

**PENERAPAN NAÏVE BAYES CLASSIFIER DENGAN ALGORITMA  
STEMMING NAZIEF DAN ADRIANI UNTUK APLIKASI  
DETEKSI UJARAN KEBENCIAN  
BERBASIS WEB**

**Alvin Pratama<sup>1</sup>, Dwi Marisa Midyanti<sup>2</sup>, Syamsul Bahri<sup>3</sup>**

<sup>123</sup>Jurusan Rekayasa Sistem Komputer, Fakultas MIPA Universitas Tanjungpura

Jalan Prof. Dr. H. Hadari Nawawi Pontianak

Telp./Fax : (0561) 577963

e-mail: <sup>1</sup>alvinpratama@student.untan.ac.id, <sup>2</sup>dwi.marisa@siskom.untan.ac.id,

<sup>3</sup>syamsul.bahri@siskom.untan.ac.id.

**ABSTRAK**

Pemahaman diperlukan dalam menentukan suatu pernyataan termasuk sebagai ujaran kebencian atau bukan. Ketika seseorang mencoba memahami maksud suatu pernyataan, orang tersebut bisa melakukan kesalahan dan merasa ragu dalam penentuan suatu pernyataan termasuk sebagai ujaran kebencian atau bukan. Untuk mempermudah proses penentuan pernyataan termasuk sebagai ujaran kebencian atau bukan, diperlukan aplikasi yang akurat dalam menentukan ujaran kebencian. Akan tetapi, untuk membangun aplikasi yang akurat, diperlukan metode dengan keakurasian tinggi dalam melakukan pendeteksian ujaran kebencian. Dalam penelitian ini digunakan metode *naïve bayes classifier* karena metode tersebut memiliki kelebihan dalam segi keakurasian. Tujuan dari adanya penelitian adalah untuk mengetahui nilai akurasi, *precision* dan *recall* serta kata yang banyak digunakan dalam ujaran kebencian. Dengan menerapkan algoritma *stemming nazief* dan *adriani* serta *naïve bayes classifier*, aplikasi deteksi ujaran kebencian memperoleh nilai akurasi sebesar 68%, nilai *precision* sebesar 93% dan nilai *recall* sebesar 69% dari pengujian 183 data uji. Selain mengetahui nilai akurasi, *precision* dan *recall*, diketahui juga kata yang banyak digunakan pada 104 data latih berkategori ujaran kebencian merupakan kata benci dengan jumlah 128 kata.

**Kata Kunci** : ujaran kebencian, *naïve bayes classifier*, *stemming nazief* dan *adriani*

**1. PENDAHULUAN**

Pernyataan berisi ujaran kebencian dapat menyebabkan konflik yang terjadi karena pembaca terprovokasi atau terhina saat membaca pernyataan berisi ujaran kebencian. Ketika banyak pembaca pernyataan berisi ujaran kebencian yang terprovokasi atau terhina, maka konflik akan menjadi lebih buruk. Untuk mencegah terjadinya konflik yang disebabkan oleh pembuat atau penyebar ujaran kebencian, diperlukan hukum yang mengatur tentang ujaran kebencian. Hukum tentang ujaran kebencian diatur dalam Kitab Undang Undang Hukum Pidana (KUHP) pada pasal 155 dan 157 [1], serta dalam Undang-Undang Informasi dan Transaksi Elektronik (UU ITE) nomor 19 tahun 2016 pada pasal 28 ayat (2) dan pasal 45A ayat (2) [2], namun masih banyak masyarakat yang membuat atau menyebarkan ujaran kebencian.

Untuk menentukan suatu pernyataan termasuk sebagai ujaran kebencian atau bukan,

masyarakat perlu pemahaman mengenai pernyataan seperti apa yang termasuk sebagai ujaran kebencian. Ketika seseorang mencoba memahami maksud suatu pernyataan, orang tersebut bisa melakukan kesalahan dan merasa ragu dalam penentuan suatu pernyataan termasuk sebagai ujaran kebencian atau bukan. Untuk mempermudah proses penentuan pernyataan, diperlukan aplikasi yang akurat dalam menentukan ujaran kebencian. Akan tetapi, untuk membangun aplikasi yang akurat, diperlukan metode dengan keakurasian tinggi dalam melakukan pendeteksian ujaran kebencian.

Terdapat penelitian yang menguji keakurasian suatu metode dalam mendeteksi ujaran kebencian. Penelitian tersebut merupakan penelitian dari Alfi Ramdhani pada tahun 2018 dengan judul Studi Algoritma *Linear Support Vector Machine* pada Deteksi Ujaran Kebencian Berbahasa Indonesia. Pada penelitian Alfi Ramdhani, akurasi yang

diperoleh sebesar 86.55% dengan menggunakan algoritma *Linear Support Vector Machine*, *feature Word N* dan *10-fold cross validation* [3].

Terdapat juga penelitian lain yang menguji keakurasian suatu metode, tapi bukan dalam mendeteksi ujaran kebencian melainkan mendeteksi jenis email termasuk *SPAM* atau *HAM*. Penelitian tersebut merupakan penelitian dari Harry Pribadi Fitriani dkk pada tahun 2018 mengenai Implementasi Metode *Naïve Bayes Classifier* untuk Aplikasi *Filtering Email SPAM* dengan *Lemmatization* Berbasis Web. Berbeda dengan penelitian Alfi Ramdhani yang menguji keakurasian algoritma *Linear Support Vector Machine* untuk mendeteksi ujaran kebencian, pada penelitian Harry Pribadi Fitriani dkk dilakukan pengujian keakurasian metode *naïve bayes classifier* dalam mendeteksi jenis email termasuk *SPAM* atau *HAM*. Nilai akurasi metode *Naïve Bayes Classifier* yang diperoleh pada penelitian Harry Pribadi Fitriani dkk sebesar 90.83% [4].

