

KLASIFIKASI SENTIMEN SARA, HOAKS DAN RADIKAL PADA POSTINGAN MEDIA SOSIAL MENGGUNAKAN ALGORITMA NAIVE BAYES MULTINOMIAL TEXT

Febry Eka Purwiantono¹⁾, Addin Aditya²⁾

¹⁾Manajemen Informatika, STIKI Malang

²⁾Sistem Informasi, STIKI Malang

^{1), 2)} Jl Raya Tidar No 100, Kota Malang 65146

Email: ¹⁾febry@stiki.ac.id, ²⁾addin@stiki.ac.id

Abstrak

Penelitian ini bertujuan untuk menerapkan sebuah algoritma klasifikasi yang dapat menjustifikasi sentimen pada kumpulan cuitan Twitter yang diposting oleh masyarakat Indonesia. Penerapan algoritma ini nantinya akan mengklasifikasikan cuitan mana yang mengandung unsur pelanggaran yang diatur dalam UU-ITE. Dengan adanya penerapan algoritma klasifikasi ini diharapkan dapat membantu pemerintah khususnya Kepolisian Republik Indonesia dan Badan Intelijen Negara dalam merumuskan kebijakan mengenai tindakan pencegahan pelanggaran UU-ITE serta mencegah penyebaran paham radikalisme, informasi palsu dan isu SARA di Negara Indonesia. Teknik pengumpulan data yang dilakukan pada penelitian ini yaitu menggunakan Twitter API (Application Programming Interface). Sedangkan algoritma klasifikasi yang digunakan pada penelitian ini yaitu Naive Bayes Multinomial Text. Algoritma ini dipilih karena mampu mengklasifikasikan dokumen dengan memperhitungkan jumlah kemunculan kata. Dari hasil kompilasi dan data yang diolah, algoritma ini mampu menjustifikasi sentimen secara akurat kurang lebih 99,62%.

Kata kunci: Hoaks, klasifikasi sentimen, Naive Bayes Multinomial Text, radikal, SARA

1. Pendahuluan

Negara Indonesia dengan beragam suku, adat dan budaya dari Sabang sampai Merauke sudah tentu menjadi ciri khas tersendiri. Namun ternyata keberagaman ini tidak sejalan dengan pesatnya perkembangan teknologi informasi. Perbedaan suku, agama, ras dan golongan menjadi sesuatu yang ramai diperdebatkan pada saat ini. Dunia maya menjadi tempat dimana pelanggaran Undang-Undang Informasi dan Transaksi Elektronik (UU-ITE) terjadi. Seringnya pelanggaran UU ITE seperti penyebaran informasi palsu, penyebaran paham radikalisme dan isu SARA (Suku, Agama, Ras dan Antar golongan) membuat betapa pentingnya upaya pencegahan tindakan inkonstitusional di dunia maya.

Seperti yang diketahui bahwa masyarakat yang terpapar paham radikalisme adalah masyarakat yang selalu

mengedepankan kepentingan golongannya sendiri dan tidak mentoleransi perbedaan di lingkungan sekitarnya khususnya permasalahan keyakinan atau agama. Saat ini, paham radikalisme, isu SARA dan informasi palsu dapat tersebar secara masif dengan perkembangan teknologi informasi khususnya melalui media sosial dan daring.

Penelitian ini bertujuan untuk menerapkan sebuah algoritma data mining klasifikasi yang dapat menjustifikasi sentimen pada kumpulan *tweet* (cuitan) Twitter yang diposting oleh masyarakat Indonesia. Media sosial Twitter dipilih pada penelitian ini karena Twitter merupakan salah satu media sosial terbesar di dunia yang sudah berumur kurang lebih 19 tahun dan digunakan oleh 328 juta orang di seluruh dunia. Di Indonesia sendiri, Twitter digunakan kurang lebih oleh 22,8 juta orang dan naik 2,4 juta pengguna setiap tahunnya (Abidin 2012; Statista 2019). Selain itu Twitter juga menyediakan API (Application Programming Interface) sehingga dapat memudahkan peneliti dalam pengumpulan data.

