ISSN 2085-4552

Deteksi Komentar *Spam* Bahasa Indonesia Pada Instagram Menggunakan *Naive Bayes*

Antonius Rachmat C[1], Yuan Lukito[2]
Prodi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Duta Wacana Yogyakarta
anton@ti.ukdw.ac.id[1], yuanlukito@ti.ukdw.ac.id[2]

Diterima 19 Mei 2017
Disetujui 16 Juni 2017

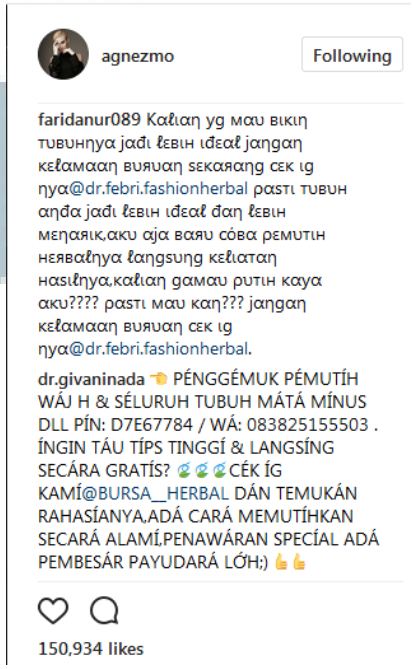
Abstract*— Instagram is the most famous pictures and videos media sharing based on the web & mobile application. Instagram users can have picture posts that can be commented by their followers. Indonesian public figures such as actors, actresses, musicians use Instagram to promote their activities to their followers. Unfortunately, there are a lot of spam comments in Instagram that need special attention and have to be removed. This research grabs Instagram comments and builds the dataset from Indonesian public figures who have more than one million followers. By using preprocessing (tokenization, stop words removal, and stemming), TF-IDF weighting, and supervised learning, Naive Bayes method is used to detect spam comments in Indonesian. Naive Bayes produces 74,31% accuracy rate on unbalanced datasets and 77,25% accuracy rate on balanced datasets. This result shows that Naïve Bayes can be used to build an automatic Indonesian spam comments detector on Instagram with high accuracy rate. The novelty of this research is that Naive Bayes can be used to detect spam comment on our Indonesian Instagram comments dataset.

***Index Terms*—Instagram, Naive Bayes, Indonesian spam comments, spam comments detection.**

I. PENDAHULUAN

Instagram (IG) merupakan salah satu aplikasi media sosial berbasis web dan *mobile* yang khusus digunakan untuk mengunggah gambar / foto. IG merupakan situs media sosial yang semakin banyak digunakan terutama oleh para artis/aktor Indonesia. Para pengguna IG akan mengunggah foto-foto kegiatan mereka dan kemudian untuk masing-masing foto / gambar tersebut akan dapat diberi *caption*, *tagging* akun IG lainnya, lokasi tempat kejadian foto tersebut diunggah, edit foto sesaat sebelum diunggah langsung dari aplikasi *smartphone*, dan *hashtag* tertentu agar foto tersebut makin banyak dilihat orang lain. Aplikasi IG dikembangkan untuk *smartphone*, baik platform iOS, Android, ataupun Windows Phone dan bersifat gratis. IG berbasis web dapat diakses di http://www.instagram.com. Melalui web pengguna hanya dapat melihat foto/gambar dari akun IG lain yang mereka ikuti, melihat dan mengedit profil pengguna aktif, dan melihat aktivitas pengguna IG lainnya, tapi tidak bisa mengunggah foto/gambar dari web, melainkan harus menggunakan aplikasi pada *smartphone*.

Salah satu hal yang menyebabkan IG banyak digunakan adalah kemudahannya untuk mengunggah foto langsung dari *smartphone*, mengingat pengguna media sosial kebanyakan adalah orang muda dan sangat menyukai *selfie*. Namun disamping kelebihan tersebut tentu terdapat kekurangan yang cukup mengganggu yaitu banyaknya komentar yang dapat dikategorikan sebagai komentar spam terhadap suatu *post* foto yang diunggah pada IG. Komentar spam akan semakin banyak terhadap IG artis/orang terkenal karena *follower*-nya juga semakin banyak. Bahkan di film Cek Toko Sebelah [1], yang baru rilis di tanggal 28 Desember 2016 saja diperlihakan bahwa komentar-komentar spam begitu mengganggu dan muncul cukup banyak di IG sampai digunakan sebagai salah satu bahan lelucon pada film itu. Contoh komentar spam pada salah satu foto milik @agnezmo dapat dilihat di Gambar 1.



Gambar 1. Komentar Spam Instagram

Beberapa solusi menghadapi komentar spam sudah ada, namun semuanya dilakukan secara manual. Pada [2] dan [3], pengguna IG dapat menghapus secara manual komentar spam tersebut namun jelas-jelas membutuhkan waktu yang besar dan harus diperiksa satu persatu. Selain dihapus secara manual IG juga menyediakan fitur untuk melaporkan semua komentar sebagai spam secara manual juga, artinya harus dilakukan satu persatu. Hal berikutnya untuk meminimalisasi komentar spam adalah dengan mengubah akun IG menjadi privat. Hal ini tentu sulit dilakukan bagi akun artis / aktor / publik figur, karena jika akun IG dibuat menjadi privat tidak bisa langsung di *follow* oleh akun lain. Hal terakhir yang dapat dilakukan adalah menggunakan pengaturan mengaktifkan fitur IG untuk menghapus komentar yang mengandung kata-kata tertentu yang dimasukkan sendiri oleh pengguna yang dianggap spam. Semua solusi tersebut hanya bisa digunakan dalam bahasa Inggris dan tidak dapat diterapkan dalam bahasa Indonesia.