Berdasarkan penelitian dari Alfi Ramdhani serta Harry Pribadi Fitriani dkk, maka dibuatlah aplikasi untuk mendeteksi ujaran kebencian dengan menerapkan *naïve bayes classifier*. Pada penelitian Harry Pribadi Fitriani dkk, diterapkan juga *Lemmatization* yang digunakan untuk mengubah kata berimbuhan menjadi kata dasar. Selain *Lemmatization* terdapat juga algoritma lain yang dapat digunakan untuk mengubah kata berimbuhan menjadi kata dasar, yaitu algoritma *stemming*. Terdapat beberapa jenis algoritma *stemming*. Di antara beberapa jenis algoritma *stemming*, algoritma *stemming nazief* dan *adriani* memiliki keakurasian yang tinggi. Keakurasian algoritma *stemming nazief* dan *adriani* terbukti dari penelitian Manase Sahat H Simarankir pada tahun 2017 mengenai Studi Perbandingan Algoritma - Algoritma *Stemming* untuk Dokumen Teks Bahasa Indonesia. Pada penelitian Manase Sahat H Simarankir, ditemukan nilai akurasi algoritma *stemming nazief* dan *adriani* sebesar 97,931%, nilai akurasi algoritma *stemming Arifin* dan *Setiono* sebesar 92,099%, nilai akurasi algoritma *stemming vega* sebesar 63,486% dan nilai akurasi algoritma *stemming tala* sebesar 78,274% [5].

Pada penelitian ini, aplikasi deteksi ujaran kebencian bukan hanya dibuat dengan menerapkan *naïve bayes classifier*, tetapi juga

menerapkan algoritma *stemming nazief* dan *adriani*. Aplikasi menerapkan metode *naïve bayes classifier* untuk mendeteksi ujaran kebencian dan algoritma *stemming nazief* dan *adriani* untuk mengubah kata berimbuhan menjadi kata dasar. Bukan hanya *stemming nazief* dan *adriani* yang digunakan dalam penelitian ini, terdapat juga *text preprocessing* lain seperti *case folding*, konversi kata, perbaikan kata, pemisahan kata, *tokenizing* dan *filtering*. Aplikasi dibuat untuk mendeteksi ujaran kebencian dalam isi pernyataan. Aplikasi yang berbasis web dapat langsung diakses dengan browser tanpa perlu *install*.

## 2. LANDASAN TEORI

### 2.1 *Naïve Bayes Classifier*

Thomas Bayes memperkenalkan metode *Naïve Bayes Classifier* sejak tahun 1702-1761. *Naïve Bayes Classifier* berguna sebagai metode pembelajaran probabilistik dalam mencari nilai probabilitas tertinggi untuk mengklasifikasi data uji pada kategori yang tepat. Teorema peluang bersyarat merupakan konsep dasar yang digunakan oleh Bayes, teoremanya seperti pada Persamaan 1.

$$P(A|B) = P(A)P(B|A) \quad (1)$$

Keterangan :

$P(A|B)$  : Peluang kejadian A bersyarat B.

$P(A)$  : Peluang A.

$P(B|A)$  : Peluang B bersyarat A.

Pada penelitian Rahayu, data uji yang digunakan berupa dokumen. Dalam algoritma *naïve bayes classifier*, setiap dokumen direpresentasikan dengan pasangan atribut " $A_1, A_2, A_3 \dots A_n$ " dimana  $A_1$  merupakan kata pertama,  $A_2$  merupakan kata kedua dan seterusnya. Saat klasifikasi, algoritma akan melakukan pencarian probabilitas tertinggi dari semua kategori dokumen yang diujikan (T). Pencarian probabilitas tertinggi dilakukan dengan menggunakan Persamaan 2.

$$T = \frac{P(A_1, A_2, A_3 \dots A_n | V_j) P(V_j)}{P(A_1, A_2, A_3 \dots A_n)} \quad (2)$$

Keterangan :

$V_j$  : kategori kata  $j = 1, 2, \dots, n$  dimana:

$j_1$  : kategori kata negatif mengandung ujaran kebencian.

$j_2$  : kategori kata positif tidak mengandung ujaran kebencian.

$P(A_i | V_j)$  : Probabilitas  $A_i$  pada kategori  $V_j$ .

$P(V_j)$  : Probabilitas dari  $V_j$ .

$P(A_1, A_2, A_3 \dots A_n)$  bernilai konstan untuk semua kategori ( $V_j$ ).  $P(V_j)$  dan  $P(A_i | V_j)$

dihitung pada saat pelatihan dengan menggunakan Persamaan 3 dan 4.

$$P(V_j) = \frac{|docs\ j|}{|contoh|} \quad (3)$$

Keterangan :

|docs j| : jumlah dokumen setiap kategori j.

|contoh| : jumlah dokumen dari semua kategori.

$$P(A_i|V_j) = \frac{nk+1}{n+|kosakata|} \quad (4)$$

Keterangan :

nk : jumlah frekuensi kemunculan setiap kata.

n : jumlah frekuensi kemunculan kata dari setiap kategori.

|kosakata| : jumlah kata dari semua kategori yang ada pada data latih.

Pada Persamaan 4 terdapat penambahan angka 1 pada pembilang untuk mengantisipasi jika terdapat kata pada dokumen uji yang bernilai nol karena tidak terdapat pada dokumen latih [6].

## 2.2 Stemming Nazief dan Adriani

Algoritma *Stemming Nazief* dan *Adriani* adalah algoritma yang berguna untuk mengubah kata berimbuhan menjadi kata dasar dalam bahasa Indonesia. Algoritma *Stemming Nazief* dan *Adriani* mempunyai aturan imbuhan sendiri dengan model seperti :

[[[AW+][AW+][AW+][Kata Dasar  
[[+AK][+KK][+P]]]

AW : Awalan

AK : Akhiran

KK : Kata Ganti Kepunyaan

P : Partikel

Imbuhan yang ada pada *stemming nazief* dan *adriani* dibagi menjadi [7]:

- a. *Inflection Suffixes*. *Inflection Suffixes* dibagi menjadi dua, yaitu:
  1. *Particles* (Partikel). Imbuhan yang termasuk sebagai *particles* adalah ‘-lah’, ‘-kah’, ‘-tah’ dan ‘-pun’.
  2. *Passive Pronouns* (Kata ganti kepunyaan). Imbuhan yang termasuk sebagai *Passive Pronouns* adalah ‘-ku’, ‘-mu’ dan ‘-nya’.
- b. *Derivation Suffixes* (Akhiran). Imbuhan yang termasuk sebagai *Derivation Suffixes* adalah ‘-i’, ‘-an’ dan ‘-kan’.
- c. *Derivation Prefixes* (Awalan). Imbuhan yang termasuk sebagai *Derivation Prefixes* adalah ‘di-’, ‘ke-’, ‘se-’, ‘te-’, ‘me-’, ‘be-’ dan ‘pe-’.

