jadi Kristen. Ketika aku anak ter	0
4	USER USER AKU ITU AKU/ninKU TAU MATAMU SIPIT TAPI DILIAT DARI MANA ITU AKU'	0

**Gambar 4.3** Implementasi halaman dataset

Pada gambar 4.3 menampilkan total data yang digunakan, total data *training*, total data *testing*, dan seluruh *dataset* yang berisikan teks serta label atau kelas dari setiap data.

No	Teks	Label
1	['saat', 'semua', 'cowok', 'usaha', 'lacak', 'perhati', 'kamu', 'lantas', 'remeh', 'perhati', 'kasih', 'khusus', 'kamu', 'basic', 'kamu', 'cowok', 'bego']	1
2	['siapa', 'lambat', 'kasih', 'tahu', 'kamu', 'gila', 'sarap', 'gaul', 'cigax', 'jifla', 'cal', 'sama', 'siapa', 'noh', 'licew']	0
3	['kadang', 'aku', 'berfikir', 'aku', 'tetap', 'percaya', 'tuhan', 'padahal', 'aku', 'selalu', 'jatuh', 'kali', 'kali', 'kadang', 'aku', 'rasa', 'tuhan', 'ninggalkan', 'aku', 'sendiri', 'orangtuaku', 'rencana', 'pisah', 'kakak', 'lebih', 'pilih', 'jadi', 'kristen', 'aku', 'anak', 'ter']	0
4	['aku', 'aku', 'tahu', 'mata', 'sipit', 'lihat', 'mana', 'aku']	0
5	['kaum', 'cebong', 'kafir', 'lihat', 'dongok', 'awal', 'tambah', 'dongok', 'hahahah']	1
6	['bani', 'taplak', 'kawan']	1
7	['deklarasi', 'pilih', 'kepala', 'daerah', 'aman', 'anti', 'isu', 'bohong', 'warga', 'dukuh', 'sari', 'jabon']	0

**Gambar 4.4** Implementasi halaman *clean dataset*

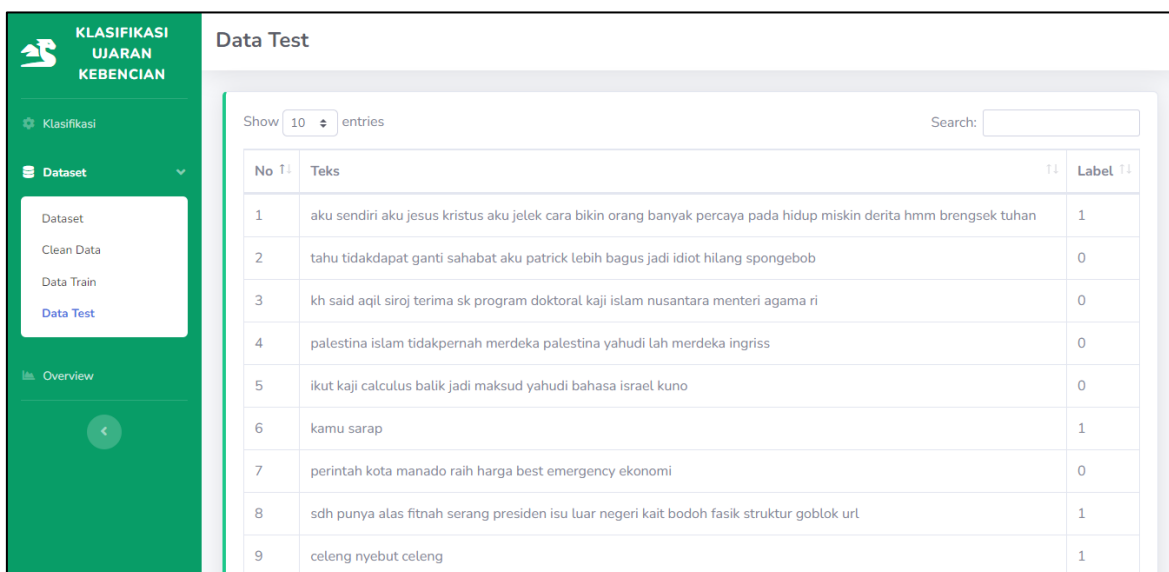
Pada gambar 4.4 menampilkan semua data hasil *preprocessing* serta label atau kelas dari setiap data.



