IMPLEMENTASI PENDETEKSIAN *SPAM* EMAIL MENGGUNAKAN METODE *TEXT MINING* DENGAN ALGORITMA NAÏVE BAYES DAN *DECISION TREE* J48

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

[1]Program Studi Manajemen Informatika, Universitas Widya Darma Klaten,
Jl. Ki Hajar Dewantoro Klaten, Indonesia
Email: Rizka.s.lutfiyani@gmail.com
[2]Program Studi Teknik Informatika, Universitas Widya Darma Klaten,
Jl. Ki Hajar Dewantoro Klaten, Indonesia
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. *Ham* adalah pesan yang diinginkan penerima pesan. Salah satu cara untuk menyortir pesan-pesan tersebut adalah dengan melakukan pengklasifikasian pesan email menjadi *spam* maupun *ham*. Naïve Bayes dan *decision tree* J48 ialah algoritma yang dapat digunakan untuk mengklasifikasikan pesan email. Oleh karena itu, penelitian ini bertujuan membandingkan efektifitas algoritma Naïve Bayes dan *decision tree* J48 dalam penyortiran email *spam*. Metode yang digunakan adalah *text 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 *stop word 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%. Kesimpulan dari penelitian ini adalah algoritma *decision tree* J48 lebih unggul dibanding Naive Bayes untuk menyortir email *spam* jika dilihat dari tingkat akurasi masing-masing algoritma.
Kata kunci: *Text mining, Decision tree*,.

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. Ham is the message that the recipient wants. One way to sort these messages is to classify email messages into spam or Ham. Naïve Bayes and decision tree J48 are the algorithms that can be used to classify email messages. Therefore, this study aims to compare the effectiveness of the Naïve Bayes algorithm and decision tree J48 in sorting spam emails. 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 stop word 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 Bayes has an accuracy of 88,5284%. The conclusion of this study is that the decision tree J48 algorithm is superior to Naive Bayes for sorting spam emails when viewed from the level of accuracy of each algorithm.
Keywords: Text Mining, Decision Tree, 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 populer adalah amail. 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 tidak dimintai persetujuan dalam penerimaan pesan. Akibatnya, pemilik email dapat menerima berbagai macam pesan, baik yang tidak diinginkan maupun yang 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 karena kapasitas email yang terbatas. Pesan email penting pun dapat tertimbun oleh pesan-pesan *spam* sehingga pesan penting tersebut justru tidak tersampaikan oleh penerimanya. Salah satu upaya mengatasi *spam* adalah dengan melakukan penyortiran *spam*.

Penyortiran tersebut dapat dilakukan dengan berbagai metode. Penelitian ini menggunakan *text mining*. *Text mining* merupakan proses ekstraksi informasi dari berbagai sumber tertulis [5]. Pesan email juga merupakan salah satu sumber tertulis sehingga penelitian ini menggunakan *text mining* untuk penyortiran pesan email *spam*. Salah satu algoritma yang dapat digunakan dalam *text mining* adalah Naïve Bayes. Algoritma Naïve Bayes adalah salah satu algoritma klasifikasi. Naïve Bayes mudah dibangun dan tidak menggunakan skema rumit [6]. *Decision* tree bersama Naïve Bayes juga merupakan 10 teratas dalam *data mining* [6]. Diharapkan dengan menggunakan kedua algoritma yang termasuk 10 teratas, penyortiran *spam* menjadi lebih efektif sehingga menjadi salah satu solusi untuk mengatasi masalah *spam* email.

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 menyuntikkan 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 [7].

Sedangkan, pesan *spam* berarti *unsolicited e-mails* atau pesan email yang tidak diminta oleh pemilik dari email [2][8]. *spam* terdiri dari beberapa tipe, yaitu *spam* iklan, Nigerian *spam* dan *phishing* [9]. Nigerian *spam* merupakan *spam* yang digunakan penipu untuk memeras penerima *spam* [9]. Sedangkan *phishing* adalah suatu tindakan untuk memperoleh informasi yang sensitive atau rahasia dari pengguna dengan membangun replica dari situs web organisasi resmi [10].

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 [7]. 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 [9]. Kebalikan dari pesan *spam* adalah pesan *ham*.

