KLASIFIKASI DIALEK BAHASA JAWA MENGGUNAKAN METODE *NAIVES BAYES*

Grace Angeline [1], Aji Prasetya Wibawa [2], Utomo Pujianto[3]
[1,2,3] Universitas Negeri Malang
aji.prasetya.ft@um.ac.id

ABSTRAK

Indonesia merupakan negara dengan keragaman suku bangsa dan budaya. Orang dengan suku yang berbeda akan berkomunikasi dengan cara yang berbeda sehingga setiap daerah memiliki dialeknya masing-masing. Pulau Jawa merupakan pulau terpadat di Indonesia dan memiliki keragaman dialek yang tinggi. Berdasarkan peta bahasa yang dikeluarkan oleh KEMDIKBUD, Pulau Jawa memiliki 12 dialek utama yang tersebar di Jawa Timur, Jawa Barat dan Jawa Tengah. Keberagaman dialek yang ada seringkali membuat masyarakat kebingungan dan menyebabkan kesulitan dalam berkomunikasi terutama pada proses penyampaian informasi atau percakapan berupa teks. Dari hasil survei yang telah dilakukan, dialek yang digunakan sebagai dataset hanya dibatasi menjadi 3 dialek terpopuler dari setiap provinsi yaitu Dialek Cirebon, Dialek Tegal dan Dialek Jawa Timur. Penyediaan data dilakukan dengan metode studi literatur yang bersumber dari buku dan dokumen tertulis yang tersedia di internet. Data akan diolah dan dianalisis menggunakan algoritma *Multinomial Naives Bayes* karena cepat dalam proses perhitungan, sederhana dan memiliki akurasi yang tinggi. Algoritma akan diuji menggunakan *K-fold Cross Validation* untuk mengetahui performa algoritma *Multinomial Naives Bayes* dalam melakukan klasifikasi dialek di Pulau Jawa. Metode *Synthetic Minority Over-Sampling Technique* (SMOTE) juga digunakan dalam penelitian ini untuk mengetahui pengaruh teknik *oversampling* terhadap performa algoritma. Dari penelitian in dihasilkan performa terbaik dengan akurasi sebesar 96,97%, presisi sebesar 97,53% dan recall sebesar 96,83%.

Keyword : *Dialek, Bahasa Jawa, Text Classification, Naives Bayes*

1. PENDAHULUAN.

Indonesia memiliki keragaman suku dan budaya yang melimpah sehingga menjadikan Indonesia salah satu negara dengan bahasa daerah terbanyak di dunia yaitu lebih dari 700 bahasa [1]. Dari sekian banyak bahasa daerah yang ada di Indonesia, Bahasa Jawa memiliki jumlah penutur yang paling banyak di Indonesia dengan diperkirakan mampu mencapai 75 juta penutur [2]. Keberagaman dialek di Pulau Jawa membuat banyak masyarakat yang bingung membedakan berbagai macam dialek tersebut.

Dialek dalam suatu bahasa merupakan salah satu aspek penting dalam sebuah bahasa dan nproses komunikasi [3]. Identifikasi dialek ternilai lebih sulit dibandingkan dengan identifikasi bahasa walaupun identifikasi dialek termasuk ke dalam sub-bagian dari identifikasi bahasa. Hal ini dikarenakan tugas dari identifikasi dialek adalah untuk dapat membedakan sebuah dialek yang biasanya memiliki tingkat kemiripan dengan dialek lainnya dalam bahasa yang sama [2].

Selama ini, dialek dikenali oleh orang-orang tertentu misalnya ahli bahasa dan masyarakat yang tinggal di daerah itu. Untuk orang-orang awam atau orang yang tinggal di luar daerah tersebut akan kesulitan dalam mengklasifikasikan dialek. Penelitian ini dilakukan untuk melakukan klasifikasi dialek Bahasa Jawa berbentuk teks menggunakan metode *data mining.*

Text mining adalah salah satu variasi dari *data mining* dan merupakan teknik yang dapat digunakan untuk melakukan klasifikasi dengan cara menemukan pola-pola menarik dari sekumpulan besar data tekstual [4]. Algoritma yang digunakan dalam text mining banyak jenisnya yaitu *Support Vector Machines* (SVM), *K-Nearest Neighbours* (KNN), *Naive Bayes*, dan *Decision Trees* [dang][5]. *Text mining* berbasis probabilitas adalah *Naive Bayes*, berbasis nilai/jarak tetangga terdekat adalah KNN, berbasis kernel adalah SVM dan berbasis jumlah pohon adalah *Decision Tree*. Setiap metode ini memiliki karakteristik berbeda-beda.

Penelitian ini menggunakan algoritma *Naive Bayes* yang cepat dalam proses perhitungan dan memiliki akurasi yang tinggi. Sebagai upaya untuk menyeimbangkan data, digunakan metode *Synthetic Minority Over-Sampling Technique* (SMOTE). Kedua skenario ini akan dianalisis menggunakan *K-fold Cross Validation* untuk mengetahui performa algoritma *Multinomial Naives Bayes* dengan dan tanpa SMOTE dalam melakukan klasifikasi dialek di Pulau Jawa. *Dataset* yang digunakan dalam penelitian ini merupakan *dataset* Bahasa Jawa dengan Dialek Cirebon, Dialek Tegal dan Dialek Jawa Timur.

