LONTAR KOMPUTER VOL. 8, NO. 3, DESEMBER 2017 **p-ISSN 2088-1541**
DOI : 10.24843/LKJITI.2017.v08.i03.p08 **e-ISSN 2541-5832**

Identifikasi Komentar Spam Pada Instagram

Antonius Rachmat Chrismanto[1], Yuan Lukito[2]

Program Studi Informatika, Fakultas Teknologi Informasi, Universitas Kristen Duta Wacana
Jl. Dr. Wahidin Sudirohusodo 5-25, Yogyakarta, Indonesia
[1]anton@ti.ukdw.ac.id
[2]yuanlukito@ti.ukdw.ac.id

Abstrak

Spam pada Instagram (IG) umumnya berupa komentar yang dianggap mengganggu karena tidak berhubungan dengan foto atau video yang dikomentari. Spam pada komentar dapat menyebabkan beberapa dampak negatif seperti menyulitkan untuk mengikuti diskusi pada komentar yang dipenuhi oleh komentar spam dan menyebabkan seseorang tampak populer karena jumlah komentarnya banyak walaupun pada kenyataannya lebih banyak komentar yang berupa spam. Penelitian ini mencoba untuk membangun model yang dapat melakukan identifikasi komentar spam pada IG. Komentar pada IG berbentuk teks, sehingga pada penelitian ini digunakan metode-metode pengolahan teks. Untuk identifikasi digunakan metode Support Vector Machine (SVM). Data komentar yang digunakan pada penelitian ini dikumpulkan dari komentar-komentar pada foto atau video yang dibagikan oleh aktor dan artis Indonesia yang memiliki pengikut (follower) paling banyak di IG. Dari hasil penelitian didapatkan model identifikasi komentar spam dengan metode SVM menghasilkan tingkat akurasi 78,49% yang lebih baik jika dibandingkan dengan model pembanding yang menggunakan metode NB (77,25%). Penelitian ini juga menguji beberapa proporsi data pelatihan yang berbeda-beda dan hasilnya metode SVM tetap lebih baik dibandingkan dengan metode NB. Hasil lain dari penelitian ini adalah tahap pre-processing dan stemming yang harus disesuaikan terutama untuk dukungan terhadap pengolahan karakter-karakter unicode dan simbol-simbol khusus yang banyak ditemukan pada komentar-komentar di IG.

Kata kunci: *Identifikasi Spam, Komentar Spam, Instagram, Naive Bayes (NB), Support Vector Machine (SVM).*

Abstract

Spam on Instagram (IG) is generally a comment that is considered as irritating because it does not relate to the photos or videos which were commented. Spam on comment section can cause some negative impacts such as making it difficult to follow the discussion on the posted status and making someone's photo or video looks very popular, commented by a lot of followers despite the fact that most of the comments are actually spam. This research tries to build a model that can identify spam comments on IG. The comment on IG is in text format, so in this research, we use text processing methods. We use Support Vector Machine (SVM) for spam identification. The comment data used in this study were collected from Indonesian actors and artists who are the most followed accounts in IG. We have tested the spam identification model using SVM method resulted in 78.49% of accuracy. This result is better than the baseline model using NB method (77.25%). This research also tested some of the different training data proportions and SVM remains better than NB. Another result of this research are some adaptations needed for preprocessing and stemming stages that must be customized to support Unicode characters and unique symbols that commonly found in IG comments section.

Keywords: *Spam Identification, Spam Comment, Instagram, Naive Bayes (NB), Support Vector Machine (SVM).*

1. Pendahuluan

Instagram (IG) merupakan media sosial berbasis foto/gambar terpopuler di dunia nomor 1, dan di urutan ke-6 untuk media sosial secara umum. Instagram dapat digunakan oleh siapapun tidak terkecuali oleh publik figur. Publik figur, terutama artis dan aktor banyak sekali yang menggunakan IG untuk berbagai keperluan terutama untuk berbagi mengenai aktivitas mereka, promosi, menjalin dan menjaga relasi dengan para penggemarnya. Dengan jumlah pengguna mencapai 500 juta serta 95 juta gambar & video yang diunggah setiap harinya, tentu hal ini sangat bermanfaat bagi para publik figur sebagai sarana promosi mereka.

Para artis/aktor Indonesia juga tidak ketinggalan dengan menggunakan IG agar memperoleh banyak *follower*. Beberapa artis bahkan memiliki *follower* lebih dari 10 juta akun [1]. Para penggemar yang mem-*follow* artis idola tentu dapat memberikan *like* dan komentar pada setiap status terbaru yang dibuat oleh artis tersebut. Sayangnya tidak semua komentar pada status adalah komentar yang berkaitan dengan status yang dibagikan, banyak sekali komentar-komentar yang disebut komentar spam yang dibuat oleh para *spammer* yang jelas-jelas tidak berkaitan dengan status yang dibagikan. Para *spammer* menuliskan berbagai komentar tentang bisnis mereka (promo/berjualan), atau link spam, dan berbagai hal lain yang tentu sangat mengganggu.

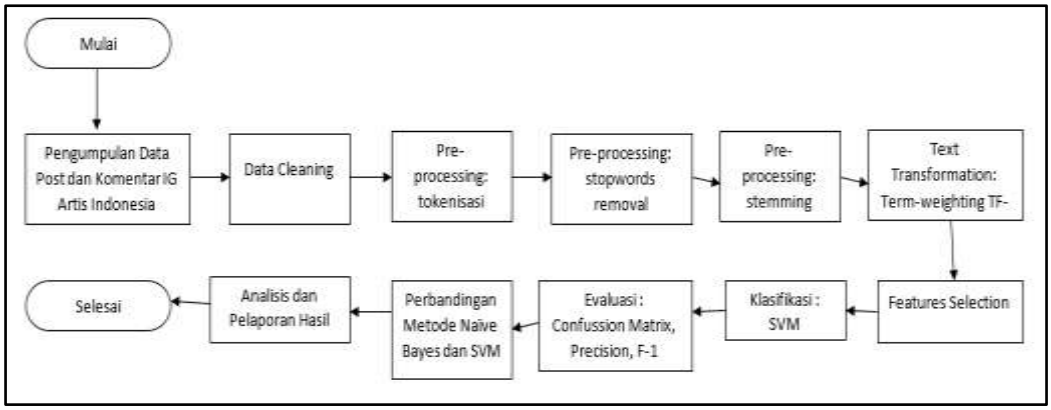
Berdasarkan latar belakang di atas, ternyata IG sendiri belum memiliki fitur deteksi atau penghapus komentar spam otomatis. Fitur yang sudah disediakan adalah fitur laporan suatu komentar adalah spam atau melalui aplikasi mobile untuk melakukan "*hide inappropriate comments*" terhadap komentar berbahasa Inggris berbasiskan kata-kata kunci yang sudah disediakan oleh IG, atau yang terakhir menonaktifkan komentar pada setiap status. Proses melaporkan komentar secara manual tentu sangat merepotkan karena harus dilakukan satu persatu. Cara lain yang dapat dilakukan untuk meminimalisasi komentar spam adalah dengan membuat profil IG menjadi privat. Hal ini tentu tidak mungkin dilakukan oleh para artis, karena jika dibuat privat maka tentu *follower* akan semakin sedikit.

