ANALISIS PERBANDINGAN ALGORITMA NAIVE BAYES DAN SUPPORT VECTOR MACHINE DALAM MENGKLASIFIKASIKAN JUMLAH PEMBACA ARTIKEL ONLINE

Umbar Riyanto
Program Studi Informatika
Fakultas Teknik Universitas Muhammadiyah Tangerang
Jl. Perintis Kemerdekaan 1/33 Cikokol Kota Tangerang
umbar71@yahoo.com

Abstraksi - PT. Linktone Indonesia merupakan salah satu perusahaan yang bergerak dalam bidang portal berita online. Semakin banyaknya portal berita online di Indonesia, para penulis yang ada di PT. Linktone Indonesia harus dapat bersaing, agar artikel yang mereka publish mendapatkan jumlah pembaca yang maksimal. Jumlah pembaca pada sebuah artikel tidaklah menentu, dan sulit untuk diprediksi. Banyaknya jumlah artikel yang dimiliki, maka dapat dilakukan penelitian data mining untuk mengklasifikasi jumlah pembaca artikel. Terdapat beberapa algoritma dalam teknik klasifikasi, akan tetapi tidak semua algoritma memiliki kinerja dan tingkat keakuratan yang baik dalam mengklasifikasi jumlah pembaca artikel. Penelitian ini membandingkan dua algoritma klasifikasi antara Naive Bayes, Support Vector Machine dan Bagging pada tiap algoritma. Peneliti membagi menjadi 5 dataset dan menggunakan tools WEKA dengan tools options K-Folds Cross Validation dan Confussion Matrix. Hasil penelitian ini, dengan jumlah dataset 7111 record. Bagging kurang memperbaiki hasil klasifikasi dengan jumlah dataset yang besar dan memerlukan waktu pembuatan model yang sangat lama dengan klasifikasi Support Vector Machine. Sementara itu Naive Bayes dalam segi waktu pembuatan model mendapatkan waktu yang paling cepat.

Kata Kunci : Klasifikasi, WEKA, Naive Bayes, Support Vector Machine, Bagging

I. PENDAHULUAN

Pada beberapa tahun ini begitu banyak portal berita daring yang tumbuh di Indonesia, salah satunya adalah PT. Linktone Indonesia dengan nama Okezone.com. Para penulis atau jurnalis harus bersaing demi nama dan karir mereka. Kreatif, berfikir luas dan kekauratan berita sangat dituntut dalam penulisan artikel agar dapat menarik minat pembaca yang mengunjungi situs portal berita daring. Jumlah pembaca pada sebuah artikel tidaklah menentu dan sulit untuk diprediksi. Banyaknya jumlah artikel yang dimiliki, maka data tersebut dapat dijadikan sebagai sarana pengambilan keputusan dengan teknik *data mining.*

Penelitian yang dilakukan oleh Sartika dan Dana, dengan melakukan perbandingan algoritma antara Naive Bayes, Nearest Neighbour dan Decision Tree pada kasus pengambilan keputusan pemilihan pola pakaian. Bahwa Decision Tree memiliki tingkat akurasi paling tinggi dengan nilai 75.6% [1]. Menurut penelitian Ariadi and Fithriasari untuk mengkategorikan artikel berita dengan perbandingan algoritma Naive Bayes dan Support Vector Machine, menghasilkan bahwa Support Vector Machine memiliki tingkat akurasi terbaik dengan nilai 88,1% [2]. Penelitian yang dilakukan Pratiwi dalam pengelompokkan siswa otomatis dengan algoritma K-Means, Decision Tree dan Naive Bayes. Menghasilkan nilai yang paling akurat adalah Naive Bayes dengan nilai 70,37% [3]. Penelitian yang dilakukan oleh Tu, M. C., Shin, D. and Shin, D memprediksi diagnosa penyakit jantung dengan algoritma Decision

Tree, Naive Bayes dan Bagging pada tiap algoritmanya, memberikan hasil bahwa Bagging pada tiap algoritma terutama pada Naive Bayes menampilkan hasil yang paling baik selama pengujian dengan nilai 82,50% [4].

Adapun pertanyaan penelitian ini adalah, "Algoritma apakah yang menghasilkan tingkat akurasi paling baik antara Naive Bayes dan Support Vector Machine dalam mengklasifikasikan jumlah pembaca artikel?".

Tujuan penelitian ini adalah membandingkan dan menentukan algoritma yang paling akurat dalam mengklasifikasikan jumlah pembaca artikel dengan membandingkan algoritma Naive Bayes dan Support Vector Machine yang akan di interpretasikan ke dalam prototipe dengan atribut-atribut yang telah ditentukan agar dalam publikasi berikutnya dapat mencapai jumlah pembaca yang ditargetkan oleh manajemen dan penulis pada PT. Linktone Indonesia.

