Klasifikasi *Email* Spam dengan Menggunakan Metode *Support Vector Machine* dan k-*Nearest Neighbor*

Shiela Novelia Dharma Pratiwi, Brodjol Sutijo Suprih Ulama
Jurusan Statistika, Fakultas MIPA, Institut Teknologi Sepuluh Nopember (ITS)
Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia
e-mail: brodjol_su@statistika.its.ac.id, shielanoveliadp@gmail.com

***Abstrak*—Akibat dari penggunaan email yang sangat intens dapat menyebabkan dampak positif dan negatif. Hal ini dikarenakan tidak semua orang dapat menggunakan email dengan baik dan diketahui banyak sekali penyalahgunaan email yang berpotensi dapat merugikan perusahaan ataupun individual. Email yang disalahgunakan ini biasa dikenal sebagai spam atau junkmail (email sampah), isi dari email tersebut bisa berupa iklan penjualan produk, penipuan berkedok menang undian atau bahkan virus dan malware. Banyaknya penyalahgunaan email ini menimbulkan kerugian yang cukup besar antara lain dapat meningkatkan data traffic dan menyebabkan kerugian ekonomis yang cukup signifikan, terutama bagi perusahaan. Hal-hal tersebut mendasari dilakukannya penelitian tentang klasifikasi email yang kemudian akan diklasifikasikan berdasar dua kategori utama yaitu email spam dan ham. Pengklasifikasian email pada penelitian ini diselesaikan dengan menggunakan metode SVM dan KNN. Metode SVM atau Support Vector Machine merupakan salah satu metode terbaik yang dapat digunakan dalam masalah klasifikasi pola, sedangkan metode KNN atau k-Nearest Neighbor metode pengklasifikasian yang berdasar pada pengukuran jarak tertangga terdekat yang memiliki performansi yang baik ketika data training yang diberikan sedikit. Beberapa referensi menyebutkan bahwa metode KNN dan SVM akan memberikan hasil ketepatan klasifikasi yang lebih baik bila dikombinasikan dengan teknik partisi data k-fold cross validation (k-fold cv), yang mana pada penelitian ini k yang digunakan adalah 10. Sehingga dari kombinasi antara metode klasifikasi dan teknik partisi diatas didapatkan kesimpulan bahwa kombinasi metode KNN pada k = 3,5,7,9,11 dengan 10-fold cv menghasilkan ketepatan klasifikasi terbaik pada saat k=3 dengan hasil ketepatan klasifikasi sebesar 92.28% dengan error 7.72 % sedangkan kombinasi metode SVM menggunakan kernel linier dan RBF dengan 10-fold cv menghasilkan ketepatan klasifikasi terbaik dengan menggunakan SVM linier dengan ketepatan klasifikasi yang diberikan sebesar 96.6% dengan error 3.4% sehingga disimpulkan metode SVM lebih baik dibanding metode KNN.**

***Kata Kunci*— email, ham, k-fold cross validation, kernel, ketepatan klasifikasi, spam, Support Vector Machine, K-Nearest Neighbor.**

I. PENDAHULUAN

Aplikasi internet sebagai sarana komunikasi dilakukan dengan banyak metode, salah satu metode yang paling sering kita gunakan adalah dengan menggunakan *email*. *Email* atau surat elektronik dan merupakan salah satu penerapan internet dalam bidang komunikasi yang paling digemari karena mudah digunakan, cepat dan berbiaya murah. Berdasarkan penelitian yang dilakukan *Radicati group* jumlah akun *email* tahun 2012 diperkirakan sebanyak 3,3 miliar akun. Dengan rincian 75% pemilik akun adalah perseorangan atau pribadi, sisanya sebanyak 25% digunakan oleh perusahaan dan diprediksi pada tahun 2016 akan menjadi 4,3 miliar akun.

Penggunaan *email* yang sangat intens ini menimbulkan dampak positif dan negatif karena pada kenyataannya tidak semua orang menggunakan *email* dengan baik dan bahkan ada banyak sekali penyalahgunaan *email* sehingga berpotensi untuk merugikan orang lain. *Email* yang disalah gunakan ini biasa kita kenal sebagai spam atau *junkmail* (*email* sampah) yang mana *email* tersebut berisikan iklan, penipuan dan bahkan virus [1].

Pentingnya dilakukan penelitian ini dikarenakan penanganan *email* spam yang efektif tidak hanya dapat mengurangi kerugian perusahaan tetapi juga meningkatkan kepuasan dari pengguna *email* itu sendiri. Penelitian ini pendeteksian *email* spam akan digali dari isi dari *email*, dengan cara menganalisis frekuensi penggunaan kata yang ada pada *email* spam dan ham. Dari frekuensi yang diberikan oleh analisis tersebut kemudian akan dilihat bagaimana karakteristik dari *email* spam dan ham pada *dataset* yang digunakan. Kemudian hasil analisa akan di klasifikasi dengan menggunakan metode SVM dan KNN.