Pada penelitian ini masalah yang dibahas adalah bagaimana membangun model identifikasi komentar spam untuk bahasa Indonesia menggunakan algoritma Naive Bayes (NB) dan Support Vector Machine (SVM). Penelitian ini merupakan kelanjutan dari penelitian sebelumnya yang telah menghasilkan hasil bahwa algoritma NB mampu mencapai akurasi tertinggi 77,25 % untuk deteksi komentar spam di IG [2]. Penelitian mengenai penggunaan metode NB dan SVM yang digunakan dalam klasifikasi atau deteksi spam juga telah banyak dilakukan. Naive Bayes telah digunakan untuk mendeteksi klasifikasi teks karena mudah digunakan, performa baik, dan fleksibel, dan banyak pula yang melakukan berbagai peningkatan algoritma ini, seperti misalnya penggunaan informasi *class* negatif yang diterapkan pada NewsGroup dataset untuk meningkatkan performa NB, dan terbukti memiliki hasil yang meningkat [3]. Naïve Bayes juga telah digunakan pada klasifikasi email spam pada dataset CERT yang memiliki hasil yang mirip dengan metode Auxiliary Features Method [4]. Pada penelitian analisis sentimen, Naïve Bayes juga telah digunakan dalam mendeteksi sentimen komentar pada Facebook Page Calon Presiden RI 2014 dan menghasilkan akurasi mencapai 82% [5]. SVM terbukti memiliki akurasi yang tinggi mencapai 96,3 % untuk mendeteksi email spam, dan meningkat menjadi 98,01 % ketika dikombinasikan dengan algoritma k-Means Clustering [6]. Ada pula penelitian yang menggabungkan SVM dan Naive Bayes guna mengklasifikasi teks ke folder secara otomatis dengan dataset sebesar 20000 (20 kategori) mampu menghasilkan akurasi rata-rata 80% dibandingkan dengan satu metode saja [7]. Hal ini membuktikan bahwa kedua metode tersebut juga memang dapat dan tepat diterapkan dalam klasifikasi komentar spam pada IG.

Penelitian ini memiliki tujuan jangka pendek untuk membangun *dataset* komentar IG berbahasa Indonesia untuk artis ber-*follower* lebih dari 10 juta terpopuler guna mendapatkan dataset training untuk sistem *supervised learning*. Batasan masalah dari penelitian ini adalah (1) menggunakan data dari 10 artis Indonesia yang memiliki *follower* lebih dari 10 juta berdasarkan referensi dari [1], di mana setiap artis diambil 50 status terbaru dengan 50 komentar terbaru, (2) proses *stemming* menggunakan library Sastrawi *Stemming* dari Andi Librian, (3) hanya digunakan untuk deteksi komentar spam dalam bahasa Indonesia, (4) *tool* yang digunakan untuk analisis adalah RapidMiner 7.x.

2. Metode Penelitian

Pada bagian ini akan dituliskan metode penelitian yang digunakan pada sub bab-sub bab berikutnya. Tahap secara keseluruhan dapat dilihat pada Gambar 1



Gambar 1. *Flowchart* dan Metode Penelitian

2.1. Tahap Pengumpulan Data

Pada tahap ini dikumpulkan data status IG dan komentar dari 10 artis terpopuler dengan jumlah *follower* lebih besar sama dengan 10 juta. Setiap satu artis diambil 50 post terbaru dan dari setiap post diambil 50 komentar terbaru. Data 10 artis diambil dari sumber [1] sebagai berikut: @ayutingting92, @princessyahrini, @raffinagita1717, @laudyacynthiabella, @prillylatuconsina96, @juliaperrezz: @chelseaoliviaa, @raisa6690, @lunamaya, @agnezmo.

Terkumpul data sejumlah 10 artis x 50 status x 50 komentar = 25000 data [2]. Data diambil dengan menggunakan tool *Instagram Tool Grabber* yang dikembangkan penulis berdasarkan pengembangan dan modifikasi dari tool PHP Instagram Grabber yang dapat diunduh secara gratis. Setelah tahap pengumpulan data tahap pemrosesan selanjutnya dilakukan seperti pada tahap pemrosesan text mining, yaitu: tokenisasi, *stopwords removal*, *stemming*, *features selection*, klasifikasi, dan evaluasi [8].

2.2. Tahap Pemrosesan Data (*Data Cleaning*)

Pada bagian ini akan dilakukan pemrosesan data berupa data *cleaning*. Data *cleaning* yang dilakukan adalah menghapus karakter-karakter khusus, menghapus angka, menghapus URL, dan data-data kosong. Hal ini penting dilakukan karena proses pengambilan data otomatis dari IG tidak selalu berhasil dengan sempurna. Setelah dilakukan data *cleaning* kemudian dilanjutkan proses pada tahap berikutnya. Dari data yang terkumpul setelah dilakukan data *cleaning* dihasilkan sejumlah 17.007 dengan data dapat dilihat pada Tabel 1 sebagai berikut [2]:

Tabel 1. Data Hasil *Cleaning*

No.	Artis	Nama Kelas dan Jumlah
1.	Ayu Ting-Ting	Spam (1262), Bukan Spam (584)
2.	Julia Perez	Spam (1362), Bukan Spam (739)
3.	Nagita Slavina	Spam (1435), Bukan Spam (610)
4.	Syahrini	Spam (922), Bukan Spam (448)
5.	Laudya Cinthia Bella	Spam (902), Bukan Spam (688)
6.	Prili Ratuconsina	Spam (437), Bukan Spam (1091)
7.	Chelsea Olivia	Spam (1625), Bukan Spam (293)
8.	Luna Maya	Spam (965), Bukan Spam (275)
9.	Raisa	Spam (666), Bukan Spam (621)
10.	Agnes Monica	Spam (1143), Bukan Spam (940)
JUMLAH SPAM 10.719		**JUMLAH BUKAN SPAM 6.288**
TOTAL KESELURUHAN 17.007		

