Klasifikasi Jenis Laporan Masyarakat dengan K-Nearest Neighbor Algorithm

Heru Pramono Hadi[1], Titien S. Sukamto[2]
Program Studi Sistem Informasi, Universitas Dian Nuswantoro
Jl Imam Bonjol 207 Semarang
e-mail: [1]heru.pramono.hadi@dsn.dinus.ac.id , [2]titien.suhartini@dsn.dinus.ac.id

Diterima: 11 Februari 2020; Direvisi: 30 April 2020; Disetujui: 6 Mei 2020

Abstrak

Feedback masyarakat terhadap pelayanan pemerintah merupakan elemen penting dalam proses evaluasi dan peningkatan kinerja. Maka dari itu pemerintah perlu untuk memiliki metode pelaporan yang efektif, efisien dan sistematis. Feedback masyarakat dapat berupa pengaduan, permintaan informasi dan aspirasi. Salah satu cara penyampain feedback masyarakat adalah melalui media sosial. Klasifikasi jenis laporan/feedback masyarakat ini penting dilakukan untuk mempercepat proses penanggapan laporan. Algoritma K-Nearest neighbor pada metode text mining ini merupakan salah satu solusi untuk dapat membantu proses klasifikasi jenis laporan. Dengan 930 data latih dan 100 data uji laporan masyarakat tahun 2017 yang disampaikan melalui media sosial, menghasilkan nilai akurasi tertinggi k=11 sebesar 82%.

Kata kunci: *klasifikasi teks, text mining, k-nearest neighbor*

Abstract

Feedbak from the citizen on government services is an important element in the process of evaluating and improving their performances. Therefore, the government needs to have an effective, efficient, and systematic method of reporting. Feedback can be in the form of complaints, requests for information, and aspirations. One way to get public feedback is through social media. In addition, it is important to classify the types of feedback, in order to speed up the report response process. K-Nearest Neighbor algorithm is a type of algorithm in text mining, which can be used to classify reports in text form. With 930 data set and 100 data test on 2017 public feedback submitted via social media, results with highest accuracy value was k=11 by 82%.

Keywords: *text classification, text mining, k-nearest neighbor*

1. PENDAHULUAN

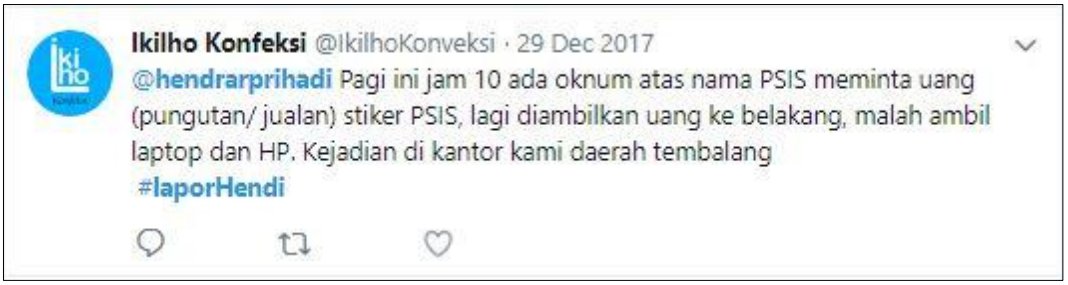
Penggunaan media sosial yang semakin meluas dapat dimanfaatkan sebagai sarana untuk menyampaikan tanggapan berupa pendapat maupun keluhan terhadap kinerja serta pelayanan publik. Kota Semarang juga telah memanfaatkan hal ini. Masyarakat Kota Semarang diberikan berbagai *platform* yang dapat digunakan untuk menyampaikan keluhan, pendapat, ide aspiratif, maupun permintaan informasi. Salah satu media yang dapat digunakan adalah media sosial, seperti *twitter*. Laporan masyarakat ini dikelola oleh Bidang Pusat Pengelolaan Pengaduan Masyarakat (P3M) yang tugasnya adalah menerima, mengelola, mengklarifikasi, monitoring, mengevaluasi serta mengkonfirmasi pengaduan masyarakat. Masyarakat dapat menyampaikan laporannya dengan menggunakan *#LaporHendi*. Beragamnya laporan yang masuk, membuat

