Klasifikasi Berita *Online* Menggunakan Metode *Support Vector Machine* dan *K- Nearest Neighbor*

Siti Nur Asiyah dan Kartika Fithriasari
Jurusan Statistika, Fakultas MIPA, Institut Teknologi Sepuluh Nopember (ITS)
Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia
e-mail: kartika_f@statistika.its.ac.id, sitinurasiyah@live.com

Abstrak*— Teknologi informasi merupakan salah satu hal yang tidak akan lepas dari kehidupan manusia. Tanpa adanya teknologi, manusia akan kesulitan dalam berkomunikasi dan menyampaikan informasi. Perlu adanya sistem yang secara otomatis yang dapat mengelompokkan berita sesuai dengan kategori berita dengan menggunakan text mining. Dalam penelitian ini, metode yang digunakan dalam klasifikasi adalah SVM dan KNN. KNN memiliki kelebihan dalam hal data training yang cukup banyak. Sebagai komparasi, dalam penelitian ini juga menggunakan SVM karena metode ini merupakan salah satu metode yang banyak digunakan untuk klasifikasi data, khususnya data teks. Kedua metode ini akan dibandingkan untuk mengetahui hasil ketepatan klasifikasi yang paling baik. Hasil dari penelitian ini bahwa SVM kernel linier dan kernel polynomial menghasilkan ketepatan klasifikasi yang paling baik adalah kernel polynomial. Apabila dibandingklan dengan KNN maka SVM lebih baik daripada KNN dengan hasil nilai akurasi, recall, precision dan F-Measure sebesar 93.2%, 93.2%, 93.63% dan 93.14%.

***Kata Kunci*—K-Nearest Neighbor, Support Vector Machine,Text Mining.**

I. PENDAHULUAN

Teknologi informasi meliputi segala hal yang berkaitan dengan proses, penggunaan sebagai alat bantu, dan pengelolaan informasi. Sedangkan teknologi komunikasi adalah segala sesuatu yang berkaitan dengan penggunaan alat bantu untuk memproses dan mentransfer data dari perangkat satu ke perangkat lainnya. Awalnya, banyak instansi menyalurkan informasi kepada masyarakat melalui media televisi, koran, majalah atau radio. Kini, seiring berkembangnya teknologi, informasi disampaikan menggunakan sistem berbasis *web* secara *update.* Kementerian Komunikasi dan Informatika menyatakan bahwa pengguna internet di Indonesia hingga saat ini telah mencapai 82 juta orang. Dengan capaian tersebut, Indonesia berada pada peringkat ke-8 di dunia [1].

Pada umumnya, berita yang disampaikan dalam *website* terdiri dari beberapa kategori seperti berita politik, olahraga, ekonomi, kesehatan, dan lain-lain (sebagai contoh pada website kompas.com, detik.com, dan vivanews.com). Sejauh ini, mengelompokkan berita dalam beberapa kategori tersebut dilakukan oleh editor secara manual. Prosesnya, sebelum diunggah harus terlebih dahulu diketahui isi berita secara keseluruhan untuk selanjutnya dikelompokkan dalam kategori yang tepat. Jika jumlah artikel berita yang diunggah semakin banyak, hal ini akan merepotkan bagi pengunggah berita. Terlebih jika dokumen sangat banyak dengan kategori yang cukup beragam. Hal tersebut akan menjadi beban kerja editor dalam mengelompokkan kategori berita. Permasalahan lain muncul ketika dokumen yang akan dikelompokkan dalam masing-masing kategori memiliki kemiripan isi. Hal ini membutuhkan ketelitian dan waktu yang tidak sebentar dalam sistem pengelompokkan. Oleh karena itu, perlu adanya sistem yang secara otomatis dapat mengelompokkan berita sesuai dengan kategori berita dengan menggunakan *text mining*.

Text mining merupakan salah satu cabang ilmu *data mining* yang menganalisis data berupa dokumen teks. Menurut Han, Kamber, dan Pei (dalam Prilianti dan Wijaya, 2014), *text mining* adalah satu langkah dari analisis teks yang dilakukan secara otomatis oleh komputer untuk menggali informasi yang berkualitas dari suatu rangkaian teks yang terangkum dalam sebuah dokumen. Ide awal pembuatan *text mining* adalah untuk menemukan pola-pola informasi yang dapat digali dari suatu teks yang tidak terstruktur [2]. Sebelum suatu data teks dianalisis menggunakan metode dalam *text mining* perlu dilakukan *pre processing text* diantaranya adalah *tokenizing*, *case folding*, *stopwords*, dan *stemming*. Setelah dilakukan *pre processing* maka selanjutnya dilakukan metode klasifikasi dalam mengelompokkan dalam masing-masing kategori. Klasifikasi merupakan suatu metode untuk memprediksi kategori atau kelas dari suatu item atau data yang telah didefinisikan sebelumnya. Berbagai macam metode klasifikasi banyak digunakan dalam melakukan klasifikasi berupa teks diantaranya adalah *Naïve Bayes Classifier* (NBC), *K-Nearest Neigbour* (KNN), *Artificial Neural Network* (ANN), dan *Support Vector Machines* (SVM).