2.3. Tahap *Pre-processing*

Tahap *pre-processing* dilakukan sebagai berikut:

a. *Tokenisasi.* Tokenisasi dilakukan untuk menghaslkan token-token data. Jumlah token yang dihasilkan adalah 35154 token unik dan 7728 token unik.
b. *Stopwords Removal. Stopwords removal* menggunakan data dari file txt yang diinputkan. Setelah dilakukan *stopwords removal*, dihasilkan data token sejumlah 31226 token. Tujuan dari tahap ini adalah mengurangi jumlah token.
c. *Stemming. Stemming* pada penelitian ini menggunakan library Sastrawi Stemming, *library stemming* bahasa Indonesia yang berbasis C, Java, Go, Ruby, PHP dan Python yang berbasis algoritma Nazief dan Adriani. Tujuan dari tahap ini juga mengurangi jumlah token.
d. *Cleansing and Symbol Handling.* Tahap ini terdiri dari c*leansing* yaitu menghapus karakter-karakter seperti: ~, `, !, $, %, ^, &, *, (,), _, -, +, =, :, “, ‘, <, >, koma, titik, ?, /, \, dan |. Kemudian dilanjutkan dengan membuang semua spasi yang berjumlah lebih dari satu dan menggabungkannya menjadi satu spasi saja. Membuang semua spasi di awal dan akhir kalimat (*trim*), dan menghapus semua baris yang kosong. Terakhir adalah membuang semua angka, string dengan format URL, dan email. Simbol-simbol pada IG perlu dikonversikan ke dalam bentuk teks. Data hasil cleansing menghasilkan Spam berjumlah 10399 dan Non Spam berjumlah 6062 data.

2.4. Tahap *Text Transformation*

Pada tahap ini dilakukan proses pengubahan data dari token teks menjadi *vector* data yang memiliki nilai berupa bobot yang dapat digunakan untuk perhitungan data / *text mining*. Proses *text transformation* dilakukan dengan menggunakan pembobotan Term Frequency – Inverse Document Frequency (TF-IDF). Dalam TF-IDF semakin banyak suatu token muncul berkali-kali di banyak dokumen maka berarti token tersebut tidak memiliki bobot yang besar sebab bobot token tersebut tidak penting dan memiliki ciri khas yang membedakannya dengan token-token lain. Sebaliknya token tertentu akan memiliki bobot tinggi jika token tersebut muncul banyak namun hanya di satu atau beberapa dokumen saja, yang artinya semakin penting dan memberikan ciri atau pengaruh kuat tentang suatu dokumen.

2.5. Tahap *Features Selection*

Pada tahap ini dilakukan pemilihan fitur-fitur dari keseluruhan token yang telah ditransformasi dan memiliki bobot TF-IDF seperti pada langkah sebelumnya. Berdasarkan [9] dan [10], tahap ini penting dilakukan dengan tujuan mengurangi jumlah fitur dan memilih fitur token-token tertentu yang memiliki bobot tertinggi sehingga dapat mewakili keunikan setiap data dokumen. Proses ini dilakukan dengan menggunakan metode *pruning* di bawah 0.1 dan di atas 0.98.

2.6. Tahap Klasifikasi

Tahap klasifikasi dilakukan dengan menggunakan algoritma SVM yang diimplementasikan pada RapidMiner 7.5 dengan pengaturan operator seperti pada Gambar 2. Algoritma NB digunakan sebagai pembanding berdasarkan penelitian sebelumnya. Sedangkan parameter-paramater yang digunakan adalah seperti pada Tabel 2 berikut.

Tabel 2. Parameter Klasifikasi

Algoritma	Parameter	Nilai
Support Vector Machine	Kernel Function	RBF
	C	1
	Gamma	1
	Epsilon	0.001
	Max iteration	100000
Naive Bayes	Laplace Correction	Yes
	Estimation mode	Greedy
	Minimum Width	0.1
	No of kernels	10

2.7. Tahap Evaluasi

Tahap evaluasi dilakukan dengan menggunakan pengujian k-fold *validation* pada RapidMiner 7.5 sesuai pengaturan pada Tabel 3 berikut.

Tabel 3. Parameter Evaluasi Penelitian

Algoritma	Parameter	Nilai
Support Vector Machine	Metode validasi	*k-Fold Validation*
	Metrik pengujian	*Confusion Matrix*
	Jumlah data	15000 data
	Tool	RapidMiner 7.5
	Kriteria output yang dihasilkan	*Accuracy* dan *Classification*
	Sampling type	*Shuffled sampling*

3. Kajian Pustaka

3.1. Tinjauan Pustaka

Pada era modern dan berkembangnya media sosial, para pengguna Internet secara *user-centric* bebas dapat melakukan dua hal yaitu proses *read*, yaitu membaca konten yang disediakan oleh orang lain di Internet dan yang kedua adalah proses *write*, yaitu mengisi konten di Internet dengan berbagai cara seperti mengunggah tulisan, dokumen, gambar, ataupun video terutama melalui situs media sosial. Era Internet seperti ini disebut sebagai era Web 2.0, di mana Internet sudah menjadi platform online yang bersifat dua arah, *read* dan *write* [11]. Dengan teknologi berbasis Web 2.0 terdapat banyak aplikasi yang akan memungkinkan akses terintegrasi terhadap berbagai layanan, konten, dan segala sesuatu di Internet. Hal ini juga menyebabkan pengguna Web 2.0 tidak hanya bersifat pasif (konsumer), sekaligus aktif (sebagai produser), yang disebut prosumer [12]. Proses *write*, yaitu menuliskan sesuatu di Internet dapat dilakukan di berbagai hal, salah satunya melalui menulis status dan komentar pada media sosial. Hal ini memiliki resiko buruk yaitu dapat menulis dengan sembarangan, termasuk munculnya tulisan *spam*.

Spam diterjemahkan sebagai suatu tulisan/pesan yang tidak sesuai/tidak berhubungan dengan topik tertentu sehingga menyebabkan ketidaknyamanan atau bahkan ketidaktepatan informasi yang diperoleh pengguna. Spam pada komentar ditemukan dalam bentuk yaitu tautan spam yang ditulis pada *web* seperti *blog* dan *wiki*. Beberapa diantaranya sering ditemukan dalam bentuk komentar, *trackback*, dan *pingback* spam pada artikel blog yang di-posting seseorang. Namun baru-baru ini komentar spam juga berupa tulisan seperti jualan barang dagangan maupun promosi sesuatu yang tidak berhubungan dengan status yang dikomentari, seperti yang banyak ditemukan pada *blog* dan IG.

Beberapa cara manual yang dapat digunakan untuk mendeteksi komentar spam adalah: (1) deteksi komentar dobel/duplikasi, (2) menggunakan *plugin* untuk blog, (3) menonaktifkan komentar tanpa login, (4) menggunakan CAPTCHA, (5) moderasi komentar secara manual, (6) tidak memperbolehkan *hyperlink*, (6) deteksi kata-kata aneh, kesalahan gramatikal, tidak rasional, tidak relevan dengan yang diberi komentar, dan biasanya bersifat sangat umum.