No	Teks	Label
1	tolak dukung anti rejim sekarang kamu sebutin kamu lakuin	1
2	ajak ulama turut sejuk suasana jelang pilih kepala daerah serentak	0
3	lokal brand global bukan baru jadi sejak lama bagai bidang agama indonesia punya hindu nusantara cukup beda segi hindu dunia hindu nusantara buat hindu dunia lemah hindu dunia berkem	0
4	ayo pancing keluar antek komunis nkri selenggara nonton bareng film penghianatan s pki	1
5	sakit panik alami kaum begundal sial bongkrek lama pak prabowo jadi presiden panik terus lanjut spinya pak jokowi periode	1
6	hayo nyok bajing	1
7	jadi begini sekarang inti orang kaya elit jamin tangan mulut didik mending kata udik kampung sampah tp punya otak buat mikir	1

**Gambar 4.5** Implementasi halaman data *training*

Pada gambar 4.5 menampilkan semua data *training* serta label atau kelas dari setiap data yang merupakan hasil dari *split dataset*.



No	Teks	Label
1	aku sendiri aku jesus kristus aku jelek cara bikin orang banyak percaya pada hidup miskin derita hmm brengsek tuhan	1
2	tahu tidakdapat ganti sahabat aku patrick lebih bagus jadi idiot hilang spongebob	0
3	kh said aqil siroj terima sk program doktoral kaji islam nusantara menteri agama ri	0
4	palestina islam tidakpernah merdeka palestina yahudi lah merdeka ingriis	0
5	ikut kaji calcutus balik jadi maksud yahudi bahasa israel kuno	0
6	kamu sarap	1
7	perintah kota manado raih harga best emergency ekonomi	0
8	sdh punya alas fitnah serang presiden isu luar negeri kait bodoh fasik struktur goblok url	1
9	celeng nyebut celeng	1

**Gambar 4.6** Implementasi halaman data *testing*

Pada gambar 4.6 menampilkan semua data *testing* serta label atau kelas dari setiap data yang merupakan hasil dari *split dataset*.

#### 4.2.3. Halaman Overview

Pada halaman *overview* akan menampilkan hasil dari *confusion matrix* dan nilai akurasi, *precision*, serta *recall* dari model yang telah dibangun. Implementasi dari halaman *overview* dapat dilihat pada gambar berikut.