2. TINJAUAN PUSTAKA

Algoritma *Naive Bayes* banyak digunakan untuk klasifikasi teks dalam berbagai Bahasa, antara lain Bali [6] dan Arab [7]. Algoritma *Naive Bayes* adalah algoritma klasifikasi menggunakan probabilitas dan statistik berdasarkan teorema *Bayes*. *Naive Bayes* memiliki beberapa jenis yaitu

Gaussian Naive Bayes, Multinomial Naive Bayes dan *Bernoulli Naive Bayes*. Model *Gaussian* digunakan dalam klasifikasi dasar dan mengasumsikan bahwa fitur dari kumpulan data mengikuti distribusi normal [8]. Model *Multinomial* bekerja pada konsep frekuensi istilah yang berapa kali kata muncul dalam dokumen sedangkan model *Bernoulli* bekerja pada konsep biner bahwa apakah istilah itu muncul atau tidak dalam dokumen [9].

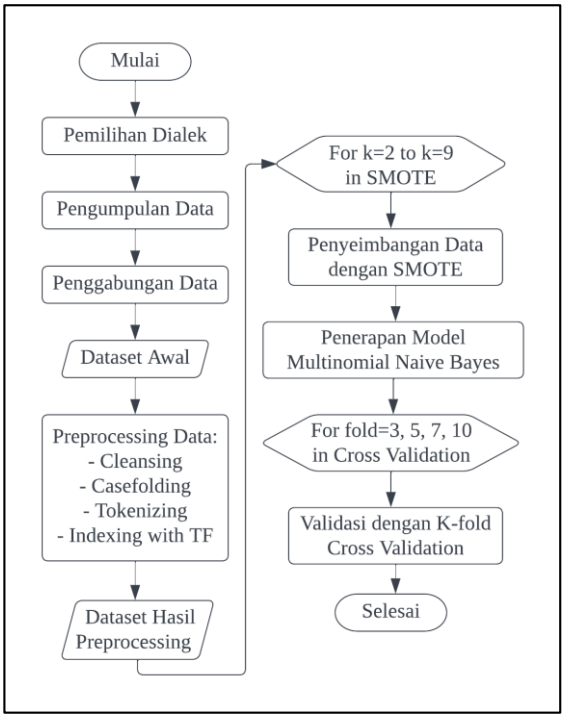
Salah satu kelebihan *Naive Bayes* adalah tingkat akurasi yang tinggi dengan perhitungan sederhana [10]. Algoritma ini dapat bekerja lebih baik dalam hal akurasi dan efisiensi komputasi dibanginkan algoritma rumit lainnya karena diasumsikan bahwa semua fitur bersifat independen [11]. Di sisi lain, asumsi bahwa semua fitur independen biasanya tidak terjadi di kehidupan nyata sehingga membuat algoritma *Naive Bayes* kurang akurat dibandingkan algoritma yang rumit.

Berbagai upaya untuk memperbaiki kekurangan *Naive Bayes* telah dilakukan antara lain *feature weighting* dan *laplace calibration* untuk meningkatkan akurasi [12]. Untuk mengatasi kekurangan *Naives Bayes* terkait ketidakseimbangan data digunakan teknik SMOTE dan *generic algorithm* untuk meningkatkan akurasi [13]. Optimasi berupa *feature extraction* dengan kombinasi *unigram, bigram,* dan *trigram* juga telah dilakukan untuk meningkatkan kinerja algoritma *Naive Bayes* [14].

Sebelumnya telah dilakukan penelitian untuk mengklasifikan Bahasa Jawa berdasarkan suara ucapan [15], semantik adjektiva [16] dan leksikon yaitu kasar (ngoko) dan halus (krama) [17]. Pada penelitian ini ada perbedaan pemilihan dialek Bahasa Jawa yang digunakan sebagai *dataset* serta algoritma yang digunakan untuk proses klasifikasi. Terdapat pembaruan pada penelitian ini dimana dialek Bahasa Jawa berbentuk teks diklasifikasikan berdasarkan regional/daerah dialek dan dilakukannya proses optimasi menggunakan SMOTE untuk menyeimbangkan *dataset*.

3. METODE PENELITIAN

Pada penelitian ini dilakukan berbagai tahapan untuk mengklasifikasi dialek Bahasa Jawa. Tahap pertama dimulai dengan pemilihan hingga proses validasi sebagai tahap akhir. Untuk mendapatkan hasil yang lebih baik dilakukan proses pembersihan data dan proses optimasi. Tahapan-tahapan yang dilakukan dalam penelitian ini dapat dilihat pada Gambar 1.



Gambar 1. *Flowchart* langkah penelitian

3.1. Pemilihan Dialek Bahasa Jawa

Dataset yang dipakai untuk pembuatan sistem ini adalah Bahasa Jawa dengan Dialek Cirebon, Dialek Tegal dan Dialek Jawa Timuran. Dialek tersebut dipilih melalui survei yang dilakukan melalui *Google Search Engine*. Pertama dilakukan pencarian dari *keyword* tiap dialek dan kemudian dilakukan pencatatan terhadap jumlah hasil pencarian (*result*) dari Google. Dari pencarian yang dilakukan, dipilih dialek terpopuler dari setiap provinsi.