Penelitian sebelumnya yang berkaitan adalah oleh Ariadi (2015) tentang klasifikasi berita Indonesia menggunakan metode NBC dan SVM dengan *Confix Stripping Stemmer* menghasilkan ketepatan klasifikasi sebesar 88,1%[3]. Selain itu oleh Buana dan Putra (2012) tentang kombinasi KNN dan K-Mean untuk klasifikasi Koran Indonesia menghasilkan ketepatan klasifikasi sebesar 87%[4]. Penelitian tentang *text mining* dilakukan oleh Widhianingsih (2016) tentang aplikasi text mining untuk automatisasi klasifikasi artikel dalam majalah *online* wanita menggunakan NBC dan ANN [5].

Dalam penelitian ini, metode yang digunakan dalam klasifikasi adalah SVM dan KNN. KNN memiliki kelebihan dalam hal data training yang cukup banyak.

Sebagai komparasi, dalam penelitian ini juga menggunakan SVM karena metode ini merupakan salah satu metode yang banyak digunakan untuk klasifikasi data, khususnya data teks. Salah satu kelebihan SVM dapat diimplementasikan relative mudah, karena proses penentuan *support vector* dapat dirumuskan dalam QP problem. Selanjutnya akan dilakukan perbandingan dari kedua metode tersebut pada data berita *online*.

II. TINJAUAN PUSTAKA

A. Text Mining

Text mining merupakan salah satu cabang ilmu *data mining* yang menganalisis data berupa dokumen teks. Menurut Han, Kamber, dan Pei (dalam Prilianti dan Wijaya), *text mining* adalah satu langkah dari analisis teks yang dilakukan secara otomatis oleh komputer untuk menggali informasi yang berkualitas dari suatu rangkaian teks yang terangkum dalam sebuah dokumen[6]. Ide awal pembuatan *text mining* adalah untuk menemukan pola-pola informasi yang dapat digali dari suatu teks yang tidak terstruktur. Dengan demikian, *text mining* mengacu juga kepada istilah *text data mining* atau penemuan pengetahuan dari basis data teks. Saat ini, *text mining* telah mendapat perhatian dalam berbagai bidang, antara lain dibidang keamanan, biomedis, pengembangan perangkat lunak dan aplikasi, media *online*, pemasaran, dan akademik. Seperti halnya dalam *data mining*, aplikasi *text mining* pada suatu studi kasus, harus dilakukan sesuai prosedur analisis. Langkah awal sebelum suatu data teks dianalisis menggunakan metode-metode dalam *text mining* adalah melakukan *pre processing* teks. Selanjutnya, setelah didapatkan data yang siap diolah, analisis *text mining* dapat dilakukan.

B. Pre Processing Text

Tahapan *pre processing* ini dilakukan agar dalam klasifikasi dapat diproses dengan baik. Tahapan dalam *pre processing text* adalah sebagai berikut:

a. *Case Folding*, merupakan proses untuk mengubah semua karakter pada teks menjadi huruf kecil. Karakter yang diproses hanya huruf 'a' hingga 'z' dan selain karakter tersebut akan dihilangkan seperti tanda baca titik (.), koma (,), dan angka.[7]
b. *Tokenizing*, merupakan proses memecah yang semula berupa kalimat menjadi kata-kata atau memutus urutan string menjadi potongan-potongan seperti kata-kata berdasarkan tiap kata yang menyusunnya.
c. *Stopwords*, merupakan kosakata yang bukan merupakan kata unik atau ciri pada suatu dokumen atau tidak menyampaikan pesan apapun secara signifikan pada teks atau kalimat [8]. Kosakata yang dimaksudkan adalah kata penghubung dan kata keterangan yang bukan merupakan kata unik misalnya "sebuah", "oleh", "pada", dan sebagainya.
d. *Stemming*, yakni proses untuk mendapatkan kata dasar dengan cara menghilangkan awalan, akhiran, sisipan, dan *confixes* (kombinasi dari awalan dan akhiran).

C. Term Frequency InverseDocument Frequency

Term Frequency Inverse Document Frequency (TF-IDF) merupakan pembobot yang dilakukan setelah ekstrasi artikel berita. Proses metode TF-IDF adalah menghitung bobot dengan cara integrasi antara *term frequency* (*tf*) dan *inverse document frequency* (*idf*). Langkah dalam TF-IDF adalah untuk menemukan jumlah kata yang kita ketahui (*tf*) setelah dikalikan dengan berapa banyak artikel berita dimana suatu kata itu muncul (*idf*). Rumus dalam menentukan pembobot dengan TF-IDF adalah sebagai berikut:

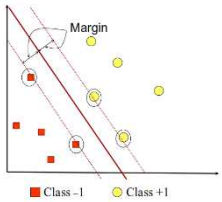
$$w_{ij} = tf_{ij} \times idf$$
$$idf = \log\left(\frac{N}{df_j}\right) \quad (1)$$

dengan : i = 1,2,…,p (Jumlah variabel)
j =1,2,…,N (Jumlah data)

Dimana w_{ij} adalah bobot dari kata i pada artikel ke j, N merupakan jumlah seluruh dokumen, tf_{ij} adalah jumlah kemunculan kata i pada dokumen j, df_j adalah jumlah artikel j yang mengandung kata i. TF-IDF dilakukan agar data dapat dianalisis dengan menggunakan *support vector machine*.