perlunya ketelitian dan kecekatan dari P3M dalam mengkalisifikasi/ mengkategorikan jenis laporan. Tujuannya adalah agar laporan dapat segara mendapat respon yang sesuai. Pemerintah Kota Semarang membagi jenis laporan ke dalam 3 (tiga) kategori yaitu, 1) pengaduan, 2) permintaan informasi, dan 3) aspirasi. Kategori tersebut ditentukan sesuai dengan Perpres No. 76 tahun 2013 tentang Pengelolaan Pengaduan Pelayanan Publik, Permenpan No. 5 tahun 2009 tentang Pedoman Umum Penanganan Pengaduan Masyarakat Bagi Instansi Pemerintah, Permenpan No. 3 tahun 2015 tentang Roadmap Pengembangan Sistem Pengelolaan Pengaduan Pelayanan Publik Nasional, dan Perwal No. 34 tahun 2017 tentang Pedoman Pengelolaan Pengaduan Masyarakat Tentang Pelayanan Publik.

Text Mining merupakan jenis metode yang umum digunakan untuk proses klasifikasi teks. *Text Mining* merupakan metode untuk mengkonversi teks tidak terstruktur menjadi data teks semi-terstruktur, untuk menemukan pola diantara teks tersebut dan menyelesaikan masalah *information overload* [1], juga *Text Mining* merupakan metode yang sangat cocok untuk diterapkan dalam proses pengambilan informasi dari media sosial, seperti *twitter* [2] . Terdapat beberapa algoritma yang biasa diterapkan pada *Text Mining*, salah satunya yang memiliki tingkat akurasi yang cukup baik adalah algoritma *K-Nearest Neighbor* (KNN) [3]. Algoritma KNN merupakan algoritma yang paling umum digunakan dalam proses klasifikasi, karena bertujuan untuk mengklasifikasikan objek berdasarkan atribut dan data latih [4], terutama untuk tujuan klasifikasi teks yang dikumpulkan melalui media sosial[5]. KNN juga merupakan algoritma klasifikasi yang paling sederhana, dimana output klasifikasi dihitung sebagai *class* dengan frekuensi tertinggi, dengan melakukan teknik *cross validation* menggunakan berbagai *value K*, untuk menentukan hasil akhir *value K* [6]. Menggunakan metode *text mining* dengan algoritma *K-Nearest Neighbor* ini diharapkan dapat membantu pemerintah Kota Semarang dalam pengelolaan pengaduan yang lebih baik, dengan mengklasifikasikan jenis laporan yang disampaikan melalui *twitter* ke dalam 3 (tiga) kategori, yaitu, 1) pengaduan, 2) permintaan informasi, dan 3) aspirasi. KNN ini akan dapat membantu sebagai algoritma pencarian dan klasifikasi teks yang mudah digunakan dan cocok untuk lingkungan yang cukup kompleks [7].

2. METODE PENELITIAN

Data yang digunakan sebagai sampel klasifikasi adalah daftar laporan masyarakat yang masuk ke Bidang Pengelolaan Pengaduan Masyarakat Kota Semarang melalui *twitter* dari 01 Januari hingga 31 Desember 2017. Daftar laporan yang dikumpulkan berjumlah 2253 data *tweet* dan pada gambar 1 merupakan contoh laporan dari masyarakat dengan menggunakan *twitter*.