II. LANDASAN TEORI

A. Data Mining

Data mining merupakan serangkaian proses untuk mendapatkan pengetahuan atau pola dari kumpulan data. Data mining memecahkan masalah dengan menganalisa data yang telah ada dalam database [5].
Definisi lain mengatakan "Data mining adalah proses mengekstrak atau menambang pengetahuan yang dibutuhkan dari sejumlah data yang besar." [6].

Pada prosesnya data mining akan mengekstrak informasi yang berharga dengan cara menganalisa adanya pola-pola ataupun hubungan keterkaitan tertentu dari data yang berukuran besar. Tujuan utama dari data mining adalah untuk mengkestrak pola data pada database, meningkatkan nilai intrinsic kemudian mentransfer data untuk menghasilkan knowledge.

Menurut MacLennan dan Tang [7]. Berikut ini adalah fungsi data mining secara umum *Classification, Clustering, Association, Regression, Forecasting, Sequence Analysis, Deviation Analysis*.

Tujuan dari adanya data mining adalah [8]:

1. *Explanatory*, yaitu untuk menjelaskan beberapa kegiatan observasi atau suatu kondisi.
2. *Confirmatory,* yaitu untuk mengkonfirmasi hipotesis yang telah ada.
3. *Explanatory,* yaitu untuk menganalisis data baru suatu relasi yang janggal.

B. Tahapan Data Mining

Sebagai suatu rangkaian proses, data mining dapat dibagi menjadi beberapa tahap, dan tahap-tahap tersebut bersifat interaktif di mana pemakai atau pengguna terlibat langsung atau dengan perantara *knowledge base*. Tahapannya adalah Pembersihan data, Integrasi data, Seleksi data, Transformasi data, Data mining, Evaluasi pola dan Presentasi pengetahuan [6].

C. Klasifikasi

Klasifikasi adalah suatu bentuk analisis yang mengekstrak model yang menggambarkan kelas data penting. Model seperti itu, yang disebut penggolongan, memprediksi label kategori kelas dengan diskrit dan tidak beraturan. Misalnya, kita bisa membangun model klasifikasi untuk mengkategorikan aplikasi pinjaman bank apakah beresiko atau aman. Analisis semacam itu dapat membantu kita untuk lebih memahami data besar lebih luas.

Klasifikasi data terdiri dari dua langkah proses. Pertama adalah fase latih, yaitu dimana algoritma klasifikasinya dibuat untuk menganalisa data training lalu direferensikan dalam bentuk rule klasifikasi. Proses kedua adalah data tes yang digunakan untuk memperkirakan dari rule klasifikasi pada fase latih. Berikut ini adalah daftar algoritma klasifikasi:

1. *Decision Tree*
2. *Naive Bayes Classification*
3. *K-Nearest Neighbors Classifier*
4. *Artificial Neural Network*
5. *Support Vector Machine*

6. *Fuzzy Set Approach*

Dalam penelitian ini akan dilakukan pengujian algoritma antara *Naive Bayes Classification* dan *Support Vector Machine* yang menggunakan data artikel PT. Linktone Indonesia.

D. Naive Bayes

Teori Bayes didasari oleh nama Thomas Bayes, seorang pendeta Inggris nonformis yang melakukan pekerjaan awal dalam teori probabilitas dan keputusan selama abad ke-18 [6].

Naive Bayes merupakan metode yang tidak memiliki aturan, Naive Bayes menggunakan cabang matematika yang dikenal teori *probabilitas* untuk mencari peluang terbesar dari kemungkinan klasifikasi, dengan cara melihat frekuensi tiap klasifikasi pada data mining. Klasifikasi Naive Bayes adalah pengklasifikasian statistik yang dapat digunakan untuk memprediksi probabilitas keanggotaan suatu *class*. Klasifikasi bayesian memiliki kemampuan klasifikasi serupa dengan decision tree dan neural network [6].

E. Support Vector Machine

Support Vector Machine (SVM) adalah sistem pembelajaran yang pengklasifikasiannya menggunakan ruang hipotesis berupa fungsi-fungsi linear dalam sebuah ruang fitur (*feature space*) berdimensi tinggi, dilatih dengan algoritma pembelajaran yang didasarkan pada teori optimasi dengan mengimplementasikan *learning bias* yang berasal dari teori pembelajaran statistik [9].

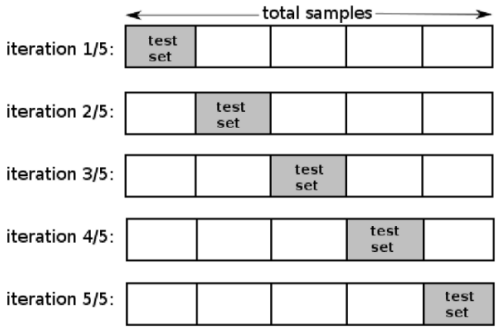
Support Vector Machine adalah suatu teknik untuk melakukan prediksi, baik dalam kasus klasifikasi maupun regresi. SVM memiliki prinsip dasar linier *classifier* yaitu kasus klasifikasi secara linier dapat dipisahkan, namun SVM telah dikembangkan agar dapat bekerja pada problem *non-linier* dengan memasukkan konsep kernel pada ruang kerja berdimensi tinggi. Pada ruang berdimensi tinggi akan dicari fungsi garis pemisah (*hyperplane*) yang dapat memaksimalkan jarak (*margin*) antara kelas data [9].