Algoritma *Stemming Nazief* dan *Adriani* memiliki aturan pemenggalan. Aturan pemenggalan diperlihatkan pada Tabel 1.

Tabel 1. Aturan Pemenggalan [7].

Aturan	Format Kata	Pemenggalan
1	berV...	ber-V...   be-rV...
2	berCAP...	ber-CAP... dimana C!= 'r' & P!='er'
3	berCAerV...	ber-CAerV... dimana C!= 'r'
4	belajar...	bel-ajar...
5	beC <sub>1</sub> erC...	be-C <sub>1</sub> erC... dimana C <sub>1</sub> !={ 'r' 'l' }
6	terV...	ter-V...   te-rV...
7	terCP...	ter-CP... dimana C!= 'r' & P!='er'
8	terCer...	ter-Cer... dimana C!= 'r'
9	teC <sub>1</sub> erC <sub>2</sub> ...	te-C <sub>1</sub> erC <sub>2</sub> ... dimana C <sub>1</sub> != 'r'
10	me {l r w y}V...	me- {l r w y}V...
11	mem {b f v}...	mem- {b f v}...
12	mempe...	mem-pe...
13	mem {rV V}...	me-m {rV V}...   me-p {rV V}...
14	men {c d j z}...	men- {c d j z}...
15	menV...	me-nV...   me-tV...
16	meng {g h q k} ...	meng- {g h q k}...
17	mengV...	meng-V...   meng-kV...
18	menyV...	meny-sV...
19	mempV...	mem-pV... dimana V!= 'e'
20	pe {w y}V...	pe- {w y}V...
21	perV...	per-V...   pe-rV...
22	perCAP...	per-CAP... dimana C!= 'r' & P!='er'
23	perCAerV...	per-CAerV... dimana C!= 'r'
24	pem {b f v}...	pem- {b f v}...
25	pem {rV V}...	pe-m {rV V}...   pe-p {rV V}...
26	pen {c d j z}...	pen- {c d j z}...
27	penV...	pe-nV...   pe-tV...
28	peng {g h q}...	peng- {g h q}...
29	pengV...	peng-V...   peng-kV...
30	penyV...	peny-sV...
31	pelV...	pe-IV... kecuali "pelajar" hasilkan "ajar"

Tabel 1. Aturan Pemenggalan (Lanjutan) [7].

Aturan	Format Kata	Pemenggalan
32	peCP...	pe-CP...dimana C!={r w y  m n} & P!='er'
33	peCerV...	per-CerV...dimana C!={r w y  m n}

### 2.3 Text Preprocessing

Untuk proses *mining* yang lebih lanjut, *text preprocessing* diperlukan untuk mengubah bentuk data yang belum memiliki struktur menjadi data yang terstruktur sesuai dengan kebutuhan [6]. Tahapan *text preprocessing* dalam penelitian ini meliputi : *case folding*, konversi kata, perbaikan kata, pemisahan kata, *tokenizing*, *stemming nazief* dan *adriani* serta *filtering*.

*Case folding* digunakan untuk mengubah semua huruf dalam dokumen menjadi huruf kecil. Hanya huruf 'a' sampai dengan 'z' yang diterima. Karakter selain huruf dan *Uniform Resources Locator* (URL) dihilangkan kecuali spasi. Konversi kata dilakukan untuk mengubah suatu kata menjadi kata baku, contohnya seperti "baLance" menjadi "seimbang" atau "hoax" menjadi "hoaks". Perbaikan kata dilakukan untuk mengubah kata yang penulisannya tidak benar menjadi benar, contohnya seperti "orng" menjadi "orang" atau "ketahuaaannn" menjadi "ketahuan". Pemisahan kata dilakukan untuk mengubah kata yang menyatu menjadi terpisah, contohnya seperti "kasuszyopi" menjadi "kasus zyopi". *Tokenizing* untuk memotong suatu kalimat berdasarkan spasi menjadi kumpulan kata. *Stemming nazief* dan *adriani* digunakan untuk mengubah suatu kata menjadi kata dasar. *Filtering* digunakan untuk mengambil kata penting. Cara yang dilakukan dalam *filtering* adalah membandingkan kata hasil *stemming nazief* dan *adriani* dengan kata tidak penting dan membuang kata tersebut jika termasuk kata tidak penting.

### 2.4 Confusion Matrix

*Confusion matrix* ditemukan oleh Karl Pearson pada tahun 1940. *Confusion matrix* merupakan sebuah tabel yang berfungsi untuk menyatakan jumlah data uji yang terklasifikasi dengan benar dan jumlah data uji yang terklasifikasi dengan salah [8].

Tabel 2. *Confusion Matrix*.

		Hasil Prediksi Kelas	
		+	-
Kelas Sebenarnya	+	True Positives	False Negatives
	-	False Positives	True Negatives

Hasil dari *Confusion matrix* digunakan untuk menghitung nilai akurasi, *precision* dan *recall* menggunakan Persamaan 5, Persamaan 6 dan Persamaan 7 [9]:

$$\text{akurasi} = \frac{TP+TN}{TP+TN+FP+FN} \quad (5)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (6)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (7)$$

Keterangan :

TP : True Positives.