Gambar 1. Tampilan laporan masyarakat Kota Semarang melalui *twitter*

Metode Klasifikasi Teks dengan Algoritma *K-Nearest Neighbor* (KNN)

1. Tahap pertama proses klasifikasi adalah dengan melakukan *pre-processing* pada data uji dan data latih. Proses ini bertujuan untuk mentransformasikan data ke dalam bentuk representasi format data lain, terdiri dari beberapa langkah, yaitu *cleaning*, *case folding*, *tokenization*, *stopword removal*, dan *stemming* [8].
2. Tahap kedua adalah *Term Weighting*, yang merupakan proses pembobotan dengan *Term*

Frequency – Inverse Document Frequency (*TF-IDF*). Pembobotan ini digunakan untuk menghitung bobot tiap *term* pada setiap dokumen, berdasarkan pada seberapa sering *term* tersebut muncul [9], dengan rumus pada persamaan (1),(2), dan (3) berikut :

$$tf_{t,d} = \sum_{i=1}^{n} t, f \tag{1}$$

$$idf_t = \log\left(\frac{N}{df_t}\right) \tag{2}$$

$$TF - IDF_{t,d} = tf_{t,d} \times idf_t \tag{3}$$

3. Langkah ketiga adalah implementasi algoritma KNN untuk memodelkan isi dokumen teks sebagai ruang vektor untuk menghitung jarak terdekatnya. Tahapan implementasi algoritma KNN adalah berikut [10],
   1) Menentukan nilai parameter *k* sebagai *neighborhood* yang paling dekat dari data latih terhadap data uji,
   2) Menghitung jarak antara nilai vektor data uji dengan semua vektor data latih dengan teorema *Euclidean Distance pada persamaan* (4),

$$d(x, y) = \sqrt{\sum_{k=1}^{n} (x_k - y_k)^2} \tag{4}$$

   3) Mengambil sejumlah nilai parameter *k* data latih terdekat,
   4) Menentukan kelas atau kategori yang dominan sebagai kelas atau kategori untuk data uji.
4. Tahap terakhir dalam proses klasifikasi teks adalah dengan evalusi dan validasi kinerja menggunakan *Confusion Matrix* pada tabel 1 dan tabel 2. *Confusion Matrix* yaitu model perhitungan perbandingan yang menggambarkan nilai positif dan negatif dari hasil klasifikasi. Hasil pengukuran evaluasi dan validasi akan dilanjutkan guna menghitung nilai *precision*, *recall*, *f-measure*, dan *accuracy* [11].

Tabel 1. Model *confusion matrix*

		Nilai Prediksi	
		Positive	*Negative*
Nilai Aktual	*Positive*	*True Positive (TP)*	*False Negative (FN)*
	Negative	*False Positive (FP)*	*True Negative (TN)*

Tabel 2. *Confusion matrix* klasifikasi laporan masyarakat

		Nilai Prediksi			Kategori
		A	B	C	
Nilai Aktual	A	TN	FP	TN	A : Pengaduan
	B	FN	TP	FN	B : Permintaan Informasi
	C	TN	FP	TN	C : Aspirasi

1) *Precision* pada persamaan (5) merupakan perbandingan dari hasil TP dengan keseluruhan dokumen yang diprediksi sebagai *positive* oleh model klasifikasi.

$$Precision = \frac{TP}{TP\ + FP} \quad (5)$$

2) *Recall* pada persamaan (6) merupakan perbandingan hasil TP dengan keseluruhan dokumen bernilai *positive*.

$$Recall = \frac{TP}{TP\ + FN} \quad (6)$$

3) *F-Measure* pada persamaan (7) merupakan gabungan hasil pengukuran *Precision* dan *Recall* untuk mengetahui estimasi kinerja klasifikasi.

$$f - measure = \frac{2TP}{2TP\ + FP\ + FN} \quad (7)$$

4) *Accuracy* pada persamaan (8) merupakan hasil presentasi kesuluruhan dokumen yang diprediksi dengan benar.

$$Accuracy = \frac{\sum TP + \sum TN}{\sum Data\ Uji} \quad (8)$$

3. HASIL DAN PEMBAHASAN

Bagian ini memaparkan hasil dari proses klasifikasi jenis laporan masyarakat. Proses dimulai dengan membagi data laporan yang dikumpulkan pada tahun 2017 ke dalam data latih dan data uji. Kedua, melaksanakan setiap langkah *pre-processing*, kemudian menghitung bobot dengan *TF-IDF* dan implementasi algoritma *K-Nearest Neighbor* (KNN). Terakhir melakukan evaluasi dan uji validasi dengan *Confusion Matrix*.