Text Mining

Text mining ialah penemuan baru oleh komputer yang sebelumnya merupakan informasi yang tidak diketahui dengan cara mengekstraksi informasi secara otomatis dari berbagai sumber tertulis [11]. Pada *text mining*, pola ditemukan tidak berasal dari rekaman basis data yang terformat tetapi dari data tekstual tidak terstuktur dalam kumpulan dokumen [12]. Arsitektur fungsional dari *text mining* memiliki

beberapa tahap, yaitu *prepocessing task, processed document collection,* dan *core mining operation* dan *presentation* [12].

Metode Penelitian

Untuk melakukan proses *text mining*, penelitian ini menggunakan aplikasi Weka sedangkan *dataset* email didapatkan dari kaggle.com [12]. File *dataset* tersebut terdiri dari 1495 pesan email *ham* dan 1495 pesan email *spam* berbahasa dataset. Awalnya, *dataset* tersebut bertipe data csv. Agar dapat dioleh dengan Weka *dataset* tersebut diubah tipe datanya menjadi arff. Untuk mengubah file csv menjadi arff, file csv dibuka dengan Weka Explorer dan disimpan dalam bentuk arff. Berikut langkah-langkah *preprocessing, atribut selection,* Naïve Bayes dan J48

Preprocessing Text Mining

Sebelum melakukan *preprocessing,* untuk membedakan data email *spam* dan *ham*, setiap data dalam *dataset* diberi label. Jika label suatu data adalah *spam*, data tersebut adalah data email yang bersifat *spam*. Sebaliknya, jika data label ialah *ham*, data email tersebut adalah data email *ham*.

Tahap *preprocesing* bertujuan untuk menyimpulkan atau mengekstrak representasi terstuktur dari data mentah yang tidak terstruktur dalam *text mining* [12]. Proses *preprocessing* dilakukan dengan membuka *dataset* email di Preprosess, Weka Explorer. Di dalam Choose pilih Weka, Filters, Unsupervised, Attribute, StringToWordVector. Untuk memulai proses *preprocesing,* pilih Apply.

Di tahap tersebut, *dataset* email tersebut dibersihkan dari data yang bukan teks (angka, tanda baca) data teks email kemudian dipotong-potong berdasarkan kata penyusunnya. Tahap pemotongan data teks disebut sebagai tokenisasi. Selain tokenisasi, proses *preprocessing* dalam *text mining* juga meliputi pembuangan kata kurang penting (*stop list*) dan penyimpanan kata penting (*word list*). *Stop word* yang 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].

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 *term frequency-inverse document frequency* (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, bobot 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 tersebuet memiliki bobot yang semakin rendah. Persamaan TF-IDF ditenjukan pada persamaan 1 [15].

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

dengan $n_{j,i}$ adalah jumlah kemunculan kata *j* dalam dokumen *i* sedangkan $\sum kn_{k,i}$ 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*.

Attribute Selection

Tidak semua kata diujicobakan dengan Naïve Bayes maupun *decision tree* J48. Kata-kata tersebut harus melewati proses *attribute selection.* Pada penelitian ini, metode *attribute selection* yang digunakan adalah *ranking* sedangkan untuk pengevaluasinya adalah *gain ratio*. Proses ini dilakukan di Weka Explorer, Select Attributes. Pada Atribute Evaluator pilih GainRatioAttributeEval. Saat memilih GainRatioAttributeEval, Weka mengarahkan Search Method untuk berubah menjadi Ranker. Untuk memulai proses *attribute selection,* pilih Apply.

Nilai Gain Ratio dari setiap kata akan diurutkan berdasarkan peringkat untuk dipilih yang dapat digunakan sebagai *data training*. Gain Ratio didapatkan dengan membagi Gain dengan SplitInfo [16]. Gain maupun SplitInfo didapatkan melalui persamaan 2.

$$I(S) = -\sum_{i=1}^{m} p_i \log_2(p_i) \quad (2)$$

dengan $I(S)$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 3 sedangkan *Gain(A)* ditunjuk persamaan 4.