Penerapan algoritma ini nantinya akan mengklasifikasikan *tweet* mana yang mengandung unsur pelanggaran yang diatur dalam UU-ITE. Dengan adanya penerapan algoritma klasifikasi ini diharapkan dapat membantu pemerintah khususnya Kepolisian Republik Indonesia dan Badan Intelijen Negara dalam merumuskan kebijakan mengenai tindakan pencegahan pelanggaran UU-ITE serta mencegah penyebaran paham radikalisme, informasi palsu (hoaks) dan isu SARA di Negara Indonesia. Algoritma klasifikasi yang diusulkan pada penelitian ini yaitu Naive Bayes Multinomial Text.

Ada beberapa bentuk representasi dari algoritma Naive Bayes antara lain Naive Bayes Gaussian, Naive Bayes Bernaulli dan Naive Bayes Multinomial Text. Naive Bayes Gaussian merupakan metode distribusi peluang yang paling umum digunakan pada data numerik (Azizah et al. 2019). Naive Bayes Bernaulli merupakan metode yang menghitung nilai peluang masing-masing kata pada sebuah kalimat tanpa memperdulikan frekuensi kemunculan tiap kata (Adityawan 2014). Sedangkan Naive Bayes Multinomial Text mengasumsikan independensi di antara kemunculan kata dalam dokumen, tanpa memperhitungkan urutan kata maupun konteks informasi. Selain itu metode ini memperhitungkan

jumlah kemunculan kata dalam dokumen (Kurniawan et al. 2017).

Algoritma klasifikasi Naive Bayes Multinomial Text dipilih pada penelitian ini karena algoritma ini dirasa paling cocok untuk mengklasifikasikan dokumen berupa *tweet* Twitter dan mampu menghasilkan akurasi yang sangat baik seperti penelitian yang dilakukan oleh (Kalokasari et al. 2017), algoritma Naive Bayes Multinomial Text mampu menghasilkan akurasi kurang lebih 89,58%. Oleh sebab itu peneliti akan menggunakan algoritma klasifikasi Naive Bayes Multinomial Text untuk mengklasifikasikan sentimen berdasarkan *tweet* SARA, radikal, hoaks dan bukan ketiganya.

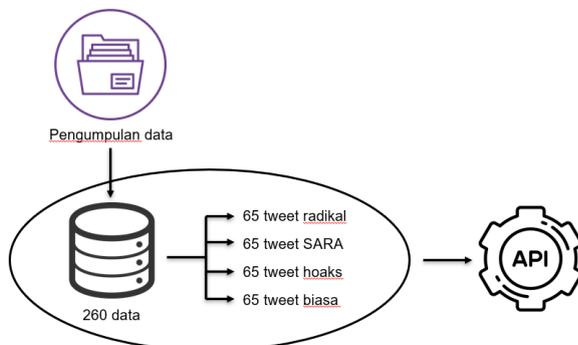
Teknik pengumpulan data yang dilakukan pada penelitian ini yaitu menggunakan *web crawler* API. *Web crawler* API merupakan sebuah sistem berbasis PHP yang menggunakan teknologi API (*Application Programming Interface*) yang sengaja dibuat pada penelitian ini untuk membantu peneliti melakukan *grabbing* data (cuitan) dari Twitter berdasarkan kata kunci yang telah ditentukan sebelumnya secara massal.

Agar data mentah tersebut dapat diolah menggunakan software Weka dan Microsoft Excel, maka harus dilakukan *pre-processing data* terlebih dahulu. Pada penelitian ini, *pre-processing data* juga berguna untuk menyeragamkan bentuk kata, menghilangkan *noise* dan memecah kalimat menjadi beberapa kata seperti yang dilakukan pada penelitian (Wahyuni et al. 2017). Hasilnya, akurasi yang diperoleh pada penelitian tersebut tembus hingga 98%.

2. Metode

2.1 Pengumpulan Data

Teknik pengumpulan data yang dilakukan yaitu dengan menggunakan *web crawler* API (*Application Programming Interface*) berbasis PHP. *Web crawler* tersebut akan melakukan *grabbing* data secara masif di Twitter berdasarkan kata kunci yang telah ditentukan sebelumnya. Jumlah data yang berhasil dikumpulkan kurang lebih 260 *tweet* (cuitan) yang terdiri dari 65 *tweet* SARA, 65 *tweet* hoaks, 65 *tweet* radikal dan 65 *tweet* bukan ketiganya (*tweet* biasa/normal).