D. Support Vector Machine

Support Vector Machine (SVM) adalah sistem pembelajaran yang menggunakan hipotesis fungsi linear dalam ruang berdimensi tinggi dan dilatih dengan algoritma berdasarkan teori optimasi dengan menerapkan *learning bias* yang berasal dari teori statistik [9]. Tujuan utama dari metode ini adalah untuk membangun OSH (*Optimal Separating Hyperplane*), yang membuat fungsi pemisahan optimum yang dapat digunakan untuk klasifikasi.



Gambar 1. Konsep Hyperplane pada SVM

Data yang berada pada bidang pembatas disebut dengan *support vector*. Dalam Gambar 1, dua kelas dapat dipisahkan oleh sepasang bidang pembatas yang sejajar. Bidang pembatas pertama membatasi kelas pertama sedangkan bidang pembatas kedua membatasi kelas kedua, sehingga diperoleh:

$$\mathbf{x_i w} + b \geq +1, y_i = +1$$
$$\mathbf{x_i w} + b \leq -1, y_i = -1 \quad (2)$$

$\mathbf{w}$ adalah normal bidang dan b adalah posisi bidang alternatif terhadap pusat koordinat. Nilai margin (jarak) antara bidang pembatas (berdasarkan rumus jarak garis ke titik pusat) adalah. $\frac{1-b-(-1-b)}{\|\mathbf{w}\|} = \frac{2}{\|\mathbf{w}\|}$ Nilai margin ini dimaksimalkan dengan tetap memenuhi persamaan (2). Dengan mengalikan b dan $\mathbf{w}$ dengan sebuah konstanta, akan dihasilkan nilai margin yang dikalikan dengan konstata yang sama. Oleh karena itu, konstrain pada persamaan (2) merupakan *scaling constraint* yang dapat dipenuhi dengan *rescaling* b dan $\mathbf{w}$. Selain itu karena memaksimalkan $1/\|\mathbf{w}\|$ sama dengan meminimumkan $\|\mathbf{w}\|^2$ dan jika kedua bidang pembatas pada persamaan (2) direpresentasikan dalam pertidaksamaan (3),

$$y_i\left(\mathbf{x_i w} + b\right) - 1 \geq 0 \quad (3)$$

maka pencarian bidang pemisah terbaik dengan nilai margin terbesar dapat dirumuskan menjadi masalah optimasi konstrain, yaitu:

$$\min \frac{1}{2}\|\mathbf{w}\|^2 \quad (4)$$

$$\text{dengan } y_i\left(\mathbf{x_i}\mathbf{w}+b\right)-1\geq 0$$

Untuk mengklasifikasikan data yang tidak dapat dipisahkan secara linier formula SVM harus dimodifikasi karena tidak akan ada solusi yang ditemukan. Oleh karena itu, kedua bidang pembatas (2) harus diubah sehingga lebih fleksibel dengan penambahan variabel $\xi_i(\xi_i \geq 0, \forall_i: \xi_i = 0$ jika x_i diklasifikasikan dengan benar) menjadi $\mathbf{X_i W} + b \geq 1 - \xi_i$untuk kelas 1 dan $\mathbf{X_i W} + b \geq -1 + \xi_i$ untuk kelas 2. Pecarian bidang pemisah terbaik dengan penambahan variabel ξ_i sering disebut dengan *soft margin hyperplane*. Dengan demikian formula pencarian bidang pemisah terbaik berubah menjadi:

$$\min \frac{1}{2}\|\mathbf{w}\|^2 + C\left(\sum_{i=1}^{n}\xi_i\right) \quad (5)$$

$$\text{dengan } y_i\left(\mathbf{x_i}\mathbf{w}+b\right)\geq 1-\zeta_i$$
$$\xi_i \geq 0$$

C adalah parameter yang menentukan besar penalti akibat kesalahan dalam klasifikasi data dan nilainya ditentukan oleh pengguna. Sehingga peran dari *C* adalah meminimalkan kesalahan pelatihan dan mengurangi kompleksitas model.

Fungsi kernel yang umum digunakan pada metode SVM adalah

1. Kernel Linier
   $K\left(\mathbf{x_i}, \mathbf{x}\right) = \mathbf{x_i^T x}$
2. Kernel Polynomial
   $K\left(\mathbf{x_i}, \mathbf{x}\right) = \left(\gamma \mathbf{x_i^T x} + r\right)^p, \gamma > 0$
3. Kernel Radial Basis Function (RBF)
   $K\left(\mathbf{x_i x}\right) = \exp\left(-\gamma\|\mathbf{x} - \mathbf{x_i}\|^2\right)$
4. Sigmoid Kernel
   $K\left(\mathbf{x_i}, \mathbf{x}\right) = \tanh\left(\gamma \mathbf{x_i^T x} + r\right)$

Dalam penelitian ini memiliki kategori lebih dari 2 atau *multiclass* makan digunakan metode *One Against One* (OAO) dalam menyelesaikan permasalahan tersebut.

E. K-Nearest Neighbor

KNN merupakan salah satu pendekatan yang sederhana untuk diimplementasikan dan merupakan metode lama yang digunakan dalam pengklasifikasian. Menurut Y. Hamamoto, dkk dan E.Alpaydin menyebutkan bahwa KNN memiliki tingkat efisiensi yang tinggi dan dalam beberapa kasus memberikan tingkat akurasi yang tinggi dalam hal pengklasifikasian [10].