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

$$E(A) = -\sum_{i=1}^{m} I(S)\frac{s_{1i}+s_{2i}+\cdots+s_{mi}}{s} \quad (3)$$

$$Gain(A) = I(S) - E(A) \quad (4)$$

$$SplitInfo_A(S) = -\sum_{i=1}^{v}(|S_i|/|S|)\log_2(|S_i|/|S|) \quad (5)$$

Persamaan 5 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 6.

$$Gain\ Ratio(A) = \frac{Gain(A)}{SplitInfo_A(S)} \quad (6)$$

Naïve Bayes

Tahap ini dilakukan dengan membuka *dataset* email yang telah melalui proses *pre-processing* di Preproses, Weka explorer. Selanjutnya, karena Naïve Bayes merupakan algoritma klasifikasi maka pilih Classify. Pada Classifier pilih Choose, Weka, Classifiers, Bayes, dan NaiveBayes. Pada Test Options pilih Cross-validation Folds 10. Langkah selanjutnya adalah memilih Start untuk memulai.

Naïve Bayes ialah pengklasifikasi *probabilistic* sederhana yang berdasarkan pada teori keputusan Bayesian [17]. Naïve Bayes mengasumsikan bahwa semua atributnya (x_1 sampai dengan x_n) independen terhadap atribut lain. Perhitungan probabilitas dari Naïve Bayes yang didasarkan teori Bayes [18].

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

$P(C = c_i \mid D = d_j)$ adalah kategori dari pobabilitas jika dokumen diketahui. Persamaan 7 dapat dibuat persamaan 8.

$$P(C = c_i \mid D = d_j) = \frac{P(C=c_i \cap D=d_j)}{P(D=d_j)} = \frac{P(D=d_j \mid C=c_i)\times P(C=c_i)}{P(D=d_j)} \quad (8)$$

$P(D = d_j \mid C = c_i)$ 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 $P\{P(C = c_i \mid D = d_j) \mid c \in C \ dan \ d \ \in D\}$.

$$P(C = c_i \mid D = d_j) = \frac{\sqcap_k P(W_k \mid C=c_i)\times P(C=c_i)}{p(W_1,W_2,W_3,\ldots,W_k,\ldots,W_n)} \quad (9)$$

Karena Naïve Bayes mengasumsikan bahwa dokumen ialah kata-kata korpus yang menyusun dokumen dan tidak memperhatikan urutan kemunculan kata-kata dalam dokumen, persamaan 8 dapat menjadi persamaan 9 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 10 dan 11 [18].

$$P(W_k = W_{kj} \mid C) = \frac{D_b(W_k = W_{kj} \mid C)+1)}{D_b(c)+|V|} \quad (10)$$

$$P(W_k = W_{kj} \mid C) = \frac{D_b(c)}{|D|} \quad (11)$$

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

$$C^* = \prod_{c \in C}^{argmax} k\ P\ (W_k|c) \times P(c) \quad (12)$$

$D_b(c)$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 nilai c maksimum dari $P(C = c_i \mid D = d_j)$ seperti pada persamaan 12. C* adalah kategori yang memiliki nilai $P(C = c_i \mid D = d_j)$ maksimum [19].

***Decision Tree* J48**

Decision Tree J48 termasuk algoritma klasifikasi seperti halnya Naive Bayes. Untuk menggunakan algoritma ini, buka Weka explorer dan pilih Classify. Pada Classifier pilih Choose, Weka, Classifiers, Trees, J48. Untuk Test options, pilih Cross-validation Folds 10. Pilih Start untuk memulai proses.

Decision Tree ialah metode analisis keputusan dengan mendapatkan dan membandingkan probabilitas dari *leave* dan *node* yang berbeda yang kemudian dievaluasi [20]. 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

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

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) [20]. IG, Splitinfo dan IGR didapatkan melalui persamaan 13, 14 dan 15.

$$IG(attr) = Entropy(S) - Entropy\ (S|Attr) \quad (13)$$

$$SplitInfo_{Attr}(S) = -\sum_{v=v_0}^{v}(\frac{|S_v|}{|S|}\log_2\frac{|S_v|}{|S|}) \quad (14)$$

$$IGR(attr) = \frac{IG(attr)}{SplitInfo(attr)} \quad (15)$$

Berdasarkan persamaan 13, 14, dan 15, jika suatu atribut memiliki lebih banyak kemungkinan nilai, *SplitInfo* yang dihasilkan akan lebih besar sehingga keputusan akhir dari algoritma *decision tree* J48 menjadi lebih sedikit dipengaruhi oleh atribut dengan lebih banyak kemungkinan nilai dan memperoleh pohon keputusan lebih akurat.