Gambar 1. Teknik Pengumpulan Data

2.2 Pre-Processing Data

Setelah semua data yang dibutuhkan terkumpul, maka selanjutnya akan dilakukan *pre-processing data*. Pada penelitian ini ada 4 *pre-processing data* yang dilakukan antara lain:

- Data Integration

Data integration adalah suatu langkah untuk menggabungkan data dari beberapa sumber menjadi 1 file (Prianto and Bunyamin 2020). Pada penelitian ini, *data integration* digunakan untuk menggabungkan *tweet* (cuitan) Twitter yang diperoleh dari *web crawler* API (*Application Programming Interface*) menjadi 1 file yaitu file .ARFF (*Attribute-Relation File Format*). Perlu diketahui bahwa ARFF adalah salah satu tipe file yang paling sering digunakan di Weka.

- Case Folding

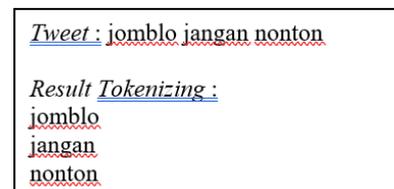
Case folding merupakan tahapan untuk merubah huruf kapital menjadi huruf kecil (Wahyuni et al. 2017). Tahapan ini perlu dilakukan pada penelitian ini untuk menghindari *case sensitive*. *Case sensitive* merupakan kasus dimana huruf besar dan huruf kecil diartikan berbeda.

- Filtering

Filtering digunakan untuk menghilangkan karakter atau kata yang tidak memiliki arti (Mahfud et al. 2020). Pada penelitian ini, *filtering* digunakan untuk menghilangkan karakter yang tidak penting pada *tweet* seperti @ (at), # (pagar), . (titik), , (koma), “ (petik dua), ‘ (petik satu), ((buka kurung),) (tutup kurung), { (buka kurung kurawal), } (tutup kurung kurawal), ? (tanda tanya), ! (tanda seru), * (bintang) dan lain-lain.

- Tokenizing

Tokenizing adalah proses pemisahan data *tweet* menjadi beberapa token atau kata (Mahfud and Tjahyanto 2017). Hal ini perlu dilakukan agar memudahkan algoritma Naive Bayes Multinomial Text dalam menghitung frekuensi kemunculan kata yang sama di dalam *data training*.



Gambar 2. Contoh Tokenizing

2.3 Klasifikasi Naive Bayes Multinomial Text

Setelah *pre-processing* selesai dilakukan, maka langkah selanjutnya yaitu menguji *data training* yang ada menggunakan algoritma Naive Bayes Multinomial Text. Langkah pertama yang dilakukan pada algoritma Naive Bayes Multinomial Text yaitu menghitung probabilitas *class* terhadap dokumen menggunakan rumus :

$$P(c) = \frac{N(c)}{N} \tag{1}$$

Dimana:

- $P(c)$: Probabilitas class terhadap dokumen.
- $N(c)$: Jumlah dokumen/tweet pada class.
- N : Jumlah dokumen/tweet.

Pada penelitian ini, jumlah dokumen/tweet yang digunakan yaitu 260 tweet yang dibagi menjadi 4 class antara lain class sara 65 tweet, class hoaks 65 tweet, class hoaks 65 tweet dan class normal 65 tweet. Sehingga masing-masing class memiliki probabilitas $65/260 = 0,25$.

Tabel 1. Probabilitas Class Terhadap Dokumen

Nama Class	Probabilitas
sara	0,25
hoaks	0,25
radikal	0,25
normal	0,25

Langkah selanjutnya adalah membuat *term-document matrix* untuk menghitung jumlah kata pada semua dokumen, jumlah kata unik pada semua dokumen, jumlah kata pada class sara, class hoaks, class radikal dan class normal. Hasilnya adalah sebagai berikut :

Tabel 2. Hasil Term-Document Matrix

Keterangan	Total
Jumlah kata pada semua dokumen	4.264
Jumlah kata unik pada semua dokumen	2.030
Jumlah kata pada class sara	1.315
Jumlah kata pada class hoaks	828
Jumlah kata pada class radikal	1.085
Jumlah kata pada class normal	1.036

Sedangkan pada Gambar 3 adalah sampel dari *term-document matrix* pada penelitian ini.