SVM atau *Support Vector Machine* merupakan salah satu metode terbaik yang dapat digunakan dalam masalah klasifikasi pola. Chen, Lu dan Huang (2009) mengatakan bahwa SVM merupakan salah satu metode pengklasifikasian yang memberikan hasil terbaik. Sedangkan KNN atau k- *Nearest Neighbor* metode pengklasifikasian yang berdasar pada pengukuran jarak tertangga terdekat serta memiliki performansi yang baik ketika data *training* yang diberikan sedikit [2].

Hal inilah yang mendasari dilakukannya penelitian tentang klasifikasi *email* spam menggunakan metode *Support Vector Machine* (SVM) dan k–*Nearest Neighbour* (KNN) sehingga dapat diketahui hasil performansi yang diberikan oleh kedua metode tersebut.

II. TINJAUAN PUSTAKA

A. Text Mining

Text mining merupakan salah satu cabang ilmu data mining yang mengacu pada pencarian informasi, dengan menganalisis data berupa dokumen teks [3]. Text mining, mengacu pada proses mengambil informasi berkualitas tinggi dari teks [4].

1) Email Spam dan Ham

Email spam dapat dikatakan penyalahgunaan *email* yang dapat merugikan orang lain [1]. Jenis dari *email*

spam ini ada beberapa macam, beberapa diantaranya adalah iklan, *malware* dan *phising*. Sedangkan email ham adalah *legitimate email* yang salah diklasifikasikan sehingga masuk dalam klasifikasi spam.

2) Information Gain

Information gain bisa dianggap masuk ke dalam *preprocessing* teks, tujuan digunakan *information gain* ini adalah untuk mengukur efektivitas suatu atribut dalam mengklasifikasikan data. Tahapan dalam *Information Gain* adalah dengan membuat:

a. ***Wordlist* atau *dictionary***, pembuatan kamus kata berdasar jumlah kata yang ditemukan pada *dataset*
b. ***Feature Selection,*** merupakan suatu pembobot yang dipilih berdasarkan perhitungan bobot *spamicity*, perhitungan bobot *spamicity* memenuhi persamaan berikut.

$$Spamicity(w) = \frac{\Pr(w|S)}{\Pr(w|S) + Pr(w|H)} \quad 2.1$$

Dengan:

Pr(w|S) = Peluang bahwa sebuah kata muncul di email spam.

Pr(w|H) = Peluang bahwa sebuah kata muncul di email ham.

c. ***Feature Vectors Content*** pemilihan 100 jumlah kata dari keseluruhan kata yang muncul pada *wordlist.* Yang mana tiap kata telah diboboti dengan pembobot yang diberi nama *feature selections*. Pencarian 100 kata ini disebut tahapan *selection of words* dan dilanjutkan mencari frekuensi dari 100 kata tersebut dalam *word frequencies*.

3) Selection of Word

Tahapan ini merupakan proses *ranking* untuk tiap kata pada berdasar bobot tertinggi [8]. Kriteria bobot tertinggi disini adalah jika bobot *spamicity* lebih tinggi tau kurang dari 0.5 (0.5 sebagai acuan) dengan penjelasan jika bobot *spamicity* mendekati nilai 1, maka bisa dikatakan kata tersebut merupakan indikator spam yang baik. Sebaliknya, jika bobot *spamicity* mendekati nilai 0, maka bisa dikatakan kata tersebut merupakan indikator ham yang baik. Namun ada kalanya nilai bobot *spamicity* ini menghasilkan nilai yang sangat kecil (jauh dari 0.5) sehingga jika hanya melihat dari *spamicity* tidaklah cukup, hal ini dapat ditanggulangi dengan melihat besar selisih absolut mengikuti persamaan berikut.

$$D = |\Pr(w|S) - \Pr(w|H)| \quad 2.2$$

Dari perhitungan dengan menggunakan persamaan 2.2 akan didapat nilai *spamicity* yang lebih besar dari 0 tetapi kurang dari 1.

4) Word Frequencies

Untuk menyelesaikan klasifikasi pada *dataset* ini, perhitungan frekuensi kata yang memiliki bobot terbesar sangat dibutuhkan. Sehingga perhitungan frekuensi kemunculan kata selain digunakan untuk melihat karakteristik *email* juga digunakan untuk menyelesaikan perhitungan dengan metode SVM dan KNN pada penelitian ini. Kesalahan awal sejak pemilihan kata dan perhitungan frekuensi pada tahap *information gain* dapat berakibat pada tidak terklasifikasinya *dataset* dengan baik dan kesalahan klasifikasi yang semakin buruk (*missclassification*)

B. N-fold Cross Validation

N- fold cross validation adalah sebuah teknik partisi data yang menggunakan keseluruhan *dataset* yang ada sebagai *training dan testing* [9]. Teknik ini mampu melakukan pengulangan data *training* dan data testing dengan algoritma N pengulangan dan partisi 1/N dari *dataset*, yang mana 1/N tersebut akan digunakan sebagai data testing.

