turnitin

5304-Research Results-16557-1-2-20211010.docx
Oct 11, 2021
3674 words / 23571 characters

5304-Research Results-16557-1-2-20211010.docx

Sources Overview

14%

OVERALL SIMILARITY

1	www.neliti.com INTERNET	2%
2	ejournal.upnjatim.ac.id INTERNET	2%
3	core.ac.uk INTERNET	<1%
4	jurnal.untan.ac.id INTERNET	<1%
5	link.springer.com INTERNET	<1%
6	Nelson Ivan Herrera Herrera. "Big Data architecture proposal for vehicular traffic detection", 2020 International Conference of Digital Tr... CROSSREF	<1%
7	ijain.org INTERNET	<1%
8	turcomat.org INTERNET	<1%
9	kinetik.umm.ac.id INTERNET	<1%
10	ejurnal.teknokrat.ac.id INTERNET	<1%
11	publications.aston.ac.uk INTERNET	<1%
12	Javier Pastor-Galindo, Mattia Zago, Pantaleone Nespoli, Sergio Lopez Bernal et al. "Spotting Political Social Bots in Twitter: A Use Cas... CROSSREF	<1%
13	sersc.org INTERNET	<1%
14	Silvia Angela Mansi, Giovanni Barone, Cesare Forzano, Ilaria Pigliautile, Maria Ferrara, Anna Laura Pisello, Marco Arnesano. "Measurin... CROSSREF	<1%
15	Avisha Das, Shahryar Baki, Ayman El Aassal, Rakesh Verma, Arthur Dunbar. "SoK: A Comprehensive Reexamination of Phishing Resea... CROSSREF	<1%
16	V Oktaviani, B Warsito, H Yasin, R Santoso, Suparti. "Sentiment analysis of e-commerce application in Traveloka data review on Googl... CROSSREF	<1%
17	jurnal.ubl.ac.id INTERNET	<1%

18	digilib.uinsby.ac.id INTERNET	<1%
19	www.scribd.com INTERNET	<1%
20	doaj.org INTERNET	<1%
21	Cruz-Monteagudo, M.. "Stochastic molecular descriptors for polymers. 4. Study of complex mixtures with topological indices of mass... CROSSREF	<1%
22	Lusa Indah Prahartiwi, Wulan Dari. "Komparasi Algoritma Naive Bayes, Decision Tree dan Support Vector Machine untuk Prediksi Peny... CROSSREF	<1%
23	Muhammad Rizki Fahdia, Dwiza Riana, Fachri Amsury, Irwansyah Saputra, Nanang Ruhyana. "Komparasi Algoritma Klasifikasi untuk O... CROSSREF	<1%
24	docplayer.net INTERNET	<1%
25	ejurnal.undana.ac.id INTERNET	<1%
26	worldcomp-proceedings.com INTERNET	<1%
27	www.frontiersin.org INTERNET	<1%

Excluded search repositories:

- None

Excluded from Similarity Report:

- Bibliography
- Quotes
- Small Matches (less than 10 words).

Excluded sources:

- None

J-ICON, Vol. 1 No. 1, Maret 2013, pp. 1~10
DOI

Implementasi Pendeteksian Spam Email Menggunakan Metode Text Mining dengan Algoritma Bayes Naïve dan Desecion Tree J48

Rizka Safitri Lutfiyani[1], Niken Retnowati[2]

[1]Program Studi Manajemen Informatika, Universitas Widya Darma Klaten, Jl. Ki Hajar Dewantoro Klaten
Email: Rizka.s.lutfiyani@gmail.com
[2]Program Studi Teknik Informatika, Universitas Widya Darma Klaten, Jl. Ki Hajar Dewantoro Klaten
Email: Retnowati.niken@yahoo.co.id

ABSTRAK

Email cukup populer sebagai salah satu media komunikasi digital. Hal tersebut dikarenakan proses pengiriman pesan dengan email yang mudah. Sayangnya, kebanyakan pesan dalam email adalah email spam. Spam adalah pesan yang tidak diinginkan penerima pesan karena spam biasanya berisi pesan iklan maupun pesan penipuan. Salah satu cara untuk menyortir pesan-pesan tersebut adalah dengan melakukan pengklasifikasian pesan email menjadi spam maupun ham. Ham adalah pesan yang diinginkan penerima pesan. Penelitian ini menggunakan Algoritma Klasifikasi seperti Naïve Bayes dan Decision Tree J48 untuk mengklasifikasikan pesan email. Metode yang digunakan adalah teks mining. Data yang berisi teks pesan email berbahasa inggris akan diproses terlebih dahulu sebelum diklasifikasikan dengan Naïve Bayes dan Decision Tree J48. Tahap pra proses tersebut meliputi tokenisasi, pembuangan *stopword list*, *stemming,* dan seleksi atribut. Selanjutnya, data teks pesan email akan diproses dengan Algoritma Naïve Bayes dan *Decision Tree* J48. Algoritma Naïve bayes adalah algoritma pengklasifikasi yang berdasarkan pada Teori Keputusan Bayesian sedangkan Algoritma *Decision Tree* J48 ialah pengembangan dari Algoritma *Decision Tree* ID3. Hasil penelitian ini adalah Algoritma *Decision Tree* J48 mendapat akurasi yang lebih tingggi dari Algoritma Naïve Bayes. Algoritma *Decision Tree* J48 mendapat 93.117% sedangkan Naïve Beyes memiliki akurasi 88,5284%.
Kata sandi: Text Mining, Decision Tree, dan Naïve Bayes.