The frequency of a word given the class

sara	hoaks	radikal	normal	
1.0	1.0	1.0	2.0	diseg
1.0	1.0	1.0	2.0	dekan
2.0	1.0	1.0	1.0	mengharuskan
1.0	2.0	1.0	1.0	surat
2.0	1.0	2.0	1.0	monyet
2.0	1.0	1.0	1.0	pamer
1.0	2.0	1.0	1.0	cimahi
1.0	2.0	1.0	1.0	penanganan
1.0	1.0	2.0	1.0	jours
2.0	1.0	1.0	1.0	halal
2.0	1.0	2.0	1.0	begini
2.0	1.0	1.0	1.0	mengganti
1.0	1.0	1.0	2.0	susisatu
1.0	1.0	2.0	2.0	bandara
1.0	2.0	1.0	1.0	apa-apa
1.0	1.0	1.0	2.0	cemas
1.0	1.0	2.0	1.0	lipat
10.0	5.0	1.0	2.0	&
4.0	1.0	1.0	1.0	ktp
4.0	2.0	1.0	3.0	setelah
1.0	1.0	1.0	2.0	+
4.0	2.0	5.0	1.0	saja
1.0	1.0	1.0	2.0	marang

Gambar 3. Sampel Term-Document Matrix

Langkah selanjutnya adalah menghitung probabilitas kata unik untuk semua class menggunakan rumus :

$$P(w/c) = \frac{\text{count}(w,c)+1}{\text{count}(c)+|V|} \tag{2}$$

Dimana:

- $P(w/c)$: Probabilitas kata terhadap class.
- $\text{count}(w,c)$: Jumlah kemunculan kata pada class.
- $\text{count}(c)$: Jumlah kata pada class.
- V : Jumlah kata unik pada dokumen.
- $//$: Nilai mutlak.

Pada penelitian ini terdapat 2.030 kata unik dan 4 class, sehingga ada $2.030 \times 4 = 8.120$ probabilitas yang harus dicari nilainya. Oleh sebab itu, peneliti menggunakan software Weka untuk membantu proses klasifikasi dan validasi. Namun perhitungan manual juga tetap dilakukan menggunakan Microsoft Excel. Berikut ini adalah salah satu contoh perhitungan manual untuk mencari probabilitas kata “jomblo”, “jangan” dan “nonton” terhadap semua class.

Sebelum melakukan perhitungan, Langkah yang harus diselesaikan terlebih dahulu adalah mencari frekuensi kemunculan kata “jomblo”, “jangan” dan “nonton” terhadap tiap class. Berikut adalah hasilnya :

Tabel 3. Frekuensi Kemunculan Kata Terhadap Class

Kata	Freq sara	Freq hoaks	Freq radikal	Freq normal
jomblo	0	0	0	1
jangan	3	6	0	1
nonton	0	1	0	4

Dari Tabel 3 dapat dilihat bahwa kata “jomblo” muncul 1x di dokumen normal dan tidak muncul sama sekali di dokumen sara, hoaks dan radikal. Kata “jangan” muncul 3x di dokumen sara, muncul 6x di dokumen hoaks, muncul 1x di dokumen normal dan tidak muncul sama sekali di dokumen radikal. Sedangkan kata “nonton” muncul 1x di dokumen hoaks, muncul 4x di dokumen normal dan tidak muncul sama sekali di dokumen sara maupun radikal.

Setelah frekuensi kemunculan kata terhadap class ditemukan, maka langkah selanjutnya adalah menghitung probabilitas kata “jomblo”, “jangan” dan “nonton” terhadap semua class menggunakan rumus (2). Berikut adalah hasil perhitungannya:

- Probabilitas kata “jomblo” terhadap class sara = $\frac{0+1}{1315+|2030|} = \frac{1}{3345} = 0,00030$
- Probabilitas kata “jomblo” terhadap class hoaks = $\frac{0+1}{828+|2030|} = \frac{1}{2858} = 0,00035$
- Probabilitas kata “jomblo” terhadap class radikal = $\frac{0+1}{1085+|2030|} = \frac{1}{3115} = 0,00032$
- Probabilitas kata “jomblo” terhadap class normal = $\frac{1+1}{1036+|2030|} = \frac{2}{3066} = 0,00065$
- Probabilitas kata “jangan” terhadap class sara = $\frac{3+1}{1315+|2030|} = \frac{4}{3345} = 0,00120$
- Probabilitas kata “jangan” terhadap class hoaks =