C. K-Nearest Neighbor (KNN)

Metode *Nearest Neighbor* adalah metode pengklasifikasian yang berdasar jarak tetangga terdekat (*nearest neighbor*). Konsep dari KNN adalah mencari nilai dari k buah data *training* yang jaraknya paling dekat dengan data baru yang labelnya belum diketahui (bisa juga disebut data *testing*). Yang mana untuk mendapatkan besaran jarak ini dihitung dengan menerapkan perhitungan jarak *minkowski*, jarak mahalanobis dan jarak *euclidean* [10].

$$d(P_i, P_j) = \sqrt{\sum_{n=1}^{k} (P_{ik} - P_{jk})^2} \quad 2.3$$

Dengan

P_{ik} = data *testing* ke-*i* pada variabel ke-k
P_{jk} = data t*raining* ke-j pada variabel ke-k
$d(P_i, P_j)$ = jarak *euclidean*
k = dimensi data variabel bebas

D. Support Vector Machine (SVM)

Support Vector Machine pertama kali diperkenalkan oleh Vapnik pada tahun 1992 sebagai rangkaian harmonis konsep-konsep unggulan dalam bidang pattern recognition [11]. SVM adalah metode *learning machine* yang bekerja atas prinsip *Structural Risk Minimization* (SRM) dengan tujuan menemukan *hyperplane* terbaik yang memisahkan dua buah *class* pada *input space* [12]. Usaha untuk mencari lokasi *hyperplane* ini merupakan inti dari proses pembelajaran pada SVM [13]. Pada penelitian ini fungsi yang digunakan untuk mencari *hyperplane* memenuhi persamaan berikut.

$$f(x) = \sum_{i=1}^{SV} \alpha_i . K(SV_i, x) + b \quad 2.4$$

Dengan keterangan :

α_i = vektor koefisien untuk *lagrange multiplier*
$K(SV_i, x)$ = fungsi kernel yang digunakan
b = error atau bias

E. Pengukuran Performansi Klasifikasi

Untuk mengevaluasi kinerja *classifier* dilakukan dengan cara mengukur ketepatan klasifikasi, *precision* dan *recall* [14]. Nilai ketepatan klasifikasi menggambarkan total klasifikasi data yang diklasifiksikan secara benar oleh *classifier.* Yang mana semakin tinggi nilai ketepatan klasifikasi yang diberikan, maka hasil klasifikasi akan semakin bagus dan akurat. Selain menghitung ketepatan klasifikasi evaluasi dengan menggunakan nilai *precision* dan *recall* dapat dilakukan Hal ini dikarenakan terkadang, nilai ketepatan klasifikasi kurang bisa merepresentasikan performa model secara signifikan [3].

III. METODOLOGI PENELITIAN

A. Sumber Data

Sumber data yang digunakan dalam penelitian ini merupakan *email* berbahasa inggris sebanyak 6000 *email*, 1500 ham dan 4500 spam, dengan format eml. Data ini

merupakan kumpulan *email* spam dan ham dari beberapa perusahaan. Data ini didapatkan dari *website Csmining Group* yaitu sebuah lembaga pemerhati *email*.

B. Langkah Analisis

Langkah analisis yang dilakukan pada penelitian ini adalah sebagai berikut.

1. Menyiapkan dan mengumpulkan *email*
2. *Text Preprocessing*
   a. *Information gain* digunakan untuk mengukur efektivitas suatu atribut dalam mengklasifikasikan data. Tahapan dalam *Information Gain* adalah dengan membuat :
      i. *Wordlist* atau *dictionary*
      ii. *Feature Vectors Content* (*selections of words* dan *word frequencies*)
3. Mendapatkan hasil *email* yang sudah di-*preprocessing*
4. Melakukan pengacakan data dengan menggunakan 10-*fold cross validation*.
5. Menganalisis karakteristik dari *email* spam dan ham.
6. Klasifikasi *email* menggunakan metode *k-nearest neighbor* dengan k = 3, 5, 7, 9, 11 dan membandingkan hasil klasifikasi yang diberikan oleh k yang berbeda pada KNN
7. Klasifikasi *email* menggunakan metode *support vector machine* dengan SVM linier dan kernel RBF serta membandingkan hasil klasifikasi yang diberikan oleh SVM *linier* dan kernel RBF
8. Melakukan uji pada 999 data *testing* dengan metode yang memberikan hasil klasifikasi terbaik dan mengambil analisis.
9. Membandingkan performansi metode SVM dan KNN berdasarkan tingkat ketepatan klasifikasi, *precision* dan *recall*

