Terakreditasi SINTA Peringkat 2
Surat Keputusan Dirjen Penguatan Riset dan Pengembangan Ristek Dikti No. 10/E/KPT/2019
masa berlaku mulai Vol. 1 No. 1 tahun 2017 s.d Vol. 5 No. 3 tahun 2021

Terbit online pada laman web jurnal: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 4 No. 2 (2020) 377 - 383 ISSN Media Elektronik: 2580-0760

Optimasi Nilai K pada Algoritma KNN untuk Klasifikasi Spam dan Ham Email

Eko Puji Laksono[1], Achmad Basuki[2], Fitra Abdurrachman Bachtiar[3]
[1,2,3]Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
[1]ekopujilaksono@student.ub.ac.id, [2]achmadbasuki@gmail.com, [3]fitraabdurrachman@gmail.com

Abstract

There are many cases of email abuse that have the potential to harm others. This email abuse is commonly known as spam, which contains advertisements, phishing scams, and even malware. This study purpose to know the classification of email spam with ham using the KNN method as an effort to reduce the amount of spam. KNN can classify spam or ham in an email by checking it using a different K value approach. The results of the classification evaluation using confusion matrix resulted in the KNN method with a value of K = 1 having the highest accuracy value of 91.4%. From the results of the study, it is known that the optimization of the K value in KNN using frequency distribution clustering can produce high accuracy of 100%, while k-means clustering produces an accuracy of 99%. So based on the results of the existing accuracy values, the frequency distribution clustering and k-means clustering can be used to optimize the K-optimal value of the KNN in the classification of existing spam emails.

Keywords: *classification, email spam, KNN, frequency distribution clustering, k-means clustering*

Abstrak

Terdapat banyak kasus penyalahgunaan email yang berpotensi merugikan orang lain. Email yang disalahgunakan ini biasa dikenal sebagai email sampah atau spam yang mana email tersebut berisikan iklan, *phising*, scam, bahkan *malware*. Penelitian ini bertujuan untuk mengetahui pengklasifikasian email spam dengan ham menggunakan metode KNN sebagai upaya mengurangi jumlah spam. KNN dapat mengklasifikasikan spam atau ham pada email dengan cara melakukan pengecekan menggunakan pendekatan nilai K yang berbeda. Hasil evaluasi klasifikasi menggunakan *confusion matrix* menghasilkan bahwa metode KNN dengan nilai K=1 memiliki nilai akurasi paling tinggi sebesar 91.4%. Dari hasil penelitian diketahui bahwa optimasi nilai K pada KNN menggunakan distribusi frekuensi *clustering* menghasilkan akurasi yang tinggi sebesar 100%, sedangkan k-means *clustering* menghasilkan akurasi sebesar 99%. Jadi berdasarkan hasil nilai akurasi yang didpatkan, distribusi frekuensi *clustering* dan k-means *clustering* dapat digunakan untuk mengoptimasi nilai K optimal pada KNN dalam klasifikasi spam email.

Kata kunci: klasifikasi, spam email, KNN, distribusi frekuensi *clustering*, k-means *clustering*

1. Pendahuluan

Spam email dapat didefinisikan sebagai *unsolicited bulk* email yaitu email yang dikirimkan kepada ribuan penerima yang mana spam tersebut berisikan iklan, *phising*, *scam*, bahkan *malware*. Hampir di semua aktifitas di internet dapat dengan mudah ditemukan spam. Keberadaan dan sifat spam yang dilakukan terus menerus dan menyampaikan hal yang kurang penting sangat mengganggu dan dapat dibilang cukup meresahkan pengguna internet. Berdasarkan penelitian yang dilakukan Radicati g*roup* jumlah akun email tahun 2019 diperkirakan sebanyak 3,3 miliar akun [1]. Dengan rincian 75% pemilik akun adalah perseorangan atau pribadi, sisanya sebanyak 25% digunakan oleh perusahaan dan diprediksi pada tahun 2020 akan menjadi 4,3 miliar akun. Pada tahun 2019 survey yang dilakukan oleh Ghani & Subekti menemukan bahwa 10% dari email yang diterima oleh suatu perusahaan adalah spam [2]. Pada tahun tersebut Spamcop yang

Diterima Redaksi : 31-03-2020 | Selesai Revisi : 13-04-2020 | Diterbitkan Online : 20-04-2020

menjalankan servis untuk menerima laporan tentang spam juga menerima lebih dari 183 juta laporan kasus spam. Akibatnya banyak pengguna email harus menghabiskan waktu mereka untuk menghapus pesan yang tidak diinginkan tersebut. Hal ini menyebabkan pesan yang penting juga ikut terhapus.