Pada penelitian ini digunakan sistem *supervised learning* di mana sistem berusaha mendeteksi secara otomatis menggunakan algoritma Naïve Bayes (NB) dan Support Vector Machine (SVM). Berdasarkan penelitian sebelumnya diperoleh bahwa algoritma NB mencapai akurasi tertinggi 77,25 %. Penelitian tersebut telah menghasilkan dataset komentar IG dari 10 artis ber-*follower* terbanyak dengan jumlah data sebanyak 17007 data (10719 spam, 6288 bukan spam) [2]. Penelitian ini membandingkan NB dan SVM karena SVM memiliki kelebihan dengan jumlah kelas kecil (biasanya 2 kelas) dan buruk untuk kelas yang sangat banyak [13], serta merupakan klasifier yang sangat baik karena memiliki tingkat akurasi yang tinggi bahkan mencapai di atas 95%, walaupun waktu komputasinya lebih lama daripada Naïve Bayes [14] [15]. SVM dipilih dalam penelitian ini karena beberapa alasan: 1). SVM mampu melakukan generalisasi dengan error yang lebih kecil, 2). SVM mampu bekerja untuk dimensi yang besar, dan 3). SVM memiliki *feasibility* yang jelas, artinya termasuk bisa diimplementasikan dan memiliki banyak *library* pendukung.

3.2. Algoritma Naïve Bayes

Algoritma Naive Bayes (NB) adalah algoritma klasifier yang menggunakan teori kemungkinan dalam bidang statistik yang digagas pertama kali oleh Thomas Bayes untuk memprediksi peluang di masa yang akan datang berdasarkan peluang dari masa sebelumnya. Metode ini kemudian digabungkan dengan kondisi natif yaitu kondisi dimana kondisi antar atribut dalam universe saling bebas dan tidak berhubungan satu sama lain. Dalam kaitannya dengan data latih, setiap data latih memiliki atribut-atribut dan satu buah label kelas, maka kemungkinan suatu data baru masuk ke dalam suatu kelas dapat didefinisikan dengan Persamaan (1) berikut [16]:

$$p(Ck|x) = \frac{p(Ck).p(x|Ck)}{p(x)} \quad (1)$$

Dalam kasus klasifikasi spam, dapat dijelaskan bahwa probabilitas suatu dokumen x masuk dalam kelas Ck jika diketahui sesuatu adalah sama dengan probabilitas keseluruhan bahwa suatu data masuk dalam kelas Ck, dikali probabilitas x ada pada kelas Ck, kemudian dibagi dengan *evidence* probabilitas x. Jika dalam bentuk klasifikasi spam adalah sebagaimana pada Persamaan (2) berikut:

$$p(S|d) = \frac{p(S).p(d|S)}{p(d|S).p(S)+p(d|NS).p(NS)} \quad (2)$$

Dengan keterangan:

- p(S|d) adalah probabilitas dokumen d masuk dalam kategori Spam (S)
- p(S) adalah probabilitas keseluruhan kategori Spam (S)
- p(d|S) adalah probabilitas kategori Spam (S) pada dokumen d
- p(d|NS) adalah probabilitas kategori Not Spam (NS) pada dokumen d
- p(NS) adalah probabilitas keseluruhan kategori Not Spam (NS)

3.3. Algoritma Support Vector Machine

Algoritma Support Vector Machine (SVM) merupakan salah satu algoritma klasifier yang berbasiskan model *supervised learning* dan diperkenalkan oleh Vapnik pada tahun 1992. Pada sejumlah data pelatihan yang memiliki sejumlah x atribut (vektornya memiliki ukuran x dimensi), metode SVM akan mencari dan menemukan sebuah *hyperplane* berukuran x-1 dimensi guna memisahkan data pelatihan berbasiskan kategori atau kelasnya. Proses menemukan *hyperplane* dilakukan dengan memaksimalkan jarak antar kelas (margin). Dengan cara ini SVM dapat menjamin kemampuan generalisasi yang tinggi untuk data-data yang akan datang [17].

Apabila diketahui data *training* merupakan data yang telah diberi label dan memiliki sejumlah x atribut (atau biasa dinamakan sebagai *tuple*), (x_i, y_i) dengan i = 1, 2, …, n, di mana n adalah jumlah data *training*, sedangkan x_i adalah kumpulan atribut pada data *training* ke-i dan y_i adalah kelas dari data *training* ke-i tersebut, maka SVM akan menghitung masalah optimisasi seperti dilihat pada Persamaan (3) [16].

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{x} \xi_i \quad (3)$$

dengan ketentuan seperti pada Persamaan (4) berikut:

$$y_i(w^T \phi(x_i) + b \geq 1 - \xi_i \, , dan \, \xi_i > 0. \quad (4)$$

Metode SVM mempunyai kelemahan pada proses perhitungan yang relatif lama dan sulit diaplikasikan pada jumlah sampel dan dimensi yang besar dibandingkan dengan metode-metode klasifikasi lainnya, namun mempunyai kelebihan dalam mengklasifikasikan data untuk kategori/kelas dengan jumlah sedikit (direkomendasikan untuk 2 kelas) sehingga sangat cocok untuk klasifikasi spam (spam dan *not spam*) [17].

3.4. Confusion Matrix

Confusion Matrix merupakan sebuah tabel atau matriks yang menggambarkan "kebingungan" dari hasil klasifikasi yang dilakukan oleh system dibandingkan dengan yang sebenarnya. Tabel *confusion matrix* dapat dilihat pada Tabel 4 berikut [18]:

Tabel 4. Confusion Matrix

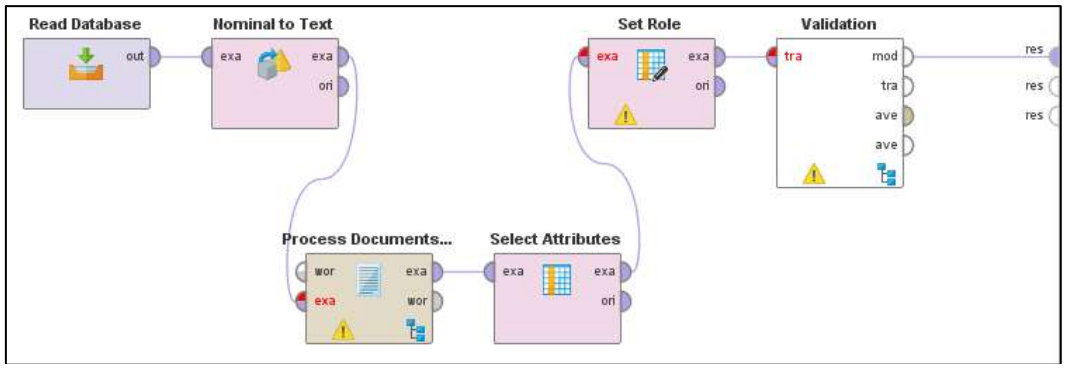
		***Class* Hasil Prediksi**	
		Negatif	Positif
***Class* sebenarnya**	Negatif	*True* Negatif (TN)	*False* Negatif (FN)
	Positif	*False* Positif (FP)	*True* Positif (TP)

Dari *confusion matrix* pada Tabel 5 dapat dilakukan perhitungan lebih lanjut untuk mendapatkan tingkat akurasi (*accuracy*), *recall*, *precision* dan *f-measure* dengan Persamaan (5-10).