Berdasarkan latar belakang tersebut pada penelitian ini akan dibangun suatu sistem yang dapat mengklasifikasikan komentar spam berbahasa Indonesia dengan mengambil data *training* komentar-komentar spam pada IG beberapa artis terkenal Indonesia. Terdapat beberapa metode untuk klasifikasi seperti Naive Bayes, K-Nearest Neighbour, Decision Tree, atau Support Vector Machine. Metode klasifikasi yang digunakan dalam penelitian ini adalah Naïve Bayes. Metode Naïve Bayes menggunakan konsep probabilitas setiap kelas dalam pembelajaran klasifikasinya, sehingga jika jarak perbedaan antar kelas tidak besar. Metode ini dipilih karena mudah diimplementasikan dan tidak terlalu membutuhkan sumber daya komputer yang besar.

Penelitian ini akan mengumpulkan data berupa status foto dan komentar dari 10 akun artis / aktor Indonesia yang memiliki *follower* lebih dari 1 juta. Untuk setiap akun artis / aktor akan diambil 50 status terbaru dan semua komentarnya. *Dataset* yang terbentuk akan dilakukan pelabelan secara manual menggunakan tenaga ahli untuk dapat digunakan sebagai data latih sistem *supervised learning* deteksi komentar spam menggunakan Naive Bayes.

II. LANDASAN TEORI

A. *Tinjauan Pustaka*

Spam adalah "*irrelevant or unsolicited messages sent over the Internet, typically to a large number of users, for the purposes of advertising, phishing, spreading malware, etc.*" [4]. Dalam terjemahan bebasnya spam berarti suatu tulisan / pesan yang tidak sesuai / tidak berhubungan dengan topik tertentu sehingga menyebabkan ketidaknyamanan atau bahkan ketidaktepatan informasi yang diperoleh pengguna. Spam dapat berwujud banyak hal, misalnya email spam, iklan spam, *click* spam, link spam, berita spam, dan tulisan (komentar) spam [5]. Istilah web spam (spam *indexing*) pertama kali muncul di tahun 1996 menurut tulisan Convey pada [6] pada harian The Boston Harald. Spam pada komentar merupakan bentuk link spam yang muncul pada web seperti *wikis*, *blogs*, dan *guestbooks*. Beberapa diantaranya sering ditemukan komentar, *trackback*, dan *pingback* spam pada tulisan (*blog*) yang diposting seseorang [7].

Menurut Hines pada [7] beberapa cara manual yang bisa digunakan untuk mendeteksi komentar spam adalah melihat konten teks komentar secara langsung apakah spesifik terhadap tulisan atau tidak, melihat apakah ada *link* pada komentar yang harus diklik atau tidak, pengguna yang berkomentar apakah menggunakan nama asli atau tidak, apakah penulis komentar menggunakan email asli atau tidak, atau menggunakan beberapa email yang berbeda-beda atau tidak. Sedangkan menurut Mishne, Carmel, & Lempel pada tahun 2005 pada [8] beberapa teknik untuk mengurangi komentar spam adalah menggunakan registrasi sebelum posting komentar, menggunakan *captcha* sebelum posting komentar, tidak mengijinkan komentar berbentuk HTML, menggunakan *blacklist* IP a*ddress*, dan memberikan batasan komentar agar tidak berkali-kali memberikan komentar.

Selain cara-cara manual seperti yang telah dijelaskan sebelumnya, deteksi spam pada teks dapat dilakukan secara otomatis menggunakan beberapa metode *content based filtering, link based filtering* dan metode yang tidak terlalu umum seperti menggunakan kebiasaan (*behaviour*) pengguna misalnya klik, *gesture*, *session*, dan lain-lain. Menurut Spirin & Han pada [9] deteksi spam kebanyakan menggunakan metode berbasis isi teks / tulisan yang ditulis. Hal ini dapat dilakukan dengan menggunakn beberapa metode seperti algoritma klasifikasi pada teks seperti algoritma Naive Bayes [10] dan Support Vector Machine [11].

Naive Bayes juga sudah digunakan sebagai metode untuk mengklasifikasikan sentimen pada dataset teks politik (SentiPol *dataset*) yang ternyata mencapai tingkat akurasi sebesar 82,3 %. Dataset SentiPol dikumpulkan dari Facebook Page dan dilabeli secara *crowdsourced labelling* menggunakan metode *Weighted Majority Voting*. Naive Bayes termasuk algoritma yang menghasilkan nilai akurasi yang cukup tinggi dalam klasifikasi data teks [12].

Penelitian yang dilakukan oleh penulis merupakan penelitian yang akan mendeteksi spam dari teks komentar pada web media sosial Instagram berbahasa Indonesia. Penelitian ini dimulai dari pengambilan data pelatihan dari Instagram menggunakan Instagram API (*Grabber*) yang akan dijelaskan pada bagian berikutnya. Setelah data diperoleh maka dilakukan pelabelan data secara manual dan kemudian dilakukan implementasi deteksi spam menggunakan algoritma

yang biasa digunakan untuk klasifikasi teks yaitu Naive Bayes. Keaslian penelitian terdapat pada studi kasus data yang berasal dari akun Instagram dan komentar-komentar bebahasa Indonesia.

B. Instagram dan Instagram API

Instagram merupakan web media sosial yang muncul sejak 2010 (kemudian diakuisisi oleh Facebook tahun 2012) dan dikhususkan untuk menggunggah koleksi foto. Menurut Pew Research Center pengguna Instagram mencapai 26% dari pengguna Internet dewasa, dimana penggunanya berusia 18-29 tahun [13]. Foto-foto yang diunggah dapat dilihat oleh semua followers dalam *timeline*-nya. Instagram memiliki kekhasan yaitu: 1). Proses unggah hanya bisa dilakukan melalui aplikasi berbasis *mobile* dan aplikasinya memiliki fitur manipulasi foto seperti efek-efek foto tertentu. 2). Pengguna IG dapat mengikuti akun pengguna lain (*follow*) atapun dapat diikuti (di *follow*) oleh akun lain. Pengguna IG dapat membuat akun profil IG nya menjadi publik dan privat. 3). Semua foto yang diunggah dapat diberi *caption* dan *hashtag*, termasuk *tagging* terhadap pengguna akun lain. 4). Semua posting foto dapat diberi komentar, di-*like*, bahkan dilaporkan ke IG sebagai *posting* yang tidak baik. 5). Selain foto IG dapat digunakan juga untuk mengunggah video berdurasi pendek (kurang lebih 1 menit) [14].