3.1 Pengumpulan Data

Data laporan masyarakat yang digunakan dikumpulkan melalui twitter dengan hashtag (#) LaporHendi selama tahun 2017, mulai dari 1 Januari – 31 Desember 2017. Total data yang dkumpulkan mencapai 2253 data tweet masyarakat. Data tweet yang terkumpul kemudian dibersihkan melalui beberapa tahapan, yaitu:

A. Data Cleaning

Proses ini dilakukan guna membersihkan data dari jenis data yang tidak valid atau relevan

karena tidak memiliki atribut yang lengkap. Data *tweet* yang dibersihakn diantaranya merupakan *tweet* yang disampaikan secara berulang dan memiliki makna sama, dan tidak termasuk dalam kategori pengaduan, permintaan ataupun aspirasi, serta laporan tersebut tidak terjadi di Kota Semarang.

B. Pemilihan Atribut

Dua (2) atribut yang digunakan pada penelitian ini yaitu, tanggal dan *tweet*. Data yang diambil merupakan data yang berhasil disaring melalui fitur pada *twitter*.

C. Pelabelan

Setelah menentukan data yang akan digunakan, selanjutnya diberi pelabelan terhadap *tweet* yang berhasil terkumpul. Label yang diberikan yaitu, pengaduan, permintaan informasi, dan aspirasi sesuai dengan tata bahasa yang disampaikan oleh masyarakat melalui *tweet*.

Dari tahapan proses diatas, ditentukan data set penelitian terdiri dari 930 data latih, dan 100 data uji. Dengan pembagian 310 untuk masing-masing kategori laporan (pengaduan, permintaan informasi, aspirasi), sedangkan untuk data uji masing-masing kategori, 40 data pengaduan, 40 data permintaan informasi, dan 20 data aspirasi seperti pada gambar 2. Pemberian kategori ini akan diuji pada perhitungan *Confusion Matrix*.

NO	KATEGORI	TWEET
1	Pengaduan	Pak saya warga moh suyudi No 5 (radana finance) Jalan suyudi ada proyek pedestrian yang menyalahi ijin PJM kami. Kontraktor
2	Aspirasi	@hendrarprihadi mhn perbkan di jl gombel lma,sbb saat ad mbil mgok jd ad 2 hmbtn, mbil tsb n jln rusak brkbat mcet **#LaporHe**
3	Pengaduan	Prihatin wes banjir jalan sempit buat parkiran macet pula jln gajahmada #LaporHendi @hendrarprihadi
4	Permintaan Informasi	@hendrarprihadi **#laporhendi** menurut lurah Srondol Kulon kalau tidak berbadan hukum tidak disetujui. Apa ini betul Pak?
5	Pengaduan	@hendrarprihadi #LaporHendi Pak, saya tu heran kenapa di sendangguwo sering mati lampu apalagi hari sabtu minggu, kok paya
6	Pengaduan	Keadaan banjir yg tak tak kunjung surut #LaporHendi @hendrarprihadi Jalan truntum Raya, tlogosari
7	Aspirasi	@hendrarprihadi **#laporhendi** pak ngeCat kampung diagendain lg di K Semawis kan tiap jmt-sbt-mgg malem biar makin rame, c
8	Pengaduan	**#LaporHendi** Pagi td Jam 02.00 WIB telah terjadi kebakaran di Jl.Tikung Baru, SMG UTARA, mohon Bantuan atas musibah ini Yth
9	Aspirasi	Sugeng Siyang Pak @hendrarprihadi , nyuwun tulung bantuannya spy pencetakkan E-KTP di kec. pedurungan bs dipercepat. Matu
10	Permintaan Informasi	bpk walikota smg yg terhormat @hendrarprihadi. pkrjaan jln difatmwati sampai kpn? pedagang di fatmawati mulai mengeluh di
11	Permintaan Informasi	@hendrarprihadi @P3Mkotasmg lampu PJU di jl durian Utara II Srondol Wetan sdh 3 hari mati. Ada apa ya? **#laporHendi**
12	Permintaan Informasi	@hendrarprihadi **#laporhendi** Pak, kenapa pemugaran paving di gang warung kok tdk sore hr? Trims
13	Aspirasi	Pak @hendrarprihadi tlg tertibkan balap liar di Kedung mundu dkt SPBU. **#LaporHendi** #kemalajateng CC @ganjarpranowo
14	Permintaan Informasi	@hendrarprihadi pak kl mau lapor mati lampu didaerah Kuningan smg Utara kemana njih..sdh 1 jam lbh. **#LaporHendi**
15	Pengaduan	Kepada Dishub Semarang, pemasangan lampu bangjo di depan kantor Bank BTPN Peterongan perlu di evaluasi khususnya bagi p
16	Pengaduan	MASALAH IMB Kepada Yth. Pemerintah Kota Semarang Mohon penjelasan mengenai IMB PT. ASSA RENT yg terletak di Tambak A
17	Permintaan Informasi	@hendrarprihadi apakah RS memberikan pengembalian dana pak ? **#LaporHendi**
18	Aspirasi	Pintu tol gayamsari tolong donk help.. setiap hari seperti ini @Dishubkotasmg @TransMarga @hendrarprihadi **#LaporHendi**
...	...	...
1743	Pengaduan	Dinas PJPR Kota Semarang dan Perumnas Palir, mohon dengan sangat agar lampu-lampu penerangan di Kompleks Perumnas Pal