Dalam istilah lain, K-*Nearest Neighbor* merupakan salahsatu metode yang digunakan dalam pengklasifikasian. Prinsip kerja K-*Nearest Neighbor* (KNN) adalah melakukan klasifikasi berdasarkan kedekatan lokasi (jarak) suatu data dengan data yang lain [11]. Dekat atau jauhnya lokasi (jarak) bisa dihitung melalui salah satu dari besaran jarak yang telah ditentukan yakni jarak *Euclidean*, jarak *Minkowski*, dan jarak Namun dalam penerapannya seringkali digunakan jarak *Euclidean* karena memiliki tingkat akurasi dan juga *productivity* yang tinggi. Jarak *Euclidean* adalah besarnya jarak suatu garis lurus yang menghubungkan antar objek. Rumus jarak *Euclidean* adalah sebagai berikut:

$$d\left(x_i, x_j\right) = \sqrt{\sum_{n=1}^{p}\left(x_{ip} - x_{jp}\right)^2} \quad (6)$$

Dengan:

x_{ip} = data *testing* ke-*i* pada variabel ke-*p*
x_{jp} = data *training* ke-*j* pada variabel ke-*p*
$d(x_i,x_j)$ = jarak *euclidean*
p = dimensi data variabel bebas

F. Pengukuran Performa

Pengukuran performa dilakukan untuk melihat hasil yang didapatkan dari klasifikasi. Terdapat beberapa cara untuk mengukur performa, beberapa cara yang sering digunakan adalah dengan menghitung akurasi total, *recall,* dan *precission* [12].

$$akurasi\ \ total = \frac{F_{11}+F_{22}+F_{33}+F_{44}+F_{55}}{F_{11}+F_{12}+F_{13}+F_{14}+F_{15}+\cdots+F_{51}+F_{52}+F_{53}+F_{54}+F_{55}}$$

$$recall = \frac{F_{11}}{F_{11}+F_{12}+F_{13}+F_{14}+F_{15}}$$

$$precision = \frac{F_{11}}{F_{11}+F_{21}+F_{31}+F_{41}+F_{51}}$$

$$F = \frac{2\times recall \times precision}{recall + precision}$$

G. K-Fold Cross Validation

K- fold cross validation adalah sebuah teknik yang menggunakan keseluruhan *dataset* yang ada sebagai *training dan testing* [13]. Teknik ini mampu melakukan pengulangan data *training* dan data testing dengan algoritma *k* pengulangan dan partisi 1/*k* dari *dataset*, yang mana 1/*k* tersebut akan digunakan sebagai data testing. Sebagai analogi misalkan keseluruhan *dataset* dibagi menjadi *k* buah subbagian *Ak* dengan ukuran sama, yang mana *Ak* merupakan himpunan bagian dari *dataset*. Kemudian dari data itu dilakukan iterasi sebanyak *k* kali. Pada iterasi ke *k*, subset *Ak* menjadi data *testing*, sedangkan subbagian lain menjadi data *training*. Hal ini ditujukan agar mendapatkan tingkat kepercayaan yang tinggi karena semua *dataset* dilibatkan sebagai data *training* maupun *testing* .

III. METODOLOGI PENELITIAN

A. Sumber Data

Sumber data yang akan digunakan dalam penelitian ini adalah artikel berita pada koran online detik.com yang terdiri dari 5 kategori. Kategori tersebut adalah *news*, *finance*,*hot* , *sport*, dan *oto*. Tiap kategori akan diambil sebanyak 100 artikel sehingga data artikel keseluruhan berjumlah 500 dengan variabel bebas sebanyak 3784 *word vector*. Berikut merupakan struktur data artikel yang telah dilakukan *pre processing*

TABEL 1. STRUKTUR DATA

No	Y	X_1	X_2	…	X_{3784}
1	1	$x_{1,1,1}$	$x_{1,1,2}$	…	$x_{1,1,3784}$
2	1	$x_{2,1,1}$	$x_{2,1,2}$	…	$x_{2,1,3784}$
3	1	$x_{3,1,1}$	$x_{3,1,2}$	…	$x_{3,1,3784}$
⋮	⋮	⋮	⋮	⋮	⋮
500	5	$x_{500,5,1}$	$x_{500,5,2}$	…	$x_{500,5,3784}$

B. Langkah Analisis

Langkah analisis data yang dilakukan pada penelitian ini adalah sebagai berikut.

1. Menyiapkan data artikel

2. Melakukan *pre processing text* yaitu *stemming, stopword, casefolding* dan *tokenizing*.
   a) Proses *stemming* menyiapkan data artikel dalam bentuk excel kemudian dilakukan *running* dengan menggunakan *xampp*
   b) Tahap *stopword* dan *casefolding* yaitu hasil dari *stemming* di *running* menggunakan *software* R. Daftar *stopwords* diambil dari tesis F. Tala yang berjudul "*A Study of Stemming Effect on Information Retrieval in Bahasa Indonesia*".
   c) Pada tahap *tokenizing* hasil dari *casefolding* dilakukan running data pada *software* Weka.
   d) Merubah teks menjadi vector dan pembobotan kata dengan *tf-idf*.
3. Membagi data menjadi data *training* dan data *testing*. Melakukan klasifikasi menggunakan SVM
   a) Menentukan pembobot parameter pada SVM tiap jenis kernel
   b) Membangun model SVM menggunakan fungsi kernel.
   c) Menghitung nilai akurasi dari model yang terbentuk.
4. Melakukan klasifikasi menggunakan KNN
   a) Menentukan nilai *k.*
   b) Menghitung kuadrat jarak *euclid (query instance)* masing-masing objek terhadap *training data* yang diberikan.
   c) Mengumpulkan label *class* Y (klasifikasi *Nearest Neighbor*).
5. Membandingkan performansi antara metode SVM dengan metode KNN berdasarkan tingkat akurasi ketepatan klasifikasi.