Instagram API (*Application Programming Interface*) merupakan *tool* pemrograman berbasis layanan yang dapat digunakan untuk mengakses dan memanipulasi pencarian tag, pencarian foto, *trending* foto, print foto, *custom item*, *feeds*, dan komentar-komentar yang terdapat pada Instagram menggunakan bahasa pemrograman tertentu dari sisi *programmer* (*developer*) [15]. Instagram API dapat diakses di https://www.instagram.com/developer/ menggunakan akun Instagram yang telah dibuat sebelumnya. Untuk dapat menggunakan Instagram API dibutuhkan : 1). register API client, 2). *Generate access token*, 3). Lakukan pembuatan aplikasi untuk mengakses hal yang terdapat pada Instagram [16].

C. Text Transformation dan TF-IDF Weighting

Di dalam pengolahan data *mining* berupa teks, proses yang paling penting adalah pengubahan dari data teks tidak terstruktur menjadi data nomimal yang terstruktur. Perubahan yang disebut *text transformation* tersebut, terdiri dari: 1). Tokenisasi, yaitu pengubahan dari string besar ke dalam satu buah kalimat/kata, 2). *Pre-processing*, yaitu proses data *cleaning*, pengubahan ke huruf kecil/besar semua, penghapusan kata-kata yang masuk dalam *stop words*, pengubahan ke bentuk kata dasar (*stemming*), dan konversi *emoticon* & kata-kata khusus, serta 4). Token weighting, yaitu pemberian bobot (w) terhadap token-token tersebut misalnya menggunakan TF-IDF [17] .

TF-IDF merupakan *Term Frequency-Inverse Document Frequency*, yang dapat digunakan untuk memberikan tingkat kepentingan token-token penentu dalam klasifikasi spam seperti pada Persamaan (1) [17]:

$$tfidf(t,d,D) = tf(t,d) * idf(t,D) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

Di mana:

- tfidf(t,d,D) adalah bobot kepentingan suatu token yang muncul dalam suatu komentar di seluruh komentar-komentar yang ada, dimana semakin sering muncul suatu token dalam suatu dokumen dan semakin banyak komentar yang memilikinya akan semakin tidak penting karena token tersebut bersifat sangat umum.
- tf(t,d) adalah jumlah token yang terdapat pada satu komentar.
- idf(t,D) adalah *Inverse Document Frequency*, yaitu log dari jumlah seluruh komentar dibanding dengan jumlah seluruh komentar dimana token tersebut muncul.
- Rumus idf(t,D) dapat dilihat pada Persamaan (2):

$$idf(t,D) = \log (N / DF) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

Di mana:
N adalah jumlah keseluruhan komentar dan DF adalah jumlah kemunculan token pada semua komentar-komentar.

D. Algoritma Naive Bayes

Algoritma Naive Bayes merupakan algoritma klasifikasi berdasarkan probabilitas dalam statistik yang dikemukakan oleh Thomas Bayes yang memprediksi peluang di masa depan berdasarkan peluang di masa sebelumnya. Metode ini kemudian digabungkan dengan situasi "*naive*" dimana kondisi antar atribut dalam sistem saling bebas dan tidak berhubungan satu sama lain. Dalam data *training*, setiap data traing memiliki atribut-atribut dan 1 label *class*, maka probabilitas suatu data masuk ke dalam suatu label *class* dapat didefinisikan sebagai berikut [18]:

1. Diketahui D adalah data *training* dan label *class*-nya. Setiap data direpresentasikan dalam bantuk n dimensi vektor atribut X = (x1, x2,... , xn)
2. Misalkan terdapat m jumlah class, C1, C2,... ,Cm. Metode Naive Bayes akan memprediksi apakah X masuk dalam *class* yang memiliki nilai *posterior* probabilitas tertinggi. Naive Bayes akan memprediksi X akan masuk ke dalam kelas Ci jika dan hanya jika: $P(Ci|X) > P(Cj|X)$ for $1 \leq j \leq m, j <> i$. Kemudian akan dimaksimumkan P(Ci|X). *Class* terbanyak dari Ci disebut

dengan *maximum* posteriori *hypothesis* yang dihitung menggunakan Persamaan (3):

$$P(C_i|\boldsymbol{X}) = \frac{P(\boldsymbol{X}|C_i)P(C_i)}{P(\boldsymbol{X})}.$$

.....(3)