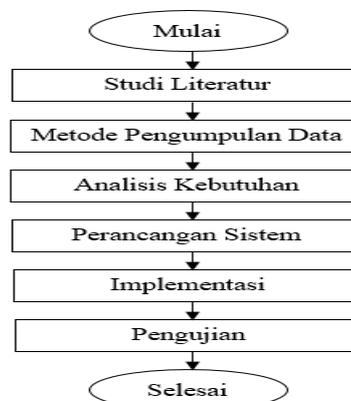
TN : True Negatives.

FP : False Positives.

FN : False Negatives.

### 3. METODE PENELITIAN

Dalam metode penelitian terdapat beberapa langkah yang dapat dilakukan untuk melakukan penelitian.



Gambar 1. Diagram Alir Metode Penelitian.

Metode penelitian dimulai dengan melakukan studi literatur yang berkaitan dengan media sosial, ujaran kebencian, *python*, *GetOldTweets3*, *text processing*, *stemming nazief* dan *adriani*, *naïve bayes classifier*, *confusion matrix*, *XAMPP*, *apache web server*, *MySQL*, *PHP* dan *HTML*.

Selanjutnya melakukan pengumpulan data dari *twitter* menggunakan *GetOldTweets3*. Data pernyataan di *twitter* dengan tagar ujaran kebencian (#hatespeech) serta dibatasi pada bahasa Indonesia diambil mulai dari Februari 2017 hingga September 2018. Data pernyataan

yang terkumpul ditentukan apakah termasuk ujaran kebencian atau bukan oleh bapak Albedhmake Vandaras Garagasi, M.pd., selaku guru bahasa Indonesia yang mengajar di Sekolah Menengah Atas Widya Pratama, bapak Dedy Ari Asfar, S.Pd., M.Lett., selaku ahli Linguistik di Balai Bahasa Kalimantan Barat dan bapak Agus Syahrani, M.M.S.Ling., selaku Dosen Fakultas Keguruan dan Ilmu Pendidikan di Universitas Tanjungpura. Kategori yang ditentukan dimasukkan ke aplikasi berdasarkan pilihan terbanyak. Data yang dikumpulkan bukan hanya data pernyataan. Data berupa daftar kata dasar serta kata tidak penting dikumpulkan dari [www.susantokun.com](http://www.susantokun.com).

Kemudian dilanjutkan dengan melakukan analisis kebutuhan apa saja yang diperlukan dalam penelitian ini. Kebutuhan yang diperlukan pada penelitian ini adalah sebuah laptop dengan spesifikasi *Intel® Celeron® quad core processor N2920 @ 1.86 GigaHertz, RAM 4 GigaByte, Harddisk 500 GigaByte dan VGA Intel® HD Graphics*. Selain sebuah laptop diperlukan juga perangkat lunak seperti *Sublime Text, XAMPP, Google Chrome, Python 3.4, GetOldTweets3, SQLizer dan Balsamiq Mockups 3*.

Lalu dilanjutkan dengan melakukan perancangan sistem. Rancangan yang dibuat adalah Diagram Alir Sistem, *Flowchart Stemming Nazief dan Adriani, Flowchart Naive Bayes Classifier, Data Flow Diagram (DFD), Entity Relationship Diagram (ERD), Conceptual Data Model (CDM)* dan Rancangan Tampilan Antarmuka. Kemudian dilakukan pengimplementasian rancangan yang dibuat ke dalam aplikasi deteksi ujaran kebencian. Pada implementasi dijabarkan tampilan antarmuka aplikasi untuk pengguna dan admin serta kode program. Selanjutnya melakukan pengujian untuk mengetahui berapa besar nilai akurasi, *precision* dan *recall* aplikasi yang menerapkan algoritma *stemming nazief dan adriani* serta *naive bayes classifier*. Selain nilai akurasi, *precision* dan *recall* akan dicari juga kata apa yang banyak digunakan pada data latih berkategori ujaran kebencian.

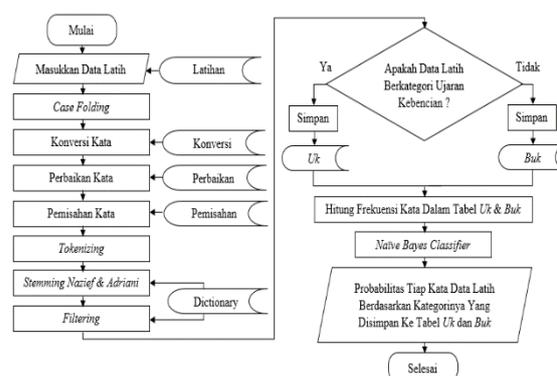
## 4. PERANCANGAN SISTEM

### 4.1 DIAGRAM ALIR

#### 4.1.1 Diagram Alir Pelatihan Data

Diagram alir pelatihan data dibuat untuk menggambarkan bagaimana aplikasi melatih suatu data. Alur pelatihan data dimulai

dengan memasukkan data latih. Alur pelatihan data dilanjutkan dengan *text preprocessing* yang meliputi *case folding*, konversi kata, perbaikan kata, pemisahan kata, *tokenizing*, *stemming nazief dan adriani* serta *filtering*. Dalam beberapa proses, ada saat dimana suatu data yang tersimpan dalam tabel diambil oleh sistem.

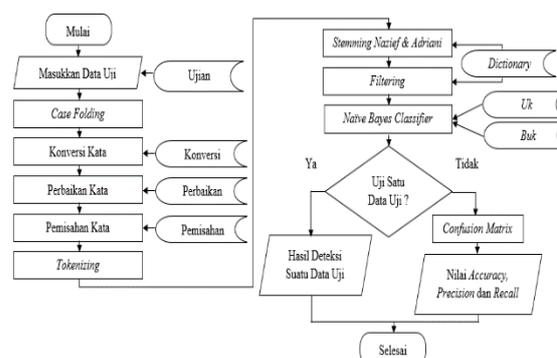


Gambar 2. Diagram Alir Pelatihan.