ABSTRACT

Email is quite popular as a digital communication media. This is because the message sending process via email is easy. Unfortunately, most messages in emails are spam emails. Spam is a message that the recipient of the message does not want because spam usually contains advertising messages or fraudulent messages. One way to sort these messages is to classify email messages into spam or ham. Ham is the message that the recipient wants. This study uses Classification Algorithms such as Naïve Bayes and Decision Tree J48 to classify email messages. The method used is text mining. Data containing the text of the email message in English will be processed before being classified with Naïve Bayes and Decision Tree J48. The pre-process stage includes tokenization, disposal of stopword lists, stemming, and attribute selection. Furthermore, Data text for email message will be processed using the Naïve Bayes Algorithm and Decision Tree J48. The Naïve Bayes Algorithm is a classification algorithm based on Bayesian Decision Theory, while the J48 Decision Tree Algorithm is the development of the ID3 Decision Tree Algorithm. The result of this research is that the Decision Tree J48 algorithm gets higher accuracy than the Naïve Bayes Algorithm. The Decision Tree J48 algorithm has an accuracy of 93.117% while Naïve Beyes has an accuracy of 88.5284%.
Keywords: Text Mining, Decision Tree, and Naïve Bayes.

1. PENDAHULUAN

Menurut Asosiasi Penyelenggara Jasa Internet Indonesia (APJII), komunikasi lewat pesan menjadi alasan ke-2 terbanyak seseorang menggunakan layanan internet[1]. Salah satu media komunikasi lewat pesan berbasis internet yang cukup popular adalah Email. Email menjadi tujuan utama dari pemakaian layanan internet berlangganan di kantor[1].

Salah satu alasan email menjadi cukup populer sebagai salah satu media komunikasi digital adalah kemudahan proses pengiriman pesan[2]. Asalkan pengirim pesan memiliki akun email dan alamat email yang dituju, pengirim email dapat melakukan pengiriman pesan ke penerima pesan. Pemilik email

J-ICON, Vol. 1 No. 1, Maret 2013, pp. 1~10
DOI

tidak dimintai persetujuan dalam penerimaan pesan. Akibatnya, pemilik email dapat menerima berbagai macam pesan, baik yang diinginkan maupun diinginkan. Pesan-pesan yang tidak diinginkan biasa disebut sebagai spam.

Tidak ada pengertian jelas mengenai spam[3]. Beberapa berpendapat bahwa spam adalah pesan komersial yang dikirimkan tanpa persetujuan penerima pesan. Pendapat lain mengatakan bahwa spam adalah pesan komersial yang palsu. Pengertian longgar dari spam ialah pesan maupun postingan dengan konten apapun yang dikirim ke banyak penerima yang tidak meminta pesan tersebut secara khusus[3].

Berdasarkan laporan CISCO, ada sekitar 85 persen dari seluruh pesan email yang dikirimkan di April 2019 dapat diklasifikasikan sebagai spam[4]. Hal tersebut tentu saja menjadi masalah. Salah satu upaya mengatasi spam adalah dengan melakukan penyortiran spam.

Penyortiran tersebut dapat dilakukan dengan berbagai algoritma. Salah satu algoritma yang dapat digunakan adalah Naïve Bayes. Algoritma Naïve Bayes adalah salah satu algorima klasifikasi. Naïve bayes mudah dibangun dan tidak menggunakan skema rumit[5]. Decision tree bersama Naïve Bayes juga merupakan 10 teratas dalam data mining [5].

Penelitian ini bertujuan membandingkan efektifitas algoritma Naïve Bayes dan Decision tree dalam penyortiran email spam. Penelitian ini diharapkan dapat memberikan masukan mengenai algoritma yang tepat dalam penyortilan email spam.

2. MATERI DAN METODE

EMAIL SPAM

Awalnya spam hanya ditujukan pada pesan email tapi kini seiring dengan perkembangan teknologi spam pun mengalami evolusi. Spam tidak hanya berwujud email tapi spam kini dapat ditemui dalam berbagai bentuk, seperti web maupun multimedia. Spam sendiri berarti upaya untuk menyalahgunakan atau memanipulasi suatu sistem tekno-sosial dengan membuat atau menyuntikan konten yang tidak diminta dan atau tidak diinginkan yang bertujuan untuk mengarahkan perilaku manusia atau sistem demi keuntungan jangka panjang maupun jangka pendek dari spammer baik secara langsung maupun tidak langsung[6].

Sedangkan, pesan spam berarti *unsolicited e-mails* atau pesan email yang tidak diminta oleh pemilik dari email[2][7]. Spam terdiri dari beberapa tipe, yaitu spam iklan, Nigerian spam dan *phishing*[8]. Nigerian spam merupakan spam yang digunakan penipu untuk memeras penerima spam[8]. Sedangkan *phishing* adalah suatu tindakan untuk memperoleh informasi yang sensitive atau rahasia dari pengguna dengan membangun replica dari situs web organisasi resmi[9].

Spam iklan bertujuan untuk mempromosikan produk, servis, ataupun konten tertentu. Metode iklan tersebut tidaklah berbeda dari metode yang dilakukan sebelumnya, iklan yang tidak diminta dikirimkan pada penerima pesan[6]. Pesan iklan tersebut dapat memenuhi kotak masuk dan menghabiskan *bandwith* internet yang ada.

Pada tipe lain, pesan spam dapat dijadikan alat penyebaran virus atau malware. Pesan tersebut digunakan untuk mendapatkan hak pengguna dalam sistem. Hak tersebutlah yang kemudian dipakai untuk menyusup dalam sistem dan melakukan penyerangan[8]. Kebalikan dari pesan span adalah pesan ham.