- *Accuracy* = (TN + TP) / (TN + FP + FN + TP) (5)
- *Recall / True Positive Rate* = TP / (FP + TP) (6)
- *False Positive Rate* = FN / (TN + FN) (7)
- *Specificity / True Negative* = FP / (FP + TP) (8)
- *Precision* = TP / (FN + TP) (9)
- *F-Measure* = 2 * TP / (2 * TP + FP + FN (10)

4. Hasil dan Pembahasan

Pada bagian ini dibahas dua hal, yaitu konfigurasi pembelajaran sistem berbasis *supervised learning* dan evaluasi pengujian sistem. Hasil konfigurasi RapidMiner dapat dilihat pada Gambar 2 berikut.



Gambar 2. Konfigurasi Sistem *Supervised Learning*

Pada konfigurasi Gambar 2 di atas, dapat dijelaskan bahwa tahapan pertama adalah pengambilan data dari basis data, kemudian dilakukan normalisasi, *pre-processing* dokumen, kemudian langkah terakhir adalah tahap klasifikasi dan validadasi.

Langkah evaluasi dilakukan dengan pengujian sistem yang terdiri dari skenario berikut:

4.1. Skenario I Tanpa *Stemming*

Skenario I tanpa *stemming* adalah pengujian di mana data yang digunakan untuk *training* berjumlah 10.399 untuk data spam dan 6062 untuk data *not spam* tanpa dilakukan *stemming* terlebih dahulu. Dari data tersebut dilakukan pengujian menggunakan teknik k-fold *validation* dengan k=10, artinya data uji untuk masing-masing pengujian berjumlah 1646 (10%) dan hasilnya akan dirata-rata serta ditampilkan dalam kurva ROC (*Receiver Operating Characteristic*). Gambar data profil skenario I dapat dilihat di Gambar 3 (a) berikut.

Gambar 3. (a) Data Profil I dan (b) Data Profil II

4.1.1. Hasil Naïve Bayes (NB)

Hasil *confusion matrix* untuk NB pada Skenario I tanpa *stemming* dapat dilihat pada Tabel 5. Dari hasil tersebut dapat dibuat kurva ROC untuk kemampuan algoritma NB pada data tidak seimbang pada Gambar 4 (a). Dari kurva tersebut dapat dilihat kinerja algoritma NB dalam melakukan klasifikasi. Pada sumbu X dapat dilihat hasil *False Positive Rate* (*fallout*) dan sumbu Y adalah *True Positive Rate* (*sensitivity*). Dari Gambar 4 (a) dapat diketahui bahwa grafik NB sudah cukup baik karena grafik NB memiliki luasan yang besar dan tidak mendekati titik 0,0, justru makin mendekati titik 1,0.

Tabel 5. *Confusion Matrix* Skenario I NB Tanpa *Stemming*

	True Spam	**True Not Spam**	Class Precission
Predicted Spam	6388 (TP)	217 (FP)	**96,71% (precision)**
Predicted Not Spam	4011 (FN)	5845 (TN)	**59,30% (fallout)**
Class Recall	**61,43% (recall)**	**96,42% (specificity)**	

Dari Tabel 5 diperoleh *accuracy* 74,31 %, *classification error* 25,69 %, dan *f-measure* 75, 13 %.

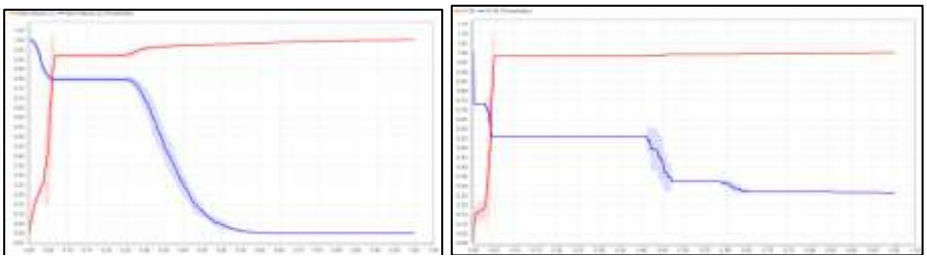
4.1.2. Hasil Support Vector Machine (SVM)

Hasil confusion matrix untuk SVM pada Skenario I tanpa *stemming* dapat dilihat pada Tabel 6. Dari hasil tersebut SVM lebih baik 4.18% daripada Naïve Bayes. Pada kurva ROC untuk algoritma SVM pada data tidak seimbang dapat dilihat pada Gambar 4 (b). Dari kurva tersebut dapat dilihat kinerja algoritma SVM dalam melakukan klasifikasi. Dari gambar tersebut dapat diketahui bahwa garis merah pada grafik SVM cukup mirip dengan grafik NB yang ada pada Gambar 4 (a).

Tabel 6. Confusion Matrix Skenario I SVM Tanpa *Stemming*

	True Spam	**True Not Spam**	Class Precission
Predicted Spam	8933 (TP)	2074 (FP)	**81.16% (precision)**
Predicted Not Spam	1466 (FN)	3988 (TN)	**73.12% (fallout)**
Class Recall	**85.90% (recall)**	**65.79% (specificity)**	

Dari Tabel 6 diperoleh *accuracy* 78.49 %, *classification error* 21.51 %, dan *f-measure* 83.46 %.



Gambar 4. (a) ROC *Curve* NB Skenario I, (b) ROC *Curve* SVM Skenario I (Tanpa *Stemming)*

4.2. Skenario II Tanpa *Stemming*

Skenario II tanpa *stemming* adalah pengujian di mana data spam dan *not spam* dibuat menjadi seimbang, sehingga yang digunakan untuk *training* berjumlah 6062 untuk data spam dan 6062 untuk data *not spam* tanpa dilakukan *stemming* terlebih dahulu. Dari data tersebut dilakukan pengujian menggunakan teknik k-fold *validation* dengan k=10, artinya data uji untuk masing-masing pengujian berjumlah 606 (10%) dan hasilnya akan dirata-rata serta ditampilkan dalam kurva ROC. Gambar data profil skenario II dapat dilihat pada Gambar 3 (b).