Tabel yang datanya diambil oleh sistem merupakan tabel latihan, konversi, perbaikan, pemisahan dan *dictionary*. Hasil yang diperoleh dari *text preprocessing* akan disimpan ke suatu variabel *array* sesuai kategori yang dimiliki oleh suatu data latih. Data variabel *array uk* dan *array buk* dihitung frekuensi kemunculan katanya. Hasil perhitungan frekuensi kemunculan kata digunakan untuk menghitung probabilitas tiap kata pada variabel *array uk* dan variabel *array buk* saat masuk ke dalam tahap *naive bayes classifier*. Hasil perhitungan probabilitas akan disimpan ke dalam tabel *uk* dan *buk*.

#### 4.1.2 Diagram Alir Pengujian Data

Diagram alir pengujian data dibuat untuk menggambarkan bagaimana alur aplikasi dalam menguji suatu data uji.

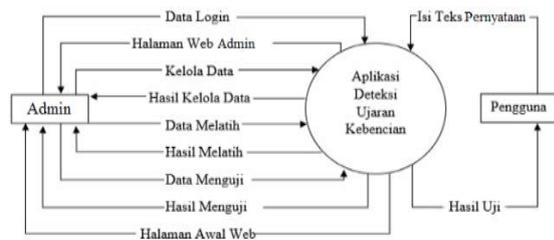


Gambar 3. Diagram Alir Pengujian.

Alur pengujian data dimulai dengan memasukkan data uji yang datanya diambil dari tabel ujian, dilanjutkan dengan *text preprocessing* mirip seperti yang ada pada alur pelatihan data. Berbeda dengan saat melatih data, pada tahap *naïve bayes classifier* terdapat proses mengambil data dari tabel *uk* dan *buk*. Pengujian dapat dilakukan untuk menguji satu atau semua data uji. Ketika menguji satu data uji, hasil deteksi berupa probabilitas ujaran kebencian dan probabilitas bukan ujaran kebencian suatu data uji akan ditampilkan. Ketika menguji semua data uji, tahap confusion matrix dijalankan dan hasil berupa nilai akurasi, *precision* dan *recall* yang diperoleh akan ditampilkan.

#### 4.2 Data Flow Diagram (DFD)

DFD level 0 seperti pada Gambar 4, dibuat untuk menggambarkan secara umum alur data dari aplikasi deteksi ujaran kebencian.



Gambar 4. DFD Level 0.

Aplikasi deteksi ujaran kebencian dapat digunakan admin dan pengguna. Admin dapat memasukkan data ke aplikasi deteksi ujaran kebencian. Data yang dimasukkan berupa data *login*, kelola data, data melatih dan data menguji. Hasil yang diperoleh dari masukkan admin terdiri dari halaman web admin, hasil kelola data, hasil melatih, hasil menguji dan halaman awal web. Untuk pengguna terdapat masukkan data berupa isi teks pernyataan. Isi teks pernyataan yang dimasukkan akan diproses aplikasi untuk memperoleh hasil uji.

## 5. HASIL DAN PEMBAHASAN

### 5.1 Implementasi Perangkat Lunak

#### 5.1.1 Tampilan Antarmuka Halaman Web Pengguna

Tampilan antarmuka halaman web pengguna diperlihatkan dalam Gambar 5. Dalam Gambar 5 disediakan kolom yang dapat diisi teks pernyataan dan tombol untuk memulai proses deteksi.



Gambar 5. Tampilan Antarmuka Halaman Web Pengguna.

Hasil deteksi yang diperoleh merupakan nilai probabilitas ujaran kebencian, nilai probabilitas bukan ujaran kebencian dan penentuan deteksinya.

#### 5.1.2 Tampilan Antarmuka Halaman Pelatihan dan Pengujian Data

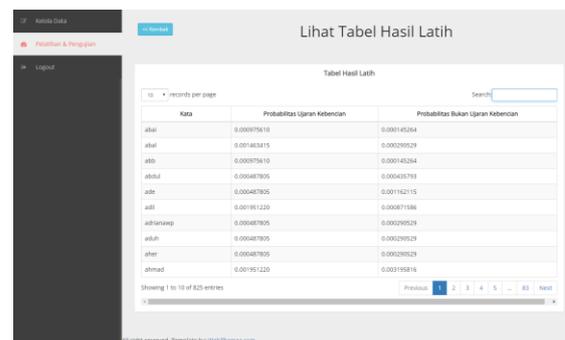
Tampilan halaman pelatihan dan pengujian data ditampilkan pada Gambar 6.



Gambar 6. Tampilan Halaman Pelatihan dan Pengujian Data.

Pada Gambar 6 terdapat tombol lihat hasil latih, tombol latih seluruh data latih, tombol uji seluruh data uji dan tombol uji data satu persatu.

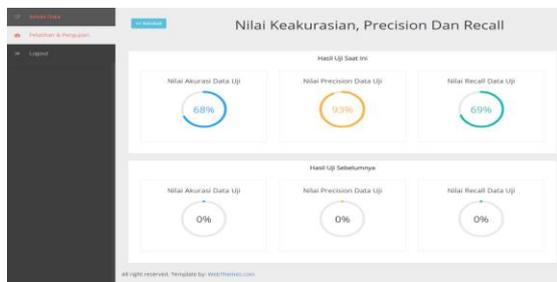
Tombol lihat hasil latih dapat ditekan untuk menampilkan halaman lihat tabel hasil latih. Tampilan halaman lihat tabel hasil latih ditampilkan pada Gambar 7.



Gambar 7. Tampilan Halaman Lihat Tabel Hasil Latih.

Pada Gambar 7 terdapat tombol kembali dan tampilan tabel hasil latihan.