3. Karena P(X) bersifat tetap untuk semua *class*, maka hanya P(X | Ci) P(Ci) yang harus dimaksimumkan. Jika *prior* probabilitas dari *class* tidak diketahui, maka secara umum diasumsikan semua *class* sama P(C1) = P(C2) = ... = P(Cm). Perlu diingat bahwa *prior* probabilitas dari *class* diestimasi dengan P(Ci) = |Ci,D| / |D|, dimana |Ci,D| adalah jumlah *training* data yang termasuk dalam *class* Ci di *dataset* D.
4. Untuk *dataset* yang memiliki banyak atribut maka kompleksitas komputasi akan sangat tinggi, sehingga perlu direduksi dengan cara mengasumsikan semua kondisi *class* bersifat saling bebas (*independence*). Hal ini menganggap bahwa nilai antar atribut saling tidak mempengaruhi satu sama lain, sehingga dapat didefinisikan :
Dimana probabilitas P(x1| Ci), P(x2| Ci),..., P(xn| Ci) dapat diperoleh dengan mudah dari data *training*. Dimana xk adalah nilai yang ada di atribut Ak untuk data X. Untuk setiap atribut, harus dilihat apakah nilai atribut bersifat kategorikal atau nilai kontinu.
   - Jika Ak bersifat kategorikal, maka P(xk|Ci) adalah jumlah data xk yang memiliki *class* Ci di data *training* D dibagi dengan |Ci,D|, jumlah seluruh data *class* Ci di data *training* D.
   - Jika Ak bersifat kontinu, seperti misalnya data umur, data angka lainnya, yang tidak bisa dikategorikan, maka data tersebut harus dibuat dalam rentang nilai, menggunakan Gaussian Distribution dengan Persamaan (4) dan (5)

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

.....(4)

Sehingga diperoleh P(xk| Ci) = g(xk, μCi, σCi) .....(5)

.

5. Untuk memprediksi label class untuk data X, P(X|Ci) P(Ci) maka prediksi dilakukan untuk setiap class Ci. Metode Naive Bayes akan memprediksi class untuk X adalah Ci jika dan hanya jika (Persamaan (6)):

P(X|Ci)P(Ci) > P(X|Cj)P(Cj) for $1 \le j \le m$, j <>i. ..... (6).

Dalam arti prediksi terhadap *class* dengan probabilitas terbesar dihitung dengan Persamaan (7).

$$P(\boldsymbol{X}|C_i) = \prod_{k=1}^{n} P(x_k|C_i) = P(x_1|C_i) \times P(x_2|C_i) \times \cdots \times P(x_n|C_i). \quad (7)$$

E. Confusion Matrix

Untuk melakukan pengujian terhadap sistem, dilakukan evaluasi akurasi sistem dalam mengklasifikasikan sentimen pada dataset dengan menggunakan *confusion matrix* [19] seperti pada Tabel 1 berikut.

Tabel 1. *Confusion Matrix*

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

Di mana:

- *True* negatif = jumlah data negatif yang benar dikategorikan sebagai *class* negatif
- *False* negatif = jumlah data negatif yang dikategorikan sebagai *class* positif
- *False* positif = jumlah data positif yang dikategorikan sebagai *class* negatif
- *True* positif = jumlah data positif yang benar dikategorikan sebagai *class* positif

Dari *confusion matrix* pada Tabel 1 dapat dilakukan perhitungan lebih lanjut untuk mendapatkan tingkat akurasi (*accuracy*), *recall*, *precision* dan *f-measure* dengan Persamaan (8) – Persamaan (13).

Accuracy = (TN + TP) / (TN + FP + FN + TP) .....(8)

Recall / True Positive Rate = TP / (FP + TP)..... (9)

False Positive Rate = FN / (TN + FN) .....(10)

Specificity / True Negative = FP / (FP + TP) .....(11)

Precision = TP / (FN + TP) .....(12)

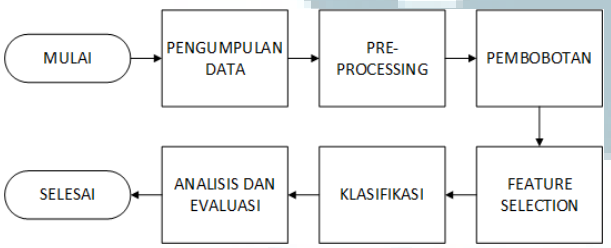
F-Measure = 2 * TP / (2 * TP + FP + FN ..... (13)

III. METODE PENELITIAN

Metode penelitian yang dilakukan dalam penelitian ini terdiri atas lima tahap sebagai berikut:

1. Tahap studi literatur digunakan untuk mempelajari algoritma Naive Bayes yang digunakan, termasuk mempelajari kode program pembuatan Instragram data *collector* menggunakan Instagram API (*tool* Instagram *Grabber*) [20].
2. Tahap pengumpulan data digunakan untuk mengumpulkan data yang berhubungan dengan penelitian yang akan dilakukan. Tahap ini akan menggunakan Instagram API untuk pengambilan data dari Instagram yang berasal dari 10 artis Indonesia yang memiliki *follower* lebih dari 1 juta. Pada tahap ini juga termasuk pre-*processing* data.
3. Tahap implementasi, yaitu tahap untuk melakukan pelabelan secara manual tentang spam / bukan spam dan kemudian dilanjutkan dengan melakukan implementasi kedua metode menggunakan aplikasi dan RapidMiner 7.5. Tahap ini termasuk pembobotan, *features selection*, dan klasifikasi.
4. Tahap analisis dan evaluasi bertujuan untuk menganalisis penggunaan kedua metode yang kemudian dibandingkan keakuratan keduanya menggunakan *confusion matrix*.

Seluruh tahapan penelitian dapat dilihat pada diagram alir Gambar 2 berikut:



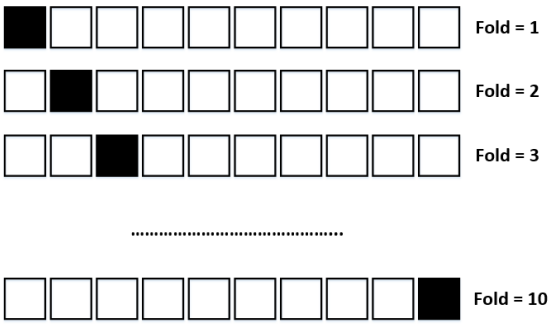
Gambar 2. Tahapan Penelitian

A. *Pengumpulan Data*

Pengujian terhadap implementasi kedua metode akan dilakukan terhadap seluruh data uji yang telah diambil dari 10 akun Instagram artis Indonesia yang memiliki *follower* lebih dari 1 juta akun. Data uji akan disimpan dalam basis data yang berisi data *username* IG artis, *posting* IG, tanggal *posting*, dan semua komentar dari 50 status terbaru, siapa yang berkomentar, dan tanggal komentar.