F. Evaluasi, Validasi dan Improvisasi

1. K-Fold Cross Validation

K-fold cross-validation merupakan salah satu dari variasi teknik pengujian *cross-validation*. *K-fold cross-validation* dilakukan dengan membagi data *training set* dan *test set*. Inti dari validasi untuk tipe ini adalah membagi data secara acak ke dalam himpunan bagian. *K-fold cross-validation* mengulang *k*-kali untuk membagi sebuah himpunan contoh secara acak menjadi *k subset* yang paling bebas. Setiap ulangan disisakan satu *subset* untuk pengujian dan satu *subset* untuk pelatihan.

Data dibagi secara acak menjadi subset atau lipatan, D1,D2,...,Dk, masing-masing berukuran kurang lebih sama. Pelatihan dan pengujian dilakukan k kali. Dalam iterasi i, partisi Di dicadangkan sebagai set tes, dan partisi yang tersisa secara kolektif digunakan untuk melatih model. Artinya, pada iterasi pertama, subset D2,...Dk secara kolektif berfungsi sebagai set pelatihan untuk mendapatkan model pertama yang diuji pada D1, iterasi kedua dilatih pada himpunan bagian D1,D3,...,Dk dan diuji pada D2 dan seterusnya [6].



Gambar 1: Ilustrasi K-Fold Cross Validation

2. Confusion Matrix

Confusion Matrix adalah cara untuk mengevaluasi metode klasifikasi pada bagian akurasi dari hasil klasifikasi. Akurasi sebuah klasifikasi berpengaruh terhadap performa dari suatu klasifikasi. Untuk melakukan analisa dapat digunakan *confusion matrix*

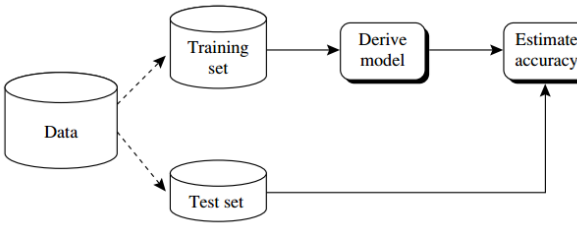
yaitu sebuah matrik dari prediksi yang akan dibandingkan dengan kelas yang asli dari data inputan.
Confusion Matrix adalah alat yang berguna untuk menganalisis seberapa baik *classifier* mengenali *tuple* dari kelas yang berbeda. TP dan TN memberikan informasi ketika *classifier* salah [6]. Berikut ini gambar *confusion matrix.*

		Predicted class		
		yes	*no*	Total
Actual class	*yes*	*TP*	*FN*	*P*
	no	*FP*	*TN*	*N*
	Total	P'	N'	$P+N$

Gambar 2: Confusion Matrix

II. Holdout Method

Holdout Method adalah data yang diberikan secara acak dibagi menjadi dua set independen yaitu *training set* dan *test set*. Biasanya, dua pertiga dari data yang dialokasikan untuk *training set* dan sepertiga sisanya dialokasikan untuk *test set*. *Training set* digunakan untuk mendapatkan model. Akurasi model kemudian diperkirakan dengan *test set*. Perkiraannya pesimis karena hanya sebagian dari data awal yang digunakan untuk mendapatkan model [6].

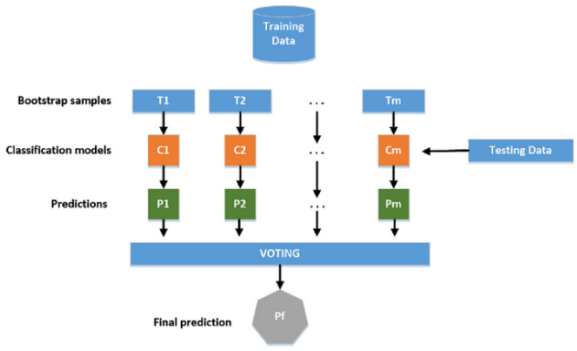


Gambar 3: Holdout Method

III. Bagging

Bagging merupakan metode yang dapat memperbaiki hasil dari algoritma klasifikasi machine learning. Metode ini disimpulkan dari phrase *Bootstrap Aggregating* [10]. Bagging adalah metode *ensamble* yang sederhana namun efektif dan telah diterapkan untuk banyak aplikasi di dunia nyata [11]. Bagging bertujuan untuk meningkatkan akurasi pengklasifikasi dengan menggabungkan pengklasifikasi tunggal dan hasilnya lebih baik daripada random sampling [12].