IV. HASIL DAN PEMBAHASAN

A. Support Vector Machine

Pada penelitian klasifikasi berita *online* digunakan metode *support vector machine.* Fungsi kernel yang akan digunakan adalah kernel linier dan polynomial. Berikut merupakan pembahasan dari kernel linier dan kernel polynomial. Pada kernel linier digunakan parameter *c* pada rentang 10^{-3} sampai dengan 10^{3} untuk data *training*.

TABEL 2. KETEPATAN KLASIFIKASI SVM KERNEL LINIER PADA DATA *TRAINING*

		Ketepatan Klasifikasi (%)						
	C	0.001	0.01	0.1	**1**	10	100	1000
Word Vector	1009	99.11	100	100	**100**	100	100	100
	1595	98.88	100	100	**100**	100	100	100
	2220	99.31	100	100	**100**	100	100	100
	2595	99.51	100	100	**100**	100	100	100
	3038	99.55	100	100	**100**	100	100	100
	3784	99.51	100	100	**100**	100	100	100

Berdasarkan Tabel 1 dapat diketahui bahwa dengan menggunakan kernel linier untuk setiap *word vector* dengan menggunakan *k-fold cross validation* sebesar 10 *fold* didapatkan nilai ketepatan paling besar 100% pada semua *word vector* 3784 dengan menggunakan *c*=0.01 sampai *c*=1000. Pada *c*=0.001 didapatkan hasil ketepatan klasifikasi yang berbeda-beda. Parameter *c* = 1 akan digunakan pada data testing dengan *word vector* sebanyak 3784. Selanjutnya dilakukan ketepatan klasifikasi pada kernel *polynomial* dengan menggunakan parameter *c* pada rentang 10^{-3} sampai 10^{3} dengan parameter γ=1, *r*=6 dan *p*=3.

TABEL 3. KETEPATAN KLASIFIKASI SVM KERNEL POLYNOMIAL PADA DATA *TRAINING*

		Ketepatan Klasifikasi (%)						
	C	0.001	0.01	**0.1**	1	10	100	1000
Word Vector	1009	93.42	99.28	**100**	100	100	100	100
	1595	85.08	98.64	**100**	100	100	100	100
	2220	82.35	98.84	**100**	100	100	100	100
	2595	81.02	98.93	**100**	100	100	100	100
	3038	79.13	98.95	**100**	100	100	100	100
	3784	79.8	99.08	**100**	100	100	100	100

Tabel 2 menunjukkan bahwa setelah dilakukan percobaan dengan menggunakan *c* 10^{-3} sampai 10^{3} didapatkan hasil pada *c* =0.1 sampai 1000 memiliki nilai akurasi sebesar 100%. Hal ini menunjukkan bahwa pada saat *c* = 0.1 didapatkan nilai akurasi yang sudah konvergen. Selanjutnya digunakan *c*=0.1 untuk digunakan pada data testing.

TABEL 4. PERFORMANSI KERNEL LINIER *10-FOLD* PADA DATA *TESTING*

Fold	Akurasi Total	*Recall*	*Precision*	*F-Measure*
10-Fold	93%	93%	93.41%	92.94%

Tabel 3 menunjukkan bahwa dari hasil pengukuran performansi untuk 10 *fold* didapatkan rata-rata dari akurasi total, *recall*, *precision* dan *F-Measure* sebesar 93%, 93%, 93.41% dan 92.94%. Dari 10 *fold* tersebut diambil *fold* ke 10 untuk melihat performansi akurasi tiap kategori. Berikut merupakan hasil dari pengukuran performansi tiap kategori yang ditampilkan pada Tabel 4

TABEL 5. PERFORMANSI KERNEL LINIER TIAP KATEGORI PADA DATA *TESTING*

	Kategori	*Recall*	*Precision*	*F-Measure*
1	Finance	100%	90.9%	95.23%
2	Hot	100%	90.9%	95.23%
3	News	80%	100%	88.89%
4	Oto	90%	90%	90%
5	Sport	100%	100%	100%
	Rata-rata	**94%**	**94.36%**	**93.87%**

Tabel 4 dapat diketahui bahwa hasil ketepatan klasifikasi dengan menggunakan kernel linier pada *word vector* 3784 didapatkan nilai rata-rata dari 10 *fold* didapatkan *recall*, *precision*, dan *F-Measure* sebesar 94%, 94.36% dan 93.87%. Kategori yang memiliki nilai akurasi sebesar 100% yaitu kategori *finance*, *hot* dan *sport.* Dari tabel tersebut maka dapat diperoleh *confusion matrix* yang ditampilkan pada Tabel 5.