B. *Metode Pengujian*

Pengujian akan dilakukan menggunakan metode *k-fold validation* dengan nilai k = 10. Metode pengujian ini dilakukan dengan cara membagi dataset menjadi 10 bagian yang sama. Dalam 10 iterasi, setiap bagian dari dataset digunakan sebagai data uji, sedangkan bagian lainnya digunakan sebagai data pelatihan. Ilustrasi *k-fold validation* dapat dilihat pada Gambar 3.



Gambar 3. Ilustrasi metode *k-fold validation* dengan nilai k = 10.

Ringkasan dari pengujian yang akan dilakukan dapat dilihat pada Tabel 2.

Tabel 2. Ringkasan Rancangan Pengujian

Parameter	Nilai
Metode validasi	*k-Fold Validation*
Metrik pengujian	*Confusion Matrix*
Jumlah data	17000 data
Tool	RapidMiner 7.5
Kriteria output yang dihasilkan	*Accuracy* dan *Classification*
Sampling tape	*Shuffled sampling*

IV. HASIL DAN PEMBAHASAN

A. *Implementasi Pengambilan Data*

Data diperoleh diambil langsung dari web Instagram menggunakan teknik *scrapping* menggunakan *tool* PHP Instagram Grabber [19] yang tersedia di GitHub dan dapat diunduh secara gratis. Tool ini dibuat oleh Thomas Bolander dan Kristian Vassard dari Denmark sejak 9 Juni 2016 (versi 1.0.4) di GitHub. *Tool* ini memungkinkan penulis untuk mengambil data-data komentar Instagram dari seorang pemilik akun Instagram yang bersifat publik seperti artis Indonesia tanpa menggunakan *username* dan *password developer*. Teknik yang digunakan oleh tool ini adalah menggunakan teknik *scrapping* halaman-halaman HTML dari Instagram dan kemudian di-*parsing* otomatis berbasis PHP.

Penggunaan *tool* PHP Instagram Grabber adalah sebagai berikut:

1. Pengaturan PHP Instagram *Grabber* pada *server* 'localhost'. Tambahkan sesuai kebutuhan seperti pengaturan: set_time_limit(0) untuk memproses data yang jumlahnya besar dan lama. Pengaturan proxy juga jika perlu menggunakan pada variabel `CURLOPT_PROXY=> <IP address proxy>` dan `CURLOPT_PROXYPORT => <no port>`.
2. Menggunakan *method* `Bolandish\Instagram::getMediaByUserID("<id`

`instagram>",<jumlah post yang hendak diambil>).`

3. Menggunakan perulangan yang dilakukan untuk mengambil semua komentar untuk masing-masing post yang diambil di langkah 2 sebelumnya dengan perintah `Bolandish\Instagram::getCommentsByMediaShortcode(<kode post yg didapat dari langkah 2>, <jumlah komentar yg hendak diambil>).`
4. Data-data komentar tersebut dimasukkan / diekspor ke format lain seperti CSV atau TXT.

Berikut adalah rencana pengambilan data untuk pembentukan *dataset* komentar spam dari Instagram artis Indonesia:

1. Data *post* akan diambil dari 10 akun aktor / artis Indonesia yang memiliki jumlah *follower* terbanyak berdasarkan sumber [21] sebagai berikut:
   - Ayu Tingting @ayutingting92: 522969993 - 18.4 juta *followers*
   - Syahrini @princessyahrini: 24239929 - 16.8 juta *followers*
   - Raffi Ahmad Nagita Slavina @raffinagita1717: 1918078581 - 15.7 juta *followers*
   - Laudya Chinthia Bella @laudyacynthiabella: 2993265 - 14.8juta *followers*
   - Prilli Ratucosina @prillylatuconsina96: 225064794 - 14.8juta *followers*
   - Julia Perezz @juliaperrezz: 30585021 - 12.3 juta *followers*
   - Chelsea Olivia @chelseaoliviaa: 5735890 - 12.6 juta *followers*
   - Raisa @raisa6690: 8115577 - 12.1 juta *followers*
   - Luna Maya @lunamaya: 1948416 - 11.6 juta *followers*
   - Agnes Monica @agnezmo: 4934196 - 11.3 juta *followers*.
2. Dari setiap akun artis / aktor akan diambil 50 *post* terbaru.
3. Dari setiap *post* yang terambil dari langkah 2 akan diambil 50 komentar terbaru, sehingga terdapat data sekitar 10 artis x 50 *post* x 50 komentar = 25000 komentar.
4. Dari 25000 komentar tersebut akan dibuat dalam format CSV per akun artis dan kemudian dilakukan *preprocessing* data pada bagian berikutnya.

Setelah data terambil menggunakan *tool* Grabber, langkah berikutnya adalah melakukan pelabelan data. Pelabelan data dilakukan secara manual menggunakan tenaga pelabel ahli. Proses pelabelan memakan waktu sekitar 1 bulan untuk semua data. Pelabelan dilakukan hanya dengan 2 kelas, yaitu "spam" dan "notspam". Pelabelan relatif mudah karena sangat jelas sekali perbedaan antara komentar spam dan bukan spam.

Tabel 3 merupakan profil hasil klasifikasi pelabelan manual menggunakan tenaga pakar untuk dataset Instagram yang telah dikumpulkan. Dari rencana jumlah data 25.000, pada saat pengambilan tidak semua post memiliki komentar 50 komentar sehingga realisasi pengambilan data terdapat pada Tabel 3.