3.2. Pengumpulan Data

Proses pengumpulan data dilakukan dengan metode studi literatur yang bersumber dari buku dan dokumen tertulis yang tersedia di internet. Data dikumpulkan dari berita, cerita pendek, kosakata, lirik lagu, puisi serta kumpulan kalimat terkait Dialek Cirebon, Dialek Tegal dan, Dialek Jawa Timuran yang tersedia di internet. Kata dan kalimat dari tiap dialek dimasukkan ke dalam 1 dokumen dan tiap dokumen dipisahkan berdasarkan dialek serta sumber pengambilan data.

Tabel 1. Jumlah kata, kalimat dan dokumen

Dialek	Kata Kunci	Kata	Kalimat	Dokumen
Cirebon	6510	4303	1477	10
Tegal	13900	5625	1103	20
Jawa Timuran	3820	4608	1037	23

3.3. Penggabungan Data

Penggabungan data dari 53 dokumen yang ada dilakukan menggunakan bahasa R. Hal ini dilakukan mengimpor 2 *library* dan membuat *list* kategori variabel baru. Selanjutnya semua *dataset* dibaca dan dibuatkan *dataframe* baru. Semua isi dari tiap *dataset* dimasukkan ke dalam kategori yang sesuai dan disimpan dalam 1 *file* baru. Berikut merupakan *pseudocode* dari proses penggabungan data yang dilakukan dalam penelitian ini.

Import library (readtext, dplyr)

Membuat list

Membaca dataset dari path and memasukkan kedalam variabel list bernama list_categories

Membuat dataframe dengan variabel df_final dan membuat kolom (File_Name, Content, dan Category)

For (category in list_categories) {

Mengambil dataset ke dalam category dimasukkan ke dalam category_path

Membaca dataset dengan readtext dan dimasukkan ke dalam variabel df

Memasukkan nama_file sebagai File_Name, teks sebagai Content, Nama Folder sebagai Category menggunakan fungsi rbind kedalam variabel df_final (df_final = rbind(df_final,df))

}

Menyimpan dataset menggunakan fungsi save dengan 'dataset_final.rda'

Melakukan load dengan fungsi load dan encoding data sebagai .csv dengan fileEncoding = 'UTF-8')

3.4. Text Preprocessing

Text preprocessing merupakan tahap sebelum proses pengklasifikasian untuk mempersiapkan teks [18]. *Preprocessing* dilakukan dengan tujuan untuk mentransformasi data yang mentah dalam hal ini baru didapatkan dari proses pengambilan data ke format data yang lebih efisien dan agar data yang digunakan lebih optimal ketika digunakan pada proses pengklasifikasiannya [19]. Tahapan *preprocessing* setiap kasus dapat berbeda- beda. Dapat dilihat pada tabel 2 tahap-tahapan *text preprocessing* yang dilakukan pada penelitian-penelitian terkait [20]–[22].

Tabel 2. Tahapan *text preprocessing*

Peneliti, Tahun	Tahapan
Kadhim, 2018	Stopword removal, stemming, tokenization, chi-square, TF-IDF
Goncalves et al., 2010	Stopword removal, stemming, wordnet, pruning
Ayedh et al., 2016	Stopword removal, word stemming, normalization
Uysal and Gunal, 2014	Tokenization, stopword removal, lowercase conversion, stemming
Krouska et al., 2016	TF-IDF weighting, stopword removal, word stemming, tokenization
Song et al., 2005	Stopword removal, word stemming, indexing with TF, weighting with IDF, normalization
Jianqiang and Xiaolin, 2017	Replacing negative mentions, removing URL, reverting words, removing numbers, removing stopwords, expanding acronyms
HaCohen-Kerner, et al., 2019	Spelling correction, HTML tag removal, converting uppercase into lowercase, punctuation mark removal, reduction of repeated characters, stopword removal
Angiani et al., 2016	Basic cleaning operations, word negation, word correction, stemming, stopword removal

Pada penelitian ini hanya digunakan 4 tahapan *text preprocessing* dari semua tahapan yang ada yaitu *cleansing, casefolding, tokenizing* dan *indexing with term frequency* (TF). Hal ini dikarenakan semua informasi dan data yang dikumpulkan terkait semua dialek dianggap penting serta aturan yang dimiliki setiap dialek berbeda-beda.

3.4.1. Cleansing

Cleansing merupakan proses membersihkan data yang akan digunakan dari karakter-karakter bahkan kata-kata yang tidak diperlukan. Hal ini bertujuan untuk mengurangi *noise* yang dapat menimbulkan proses perhitungan dalam pengklasifikasian tidak optimal [19]. *Cleansing data* dilakukan dengan membersihkan data dari tanda baca seperti titik, koma, tanda seru dan tanda tanya. Untuk tanda hubung dan tanda petik satu masih digunakan sehingga tidak dihapus dalam proses *cleansing*. Pada proses ini juga dilakukan pembersihan data dari alfabet yang tidak digunakan seperti é, ü, dan ý serta angka.