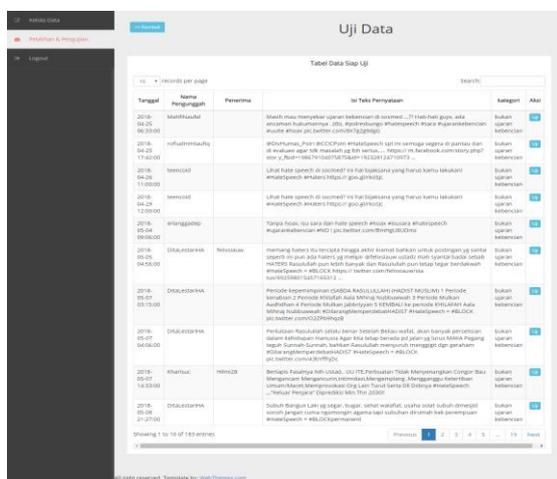
Tombol uji seluruh data uji digunakan untuk memulai proses pengujian dan menampilkan hasil pengujian. Hasil pengujian ditampilkan dalam Gambar 8.



Gambar 8. Tampilan Halaman Nilai Keakurasian, Precision Dan Recall.

Pada Gambar 8 terdapat tampilan nilai akurasi data uji, nilai precision data uji dan nilai recall data uji. Hasil uji yang ditampilkan merupakan hasil uji saat melakukan pengujian dan hasil pengujian sebelumnya.

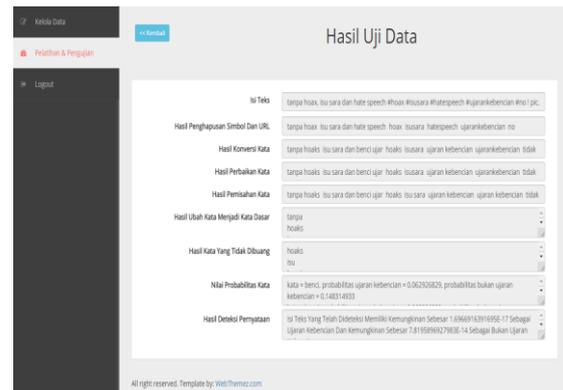
Tombol uji data satu persatu dapat digunakan untuk membuka tampilan halaman uji data satu persatu. Tampilan halaman uji data satu persatu ditampilkan pada Gambar 9.



Gambar 9. Tampilan Halaman Uji Data Satu Persatu.

Pada Gambar 9 terdapat tombol kembali dan tombol uji.

Tombol uji pada Gambar 9 dapat digunakan untuk memulai proses menguji satu data uji. Hasil uji satu data uji akan ditampilkan pada Gambar 10. Pada Gambar 10 terdapat tombol kembali dan kolom yang menampilkan hasil uji.



Gambar 10. Tampilan Halaman Hasil Uji Data.

## 5.2 Perhitungan Manual

### 5.2.1 Naïve Bayes Classifier

Perhitungan manual *naïve bayes classifier* dibuat untuk menggambarkan perhitungan yang terdapat pada metode *naïve bayes classifier*. Perhitungan yang dilakukan dalam metode *naïve bayes classifier* dimulai dengan menghitung nilai probabilitas kata data latih. Perhitungan nilai probabilitas kata data latih dicontohkan dengan menggunakan 5 data latih yang terdapat pada Tabel 3.

Tabel 3. Data Latih.

Nomor	Teks	Kategori
78	"Sangat meresahkan.."    #hatespeech , #turnbackHoax https://twitter.com/Fahrihamzah/status/863915185194455041&e=1	bukan ujaran kebencian atau <i>buk</i>
170	Harusnya produk gagal yg dikirim ke malingsial #HateSpeech https://twitter.com/WOWFAKTA/status/899267275605237760&e=1	ujaran kebencian atau <i>uk</i>
353	Sumpah wkwk #hatespeech	bukan ujaran kebencian atau <i>buk</i>
518	Lucu lu bgst #HateSpeech	ujaran kebencian atau <i>uk</i>
671	Sepi tapi berisik. #HateSpeech	bukan ujaran kebencian atau <i>buk</i>

Data latih pada Tabel 3 diproses dalam pemrosesan teks dan hasilnya ditampilkan pada Tabel 4.

Tabel 4. Hasil Pemrosesan Teks.

Nomor	Kata	Kategori
78	resah	bukan ujaran kebencian atau <i>buk</i>
	benci	
	hoaks	
170	harus	ujaran kebencian atau <i>uk</i>
	gagal	
	malingsial	
353	benci	bukan ujaran kebencian atau <i>buk</i>
	benci	
518	lucu	ujaran kebencian atau <i>uk</i>
	bangsat	
	benci	
671	sepi	bukan ujaran kebencian atau <i>buk</i>
	berisik	
	benci	

Data pada Tabel 4 digunakan untuk menghitung frekuensi dan probabilitas kata data latih. Hasil perhitungan frekuensi dan probabilitas kata data latih ditampilkan pada Tabel 5. Perhitungan Probabilitas kata data latih dilakukan dengan menggunakan Persamaan 4.

Tabel 5. Hasil Perhitungan Frekuensi Dan Probabilitas Kata Data Latih.

Nomor	Kata	Frekuensi <i>uk</i>	Frekuensi <i>buk</i>	Probabilitas <i>uk</i>	Probabilitas <i>buk</i>
1	resah	0	1	$P(\text{resah} uk) = \frac{0+1}{7+10} = 0.059$	$P(\text{resah} buk) = \frac{1+1}{7+10} = 0.118$
2	benci	2	3	$P(\text{benci} uk) = \frac{2+1}{7+10} = 0.177$	$P(\text{benci} buk) = \frac{3+1}{7+10} = 0.235$
3	hoaks	0	1	$P(\text{hoaks} uk) = \frac{0+1}{7+10} = 0.059$	$P(\text{hoaks} buk) = \frac{1+1}{7+10} = 0.118$
4	harus	1	0	$P(\text{harus} uk) = \frac{1+1}{7+10} = 0.118$	$P(\text{harus} buk) = \frac{0+1}{7+10} = 0.059$
5	gagal	1	0	$P(\text{gagal} uk) = \frac{1+1}{7+10} = 0.118$	$P(\text{gagal} buk) = \frac{0+1}{7+10} = 0.059$

Tabel 5. Hasil Perhitungan Frekuensi Dan Probabilitas Kata Data Latih (Lanjutan).