Gambar 2. Hasil pelabelan *tweet*

3.2 *Pre-processing*

Pada tahap ini, data latih dan data uji akan melalui tahapan *pre-processing text*. Tahapan ini akan menghasilkan kata-kata dasar dalam Bahasa Indonesia yang mewakili tiap-tiap kategori yang telah ditentukan sebagai acuan untuk menentukan kategori laporan masyarakat. Tahapan ini menggunakan bahasa pemrograman PHP dengan memanfaatkan *library sastrawi*. Tabel 3 menunjukkan proses dan hasil *pre-processing text*.

1) Tahapan pertama dalam *pre-processing text* yaitu *Cleaning*. Pada tahapan ini, karakter seperti angka, tanda baca, link, dan karakter lain yang terdapat dalam sebuah dokumen dihilangkan karena tidak memiliki pengaruh terhadap pemrosesan dokumen teks tersebut.
2) Selanjutnya adalah tahap *Case Folding* dengan menggunakan perintah seperti pada gambar 3. Pada tahap ini, semua huruf dalam dokumen teks diubah menjadi huruf kecil atau *lowercase*, serta penghapusan karakter seperti *emoticon*.

```
$sentence = strtolower($sentence);
$sentence = $this->replaceEmoji($sentence);
```

Gambar 3. Tampilan *source code* dengan *library sastrawi* pada proses *case folding*

3) *Tokenization.* Pada tahap ini dilakukan pemisahan kalimat ke dalam suku-suku kata. Pada tahap ini dokumen teks sepenuhnya hanya berisi kalimat. Setiap suku kata harus merupakan sebuah kata yang memiliki makna yang merepresentasikan isi dari sebuah dokumen terkait. Kata-kata yang memiliki makna dinamakan *Token*.
4) *Stopword Removal*. Setelah mendapatkan *token* dalam dokumen, *token* tersebut kemudian dipisahkan dari kata penghubung seperti "dan", "yang", "ke", dan kata penghubung lainnya. Dilakukan pula penghapusan terhadap kata yang berulang, dan mengambil satu kata yang saja yang mewakili dengan perintah seperti pada gambar 4. *Token* yang sudah ada dalam *database stopword* pada *library sastrawi* akan dihapus, dan untuk *token* yang belum ada akan diproses.

```
$stopwordFactory = new \Sastrawi\StopwordRemover\StopwordRemoverFactory();
$stopword  = $stopwordFactory->createStopWordRemover();
$hasil_stopword_removal =  $stopword->remove($hasil_formalisasi);
```

Gambar 4. Tampilan *source code* dengan *library sastrawi* pada proses *stopword removal*

5) *Stemming.* Tahap terakhir pada *pre-processing text* merupakan tahapan untuk mengubah kata-kata dari hasil *Stopword Removal* menjadi kata dasar. Pengubahan dilakukan dengan menghapus awalan dan akhiran pada setiap kata.