Dalam mengatasi masalah tersebut diperlukan upaya untuk menyaring konten-konten yang dibagikan secara otomatis. Penanganan terkait spam telah dilakukan beberapa penelitian terdahulu. Novelia dkk [1] melakukan penelitian untuk menguji beberapa metode dengan pendekatan machine learning untuk memfilter spam pada email. Metode yang digunakan yaitu KNN dengan nilai K=3, Support Vector Machine dengan kernel RBF yang menggunakan parameter C dan γ (gamma) dan SVM Linier yang hanya menggunakan parameter C. Hasilnya adalah SVM linier memiliki akurasi paling tinggi sebesar 96.6%. Anugroho & Winarno [3] melakukan klasifikasi email spam menggunakan Naïve Bayes Classifier membuktikan bahwa Naïve Bayes mampu mengidentifikasi spam, dengan beberapa syarat dan kondisi secara lebih akurat. Penelitian lainnya dilakukan oleh Fitriyanto & Saifudin [4] untuk mengklasifikasi spam pada email. Metode yang digunakan yaitu K-Nearest Neighbors yang memiliki hasil nilai akurasi paling tinggi dengan nilai k=8 yaitu 85.2% dengan berbagai fitur yang digunakan.

Penggunaan metode yang berbeda membuat hasil yang berbeda pula. Oleh karena itu, penelitian mengenai perbandingan metode klasifikasi terus dilakukan pada berbagai kasus untuk mencari metode yang optimal [5]. Pada penelitian ini dilakukan perbandingan metode klasifikasi dengan kasus membedakan spam dan ham pada email. Metode yang digunakan adalah K-Nearest Neighbors beserta optimasi nilai K menggunakan distribusi frekuensi clustering dan k-means clustering. Kedua optimasi tersebut menggunakan *confusion matrix* untuk evaluasi nilai hasilnya [6].

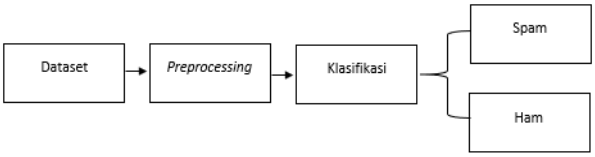
2. Metode Penelitian

Bagian ini merupakan tata urut proses penelitian yang dilakukan antara lain dimulai dari desain sitem, pengumpulan data, *preprocessing*, kemudian klasifikasi menggunakan KNN, serta evaluasi dan validasi hasil yang setiap sub babnya akan dijelaskan di bagian selanjutnya.

2.1. Desain Sistem

Gambaran dari sistem yang dibangun dimulai dengan tahap pengumpulan dataset, kemudian *preprocessing*, proses klasifikasi, dan hasil klasifikasi berupa spam atau ham ditampilkan pada Gambar 1.

Perbedaan spam dan ham (bukan spam) berdasarkan struktur email dapat diklasifikasikan sebagai berikut [1] header, email header menunjukkan informasi perjalanan setiap email. Secara umum, email header terdiri dari pengirim, jaringan dan penerima email. Subject, subject suatu email merupakan suatu judul topik yang mewakili isi pada email. Subject email dapat dijumpai pada header setiap email. Body, pada email body adalah isi dari suatu pesan email, dan dengan adanya body email, pengirim (sender) menyampaikan maksud yang akan disampaikan kepada penerima. Pada penelitian ini selain mengklasifikasikan email spam dari header, dapat pula diklasifikasikan melalui bodynya. Karena dengan body email, dapat ditentukan bahwa email tersebut email yang penting atau tidak.



Gambar 1. Desain Sistem

2.2. Dataset

Data yang digunakan pada penelitian ini berupa data teks yang tersebar ke dalam 5 buah enron yang berjumlah 27716 dokumen email berbahasa Inggris dari 154 pengguna email yang berekstensi .txt. Sumber dokumen email yang digunakan didapatkan dari http://www2.aueb.gr/users/ion/data /enron-spam/. Daftar atribut yang ada pada dataset dapat dilihat pada Tabel 1.

Tabel 1. Daftar Atribut Dalam Dataset

Nama Atribut	Penjelasan Atribut	Tipe data	Rentang nilai / enumerasi
Id Teks	Nama file teks	String	-
Teks	Isi dari email	String	1 -1004
Label	Spam vs. Ham	String	Spam atau Ham

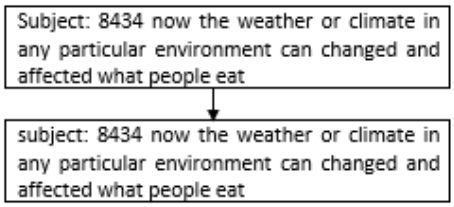
Pada penelitian ini presentase pembagian data *training* dengan data testing sebesar 70% banding 30% dengan total dataset sebanyak 9390 dengan data berlabel spam sebanyak 4695 dan data berlabel ham sebanyak 4695. Contoh isi dari dataset yang digunakan adalah sebagai berikut : "*Subject: 8434 now the weather or climate in any particular environment can changed and affected what people eat*". Sebelum menuju proses klasifikasi, dataset harus di *preprocessing* terlebih dahulu.