- $\frac{6+1}{828+|2030|} = \frac{7}{2858} = \mathbf{0,00245}$
- Probabilitas kata “jangan” terhadap class radikal = $\frac{0+1}{1085+|2030|} = \frac{1}{3115} = \mathbf{0,00032}$
- Probabilitas kata “jangan” terhadap class normal = $\frac{1+1}{1036+|2030|} = \frac{2}{3066} = \mathbf{0,00065}$
- Probabilitas kata “nonton” terhadap class sara = $\frac{0+1}{1315+|2030|} = \frac{1}{3345} = \mathbf{0,00030}$
- Probabilitas kata “nonton” terhadap class hoaks = $\frac{1+1}{828+|2030|} = \frac{2}{2858} = \mathbf{0,00070}$
- Probabilitas kata “nonton” terhadap class radikal = $\frac{0+1}{1085+|2030|} = \frac{1}{3115} = \mathbf{0,00032}$
- Probabilitas kata “nonton” terhadap class normal = $\frac{4+1}{1036+|2030|} = \frac{5}{3066} = \mathbf{0,00163}$

Setelah 8.120 probabilitas berhasil dihitung, Langkah berikutnya adalah menghitung probabilitas dokumen/kalimat/tweet terhadap class menggunakan rumus :

$$P(c/d_{(n)}) = P(c) \times \prod P(w/c) \tag{3}$$

Dimana:

- $P(c/d_{(n)})$: Probabilitas kalimat terhadap class.
- $P(c)$: Probabilitas class terhadap dokumen.
- \prod : Product (perkalian beruntun).
- $P(w/c)$: Probabilitas kata terhadap class.

Seperti yang diketahui sebelumnya, pada penelitian ini terdapat 260 tweet yang terbagi menjadi 4 class, sehingga jumlah probabilitas kalimat terhadap class yang harus dicari pada penelitian ini adalah $260 \times 4 = 1.040$ probabilitas. Berikut adalah salah satu contoh perhitungan probabilitas kalimat terhadap class menggunakan dokumen/kalimat/tweet dengan nomor 231 yang berbunyi “jomblo jangan nonton”:

- $P(sara/d_{(231)}) = \mathbf{0,25} \times 0,00030 \times 0,00120 \times 0,00030 = \mathbf{0,00000000026718}$
- $P(hoaks/d_{(231)}) = \mathbf{0,25} \times 0,00035 \times 0,00245 \times 0,00070 = \mathbf{0,000000000149928}$
- $P(radikal/d_{(231)}) = \mathbf{0,25} \times 0,00032 \times 0,00032 \times 0,00032 = \mathbf{0,00000000008271}$
- $P(normal/d_{(231)}) = \mathbf{0,25} \times 0,00065 \times 0,00065 \times 0,00163 = \mathbf{0,000000000173482}$

Hasil dari perhitungan tersebut kemudian dibandingkan. Ternyata $P(normal/d_{(231)})$ memiliki nilai probabilitas tertinggi bila dibandingkan dengan probabilitas kalimat terhadap class yang lain, sehingga dokumen dengan nomor 231 yang berbunyi “jomblo jangan nonton” diklasifikasikan sebagai sentimen normal (bukan SARA, radikal maupun hoaks). Perhitungan tersebut juga diimplementasikan kepada 259 dokumen lainnya. Kemudian hasil dari klasifikasi tersebut direkap ke dalam confusion matrix. Berikut adalah penampakkannya :

Tabel 4. Confusion Matrix for 4 Classes

Confusion Matrix	Pred sara	Pred hoaks	Pred radikal	Pred normal
Act sara	65	0	0	0
Act hoaks	0	65	0	0
Act radikal	0	0	65	0
Act normal	1	0	0	64

Pada Tabel 4, act diasumsikan sebagai data aktual dan pred sebagai data prediksi. Sehingga dapat ditarik kesimpulan bahwa “tidak ada tweet SARA yang dideteksi sebagai tweet bukan SARA”, “tidak ada tweet hoaks yang dideteksi sebagai tweet bukan hoaks”, “tidak ada tweet radikal yang dideteksi sebagai tweet bukan radikal” dan “ada 1 tweet normal yang dideteksi sebagai tweet SARA dan sisanya dideteksi sebagai tweet normal”.