Tabel 2. Contoh hasil *pre-processing text* pada *sample* data latih

Sample Data Mentah:		
Pak @hendrarprihadi kemarin waktu lewat jalan turunan gombel lama motor saya kena lubang sampe bikin ban bocor #LaporHendi		
Sample Data Latih		
D1	Pak @hendrarprihadi kemarin waktu lewat jalan turunan gombel lama motor saya kena lubang sampe bikin ban bocor #LaporHendi	Kelas : Pengaduan
Pre-processing Data Latih		
D1	*Cleaning*	Pak kemarin waktu lewat jalan turunan gombel lama motor saya kena lubang sampe bikin ban bocor
	Case Folding	pak kemarin waktu lewat jalan turunan gombel lama motor saya kena lubang sampe bikin ban bocor
	Tokenization	pak kemarin waktu lewat jalan turunan gombel lama motor saya kena lubang sampai bikin ban bocor
	Stopword Removal	pak kemarin jalan turunan gombel motor saya kena lubang ban bocor
	Stemming	pak kemarin jalan turun gombel motor saya kena lubang ban bocor

3.3 Pembobotan *Term Frequency – Inverse Document Frequency* (TF-IDF)

Pada tahap ini dilakukan perhitungan bobot tiap kata (*term*) pada sebuah dokumen, berdasarkan seberapa sering *term* muncul pada beberapa dokumen data latih dan data uji yang telah melalui *pre-processing* dengan menggunakan metode TF-IDF. *Term Frequency* (TF) digunakan untuk menghitung frekuensi kemunculan *term* pada setiap dokumen. Kemudian dilakukan perhitungan *Inverse Document Frequency* (IDF) guna menghitung bobot *term* dari kemunculannya di beberapa dokumen. Hasil dari pembobotan ini akan menghasilkan nilai untuk perhitungan *K-Nearest Neighbor* (KNN). Contoh hasil perhitungan ditampilkan pada tabel 3 dan tabel 4.

Tabel 3. Contoh perhitungan *term frequency (TF),* dan *inverse document frequency (IDF)*

Term	*Tf*							*idf*
	DUji	D1	D2	D3	D4	D5	*Df*	*log(Dn/df)*
Pak	0	1	1	1	1	1	5	0,0792
Kemarin	0	1	0	0	0	0	1	0,7782
Jalan	1	1	1	1	1	1	6	0
Turun	0	1	0	0	0	0	1	0,7782
Gombel	0	1	0	0	0	0	1	0,7782
Motor	1	1	1	1	0	0	4	0,1761
Saya	0	1	0	0	0	0	1	0,7782
Kena	0	1	0	0	0	0	1	0,7782
Lubang	2	1	1	0	0	0	3	0,3010
Ban	0	1	0	0	0	0	1	0,7782
Bocor	0	1	0	0	0	0	1	0,7782

Tabel 4. Contoh perhitungan TF-IDF

Term	Di = *tf * idf (Panjang Vektor Term)*					
	DUji	D1	D2	D3	D4	D5
Pak	0	0,0792	0,0792	0,0792	0,0792	0,0792
Kemarin	0	0,7782	0	0	0	0
Jalan	0	0	0	0	0	0
Turun	0	0,7782	0	0	0	0
gombel	0	0,7782	0	0	0	0
Motor	0,1761	0,1761	0,1761	0,1761	0	0
Saya	0	0,7782	0	0	0	0
Kena	0	0,7782	0	0	0	0
lubang	0,6021	0,3010	0,3010	0	0	0
Ban	0	0,7782	0	0	0	0
bocor	0	0,7782	0	0	0	0

3.4 Penerapan Algoritma *K-Nearest Neighbor*

Tahap selanjutnya adalah penghitungan jarak nilai vektor TF-IDF pada *sample* data uji terhadap *sample* data latih, kemudian mengklasifikasikan prediksi kelas dari *sample* data uji menggunakan algoritma *K-Nearest Neighbor*. *Sample* data uji akan dikategorikan ke dalam kelas, antara pengaduan, permintaan informasi, atau aspirasi.