2.3. *Preprocessing*

Preprocessing merupakan tahap awal menyiapkan dataset untuk mempermudah pemrosesan data yang bertujuan untuk meningkatkan kualitas data agar mendapatkan hasil dengan tingkat performa yang tinggi. Dalam penelitian ini terdapat proses perubahan dataset yang dilakukan secara manual dari default data yang didapat .txt ke dalam format .csv yang bertujuan agar data yang ada dapat diolah untuk dilanjutkan ke dalam proses klasifikasi. Tahapan dalam preprocessing text adalah sebagai berikut [2] :

1. *Case Folding*

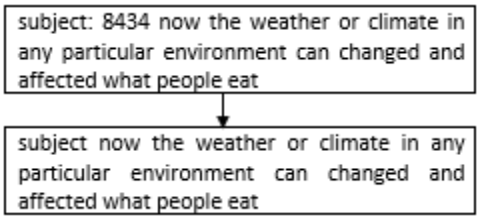
Pada proses *Case Folding* dilakukan perubahan terhadap huruf kapital menjadi huruf kecil. Berikut ini merupakan contoh dan proses *case folding* yang dapat dilihat pada Gambar 2.



Gambar 2. Proses *Case Folding*

2. *Data Cleaning*

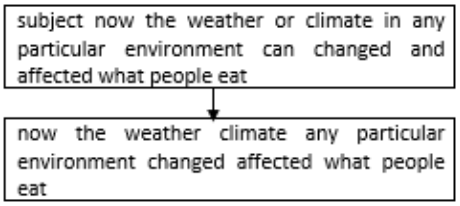
Proses *Data cleaning* digunakan untuk menghilangkan simbol, tanda baca beserta angka. Berikut ini adalah gambaran dari proses *data cleaning* yang ditampilkan pada Gambar 3.



Gambar 3. Proses *Data Cleaning*

3. *Stopword*

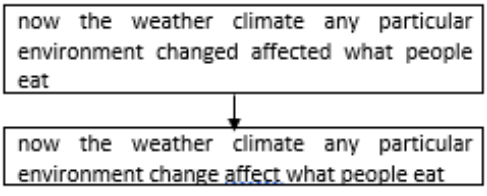
Stopword merupakan proses untuk mengurangi kata-kata yang dianggap tidak perlu atau tidak mempengaruhi informasi dari data yang ada [3]. Berikut ini merupakan contoh dan proses *stopword* yang ditampilkan pada Gambar 4.



Gambar 4. Proses *Stopword*

4. *Stemming*

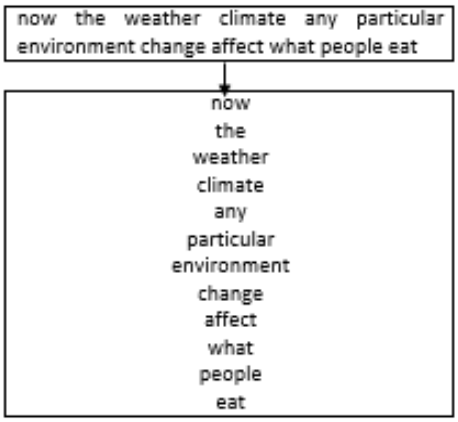
Penerapan untuk proses *stemming* menggunakan model *Porter Stemmer*. Tujuan dari proses ini adalah untuk menyederhanakan kembali kata-kata yang sudah mengalami perubahan [4]. Berikut ini adalah gambaran dari proses *stemming* yang ada pada Gambar 5.



Gambar 5. Proses *Stemming*

5. Tokenisasi

Tokenisasi merupakan proses yang digunakan untuk memotong setiap dokumen menjadi kata-kata yang berdiri sendiri. Berikut ini merupakan contoh dan proses tokenisasi yang ditampilkan pada Gambar 6.



Gambar 6. Proses Tokenisasi

6. Pembobotan TF-IDF

Dalam tahap ini dilakukan vektorisasi berdasarkan *Term Frequency* (TF) pada setiap data untuk membangun fitur atau *attribute* secara keseluruhan. Selanjutnya pembobotan TF dikonversi menjadi bentuk *Term Frequency Inverse Document Frequency* (TF-IDF). TF-IDF merupakan metode untuk menghitung bobot setiap kata berdasarkan frekuensi kemunculan kata. Metode ini akan menghitung nilai TF dan IDF pada setiap kata yang terdapat pada dokumen [5]. Berikut merupakan contoh dari hasil pembobotan TF-IDF yang ditampilkan pada Gambar 7.

Term	TF-IDF									
	D1	D2	D3	D4	D5	D6	D7	D8	D9	10
now	0	0	0	0	0	0	0	159,6	0	0
the	0	0	159,6	0	0	0	0	0	0	0
weather	159,6	0	0	0	0	0	0	0	0	0
climate	0	0	0	0	0	159,6	0	0	0	0
any	0	0	0	0	0	0	0	0	0	159,6
particular	0	0	0	0	0	0	159,6	0	0	0
environment	0	0	0	0	0	0	0	0	159,6	0
change	0	0	0	0	0	0	159,6	0	0	159,6
affect	0	0	0	0	0	0	0	159,6	0	0
what	0	0	0	0	0	0	0	0	159,6	0
eat	0	0	0	0	159,6	0	0	0	0	0

Gambar 7. Hasil Pembobotan TF-IDF

Setelah didapatkan atribut berupa nilai TF-IDF di setiap datanya, proses klasifikasi bisa dilakukan.