Nomor	Kata	Frekuensi <i>uk</i>	Frekuensi <i>buk</i>	Probabilitas <i>uk</i>	Probabilitas <i>buk</i>
6	maling sial	1	0	$P(\text{maling sial} uk) = \frac{1+1}{7+10} = 0.118$	$P(\text{maling sial} buk) = \frac{0+1}{7+10} = 0.059$
7	lucu	1	0	$P(\text{lucu} uk) = \frac{1+1}{7+10} = 0.118$	$P(\text{lucu} buk) = \frac{0+1}{7+10} = 0.059$
8	bangsat	1	0	$P(\text{bangsat} uk) = \frac{1+1}{7+10} = 0.118$	$P(\text{bangsat} buk) = \frac{0+1}{7+10} = 0.059$
9	sepi	0	1	$P(\text{sepi} uk) = \frac{0+1}{7+10} = 0.059$	$P(\text{sepi} buk) = \frac{1+1}{7+10} = 0.118$
10	berisik	0	1	$P(\text{berisik} uk) = \frac{0+1}{7+10} = 0.059$	$P(\text{berisik} buk) = \frac{1+1}{7+10} = 0.118$

Setelah nilai probabilitas kata data latih diperoleh, proses menentukan suatu data pernyataan termasuk sebagai ujaran kebencian atau bukan dapat dilakukan. Data pernyataan yang digunakan memiliki isi teks “tanpa hoax, isu sara dan hate speech #hoax #isusara #hatespeech #ujarankebencian #no ! pic.twitter.com/bmhgu8udmz” dan berkategori bukan ujaran kebencian. Isi teks pernyataan diproses dalam tahap pemrosesan teks dan dicari nilai probabilitas katanya, hasil yang diperoleh ditampilkan pada Tabel 6. Nilai probabilitas kata yang terdapat pada Tabel 6 diperoleh berdasarkan hasil pelatihan 731 data latih yang terdiri dari 104 data berkategori ujaran kebencian dan 627 data berkategori bukan ujaran kebencian.

Tabel 6. Nilai Probabilitas Kata Data Uji.

Nomor	Kata	Probabilitas <i>uk</i>	Probabilitas <i>buk</i>
1	hoaks	0.009268293	0.058105752
2	isu	0.000975610	0.000871586
3	benci	0.062926829	0.148314933
4	hoaks	0.009268293	0.058105752
5	isu	0.000975610	0.000871586
6	benci	0.062926829	0.148314933
7	benci	0.062926829	0.148314933
8	tidak	0.005853659	0.010894829

Perhitungan kemudian dilanjutkan dengan menghitung probabilitas setiap kategori pada data latih menggunakan Persamaan 3.

$$P(uk) = \frac{104}{731} = 0.14227086183311$$

$$P(buk) = \frac{627}{731} = 0.85772913816689$$

Setelah probabilitas setiap kategori pada data latih dihitung, perhitungan probabilitas data pernyataan dilakukan.

$$P(\text{data pernyataan}|uk) = P(uk) \times P(\text{hoaks}|uk) \times P(\text{isu}|uk) \times P(\text{benci}|uk) \times P(\text{hoaks}|uk) \times P(\text{isu}|uk) \times P(\text{benci}|uk) \times P(\text{tidak}|uk)$$

$$P(\text{data pernyataan}|uk) = 0.14227086183311 \times 0.009268293 \times 0.000975610 \times 0.062926829 \times 0.009268293 \times 0.000975610 \times 0.062926829 \times 0.062926829 \times 0.005853659 = 1.69669E-17$$

$$P(\text{data pernyataan}|buk) = P(buk) \times P(\text{hoaks}|buk) \times P(\text{isu}|buk) \times P(\text{benci}|buk) \times P(\text{hoaks}|buk) \times P(\text{isu}|buk) \times P(\text{benci}|buk) \times P(\text{tidak}|buk)$$

$$P(\text{data pernyataan}|buk) = 0.85772913816689 \times 0.058105752 \times 0.000871586 \times 0.148314933 \times 0.058105752 \times 0.000871586 \times 0.148314933 \times 0.148314933 \times 0.010894829 = 7.81959E-14$$

Dari perhitungan probabilitas data pernyataan, dapat diketahui bahwa nilai  $P(\text{data pernyataan}|uk) < P(\text{data pernyataan}|buk)$  maka data pernyataan ditentukan sebagai bukan ujaran kebencian.

### 5.3 Pengujian

Dari 1033 data pernyataan yang terkumpul, hanya 914 data pernyataan yang digunakan untuk pelatihan dan pengujian. 119 data pernyataan tidak digunakan karena isi pernyataan sepenuhnya tidak berbahasa Indonesia dan ada juga yang menggunakan bahasa yang tidak diketahui apa artinya, sehingga data pernyataan yang tidak digunakan diberi kategori tidak diketahui. Dari 914 data pernyataan, 731 data pernyataan digunakan untuk data latih dan 183 digunakan untuk data uji. 183 data uji terdiri dari 17 data berkategori ujaran kebencian dan 166 data berkategori bukan ujaran kebencian.

Pengujian dengan menggunakan 183 data uji memperoleh hasil pendeteksian yang ditampilkan dalam bentuk *confusion matrix*.

Tabel 7. *Confusion Matrix* Hasil Deteksi.

		Hasil Prediksi Kelas	
		Bukan Ujaran Kebencian	Ujaran Kebencian
Kelas Sebenarnya	Bukan Ujaran Kebencian	115	51
	Ujaran Kebencian	8	9

Dengan menggunakan nilai yang terdapat pada Tabel 7, penghitungan akurasi, *precision* dan *recall* dapat dilakukan. Penghitungan akurasi, *precision* dan *recall* menggunakan Persamaan 5, Persamaan 6 dan Persamaan 7.

$$\text{akurasi} = \frac{115+9}{183} = 0.6776 = 68\%$$

$$\text{Precision} = \frac{115}{115+8} = 0.9349 = 93\%$$

$$\text{Recall} = \frac{115}{115+51} = 0.6928 = 69\%$$

Dari perhitungan yang dilakukan diketahui bahwa nilai akurasi yang diperoleh sebesar 68%, nilai *precision* yang diperoleh sebesar 93% dan nilai *recall* yang diperoleh sebesar 69%.