3. Hasil dan Pembahasan

Untuk memastikan perhitungan manual dari probabilitas kalimat “jomblo jangan nonton” terhadap masing-masing class sudah sesuai atau belum, maka peneliti menguji validitas dari perhitungan tersebut menggunakan software Weka dengan rumus :

$$np(c) = \frac{P(c|d(n))}{\sum P(c|d(n))} \tag{4}$$

Dimana:

- $np(c)$: Nilai prediksi class.
- $P(c/d_{(n)})$: Probabilitas kalimat terhadap class.
- \sum : Menjumlahkan seluruh bilangan

Berikut adalah hasil perhitungannya:

- $np(sara) = 0,00000000026718 / (0,00000000026718 + 0,000000000149928 + 0,00000000008271 + 0,000000000173482) = 0,00000000026718 / 0,000000000358399 = \mathbf{0,075}$
- $np(hoaks) = 0,000000000149928 / (0,00000000026718 + 0,000000000149928 + 0,00000000008271 + 0,000000000173482) = 0,000000000149928 / 0,000000000358399 = \mathbf{0,418}$
- $np(radikal) = 0,00000000008271 / (0,00000000026718 + 0,000000000149928 + 0,00000000008271 + 0,000000000173482) = 0,00000000008271 / 0,000000000358399 = \mathbf{0,023}$
- $np(normal) = 0,000000000173482 / (0,00000000026718 + 0,000000000149928 + 0,00000000008271 + 0,000000000173482) = 0,000000000173482 / 0,000000000358399 = \mathbf{0,484}$

Kemudian nilai prediksi dibandingkan satu sama lain untuk menemukan nilai prediksi yang paling besar. Nilai prediksi yang paling besar akan dipilih sebagai nilai akurasi dari klasifikasi sebuah kalimat/tweet. Dalam sampel ini, tweet “jomblo jangan nonton” diprediksi 0,484 atau **48,4%** sebagai sentimen normal. Gambar 4 adalah sepenggal penampakkan dari hasil nilai prediksi yang dihasilkan oleh software Weka. Di sana terlihat bahwa nilai prediksi dari dokumen 231 (“jomblo jangan nonton”) adalah 0.484. Artinya, perhitungan manual

yang telah dilakukan valid.

211	4:normal	4:normal	1
212	4:normal	4:normal	0.781
213	4:normal	4:normal	0.989
214	4:normal	4:normal	0.987
215	4:normal	4:normal	0.999
216	4:normal	4:normal	1
217	4:normal	4:normal	1
218	4:normal	4:normal	0.953
219	4:normal	4:normal	0.977
220	4:normal	4:normal	1
221	4:normal	4:normal	0.994
222	4:normal	4:normal	1
223	4:normal	4:normal	0.921
224	4:normal	4:normal	0.509
225	4:normal	4:normal	0.991
226	4:normal	4:normal	0.993
227	4:normal	4:normal	1
228	4:normal	4:normal	1
229	4:normal	4:normal	0.981
230	4:normal	4:normal	0.999
231	4:normal	4:normal	0.484 ✓
232	4:normal	4:normal	1
233	4:normal	4:normal	1
234	4:normal	4:normal	1
235	4:normal	4:normal	1
236	4:normal	4:normal	1
237	4:normal	4:normal	1

Gambar 4. Hasil Nilai Prediksi Dokumen pada Weka

Perlu diketahui bahwa 48,4% bukanlah akurasi keseluruhan dari klasifikasi pada penelitian ini, melainkan nilai prediksi dari dokumen 231. Jika melihat Gambar 4, nilai prediksi yang dihasilkan sangat bervariasi seperti dokumen 224 yang memiliki nilai prediksi 0,509, dokumen 230 yang memiliki nilai prediksi 0,999 dan lain sebagainya. Untuk menghitung keseluruhan akurasi pada penelitian ini, maka harus berdasarkan *confusion matrix* pada Tabel 4 dengan menggunakan rumus sebagai berikut :

$$A = \frac{TP}{N} \tag{5}$$

Dimana:

- A : Akurasi.
- TP : True Positive.
- N : Jumlah dokumen/tweet.

TP (True Positive) adalah jumlah dokumen yang diprediksi benar. Pada penelitian ini terdapat 4 class, sehingga bentuk *confusion matrix* dan letak TP tidak sama dengan klasifikasi 2 class. Berikut adalah bagan *confusion matrix* yang bisa digunakan untuk mencari TP pada klasifikasi 4 class.