**KLASIFIKASI UJARAN KEBENCIAN**

Overview

Optimasi Naive Bayes Dengan Seleksi Fitur Mutual Information

Tabel Perbandingan Skenario Split Data Training dan Data Testing

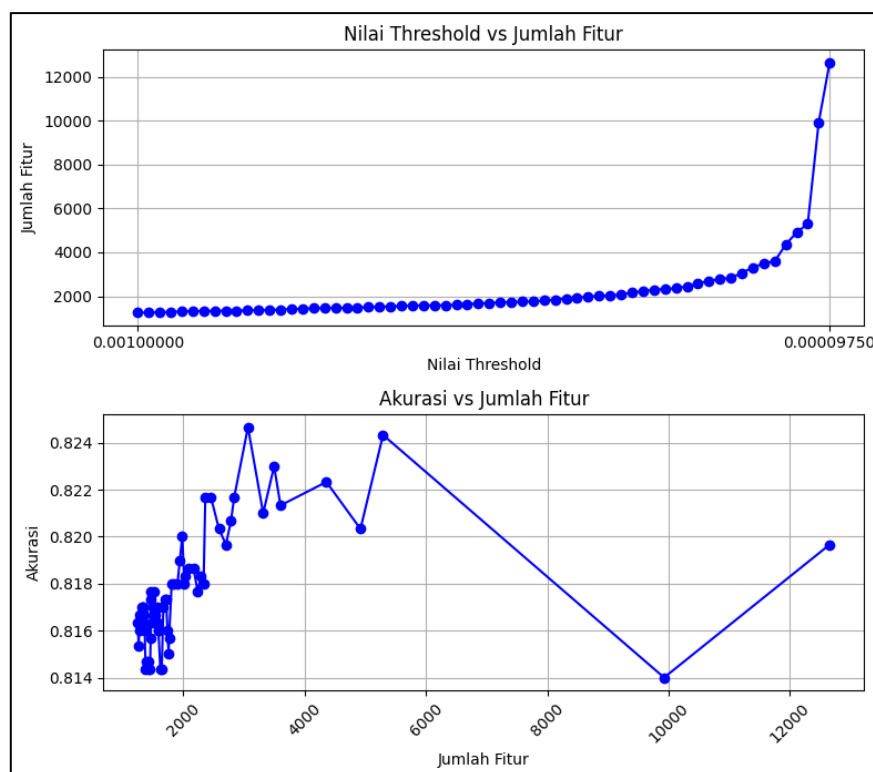
No	Perbandingan Data Training Dan Data Testing	Model Naive Bayes	Akurasi	Presisi	Recall
1	70:30	Tanpa Mutual Information	81,97%	77,25%	89,34%
2	70:30	Dengan Mutual Information	81,4%	75,15%	92,41%
3	80:20	Tanpa Mutual Information	82,25%	78,38%	89,88%
4	80:20	Dengan Mutual Information	83,4%	79,60%	89,27%
5	90:10	Tanpa Mutual Information	83%	79,75%	89,17%
6	90:10	Dengan Mutual Information	83%	80,18%	88,39%

Perbandingan Confusion Matrix

**Gambar 4.7** Implementasi halaman *overview*

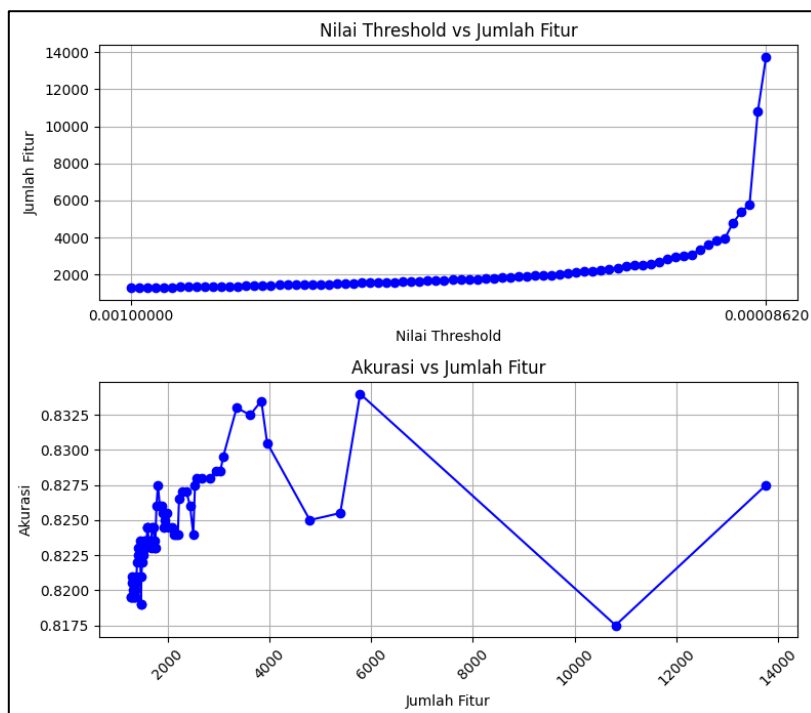
### 4.3. Pembahasan

Dalam melakukan penerapan seleksi fitur *Mutual Information* pada metode *Naive Bayes*, penelitian ini melakukan beberapa percobaan untuk mendapatkan nilai *threshold* dengan akurasi tertinggi. Percobaan ini dilakukan dengan mencari akurasi tertinggi yang bisa didapatkan pada rentang nilai *threshold* 0,001 hingga 0,00001 dengan *step* atau *decrement* sebesar 0,0000001. Percobaan ini dilakukan pada setiap skenario *split dataset* yaitu 70:30, 80:20, dan 90:10. Hasil dari percobaan tersebut dapat dilihat sebagai berikut.