4.2.1. Hasil Naïve Bayes (NB)

Hasil *confusion matrix* untuk NB pada Skenario II tanpa *stemming* dapat dilihat pada Tabel 7. Dari hasil tersebut terlihat ada peningkatan 2,75 % dari Skenario I ke Skenario II pada NB. Kurva ROC untuk melihat kemampuan algoritma NB pada data seimbang dapat dilihat pada Gambar 5 (a). Terlihat bahwa ROC NB pada Skenario I dan II sangat mirip seperti pada Gambar 4 (a) dan 5 (a).

Tabel 7. Confusion Matrix NB Skenario II Tanpa *Stemming*

	True Spam	**True Not Spam**	Class Precission
Predicted Spam	3468 (TP)	164 (FP)	**95.48% (precision)**
Predicted Not Spam	2594 (FN)	5898 (TN)	**69.45% (fallout)**
Class Recall	**57.21% (recall)**	**97.29% (specificity)**	

Dari Tabel 7 diperoleh *accuracy* 77,25 %, *classification error* 22,75 %, dan *f-measure* 71,5 %.

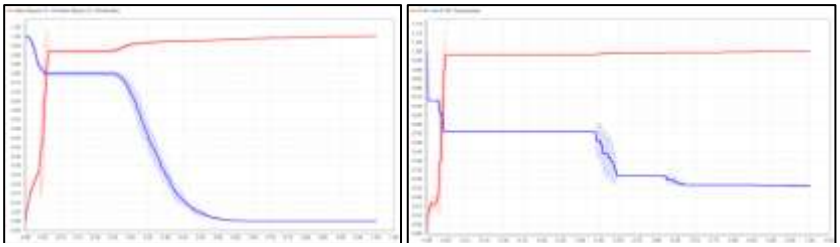
4.2.2. Hasil Support Vector Machine (SVM)

Hasil Confusion Matrix untuk SVM pada Skenario II tanpa *stemming* dapat dilihat pada Tabel 8. Berbeda dengan hasil SVM menggunakan data tidak seimbang (skenario I), dari hasil perbandingan akurasi antara SVM skenario I dan II terjadi penurunan kecil, yaitu sebesar 2.71 %. Kurva ROC untuk SVM data seimbang (skenario II) juga mengalami penurunan luasan seperti pada Gambar 5 (b). Dari Gambar 5 (b) dapat diketahui bahwa grafik SVM untuk data seimbang mirip sekali dengan SVM data tidak seimbang (Gambar 4 (b), yang berarti tidak lebih baik kinerjanya daripada metode NB, karena grafik SVM memiliki luasan yang lebih kecil daripada NB.

Tabel 8. Confusion Matrix SVM Skenario II Tanpa *Stemming*

	True Spam	**True Not Spam**	Class Precission
Predicted Spam	3224 (TP)	98 (FP)	**97.05% (precision)**
Predicted Not Spam	2838 (FN)	5964 (TN)	**67.76% (fallout)**
Class Recall	**53.18% (recall)**	**98.38% (specificity)**	

Dari Tabel 8 diperoleh *accuracy* 75.78 %, *classification error* 24.22 %, dan *f-measure* 68.71 %.



Gambar 5. (a) ROC *Curve* Naïve Bayes Skenario II, (b) ROC *Curve* SVM Skenario II (Tanpa *Stemming)*

4.3. Pembahasan Perbandingan Skenario I dan II Tanpa *Stemming*

Dilihat dari kedua pengujian menggunakan skenario I dan II (tanpa *stemming*) diperoleh peningkatan akurasi sebesar 2,94 % untuk algoritma NB namun justru terjadi penurunan akurasi kecil sebesar 2.71 % untuk algoritma SVM, namun pada prinsipnya penurunan tersebut tidak signifikan. Jika dilihat dari kurva ROC, kinerja NB antara data seimbang dan tidak seimbang hampir sama dan untuk ROC SVM lebih baik daripada NB walaupun peningkatan

sangat kecil. Kurva ROC juga menunjukkan bahwa kinerja algoritma SVM lebih baik daripada NB dan keduanya memiliki akurasi dalam kisaran 74% – 79%. Perbandingan akhir kedua metode untuk skenario I dan II diperoleh bahwa algoritma SVM dan algoritma NB sebenarnya memiliki kemampuan yang hampir sama (terjadi perbedaan namun tidak signifikan) untuk kasus Instagram bahasa Indonesia tanpa *stemming*.

4.4. Skenario I dengan *Stemming*

Skenario I dengan *stemming* adalah pengujian di mana data yang digunakan untuk *training* berjumlah 10.399 untuk data spam dan 6062 untuk data *not spam* denga terlebih dahulu dilakukan pemrosesan *stemming*. Dari data tersebut dilakukan pengujian menggunakan teknik k-fold *validation* dengan k=10, artinya data uji untuk masing-masing pengujian berjumlah 1646 (10%) dan hasilnya dirata-rata serta ditampilkan dalam kurva ROC.

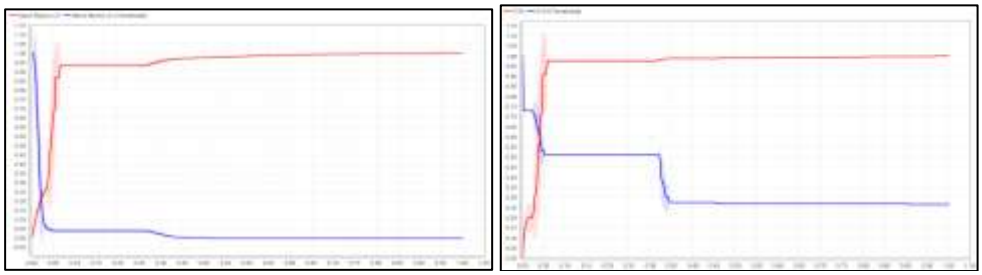
4.4.1. Hasil Naïve Bayes (NB)

Hasil *confusion matrix* untuk NB pada Skenario I dengan *stemming* dapat dilihat pada Tabel 9. Dilihat dari data pada tabel tersebut, tingkat akurasi menurun dibandingkan dengan data yang tidak dilakukan *stemming*. Hal ini terjadi karena data-data teks menggunakan Unicode namun *library stemming* yang digunakan tidak mendukung Unicode dengan baik. Grafik ROC NB untuk data seimbang untuk *stemming* dapat dilihat pada Gambar 6 (a). Pada Gambar 6 (a) kinerja algoritma NB masih cukup baik dan hampir mirip dengan kinerja NB pada Gambar 4 (a) dan 5(a).