Berikut ini adalah cara kerja metode bagging:



Gambar 4: Bagging *[12]*

Pada Gambar 4 *bagging* akan membagi data training T menjadi beberapa bagian sejumlah m. Kemudian dibuat model klasifikasi C sejumlah m. Hasil prediksi setiap model P juga akan berjumlah m. Untuk mendapatkan prediksi akhir Pf dilakukan dengan cara voting dari hasil prediksi setiap model klasifikasi C. Ciri utama dari teknik bagging ini adalah setiap model klasifikasi menggunakan algoritma yang sama. Cara voting yang dilakukan untuk memilih prediksi akhir Pf dapat dilakukan dengan dua cara, yaitu [12]:

1) Suara terbanyak, artinya nilai prediksi yang dihasilkan oleh lebih 50% dari jumlah *classifier* yang ada akan dijadikan prediksi akhir Pf.
2) *Bounded minority rule*, cara ini bertujuan untuk menentukan prediksi akhir Pf sebagai class minoritas dan hanya jika seluruh classifier memprediksi sebagai class minoritas.

IV. Slovin

Pertanyaan yang sering diajukan dalam penelitian adalah berapa jumlah sampel yang dibutuhkan dalam penelitian. Sampel yang terlalu kecil dapat menyebabkan penelitian tidak dapat menggambarkan kondisi populasi yang sesungguhnya. Sebaliknya, sampel yang terlalu besar dapat mengakibatkan pemborosan biaya penelitian. Salah satu metode yang digunakan untuk menentukan

jumlah sampel adalah menggunakan rumus slovin, sebagai berikut:

$$n = \frac{N}{1 + Ne^2}$$

Keterangan:
n = Jumlah sampel
N = Jumlah populasi
e = Batas toleransi kesalahan (*error tolerance*)

III. METODOLOGI PENELITIAN

A. Metode Penelitian

Metode penelitian yang akan adalah sebagai berikut:

1. Penelitian eksperimental merupakan penelitian yang bersifat uji coba, dengan berbagai jumlah data sampel yang telah ditentukan dengan metode *slovin* dan dengan mengeliminasi beberapa atribut yang telah ditentukan oleh ahli.
2. Penelitian perbandingan atau studi komparasi dengan membandingkan algoritma Naive Bayes, Decision Tree C4.5 dan Support Vector Machine dan dengan Bagging pada tiap algoritma yang akan diuji dengan menggunakan *tools* WEKA untuk mendapatkan tingkat akurasi yang paling baik.

B. Metode Pemilihan Sampel

Populasi dalam penelitian ini adalah data artikel yang telah diterbitkan pada PT. Linktone Indonesia Okezone.com. Dalam pemilihan sampel perlu dipertimbangkan dengan kesesuaian tujuan dalam penelitian, dimana tujuan dalam penelitian ini adalah pengklasifikasian jumlah pembaca artikel.
Maka rumus slovin dapat ditentukan jumlah artikel berita yang akan dijadikan sampel berdasarkan total jumlah keseluruhan berita 24.619 sebagai berikut:

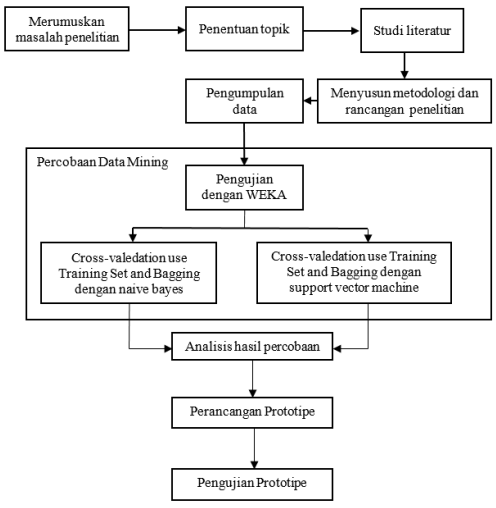
$N = N/(1+Ne^2)$
$= 24.619/(1+ 24.619*0,05^2)$
$= 393,6049 = 394$ artikel berita

C. Teknik Analisis dan Pengujian

Teknik analisa yang akan digunakan adalah data mining dengan algoritma Naive Bayes, Support Vector Machine dan dengan Bagging pada tiap algoritma sebagai peningkat akurasi hasil klasifikasi. Algoritma dengan nilai tingkat akurasi terbaik akan digunakan untuk mengolah data artikel berita guna menghasilkan prediksi jumlah pembaca berita.