TABEL 6. *CONFUSION MATRIX* KERNEL LINIER

Kelas Asli		Kelas Prediksi				
		a	b	c	d	e
a	News	**8**	0	1	0	1
b	Finance	0	**10**	0	0	0
c	Hot	0	0	**10**	0	0
d	Sport	0	0	0	**10**	0
e	Oto	0	1	0	0	**9**

Tabel 5 menunjukkan bahwa kategori *finance, hot* dan *sport* tidak terdapat kesalahan klasifikasi pada kategori tersebut. Sedangkan pada kategori *news* dan *oto* terdapat artikel berita yang dikategorikan kedalam kategori lainnya terdapat 2 artikel dan 1 artikel. Berikut merupakan hasil

dari pengukuran performansi pada kernel *polynomial* dengan menggunakan parameter *c* yang sudah terpilih yaitu *c*=0.1 dengan $\gamma = 1$, *r*=6 dan *p*=3.

TABEL 7. PERFORMANSI KERNEL POLYNOMIAL *10-FOLD* PADA DATA *TESTING*

Fold	Akurasi Total	*Recall*	*Precision*	*F-Measure*
10-Fold	93.2%	93.2%	93.63%	93.14%

Tabel 6 dapat diketahui bahwa hasil nilai rata-rata *10-fold* didapatkan nilai akurasi total, *recall*, *precision*, dan *F-Measure* sebesar 93.2%, 93.2%, 93.63% dan 93.14%. Untuk melihat performansi tiap kategori diambil *fold* ke 2 agar dapat diketahui tingkat akurasi tiap kategori.

TABEL 8. PERFORMANSI KERNEL POLYNOMIAL TIAP KATEGORI PADA DATA *TESTING*

	Kategori	*Recall*	*Precision*	*F-Measure*
1	Finance	90%	100%	94.74%
2	Hot	100%	90.91%	95.24%
3	News	90%	100%	94.74%
4	Oto	100%	90.91%	95.24%
5	Sport	100%	100%	100%
	Rata-rata	**96%**	**96.36%**	**95.99%**

Tabel 7 menunjukkan bahwa hasil ketepatan klasifikasi dengan menggunakan kernel *polynomial* pada *word vector* 3784 didapatkan rata-rata nilai dari 10 *fold* didapatkan *recall*, *precision*, dan *F-Measure* sebesar 96%, 96.36% dan 95.99%. Kategori *hot, oto* dan *sport* memiliki nilai akurasi 100%. Sedangkan kategori *finance, sport* dan *news* memiliki nilai *precision* sebesar 100%. Selanjutnya didapatkan hasil *confusion matrix* pada Tabel 8.

TABEL 9. *CONFUSION MATRIX* KERNEL POLYNOMIAL

Kelas Asli		Kelas Prediksi				
		a	b	c	d	e
a	News	**9**	0	1	0	0
b	Finance	0	**9**	0	0	1
c	Hot	0	0	**10**	0	0
d	Sport	0	0	0	**10**	0
e	Oto	0	0	0	0	**10**

Tabel 8 menunjukkan bahwa dari 10 artikel berita, kategori *news* terdapat 1 artikel berita yang diklasifikasikan kedalam kategori lain. Sedangkan kategori *finance* terdapat 1 artikel berita yang diklasifikasikan kedalam kategori *finance*. Pada kategori *hot, sport* dan *oto* tidak terdapat artikel berita yang diklasifikasikan kedalam kategori lainnya.

TABEL 10. PENGUKURAN PERFORMANSI SVM

	Akurasi Total	*Recall*	*Precision*	*F-Measure*
Linier	93%	93%	93.41%	92.94%
Polynomial	93.2%	93.2%	93.63%	93.14%

Tabel 9 merupakan hasil dari rata-rata tiap fold untuk tiap nilai dari akurasi total, *recall, precision* dan *F-Measure*. Dapat dilihat bahwa pada kernel tersebut memiliki nilai yang sama baiknya akan tetapi nilai akurasi kernel polynomial lebih tinggi dari pada linier. Untuk dibandingkan dengan KNN maka digunakan SVM dengan menggunakan kernel *polynomial.* Setelah didapatkan kernel *polynomial* lebih baik dari pada linier dengan menggunakan persamaan kernel polynomial $K = (\mathbf{x_i}, \mathbf{x}) = (\gamma\, \mathbf{x_i^T}\, \mathbf{x} + r)^p$ menjadi $K = (x_i, x) = (\gamma \phi(x_i)^T \phi(x) + r)^p$ dengan menggunakan metode *one against one* didapatkan 10 persamaan biner SVM sebagai berikut

SVM Biner kategori 1 vs 2

$$f^{12}(x) = \sum_{i=1}^{62} \alpha_i y_i (1 \times \phi(\mathbf{x_i})^T \phi(\mathbf{x}) + 6)^3 - 0{,}06142692$$

SVM Biner kategori 1 vs 3

$$f^{13}(x) = \sum_{i=1}^{62} \alpha_i y_i (1 \times \phi(\mathbf{x_i})^T \phi(\mathbf{x}) + 6)^3 + 0.13187868$$

SVM Biner kategori 1 vs 4

$$f^{14}(x) = \sum_{i=1}^{62} \alpha_i y_i (1 \times \phi(\mathbf{x_i})^T \phi(\mathbf{x}) + 6)^3 - 0.23493447$$

SVM Biner kategori 1 vs 5

$$f^{15}(x) = \sum_{i=1}^{62} \alpha_i y_i (1 \times \phi(\mathbf{x_i})^T \phi(\mathbf{x}) + 6)^3 - 0.03214075$$