Tabel 3. Profil Hasil Pelabelan Instagram 10 Artis Indonesia

No.	Artis	Nama Kelas dan Jumlah
1.	Ayu Ting-Ting	Spam (1262), Bukan Span (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

B. Implementasi Preprocessing Data

Setelah data terkumpul dalam format CSV langkah berikutnya adalah tahap *pre-processing (cleaning data)* agar nantinya dapat dimasukkan dalam basis data MySQL dengan mudah. Proses *cleaning* dilakukan secara langsung pada file CSV nya dengan cara:

1. Menggunakan format karakter *unicode* UTF-8 karena semua karakter hasil Instagram menggunakan simbol-simbol dan karakter *unicode*.
2. Konversi karakter khusus seperti *emoticon* menjadi istilah-istilah seperti yang ditampilkan pada Tabel 4 berikut:

Tabel 4. Konversi *Emoticon* dari Instagram

No.	Karakter Emoticon	Konversi
1	:)	Senyum
2	(y)	Suka
3	~_~	Netral
4	(Y)	Suka
5	:*	Cium
6	=))	Senyum
7	:D	Senyum
8	*Like* atau *like*	Suka
9	*YES*, *Yes*, atau *yes*	Suka

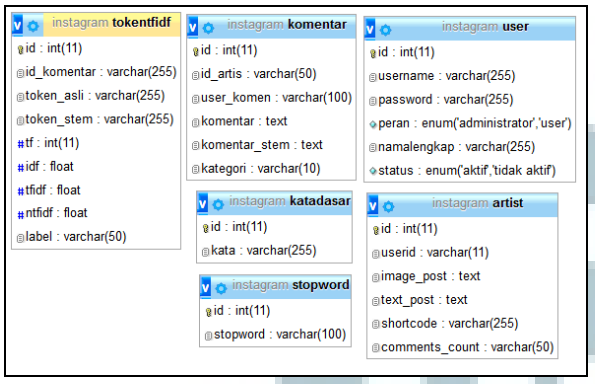
3. *Cleansing*: menghapus karakter-karakter seperti: ~, `, !, $, %, ^, &, *, (,), _, -, +, =, :, ", ', <, >, koma, titik, ?, /, \, dan |.
4. Membuang semua spasi yang berjumlah lebih dari satu dan menggabungkannya menjadi satu spasi saja. Membuang semua spasi di

awal dan akhir kalimat (*trim*), dan menghapus semua baris yang kosong.

5. Membuang semua angka, string dengan format URL, dan email.

C. *Implementasi Basis Data*

Setelah semua data CSV sudah melalui tahapan *cleaning* data, maka langkah selanjutnya dilakukan pembuatan basis data yang terdiri dari 3 tabel: stopwords, komentar, dan artist, sesuai skema pada Gambar 4 berikut.



Gambar 4. Skema Basis Data

Setelah dibuat semua tabel pada basis data MySQL langkah berikutnya adalah mengimpor semua CSV ke dalam tabel artis dan komentar menggunakan *tool import* pada MySQL. Contoh data komentar setelah diimpor dapat dilihat pada Gambar 5 berikut.

widya prasasti2212	butuh followers instagram atau likes instagram sedikit yuk buruan order hanya unicornappsstores
ficwidhi	followers instagram kamu sedikit atau likes instagram kamu sedikit yuk order hanya unicornappsstores
opet awsm	butuh followers instagram atau likes instagram kamu sedikit yuk cek instagram unicornappsstores melayani dengan ramah

Gambar 5. Contoh Data Komentar

Setelah data berada di dalam MysQL, tahap berkutnya adalah proses *cleaning* yang dilakukan dengan langkah sebagai berikut:

1. Mengatur jenis karakter pada MySQL menjadi UTF-8.
2. Menghapus semua karakter ? setelah proses impor dari CSV karena perbedaan karakter, dengan SQL: `update komentar set komentar=replace(komentar,'?',' ')`
3. Menghapus semua komentar yang jumlah hurufnya <= 1 dengan SQL: `delete from komentar where length(komentar)<=1`
4. Mengubah semua spasi yang lebih dari satu menjadi satu spasi dengan memanggil *stored procedure* DELETE_DOUBLE_SPACES, dengan perintah SQL: `update komentar set komentar=DELETE_DOUBLE_SPACES(komentar)`
5. Menghapus semua kata pada komentar yang memiliki jumlah karakter <= 1, dengan memanggil *stored procedure / function* ALPHA, dengan perintah SQL: `update komentar set komentar=ALPHA(komentar)`
6. Menghapus spasi di awal dan akhir komentar dengan SQL: `update komentar set komentar=trim(komentar).`

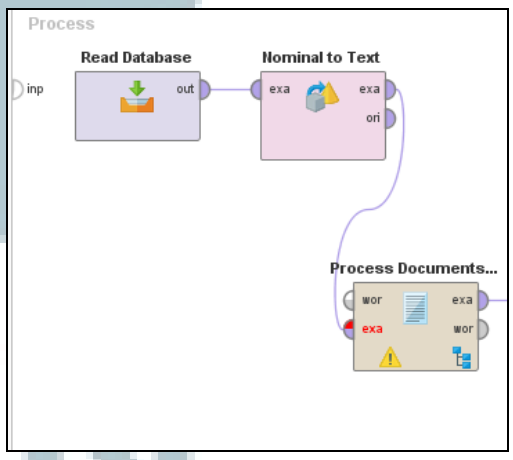
Setelah semua proses *cleaning* data pada MySQL dilakukan maka hasil data terakhir yang siap digunakan untuk proses *learning* klasifikasi berjumlah 16.641 data yang dapat dilihat pada Tabel 5 berikut.