Pseudocode:

Import library (pandas, numpy, matplotlib.pyplot, re)

Membaca dataset .csv (dataset_final.csv) dengan library pd.read_csv sebagai variabel df_FF

Mengubah isi Category menjadi Cirebon, Tegal dan Jawa Timuran menggunakan fungsi str.replace

Membuat variabel cleaning = list ('0123456789éê!#$%&()*+,./:;<=>?@\[\\\]_`{|}~') dan petik_2 = list('"')

For cln in cleaning:
hapus cln (df_FF.str.replace(cln,'')

For ptk in petik_2:
ubah ptk menjadi petik satu (df_FF.str.replace(ptk, '"'))

Tabel 3. Contoh hasil *cleansing*

Data Masukan	Hasil Cleasing
Ya iku mau, to!	Ya iku mau to
Kapan Nur olehe mulih?	Kapan Nur olehe mulih
tempéléng	tempeleng

3.4.2. Casefolding

Case folding merupakan proses pengubahan data menjadi format yang sesuai dan dilakukan untuk menyeragamkan karakter pada data. Hal ini dilakukan dengan mengubah semua huruf pada teks menjadi huruf kecil dan semua karakter selain huruf dihilangkan [18]. Dalam penelitian ini proses *case folding* dilakukan dengan mengubah seluruh huruf menjadi huruf kecil (*lowercase*).

Pseudocode:
Mengubah format data pada variabel df_FF menjadi lowercase menggunakan fungsi ft_FF.Text = df_FF.Text.str.lower()

Tabel 4. Contoh hasil *casefolding*

Data Masukan	Hasil Casefolding
Ancene iku cara Malang	ancene iku cara malang
Tapi nggone rusuh lho	tapi nggone rusuh lho
Dhik Sekolah gae bahasa Indonesia	dhik sekolah gae bahasa indonesia

3.4.3. Tokenizing

Tokenizing adalah tahap pemotongan atau pemisahan data baik berupa frasa, klausa, atau kalimat menjadi kata perkata berdasarkan *delimeter* yang digunakan yaitu spasi [23]. Proses ini dilakukan dengan memotong *string* masukan berdasarkan kata-kata yang menyusunnya atau dengan kata lain pemecahan kalimat menjadi kata. Pada penelitian pemecahan kalimat dilakukan dengan metode *unigram*. Jadi setiap *string* yang diproses akan menjadi potongan-potongan 1 kata.

Pseudocode:
Import library (nltk, word tokenize)
Melakukan tokenizing pada teks dengan fungsi tokenize = nltk.tokenize.word_tokenize(df_category)

Tabel 5. Contoh hasil *tokenizing*

Data Masukan	Hasil Tokenizing
padha ae kok	padha ae kok
aja isin-isin lho	aja isin-isin lho
tapi larang-larang	tapi larang-larang to

3.4.4. Term Frequency (TF)

Pada penelitian ini dilakukan proses perhitungan *term frequency* yang ada di keseluruhan dataset. Frekuensi kemunculan kata/term di dalam dataset yang diberikan menunjukkan seberapa penting kata itu di dalam dataset tersebut. Pada penelitian ini, nilai TF yang tinggi mengindikasikan bahwa kata tersebut penting dalam dalam proses identifikasi kelas dialek tersebut. Jika terdapat kata yang sama pada beberapa dialek, TF dapat digunakan untuk menentukan dimana kata tersebut paling mempengaruhi identifikasi kelas dialek dengan melihat nilai TF yang lebih tinggi.

Frekuensi kemunculan kata juga digunakan dalam perhitungan *Multinomial Naives Bayes* untuk menghitung probabilitas kemunculan suatu kata dalam suatu kelas. Setelah dilakukan *tokenizing*, total kata/*term* adalah 34.807. Frekuensi kemunculan kata paling tinggi untuk dialek Cirebon adalah kata "bebasan", untuk dialek Tegal adalah kata "ora" dan untuk dialek Jawa Timuran adalah kata "sing".

3.5. SMOTE

Dalam penelitian ini digunakan metode *oversampling* agar tidak menghapus *instance* dari data dialek yang mungkin membawa beberapa informasi penting. SMOTE atau *Synthetic Minority Over-Sampling Technique* yang diusulkan oleh Chawla [24] tidak hanya menduplikasi data yang sama melainkan akan membuat sampel baru yang menyerupai data asli dari kelas minoritas (data sintetis) untuk menyeimbangkan dataset.

SMOTE telah dilakukan sebelumnya untuk meningkatkan kinerja *classifier* dikarenakan data tidak seimbang yaitu pada teks pendek bahasa Arab [25]. Beberapa penelitian menyimpulkan bahwa SMOTE merupakan salah satu teknik yang menghasilkan performa terbaik untuk menangani keseimbangan data dalam klasifikasi teks [26], [27]. Oleh karena itu, pada penelitian ini digunakan teknik SMOTE untuk menyeimbangkan *dataset*.

Dalam penelitian ini digunakan metode SMOTE karena jumlah label yang tidak seimbang antar kelas Cirebon, Tegal dan Jawa Timuran. Melalui metode ini jumlah proporsi data kelas minoritas yaitu kelas Cirebon dan kelas Tegal akan ditambah sehingga sama dengan kelas mayoritas yaitu kelas Jawa Timuran. Prosedur pemerataan ini digunakan untuk mencegah model untuk condong ke arah kelas Jawa Timuran sehingga dapat menurunkan *bias* dalam model.