Teknik pengujian terhadap algoritma dibandingkan dengan menggunakan tools WEKA, dengan mode pengujian mengevaluasi algoritma melalui *cross-validation*, menggunakan nilai *folds* yang ditentukan. Data yang telah siap di proses akan di konversi ke dalam file berformat .arff dengan 5 jenis dataset, yaitu 394 data, 609 data, 1063 data, 2270 data dan 7111 data. Dimana data tersebut akan di manipulasi atributnya dan dataset akan di uji dengan *Cross-validation.* Dataset tersebut juga akan di uji dengan Bagging. Maka hasil dari evaluasi adalah algoritma yang memiliki tingkat akurasi paling tinggi.

D. Langkah-langkah Penelitian



Gambar 5: Langkah-langkah Penelitian

IV. HASIL DAN PEMBAHASAN

A. Pengumpulan Data

Dalam penelitian ini data yang diperoleh dari objek penelitian akan dianalisa. Data yang dikumpulkan sebanyak 24.619 artikel telah di terbitkan okezone.com. Kemudian dilakukan seleksi data dan pembersihan data, dimana atribut akan dipilih sesuai dengan tujuan data mining, dan didapatkan atribut sebagai berikut:

Tabel 1: Sampel Data Asli

Attribut	Nilai Value
title	Penusuk Polisi di Masjid Falatehan Ikut Bersalaman Usai Salat Berjamaah
channel_name	Nasional
content	(BLOB) 1.42KB
date_created	30/06/2017 22:09
date_publish	30/06/2017 22:10
jenis_berita	Breaking
topic_berita	Terorisme, Kejahatan
hit	8295

B. Transformasi Data

Sebelum data diolah dengan data mining, tahapan selanjutnya adalah proses transformasi. Proses ini mengubah data yang akan diproses untuk memperoleh akurasi dan performa yang baik, proses ini disebut *binning*. *Binning* digunakan untuk mengurangi *noise* data, karena *noise* akan mengurangi performasi algoritma data mining. Transformasi dilakukan dengan membentuk nilai value menjadi interval dengan metode *Rating Scale* yang disesuaikan dengan kaidah yang ada pada PT. Linktone Indonesia dan pemecahan atribut agar dapat memaksimalkan hasil dari pengkalsifikasian jumlah pembaca, berikut ini adalah transformasi data.

Tabel 2: Transformasi Data

Attribut Awal	Attribut	Nilai Value	Keterangan
title	title_word	**Pendek** **Sedang** **Panjang**	Jumlah Kata Dalam Judul Berdasarkan SEO dan Editor PT. Linktone Indonesia
channel_name	channel_name	**Nasional** **Mgapolitan** **International** **Jateng** **Nusantara** **Kampus** **Jatim** **Yogya** **Autos** **Jabar** **News** **News-adv**	Kanal Berita News
content	content_word	**Pendek** **Sedang** **Panjang**	Jumlah Kata Dalam Konten Berdasarkan SEO dan Editor PT. Linktone Indonesia
content	text_artike	**Ya atau Tidak**	Jumlah Media Gambar atau Video
date_publish	publish_day	Hari Pada Tanggal Diterbitkan	Hari Diterbitkan
	publish_time	**Pagi** **Siang** **Sore** **Malam** **Dini Hari**	Waktu Diterbitkan
date_publish - date_created	kehangatan_berita	**Hangat** **Sedang** **Dingin**	Kehangatan Berita Berdasarkan Redaksi PT. Linktone Indonesia
jenis_berita	jenis_berita	**Breaking** **Headline**	Jenis Berita yang diterbitkan
topic_berita	topic_count	**Satu** **Dua** **Tiga** **Lebih Empat**	Jumlah Topic
hit	jumlah_pembaca	**Sedikit** **Banyak** **Sangat Banyak**	Jumlah Pembaca Berita Berdasarkan SEO PT. Linktone Indonesia

C. Percobaan Mining

Langkah-langkah percobaan pada dataset adalah untuk menemukan algoritma yang paling akurat, dalam hal ini penulis melakukan percobaan dengan 5 dataset yang telah ditentukan jumlah sampelnya. Algoritma yang diuji adalah Naive Bayes, Decision Tree dan Support Vector Machine dan dengan bagging pada setiap algoritma.

Test options yang digunakan adalah 5-*Fold Cross Validation*. Selain itu, pengujian dilakukan dengan 10 atribut dan mengeliminasi atribut text_artikel, setelah itu atribut content_word. Setelah semua tahapan percobaan dilakukan maka akan didapat nilai akurasi terbaik.

D. Percobaan Pertama

Percobaan pertama dengan jumlah 10 atribut dan dengan algoritma *Naive Bayes*, *Decision Tree* dan *Support Vector Machine*, dan dengan bagging pada tiap algoritma. Pengujian ini menggunakan test options 5-*fold cross validation*:

Tabel 3: Tingkat Akurasi Percobaan Pertama

Dataset	Pengujian	NB	SVM
7111	Tanpa Bagging	61.44%	61.70%
	Dengan Bagging	61.49%	61.39%
2270	Tanpa Bagging	61.54%	61.40%
	Dengan Bagging	61.32%	60.79%
1163	Tanpa Bagging	60.36%	60.96%
	Dengan Bagging	60.36%	61.65%
609	Tanpa Bagging	57.79%	60.75%
	Dengan Bagging	58.94%	60.59%
394	Tanpa Bagging	55.32%	57.61%
	Dengan Bagging	56.09%	55.58%

Berdasarkan Tabel 3, tingkat akurasi yang paling tinggi diperoleh SVM dengan nilai 61,70% dengan dataset 7111.