2.4. Klasifikasi Menggunakan KNN

K-Nearest Neighbor (KNN) adalah salah satu metode yang dipakai dalam klasifikasi data. Prinsip kerja K-Nearest Neighbor adalah melakukan klasifikasi data berdasarkan kedekatan jarak suatu data dengan data lainnya.

Untuk menggunakan algoritma K-Nearest Neighbors perlu ditentukan banyaknya K tetangga terdekat yang digunakan untuk melakukan klasifikasi data baru. Banyaknya K sebaiknya merupakan angka ganjil,

misalnya K = 1, 2, 3, dan seterusnya [6]. Penentuan nilai K dipertimbangkan berdasarkan banyaknya data yang ada dan dimensi data. Semakin banyak data yang ada, nilai K yang dipilih sebaiknya semakin rendah. Namun semakin besar ukuran dimensi data, nilai K yang dipilih sebaiknya semakin tinggi.

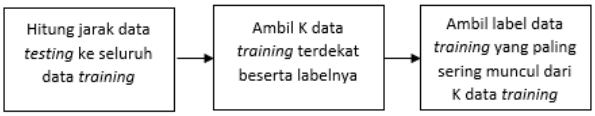
```
Pseudocode KNN

k-Nearest Neighbor
Classify (X,Y,x) // X: training data, Y: class
labels of X, x: unknown sample
for i = to m do
   Compute distance d(Xi, x)
end for
Compute set I containing indices for the k
smallest distances d(Xi, x).
return majority label for [Y, where i € I]
```

Dekat atau jauhnya jarak bisa dihitung dengan besaran jarak. Dalam penelitian ini menggunakan jarak *Euclidean*. Jarak *Euclidean* adalah besarnya jarak suatu garis lurus yang menghubungkan antar objek. Jarak *Euclidean* digunakan karena data teks yang ada sudah mengalami perubahan menjadi angka pada saat proses pembobotan TF-IDF dalam *preprocessing* [7]. Persamaan jarak *Euclidean* ditunjukkan pada persamaan (1).

$$d(x1, x2) = \sqrt{\sum_{i=1}^{p}(x2i - x1i)^2} \qquad (1)$$

dimana *d(x1, x2)* adalah jarak *Euclidean*, *x2i* adalah data uji ke-*i* pada variabel ke-*p*, *x1i* data sampel ke-*i* pada variabel ke-*p* dan *p* adalah dimensi data variabel bebas. Alur dari proses KNN dapat dilihat pada Gambar 8.



Gambar 8. Alur Proses KNN

Setiap data *testing* akan dihitung jaraknya terhadap seluruh data *training*. Perhitungan jarak tersebut dilakukan menggunakan persamaan *Euclidean Distance*. Setelah diperoleh jarak untuk seluruh data *training*, diambil sebanyak K data *training* terdekat dengan label data-data tersebut. Label yang paling sering muncul dari K data *training* tersebut akan menjadi label data *testing* yang sedang diproses. Jika terdapat banyak kemunculan label data *training* yang bernilai sama, maka akan diambil label data *training* terdekat.

2.5. Optimasi Menggunakan Distribusi Frekuensi *Clustering*

Nilai K yang bagus dapat dipilih dengan optimasi parameter untuk meningkatkan akurasi dari KNN [8], salah satu proses optimasi yang dapat dilakukan ini ialah dengan cara mengelompokkan data menggunakan distribusi frekuensi *clustering*. Optimasi dengan distribusi frekunesi *clustering* merupakan pengelompokan data ke dalam beberapa kategori yang menunjukkan banyaknya data dalam setiap kategori. Setiap data tidak dapat dimasukkan ke dalam dua atau lebih kategori. Pengelompokan data menjadi tabulasi menggunakan data class label dan dikaitkan dengan masing-masing frekuensinya [3]. Sehingga hal ini dapat meningkatkan hasil yang ada setelah dilakukan proses optimasi.

2.6. Optimasi Menggunakan K-Means *Clustering*

K-Means merupakan suatu metode data mining yang melakukan proses pemodelan tanpa *supervise* dan juga salah satu metode yang menggunakan metode pengelompokan data secara partisi. Sehingga dengan hal tersebut maka proses optimasi menggunakan K-Means dapat dilakukan untuk meningkatkan hasil dari nilai K optimal yang ada. Optimasi dengan K-Means dilakukan dengan cara mengelompokkan data yang ada ke dalam beberapa kelompok. Dimana data yang terdapat dalam suatu kelompok memiliki karakteristik yang sama antara satu dengan yang lain namun memiliki karakteristik yang berbeda dengan data yang berada pada kelompok yang lainnya, dengan begitu metode ini dapat digunakan untuk meminimalkan variasi antar data yang terdapat dalam suatu *cluster* serta memaksimalkan variasi dengan data-data yang terdapat dalam *cluster* yang lainnya [9].

2.7. Evaluasi dan Validasi Hasil

Validasi dilakukan dengan menggunakan 10 *fold cross validation*. Untuk 10 *fold cross validation* data eksperimen akan dibagi menjadi 10 bagian. Satu bagian untuk data testing Sembilan bagian lainnya untuk data training [10]. Sedangkan evaluasi pengukuran akurasi diukur dengan menggunakan *confusion matrix* yang juga mengukur hasil presisi dan recall [11]. Selain itu juga ditambahkan parameter waktu untuk mengetahui lama proses klasifikasi yang digunakan.

Tabel 2. *Confusion Matrix*

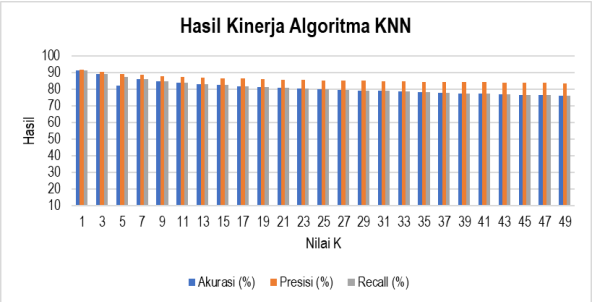
Aktual	Prediksi Kelas Positif	Kelas Negatif
Kelas Positif	TP (*True Positive)*	FN (*False Negative)*
Kelas Negatif	FP (*False Positive)*	TN (*True Negative)*

$$Akurasi = \frac{TP+TN}{TP+FN+FP+TN}\ x\ 100 \qquad (2)$$
$$Presisi = \frac{TP}{TP+FP}\ x\ 100$$
$$Recall = \frac{TP}{TP+FN}\ x\ 100$$

3. Hasil dan Pembahasan

Dalam penelitian ini nilai K pada KNN yang digunakan bernilai ganjil dari K=1 – K=49, hal ini bertujuan agar peneliti dapat mengetahui apakah hasil yang ada telah menunjukkan kestabilan data atau masih berubah-ubah [8]. Berdasarkan percobaan penelitian yang telah dilakukan diatas didapatkan nilai akurasi, nilai recall,

nilai precision, dan waktu pemrosesan dari klasifikasi spam atau ham pada email. Hasilnya adalah sebagaimana berikut yang ada pada Gambar 9:



Gambar 9. Hasil Kinerja Algoritma KNN

Dari hasil penelitian pada Gambar 9 dapat dilihat grafik nilai akurasi berhimpitan dengan nilai recall, hal tersebut menunjukkan bahwa nilai akurasi yang dihasilkan memiliki nilai yang mayoritas sama dengan nilai recall. Nilai presisi sebagaimana yang ditunjukkan pada grafik Gambar 9 memiliki nilai yang lebih besar dibandingkan nilai akurasi dan recall, sehingga hal tersebut menunjukkan bahwa tingkat ketepatan klasifikasi yang dilakukan diminta oleh user dengan jawaban yang diberikan oleh sistem sangat tepat. KNN dengan nilai parameter K=1 hingga K=49 dari hasil penelitian menunjukkan bahwa nilai K=1 merupakan nilai K yang paling optimal dengan tingkat akurasi sebesar 91.4% dengan nilai presisi sebesar 91.9% dan nilai recall sebesar 91.4%. Hasil persentase ini didapatkan dari proses percobaan sebanyak 10 kali dengan data yang diacak (10 *fold cross validation*).

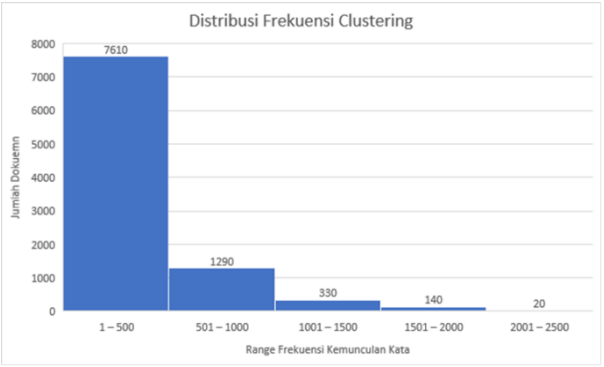
Untuk parameter waktu, dari hasil yang ada pada Gambar 9 dapat diketahui bahwa lama waktu pemrosesan yang dibutuhkan untuk mengklasifikasikan data spam email lebih cepat diproses saat menggunakan K=1 dan K=3 yang membutuhkan waktu 93 detik dibandingkan dengan nilai K yang lebih besar membutuhkan waktu lebih lama yaitu 95 sampai 180 detik. Sehingga dapat disimpulkan bahwa terdapat perbedaan yang signifikan antara nilai K yang digunakan dalam menghasilkan nilai evaluasi dari proses klasifikasi. Dalam metode KNN berdasarkan hasil klasifikasi yang ada berdasarkan grafik pada Gambar 9, semakin besar nilai K yang digunakan maka hasil akurasi, presisi, dan recall yang ada semakin menurun [6]. Jika semakin besar nilai K yang digunakan maka waktu yang dibutuhkan untuk pemrosesan juga dominan semakin lama. Sedangkan semakin kecil nilai K yang digunakan maka hasil yang ada semakin baik.

Optimasi parameter nilai K yang kecil pada KNN agar dapat menghasilkan nilai yang lebih besar dapat menggunakan metode distribusi frekuensi *clustering* untuk pembagian datanya. Dalam hal ini, dari proses pembobotan TF-IDF 9390 dokumen data yang ada didapatkan 1004 kata yang memiliki jumlah frekuensi kemunculan katanya sebanyak 17 sampai 2186 kali. Kemudian dari hal tersebut dibuat distribusi frekuensi *clustering* berdasarkan range frekuensi kemunculan kata tersebut pada setiap dokumen yang ada.

Tabel 3. Distribusi Frekuensi *Clustering*

Distribusi Frekuensi	Range Frekuensi Kemunculan Kata	Jumlah Dokumen
D1	1 – 500	7610
D2	501 – 1000	1290
D3	1001 – 1500	330
D4	1501 – 2000	140
D5	2001 – 2500	20
	Total	9390

Pada Tabel 3 dari hasil pembagian data menggunakan distribusi frekuensi yang ada maka terdapat 5 range frekuensi yang setiap intervalnya memiliki panjang kelas sebanyak 500. Berikut merupakan histogram hasil dari pembagian data berdasarkan frekuensi kemunculan kata yang ada dengan bantuan distribusi frekuensi *clustering* yang ada pada Gambar 10.



Gambar 10. Histogram Distribusi Frekuensi *Clustering*

Dari persebaran distribusi frekuensi yang ada, maka didapatkan 5 model pembagian data sebagaimana yang ada pada range frekuensi kemunculan data pada Tabel 3. Dari 5 model ini maka didapatkan 9 skenario penelitian yang akan digunakan untuk mengoptimalkan nilai K yang kecil (K=1) pada KNN agar nilai akurasi yang dihasilkan lebih baik yaitu :

1. Skenario 1 = D 1
2. Skenario 2 = D 2
3. Skenario 3 = D 3
4. Skenario 4 = D 4
5. Skenario 5 = D 5
6. Skenario 6 = D 1 dan D 2
7. Skenario 7 = D 1, D 2 dan D 3
8. Skenario 8 = D 1, D 2, D 3 dan D 4
9. Skenario 8 = D 1, D 2, D 3, D 4 dan D 5

Berikut merupakan hasil klasifikasi menggunakan KNN dengan nilai K paling kecil (K=1) yang ada dari skenario yang dihasilkan dari distribusi frekuensi *clustering* pada Tabel 4.

Dari Tabel 4 dapat diketahui jika nilai KNN dengan skenario 2 pada distribusi frekuensi clustering ini

memiliki akurasi yang paling tinggi sebesar 100% kemudian diikuti skenario 3 memiliki nilai akurasi 95.8%. Dalam hal ini menunjukkan bahwa nilai awal akurasi tertinggi yang dimiliki oleh KNN saat K=1 sebesar 91.4% dapat melebihi nilai akurasinya dengan optimasi nilai K metode KNN menggunakan distribusi frekuensi *clustering*.

Tabel 4. Hasil Klasifikasi Dengan Distribusi Frekuensi *Clustering*

Skenario	Evaluasi			
	Akurasi (%)	*Presisi (%)*	*Recall (%)*	*Waktu (s)*
1	89.4	90.1	89.4	71.27
2	100	100	100	2.29
3	95.8	96	95.8	0.75
4	95	95.3	95	0.50
5	90	90	90	0.49
6	90.6	91	90.6	63.17
7	91.1	91.7	91.1	61.54
8	91.4	91.8	91.2	64.03
9	91.4	91.9	91.4	104.99

Optimasi parameter nilai K yang kecil pada KNN agar dapat menghasilkan nilai yang lebih besar juga dapat menggunakan metode K-Means Clustering untuk pembagian datanya. Dalam hal ini, dari 9390 data dokumen yang ada kemudian dilakukan proses cluster menggunakan K-Means. Langkah-langkah untuk melakukan clustering dengan metode K-Means adalah [9]:

1. Memilih jumlah k cluster.
2. Inisialisasi k pusat cluster secara random. Pusat-pusat cluster (centroid) diberikan nilai awal secara random.
3. Melakukan alokasi semua data objek yang ada pada cluster terdekat, pada tahap ini penghitungan jarak tiap-tiap data ke centroid menggunakan *Euclidean Distance*.
4. Kemudian hitung kembali jarak antara centroid dengan data yang sekarang.
5. Mengulangi langkah 2-4 hingga nilai centroid tidak berubah.

Dalam penelitian ini untuk mengetahui performa kinerja yang ada pada K-Means pembagian k cluster yang ada akan dibagi ke dalam 3 model. Kemudian dari proses cluster tersebut didapatkan model yang tersedia pada Tabel 5.

Tabel 5. Hasil Pembagian Data K-Means *Clustering*

Model K-Means	Jumlah Data C_0	Jumlah Data C_1	Jumlah Data C_2	Jumlah Data C_3
1	2487	6903	-	-
2	1743	4186	3461	-
3	2175	6317	691	207

Setelah data terkluster menjadi 3 model, berikut merupakan skenario hasil klasifikasi yang didapat menggunakan KNN dengan nilai K paling kecil (K=1) :

a. Model 1 dengan nilai k sebanyak 2 cluster menggunakan 2 skenario
   1. Skenario C_0
   2. Skenario C_1

Tabel 6. Hasil Klasifikasi Model 1

Skenario	*Akurasi (%)*	*Presisi (%)*	*Recall (%)*	*Waktu (s)*
1	86.7	89.1	86.7	6.69
2	90.8	91.5	90.8	43.33

Dari Tabel 6 dapat diketahui jika nilai klasifikasi KNN dengan skenario 2 pada K-means clustering ini memiliki akurasi yang lebih tinggi jika dibandingkan dengan skenario 2 yaitu sebesar 90.8%. Dalam hal ini menunjukkan bahwa nilai yang ada pada k sebanyak 2 cluster pada K-means clustering ini masih belum bisa sebanding dengan yang dimiliki oleh KNN K=1 sebesar 91.4%.

b. Model 2 dengan nilai k sebanyak 3 cluster menggunakan 6 skenario
   1. Skenario C_0
   2. Skenario C_1
   3. Skenario C_2
   4. Skenario C_0 dan C_1
   5. Skenario C_0 dan C_2
   6. Skenario C_1 dan C_2

Tabel 7. Hasil Klasifikasi Model 2

Skenario	*Akurasi (%)*	*Presisi (%)*	*Recall (%)*	*Waktu (s)*
1	86.5	91.9	86.5	3.37
2	90.3	91.1	90.3	26.45
3	91.3	91.8	91.3	9.40
4	89.6	90.5	89.6	44.30
5	90.8	91.6	90.8	28.07
6	91.1	91.7	91.1	69.21

Dari Tabel 7 dapat dilihat bahwa nilai KNN dengan skenario 3 pada K-means clustering ini memiliki akurasi yang paling tinggi sebesar 91.3% jika dibandingkan dengan skenario yang lain. Dalam hal ini menunjukkan bahwa nilai awal akurasi tertinggi yang dimiliki oleh KNN K=1 sebesar 91.4% masih belum bisa ditandingi oleh nilai yang ada pada k sebanyak 3 cluster pada K-means clustering ini.

c. Model 3 dengan nilai k sebanyak 4 cluster menggunakan 14 skenario
   1. Skenario C_0
   2. Skenario C_1
   3. Skenario C_2
   4. Skenario C_3
   5. Skenario C_0 dan C_1
   6. Skenario C_0 dan C_2
   7. Skenario C_0 dan C_3
   8. Skenario C_1 dan C_2
   9. Skenario C_1 dan C_3
   10. Skenario C_2 dan C_3
   11. Skenario C_0, C_1 dan C_2
   12. Skenario C_0, C_1 dan C_3
   13. Skenario C_0, C_2 dan C_3
   14. Skenario C_1, C_2 dan C_3

Tabel 8. Hasil Klasifikasi Model 3

Skenario	*Akurasi (%)*	*Presisi (%)*	*Recall (%)*	*Waktu (s)*
1	86.2	89.7	86.2	4.36
2	89.3	90.3	89.3	33.78
3	99	99	99	1.08
4	93.7	94.1	93.7	0.76
5	89.9	90.5	89.9	86.88
6	90.8	91.6	90.8	20.07
7	87.4	89.8	87.4	6.89
8	90.6	91.3	90.6	46.26
9	89.6	90.7	89.6	41.76
10	98	98.1	98.1	1.53
11	91.3	91.8	91.3	99.51
12	90.4	91	90.4	68.25
13	90.4	91.2	90.4	8.63
14	90.8	91.5	90.8	50.45

Dari Tabel 8 dapat diketahui jika klasifikasi nilai KNN dengan skenario 3 pada K-means clustering ini memiliki akurasi yang paling tinggi diantara skenario lainnya yaitu sebesar 99% diikuti dengan skenario 10 dengan hasil akurasi sebesar 98%. Dalam hal ini menunjukkan bahwa nilai awal akurasi tertinggi yang dimiliki oleh KNN K=1 sebesar 91.4% dapat melebihi nilai akurasinya dengan optimasi nilai K metode KNN menggunakan K-means clustering dengan nilai k sebanyak 4 cluster.