Text Mining

Text mining ialah penemuan baru oleh komputer yang sebelumnya merupakan informasi yang tidak diketahui dengan cara menekstraksi informasi secara otomatis dari berbagai sumber tertulis[10]. Pada *text mining*, pola ditemukan tidak berasal dari rekaman basis data yang terformat tetapi dari data tekstual tidak terstuktur dalam kumpulan dokumen[11]. Arsitektur fungsional dari text mining memiliki beberapa tahap, yaitu *prepocessing task, processed document collection,* dan *Core mining operation* dan *presentation*[11].

Preprocessing Text Mining

Pada intinya, tahap *preprocesing* bertujuan untuk menyimpulkan atau mengekstrak representasi terstuktur dari data mentah yang tidak terstruktur dalam text mining[11]. Penelitian ini menggunakan dataset email berbahasa inggris yang berasal dari kaggle.com[12]. Dataset tersebut kemudian diubah struktur agar dapat digunakan di aplikasi Weka. File dataset tersebut terdiri dari 1495 pesan email ham dan 1495 pesan email spam.

Untuk membedakan data email spam dan ham, setiap data dalam data set diberi label. Jika label suatu data adalah spam, data tersebut adalah data email yang bersifat spam. Sebal iknya, jika data label ialah ham, data email tersebut adalah data email ham.

Data email tersebut kemudian dibersihkan dari data yang bukan teks (angka, tanda baca) data teks email kemudian dipotong-potong berdasarkan kata penyusunnya. Tahap pemotongan data teks

ISSN:2337-7631 (printed)
ISSN: 2654-4091 (Online)

J-ICON, Vol. 1 No. 1, Maret 2013, pp. 1~10
DOI

disebut sebagai tokenisasi. Selain tokenisasi, proses *preprocessing* dalam text mining juga meliputi pembuangan kata kurang penting(*stop list*) dan penyimpanan kata penting(*word list*). Stopword yag digunakan dalam penelitian ini berasal github.com.

Selanjutnya, data teks tersebut dihilangkan imbuhan yang melekat pada kata atau dicari dasarnya. Proses tersebut disebut *Stemming*. Penelitian ini menggunakan Metode Lovins Stemming. Metode Lovin Stemming ialah salah satu metode *stemming* penghapusan akhiran yang berdasarkan pada prinsip *longest-match*[13]. Metode ini memiliki keunggulan dalam kecepatan, kemampuan dalam menentukan akar kata jamak yang tidak beraturan, serta menghilangkan huruf ganda[14].

Pembobotan Kata

Sebelum diolah dengan Algoritma Naive Bayes dan J48, kata-kata dalam dokumen tersebut dicari bobotnya. Pembobotan kata atau *Term Weighting* ialah pemberian bobot pada kata yang berdasarkan frekuensi kemunculan kata tersebut dalam dokumen. Pembobotan kata pada penelitian ini menggunakan persamaan TF-IDF. TF-IDF merupakan gabungan dari dua skema pembobotan yaitu Term Frequency (TF) dan Inverse Document Frequenci (IDF). Jika sebuah kata dalam sebuah dokumen memiliki kemunculan yang sering dan kemunculan kata tersebut dalam dokumen lain jarang, bobok kata tersebut akan semakin tinggi. Sebaliknya, jika sebuah kata dalam sebuah dokumen memiliki kemunculan yang jarang dan kemunculan kata tersebut dalam dokumen lain sering, kata tersebut memiliki bobot yang semakin rendah. PErsamaan TF-IDF adalah sebagai berikut [15]:

$$W_{j,i} = \frac{n_{j,i}}{\sum k n_{k,i}} \times log_2 \frac{D}{d_j} \quad (1)$$

$n_{j,i}$ adalah jumlah kemunculan kata j dalam dokumen i sedangkan $\sum k n_{k,i}$ adalah keseluruhan dari jumlah kemunculan kata dalam dokumen i. D ialah jumlah dokumen yang digunakan dan d_j ialah jumlah dari dokumen yang menggunakan kata j.

Attribut Selection

Tidak semua kata diujicobakan dengan Naïve Bayes maupun Decision Tree J48. Kata-kata tersebut harus melewati proses Attribut Selectioan. Pada penelitian ini, Metode Atribut selection yang digunakan adalah rangking sedangkan untuk pengevaluasinya adalah Gain Ratio. Nilai Gain Ratio dari setiap kata akan di rangking untuk dipilih yang dapat digunakan sebagai data training. Gain Ratio didapatkan dengan membagi Gain dengan SplitInfo[16]. Gain maupun SplitInfo didapatkan melalui langkah-langkah dibawah ini :

$$I(S) = -\sum_{i=1}^{m} p_i \log_2(p_i) \quad (1)$$

$I(S)$menunjukan informasi yang diharapkan atau entropi dari S data training dengan m kelas yang berbeda. Sedangkan p_i ialah probabilitas sampel milik kelas C_i yang diestimasi oleh s_i/s. Misalnya, A adalah atribut yang memiliki v nilai yang berbeda dan s_y adalah jumlah sampel dari kelas C_i di himpunan bagian S_j. S_j merupakan sampel di S yang memiliki nilai a_j dari A. Entropi dari himpunan bagian A ditunjuk oleh persamaan (2) sedangkan *Gain(A)* ditunjuk persamaan (3).