IV. ANALISIS DAN PEMBAHASAN

A. Text Preprocessing dan Karakteristik email

Tahapan ini adalah tahap *screening* data yang bertujuan mendapatkan karakteristik *email* dengan melihat kata yang muncul pada *email* dan frekuensi kemunculan kata dan untuk meningkatkan ketepatan klasifikasi dari data. Proses *case folding*, *stemming, stopwords dan tokenizing* tidak dilakukan. Hal ini dikarenakan kata penghubung, kata yang bukan kata dasar, dan kata yang tidak relevan bisa menjadi variabel yang signifikan dalam penelitian pada penelitian ini. sehingga tahapan yang dilakukan hanyalah *information gain*, yang meliputi pembuatan *wordlist*. Dengan menggunakan bantuan *python*, *sublime text* dan *R* didapat hasil seperti pada Tabel 4.1

TABEL 4.1 KUMPULAN MACAM KATA YANG DITEMUKAN DALAM *WORDLIST*

Kata	Kata	Kata	Kata	Kata	Kata
subject	can	re	number	best	meeting
to	any	only	were	report	lokay
the	more	;	name	click	full
...	...	...	...	...	...
...	...	...	...	...	...
all	contact	order	Then	market	keep
has	like	when	Take	note	visit

Dari Tabel 4.1 ditemukan terdapat 18422 macam kata yang muncul pada keseluruhan *dataset*. Selanjutnya tahapan berikutnya yaitu *feature vectors content* yaitu pemilihan 100 jumlah kata yang didasarkan pada pembobotan *feature selections*. Untuk menyelesaikan tahapan *feature selections*, digunakan *software Sublime Text* (*Unregistered* Vesion) dengan menambahkan formula pembobot yang memenuhi persamaan (2.2) dan persamaan (2.3) pada bab II. Tahapan ini juga mencakup perhitungan *selection of* word sehingga hasil *feature vectors content* dapat dilihat pada Tabel 4.2.

TABEL 4.2 HASIL FEATURE VECTORS CONTENT DENGAN BOBOT SPAMICITY

No.	Kata	Pr(w|H)	Pr(w|S)	D	Spamicity
1	enron	0.48	0	0.48	0
2	2001	0.306667	0.01566	0.291007	0.048584
3	please	0.566667	0.290828	0.275839	0.33916
...	...	...	...	...	...
...	...	...	...	...	...
98	continue	0.1	0.194631	0.094631	0.660592
99	out	0.066667	0.161074	0.094407	0.707269
100	price	0.013333	0.107383	0.094049	0.889548

Dari Tabel 4.2 dapat diketahui bahwa Pr(w|H) merupakan nilai probabilitas kemunculan kata tersebut pada *email* ham. Pr(w|S) merupakan nilai probabilitas kemunculan kata tersebut pada *email* spam. D adalah nilai mutlak dari selisih probabilitas *email* spam (Pr(w|S)) dan ham (Pr(w|H)). Sedangkan *spamicity* adalah bobot kecenderungan kata pada data yang digunakan untuk menentukan kelas dari kata tersebut. Jika nilai *spamicity* mendekati nol maka kata tersebut merupakan indikator ham yang baik, sedangkan jika semakin mendekati satu maka merupaka indikator spam yang baik.

TABEL 4.3 HASIL WORD FREQUENCIES

Kata	Frekuensi	Kata	Frekuensi	Kata	Frekuensi
the	49583	market	1467	paso	478
you	15141	year	1195	regards	474
this	13650	message	1163	contract	465
...	...	**...**	...	**...**	...
...	...	**...**	...	**...**	...
with	7852	power	962	markets	444
will	6235	2001	930	point	444
our	5664	best	904	meeting	435

Untuk melihat kata yang menyusun salah satu kelas, dapat dilakukan dengan melihat nilai nilai bobot Pr(w|H) dan Pr(w|S). Jika nilai bobot, baik Pr(w|H) dan Pr(w|S), lebih dari nol tetapi kurang dari satu, maka kata tersebut muncul di kedua *email*. Jika nilai bobot Pr(w|H) = 0 maka kata tersebut hanya ada pada *email* spam dan sebaliknya. Sedangkan frekuensi dari 100 kata yang menjadi variabel ditampilkan pada Tabel 4.3.

B. KNN untuk Klasifikasi email spam dan ham

Setelah dilakukan *text preprocessing* dan pencarian karakteristik email, tahapan berikutnya adalah mencari ketepatan klassifikasi *email* dengan menggunakan metode KNN. K yang akan digunakan pada analisis penelitian ini yaitu 3, 5, 7, 9, 11. Berikut hasil klasifikasi dengan metode KNN.