Confusion Matrix	Predict A	Predict B	Predict C	Predict D
Act A	TP			
Act B		TP		
Act C			TP	
Act D				TP

Gambar 5. Bagan Confusion Matrix untuk 4 Class

Sehingga akurasi yang dihasilkan adalah $\frac{65+65+65+64}{260} = \frac{259}{260} = 0,996154$ atau setara **99,6154%**. Sesuai dengan hasil yang diperoleh bila menggunakan software Weka seperti yang terlihat pada Gambar 6.

```
Time taken to test model on training data: 0.16 seconds

=== Summary ===

Correctly Classified Instances      259      99.6154 % ✓
Incorrectly Classified Instances     1         0.3846 %
Kappa statistic                     0.9949
Mean absolute error                  0.0098
Root mean squared error              0.051
Relative absolute error              2.6125 %
Root relative squared error          11.7767 %
Total Number of Instances           260
```

Gambar 6. Akurasi pada Softwarei Weka

4. Kesimpulan

Dari penelitian yang telah dilakukan dapat diambil kesimpulan bahwa algoritma Naive Bayes Multinomial Text sangat cocok digunakan untuk mengklasifikasikan data yang berbentuk dokumen, karena algoritma ini mampu menghasilkan akurasi yang sangat tinggi yaitu 99,62%. Untuk proses klasifikasinya tidak memakan waktu yang cukup lama yaitu hanya sekitar 0,16 detik, karena pada algoritma Naive Bayes Multinomial Text tidak perlu mencari nilai *Evidence* seperti algoritma Naive Bayes Gaussian.

Secara praktis hasil dari penelitian ini dapat membantu pemerintah khususnya Kepolisian Republik Indonesia dan Badan Intelijen Negara dalam merumuskan kebijakan mengenai tindakan pencegahan pelanggaran UU-ITE serta mencegah penyebaran paham radikalisme, informasi palsu dan isu SARA di Negara Indonesia melalui media sosial apabila penelitian ini diimplementasikan menjadi sistem deteksi berbasis Android, web maupun *Add-On* pada *web browser*.

Daftar Pustaka

Abidin, T.F. 2012. *Accuracy Measure*. Program Studi Teknik Informatika FMIPA Universitas Syiah Kuala.

Adityawan, E. 2014. Analisis Sentimen Dengan Klasifikasi Naive Bayes Pada Pesan Twitter Menggunakan Data Seimbang.

Azizah, N. et al. 2019. Metode Naive Bayes Dengan Pendekatan Distribusi Gauss Untuk Klasifikasi Peminatan Peserta Didik. *Prosiding Seminar Nasional Matematika, Statistika, dan Aplikasinya 2019*, pp. 8–14.

Kalokasari, D.H. et al. 2017. IMPLEMENTASI ALGORITMA MULTINOMIAL NAIVE BAYES CLASSIFIER PADA SISTEM KLASIFIKASI SURAT KELUAR (Studi Kasus : DISKOMINFO Kabupaten Tangerang). *JURNAL TEKNIK INFORMATIKA*. doi: 10.15408/jti.v10i2.6199.

Kurniawan, B. et al. 2017. Klasifikasi Berita Twitter Menggunakan Metode Improved Naive Bayes. *Jurnal Pengembangan Teknologi Informasi dan*

- Ilmu Komputer (J-PTIHK) Universitas Brawijaya* 1(10), pp. 1193–1200.
- Mahfud, F.K.R. et al. 2020. Sentiment Analysis of Perpustakaan Nasional Republik Indonesia Through Social Media Twitter. *Matics* 12(1), p. 90. doi: 10.18860/mat.v12i1.8973.
- Mahfud, F.K.R. and Tjahyanto, A. 2017. Improving Classification Performance of Public Complaints With TF-IGM Weighting. *2017 International Conference on Sustainable Information Engineering and Technology (SIET)* , pp. 220–225.
- Prianto, C. and Bunyamin, S. 2020. *Pembuatan aplikasi clustering gangguan jaringan menggunakan metode K-Means Clustering*. Kreatif Industri Nusantara.
- Statista 2019. Twitter: number of users in Indonesia 2019 | Statista. Available at: <https://www.statista.com/statistics/490548/twitter-users-indonesia/> [Accessed: 15 August 2019].
- Wahyuni, R.T. et al. 2017. Penerapan Algoritma Cosine Similarity dan Pembobotan TF-IDF pada Sistem Klasifikasi Dokumen Skripsi. *Jurnal Teknik Elektro* 9(1), pp. 18–23.