SVM Biner kategori 2 vs 3

$$f^{23}(x) = \sum_{i=1}^{62} \alpha_i y_i (1 \times \phi(\mathbf{x_i})^T \phi(\mathbf{x}) + 6)^3 + 0.19821238$$

SVM Biner kategori 2 vs 4

$$f^{24}(x) = \sum_{i=1}^{62} \alpha_i y_i (1 \times \phi(\mathbf{x_i})^T \phi(\mathbf{x}) + 6)^3 + 0.06061854$$

SVM Biner kategori 2 vs 5

$$f^{25}(x) = \sum_{i=1}^{62} \alpha_i y_i (1 \times \phi(\mathbf{x_i})^T \phi(\mathbf{x}) + 6)^3 - 0.35047705$$

SVM Biner kategori 3 vs 4

$$f^{34}(x) = \sum_{i=1}^{62} \alpha_i y_i (1 \times \phi(\mathbf{x_i})^T \phi(\mathbf{x}) + 6)^3 - 0.35047705$$

SVM Biner kategori 3 vs 5

$$f^{35}(x) = \sum_{i=1}^{62} \alpha_i y_i (1 \times \phi(\mathbf{x_i})^T \phi(\mathbf{x}) + 6)^3 - 0.15152921$$

SVM Biner kategori 4 vs 5

$$f^{45}(x) = \sum_{i=1}^{62} \alpha_i y_i (1 \times \phi(\mathbf{x_i})^T \phi(\mathbf{x}) + 6)^3 + 0.21545623$$

B. K-Nearest Neighbor

Penelitian ini menggunakan 2-NN, 3-NN dan 5-NN untuk dilakukan analisis. Berikut merupakan hasil dari ketepatan klasifikasi KNN dengan menggunakan data training.

TABEL 11. KETEPATAN KLASIFIKASI KNN PADA DATA *TRAINING*

	Akurasi Total	*Recall*	*Precision*	*F-Measure*
2-NN	**83.97%**	**83.97%**	**90.27%**	**87.00%**
3-NN	75.60%	75.60%	87.48%	81.80%
5-NN	68.86%	68.86%	85.97%	76.43%

Tabel 10 dapat diketahui bahwa tingkat akurasi yang tertinggi dengan menggunakan 2-NN didapatkan hasil nilai rata-rata akurasi total,*recall, precision* dan *F-Measure* masing-masing sebesar 83.97%, 83.97%, 90.27% dan 87%. Semakin besar k yang digunakan akan menghasilkan nilai akurasi semakin kecil. Maka akan digunakan 2-NN untuk dilanjutkan kedalam analisis menggunakan data testing. Berikut merupakan hasil dari pengukuran performansi rata-rata *10-fold.*

TABEL 12. PERFORMANSI KNN *10-FOLD* PADA DATA *TESTING*

Fold	Akurasi Total	*Recall*	*Precision*	*F-Measure*
10-Fold	**60%**	**60%**	**81.15%**	**60.15%**

Tabel 11 dengan menggunakan *word vector* 3784 dengan k=2 didapatkan hasil nilai rata-rata akurasi total, *recall, precision,* dan *F-Measure* yaitu sebesar 60%, 60%, 81.15% dan 60.15%. Untuk melihat pengukuran performa tiap kategori maka diambil salah satu *fold* agar didapatkan nilai akurasi tiap kategori. Berikut merupakan performansi per kategori pada *fold* ke 4

TABEL 13. PERFORMANSI KNN TIAP KATEGORI PADA DATA *TESTING*

	Kategori	*Recall*	*Precision*	*F-Measure*
1	Finance	70%	100%	82.35%
2	Hot	100%	38.46%	55.55%

3	News	40%	100%	57.14%
4	Oto	50%	100%	66.67%
5	Sport	80%	100%	88.89%
	Rata-rata	**68%**	**87.69%**	**70.12%**

Tabel 12 dapat diketahui bahwa dengan *word vector* 3784 didapatkan hasil dari nilai rata-rata dari akurasi total, *recall, precision,* dan *F-Measure* sebesar 68%, 68%, 87.69% dan 70.12%. Kategori *hot* memiliki nilai akurasi sebesar 100% sedangkan kategori *news* memiliki nilai akurasi yang paling rendah yaitu sebesar 40%. Pada kategori *hot* memiliki nilai *precision* paling rendah yaitu 38.46%. Selanjutnya akan didapatkan *confusion matrix* pada Tabel 13

TABEL 14.CONFUSION MATRIX KNN

Kelas Asli		Kelas Prediksi				
		a	b	c	d	e
a	News	**4**	0	6	0	0
b	Finance	0	**7**	3	0	0
c	Hot	0	0	**10**	0	0
d	Sport	0	0	2	**8**	0
e	Oto	0	0	5	0	**5**

Berdasarkan Tabel dapat diketahui bahwa kategori *hot* tidak terdapat kesalahan klasifikasi. Pada kategori *news*, artikel yang diklasifikasikan dengan tepat hanya 4 artikel sisanya terdapat kesalahan klasifikasi pada kategori *hot* sebanyak 6 artikel. Kategori *finance* terdapat 7 artikel yang tepat diklasifikasikan. Pada kategori sport terdapat 8 artikel yang tepat diklasifikasikan pada kategori tersebut dan kategori *oto* terdapat 5 artikel yang diklasifikasikan dengan benar.