TABEL 4.4 HASIL KLASIFIKASI DENGAN METODE KNN PADA SEMUA *FOLD*

k-	*Ketepatan Klasifikasi*	*Weighted Precision*	*Weighted Recall*	Error
3	**92.28%**	**92.3%**	**92.3%**	7.72%
5	92.17%	92.3%	92.2%	7.83%
7	91.87%	92.1%	91.9%	8.13%
9	91.75%	92.0%	91.8%	8.25%
11	91.3%	91.6%	91.3%	8.70%

Dari Tabel 4.4 dapat dilihat bahwa nilai ketepatan klasifikasi paling optimal adalah saat k = 3, dengan ketepatan klasifikasi sebesar 0.9228 atau 92.28%. Artinya

peluang *email* spam dan ham yang mengalami klasifikasi hanya sebesar 7.72%. artinya dari 6000 *email* yang salah di klasifikasikan ada sebesar 463 *email*. Sedangkan untuk nilai *precision* adalah tingkat ketepatan antara informasi yang diminta oleh pengguna dengan hasil yang diberikan oleh sistem, dari hasil Tabel 4.4 diketahui bahwa hasil *precision* terbaik adalah 92.3% artinya adalah hasil perhitungan yang diberikan oleh sistem mampu mengklasifikasikan data dengan ketepatan hingga 92.3%.

Sedangkan untuk nilai *recall* adalah ketepatan sistem untuk mengklasifikasikan data dengan benar, dari hasil Tabel 4.4 diketahui bahwa hasil *recall* terbaik adalah 92.3% artinya adalah sistem mampu mengklasifikasikan data dengan benar dengan ketepatan 92.3%.

TABEL 4.5 HASIL *CONFUSION MATRIX* UNTUK 3-NN PADA SEMUA *FOLD*

Confusion Matrix				
		Prediksi		
		Ham	**Spam**	**Total**
Pengamatan	**Ham**	1121	379	**1500**
	Spam	84	4416	**4500**
	Total	**1205**	**4795**	**6000**

Dengan rincian *email* ham yang salah diklasifikasikan sebagai *email* spam ada sebanyak 379 dari total 1500 *email* ham, serta terdapat 84 dari total 4500 *email* spam yang salah diklasifikasikan sebagai *email* ham. Dari hasil pembahasan pada paragraf sebelumnya dapat diketahui bahwa metode 3-NN memberikan hasil yang terbaik dengan error terkecil yaitu 7.72% dan ketepaan klasifikasi sebesar 92.28%. Nilai pada hasil tersebut merupakan nilai yang didapat dari rata-rata keseluruhan semua *fold*. Perlu diketahui bahwa setiap *fold* memiliki jumlah data sebanyak 600 *email* dengan rincian 150 *email* spam dan 450 *email* ham. Hasil pengukuran performansi klasifikasi tiap *folds* ditunjukkan Tabel 4.12.

Dari Tabel 4.12 diketahui bahwa dari 10 *folds* yang digunakan, *fold* yang memberikan hasil paling optimum adalah *fold* ke-8. Nilai tersebut menunjukkan bahwa dari 600 data *email* pada fold ke-8 pada metode 3-NN terdapat 48 *email* yang mengalami misklasifikasi dengan rincian dari 150 *email* ham ada 42 *email* ham yang salah diklasifikasikan sebagai *email* spam dan dari 450 *email* spam ada 6 *email* spam yang salah diklasifikasikan sebagai *email* ham.

TABEL 4.12 PENGUKURAN PERFORMANSI KLASIFIKASI METODE 3-NN TIAP *FOLD*

Folds	**Ketepatan Klasifikasi**	***Precision***	***Recall***	**Error**
1	91.33%	92.24%	91.12%	8.67%
2	91.67%	90.98%	91.84%	8.33%
3	93.50%	93.70%	93.45%	6.50%
4	92.33%	94.83%	91.74%	7.67%
5	91.67%	93.10%	91.32%	8.33%
6	92.33%	91.27%	92.62%	7.67%
7	91.33%	92.24%	91.12%	8.67%
8	**93.67%**	**95.16%**	**93.28%**	**6.33%**
9	93.17%	93.60%	93.05%	6.83%
10	92.00%	93.22%	91.70%	8.00%

C. SVM untuk Klasifikasi email spam dan ham

1) SVM dengan Kernel RBF

Untuk kernel RBF, ada beberapa parameter yang harus diperhatikan yaitu parameter C dan γ (gamma). Parameter C berfungsi sebagai koefisien optimasi sedangkan γ merupakan parameter dari persamaan kernel RBF. Penentuan parameter pada kernel RBF ini sangat penting karena nilai parameter C dan γ yang berbeda akan menghasilkan hasil performansi yang berbeda. Untuk mencari nilai parameter C dan γ yang baik dapat dilakukan dengan menggunakan metode *trial error* yaitu mencobakan nilai parameter C dan γ dari 10^{-2} hingga 10^{2} pada *dataset* yang telah disiapkan sebelumnya [15].