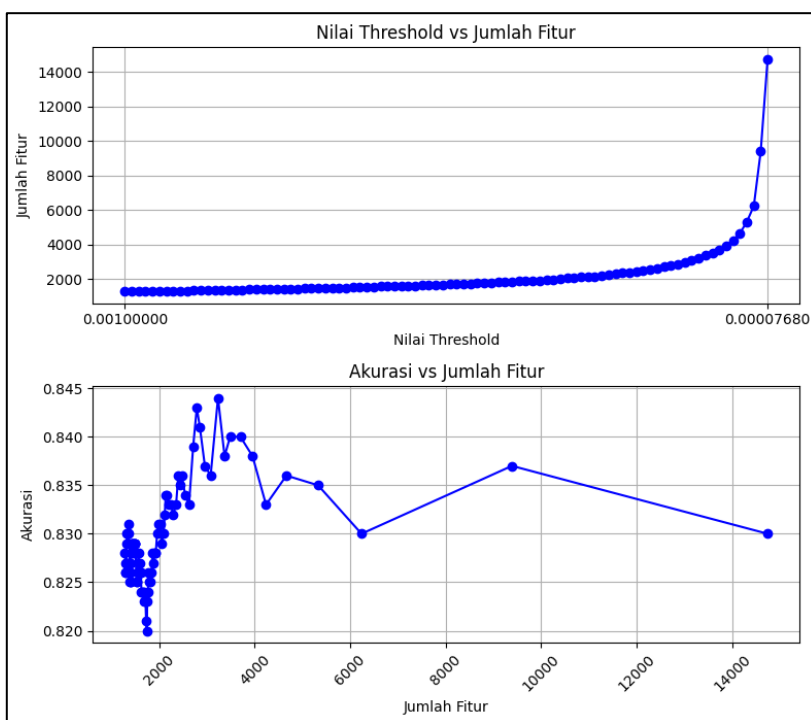
**Gambar 4.8** Grafik perbandingan seleksi fitur pada *split dataset* 70:30

Berdasarkan gambar 4.8 didapatkan akurasi tertinggi sebesar 82,47% dengan nilai *threshold* sebesar 0,0003017 dan pengurangan jumlah fitur sebanyak 9593 fitur dari 12657 fitur menjadi 3064 fitur.



**Gambar 4.9** Grafik perbandingan seleksi fitur pada *split dataset* 80:20

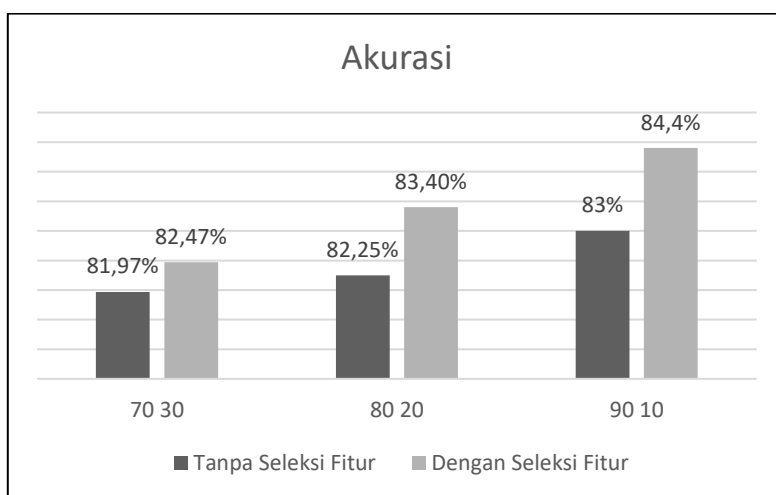
Berdasarkan gambar 4.9 didapatkan akurasi tertinggi sebesar 83,4% dengan nilai *threshold* sebesar 0,0001725 dan pengurangan jumlah fitur sebanyak 7981 fitur dari 13764 fitur menjadi 5783 fitur.