Tabel 4: Accuracy, F-Measure dan Time Build Percobaan Pertama

7111 Dataset	TP	FP	Prec	Rcall	F-Mea	Time Build	Accura
NB Tanpa Bagging	0,614	0,483	0,571	0,614	0,557	0,02	61.44%
NB Dengan Bagging	0,615	0,482	0,570	0,615	0,558	0,03	61.49%
SVM Tanpa Bagging	0,617	0,491	0,549	0,617	0,553	18,8	61.70%
SVM Dengan Bagging	0,614	0,490	0,545	0,614	0,553	154	61.39%

Berdasarkan Tabel 4, TP Rate paling tinggi dengan nilai 0,617 diperoleh SVM tanpa bagging dan menurun menjadi 0,614 dengan bagging. FP Rate paling rendah diperoleh NB dengan bagging sebesar 0,614 dan meningkat tanpa bagging dengan nilai 0,615. Nilai Precision dengan bagging mendapatkan hasil yang lebih baik 0,001. Nilai Recall paling tinggi diperoleh SVM tanpa bagging dengan nilai 0,617 dan dengan bagging 0,614. Hal ini menunjukkan bahwa tingkat keberhasilan algoritma dalam menemukan kembali sebuah informasi dengan bagging menurun. Hasil *F-Measure* dengan bagging lebih unggul 0,001 dibanding dengan tanpa bagging. Dilihat dari segi waktu pembuatan model, bagging dengan *base learning* Naive Bayes memperoleh waktu yang paling cepat dengan 0,02 detik, akan tetapi nilai keakuratannya kecil. Sedangkan bagging dengan *base learning* Decision Tree memperoleh waktu pembuatan model 0,05 detik dengan nilai ke akuratan 63,35%.

Tabel 5: Confsion Matrix Decision Tree Percobaan Pertama

Decision Tree Tanpa Bagging			
a	**b**	**c**	**classified as**
3667	558	3	**a = banyak**
1437	837	4	**b = sedikit**
520	84	1	**c = sangat-banyak**

E. Percobaan Kedua

Pada percobaan kedua ini, penulis mengeliminasi atribut text_artikel untuk melihat apakah ada perubahan terhadap hasil akurasi. Total atribut yang akan di uji menjadi 9. Menggunakan algoritma Naive Bayes dan Support Vector Machine, dan bagging pada setiap algoritma. Pengujian ini menggunakan test options 5-*fold cross validation*:

Tabel 6: Tingkat Akurasi Percobaan Kedua

Dataset	Pengujian	NB	SVM
7111	Tanpa Bagging	61.39 %	61.67 %
	Dengan Bagging	61.46 %	61.87 %
2270	Tanpa Bagging	61.23 %	61.40 %
	Dengan Bagging	60.96 %	60.61 %
1163	Tanpa Bagging	60.36 %	60.96 %
	Dengan Bagging	60.27 %	61.47 %
609	Tanpa Bagging	57.79 %	60.59 %
	Dengan Bagging	59.27 %	60.91 %
394	Tanpa Bagging	55.58 %	58.62 %
	Dengan Bagging	56.59 %	55.83 %

Berdasarkan Tabel 6, tingkat akurasi yang paling tinggi diperoleh SVM dengan nilai 61,67% dengan dataset 7111 dan mengalami kenaikan nilai dengan *bagging* menjadi 61,87%.

Tabel 7: Accuracy, F-Measure dan Time Build Percobaan Kedua

7111 Dataset	TP	FP	Prec	Rcall	F-Mea	Time Build	Accura
NB Tanpa Bagging	0,614	0,485	0,567	0,614	0,556	0,01	61.39%
NB Dengan Bagging	0,615	0,483	0,566	0,615	0,557	0,03	61.46%
SVM Tanpa Bagging	0,617	0,490	0,548	0,617	0,554	14,35	61.67%
SVM Dengan Bagging	0,619	0,473	0,550	0,619	0,564	143	61.87%

Berdasarkan tabel 7, TP Rate paling tinggi deng\an nilai 0,619 diperoleh SVM dengan bagging dan naik menjadi 0,619 dengan bagging. FP Rate paling rendah diperoleh NB tanpa bagging sebesar 0,614 dan meningkat dengan bagging dengan nilai 0,615. Precision dengan nilai yang paling tinggi diperoleh NB dengan bagging dengan nilai 0,567. Nilai Recall paling tinggi diperoleh SVM dengan bagging dengan nilai 0,614 dan NB nilai terendah dengan bagging 0,614. Hasil ini menunjukkan bahwa tingkat keberhasilan algoritma dalam menemukan kembali sebuah informasi dengan bagging menurun. Hasil *F-Measure SVM* dengan bagging lebih unggul 0,010 dibanding dengan tanpa bagging. Dari segi waktu pembuatan model, Naive Bayes tanpa bagging memperoleh waktu yang paling cepat dengan 0,01 detik, akan tetapi nilai keakuratan kecil dengan nilai 61,39%.