Pseudocode:
Input: Jumlah minoritas kelas, Jumlah SMOTE N%, nilai dari k
Ouput: (N/100) * T

If N < 100
Then random dari jumlah T minoritas
T = (N/100)*T
N = 100
N = (int(N/100))
for i <- 1 to T

```
        Hitung k tetangga terdekat untuk i dan simpan
        N-array

Menidentifikasi populasi (N, i, N-array)

Memilih jumlah random antara 1 dan k yang disebut nn.
Tahap ini memilih 1 k dari i
for attr <- 1 to jumlah atribut
        Hitung
        dif = Sample[N-array[nn][attr]] -
        Sample[[i][attr]]
        gap = rand(0,1)
        Sintetis[new][attr] = Sample[i][attr] + gap *dif
new++
N = N – 1

return(populasi akhir)
```

Langkah pertama pada SMOTE adalah perhitungan perbedaan jumlah label antara kelas mayoritas dan minoritas. Untuk kelas Cirebon memiliki perbedaan 13 label dan untuk kelas Tegal memiliki perbedaan 3 label. Pada langkah kedua dilakukan perhitungan persentase duplikasi yang diiginkan pada kelas minoritas. Untuk kelas Cirebon memiliki persentase 130% dan untuk kelas Tegal memiliki persentase 15%. Proses selanjutnya dilakukan dengan memilih jumlah k. Pada penelitian ini dilakukan beberapa kali percobaan dengan nilai k=2-9, dimana nilai k=9 merupakan nilai maksimal dikarenakan jumlah label pada kelas paling minoritas adalah 10. Data sintetis dibentuk sesuai jumlah ketetanggan terdekat (k) yang dipilih sebanyak persentase duplikasi yang diinginkan. Kelas Cirebon akan memiliki 13 data sintetis baru dan kelas Tegal akan memiliki 3 data sintetis baru.

3.6. Multinomial Naive Bayes

Algoritma *Multinomial Naive Bayes* merupakan salah satu bentuk pengembangan dari algoritma bayes yang cocok dalam pengklasifikasian teks atau dokumen [28]. Metode *Multinomial Naive Bayes* sering digunakan dalam penelitian tentang klasifikasi teks karena kesederhanaan dan efektivitasnya yang menggunakan ide dasar probabilitas gabungan dari kata-kata dan kategori untuk memperkirakan probabilitas kategori pada suatu dokumen [29]. Pada formula *Multinomial Naive Bayes*, kelas dialek tidak hanya ditentukan dengan kata yang muncul tetapi juga jumlah kemunculannya. *Multinomial Naive Bayes* adalah salah satu metode yang dipakai dengan memperhitungkan frekuensi kemunculan token atau kata dalam sebuah dokumen [30].

Pseudocode:

```
Import library (numpy, pandas, matplotlib, operator)

Melakukan training label dengan fungsi
train_label = open('dataset’)

Ekstraksi values/nilai dari label dengan fungsi
lines = train_label.readlines()

Mengambil jumlah dokumen dengan fungsi
total = len(lines)

Menghitung frekuensi kemunculan tiap kelas dialek
for line in lines:
   val = int(line.split()[0])
   pi[val] += 1

Probabilitas kelas dialek dengan total dokumen
for key in pi:
   pi[key] /= total

Menghitung probabilitas tiap kata sesuai kelas dialek
pb_ij = df.groupby(['classIdx','wordIdx'])
pb_j = df.groupby(['classIdx'])
Pr =  (pb_ij['count'].sum() + a) / (pb_j['count'].sum())

Melakukan smoothing
if smooth:
   probability=Pr_dict[wordIdx][classIdx]
   power = np.log(1+ new_dict[docIdx][wordIdx])

Mengambil kelas dengan probabilitas tertinggi
max_score = max(score_dict, key=score_dict.get)
prediction.append(max_score)
```

Algoritma ini dimulai dengan menetapkan label dari data yang ada dan melakukan ekstraksi nilai dari label tersebut. Tahap kedua dilakukan perhitungan jumlah dokumen, jumlah kelas, dan jumlah kata dari semua data. Tahapan selanjutnya dilakukan perhitungan *prior probability* untuk menghitung probabilitas kelas dialek dan *post probability* untuk menghitung probabilitas suatu kata masuk ke dalam sebuah kelas dialek. Untuk menghindari *zero probability* dilakukan *laplacian smoothing* dan kemudian diambil kelas dengan probabilitas tertinggi sebagai hasil prediksi.

3.7. K-fold Cross Validation

K-fold cross validation merupakan metode yang digunakan untuk mengevaluasi kinerja *Multinomial Naives Bayes* dalam melakukan klasifikasi dialek. Metode ini digunakan karena dalam penelitian ini jumlah data dialek terbatas (jumlah *instance* tidak banyak). *Cross validation* dilakukan dengan membagi data menjadi himpunan bagian k dengan ukuran yang hampir sama, model dalam klasifikasi dilatih dan diuji sebanyak k. Di setiap pengulangan, salah satu himpunan bagian akan digunakan sebagai data latih dan sub kelompok data k lainnya berfungsi sebagai data penguji [31].

Sebagian besar penggungaan *fold*=10 dapat memberikan hasil yang baik tetapi terkadang di beberapa kasus *fold*=5 sudah cukup memadai [32]. Pada penelitian yang dilakukan Nti [33], nilai *fold*=7 dapat memberikan peningkatan dalam akurasi validasi. Selain itu penelitian yang dilakukan Tempola [34] dibagi kedalam 3 *fold* disetiap metode klasifikasi diperoleh perbandingan akurasi sistem rata-rata tertinggi. Oleh karena itu, pada peneilitian ini digunakan nilai *fold*=3,5,7,10. Terdapat 36

skenario dimana skenario 1-4 dilakukan tanpa menggunakan SMOTE. Skenario 5-36 dilakukan menggunakan SMOTE dengan nilai k=2-9.