$$E(A) = -\sum_{i=1}^{m} I(S) \frac{s_{1i}+s_{2i}+\cdots+s_{mi}}{s} \quad (2)$$

$$Gain(A) = I(S) - E(A) \quad (3)$$

$$SplitInfo_A(S) = -\sum_{i=1}^{v} (|S_i|/|S|) \log_2(|S_i|/|S|) \quad (4)$$

Persamaan (4) merupakan persamaan dari SplitInfo untuk A. SplitInfo adalah entropi yang dihasilkan dengan membagi dataset traning S menjadi partisi v yang sesuai dengan v dari pengujian atribut[16]. Sedangkan Gain Ratio dihasilkan dari persamaan (5)

$$Gain\ Ratio(A) = \frac{Gain(A)}{SplitInfo_A(S)} \quad (5)$$

Cross-Validation

Pengujian Algoritma Naïve Bayes dan J48 pada penelitian ini menggunakan metode Cross-Validation. Cross-Validation ialah metode pengambilan sampel ulang data untuk menguji kemampuan

J-ICON, Vol. 1 No. 1, Maret 2013, pp. 1~10
DOI

dari suatu model prediktif serta untuk mencegah *overfitting*[17][18]. *Overfitting* terjadi karena *noise*, ukuran set pelatihan yang terbatas, serta kompleksitas dari pengklasifikasi[19].

Penelitian ini menggunakan *10-fold Cross-Validation* yang berarti dataset akan dibentuk kedalam 10 subset. 9 subset data akan digunakan sebagai *training set* dan 1 subset data akan menjadi *testing set* kemudian akan diiterasi sebanyak 10 kali. Hasil pengujian adalah rata-rata dari 10 iterasi tersebut.

Naïve Bayes

Naïve Bayes ialah pengklasifikasi *probabilistic* sederhana yang berdasarkan pada Teori Keputusan Bayesian[20]. Naïve bayes mengasumsikan bahwa semua atributnya (x1 sampai dengan xn) independen terhadap atribut lain. Perhitungan probabilitas dari Naïve Bayes yang didasarkan teori Bayes adalah sebagai berikut[21] :

$$P(C = c_i \mid D = d_j) = \frac{P(C=c_i \cap D=d_j)}{P(D=d_j)} \quad \text{.....(6)}$$

$P(C = c_i \mid D = d_j)$ adalah kategori dari pobabilitas jika dokumen diketahui. Persamaan (6) dapat dibuat persamaan (7)

$$P(C = c_i \mid D = d_j) = \frac{P(C=c_i \cap D=d_j)}{P(D=d_j)}$$

$$= \frac{P(D=d_j \mid C=c_i) \times P(C=c_i)}{P(D=d_j)} \quad \text{.....(7)}$$

$P(D = d_j \mid C = c_i)$ ialah nilai probabilitas dari kejadian dokumen d_j jika dokumen diketahui memiliki kategori c_i sedangkan $P(C = c_i)$ adalah nilai probabilitas dari kejadian dokumen c_i. $P(D = d_j)$ merupakan nilai probabilitas dari kejadian dokumen d_j. Proses Klasifikasi teks dimulai dengan menentukan kategori $c \in C$ dari dokumen $d \in D$ dengan $C=\{c_1,c_2,c_3 \ldots c_i\}, D=\{d_1,d_2,d_3 \ldots d_j\}$, dan $P(C = c_i \mid D = d_j)$ mempunyai nilai maksimal $P\{P(C = c_i \mid D = d_j) \mid c \in C \; dan \; d \in D\}$.

$$P(C = c_i \mid D = d_j) = \frac{\sqcap_k P(W_k \mid C=c_i) \times P(C=c_i)}{p(W_1,W_2,W_3,\ldots,W_k,\ldots,W_n)} \quad \text{.....(8)}$$

Karena Naïve Bayes mengasumsikan bahwa dokumen ialah kata-kata korpus yang menyusun dokumen itu sendiri dan tidak memperhatikan urutan kemunculan kata-kata dalam dokumen, persamaan (7) dapat menjadi persamaan (8) dengan $\sqcap_k P(W_k \mid C = c_i)$ ialah perhitungan dan perkalian dari probabilitas kemunculan semua kata dokumen d_j.

Model probabilistik dari dokumen pelatihan dibuat selama proses klasifikasi dengan menghitung nilai $P(W_{kj} \mid C)$ sehingga probabilitas semua nilai dapat ditentukan dengan persamaan (9) dan (10) [21].

$$P(W_k = W_{kj} \mid C) = \frac{D_b(W_k = W_{kj} \mid C)+1)}{D_b(c)+|V|} \quad \text{.....(9)}$$

$$P(W_k = W_{kj} \mid C) = \frac{D_b(c)}{|D|} \quad \text{.....(10)}$$

$D_b(W_k = W_{kj} \mid C)$ ialah fungsi yang mengembalikan jumlah dokumen b dalam kategori c, yang mempunyai nilai kata $wk = wkj$ sedangkan $D_b(c)$ adalah fungsi yang mengembalikan jumlah dokumen b yang mempunyai kategori c, dan |V| besarnya kemungkinan nilai wkj. Laplace Smooting sering dikominasikan dengan$D_b(W_k = W_{kj} \mid C)$ untuk mencegah nilai menjadi 0.

$$C^* = \prod_{c \in C}^{argmax} k \; P\,(W_k \mid c) \times P(c) \quad \text{.....(11)}$$

$D_b(c)$adalah fungsi yang mengembalikan jumlah dokumen b dengan kategori c dan jumlah semua pelatihan dapat diekspresikan dengan |D|. Pemberian kategori pada text dilakukan dengan memilih

J-ICON, Vol. 1 No. 1, Maret 2013, pp. 1~10
DOI

nilai c maksimum dari $P(C = c_i \mid D = d_j)$ seperti pada model (11). C* adalah kategori yang memiliki nilai $P(C = c_i \mid D = d_j)$ maksimum[22].