Tabel 9. Confusion Matrix Skenario I NB Dengan *Stemming*

	True Spam	**True Not Spam**	Class Precission
Predicted Spam	10176 (TP)	4722 (FP)	**68.30% (precision)**
Predicted Not Spam	223 (FN)	1340 (TN)	**85.73% (fallout)**
Class Recall	**97.86% (recall)**	**22.10% (specificity)**	

Dari Tabel 9 diperoleh *accuracy* 69,96 %, *classification error* 30.04 %, dan *f-measure* 80.4 %.



Gambar 6. (a) ROC *Curve* NB Skenario I dan (b) ROC *Curve* Skenario I SVM (*Stemming)*

4.4.2. Hasil Support Vector Machine (SVM)

Hasil confusion matrix untuk SVM pada Skenario I dengan *stemming* dapat dilihat pada Tabel 10. Dilihat dari data tersebut, tingkat akurasi dengan *stemming* sama saja dibandingkan dengan data yang tidak dilakukan *stemming*. Dalam hal ini *stemming* tidak membawa perubahan apapun. Gambar grafik ROC dapat dilihat pada Gambar 6 (b). Dari grafik ROC SVM sangat mirip seperti pada SVM tidak seimbang maupun seimbang tanpa *stemming*, alias tidak terjadi perubahan. Dari hasil pengujian yang sudah dilakukan untuk skenario I (data tidak seimbang) baik tanpa *stemming* ataupun dengan *stemming* ternyata algoritma SVM lebih baik daripada NB dengan selisih keakuratan antara 4 – 9 % lebih tinggi.

Tabel 10. Confusion Matrix Skenario I SVM Dengan *Stemming*

	True Spam	**True Not Spam**	Class Precission
Predicted Spam	6958 (TP)	131 (FP)	**98.15 % (precision)**
Predicted Not Spam	3441 (FN)	5931 (TN)	**63.28 % (fallout)**
Class Recall	**66.91 % (recall)**	**97.84 % (specificity)**	

Dari Tabel 10 diperoleh *accuracy* 78.3 %, *classification error* 21.7 %, dan *f-measure* 79.57 %.

4.5. Skenario II dengan *Stemming*

Skenario II dengan *stemming* adalah pengujian di mana data spam dan *not spam* dibuat menjadi seimbang, sehingga yang digunakan untuk *training* berjumlah 6062 untuk data spam dan 6062 untuk data *not spam* dengan terlebih dahulu dilakukan pemrosesan *stemming*. Dari data tersebut dilakukan pengujian menggunakan teknik k-fold *validation* dengan k=10, artinya data uji untuk masing-masing pengujian berjumlah 606 (10%) dan hasilnya dirata-rata serta ditampilkan dalam kurva ROC.

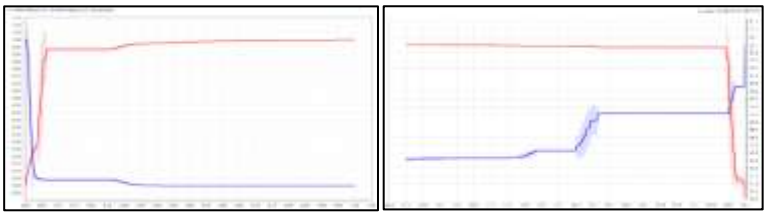
4.5.1. Hasil Naïve Bayes (NB)

Hasil confusion matrix untuk NB pada Skenario II dengan *stemming* dapat dilihat pada Tabel 11. Pada Gambar 7 (a) dapat dilihat grafik ROC dari NB skenario II dengan *stemming*. Dari gambar tersebut dapat diketahui bahwa kinerja algoritma NB masih cukup baik, walaupun tetap lebih baik pada Skenario I dengan *stemming*.

Tabel 11. Confusion Matrix Skenario II NB Dengan *Stemming*

	True Spam	**True Not Spam**	Class Precission
Predicted Spam	5969 (TP)	5072 (FP)	**54.06 % (precision)**
Predicted Not Spam	93 (FN)	990 (TN)	**91.41 % (fallout)**
Class Recall	**98.47 % (recall)**	**16.33 % (specificity)**	

Dari Tabel 11 diperoleh *accuracy* 78.30 %, *classification error* 21.70 %, dan *f-measure* 79.57 %.



Gambar 7. (a) ROC *Curve* Skenario II NB, (b) ROC Curve Skenario II SVM (Dengan *Stemming)*

4.5.2. Hasil Support Vector Machine (SVM)

Hasil confusion matrix untuk SVM pada Skenario II dengan *stemming* dapat dilihat pada Tabel 12. Gambar 7 (b) merupakan grafik ROC algoritma SVM untuk skenario II dengan *stemming* yang jelas lebih baik daripada NB.

Tabel 12. Confusion Matrix Skenario II SVM Dengan *Stemming*

	True Spam	**True Not Spam**	Class Precission
Predicted Spam	3253 (TP)	89 (FP)	**97.34 % (precision)**
Predicted Not Spam	2809 (FN)	5973 (TN)	**68.01 % (fallout)**
Class Recall	**53.66 % (recall)**	**98.53 % (specificity)**	

Dari Tabel 13 diperoleh *accuracy* 76.10 %, *classification error* 23.90 %, dan *f-measure* 69.2 %.

4.6. Pembahasan Perbandingan Algoritma NB dan SVM dengan Stemming

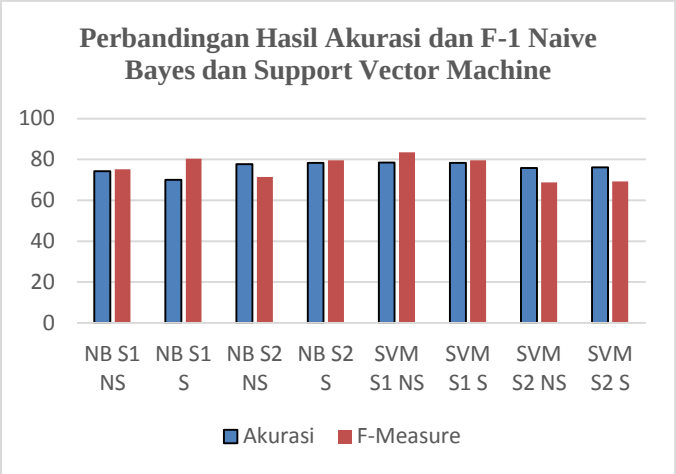
Dari hasil akurasi algoritma SVM lebih tinggi daripada algoritma Nb untuk data dengan *stemming*, walaupun akurasi dan F-Measure-nya lebih kecil / turun daripada yang tanpa *stemming*. Hal ini terjadi karena pemrosesan karakter *Unicode* yang tidak terproses dengan baik. Dari Gambar 11 dan Gambar 12 juga dapat diketahui bahwa grafik kinerja SVM baik karena grafik SVM memiliki luasan yang lebih besar daripada NB pada skenario II menggunakan *stemming*. SVM unggul dari NB baik untuk tanpa *stemming* maupun dengan *stemming*.