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 dari suatu model prediktif serta untuk mencegah *overfitting* [21][22]. *Overfitting* terjadi karena *noise*, ukuran *set* pelatihan yang terbatas, serta kompleksitas dari pengklasifikasi [23].

Penelitian ini menggunakan *10-fold cross-validation* yang berarti *dataset* akan dibentuk ke dalam 10 *subset*. Sembilan (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.

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.

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 127 pesan email *ham* dikategorikan sebagai *spam*. Sedangkan 1418 pesan email *spam* dikategorikan sebagai *spam* dan 77 pesan email *spam* dikategorikan sebagai *ham*.

Akurasi detail oleh kelas pada algoritma J48 ditunjukan oleh tabel 2. Berdasarkan tabel 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]:

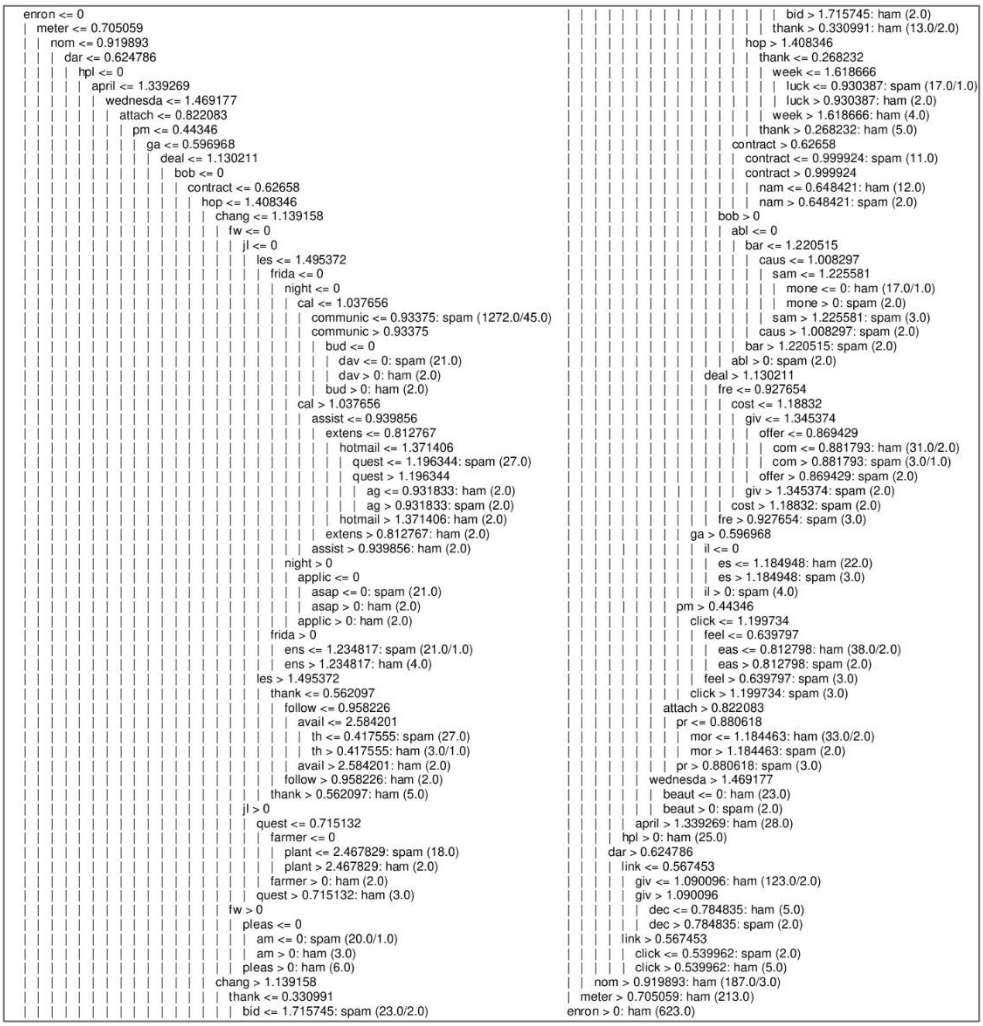
1. *TP-Rate* adalah penapsiran dari data yang bernilai positif dan berhasil diklasifikasikan di kelas positif. Nilai rata-rata dari *TP-Rate ham* dan *spam* algoritma ini adalah 0,932.
2. *FP-Rate* adalah penapsiran dari data yang bernilai negatif dan diklasifikasikan di kelas positif. Hasil rata-rata 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 1. Hasil dari pengujian dengan algoritma J48