C. Perbandingan Antara SVM dan KNN

Setelah didapatkan hasil ketepatan klasifikasi pada kedua metode maka langkah selanjutnya adalah membandingkan. Berikut merupakan perbandingan antara kedua metode berdasarkan akurasi total, *precision, recall,* dan *F-Measure*.

TABEL 15. PERBANDINGAN SVM DAN KNN

Metode	Akurasi Total	Recall	Precision	F-Measure
SVM	93.2%	93.2%	93.63%	93.14%
KNN	60%	60%	81.15%	68.90%

Tabel 14 dapat dilihat bahwa dari hasil pengukuran performansi yang dilihat dari akurasi, *precision, recall,* dan *F-Measure* SVM kernel linier lebih baik dari pada KNN. Hasil dari KNN memberikan tingkat akurasi paling kecil dibandingkan dengan metode SVM.

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan analisis dan pembahasan yang telah dilakukan dapat diambil kesimpulan dari penelitian ini. Metode *Support Vector Machine* dengan menggunakan kernel linier dan *polynomial* didapat hasil kernel linier sama baik dengan kernel polynomial pada *word vector* 3784. Untuk dibandingkan dengan hasil KNN digunakan kernel *polynomial* dengan hasil yang didapatkan pada data *testing* untuk masing-masing pengukuran performa nilai rata-rata 10 *fold* didapatkan akurasi total, *recall*, *precision*, dan *F-Measure* sebesar 93.2%, 93.2%, 93.63% dan 93.14%.

Metode *K-Nearest Neighbor* dengan menggunakan 2-NN pada data *testing* dengan *word vector* sebesar 3784 didapatkan hasil dari tiap nilai rata-rata dari 10 *fold* performa akurasi total, *recall, precision,* dan *F-Measure* adalah 60%, 60%, 81.15%, 68.90%.

Perbandingan antara kedua metode SVM dan K-NN didapatkan hasil SVM kernel linier lebih baik dibandingkan dengan K-NN.

B. Saran

Saran untuk penelitian selanjutnya adalah agar didapatkan performansi lebih baik maka menggunakan kernel yang sesuai dengan jenis data. Untuk prediksi kelas pada *multiclass* SVM hanya menggunakan metode *one against one* dimana terdapat metode lainnya seperti *one against all* pada kasus *multiclass*.

DAFTAR PUSTAKA

[1] Kementrian Komunikasi dan Informatika. (2014). *Pengguna Internet Di Indonesia Capai 82 Juta*. Diakses pada 20 Januari 2016, dari URL:http://kominfo.go.id/publikasi/content/detail/3980/kemkominfo-pengguna-internet-di-indonesia-capai-82-juta/0/berita_satker

[2] Hamzah, A. (2012). Klasifikasi Teks dengan Naïve Bayes Classifier (NBC) untuk Pengelompokan Teks Berita dan Abstract Akademis. In Prosiding Seminar Nasional

[3] Ariadi, D. & Fithriasari, K. (2015). Klasifikasi Berita Indonesia Menggunakan Metode Naïve Bayessian Classification dan Support Vector Machine dengan Confix Stripping Stemmer. *Jurnal Sains dan Seni ITS, 4*(2), 2337-3520.

[4] Buana, P. W., & Putra, I. K.G.D. (2012). Combination of K-Nearest Neighbor and K-Means based on Term Re-weighting for Classify Indonesian News. *International Journal of Computer Applications* 11(50).0975-8887.

[5] Widhianingsih, T.D.A. & Fithriasari, K. (2016). Aplikasi Text Mining untuk Automatisasi Klasifikasi Artikel dalamnMajalah *Online* Wanita Menggunakan Naïve Bayessian Classification (NBC) Dan Artificial Neural Network (ANN). *Jurnal Sains dan Seni ITS, 5*(1).

[6] Prilianti, K. R., & Wijaya, H. (2014). Aplikasi Text Mining untuk Automasi Penentuan Tren Topik Skripsi dengan Metode K-Means Clustering. Jurnal Cybermatika, 2(1).

[7] Weiss, S. M. (2010). Text mining: Predictive Methods for Analyzing Unstructured Information. New York: Springer.

[8] Dragut, E., Fang, F., Sistla, P., Yu, C., & Meng, W. (2009). Stop Word and Related Problems in Web Interface Integration. VLDB Endowment.

[9] Cristianini, N., & Shawe-Taylor, J. (2000). An Introduction to Support Vector Machine. Cambridge: Cambridge University Press.

[10] Y. Hamamoto, S. Uchimura, and S. Tomita.(1997) "A Bootstrap Technique for Nearest Neighbours Classifier Design," IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, vol. 19, no. 1, pp. 73-79.

[11] Prasetyo, E. (2012). Data Mining Konsep dan Aplikasi Menggunakan MATLAB. Yogyakarta: ANDI Yogyakarta.

[12] Hotho, A., Nurnberger, A., & Paass, G. (2005). *A Brief Survey of Text Mining.* Kassel: University of Kassel.

[13] Bengio, Y. (2004). No Unbiased Estimator of the Variance of K-Fold Cross-Validation. *Journal of Machine Learning Research 5 (2004) 1089–1105*.