**Gambar 4.10** Grafik perbandingan seleksi fitur pada *split dataset* 90:10

Berdasarkan gambar 4.10 didapatkan akurasi tertinggi sebesar 84,4% dengan nilai *threshold* sebesar 0,0003076 dan pengurangan jumlah fitur sebanyak 11518 fitur dari 14749 fitur menjadi 3231 fitur. Pada skenario ini jugalah didapatkan akurasi tertinggi dari seluruh percobaan yang telah dilakukan.

Berdasarkan hasil pengujian tersebut penggunaan seleksi fitur *Mutual Information* dapat meningkatkan akurasi dari metode *Naïve Bayes* dalam melakukan klasifikasi teks ujaran kebencian. Perbandingan akurasi dari metode *Naïve Bayes* tanpa seleksi fitur dan dengan seleksi fitur serta skenario *split dataset* 70:30, 80:20, dan 90:20 dapat dilihat pada gambar grafik berikut.



**Gambar 4.11** Grafik perbandingan akurasi model

Berdasarkan gambar 4.11 pada skenario *split dataset* 90:10 didapatkan peningkatan akurasi dan merupakan akurasi tertinggi ketika menggunakan seleksi fitur *Mutual Information* yaitu dari 83% menjadi 84,4%. Sementara itu pada skenario *split dataset* 70:30 juga terjadi peningkatan akurasi ketika menggunakan seleksi fitur *Mutual Information* yaitu dari 81,97% menjadi 82,47% dan pada skenario *split dataset* 80:20 terjadi peningkatan akurasi setelah menggunakan seleksi fitur *Mutual Information* dimana akurasi yang didapatkan dari 82,25% menjadi 83,4.

Penerapan seleksi fitur *Mutual Information* juga mampu mengurangi jumlah fitur yang digunakan dalam pembuatan model. Hal ini berdasarkan hasil yang dapat dilihat pada gambar 4.8, gambar 4.9, dan gambar 4.10. Sehingga kelemahan dari metode *Naïve Bayes* yang mengasumsikan setiap fitur bersifat independen dapat diatasi dengan pengurangan jumlah fitur menggunakan seleksi fitur *Mutual Information* dimana akurasi tertinggi yang didapatkan sebesar 84,4%.

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1. Kesimpulan**

Berdasarkan hasil pengujian, dapat disimpulkan bahwa seleksi fitur *Mutual Information* memiliki kemampuan untuk mengatasi kelemahan dari metode *Naïve Bayes* yaitu mengasumsikan setiap fitur bersifat independen dan mendapatkan peningkatan akurasi sebesar 1,4% dari 83% menjadi 84,4% serta nilai akurasi tersebut merupakan nilai akurasi tertinggi yang didapatkan dari penelitian ini. Penelitian ini juga mampu melakukan klasifikasi teks ujaran kebencian menggunakan metode *Naïve Bayes* dengan seleksi fitur *Mutual Information*.

#### **5.2. Saran**

Adapun saran yang dapat digunakan pada penelitian selanjutnya untuk mengembangkan penelitian ini adalah sebagai berikut :

1. Menggunakan persentase dalam memilih fitur yang akan digunakan, sehingga dapat mengurai waktu dan *resources* yang digunakan.
2. Menggunakan metode seleksi fitur lain untuk mengetahui metode mana yang lebih optimal dalam mengatasi kelemahan metode *Naïve Bayes*.
3. Menggunakan metode *machine learning* atau *deep learning* lain untuk membandingkan seberapa efektif seleksi fitur menggunakan metode *Mutual Information*.