Selain nilai akurasi, *precision* dan *recall* yang dapat diketahui, kata apa yang banyak digunakan pada data latih berkategori ujaran kebencian juga dapat diketahui dengan menghitung frekuensi kata pada tabel *uk*. Kata yang banyak digunakan dalam ujaran kebencian merupakan kata benci dengan jumlah 128 kata.

### 5.4 Pembahasan

Pada subbab ini, dilakukan pembahasan mengenai implementasi, kode program, perhitungan manual dan pengujian. Pada implementasi telah dijabarkan tampilan antarmuka halaman web pengguna, tampilan antarmuka halaman *login* admin, tampilan antarmuka halaman web admin kelola data serta tampilan antarmuka halaman pelatihan dan pengujian data. Pada kode program telah dijabarkan kode program *case folding*, kode program konversi kata, kode program pemisahan kata, kode program *tokenizing*, kode program *stemming nazief* dan *adriani*, kode program *filtering*, kode program *naïve bayes classifier* serta kode program *confusion matrix*. Pada perhitungan manual, dijabarkan proses perhitungan probabilitas kata data latih dan probabilitas data pernyataan.

Dari pengujian yang dilakukan dapat diketahui aplikasi dapat mendeteksi dengan benar 124 data uji dan mendeteksi dengan salah 59 data uji. Dari 124 data uji yang dideteksi dengan benar, 9 data uji berkategori ujaran kebencian dan 115 data uji berkategori bukan ujaran kebencian. Aplikasi melakukan kesalahan pendeteksian data uji sebagai ujaran kebencian sebanyak 51 data uji dan pendeteksian data uji sebagai bukan ujaran kebencian sebanyak 8 data uji. Dari hasil pendeteksian aplikasi yang benar dan salah dapat diperoleh nilai akurasi sebesar 68%, nilai *precision* sebesar 93% dan nilai *recall* sebesar 69%.

Selain mengetahui nilai akurasi, *precision* dan *recall*, pada penelitian ini dapat diketahui juga kata apa yang banyak digunakan pada data latih berkategori ujaran kebencian dengan menghitung frekuensi kata pada tabel *uk*. Kata yang banyak digunakan pada data latih berkategori ujaran kebencian merupakan kata benci dengan jumlah 128 kata.

## 6. KESIMPULAN DAN SARAN

### 6.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan terdapat beberapa kesimpulan yang dapat diambil, yaitu :

1. Dengan menerapkan algoritma *stemming nazief* dan *adriani* serta *naive bayes classifier*, aplikasi deteksi ujaran kebencian dapat memperoleh nilai akurasi sebesar 68%, nilai *precision* sebesar 93% dan nilai *recall* sebesar 69% dari pengujian 183 data uji.
2. Kata yang banyak digunakan pada 104 data latih berkategori ujaran kebencian merupakan kata benci dengan jumlah 128 kata.

### 6.2 Saran

Berdasarkan penelitian yang telah dilakukan terdapat saran yang dapat diberikan, yaitu menambahkan data dari ahli atau lembaga yang dapat dipertanggungjawabkan.

## DAFTAR PUSTAKA

- [1] Moeljatno. (2018). *KUHP Kitab Undang-Undang Hukum Pidana*. Jakarta: Bumi Aksara.
- [2] JDIH Kemkominfo. (2016). *Undang-Undang Nomor 19 Tahun 2016 tanggal 25 November 2016*. Dipetik Januari 18, 2020, dari [jdih.kominfo.go.id: https://jdih.kominfo.go.id/produk\\_hukum/view/id/555/t/undangundang+nomor+19+tahun+2016+tanggal+25+november+2016](https://jdih.kominfo.go.id/produk_hukum/view/id/555/t/undangundang+nomor+19+tahun+2016+tanggal+25+november+2016)
- [3] Ramdhani, A. (2018). Studi Algoritma Liner Support Vector Machine pada Deteksi Ujaran Kebencian Berbahasa Indonesia. *Seminar Nasional TEKNOKA ke - 3. Vol. 3*, hal. I-42-I-44. Jakarta: Fakultas Teknik, Universitas Muhammadiyah Prof. Dr. Hamka.
- [4] Fitriani, H. P., Ruslianto, I., & Hidayati, R. (2018). IMPLEMENTASI METODE NAIVE BAYES CLASSIFIER UNTUK APLIKASI FILTERING EMAIL SPAM DENGAN LEMMATIZATION BERBASIS WEB. *Volume 06, No.02*, 13-24.
- [5] Simarangkir, M. S. (2017). Studi Perbandingan Algoritma - Algoritma Stemming untuk Dokumen Teks Bahasa Indonesia. *2-14-1-PB, 1 No. 1*, 40-46. Diambil kembali dari <http://www.politeknikmeta.ac.id/meta/objects/>
- [6] Rahayu, N. M. (2018). Rancangan Penerapan Metode Naive Bayes dalam Mendeteksi Hate Speech di Media Sosial.
- [7] Adriani, M., Nazief, B., Asian, J., Tahaghoghi, S., & Williams, H. E. (2007). Stemming Indonesian: A Confix-Stripping Approach. *2007.Adriani.TALIP1*. doi:10.1145/1316457.1316459
- [8] Indriani, A. (2014). Klasifikasi Data Forum dengan menggunakan Metode Naive Bayes Classifier. *Seminar Nasional Aplikasi Teknologi Informasi (SNATI)* (hal. G-5-G-10). Yogyakarta: Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia.
- [9] Han, J., Kamber, M., & Pei, J. (2012). *Data Mining Concepts and Techniques* (Third Edition ed.). waltham, USA: Morgan Kaufmann.