4.7. Pembahasan Perbandingan Algoritma NB dan SVM Secara Keseluruhan

Hasil akurasi dan F-measure dari Algoritma NB dan SVM dapat dilihat pada Tabel 13 dan Gambar 13 berikut. Dari Tabel dan gambar tersebut dapat dilihat bahwa akurasi dan *f-measure* terbaik diperoleh SVM S1 NS dengan nilai 78.49 % dan 83.46.

Tabel 13. Perbandingan Akurasi Dan F-Measure Naïve Bayes Dan SVM

	Akurasi	F-Measure
NB S1 NS	74.31 %	75.13
NB S1 S	69.96 %	80.4
NB S2 NS	77.75 %	71.5
NB S2 S	78.3 %	79.57
SVM S1 NS	78.49 %	83.46
SVM S1 S	78.3 %	79.57
SVM S2 NS	75.78 %	68.71
SVM S2 S	76.1 %	69.2



Gambar 8. Grafik Perbandingan Akurasi & F-Measure Naïve Bayes - Support Vector Machine

5. Kesimpulan

Kesimpulan yang diperoleh dari penelitian ini adalah SVM memiliki kinerja yang lebih baik daripada NB namun tidak terlalu signifikan peningkatannya. Tingkat akurasi antara NB dan SVM berkisar antara 70 – 79 % di mana kemampuan deteksi keduanya termasuk dalam kategori baik. Akurasi untuk klasifikasi menggunakan NB adalah 74,31 % untuk skenario I (data tidak seimbang) dan sebesar 77,25% untuk skenario II (data seimbang). Terjadi peningkatan sebesar 2,94 % untuk data seimbang. Akurasi untuk klasifikasi menggunakan SVM adalah sebesar 78,49 % untuk skenario I (data tidak seimbang) dan sebesar 75,78% untuk skenario II (data seimbang). Terjadi penurunan sebesar 2,71 % untuk data seimbang. Proses *stemming* yang digunakan pada data skenario I dan II tidak menghasilkan akurasi yang lebih baik pada algoritma NB maupun SVM karena adanya karakter *Unicode* dan simbol yang belum dapat ditangani sepenuhnya. Penggunaan *stemming* juga tidak meningkatkan akurasi baik pada NB (tingkat akurasi 69.96 % untuk skenario I dan 76.1 % untuk skenario II*)* maupun SVM (tingkat akurasi 78.30 % untuk skenario I dan 76.1 % untuk skenario II*)*. Tahapan *pre-processing* data Instagram bahasa Indonesia yang perlu dilakukan untuk pemrosesan data deteksi komentar spam dari Instagram adalah: *setting encoding* teks ke *encoding* Unicode (UTF-8), tokenisasi, *case folding*, *stop words removal, stemming*, dan konversi simbol-simbol, serta *emoticon*.

Daftar Pustaka

[1] M. Deoranje, "10+ Akun Instagram Dengan Followers Terbanyak di Indonesia," Musdeoranje.net, August 2016. [Online]. Available: http://www.musdeoranje.net/2016/08/akun-instagram-dengan-followers-terbanyak-di-indonesia.html. [Acessed on 9 August 2017].

[2] A. Rachmat and Y. Lukito, "Deteksi Komentar Spam Bahasa Indonesia Pada Instagram Menggunakan Naive Bayes," *Ultimatics - Jurnal Informatika,* vol. 9, no. 1, pp. 50-58, 1 June 2017.

[3] Y. Ko, "How to use negative class information for Naive Bayes classification," *Information Processing & Management,* vol. 53, no. 6, pp. 1255-1268, 2017.

[4] F. G. Wei Zhang, "An Improvement to Naive Bayes for Text Classification," *Procedia Engineering,* vol. 15, no. 15, pp. 2160-2164, 2011.

[5] A. Rachmat C e Y. Lukito, "Klasifikasi Sentimen Komentar Politik dari Facebook," *JUISI,* vol. 02, no. 02, 2016.

[6] N. O. F. Elssied, O. Ibrahim and A. H. Osman, "Enhancement of spam detection mechanism based on hybrid kk," *Soft Computing,* vol. 19, no. 11, p. 3237–3248, 2015.

[7] L. H. Lee, R. Rajkumar and D. Isa, "Automatic folder allocation system using Bayesian-support vector," *Applied Intelligence,* vol. 36, no. 2, pp. 295-307, March 2012.

[8] S. M. Weiss, N. Indurkhya and T. Zhang, Fundamentals of Predictive Text Mining, 1st ed., London: Springer, 2010, pp. XIV, 226.

[9] G. Forman, "An extensive empirical study of feature selection metrics for text classification," *Journal of machine learning research,* vol. 3, no. March, pp. 1289-1305, 2003.

[10] W. Zhang, T. Yoshida and X. Tang, "A comparative study of TF-IDF, LSI, and multi-words for text classification," *Expert Systems with Application,* vol. 38, no. 2011, pp. 2758-2765, 2010.

[11] R. Hail, "Towards a Fusion of Formal and Informal Learning Environments: The Impact of the Read/Write Web," *Electronic Journal of e-Learning,* vol. 7, no. 1, pp. 29-40, 2009.

[12] J. A. Lara, D. Lizcano, M. A. Martínez and J. Pazos, "Developing front-end Web 2.0 technologies to access services, content and things in the future Internet," *Future Generation Computer Systems,* vol. 29, no. 5, pp. 1184-1195, 2013.

[13] Y. Lukito and A. R. Chrismanto, "Perbandingan Metode-Metode Klasifikasi untuk Indoor Positioning System," *Jutisi (Jurnal Teknik Informatika dan Sistem Informasi),* vol. 1, no. 2, pp. 123-131, 2015.

[14] D. Ariadi and K. Fithriasari, "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, pp. D248-D253, 2015.

[15] S. N. D. Pratiwi and B. S. S. Ulama, "Klasifikasi Email Spam dengan Menggunakan Metode Support Vector Machine dan k-Nearest Neighbor," *JURNAL SAINS DAN SENI ITS,* vol. 5, no. 2, pp. D-344 - D-349, 2016.

[16] H. Jiawei, K. Micheline and P. Jian, Classification: basic concepts. In Data mining Concepts and techniques (3rd ed.), Amsterdam: Elsevier, 2011.

[17] S. M. Dr. Suyatno, Data Mining untuk Klasifikasi dan Klasterisasi Data, Bandung: Informatika, 2017.

[18] X. Deng, Q. Liu, Y. Deng e S. Mahadevan, "An improved method to construct basic probability assignment based on the confusion matrix for classification problem," *Information Sciences,* vol. 340–341, no. 1 May 2016, pp. 250-261, 2016.