## DAFTAR PUSTAKA

- Agusta, L. (2009). *Perbandingan Algoritma Stemming Porter Dengan Algoritma Nazief & Andriani Untuk Stemming Dokumen Teks Bahasa Indonesia. Konferensi Nasional Sistem dan Informatika. Bali: KNSI.* 196–201.
- Ali Fauzi, M., Arifin, A. Z., Gosaria, S. C., & Prabowo, I. S. (2017). Indonesian news classification using naïve bayes and two-phase feature selection model. *Indonesian Journal of Electrical Engineering and Computer Science*, 8(3), 610–615. <https://doi.org/10.11591/ijeecs.v8.i3.pp610-615>
- Ali, M. (2017). *An Accurate CT Saturation Classification Using a Fine-Tuning Strategy. Dc.* <https://doi.org/10.3390/en10111830>
- Andono, P. N., T.Sutojo, & Muljono. (2017). *Pengolahan Citra Digital.*
- Antariksa, K., Purnomo WP, Y. S., & Ernawati, E. (2019). Klasifikasi Ujaran Kebencian pada Cuitan dalam Bahasa Indonesia. *Jurnal Buana Informatika*, 10(2) Antariksa, K., Purnomo WP, Y. S., Ernawati, E. (2019). Klasifikasi Ujaran Kebencian pada Cuitan dalam Bahasa Indonesia. *Jurnal Buana Informatika*, 10(2), 164. <https://doi.org/10.24002/jbi.v10i2.2451>, 164. <https://doi.org/10.24002/jbi.v10i2.2451>
- Anugerah, F., & Djunaidy, A. (2017). *Improving the Performance of Repeated Character Preprocessing in Recognizing Words in the Indonesian Sentiment Classification.* 7(9), 1–9.
- Birjali, M., Kasri, M., & Beni-Hssane, A. (2021). A comprehensive survey on sentiment analysis: Approaches, challenges and trends. *Knowledge-Based Systems*, 226, 107134. <https://doi.org/10.1016/j.knosys.2021.107134>
- Davidson, T., Warmsley, D., Macy, M., & Weber, I. (2017). Automated hate speech detection and the problem of offensive language. *Proceedings of the 11th International Conference on Web and Social Media, ICWSM 2017*, 512–515.
- Devita, R. N., Herwanto, H. W., & Wibawa, A. P. (2018). Perbandingan Kinerja Metode Naive Bayes dan K-Nearest Neighbor untuk Klasifikasi Artikel Berbahasa Indonesia. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 5(4), 427. <https://doi.org/10.25126/jtiik.201854773>
- Hegazi, M. O., Al-Dossari, Y., Al-Yahy, A., Al-Sumari, A., & Hilal, A. (2021). Preprocessing Arabic text on social media. *Heliyon*, 7(2), e06191. <https://doi.org/10.1016/j.heliyon.2021.e06191>
- Hendy Evan, F., & Sigit Purnomo, Y. W. (2014). Pembangunan Perangkat Lunak Peringkat Dokumen dari Banyak Sumber Menggunakan Sentence Scoring dengan Metode TF-IDF. *Seminar Nasional Aplikasi Teknologi Informasi (SNATI) Yogyakarta*, G17–G22.
- Hermanto, H., Mustopa, A., & Kuntoro, A. Y. (2020). Algoritma Klasifikasi Naive Bayes Dan Support Vector Machine Dalam Layanan Komplain Mahasiswa. *JITK (Jurnal Ilmu Pengetahuan Dan Teknologi Komputer)*, 5(2), 211–220. <https://doi.org/10.33480/jitk.v5i2.1181>
- Ibrohim, M. O., & Budi, I. (2019). Multi-label Hate Speech and Abusive Language Detection in Indonesian Twitter. *Proceedings of the Third Workshop on Abusive*