Tabel 5. Hasil Akhir Proses Data *Cleaning*

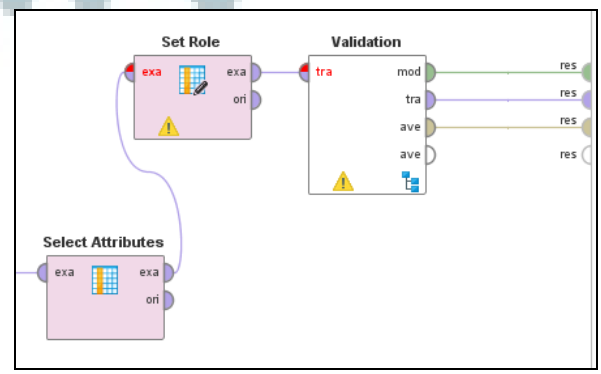
No.	Kelas	Jumlah
1.	Spam	10.399
2.	Not Spam	6.062
	TOTAL	**16.641**

D. *Implementasi Deteksi Spam*

Deteksi spam menggunakan *software* RapidMiner versi 7.5 dengan konfigurasi operator-operator seperti pada Gambar 6 dan 7 berikut.



Gambar 6. Konfigurasi RapidMiner (Bagian 1)

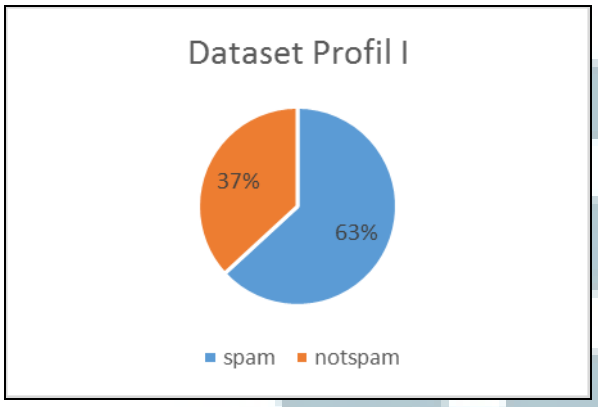


Gambar 7. Konfigurasi RapidMiner (Bagian 2)

E. Pengujian Skenario I (Unbalanced Dataset)

Skenario I adalah pengujian di mana data yang digunakan untuk *training* berjumlah 10.399 untuk data spam dan 6062 untuk data *not* spam. 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.

Dari hasil percobaan menggunakan skenario I (seperti pada Tabel 5 sebelumnya) yang ditampilkan dalam bentuk *pie chart* seperti pada Gambar 8, diperoleh hasil seperti pada Tabel 6.



Gambar 8. Profil Dataset Skenario I

Tabel 6. Hasil *Confusion Matrix* Skenario I menggunakan Naïve Bayes

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

Waktu proses keseluruhan : 12.33 menit
Proses validasi: 10.24 menit

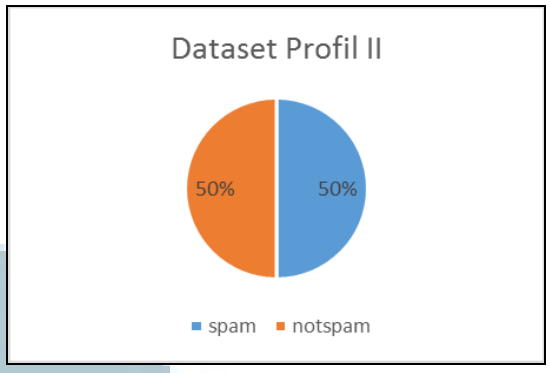
Dari hasil pada Tabel 6 tersebut diperoleh hasil:

- *Accuracy* = 74,31 %
- *Classification error* = 25,69 %
- *Sensitivity (Recall)* = 61,43 %
- *Specificity* = 96,4 %
- *Precision* = 96,71 %
- *F-Measure* = 75, 13 %

Hal ini menujukkan bahwa untuk dataset pelatihan yang tidak seimbang antara spam dan bukan spam akurasi klasifikasinya sudah cukup baik (di atas 70%).

F. Pengujian Skenario II (Balanced Dataset)

Skenario II 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*. 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. Dari hasil percobaan menggunakan skenario II dan ditampilkan seperti dalam bentuk *pie chart* pada Gambar 9 diperoleh hasil pada Tabel 7 sebagai berikut:



Gambar 9. Profil Data Skenario II

Tabel 7. Hasil *Confusion Matrix* Skenario II menggunakan Naïve Bayes

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

Waktu proses keseluruhan: 6.31 menit
Proses validasi: 4.56 menit

Dari hasil pada Tabel 7 tersebut diperoleh hasil:

- *Accuracy* = 77,25 %
- *Classification error* = 22,75 %
- *Sensitivity (Recall)* = 57,21 %
- *Specificity* = 97, 29 %
- *Precision* = 95,48 %
- *F-Measure* = 71,5 %

Hal ini menujukkan bahwa untuk dataset pelatihan yang seimbang antara spam dan bukan spam akurasi klasifikasinya baik (di atas 75%).

Dilihat dari kedua pengujian menggunakan skenario I dan II diperoleh peningkatan akurasi sebesar 2,94% pada data profil yang seimbang. Dari penelitian ini algoritma Naïve Bayes jelas dapat digunakan untuk sistem detektor spam pada data komentar Instagram berbahasa Indonesia. Berdasarkan hal tersebut, beberapa penelitian yang masih dapat dikembangkan dari penelitian ini adalah membangun aplikasi / sistem dalam bentuk service dan diimplementasikan pada plugin browser baik desktop ataupun mobile untuk mendeteksi dan

menandai komentar spam pada Instagram saat pengguna membuka halaman Instagram.