Decision Tree J48

Decision Tree ialah metode analisis keputusan dengan mendapatkan dan membandingkan probabilitas dari *leave* dan *node* yang berbeda yang kemudian dievaluasi[23]. Model terbaik adalah model dengan kelayakan tertinggi. Algoritma J48 yang digunakan dalam penelitian ini adalah algoritma yang berbasis pada Algoritma ID3 dengan beberapa peningkatan seperti kemampuannya menangani data yang tidak lengkap. Algoritma ini menggunakan teori informasi, yaitu *information entropy* dan *information gain degree* sebagai standar pengukuran.

Algoritma ID3 ialah algoritma mengambil *information gain (IG)* sebagai standar ukur maupun kriteria selama melakukan optimasi atribut keputusan sementara J48 menggunakan *Splitinfo* dan *Information gain rate (IGR)* [23]. *IG, Splitinfo* dan *IGR* didapatkan melalui persamaan dibawah ini :

$$IG(attr) = Entropy(S) - Entropy\ (S|Attr) \quad (12)$$

$$SplitInfo_{Attr}(S) = -\sum_{v=v_0}^{v}\left(\frac{|S_v|}{|S|}\log_2\frac{|S_v|}{|S|}\right) \quad (13)$$

$$IGR(attr) = \frac{IG(attr)}{SplitInfo(attr)} \quad (14)$$

Berdasarkan persamaan (12), (13), dan (14), jika suatu atribut memiliki lebih banyak kemungkinan nilai, *SplitInfo* yang dihasilkan akan lebih besar sehingga keputusan akhir dari algoritma J48 menjadi lebh sedikit dipengaruhi oleh atribut dengan lebih banyak kemungkinan nilai dan memperoleh pohon keputusan lebih akurat.

3. HASIL DAN PEMBAHASAN

Decision Tree J48

Pengujian Algoritma Decision Tree J48 menggunakan metode *10-Fold cross validation.* Metode tersebut menghasilkan Pohon Keputusan seperti Gambar 1. Ada sebanyak 71 *leaves* yang terbentuk dengan 141 *nodes*. Waktu yang dibutuhkan untuk melakukan pengujian adalah 112.16 detik.

```
enron <= 0
|  meter <= 0.705059
|  |  nom <= 0.919893
|  |  |  dar <= 0.624786
|  |  |  |  hpl <= 0
|  |  |  |  |  april <= 1.339269
|  |  |  |  |  |  wednesda <= 1.469177
|  |  |  |  |  |  |  attach <= 0.822083
|  |  |  |  |  |  |  |  pm <= 0.44346
|  |  |  |  |  |  |  |  |  ga <= 0.596968
|  |  |  |  |  |  |  |  |  |  deal <= 1.130211
|  |  |  |  |  |  |  |  |  |  |  bob <= 0
|  |  |  |  |  |  |  |  |  |  |  |  contract <= 0.62658
|  |  |  |  |  |  |  |  |  |  |  |  |  hop <= 1.408346
|  |  |  |  |  |  |  |  |  |  |  |  |  |  chang <= 1.139158
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  fw <= 0
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  jl <= 0
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  les <= 1.495372
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  frida <= 0
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  night <= 0
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  cal <= 1.037656
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  communic <= 0.93375: spam (1272.0/45.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  communic > 0.93375
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  bud <= 0
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  dav <= 0: spam (21.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  dav > 0: ham (2.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  bud > 0: ham (2.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  cal > 1.037656
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  assist <= 0.939856
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  extens <= 0.812767
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  hotmail <= 1.371406
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  quest <= 1.196344: spam (27.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  quest > 1.196344
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  ag <= 0.931833: ham (2.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  ag > 0.931833: spam (2.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  hotmail > 1.371406: ham (2.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  extens > 0.812767: ham (2.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  assist > 0.939856: ham (2.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  night > 0
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  applic <= 0
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  asap <= 0: spam (21.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  asap > 0: ham (2.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  applic > 0: ham (2.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  frida > 0
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  ens <= 1.234817: spam (21.0/1.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  ens > 1.234817: ham (4.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  les > 1.495372
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  thank <= 0.562097
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  follow <= 0.958226
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  avail <= 2.584201
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  th <= 0.417555: spam (27.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  th > 0.417555: ham (3.0/1.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  avail > 2.584201: ham (2.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  follow > 0.958226: ham (2.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  thank > 0.562097: ham (5.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  jl > 0
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  quest <= 0.715132
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  farmer <= 0
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  plant <= 2.467829: spam (18.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  plant > 2.467829: ham (2.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  farmer > 0: ham (2.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  quest > 0.715132: ham (3.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  fw > 0
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  pleas <= 0
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  am <= 0: spam (20.0/1.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  am > 0: ham (3.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  pleas > 0: ham (6.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  chang > 1.139158
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  thank <= 0.330991
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  bid <= 1.715745: spam (23.0/2.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  bid > 1.715745: ham (2.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  thank > 0.330991: ham (13.0/2.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  hop > 1.408346
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  thank <= 0.268232
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  week <= 1.618666
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  luck <= 0.930387: spam (17.0/1.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  luck > 0.930387: ham (2.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  week > 1.618666: ham (4.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  thank > 0.268232: ham (5.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  contract > 0.62658
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  contract <= 0.999924: spam (11.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  contract > 0.999924
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  nam <= 0.648421: ham (12.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  nam > 0.648421: spam (2.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  bob > 0
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  abl <= 0
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  bar <= 1.220515
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  caus <= 1.008297
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  sam <= 1.225581
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  mone <= 0: ham (17.0/1.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  mone > 0: spam (2.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  sam > 1.225581: spam (3.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  caus > 1.008297: spam (2.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  bar > 1.220515: spam (2.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  abl > 0: spam (2.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  deal > 1.130211
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  fre <= 0.927654
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  cost <= 1.18832
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  giv <= 1.345374
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  offer <= 0.869429
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  com <= 0.881793: ham (31.0/2.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  com > 0.881793: spam (3.0/1.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  offer > 0.869429: spam (2.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  giv > 1.345374: spam (2.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  cost > 1.18832: spam (2.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  fre > 0.927654: spam (3.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  ga > 0.596968
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  il <= 0
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  es <= 1.184948: ham (22.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  es > 1.184948: spam (3.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  il > 0: spam (4.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  pm > 0.44346
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  click <= 1.199734
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  feel <= 0.639797
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  eas <= 0.812798: ham (38.0/2.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  eas > 0.812798: spam (2.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  feel > 0.639797: spam (3.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  click > 1.199734: spam (3.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  attach > 0.822083
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  pr <= 0.880618
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  mor <= 1.184463: ham (33.0/2.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  mor > 1.184463: spam (2.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  pr > 0.880618: spam (3.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  wednesda > 1.469177
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  beaut <= 0: ham (23.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  beaut > 0: spam (2.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  april > 1.339269: ham (28.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  hpl > 0: ham (2
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  dar > 0.624786
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  link <= 0.567453
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  giv <= 1.090096: ham (123.0/2.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  giv > 1.090096
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  dec <= 0.784835: ham (5.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  dec > 0.784835: spam (2.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  link > 0.567453
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  click <= 0.539962: spam (2.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  click > 0.539962: ham (5.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  nom > 0.919893: ham (187.0/3.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  meter > 0.705059: ham (213.0)
enron > 0: ham (623.0)
```