4. HASIL DAN PEMBAHASAN

Tiap data diberi label sesuai dialek yang ada yaitu dialek Cirebon, dialek Tegal dan dialek Jawa Timuran untuk dilatih dan kemudian diuji. Berikut merupakan jumlah dokumen dan kata dari tiap dialek sebelum dan setelah dilakukan SMOTE.

Tabel 6. Jumlah dokumen dan kata

Keterangan	Sebelum	Setelah
Jumlah Dokumen	53	69
Jumlah Kata	34807	51067

Kelas minoritas yaitu dialek Cirebon dan Tegal diseimbangkan dengan kelas mayoritas menggunakan nilai k=2-9. Setelah dilakukan SMOTE, semua data naik ke kelas mayoritas yaitu masing-masing 23 label. Pengujian dilakukan tanpa menggunakan SMOTE serta menggunakan SMOTE dengan nilai k=2-9.

Tabel 7. Hasil pengujian tanpa SMOTE

fold	akurasi (%)	presisi (%)	recall (%)
3	77,34	52,81	63,89
5	75,45	52,67	62,67
7	75,26	51,27	62,70
10	75,33	53,06	62,78

Tabel 7 menunjukan hasil pengujian menggunakan *K-fold Cross Validation* dengan nilai fold=3,5,7,10 dengan metode *Multinomial Naives Bayes*. Dari hasil pengujian dapat disimpulkan nilai akurasi, presisi dan recall terbaik tanpa *oversampling* didapatkan dengan nilai *fold*=5.

Tabel 8. Hasil pengujian dengan SMOTE

fold	k	akurasi (%)	presisi (%)	recall (%)
3	2,3,4,5,7,8	95,45	95,94	95,44
	6,9	96,97	97,53	96,83
5	2,5,6,7	94,07	95,10	93,67
	3,4,8,9	95,60	96,43	95,33
7	2,3,4,5,7,8	93,65	95,71	94.05
	6,9	95,24	96,51	95,63
10	2,3,4,6,7,8,9	95,71	96,67	95,56
	5	95,71	96,67	95,56

Tabel 8 menunjukan performa *Naive Bayes* menggunakan SMOTE dengan berbagai macam skenario *fold* dan nilai k. Pada *fold*=3, akurasi tertinggi ada pada k=9 (96,97%). Nilai yang sama juga didapat pada k=6. Pada kedua skenario ini, nilai presisi dan recall juga mencapai tertinggi yaitu 97,53% dan 96,83%. Secara umum, nilai k yang tinggi akan menghasilkan akurasi yang lebih baik. Hal ini dapat dikarenakan pengambilan kata-kata untuk pembentukan data sintesis yang lebih beragam pada teknik SMOTE.

Dari hasil pengujian dapat dilihat bahwa nilai akurasi, presisi dan recall meningkat saat SMOTE diterapkan. Jika SMOTE tidak digunakan, jumlah label yang tidak seimbang antar kelas dapat meningkatkan *bias* dalam model. Hal ini dapat menyebabkan performa model menjadi lebih rendah karena model akan cenderung condong ke kelas mayoritas yaitu dialek Jawa Timuran.

SMOTE dapat meningkatakn performa *classifier* karena dapat mempengaruhi *prior probability* masing-masing kelas dialek. *Prior probability* merupakan salah satu perhitungan yang digunakan dalam proses klasifikasi menggunakan metode *Multinomial Naïve Bayes.* Kelas dengan *prior probability* yang kecil akan cenderung diklasifikasn ke kelas dengan nilai yang lebih besar. Oleh karena itu, dengan menggunakan SMOTE nilai *prior probability* masing-masing kelas akan seimbang sehingga *classifier* dapat melakukan klasifikasi dengan lebih tepat.

Dari hasil pengujian tanpa SMOTE dan dengan SMOTE terdapat kesalahan dalam prediksi antar dialek. Secara umum kesalahan terdapat pada data dialek Cirebon dan Jawa Timuran yang diprediksi sebagai dialek Tegal dikarenakan beberapa kemiripan kata yang ada di dalam kelas-kelas tersebut. Contohnya kata abah yang muncul di dialek Tegal dan Cirebon dan kata nang yang muncul di dialek Tegal dan Jawa Timuran. Selain itu terdapat kata-kata yang muncul pada ketiga kelas tersebut dengan frekuensi yang hampir sama sehingga proses klasifikasi menjadi lebih sulit. Contohnya kata wis yang muncul pada semua kelas dialek dan memiliki frekuensi yang sama pada dialek Tegal dan Jawa Timuran yaitu 107 kali.

5. KESIMPULAN DAN SARAN

Dari hasil penelitian dapat disimpulkan bahwa teknik SMOTE dapat meningkatkan kinerja model *Multinomial Naive Bayes* walaupun masih ditemukan beberapa kesalahan klasifikasi karena kemiripan kata antar dialek. Penggunaan SMOTE dapat meningkatkan hasil akurasi hingga 20%, presisi hingga 45% dan recall hingga 35% pada klasifikasi dialek bahasa Jawa. Nilai *fold*=3 pada *K-fold Cross Validation* dan k=9 pada SMOTE memiliki hasil terbaik. Pada penelitian ini *dataset* hanya dibatasi untuk 3 dialek bahasa Jawa terpopuler sehingga kedepannya dapat ditambahkan dialek lainnya.