Perhitungan algoritma *K-Nearest Neighbor* pada tabel 5 pertama adalah dengan menentukan nilai parameter *k* atau jumlah dokumen ketetanggan terdekat. Pemberian kategori untuk sample data uji ditentukan dari jumlah kategori paling dominan pada nilai *k sample* data latih. Pada penelitian ini, menggunakan nilai parameter *k* = 3. Selanjutnya menghitung jarak antara hasil vektor *sample* data uji dengan semua vektor *sample* data latih dengan menggunakan teorema perhitungan jarak *Euclidean Distance*, cara perhitungan adalah dengan mengakar dari kuadrat perbedaan 2 vektor. Tabel 5 dan tabel 6 menunjukkan kesimpulan dari hasil perhitungan *Euclidean Distance* pada tabel 6 jarak antara 2 vektor pada contoh 5 buah *sample* data latih. Berdasarkan hasil perhitungan, ditemukan jarak beserta ranking antara vektor *sample* data uji sengan semua vektor data latih. Dengan menggunakan nilai parameter *k*=3, diambil jarak terdekat yaitu dokumen D2, D1, dan D3. Masing-masing dokumen *sample* data latih tersebut memiliki kategori kelas yang berbeda, dokumen D1, D2, dan D3 termasuk dalam kelas pengaduan. Dokumen D4 termasuk ke dalam kelas permintaan informasi, dan dokumen D5 masuk ke dalam kelas aspirasi. Dokumen dengan jarak terdekat berdasarkan nilai parameter *k* yaitu D2, D1 dan D3, sehingga dapat disimpulkan bahwa *sample* data uji termasuk ke dalam kelas Pengaduan.

Tabel 5. Perhitungan jarak antara 2 vektor pada *euclidean distance*

	(Di - DUji) ^ 2				
	D1	D2	D3	D4	D5
	0,0063	0,0063	0,0063	0,0063	0,0063
	0,6055	0	0	0	0
	0	0	0	0	0
	0,6055	0	0	0	0
	0,6055	0	0	0	0
	...	...	...	...	...
Total	9,5886	9,4979	10,2251	10,6340	14,0394
Akar Kuadrat Total	3,0965	3,0819	3,1977	3,2610	3,7469

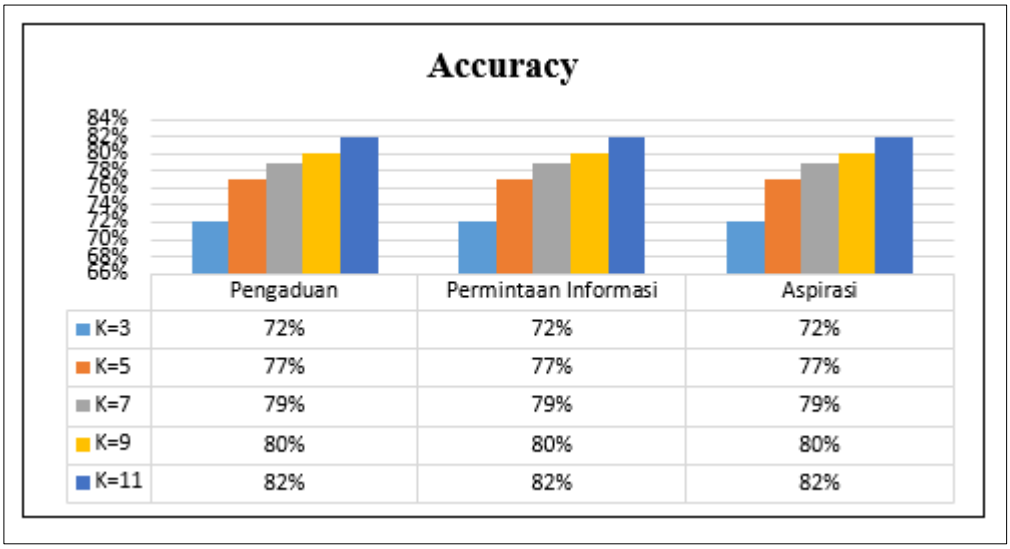
Tabel 6. Hasil perhitungan *euclidean distance* pada 5 *sample* data latih