J-ICON, Vol. 1 No. 1, Maret 2013, pp. 1~10
DOI

Gambar 1. Pohon Keputusan yang Dibuat Algoritma Decision Tree J48

Hasil pengujian ditunjukan oleh Tabel 1. Berdasarkan tabel tersebut, jumlah data email yang dikategorikan secara benar adalah 2786 atau 93.1773 % sedangkan data yang dikategorikan secara tidak benar berjumlah 204 atau 6.8227 %. Matrik dari pengkategorian dapat dilihat pada Gambar 2, yaitu 1368 pesan email ham dikategorikan sebagai ham dan 172 pesan email ham dikategorikan sebagai spam. Sedangkan 1418 pesan email spam dikategorikan sebagai spam dan 77 pesan email spam dikategorikan sebagai ham.

Tabel 1. Hasil dari Pengujian dengan Algoritma J48

	Hasil
Correctly Classified Instance	2786 (93.1773%)
Incorrectly Classified Instance	204 (6.8227 %)
Total instances	2990

```
=== Confusion Matrix ===

   a    b   <-- classified as
1368  127 |    a = ham
  77 1418 |    b = spam
```

Gambar 2. Matrik yang Diperoleh pada Algoritma Decision Tree J48

Akuasi detail oleh kelas pada Algoritma J48 ditunjukan oleh Tabel 2. Berdasarkan table tersebut hanya FP Rate saja yang mendapatkan hasil rendah sedangkan TP Rate, Precision, Recall, F-Measure mendapatkan hasil yang tinggi. Berikut penjelasan tiap kelas[24] :

1. TP Rate adalah penapsiran dari data yang bernilai positif dan berhasil diklasifikasikan di kelas positif. Nilai rata-rata dari TP Rate Ham dan Spam algoritma ini adalah 0.932.
2. FP Rate adalah penapsiran dari data yang bernilai negative dan diklasifikasikan di kelas positif. Hasil rata-rata dari FP Rate Ham dan Spam algoritma ini adalah 0.068.
3. Precision ialah perbandingan dari akurasi klasifikasi yang diinginkan oleh user dan jawaban yang dibagikan sistem. Nilai rata-rata Precision Ham dan Spam algoritma ini adalah 0.932.
4. Recall ialah derajat keberhasilan sistem untuk mendapatkan informasinya kembali. Nilai rata-rata Recall Ham dan Spam algoritma ini adalah 0.932.
5. F- Measure ialah perhitungan dari rata-rata precision dan recall. Nilai rata-rata F-Measure Ham dan Spam algoritma ini adalah 0.932.

Hal tersebut menunjukan akurasi tinggi yang didapat Algoritma J48 meskipun algoritma ini memiliki struktur yang sederhana dan interpretasi dan visualisasi yang mudah[25].

Tabel 2. Akurasi Detail oleh Kelas pada Algoritma J48

	TP Rate	***FP Rate***	***Precision***	***Recall***	***F-Measure***	***Class***
	0.915	0.052	0.947	0.915	0.931	Ham
	0.948	0.085	0.918	0.948	0.933	Spam
Weighted Avg.	0.932	0.068	0.932	0.932	0.932	

Naïve Bayes

J-ICON, Vol. 1 No. 1, Maret 2013, pp. 1~10
DOI


```
                         Class
Attribute                  ham    spam
                          (0.5)  (0.5)
=====================================
abl
  mean                   0.0829 0.0452
  std. dev.              0.3896 0.2838
  weight sum               1495   1495
  precision              0.0379 0.0379

abov
  mean                   0.1074 0.0652
  std. dev.              0.4243 0.2931
  weight sum               1495   1495
  precision              0.0173 0.0173

accept
  mean                   0.0255 0.0694
  std. dev.              0.2352 0.3469
  weight sum               1495   1495
  precision              0.0415 0.0415
```