DAFTAR PUSTAKA

[1] D. Tuhenay, “Perbandingan Klasifikasi Bahasa Menggunakan Metode Naïve Bayes Classifier (NBC) Dan Support Vector Machine (SVM),” *JIKO (Jurnal Inform. dan Komputer)*, vol. 4, no. 2, pp. 105–111, 2021, doi: 10.33387/jiko.v4i2.2958.

[2] R. D. Pamungkas and A. F. Hidayatullah, "Tinjauan Literatur : Identifikasi Dialek Dengan Deep Learning," *Automata*, vol. 2, no. 1, 2021.

[3] S. Siregar, "The Influence of Dialect on the Student's Pronunciation in Speaking Ability," *Pedagog. J. English Lang. Teach.*, vol. 5, no. 1, pp. 28–36, Jan. 2017.

[4] E. E. Pratama and R. L. Atmi, "A Text Mining Implementation Based on Twitter Data to Analyse Information Regarding Corona Virus in Indonesia," *J. Comput. Soc.*, vol. 1, no. 1, pp. 91–100, 2020, [Online]. Available: https://ejournal.upi.edu/index.php/JCS/article/view/25502

[5] S. Dang and P. H. Ahmad, "A Review of Text Mining Techniques Associated with Various Application Text Mining View project A Review of Text Mining Techniques Associated with Various Application Areas," *Areas Artic. Int. J. Sci. Res.*, vol. 4, 2015, Accessed: Apr. 25, 2022. [Online]. Available: www.ijsr.net

[6] I. B. G. W. Putra, M. Sudarma, and I. N. S. Kumara, "Klasifikasi Teks Bahasa Bali dengan Metode Supervised Learning Naive Bayes Classifier," *Teknol. Elektro*, vol. 15, no. 2, pp. 81–86, 2016, [Online]. Available: https://ojs.unud.ac.id/index.php/JTE/article/view/ID21577

[7] M. El Kourdi, A. Bensaid, and T. Rachidi, "Automatic Arabic Document Categorization Based on the Naïve Bayes Algorithm," in *Computational Approaches to Arabic Script-based Languages*, 2004, p. 51. doi: 10.3115/1621804.1621819.

[8] M. Ismail, N. Hassan, and S. S. Bafjaish, "Comparative Analysis of Naive Bayesian Techniques in Health-Related for Classification Task," *J. Soft Comput. Data Min.*, vol. 1, no. 2, pp. 1–10, 2020, [Online]. Available: https://www.researchgate.net/publication/352312714_Comparative_Analysis_of_Naive_Bayesian_Techniques_in_Health-Related_for_Classification_Task

[9] G. Singh, B. Kumar, L. Gaur, and A. Tyagi, "Comparison between Multinomial and Bernoulli Naïve Bayes for Text Classification," in *2019 International Conference on Automation, Computational and Technology Management (ICACTM)*, Apr. 2019, pp. 593–596. doi: 10.1109/ICACTM.2019.8776800.

[10] F. Handayani and F. S. Pribadi, "Implementasi Algoritma Naive Bayes Classifier dalam Pengklasifikasian Teks Otomatis Pengaduan dan Pelaporan Masyarakat melalui Layanan Call Center 110," *J. Tek. Elektro*, vol. 7, no. 1, pp. 19–24, 2015, doi: 10.15294/JTE.V7I1.8585.

[11] S. L. Ting, W. H. Ip, and A. H. C. Tsang, "Is Naïve bayes a good classifier for document classification?," *Int. J. Softw. Eng. its Appl.*, vol. 5, no. 3, pp. 37–46, 2011.

[12] H. Chen, S. Hu, R. Hua, and X. Zhao, "Improved naive Bayes classification algorithm for traffic risk management," *EURASIP J. Adv. Signal Process.*, vol. 2021, no. 30, pp. 1–12, 2021, doi: 10.1186/s13634-021-00742-6.

[13] A. R. Safitri and M. A. Muslim, "Improved Accuracy of Naive Bayes Classifier for Determination of Customer Churn Uses SMOTE and Genetic Algorithms," *J. Soft Comput. Explor. JOSCEX*, vol. 6, no. 1, pp. 70–75, 2020.

[14] A. Solikhatun and E. Sugiharti, "Application of the Naïve Bayes Classifier Algorithm using N-Gram and Information Gain to Improve the Accuracy of Restaurant Review Sentiment Analysis," *J. Adv. Inf. Syst. Technol.*, vol. 2, no. 2, pp. 1–12, 2020.