Tabel 8: Confusion Matrix Decision Tree Percobaan Kedua

Decision Tree Tanpa Bagging			
a	**b**	**c**	**classified as**
3653	571	4	**a = banyak**
1422	853	3	**b = sedikit**
516	87	2	**c = sangat-banyak**

F. Percobaan Ketiga

Pada percobaan ketiga ini, mengeliminasi satu lagi atribut untuk melihat apakah ada perubahan terhadap hasil perhitungan. Atribut yang dieliminasi adalah content_word_count, sehingga total atribut yang akan di uji adalah 8. Menggunakan algoritma Naive Bayes, Decision Tree dan Support Vector Machine, dan bagging pada tiap algoritma. Pengujian ini menggunakan test options 5-*fold cross validation*.

Tabel 9: Tingkat Akurasi Percobaan Ketiga

Dataset	Pengujian	NB	SVM
7111	Tanpa Baging	61.44 %	61.72 %
	Dengan Baging	61.42 %	61.69 %
2270	Tanpa Baging	61.32 %	61.40 %
	Dengan Baging	61.23 %	60.74 %
1163	Tanpa Baging	60.36 %	60.87 %
	Dengan Baging	60.27 %	61.56 %
609	Tanpa Baging	59.27 %	60.59 %
	Dengan Baging	59.44 %	60.75 %
394	Tanpa Baging	55.83 %	59.13 %
	Dengan Baging	56.09 %	58.12 %

Berdasarkan Tabel 9, tingkat akurasi yang paling tinggi diperoleh *Decision Tree* dengan nilai 63,39% dengan dataset 7111. Dengan *bagging*, tingkat akurasi yang dihasilkan 63.11%.

Tabel 10: Accuracy, F-Measure dan Time Build Percobaan Ketiga

7111 Dataset	TP	FP	Prec	Rcall	F-Mea	Time Build	Accura
NB Tanpa Bagging	0,614	0,485	0,564	0,614	0,556	0	61.44%
NB Dengan Bagging	0,614	0,483	0,561	0,614	0,556	0,02	61.42%
SVM Tanpa Bagging	0,617	0,490	0,549	0,617	0,554	12,31	61.72%
SVM Dengan Bagging	0,617	0,483	0,548	0,617	0,558	119,4	61.69%

Berdasarkan Tabel 10, TP Rate paling tinggi dengan nilai 0,617 diperoleh SVM dengan bagging maupun tanpa bagging. FP Rate paling rendah diperoleh SVM dengan bagging sebesar 0,483 dan meningkat tanpa bagging dengan nilai 0,490. Precision dengan nilai yang paling tinggi diperoleh NB dengan bagging dengan nilai 0,564. Recall paling tinggi diperoleh SVM dengan maupun tanpa bagging dengan nilai 0,617. Hasil ini menunjukkan bahwa tingkat keberhasilan algoritma dalam menemukan kembali sebuah informasi dengan bagging menurun. Hasil *F-Measure SVM* dengan bagging lebih unggul 0,004 dibanding dengan tanpa bagging. Waktu pembuatan model, Naive Bayes tanpa bagging memperoleh waktu yang paling cepat yaitu 0 detik, akan tetapi nilai ke akuratan kecil dengan nilai 61,44%.

Tabel 11: Confusion Matrix Decision Tree Percobaan Ketiga

Decision Tree Tanpa Bagging			
a	**b**	**c**	**classified as**
3697	529	2	**a = banyak**
1466	809	3	**b = sedikit**
519	84	2	**c = sangat-banyak**

G. Analisa Hasil Percobaan

Hasil yang dapat disimpulkan dari perbandingan klasifikasi jumlah pembaca yang telah dilakukan dengan tiga percobaan dan dengan lima dataset yang berbeda adalah sebagai berikut:

Tabel 12: Perbandingan Hasil 8 dan 9 Atribut Decision Tree

	TP	FP	Prec	Rcall	F-Mea	Time Build	Accuracy
9 Atribut	0,634	0,443	0,588	0,634	0,587	0,02	63.39%
8 Atribut	0,634	0,450	0,593	0,634	0,584	0,02	63.39%

Hasil dari Tabel 12 menunjukkan bahwa nilai TP Rate keduanya sama. FP Rate pada 9 atribut lebih kecil dibanding FP Rate 8 atribut, menunjukkan bahwa data salah diprediksi sebagai data benar dengan nilai 0,443 dan 0,450. Untuk nilai *F-Measure* tertinggi diperoleh 9 atribut sebesar 0,587, 8 atribut 0,584. Algoritma yang terbaik dalam tingkat akurasi adalah yang memiliki tingkat akurasi dan *f-measure* yang tinggi.