	Hasil
Correctly Classified Instance	2786 (93,1773%)
Incorrectly Classified Instance	204 (6,8227 %)
Total instances	2990

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



Gambar 1. Pohon Keputusan yang dibuat algoritma *decision tree* J48

```
=== Confusion Matrix ===

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

Gambar 2. Matrik yang diperoleh pada algoritma *decision tree* J48

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

Pengujian dengan metode *cross-validation* dengan 10 *fold* pada Naïve Bayes menghasilkan tabel seperti di gambar 3. Pada gambar tersebut, Attribute menunjukan kata dalam pesan email yang diuji. Sebagai contoh: kata 'abl' memiliki *mean* 0,0829 untuk kelas *ham* dan 0,0452 maka kata 'abl' memiliki nilai rata-rata sebesar 0,0829 dari seluruh atribut di kelas *ham* dan nilai rata-rata sebesar 0,0452 dari seluruh atribut di kelas *spam*. Std. dev pada baris selanjutnya menunjukan nilai standar deviasi atau nilai simpangan baku atribut tersebut di kelas *ham* dan di kelas *spam*. Pada contoh atribut 'abl' memiliki nilai standar deviasi sebesar 0,3896 untuk kelas *ham* dan 0,2838 untuk kelas *spam*. Weight sum menunjukan bobot kata atribut tersebut, misalnya atribut 'abl' memiliki bobot kata 1495 untuk kelas *ham* dan kelas *spam*. Sedangkan *precision* menunjukan rasio prediksi benar positif atribut jika dibandingkan dengan semua atribut yang

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

diprediksi positif sebagai contoh atribut abl memiliki precision sebesar 0,0379 untuk kelas *ham* maupun *spam*.

```
                          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 Bayes

Model tersebut diperoleh dengan waktu 6,92 detik. Kesimpulan dari hasil pengujian Naive Bayes ditunjukan oleh tabel 3. 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 [6]. Algoritma ini tidak membutuhkan skema estimasi parameter iteratif rumit tapi sering kali berhasil dengan baik [6].

Tabel 4. Akurasi Detail oleh Kelas

	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 menunjukkan perbandingan akurasi *decision tree* J48 dan Naïve Bayes. Berdasarkan tabel tersebut, algoritma *decision tree* J48 menunjukan keunggulannya daripada algoritma Naïve Bayes dalam

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

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* memperoleh *incorrectly classified instance* 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].

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 %

4. KESIMPULAN DAN SARAN

Metode *text mining* untuk mengklasifikasi email *ham* dan *spam* dapat dilakukan dengan algoritma J48 dan Naive Bayes. Algoritma J48 yang merupakan algoritma *decision tree* J48 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 terklasifikasi.

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/surveii.

[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," University of Khasmir Srinagar, 2011.

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

[5] T. Kwartler, *What is Text Mining?* 2017.

[6] 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.

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

[8] 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.

[9] 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.

[10] 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.

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

[12] R. Feldman and J. Sanger, *The Text Mining Handbook Advanced Approaches in Analyzing Unstructured Data*. New York: Cambridge University Press, 2007.

[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. 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.

[18] 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.
[19] 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.
[20] H. Fan, "Network Activities Recognition and Analysis Based on Supervised Machine Learning Classification Methods Using J48 and Naïve Bayes Algorithm."
[21] T. Hastie, R. Tibshirani, J. Frie, and Dman, *The Elemen of Statistical Learning Data mining, Inference, and Prediction*, 2nd ed. california, 2008.
[22] 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.
[23] 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.
[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.