TABEL 4.8 HASIL PERFORMANSI UNTUK MENCARI PARAMETER TERBAIK DENGAN METODE *TRIAL ERROR*

C	**Gamma (γ)**				
	0.01	**0.1**	**1**	**10**	**100**
0.01	75%	75%	75%	75%	75%
0.1	87.98%	77.90%	75.26%	75.23%	76.65%
1	95.18%	87.78%	77.30%	76.65%	76.65%
10	**96.07%**	88.50%	77.45%	76.65%	76.65%
100	95.80%	87.65%	77.45%	76.65%	76.65%

Dari hasil Tabel 4.8 diketahui bahwa untuk mendapatkan hasil klasifikasi yang baik nilai parameter C dan γ yang harus di gunakan adalah C = 10 dan γ = 0.01. sehingga didapat hasil seperti pada tabel 4.9.

TABEL 4.9 HASIL KETEPATAN KLASIFIKASI SVM-KERNEL RBF PADA SEMUA *FOLD*

Ketepatan Klasifikasi	***Weighted Precision***	***Weighted Recall***	**Error**
96.07%	96.10%	96.10%	3.93%

Merujuk Tabel 4.9 didapat hasil bahwa ketepatan klasifikasi yang diberikan oleh SVM dengan kernel RBF adalah sebesar 96.07%. Sedangkan nilai *precision* dan *recall* berturut-turut adalah 96.10% dan 96.10%

TABEL 4.10 HASIL *CONFUSION MATRIX* SVM-KERNEL RBF PADA SEMUA *FOLD*

***Confusion* Matrix**				
		Prediksi		
		Ham	**Spam**	**Total**
Pengamatan	**Ham**	1326	174	**1500**
	Spam	62	4438	**4500**
	Total	**1388**	**4612**	**6000**

Error maksimum yang diberikan sebesar 3.93% artinya dari 6000 *email* email yang salah di klasifikasikan ada sebesar 236 *email*. Merujuk Tabel 4.10 diketahui bahwa *email* ham yang salah diklasifikasikan sebagai *email* spam ada sebanyak 174 dari total 1500 *email* ham. Selanjutnya *email* spam yang salah diklasifikasikan sebagai ham ada sebanyak 62 dari total 4500 *email* spam.

2) SVM Linier

Sama seperti kernel RBF, SVM linier juga memiliki parameter. Parameter yang digunakan adalah parameter C yang mana nilai parameter C yang akan memberikan hasil terbaik adalah C = 1 nilai ini didasarkan pada metode *trial error* dengan mencobakan nilai C dari 1 hingga 10. Hasil metode *trial error* ditunjukkan pada Tabel 4.11

TABEL 4.11 KETEPATAN PARAMETER C DENGAN *TRIAL ERROR*

C	**1**	**2**	**3**	**4**	**5**
Ketepatan	**96.6%**	96.55%	96.57%	96.58%	96.58%

TABEL 4.11 KETEPATAN PARAMETER C DENGAN *TRIAL ERROR* (LANJUTAN)

C	**6**	**7**	**8**	**9**	**10**
Ketepatan	96.58%	96.55%	96.58%	96.55%	96.55%

Berdasarkan hasil pada Tabel 4.11 diketahui bahwa ketepatan tertinggi diberikan oleh C = 1 yaitu sebesar 96.6 %. Sehingga analisis akan dilakukan dengan menggunakan C = 1 agar mendapat hasil yang optimal. Berikut adalah hasil analisis dengan menggunakan SVM linier dengan C = 1.

TABEL 4.12 HASIL KETEPATAN KLASIFIKASI SVM- KERNEL LINIER PADA SEMUA FOLD

Ketepatan Klasifikasi	*Precision*	*Recall*	Error
96.6%	96.6%	96.6%	3.4%

Merujuk Tabel 4.12 didapat hasil bahwa ketepatan klasifikasi yang diberikan oleh SVM dengan SVM Linier adalah sebesar 96.6%. Diketahui bahwa nilai *precision* adalah tingkat ketepatan antara informasi yang diminta oleh pengguna dengan hasil yang diberikan oleh sistem, dari hasil Tabel 4.12 diketahui bahwa hasil *precision* terbaik adalah 96.6% artinya adalah hasil perhitungan sistem mampu mengklasifikasikan data dengan ketepatan hingga 96.6%.

Sedangkan untuk nilai *recall* adalah ketepatan sistem untuk mengklasifikasikan data dengan benar, dari hasil Tabel 4.12 diketahui bahwa hasil *recall* terbaik adalah 96.6% artinya adalah sistem mampu mengklasifikasikan data dengan benar dengan ketepatan 96.6%. Error maksimum yang diberikan sebesar 3.4% artinya dari 6000 *email* email yang salah di klasifikasikan ada sebesar 204 *email*, dengan rincian seperti pada Tabel 4.13.