[15] E. R. Andrianti and C. Atmaji, "Klasifikasi Dialek Bahasa Jawa Berdasarkan Suara Ucapan Menggunakan Jaringan Syaraf Tiruan," Universitas Gadjah Mada, Yogyakarta, 2019. Accessed: Jun. 29, 2022. [Online]. Available: http://etd.repository.ugm.ac.id/penelitian/detail/173221

[16] I. A. Shitadevi and N. M. Dhanawaty, "View of Klasifikasi Semantik Adjektiva Bahasa Jawa Dialek Malang," *Linguistika*, vol. 28, no. 1, pp. 29–39, Mar. 2021, Accessed: Jun. 29, 2022. [Online]. Available: https://ojs.unud.ac.id/index.php/linguistika/article/view/67414/37445

[17] W. Aulia Zakki, "Klasifikasi Dokumen Teks Bahasa Jawa dengan Menggunakan Metode Naive Bayes," Universitas Dian Nuswantoro, Semarang, 2017. Accessed: Jun. 29, 2022. [Online]. Available: www.dinus.ac.id

[18] Agung Hasbi Ardiansyah, K. Paranita Kartika, and S. Nur Budiman, "Penerapan Latent Semantic Indexing Pada Sistem Temu Balik Informasi Pada Undang-Undang Pemilu Berdasarkan Kasus," *J. Mnemon.*, vol. 4, no. 2, pp. 64–70, 2021, doi: 10.36040/mnemonic.v4i2.4165.

[19] F. A. Muttaqin and A. M. Bachtiar, "Implementasi Teks Mining Pada Aplikasi Pengawasan Penggunaan Internet Anak 'Dodo Kids Browser,'" *J. Ilm. Komput. dan Inform.*, pp. 1–8, 2016.

[20] A. I. Kadhim, "An Evaluation of Preprocessing Techniques for Text Classification," *Int. J. Comput. Sci. Inf. Secur.*, vol. 16, no. 6, pp. 22–32, 2018, [Online]. Available: https://sites.google.com/site/ijcsis/

[21] Y. HaCohen-Kerner, D. Miller, and Y. Yigal,

"The influence of preprocessing on text classification using a bag-of-words representation," *PLoS One*, vol. 15, no. 5, pp. 1–22, May 2020, doi: 10.1371/JOURNAL.PONE.0232525.

[22] G. Angiani *et al.*, "A comparison between preprocessing techniques for sentiment analysis in Twitter," 2016.

[23] F. S. Jumeilah, "Penerapan Support Vector Machine (SVM) untuk Pengkategorian Penelitian," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 1, no. 1, pp. 19–25, 2017, doi: 10.29207/resti.v1i1.11.

[24] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002, doi: 10.1613/JAIR.953.

[25] S. Al-Azani and E. S. M. El-Alfy, "Using Word Embedding and Ensemble Learning for Highly Imbalanced Data Sentiment Analysis in Short Arabic Text," *Procedia Comput. Sci.*, vol. 109, pp. 359–366, 2017, doi: 10.1016/J.PROCS.2017.05.365.

[26] C. Padurariu and M. E. Breaban, "Dealing with Data Imbalance in Text Classification Dealing with Data Imbalance in Text Classification," *Procedia Comput. Sci.*, vol. 159, pp. 736–745, 2019, doi: 10.1016/j.procs.2019.09.229.

[27] A. Indrawati, H. Subagyo, A. Sihombing, Wagiyah, and S. Afandi, "Analyzing the Impact of Resampling Method for Imbalanced Data Text in Indonesian," *BACA J. Dokumentasi dan Inf.*, vol. 42, no. 2, pp. 133–141, 2020.

[28] A. H. Setianingrum, D. H. Kalokasari, and I. M. Shofi, "Implementasi Algoritma Multinomial Naive Bayes Classifier," *J. Tek. Inform.*, vol. 10, no. 2, pp. 109–118, 2017, doi: 10.15408/jti.v10i2.6822.

[29] A. Sabrani, I. G. W. Wedashwara W., and F. Bimantoro, "Multinomial Naïve Bayes untuk Klasifikasi Artikel Online tentang Gempa di Indonesia," *J. Teknol. Informasi, Komputer, dan Apl. (JTIKA)*, vol. 2, no. 1, pp. 89–100, 2020, doi: 10.29303/jtika.v2i1.87.

[30] J. Winahyu and I. Suharjo, "Aplikasi Web Analisis Sentimen Dengan Algoritma Multinomial Naïve Bayes," *Kumpul. Artik. Mhs. Pendidik. Tek. Inform.*, vol. 10, no. 2, p. 206, 2021, doi: 10.23887/karmapati.v10i2.36609.

[31] L. Mardiana, D. Kusnandar, and N. Satyahadewi, "Analisis Diskriminan Dengan K Fold Cross Validation Untuk Klasifikasi Kualitas Air Di Kota Pontianak," *BIMASTER*, vol. 11, no. 1, pp. 97–102, 2022.

[32] B. G. Marcot and A. M. Hanea, "What is an optimal value of k in k-fold cross-validation in discrete Bayesian network analysis?," *Comput. Stat.*, vol. 36, pp. 2009–2031, 2020, doi: 10.1007/s00180-020-00999-9.

[33] I. K. Nti, O. Nyarko-Boateng, and J. Aning, "Performance of Machine Learning Algorithms with Different K Values in K-fold CrossValidation," *Int. J. Inf. Technol. Comput. Sci.*, vol. 13, no. 6, pp. 61–71, Dec. 2021, doi: 10.5815/IJITCS.2021.06.05.

[34] F. Tempola, M. Muhammad, and A. Khairan, "Perbandingan Klasifikasi Antara KNN dan Naive Bayes pada Penentuan Status Gunung Berapi dengan K-Fold Cross Validation," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 5, pp. 577–584, 2018, doi: 10.25126/jtiik.201855983.