Gambar 3. Salah Satu Contoh Isi Tabel Yang Dibuat Algoritma Naive Beyes

Pengujian dengan metode *10-Fold cross validation* pada Naïve Bayes menghasilkan table seperti di Gambar 5. Model tersebut diperoleh dengan waktu 6.92 detik. Kesimpulan dari hasil pengujian Naive Bayes ditunjukan oleh Gambar 6. Berdasarkan gambar tersebut, jumlah data email yang dikategorikan secara benar adalah 2647 atau 88.5284 % sedangkan data yang dikategorikan secara tidak benar berjumlah 259 atau 8.6622 %. Ada sebanyak 84 pesan email atau 2.8094 % yang tidak dapat dikategorikan oleh algoritma ini.

Matrik yang terbentuk dari algoritma ini ditunjukan oleh Gambar 4. Berdasarkan gambar tersebut 1367 pesan ham dikategorikan sebagai pesan ham sedangkan sisanya sebanyak 72 pesam ham dikategorikan sebagai spam. Sementara itu sebanyak 1280 pesan spam dikategorikan sebagai pesan spam dan 187 pesan spam dikategorikan sebagai ham.

Tabel 3. Hasil dari Pengujian dengan Algoritma Naïve Bayes

	Hasil
Correctly Classified Instance	2647 (88.5284%)
Incorrectly Classified Instance	259 (8.6622 %)
Unclassified instances	84 2.8094 %
Total instances	2990

```
=== Confusion Matrix ===

   a    b   <-- classified as
1367   72 |    a = ham
 187 1280 |    b = spam
```

Gambar 4. Matrik yang Diperoleh pada Algoritma Naïve Bayes

Detail akurasi berdasarkan kelas ditunjukan oleh Tabel 4. TP Rate, Precision, Recall, dan F-Measure rata-rata Naive Bayes tinggi yaitu 0.911, 0.914, 0.911, dan 0.911. Sementara itu, FP Rate rata-rata Naive Bayes cukup rendah, yakni 0.088. Akurasi Naive Bayes tergolong tinggi.

Naive Bayes juga termasuk algoritma yang mudah diterapkan dan diinterprestasikan[5] . Algoritma ini tidak membutuhkan skema estimasi parameter iteratif rumit tapi sering kali berhasil dengan baik[5].

Tabel 4. Akurasi Detail oleh Kelas

J-ICON, Vol. 1 No. 1, Maret 2013, pp. 1~10
DOI

	TP Rate	*FP Rate*	*Precision*	*Recall*	*F-Measure*	*Class*
	0.950	0.127	0.880	0.950	0.913	Ham
	0.873	0.050	0.947	0.873	0.908	Spam
Weighted Avg.	0.911	0.088	0.914	0.911	0.911	

Perbandingan Akurasi Naïve Bayes dan Decision Tree J48

Tabel 5. Perbandingan Akurasi Decision Tree J48 dan Naïve Bayes

	Correctly Classified Instance	**Incorrectly Classified Instance**
Decision Tree J48	93.1773%	6.8227 %
Naïve Bayes	88.5284%	8.6622 %

Tabel 5. menunjukkan perbandingan akurasi Decision Tree J48 dan Naïve Bayes. Berdasarkan tabel tersebut, Algoritma Decision Tree J48 menunjukan keunggulannya daripada Algoritma Naïve Bayes dalam data email. Algoritma Decision Tree mendapatkan *Correctly Classified Instance* sebesar 93.1773% sedangkan Naïve bayes mendapatkan sebesar 88.5284%. Begitu juga pada *Incorrectly Classified Instance*, Algoritma Decision Tree menperoleh sebesar 6.8227 %. Angka tersebut lebih kecil jika dibandingkan dengan angka yang didapat Naïve bayes, yaitu sebesar 8.6622 %. Hal tersebut, akibat dari algoritma Decision Tree yang mendapat keuntungan dengan data yang tidak lengkap dan *noisy* [26].

4. KESIMPULAN DAN SARAN

Metode Teks Mining untuk mengklasifikasi email Ham dan Spam dapat dilakukan dengan Algoritma J48 dan Naive Bayes. Algoritma J48 yang merupakan algoritma *Decision tree* menghasilkan pohon keputusan dengan 71 *leaves* dan 141 *nodes*. Naïve Bayes menghasilkan tabel seperti di Gambar 3.

Pengujian menggunakan Algoritma Decision Tree J48 menunjukan hasil yang lebih baik dari pada Naive Bayes. Hal tersebut menunjukan keunggulan Algoritma Decision Tree J48 dibanding Naive Bayes pada data email. Keunggulan tersebut dapat dilihat melalui *Correctly Classified Instance* Algoritma J48 sebesar 93.1773%. Naïve Bayes memiliki *Correctly Classified Instance* 88.5284% dengan 2.8094 % data yang tak terclasifikasi.

Penelitian ini juga diharapkan dapat dikembangkan untuk algoritma-algoritma klasifikasi lain, seperti SVM dan K-Means.

DAFTAR PUSTAKA

[1] A. W. Irawan, A. Yusufianto, D. Agustina, and R. Dean, "Laporan Survei Internet APJII 2019 – 2020," 2020. [Online]. Available: https://apjii.or.id/survei.

[2] J. Batra, R. Jain, V. A. Tikkiwal, and A. Chakraborty, "A comprehensive study of spam detection in e-mails using bio-inspired optimization techniques," *Int. J. Inf. Manag. Data Insights*, vol. 1, no. 1, p. 100006, 2021, doi: 10.1016/j.jjimei.2020.100006.