Dokumen	D1	D2	D3	D4	D5
Jarak	3,0965	3,0819	3,1977	3,2610	3,7469
Ranking	2	1	3	4	5

3.5 Evaluasi dan Validasi

Tahapan evaluasi dan validasi dilakukan setelah proses klasifikasi masing-masing nilai parameter *k*. Evaluasi dan validasi menggunakan *Confusion Matrix*, dimana evaluasi dan validasi dengan membandingkan pada 2 (dua) dimensi, yaitu, kelas prediksi dan kelas aktual. Bentuk *matrix* tertera pada tabel 2 bab sebelumnya, yaitu mencari nilai *True Positive (TP)*, *False Positive (FP)*, *True Negative (TN)*, dan *False Negative (FN)*. Setelah itu mencari nilai *precision*, *recall*, *f-measure* dan *accuracy* untuk uji evaluasi dan validasi.

Nilai parameter *k* yang diukur adalah *k*=3, *k*=5, *k*=7, *k*=9, dan *k*=11. Hasil perhitungan uji evaluasi dan validasi disimpulkan untuk penentuan *accuracy*, ditunjukkan pada gambar 5.



Gambar 5. Grafik hasil perhitungan *accuracy* pada uji parameter *k*

Pada gambar 5 diatas dapat disimpulkan bahwa nilai k tertinggi untuk kelas pengaduan, permintaan informasi, dan aspirasi yaitu pada k=11, dengan hasil 82%.

4. KESIMPULAN DAN SARAN

Algoritma *K-Nearest Neighbor* memberikan hasil akurasi yang baik untuk proses klasifikasi data laporan masyarakat, ke dalam 3 (kategori), yaitu, pengaduan, permintaan informasi, dan aspirasi. Menggunakan 2235 data laporan yang dikumpulkan selama tahun 2017 oleh pemerntah Kota Semarang. Kemudian melalui pembersihan didapatkan 930 data latih dengan masing-masing 310 data latih untuk setiap kategori laporan masyarakat (aspirasi, permintaan informasi, aspirasi), dan 100 data uji. Penerapan algoritma *K-Nearest Neighbor* untuk menentukan jarak parameter k. Terakhir, berdasarkan uji evaluasi dan validasi dengan *Confusion Matrix*, ditemukan bahwa parameter k = 11 memiliki nilai *accuracy* tertinggi dalam penentuan kelas kategori laporan masyarakat, yaitu 82%.

Penelitian ini dapat disempurnakan dengan menambahkan data aktual pada data uji kelas aspirasi, sehingga memberikan hasil *Confusion Matrix* yang lebih baik. Serta dapat dijalankan perbandingan dengan algoritma *text mining* lain dalam menentukan kelas laporan masyarakat Kota Semarang.

DAFTAR PUSTAKA

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

[2] P. B. Dastanwala and V. Patel, "A review on social audience identification on twitter using text mining methods," *Proc. 2016 IEEE Int. Conf. Wirel. Commun. Signal Process. Networking, WiSPNET 2016*, pp. 1917–1920, 2016.

[3] W. Gata and Purnomo, "Akurasi Text Mining Menggunakan Algoritma K-Nearest Neighbor pada Data Center Berita SMS," *J. Format*, vol. 6, no. 1, 2017.

[4] Okfalisa, I. Gazalba, Mustakim, and N. G. I. Reza, "Comparative analysis of k-nearest neighbor and modified k-nearest neighbor algorithm for data classification," *Proc. - 2017 2nd Int. Conf. Inf. