

Nisha Marthasari Wicaksono - Classification Neighbor Weighted K-Neares Neighbor

by Galih Wasis Wicaksono

Submission date: 01-Aug-2023 09:54PM (UTC+0700)

Submission ID: 2140008441

File name: caksono_-_Classification_Neighbor_Weighted_K-Neares_Neighbor.pdf (970.55K)

Word count: 3271

Character count: 18160

Klasifikasi Abstrak Jurnal Repositor di Teknik Informatika UMM Menggunakan Metode Neighbor Weighted K-Nearest Neighbor

Andjani Chaerun Nisha¹, Gita Indah Marthasari², Galih Wasis Wicaksono³

^{1,2,3}Universitas Muhammadiyah Malang
anjanie.nisa@gmail.com

Abstrak

Jurnal atau artikel ilmiah adalah sebuah data teks berupa karya tulis yang telah dibuat oleh peneliti yang melakukan sebuah penelitian pada suatu topik dalam bidang tertentu. Secara umum jurnal ilmiah ini diterbitkan online oleh situs penyedia informasi ilmiah dan perguruan tinggi. Pada penelitian ini mengambil studi kasus jurnal repositor Teknik Informatika UMM dimana masih belum dikategorikan berdasarkan jurusan bidang minat seperti RPL, Jaringan, Game, dan Data Science. Penelitian ini menggunakan metode Neighbor Weighted K-Nearest Neighbor (NWKNN) yaitu modifikasi dari metode KNN secara umum. Hasil klasifikasi menggunakan metode ini menunjukkan bahwa nilai k yang optimal adalah berada pada angka 17 dengan $e = 2$ memiliki akurasi sebesar 60,25641026%. Dikarenakan bahwa menurut penelitian sebelumnya yang menggunakan metode NWKNN, jika k bernilai besar maka suatu data uji akan mempunyai besar probabilitas benar masuk kedalam kategori kelas dengan hasil score tertinggi.

Kata Kunci: Klasifikasi, Neighbor Weighted K-Neares Neighbor

Abstract

Journals or scientific articles are text data in the form of written works that have been made by researchers who conduct research on a topic in a particular field. In general this scientific journal is published online by scientific information provider sites and universities. In this research, take a case study UMM Repositor Informatics Engineering journal which is still not categorized based on the fields of interest such as RPL, Network, Game, and Data Science. This study uses the Neighbor Weighted K-Nearest Neighbor (NWKNN) method, a modification of the KNN method in general. The classification results using this method show that the optimal k value is at number 17 with $e = 2$ which has 60,25641026% of accuracy. Due to that according to previous studies using the NWKNN method, if k is of great value then a test data will have a big probability of true entering the class category with the most highest score calculate.

Keywords: Classification, Neighbor Weighted K-Neares Neighbor

1. Pendahuluan

Pertumbuhan data teks dari tahun ke tahun semakin meningkat. Penyebabnya adalah tak lain dari berkembang pesatnya dunia teknologi informasi yang berisikan data teks. Perkembangan data teks untuk dijadikan sumber yang potensial saat ini telah menjamur di masyarakat seperti halnya media online yaitu website blog, portal berita, dan jejaring sosial [1]. Selain itu kebutuhan informasi dalam bentuk dokumen yaitu berupa jurnal ataupun artikel ilmiah telah meningkat setiap tahun. Kebutuhan jurnal ataupun artikel ilmiah meningkat ini juga dipengaruhi oleh bertambahnya kapasitas mahasiswa pertahunnya disetiap universitas serta peneliti yang anggotanya mahasiswa dan dosen [2].

Jurnal atau artikel ilmiah adalah sebuah data teks berupa karya tulis yang telah dibuat oleh peneliti yang melakukan sebuah penelitian pada suatu topik dalam bidang tertentu. Secara umum jurnal ilmiah ini diterbitkan online oleh situs penyedia informasi ilmiah dan perguruan tinggi. Dengan adanya perkembangan informasi yang cepat didukung oleh perkembangan internet saat ini memungkinkan proses penerbitan jurnal baik dari dalam dan luar negeri bisa dengan cepat dan mudah. Peneliti baik dari dalam dan luar negeri pun berlomba-lomba membuat jurnal ilmiah untuk keperluan studi mereka ataupun hanya sekedar meneliti saja. Jumlah dokumen jurnal ilmiah dalam satu situs penerbitan gratis bisa mencapai kurang lebih 25 juta dari seluruh dunia [3]. Tidak hanya dokumen dalam situs penerbitan gratis, penerbitan didalam perguruan tinggi pun tidak kalah jumlahnya yang kurang lebih 2000 file, dimana setiap jurusan dalam perguruan tinggi

menyetorkan jurnal ilmiah dengan jumlah 200 lebih *file*. Oleh karena itu dibutuhkan pengelompokan jurnal agar lebih mudah dalam mencari informasi

Pada penelitian ini bertujuan untuk menerapkan metode *Neighbor Weighted K-Nearest Neighbor* untuk melakukan proses klasifikasi ke dalam kategori yang tepat berdasarkan bidang minat pada aplikasi yang akan dibuat, sehingga mahasiswa dan masyarakat umum yang membutuhkan referensi berupa jurnal dapat mencarinya sesuai dengan bidang minat yang akan dicari. Pengaksesan repositor jurnal untuk pengambilan data pada saat ini tidak bisa diakses dari luar dikarenakan *server* sedang mengalami masalah maka untuk memperoleh data repositor jurnal pada prodi Teknik Informatika UMM harus meminta data pada dosen yang terkait dengan penerbitan jurnal. Pemilihan Klasifikasi Abstrak Jurnal Repositor di Teknik Informatika UMM Menggunakan Metode *Neighbor Weighted K-Nearest Neighbor* berdasarkan beberapa penelitian sebelumnya yang menggunakan metode tersebut dengan tingkat akurasi yang baik dan efektif.

Pada penelitian sebelumnya, penulis belum bisa menyelesaikan masalah yaitu belum dapat mengklasifikasikan jurnal ke kategori yang tepat. Untuk menyelesaikan permasalahan tersebut, pada penelitian ini proses mengkategorikan jurnal dengan tepat dapat diselesaikan dengan menggunakan metode *Neighbor Weighted K-Nearest Neighbor*. Metode NWKNN adalah metode modifikasi atau pengembangan dari metode tradisional KNN. Metode ini menghasilkan nilai bobot kecil untuk tetangga dari kelas mayoritas dan bobot besar untuk tetangga dari kelas minoritas [4]. Berdasarkan penelitian lain sebelumnya menggunakan metode *Neighbor Weighted K-Nearest Neighbor* diperoleh hasil kemajuan signifikan pada penyebaran data tidak seimbang yang terbukti bahwa perbandingan kinerja metode *Neighbor Weighted K-Nearest Neighbor* lebih baik daripada kinerja metode *K-Nearest Neighbor* tradisional [5].

Kategori bidang minat yang ada saat ini berdasarkan kurikulum baru yaitu Rekayasa Perangkat Lunak (RPL), Sistem Keamanan Jaringan (SKJ), *Game Cerdas*, dan *Data Science*. Dengan adanya aplikasi klasifikasi jurnal repositor menggunakan metode *K-Nearest Neighbor* ini bisa menjadi solusi masalah pada jurnal sebelumnya yang menggunakan dua metode yaitu *Naive Bayes Classifier* dan *Support Vector Machine*.

2.3 Metode Penelitian

2.1 Metode Pengumpulan Data

Metode ini dilakukan untuk mendapatkan informasi yang digunakan untuk tercapainya tujuan penelitian. Terdapat dua metode pengumpulan data yang digunakan yaitu : Studi Literatur dan Observasi.

2.2 Studi Literatur

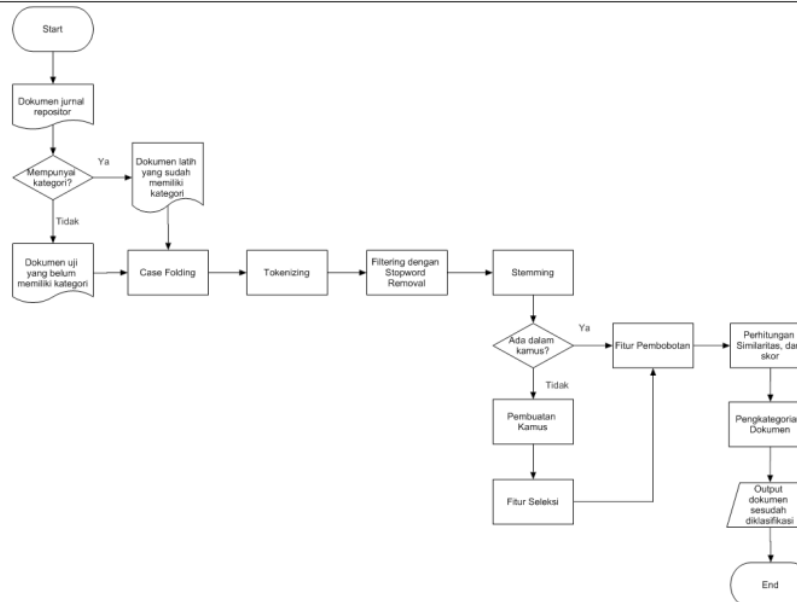
Studi literatur merupakan pencarian dan pengumpulan informasi bersumber dari jurnal ilmiah, buku ataupun artikel yang membahas terkait dengan penelitian. Studi literatur yang digunakan untuk penelitian ini adalah jurnal klasifikasi, penjelasan metode NWKNN, dan referensi terkait lainnya. Jurnal yang digunakan adalah yang membahas tentang pengertian metode NWKNN dan perbedaan metode NWKNN dengan KNN ditinjau dari perbandingan nilai performansi keduanya.

2.3 Observasi

Observasi merupakan metode pengumpulan data yang langsung melihat situasi secara nyata untuk mendapat pengetahuan apa saja terkait dengan penelitian. Observasi dilakukan di kantor dosen Teknik Informatika UMM yaitu dengan meminta data jurnal pada dosen yang telah menjadi admin jurnal repositor Teknik Informatika UMM.

2.4 Analisa Kebutuhan

Berdasarkan tujuan dari *tools* yang akan dibuat maka perlu menyediakan kebutuhan. Gambar 1 berikut penjelasan dari alur *tools* yang akan dibuat.



Gambar 1. Alur Tools Klasifikasi Repositor Jurnal Teknik Informatika UMM

2.5 Proses Klasifikasi

Proses ini dimulai dengan membuat *training* data dan data *test* dan mengekstrasi dokumen jurnal repositor yang kemudian ditambahkan ke dalam *tools* bantuan. Terdapat 6 proses dalam proses klasifikasi menggunakan metode *Neighbor Weighted K-Nearest Neighbor*.

2.5.1 Ekstraksi Data

Pada Tabel 1 ini dilakukan pembacaan dan penerjemahan *input dataset* abstrak jurnal repositor prodi Teknik Informatika menjadi sebuah objek yang mudah dipahami oleh sistem. Abstrak jurnal diekstrak menggunakan *tools* yang kemudian hasilnya akan menentukan banyaknya jumlah data *training* dan data *test*.

Tabel 1. Sample Data Abstrak Jurnal

Jenis Data	Abstrak	Kategori
Data Latih	Laptop memiliki peran penting dalam pekerjaan sehari-hari...	Jaringan
Data Latih	Pada penelitian ini, dilakukan klasifikasi dokumen Tugas Akhir ...	Data Science
Data Uji	Algoritma Metaphone dapat diterapkan pada berbagai ...	Data Science
Data Uji	Penulis mengangkat topik penelitian guna mengetahui ...	Jaringan

2.5.2 Text Pre-Processing

Pada Tabel 2 ini dilakukan proses *tokenizing*, *case folding*, *filtering*, dan *stemming*. Tujuan dilakukan beberapa proses tersebut adalah untuk memilah-milah teks yang ada pada abstrak jurnal menjadi beberapa *term* agar siap diolah pada tahap selanjutnya. Pada proses *tokenizing* dilakukan penghilangan karakter-karakter pemisah kata seperti spasi, tanda baca, angka. Pada proses *case folding* dilakukan perubahan huruf besar menjadi huruf kecil. Pada proses *stopword removal* dilakukan penghapusan kata yang merupakan ciri atau kata unik dalam suatu dokumen. Pada proses *stemming* dilakukan perubahan kata menjadi kata dasar yang sesuai dengan ejaan EYD [6].

Tabel 2. Proses Tokenizing, Stopword Removal, Stemming

Jenis Data	Abstrak	Kategori
Data Latih	[laptop, peran, kerja, ri, forma, laptop, kondisi, laptop, rusak, ...]	Jaringan
Data Latih	[teliti, klasifikasi, dokumen, tugas, teknik, informatika, masalah, ...]	Data Science
Data Uji	[terap, bahasa, atur, modifikasi, bahasa, dasar, sil, uji, ...]	Data Science
Data Uji	[tulis, angkat, topik, akurasi, baca, sensor, gas, lpg, pasar, ...]	Jaringan

2.5.3 Term Weighting

Pada Tabel 3 ini dilakukan perhitungan pembobotan dengan metode pembobotan TF-IDF. TF adalah jumlah kemunculan kata pada satu dokumen. Sedangkan IDF adalah pembobotan kata berdasarkan banyaknya dokumen yang memuat kata tertentu.

Tabel 3. Penerapan Pembobotan Menggunakan TF-IDF

Kata	Tf					idf (log(n/df))	Wdt = tf.idf			
	DL1	DL2	DU1	DU2	df		DL1	DL2	DU1	DU2
laptop	3	0	0	0	3	0,125	0,375	0,000	0,000	0,000
peran	1	0	0	0	1	0,602	0,602	0,000	0,000	0,000
kerja	1	0	0	0	1	0,602	0,602	0,000	0,000	0,000
ri	1	0	0	0	1	0,602	0,602	0,000	0,000	0,000
forma	1	0	0	0	1	0,602	0,602	0,000	0,000	0,000
.....

2.5.4 Perhitungan Euclidean Distance

Pada Tabel 4 ini dilakukan perhitungan ketetanggaan terdekat dengan memasukkan nilai kemiripan antara dokumen data latih dan data uji setelah dilakukan pembobotan menggunakan TF-IDF. Tahap berikutnya perhitungan nilai ketetanggaan menggunakan *Euclidean Distance*, hasil dari perhitungan tersebut dijumlah berdasarkan dokumen yang telah diuji.

Tabel 4. Perhitungan Jarak Data Uji terhadap Data Latih

	DU1		DU2	
	DL1	DL2	DL1	DL2
	0,375	0	0,375	0
	0,602	0	0,602	0
	0,602	0	0,602	0
	0,602	0	0,602	0
	0,602	0	0,602	0

Setelah melewati tahap perhitungan nilai ketetanggaan menggunakan *Euclidean Distance*, hasil dari perhitungan tersebut dijumlah berdasarkan dokumen yang telah diuji, seperti pada Tabel 5.

Tabel 5. Total Jarak Data Uji Terhadap Data Latih

	DU1		DU2	
	DL1	DL2	DL1	DL2
SUM	22,672	23,07	26,636	26,08
Sorting	22,672	23,07	26,08	26,636

2.5.5 Pembobotan Kelas Kategori

Pada Persamaan 1 ini dilakukan pembobotan sebelum dilakukan perhitungan skor. Tahap ini merupakan letak perbedaan antara metode KNN dan NWKNN sehingga diharapkan dapat melakukan pengkategorian secara tepat.

$$Weight(i) = \frac{1}{\left(\frac{\text{jumlah data latih } i}{\text{jumlah data minimum pada } k}\right)^{\frac{1}{exp}}} = \text{hasil}$$

$$Weight(\text{Jaringan}) = \frac{1}{\left(\frac{1}{2}\right)^{\frac{1}{2}}} = 1,414213562 \quad (1)$$

$$Weight(\text{Data Science}) = \frac{1}{\left(\frac{1}{2}\right)^{\frac{1}{2}}} = 1,414213562$$

2.5.6 Nilai Skor

Pada Tabel 6 menunjukkan perhitungan nilai skor dengan memasukkan nilai *Euclidian Distance* dan bobot kelas pada proses sebelumnya. DL1 termasuk kategori jaringan sehingga termasuk elemen *score* kategori yang akan diuji, maka hasil perhitungan nilai jarak / *Euclidian Distance* DL1 dikali dengan 1. Sedangkan untuk DL2 termasuk kategori *Data Science* sehingga tidak termasuk elemen *score* jaringan, bernilai 0. Berikut perhitungannya :

- *Score* Pengujian DU1
 $Score\ DL1(\text{Jaringan}) = 1,414213562 * ((22,672 * 1) + (23,07 * 0)) = 32,06304989$
 $Score\ DL2(\text{Data Science}) = 1,414213562 * ((22,672 * 0) + (23,07 * 1)) = 32,62590688$
 $Score\ DL2$ lebih besar daripada $score\ DL1$, maka DU1 terklasifikasi sebagai dokumen dengan kelas kategori *Data Science*.
- *Score* Pengujian DU2
 $Score\ DL1(\text{Jaringan}) = 1,414213562 * ((26,36 * 1) + (26,08 * 0)) = 37,66899245$
 $Score\ DL2(\text{Data Science}) = 1,414213562 * ((26,36 * 0) + (26,08 * 1)) = 36,88268971$
 $Score\ DL1$ lebih besar daripada $score\ DL2$, maka DU2 terklasifikasi sebagai dokumen dengan kelas kategori Jaringan.

Tabel 6. Kelas Prediksi Dokumen Uji

Dokumen Uji	Kelas Awal	Kelas Prediksi
DU1	<i>Data Science</i>	<i>Data Science</i>
DU2	Jaringan	Jaringan

2.5.7 Pengukuran Akurasi

Pada penelitian ini untuk pengukuran akurasinya menggunakan Persamaan 2 sebagai berikut.

$$Akurasi = \frac{\text{jumlah data yang diprediksi benar}}{\text{banyak data dalam satu set data testing}} \times 100\% \quad (1)$$

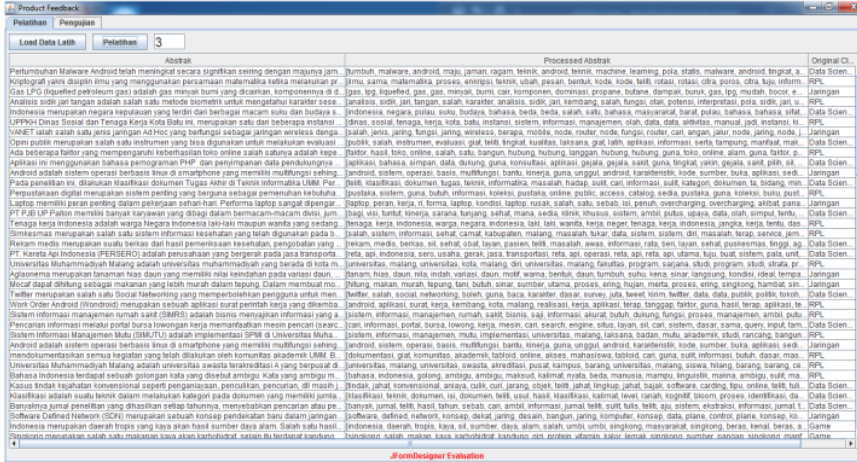
$$Akurasi = \frac{2}{2} \times 100\% = 100\%$$

3. Hasil Penelitian dan Pembahasan

Pada penelitian ini *dataset* diambil dari jurnal repositor Teknik Informatika Universitas Muhammadiyah Malang yang belum [posting](#) pada halaman *website* jurnal repositor. Data yang diambil berjumlah 130 *file* kemudian data tersebut dibagi menjadi 80% *data latih* dan 20 % uji dan divalidasi oleh 3 pakar yaitu dosen Informatika Universitas Muhammadiyah Malang sesuai bidang minat.

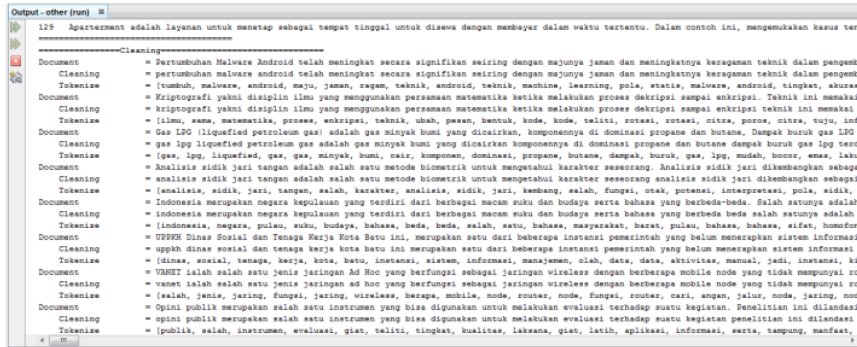
3.1 Tools

Gambar 2 berikut merupakan *tools* yang telah dibuat menggunakan *java*. Pada tahap pertama yaitu *preprocessing* dokumen.



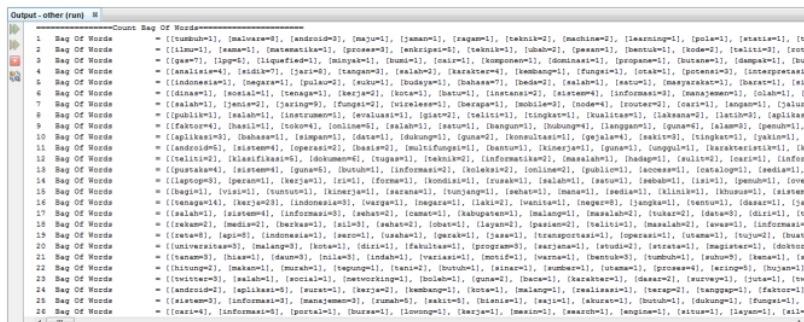
Gambar 2. Interface Preprocessing Data.

Pada Gambar 3 saat tombol "Pelatihan" di klik, setelah klik tombol "Load Data Latih", pada console Netbeans, akan muncul proses *tokenizing* dan *cleaning*. Pada proses ini, kalimat pada dokumen akan dirubah menjadi dokumen, kemudian dilakukan proses pembersihan dengan *stopword removal* dan *stemming*.



Gambar 3. Tokenizing dan Cleaning Document

Kalimat yang sudah dilakukan proses pembersihan akan masuk ke dalam *Bag Of Words*, yaitu tempat frekuensi *terms* pada suatu dokumen, seperti pada Gambar 4.



Gambar 4. Bag of Words dari Setiap Term didalam Document

Gambar 5 menunjukkan hasil setelah *terms* masuk kedalam *Bag Of Words*, masing-masing term yang ada didalam dokumen akan dibobot menggunakan TF-IDF.

```

Output - other (run)
Calculate TF-IDF
1 TF-IDF = [tubuh=1,414973], [malware=3,094977], [android=2,650493], [saja=1,395792], [saman=2,119943], [sagan=1,812313], [teknik=2,000000], [
2 TF-IDF = [lilam=1,511893], [sana=1,511893], [doktermalika=2,636222], [pilot=1,471022], [teknisi=1,284113], [teknisi=1,000000], [sabah=2,187492],
3 TF-IDF = [saga=14,797493], [lpg=10,649717], [lilique=1=2,119943], [malika=2,119943], [dusa=1,636222], [caki=1,812313], [komponen=1,414973], [do
4 TF-IDF = [malika=1,009492], [id=1=14,797493], [sai=16,815477], [tangan=1,652640], [sial=0,812744], [karakter=6,647288], [kemungkinan=4,22482
5 TF-IDF = [lindonesia=1,741724], [sagan=1,511893], [pilot=1=4,425627], [saja=1,812313], [doktermalika=1,511893], [doktermalika=4,636222], [dokter=1,425627], [a
6 TF-IDF = [dina=1,636222], [sial=1=1,210843], [sagan=1,636222], [dokter=1,819477], [kota=1,072551], [kota=1,812313], [satan=1,523747], [sai
7 TF-IDF = [sial=0,408373], [sial=1,633442], [saling=1,710339], [tangan=1,760939], [doktermalika=1,636222], [sagan=1,119943], [sial=1=4,244920], [
8 TF-IDF = [sial=1=1,395792], [sial=0,408373], [sial=1=1,119943], [doktermalika=1,034742], [sai=1=1,210843], [sial=0,219439], [saling=1,812313
9 TF-IDF = [karakter=1,071261], [sial=1=0,882444], [sok=0,047533], [sial=1=4,838077], [sial=0,408373], [sial=0,303923], [sagan=1,716003], [tubuh
10 TF-IDF = [aplikasi=1,244920], [doktermalika=0,826130], [sagan=1,210843], [data=1,321522], [doktermalika=1,597011], [sagan=1,664044], [komponen=1,119943]
11 TF-IDF = [android=1,417472], [sial=1=1,432744], [sagan=1=1,412707], [doktermalika=1,034742], [sial=1=1,034742], [sial=0,303923], [sagan=1,039701], [sial=1=1,000
12 TF-IDF = [sial=1=0,887859], [sial=1=1,812313], [tangan=1,119943], [sial=1=1,000000], [sagan=1,034742], [teknik=1=2,000000], [informasi=1=0,023747], [malika=
13 TF-IDF = [aplikasi=1,047938], [sial=1=1,602744], [sagan=1,169811], [doktermalika=0,811893], [informasi=1,812744], [teknik=1=3,425627], [komponen=1,119943]
14 TF-IDF = [lilique=1,041892], [sagan=1,159702], [sai=1=0,909823], [sai=1,812313], [sagan=1,414973], [doktermalika=1,210843], [sagan=1,159702], [sial=1=
15 TF-IDF = [sagan=1,047938], [sial=1=1,812313], [tangan=1,119943], [sial=1=1,000000], [sagan=1,811893], [sagan=1,119943], [sial=1,119943], [s
16 TF-IDF = [sagan=2,815509], [sai=1=0,909823], [lindonesia=1,075172], [sagan=1,119943], [sagan=1,511893], [sai=1=4,227897], [sial=1=1,812313], [s
17 TF-IDF = [sial=1=1,402374], [sial=1=1,402374], [lindonesia=1,210843], [sial=1=2,278977], [sagan=1,414973], [saling=1,268441], [saling=1,812313
18 TF-IDF = [sagan=1,023747], [sial=1=1,023747], [doktermalika=1,812313], [sai=1=0,760322], [sial=1=2,278977], [sagan=1,511893], [sai=1=1,447444], [sagan=1
19 TF-IDF = [sai=1=4,023307], [sai=1=1,197877], [lindonesia=1=0,781724], [sai=1=1,425627], [sagan=1,828494], [sagan=1,812313], [sagan=1,414973], [sagan
20 TF-IDF = [lindonesia=1=0,478103], [saling=1,438742], [sai=1=0,072821], [sai=1=1,811893], [sial=1=1,119943], [sagan=1,478103], [sagan=1,119
21 TF-IDF = [sagan=1,938450], [sagan=1,119943], [sagan=1,438742], [sai=1=1,656341], [lindonesia=1,636222], [sial=1=1,811893], [sial=1=2,119943], [sagan=1
22 TF-IDF = [saling=1,717942], [sagan=1,414973], [sagan=1,119943], [sagan=1,395792], [sai=1=4,227897], [tubuh=1,511893], [sagan=1,812313], [sagan
23 TF-IDF = [sial=1=1,010466], [sial=1=1,636222], [sial=1=1,119943], [sial=1=1,119943], [sagan=1,868442], [sagan=1,000000], [
24 TF-IDF = [android=1,746989], [aplikasi=1=0,748477], [sagan=1,636222], [sai=1=1,819477], [kemungkinan=1,622982], [sai=1=0,720851], [saling=1,812313], [
25 TF-IDF = [sial=1=0,742042], [informasi=1,219120], [sagan=1=1,636222], [sagan=1,344227], [sai=1=6,649717], [sial=1=1,414973], [sai=1=1,414973], [s
26 TF-IDF = [sai=1=1,449341], [informasi=1,031841], [sagan=1=1,119943], [sagan=1,119943], [sagan=1,119943], [sai=1=0,909823], [sagan=1,414973], [s

```

Gambar 5. Perhitungan TF-IDF

TF-IDF yang sudah terhitung dimasukkan ke *file excel* yang kemudian akan menjadi data latih dan data uji pada proses NWKNN. Data latih berupa *excel* yang sudah ada kemudian di *load* untuk diproses, seperti pada Gambar 6

```

Output - Skripsi (run)
run
*****Data Ke- 0*****
Jarak Kelas Original
0.696 4.0
0.463 1.0
0.667 2.0
0.847 1.0
0.391 4.0
0.489 1.0
0.401 2.0
0.347 4.0
0.551 1.0
0.314 4.0
0.401 2.0
0.533 4.0
0.521 1.0
0.425 2.0
0.392 4.0
1.144 1.0
0.632 1.0
0.384 4.0
0.698 4.0
0.344 1.0
0.824 2.0
0.447 2.0
0.785 4.0
0.499 1.0
0.49 1.0
0.381 4.0
0.419 1.0
0.481 2.0
0.348 1.0
0.398 1.0
0.576 1.0
0.61 4.0
0.466 4.0

```

Gambar 6. Load Data Latih Proses NWKNN

3.2 Pengujian Sistem

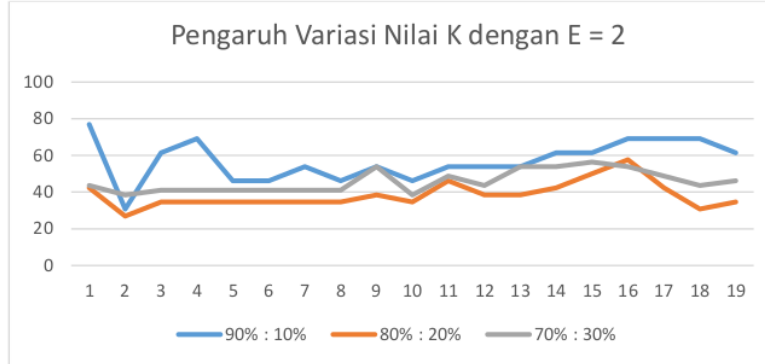
3.2.1 Pengaruh Variasi Nilai K

Pada pengujian sistem ini dilakukan perbandingan performa nilai k dari 20 variasi nilai k yaitu 1 – 20 dengan nilai e yaitu 2 dan 4 . Tabel 7, Tabel 8, Gambar 7, dan Gambar 8 berikut merupakan hasil akurasi.

Tabel 7. Pengaruh Variasi Nilai k dengan Nilai e = 2

Nilai e	Nilai k	Akurasi Perbandingan Data (%)		
		90% : 10%	80% : 20%	70% : 30%
2	2	76,92308	42,30769	43,58974
	3	30,76923	26,92308	38,46154

	20	61,53846	34,61538	46,15385

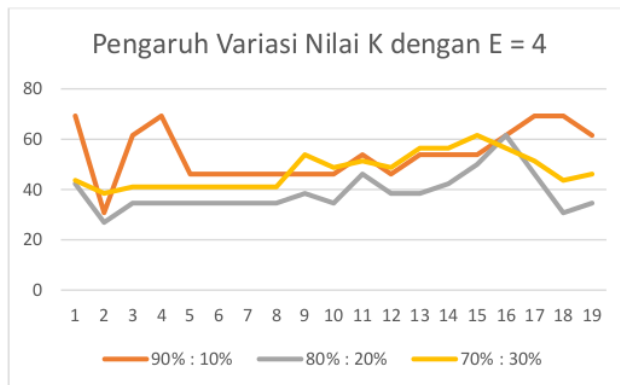


Gambar 7. Grafik Perubahan Nilai k dengan Nilai $e = 2$

Tabel 8. Pengaruh Variasi Nilai k dengan Nilai $e = 4$

Akurasi Perbandingan Data (%)				
Nilai E	Nilai k	90% : 10%	80% : 20%	70% : 30%
4	2	69,23077	42,30769	43,58974
	3	30,76923	26,92308	38,46154

	20	61,53846	34,61538	46,15385



Gambar 8. Grafik Perubahan Nilai k dengan Nilai $e = 4$

Berdasarkan hasil skenario pengujian pada variasi nilai k 1-20 dengan $e = 2$ dan $e = 4$, nilai akurasi cenderung naik turun disetiap rentang jumlah data. Hal ini menunjukkan bahwa perubahan nilai K pada rentang data latih jika semakin besar nilainya maka jumlah data yang mendominasi masuk kedalam ketetangaan yang telah ditentukan. Untuk mendapatkan nilai k optimal, masing-masing nilai k pada rentang data di rata-rata.

Tabel 9. Rata-rata Nilai k pada $e = 2$ dan $e = 4$

Nilai k	Rata-rata Nilai k	
	$e = 2$	$e = 4$
2	54,2735	51,7094
3	32,05128	32,05128
...
20	47,4359	47,4359
Max (17)	60,25641	59,82906

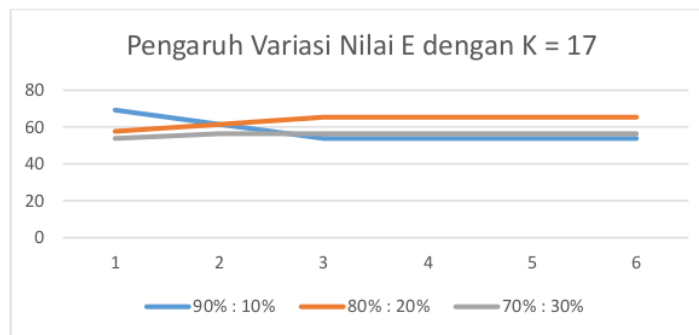
Hasil Tabel 9 dari perhitungan rata-rata ditemukan bahwa nilai k optimal berada pada $e = 2$, dan $e = 4$ dengan nilai $k = 17$.

3.2.2. Pengaruh Variasi Nilai E

Pada pengujian sistem ini dilakukan perbandingan performa nilai $k = 17$ dengan variasi nilai e yaitu 2, 4, 6, 8, 10, 12. Tabel 10 dan Gambar 9 berikut merupakan hasil akurasi nya :

Tabel 10. Nilai Bobot Setiap Kategori Kelas Berdasarkan Nilai e Akurasi Perbandingan Data (%)

Nilai k	Nilai e	90% : 10%	80% : 20%	70% : 30%
17	2	69,23077	57,69231	53,84615
	4	61,53846	61,53846	56,41026
	6	53,84615	65,38462	56,41026
	8	53,84615	65,38462	56,41026
	10	53,84615	65,38462	56,41026
	12	53,84615	65,38462	56,41026



Gambar 9. Grafik Perubahan Nilai e dengan Nilai $k = 17$

Berdasarkan Tabel 11 hasil skenario pengujian pada variasi nilai $e = 2, 4, 6, 8, 10, 12$ dengan $k = 17$, nilai akurasi cenderung stabil disetiap rentang jumlah data. Hal ini menunjukkan bahwa perubahan nilai e pada rentang data latih tidak memiliki pengaruh perubahan akurasi yang besar. Untuk mendapatkan nilai e optimal, masing-masing nilai e pada rentang data di rata-rata.

Tabel 11. Rata-rata Nilai e Pada $k = 17$

Nilai k	Nilai e	AVG
17	2	60,25641
	4	59,82906
	6	58,54701
	8	58,54701
	10	58,54701
	12	58,54701
Max (2)	60,25641	

Hasil dari perhitungan rata-rata ditemukan bahwa nilai e optimal pada $k = 17$ dengan variasi nilai e yang telah ditentukan yaitu $e = 2$. Maka dapat disimpulkan bahwa klasifikasi abstrak jurnal repositor Teknik Informatika UMM menggunakan algoritma NWKNN diperoleh nilai optimal $k = 17$ dan $e = 2$.

4. Kesimpulan

Berdasarkan hasil penelitian menggunakan metode *Neighbor Weighted K-Nearest Neighbor* (NWKNN), ada beberapa tahapan yang telah dilakukan. Tahap pertama adalah preprocessing abstrak yang dilakukan ada beberapa tahap meliputi *tokenizing*, *case folding*, *filtering*, dan *stemming*. Kemudian dilakukan *term weighting* menggunakan TF-IDF yang dimana

hasil dari proses tersebut digunakan untuk menghitung jarak ketetangga. Proses NWKNN diawali dengan tahap perhitungan jarak ketetangga menggunakan *Euclidean Distance*. Hasil dari jarak tersebut dilakukan pengurutan / *sorting* guna memudahkan perhitungan. Setelah itu masuk ke tahap pembobotan kategori pada data latih dan dari hasil pembobotan dilakukan perhitungan *score* untuk menentukan kelas prediksi. Tahap terakhir adalah menghitung tingkat keakurasian dimana membagi jumlah data benar sesuai dengan diprediksi dengan jumlah total data latih. Sehingga didapatkan nilai optimal $k = 17$ dan $e = 2$ pada setiap rentang data latih dengan rata-rata akurasi sebesar 60,25641026%.

Daftar Notasi

k : jumlah data iterasi
 e / exp : *exponent*
Weight(i) : Bobot Per Kategori.

Referensi

- [1] A. S. Sri Widaningsi, "Klasifikasi Jurnal Ilmu Komputer Berdasarkan Pembagian," *Semin. Nas. Teknol. Inf. dan Komun. 2018 (SENTIKA 2018)*, vol. 2018, no. Sentika, pp. 23–24, 2018.
- [2] A. Indranandita, "Sistem Klasifikasi Dan Pencarian Jurnal Dengan Menggunakan Metode Naive Bayes Dan Vector SP Ace Model," *J. Inform.*, vol. 4, no. 2, p. 10, 2008, doi: 10.21460/inf.
- [3] A. Fathan Hidayatullah, M. Rifqi Ma, and A. Program Studi Manajemen Informatika STMIK Jenderal Achmad Yani Yogyakarta Jl Ringroad Barat, "Penerapan Text Mining dalam Klasifikasi Judul Skripsi," *Semin. Nas. Apl. Teknol. Inf. Agustus*, pp. 1907–5022, 2016.
- [4] M. T. F. Annisya Aprilia Prasanti, M. Ali Fauzi, "Klasifikasi Teks Pengaduan Pada Sambat Online Menggunakan Metode N-Gram dan Neighbor Weighted K-Nearest Neighbor (NW-KNN)," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, no. 2, pp. 594–601, 2018.
- [5] S. Tan, "Neighbor-weighted K-nearest neighbor for unbalanced text corpus," *Expert Syst. Appl.*, vol. 28, no. 4, pp. 667–671, 2005, doi: 10.1016/j.eswa.2004.12.023.
- [6] A. Ridok, "Sentiment Analysis for Review Mobile Applications Using Neighbor Method Weighted K-Nearest Neighbor (Nwknn)," *J. Environ. Eng. Sustain. Technol. 2016*, vol. 03, no. 01, pp. 23–32, 2016.

Nisha Marthasari Wicaksono - Classification Neighbor Weighted K-Neares Neighbor

ORIGINALITY REPORT

5%

SIMILARITY INDEX

4%

INTERNET SOURCES

1%

PUBLICATIONS

2%

STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Universitas Terbuka Student Paper	2%
2	zombiedoc.com Internet Source	<1%
3	repository.uinjkt.ac.id Internet Source	<1%
4	www.coursehero.com Internet Source	<1%
5	Evan Kriminger, Jose C. Principe, Choudur Lakshminarayan. "Nearest Neighbor Distributions for imbalanced classification", The 2012 International Joint Conference on Neural Networks (IJCNN), 2012 Publication	<1%
6	eprints.umg.ac.id Internet Source	<1%
7	id.123dok.com Internet Source	<1%

8

kinetik.umm.ac.id

Internet Source

<1 %

9

repositor.umm.ac.id

Internet Source

<1 %

10

www.journaltoocs.ac.uk

Internet Source

<1 %

11

Tulus Harry Lamramot Tulus, Asep Id
Hadiana, Irma Santikarama. "Sistem Prediksi
Awal Terhadap Atrisi Karyawan Menggunakan
Algoritma C4.5", Informatics and Digital
Expert (INDEX), 2022

Publication

<1 %

Exclude quotes Off

Exclude matches Off

Exclude bibliography On