[3] J. Qadri, "SPAM -- Technological and Legal Aspects," 2011.

[4] CISCO, "Email: Click with Caution," 2019.

[5] X. Wu *et al.*, "Top 10 algorithms in data mining," *Knowl. Inf. Syst.*, vol. 14, no. 1, pp. 1–37, 2008, doi: 10.1007/s10115-007-0114-2.

[6] E. Ferrara, "The history of digital spam," *Commun. ACM*, vol. 62, no. 8, pp. 82–91, 2019, doi: 10.1145/3299768.

[7] O. Saad, A. Darwish, and R. Faraj, "A survey of machine learning techniques for Spam filtering," *J. Comput. Sci.*, vol. 12, no. 2, pp. 66–73, 2012.

[8] Y. Kontsewaya, E. Antonov, and A. Artamonov, "Evaluating the Effectiveness of Machine Learning Methods for Spam Detection," *Procedia Comput. Sci.*, vol. 190, no. 2019, pp. 479–486, 2020, doi: 10.1016/j.procs.2021.06.056.

[9] A. A. Akinyelu and A. O. Adewumi, "Classification of phishing email using random forest machine learning technique," *J. Appl. Math.*, vol. 2014, no. April, 2014, doi: 10.1155/2014/425731.

[10] K. Borgwardt and C. Biology, "What is text mining ?," 2010. http://people.ischool.berkeley.edu/~hearst/text-mining.html (accessed Apr. 24, 2015).

ISSN:2337-7631 (printed)
ISSN: 2654-4091 (Online)

J-ICON, Vol. 1 No. 1, Maret 2013, pp. 1~10
DOI

[11] R. Feldman and J. Sanger, *The Text Mining Handbook Advanced Approaches in Analyzing Unstructured Data*. New York: Cambridge University Press, 2007.
[12] "Spam Mails Dataset | Kaggle." https://www.kaggle.com/venky73/spam-mails-dataset.
[13] J. B. Lovins, "Development of a Stemming Algorithm," *Mech. Transl. Comput. Linguist.*, vol. 11, no. 1, pp. 22–31, 1968.
[14] S. Yucebas and R. Tintin, "Govdeturk: A novel turkish natural language processing tool for stemming, morphological labelling and verb negation," *Int. Arab J. Inf. Technol.*, vol. 18, no. 2, pp. 148–157, 2021, doi: 10.34028/IAJIT/18/2/3.
[15] G. Salton and C. Buckley, "Term-Weifhting Approaches in Automatic Text Retrieval," *Inf. Process. Manag.*, 1988, doi: 10.1016/0306-4573(88)90021-0.
[16] A. G. Karegowda, A. S. Manjunath, G. Ratio, and C. F. Evaluation, "COMPARATIVE STUDY OF ATTRIBUTE SELECTION USING GAIN RATIO," *Int. J. Inf. Technol. Knowl. Manag.*, vol. 2, no. 2, pp. 271–277, 2010.
[17] T. Hastie, R. Tibshirani, J. Frie, and Dman, *The Elemen of Statistical Learning Data mining, Inference, and Prediction*, 2nd ed. california, 2008.
[18] R. O. Duda, P. E. Hart, D. G. Stork, and J. Wiley, "Pattern Classification All materials in these slides were taken from Pattern Classification (2nd ed)," no. April, 2016.
[19] X. Ying, "An Overview of Overfitting and its Solutions," *J. Phys. Conf. Ser.*, vol. 1168, no. 2, 2019, doi: 10.1088/1742-6596/1168/2/022022.
[20] T. K. Bhowmik, "Naive bayes vs logistic regression: Theory, implementation and experimental validation," *Intel. Artif.*, vol. 18, no. 56, pp. 14–30, 2015, doi: 10.4114/ia.v18i56.1113.
[21] T. M. Mitchell, "GENERATIVE AND DISCRIMINATIVE CLASSIFIERS : NAIVE BAYES AND LOGISTIC REGRESSION Learning Classifiers based on Bayes Rule," in *Machine Learning*, vol. 1, no. Pt 1-2, 2010, pp. 1–17.
[22] V. Oktaviani, B. Warsito, H. Yasin, R. Santoso, and Suparti, "Sentiment analysis of e-commerce application in Traveloka data review on Google Play site using Naïve Bayes classifier and association method," *J. Phys. Conf. Ser.*, vol. 1943, no. 1, 2021, doi: 10.1088/1742-6596/1943/1/012147.
[23] H. Fan, "Network Activities Recognition and Analysis Based on Supervised Machine Learning Classification Methods Using J48 and Naïve Bayes Algorithm."
[24] A. H. Rakhmah and T. A. Putri, "Analisis Sentimen Terhadap Pasangan Calon Presiden 2019 Pada Media Sosial Twitter," *J. Lentera Ict*, no. ISSN 2338-3143, pp. 1–11, 2019.
[25] S. Cepeda and S. García-garcía, "Advantages and limitations of intraoperative ultrasound strain elastography applied in brain tumor surgery : a single-center experience," 2021.
[26] Y. N. Feng, Z. H. Xu, J. T. Liu, X. L. Sun, D. Q. Wang, and Y. Yu, "Intelligent prediction of RBC demand in trauma patients using decision tree methods," *Mil. Med. Res.*, vol. 8, no. 1, pp. 1–12, 2021, doi: 10.1186/s40779-021-00326-3.

J-ICON Jurnal Komputer & Informatika
ISSN:2337-7631 (printed)
ISSN: 2654-4091 (Online)