TABEL 4.13 HASIL *CONFUSION MATRIX* SVM-KERNEL LINIER PADA SEMUA *FOLD*

Confusion Matrix				
		Prediksi		
		Ham	**Spam**	**Total**
Pengamatan	**Ham**	1333	167	**1500**
	Spam	37	4463	**4500**
	Total	**1370**	**4630**	**6000**

Berdasarkan hasil pada Tabel 4.13 diketahui bahwa *email* ham yang salah diklasifikasikan sebagai *email* spam ada sebanyak 167 dari total 1500 *email* ham. Sedangkan untuk *email* spam yang salah diklasifikasikan sebagai ham ada sebanyak 37 dari total 4500 *email* spam.Dari hasil pembahasan pada paragraf sebelumnya dapat diketahui bahwa metode SVM dengan kernel linier memberikan hasil yang terbaik dengan error terkecil yaitu 3.4% dan ketepaan klasifikasi sebesar 96.6%.

Nilai pada hasil tersebut merupakan nilai yang didapat dari rata-rata keseluruhan semua *fold* padahal seharusnya untuk menghitung *hyperplane* baru diperlukan model dari salah satu *fold* yang memberikan hasil paling optimum sehingga pada Tabel 4.14 akan ditunjukkan hasil perhitungan klasifikasi tiap *fold*-nya. Sehingga dapat diketahui *fold* yang memberikan hasil paling optimum dan model terbaik untuk menghitung *hyperplane* baru. Perlu diketahui bahwa setiap *fold* memiliki jumlah data sebanyak 600 *email* dengan rincian 150 *email* spam dan 450 *email* ham. Berikut adalah Tabel 4.14.

TABEL 4.14 PENGUKURAN PERFORMANSI KLASIFIKASI SVM-KERNEL LINIER TIAP *FOLD*

Folds	Ketepatan Klasifikasi	*Precision*	*Recall*	Error
1	96.67%	97.10%	96.54%	3.33%
2	95.33%	97.66%	94.70%	4.67%
3	96.83%	96.45%	96.95%	3.17%
4	**97.67%**	**97.22%**	**97.81%**	**2.33%**
5	96.33%	97.76%	95.92%	3.67%
6	96.67%	97.10%	96.54%	3.33%
7	96.17%	95.68%	96.31%	3.83%
8	96.67%	97.10%	96.54%	3.33%
9	96.33%	97.76%	95.92%	3.67%
10	97.33%	99.26%	96.77%	2.67%

Dari Tabel 4.14 diketahui bahwa dari 10 *folds* yang digunakan, *fold* yang memberikan hasil paling optimum adalah *fold* ke-4. Nilai tersebut menunjukkan bahwa dari 600 data *email* pada fold ke-4 pada metode SVM linier terdapat 14 *email* yang mengalami misklasifikasi dengan rincian dari 150 *email* ham ada 10 *email* ham yang salah diklasifikasikan sebagai *email* spam dan dari 450 *email* spam ada 4 *email* spam yang salah diklasifikasikan sebagai *email* ham.

D. Perbandingan antara Metode KNN dan SVM

Diketahui bahwa untuk metode KNN hasil klasifikasi terbaik adalah saat k = 3 atau 3-NN, sedangkan untuk SVM adalah dengan kernel linier. Tabel 4.15 berikut menunjukkan perbandingan antara kedua metode berdasarkan ketepatan klasifikasi, *precision* dan *recall*.

TABEL 4.15 PERBANDINGAN KETEPATAN KLASIFIKASI ANTARA KNN DAN SVM

Metode	Ketepatan Klasifikasi	*Weighted Precision*	*Weighted Recall*	Error
KNN	92.28%	92.3%	92.3%	7.72%
SVM	**96.6%**	**96.6%**	**96.6%**	**3.4%**

Merujuk hasil dari 4.15 maka untuk semua cara pengukuran performa baik ketepatan klasifikasi, *precision, recall* dan nilai error, SVM linier memberikan performansi yang lebih baik dibandingkan 3-NN. Dengan demikian bisa dikatakan bahwa metode SVM mampu memberikan performa yang lebih baik dibanding KNN.

E. Fungsi Hyperplane pada Metode Terbaik

Fungsi *hyperplane* terbaik yang diberikan oleh metode SVM yang mana *hyperplane* yang diberikan memenuhi persamaan

$$f(x) = \sum_{i=1}^{483} \alpha_i . K(SV_i, x) - 1.0229$$

Dengan fungsi kernel linier memenuhi persamaan

$$K(SV_i, x) = SV_i{}^T \mathrm{x}$$

Sehingga fungsi *hyperplane* menjadi

$$f(x) = \sum_{i=1}^{483} \alpha_i (SV_i{}^T \mathrm{x}) - 1.0229$$

Diketahui bahwa nilai -1.0229 adalah nilai bias atau ρ (*rho*), kemudian α_i merupakan vektor koefisien atau *lagrange multiplier* dari *support vector* yang berukuran (483,1). Sedangkan SV adalah matriks *support vektor* berukuran (483,2), 483 merupakan jumlah SV dan 2 menunjukkan banyak kategori yang merupakan kelas *email* spam atau ham dengan penjelasan jika hasil akhir didapat nilai $f(x) \geq 0$ maka akan termasuk pada kelas *email* ham dan sebaliknya, jika nilai $f(x) \leq 0$ maka akan termasuk pada kelas *email* spam.