4. Kesimpulan

Berdasarkan hasil penelitian didapatkan bahwa metode KNN dapat digunakan untuk klasifikasi email spam dan ham. Dalam pemrosesan dan hasil yang ada, nilai K=1 memiliki akurasi yang paling tinggi jika dibandingkan dengan nilai K yang lebih besar. KNN dengan nilai K=1 menghasilkan akurasi sebesar 91.4% dengan nilai presisi dan recallnya sebesar 91,9% dan 91.4% yang membutuhkan waktu selama 93 detik untuk prosesnya. Untuk lebih meningkatkan hasil akurasi yang ada maka dapat dilakukan cara mengoptimasi nilai K yang ada pada KNN. Optimasi dapat dilakukan dengan cara mengelompokkan data yang ada ke dalam beberapa kategori yang sejenis sebelum dimasukkan ke dalam proses klasifikasi. Metode optimasi yang dapat digunakan diantaranya adalah distribusi frekuensi clustering dan K-means clustering.

Dari hasil yang ada, pada optimasi menggunakan distribusi frekuensi clustering skenario ke-2 memiliki nilai akurasi yang tinggi sebesar 100% dan skenario ke-3 sebesar 95.8%, sedangkan pada optimasi menggunakan k-means clustering model 3 skenario ke-3 memiliki nilai akurasi yang tinggi sebesar 99% dan skenario ke-10 sebesar 98%. Untuk parameter waktu maka didapatkan hasil bahwa semakin banyak gabungan model yang ada dalam suatu skenario maka waktu pemrosesan yang dibutuhkan mayoritas akan semakin lama. Dari hasil perbandingan tersebut maka optimasi parameter nilai K pada algoritma KNN memiliki performa nilai akurasi yang lebih tinggi sehingga dapat digunakan dalam mengklasifikasikan email yang bersifat spam atau ham..

Daftar Rujukan

[1] P. Anugroho and I. Winarno, "Klasifikasi email spam dengan metode naïve bayes classifier menggunakan java programming," *ITS*, pp. 1–11, 2018.

[2] M. B. Hartanto, "Analisis dan Implementasi Pengklasifikasian Pesan Singkat pada Penyaringan SMS Spam Menggunakan Algoritma Multinomial Naïve Bayes," *e-Proceeding Eng.*, vol. 2, no. 2, pp. 6353–6357, 2015.

[3] V. Christanti *et al.*, "Perbandingan Pengklasifikasi K-Nearest Neighbor Dan Neighbor-Weighted K-Nearest Neighbor Pada Sistem Analisis Sentimen Dengan Data Microblog," *Front. J. Sains Dan Teknol.*, vol. 1, no. April, pp. 81–90, 2018, doi: 10.36412/frontiers/001035e1/april201801.08.

[4] J. Ling, I. P. E. N. Kencana, and T. B. Oka, "Analisis Sentimen Menggunakan Metode Naïve Bayes Classifier Dengan Seleksi Fitur Chi Square," *E-Jurnal Mat.*, vol. 3, no. 3, p. 92, 2014, doi: 10.24843/mtk.2014.v03.i03.p070.

[5] R. K. Roul, J. K. Sahoo, and K. Arora, "Modified TF-IDF Term Weighting Strategies for Text Categorization," *2017 14th IEEE India Counc. Int. Conf. INDICON 2017*, no. October, 2018, doi: 10.1109/INDICON.2017.8487593.

[6] M. Nanja and P. Purwanto, "Metode K-Nearest Neighbor Berbasis Forward Selection Untuk Prediksi Harga Komoditi Lada," *Pseudocode*, vol. 2, no. 1, pp. 53–64, 2015, doi: 10.33369/pseudocode.2.1.53-64.

[7] A. A. Irfa, Adiwijaya, and M. S. Mubarok, "Klasifikasi Topik Berita Berbahasa Indonesia Menggunakan k-Nearest Neighbor," *e-Proceeding Eng.*, vol. 5, no. 2, p. 3631, 2018.

[8] Indrayanti, D. Sugianti, and M. A. Al Karomi, "Optimasi Parameter K Pada Algoritma K-Nearest Neighbour Untuk Klasifikasi Penyakit Diabetes Mellitus," *Pros. SNATIF Ke-4 2017*, pp. 823–829, 2017, doi: 10.1007/s10115-007-0114-2.

[9] T. Widiyaningtyas, M. Prabowo, and M. Pratama, "Implementation of K-means clustering method to distribution of high school teachers," *EECSI*, pp. 1–6, 2017.

[10] Burhanudin, Y. Musa'adah, and Y. Wihardi, "Klasifikasi Komentar Spam Pada Youtube Menggunakan Metode Naïve Bayes, Support Vector Machine, dan K-Nearest Neighbors," *J. Inform. dan Komput.*, vol. 3, no. 2, pp. 54–59, 2018.

[11] D. Z. Nathania and F. A. Bachtiar, "Klasifikasi Spam Pada Twitter Menggunakan Metode Improved K-Nearest Neighbor," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, no. 10, pp. 3948–3956, 2018.