*Language Online*, 46–57. <https://doi.org/10.18653/v1/w19-3506>

- Irham, L. G., Adiwijaya, A., & Wisesty, U. N. (2019). Klasifikasi Berita Bahasa Indonesia Menggunakan Mutual Information dan Support Vector Machine. *Jurnal Media Informatika Budidarma*, 3(4), 284. <https://doi.org/10.30865/mib.v3i4.1410>
- Irsad, A. E., Sari, Y. A., & Fauzi, M. A. (2019). *Seleksi Fitur Information Gain untuk Klasifikasi Informasi Tempat Tinggal di Kota Malang Berdasarkan Tweet Menggunakan Metode Naïve Bayes dan Pembobotan TF-IDF-CF*. 3(5), 4907–4913.
- Kowsari, K., Meimandi, K. J., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text classification algorithms: A survey. In *Information (Switzerland)* (Vol. 10, Issue 4). <https://doi.org/10.3390/info10040150>
- Kurnia, R. I. (2020). Classification of User Comment Using Word2vec and SVM Classifier. *International Journal of Advanced Trends in Computer Science and Engineering*, 9(1), 643–648. <https://doi.org/10.30534/ijatcse/2020/90912020>
- Maylawati, D. S., Zulfikar, W. B., Slamet, C., Ramdhani, M. A., & Gerhana, Y. A. (2019). An Improved of Stemming Algorithm for Mining Indonesian Text with Slang on Social Media. *2018 6th International Conference on Cyber and IT Service Management, CITSM 2018, Citsm*. <https://doi.org/10.1109/CITSM.2018.8674054>
- Munitasri, I., Santosa, S., & Supriyanto, C. (2018). Klasifikasi Pesan Sms Menggunakan Algoritma Naive Bayes Dengan Seleksi Fitur Genetic Algorithm. *Jurnal Teknologi Informasi*, 14.
- Nugroho, A. (2018). Analisis Sentimen Pada Media Sosial Twitter Menggunakan Naive Bayes Classifier Dengan Ekstraksi Fitur N-Gram. *J-SAKTI (Jurnal Sains Komputer Dan Informatika)*, 2(2), 200. <https://doi.org/10.30645/j-sakti.v2i2.83>
- Nurfikri, F. S., Mubarak, M. S., & Adiwijaya. (2018). News topic classification using mutual information and Bayesian network. *2018 6th International Conference on Information and Communication Technology, ICoICT 2018*, 0(c), 162–166. <https://doi.org/10.1109/ICoICT.2018.8528806>
- Prakoso, B. S., Rosiyadi, D., Utama, H. S., & Aridarma, D. (2019). Klasifikasi Berita Menggunakan Algoritma Naive Bayes Classifier Dengan Seleksi Fitur Dan Boosting. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 3(2), 227–232. <https://doi.org/10.29207/resti.v3i2.1042>
- Rahat, A. M., Kahir, A., & Masum, A. K. M. (2020). Comparison of Naive Bayes and SVM Algorithm based on Sentiment Analysis Using Review Dataset. *Proceedings of the 2019 8th International Conference on System Modeling and Advancement in Research Trends, SMART 2019*, 266–270. <https://doi.org/10.1109/SMART46866.2019.9117512>
- Rahman, A., & Hossen, M. S. (2019). Sentiment Analysis on Movie Review Data Using Machine Learning Approach. *2019 International Conference on Bangla Speech and Language Processing, ICBSLP 2019*, 27–28. <https://doi.org/10.1109/ICBSLP47725.2019.201470>
- Rini, D. C., Farida, Y., & Puspitasari, D. (2016). Klasifikasi Menggunakan Metode Hybrid Bayesian-Neural Network (Studi Kasus: Identifikasi Virus Komputer). *Jurnal Matematika "MANTIK,"* 1(2), 38. <https://doi.org/10.15642/mantik.2016.1.2.38-43>