KESIMPULAN

Hasil dari penelitian dapat disimpulkan, algoritma dengan tingkat akurasi terbaik dalam pengklasifikasian jumlah pembaca adalah SVM tanpa bagging dengan memperoleh nilai 63,39% dan 9 atribut. Pengujian dengan bagging mengalami penurunan nilai akurasi pada percobaan 10 atribut, Naive Bayes dengan nilai 57,10% menjadi 58,37% dan 55,32% menjadi 56,09%. Akan tetapi, bagging menghasilkan tingkat akurasi yang cukup baik dengan jumlah atribut 9 dan 8. untuk Naive Bayes dan 60,59% menjadi 60,91% untuk Support Vector Machine. Hasil pada waktu pembuatan model disetiap percobaan, Naive Bayes selalu mendapatkan waktu yang paling cepat dan Support Vector Machine yang paling lambat. Evaluasi dengan *confusion matrix* terhadap dua percobaan terbaik mendapatkan kesimpulan bahwa data dengan kriteria sangat-banyak yang kategorikan sangat-banyak sangat sedikit, sehingga perlu ada penambahan atribut atau pengkajian terhadap data yang berkategori sangat-banyak.

REFERENSI

[1] Sartika, D. dan Indra, D. (2017) 'Perbandingan Algoritma Klasifikasi Naive Bayes , Nearest Neighbour , dan Decision Tree pada Studi Kasus Pengambilan Keputusan Pemilihan Pola Pakaian', Jurnal Teknik Informatika dan Sistem Informasi (JATISI), 1(2), pp. 151–161.

[2] Ariadi, D. dan Fithriasari, K. (2015) 'Klasifikasi Berita Indonesia Menggunakan Metode Naive Bayesian Classification dan Support Vector Machine dengan Confix Stripping Stemmer', JURNAL SAINS DAN SENI ITS Vol. 4, No.2, 4(2), pp. 248–253.

[3] Pratiwi, O. N. (2016) 'Analisa Perbandingan Algoritma K-Means, Decision Tree, Dan Naive Bayes Untuk Sistem Pengelompokkan Siswa Otomatis', II(2).

[4] Tu, M. C., Shin, D. and Shin, D. (2009) 'A Comparative Study of Medical Data Classification Methods Based on Decision Tree and Bagging Algorithms', in 2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing, pp. 183–187. doi: 10.1109/DASC.2009.40

[5] Witten, I. H., Frank, E. and Hall, M. A. (2011) 'Tutorial Exercises for the Weka Explorer', in Data Mining: Practical Machine Learning Tools and Techniques, pp. 559–585. doi: 10.1016/B978-0-12-374856-0.00017-1

[6] Jiawei Han, M. K. A. J. P. (2012) 'Data Mining: Concepts and Techniques, Third Edition – Books24x7', Morgan Kaufmann Publishers, p. 745. doi: 10.1002/1521-3773(20010316)40:6

[7] MacLennan, J. and Tang, Z. H. (2011) Data mining with Microsoft SQL server 2008,

Cancer gene therapy. doi: 10.1038/cgt.2011.47.
[8] Hoffer, J. A., Ramesh, V. and Topi, H. (2015) Modern Database Management, Modern Database Management. doi: 10.1017/CBO9781107415324.004
[9] Cristianini, N. and Shawe-Taylor, J. (2000) 'An Introduction to Support Vector Machines and other kernel based learning methods', Ai Magazine, p. 190. doi: citeulike-article-id:114719
[10] Breiman, L. (1994) Bagging predictors: Technical Report No. 421, Machine Learning
[11] Liang, G., Zhu, X. and Zhang, C. (2014) 'The effect of varying levels of class distribution on bagging for different algorithms: An empirical study', International Journal of Machine Learning and Cybernetics, 5(1), pp. 63–71. doi: 10.1007/s13042-012-0125-5
[12] Alfaro, E., Gamez, M. and García, N. (2013) 'adabag : An R Package for Classification with Boosting and Bagging', Journal of Statistical Software, 54(2), pp. 1–35. doi: http://dx.doi.org/10.18637/jss.v054.i02
[13] Oktafia, D. and Pardede, D. D. L. C. (2008) 'Perbandingan Kinerja Algoritma Decission Tree Dan Naive Bayes Dalam Memprediksi Kebangkrutan', 2008, p. 2008.
[14] Duriqi, R., Raca, V. and Cico, B. (2016) 'Comparative Analysis of Classification Algorithms on Three Different Datasets using WEKA', Embedded Computing (MECO), 2016 5th Mediterranean Conference on. IEEE, pp. 96–101