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan penelitian yang telah dilakukan pada bab sebelumnya didapatkan kesimpulan sebagai berikut :

1. Banyak macam kata yang hanya menyusun *dataset email* ham ada sebanyak 12536 kata. Banyak macam kata yang hanya menyusun *dataset email* spam ada sebanyak 2804 kata. Sedangkan sisanya sebanyak 3082 kata adalah kata yang terdapat pada kedua *email*.
2. Metode KNN memberikan hasil performansi klasifikasi terbaik saat k = 3 dengan hasil ketepatan klasifikasi, *precision* dan *recall* berturut-turut sebesar

92.28%, 92.3% dan 92.3% serta error sebesar 7.72% dari total 6000 *email*.

3. Metode SVM memberikan hasil klasifikasi terbaik dengan menggunakan kernel linier, nilai yang diberikan untuk ketepatan klasifikasi, *precision* dan *recall* berturut-turut sebesar 96.6%, 96.6% dan 96.6% serta error sebesar 3.4% dari total 6000 *email*.
4. SVM dengan kernel linier mampu memberikan hasil klasifikasi yang lebih baik dibandingkan dengan 3-NN.
5. Diketahui bahwa SVM linier memberikan hasil yang terbaik dengan total *support vector* yang diberikan adalah sebanyak 483 SV dan bias sebesar negatif 1.0229. dengan fungsi hyperplane yang terbentuk adalah seperti berikut.

$$f(x) = \sum_{i=1}^{483} \alpha_i (SV_i{}^T \mathrm{x}) - 1.0229$$

B. Saran

Saran untuk penelitian selanjutnya adalah Disarankan data yang digunakan memiliki proporsi yang seimbang untuk tiap kategori dan membuat *wordlist* 2 kata atau 3 kata serta melakukan *preprocessing text* untuk meningkatkan ketepatan klasifikasi.

DAFTAR PUSTAKA

[1] Suyanto. (2014). *Artificial Intelligence, Searching - Reasoning - Planning - Learning.* Bandung: Informatika Bandung.

[2] Colas, F., & Brazdil, P. (2006). Comparison of SVM and Some Older Classification Algorithms in Text Classification Tasks.

[3] Han, J., Kamber, M., & Pei, J. (2012). *Data Mining : Concepts and Technique.* United States of America: Morgan Kaufmann Publishers for Elsevier.

[4] Saraswati, N. W. (2011). Text Mining dengan Metode Naive Bayes Classifier dan Support Vector Machine untuk Sentiment Analysis.

[5] Rifqi, N., Maharani, W., & Shaufiah. (2011). Analisis dan Implementasi Klasifiksi Data Mining Menggunakan Jaringan Syarah Tiruan dan Evolution Strategis.

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

[7] Metsis, V., Androutsapoulos, I., & Paliouras, G. (2006). Spam Filtering with Naive Bayes – Which Naive Bayes? . *CEAS PAPER* .

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

[9] Deokar, S. (2009). *University of Minnesota Duluth.* Diambil kembali dari CSEEWebsite:http://www.csee.umbc.edu/~tinoosh/cmpe650/slides/K_Nearest_Neighbor_Algorithm.pdf

[10] Cortes, C., & Vapnik, V. (1995). *"Support-Vector Networks," Machine Learning, vol. 20, no. 3, pp. 273-297.* Machine Learning.

[11] Feldman, R., & Sanger, J. (2007). *The Text Mining Hand Book.* New York: Cambridge University Press.

[12] Nugroho, A. S., Witarto, A. B., & Handoko, D. (2003). Support Vector Machine : Teori dan Aplikasinya dalam Bioinformatika.

[13] Miner, G., Nisbet, B., Elder, J., Delen, D., Fast, A., & Hill, T. (2012). *Practical Text Mining and Statistical Analysis for Unstructured Text Data Applications.* United State of America: Academic Press.

[14] Chen, Y.-N., Lu, C.-A., & Huang, C.-Y. (2009). Anti Spam Filter Based on Naive Bayes, SVM and KNN Model.

[15] Huang, C.-M., Lee, Y.-J., Lin, D. K., & Huang, S.-Y. (2007). Model Selection For Support Vector Machines Via Uniform Design. *Computational Statistics & Data Analysis*, 335-346.