- Santoso, J., Soetiono, A. D. B., Gunawan, G., Setyati, E., Yuniarno, E. M., Hariadi, M., & Purnomo, M. H. (2018). Self-Training Naive Bayes Berbasis Word2Vec untuk Kategorisasi Berita Bahasa Indonesia. *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi (JNTETI)*, 7(2), 158–166. <https://doi.org/10.22146/jnteti.v7i2.418>
- Sartika, D., & Sensuse, D. I. (2017). Perbandingan Algoritma Klasifikasi Naive Bayes, Nearest Neighbour, dan Decision Tree pada Studi Kasus Pengambilan Keputusan Pemilihan Pola Pakaian. *Jatisi*, 1(2), 151–161.
- Septian, J. A., Fachrudin, T. M., & Nugroho, A. (2019). Analisis Sentimen Pengguna Twitter Terhadap Polemik Persepakbolaan Indonesia Menggunakan Pembobotan TF-IDF dan K-Nearest Neighbor. *Journal of Intelligent System and Computation*, 1(1), 43–49. <https://doi.org/10.52985/insyst.v1i1.36>
- Sharma, S., Agrawal, S., & Shrivastava, M. (2018). Degree based Classification of Harmful Speech using Twitter Data. *COLING 2018 - 1st Workshop on Trolling, Aggression and Cyberbullying, TRAC 2018 - Proceedings of the Workshop*, 106–112.
- Sya'bani, T. N., Fitriawati, C., Ratnasari, D. Y., & Habibah, S. M. (2021). Responses to the Practice of Hate Speech at the Pancasila Youth Organization in Sidoarjo. *JED (Jurnal Etika Demokrasi)*, 6(2), 292–298. <https://doi.org/10.26618/jed.v6i2.4515>
- Syafar, D. N. (2016). Negasi Dalam Bahasa Indonesia Dan Bahasa Inggris. *Jurnal Arbitrer*, 3(1), 1. <https://doi.org/10.25077/ar.3.1.1-11.2016>
- Tarecha, R. I., Wahyudi, F., & Jannah, U. M. (2022). Penanganan Negasi dalam Analisa Sentimen Bahasa Indonesia. *Jurnal Sistem Informasi Dan Informatika (JUSIFOR)*, 1(1), 51–58. <https://doi.org/10.33379/jusifor.v1i1.1276>
- Thangaraj, M., & Sivakami, M. (2018). Text classification techniques: A literature review. *Interdisciplinary Journal of Information, Knowledge, and Management*, 13. <https://doi.org/10.28945/4066>
- Ulfa, M. A., Irmawati, B., & Husodo, A. Y. (2018). Twitter Sentiment Analysis using Naïve Bayes Classifier with Mutual Information Feature Selection. *Journal of Computer Science and Informatics Engineering (J-Cosine)*, 2(2), 106–111. <https://doi.org/10.29303/jcosine.v2i2.120>
- VijayGaikwad, S., Chaugule, A., & Patil, P. (2014). Text Mining Methods and Techniques. *International Journal of Computer Applications*, 85(17), 42–45. <https://doi.org/10.5120/14937-3507>
- Wati, R. (2020). Penerapan Algoritma Naive Bayes Dan Particle Swarm Optimization Untuk Klasifikasi Berita Hoax Pada Media Sosial. *JITK (Jurnal Ilmu Pengetahuan Dan Teknologi Komputer)*, 5(2), 159–164. <https://doi.org/10.33480/jitk.v5i2.1034>
- Wibisono, A. B., & Fahrurrozi, A. (2019). Perbandingan Algoritma Klasifikasi Dalam Pengklasifikasian Data Penyakit Jantung Koroner. *Jurnal Ilmiah Teknologi Dan Rekayasa*, 24(3), 161–170. <https://doi.org/10.35760/tr.2019.v24i3.2393>
- Widayati, L. S. (2018). Ujaran Kebencian: Batasan Pengertian Dan Larangannya. *Pusat Penelitian Badan Keahlian DPR RI*.
- Zareapoor, M., & K. R, S. (2015). Feature Extraction or Feature Selection for Text Classification: A Case Study on Phishing Email Detection. *International Journal of*

*Information Engineering and Electronic Business*, 7(2), 60–65.  
<https://doi.org/10.5815/ijieeb.2015.02.08>

Zulfikar, W. B., & Lukman, N. (2016). Perbandingan Naive Bayes Classifier Dengan Nearest Neighbor Untuk Identifikasi Penyakit Mata. *Jurnal Online Informatika*, 1(2), 82–86. <https://doi.org/10.15575/join.v1i2.33>