V. SIMPULAN

Berdasarkan penelitian yang telah dilakukan maka diperoleh beberapa kesimpulan sebagai berikut:

1. Penelitian ini sudah berhasil mengimplementasikan algoritma Naïve Bayes pada studi kasus deteksi komentar spam menggunakan data Instagram berbahasa Indonesia. Tingkat akurasi yang didapatkan sudah cukup baik di atas 75%, yaitu 77,25%.
2. Penelitian ini telah menguji kemampuan algoritma klasifikasi Naive Bayes dan diperoleh akurasi sebesar 74,31 % untuk skenario I (*dataset* tidak seimbang) dan akurasi sebesar 77,25% untuk skenario II (*dataset* seimbang). Terjadi peningkatan keakuratan sebesar 2,94 % untuk *dataset* seimbang.
3. Tahapan *preprocessing* 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 *emoticon*.

Sedangkan saran-saran yang masih perlu dikerjakan untuk penelitian selanjutnya adalah:

1. Tahap *stemming* masih perlu dilakukan dan diuji apakah memiliki pengaruh terhadap hasil akurasi atau tidak.
2. Melakukan tahapan perbandingan dengan algoritma lain yaitu menggunakan metode Support Vector Machine.
3. Dapat dikembangkan lebih lanjut ke pengembangan *plugin browser* yang secara otomatis mendeteksi komentar spam saat menampilkan halaman Instagram dan menandainya sehingga pengguna mengetahuinya.

UCAPAN TERIMA KASIH

Terima kasih penulis berikan kepada Lembaga Penelitian dan Pengabdian Kepada Masyarakat Universitas Kristen Duta Wacana (UKDW) Yogyakarta yang telah memberikan dana penelitian untuk tahun 2017.

DAFTAR PUSTAKA

[1] E. Prakasa, *Cek Toko Sebelah*. Jakarta: Starvision Plus, 2016.

[2] M. Eskelinen, "How to get rid of spam on Instagram", *Mervi Emilia*, 2015. [Online]. Available: http://merviemilia.com/blog/2015-02-how-to-get-rid-of-spam-on-instagram. [Accessed: 24- Jan- 2017].

[3] D. Tamir, "How To Protect Yourself From Instaspam - ReadWrite", *ReadWrite*, 2015. [Online]. Available: http://readwrite.com/2015/04/15/instagram-spam-instaspam-how-to-avoid/. [Accessed: 24- Jan- 2017].

[4] M. Webster, "Definition of SPAM", *Merriam-webster.com*, 2016. [Online]. Available: https://www.merriam-webster.com/dictionary/spam. [Accessed: 26- Jan- 2016].

[5] Geerthik S., "Survey on Internet Spam: Classification and Analysis", *Int.J.Computer Technology & Applications*, vol. 4, no. 3, pp. 384-391, 2013.

[6] E. Convey, "Porn sneaks way back on web", *The Boston Herald*, 1996.

[7] K. Hines, "How to Identify and Control Blog Comment Spam. Kissmetrics Blog", *Kissmetrics Blog*, 2012.

[8] G. Mishne, D. Carmel and R. Lempel, "Blocking Blog Spam with Language Model Disagreement", in *The First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, Chiba, Japan, 2005.

[9] N. Spirin and J. Han, "Survey on web spam detection", *ACM SIGKDD Explorations Newsletter*, vol. 13, no. 2, p. 50, 2012.

[10] S. Raschka, "Naive Bayes and Text Classification {I} - Introduction and Theory", *CoRR*, vol. 14105329, no. 14105329, pp. 1-20, 2014.

[11] D. Sculey and G. Wachman, "Relaxed Online SVMs for Spam Filtering", in *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Amsterdam, The Netherlands, 2007, pp. 415--422.

[12] A. Rachmat and Y. Lukito, "SENTIPOL: Dataset Sentimen Komentar Pada Kampanye PEMILU Presiden Indonesia 2014 dari Facebook Page", in *Konferensi Nasional Teknologi Informasi dan Komunikasi 2017*, Univerasitas Kristen Duta Wacana, 2016, pp. 218-228.

[13] M. Duggan, N. Ellison, C. Lampe, A. Lenhart and M. Madden, "Social Media Update 2014", *Pew Research Center: Internet, Science & Tech*, 2014. [Online]. Available: http://www.pewinternet.org/2015/01/09/social-media-update-2014/. [Accessed: 30- Jan- 2017].

[14] Instagram, "Instagram Developer Documentation", *Instagram.com*. [Online]. Available: https://www.instagram.com/about/. [Accessed: 28- Jan- 2016].

[15] S. Roncero-Menendez, "8 Ways to Use Instagram's API", *Mashable*, 2017. [Online]. Available: http://mashable.com/2013/09/19/instagram-api-uses/#mSrayzI6Dmq3. [Accessed: 28- Feb- 2016].

[16] E. This, "Everything You Need to Know About Instagram API Integration | CSS-Tricks", *CSS-Tricks*, 2016. [Online]. Available: https://css-tricks.com/everything-need-know-instagram-api-integration/. [Accessed: 28- Jan- 2016].

[17] S. M. Weiss, N. Indurkhya, T. Zhang and F. Damerau , Text mining: Predictive Methods for Analyzing Unstructured Information, New York: Springer, 2005.

[18] J. Han and M. Kamber, *Data mining*. Haryana, India: Elsevier, 2012.

[19] U. DBD, "Confusion Matrix", *Www2.cs.uregina.ca*, 2017. [Online]. Available: http://www2.cs.uregina.ca/~dbd/cs831/notes/confusion_matrix/confusion_matrix.html. [Accessed: 28- Jan- 2016].

[20] T. Bolander, "Bolandish/PHP-Instagram-Grabber", *GitHub*, 2016. [Online]. Available: https://github.com/Bolandish/PHP-Instagram-Grabber. [Accessed: 01- Feb- 2017].

[21] M. Deoranje, "musdeoranje.net: 10+ Akun Instagram Dengan Followers Terbanyak Di Indonesia", *musdeoranje.net*, 2015. [Online]. Available: http://www.musdeoranje.net/2016/08/akun-instagram-dengan-followers-terbanyak-di-indonesia.html. [Accessed: 8- Feb- 2017].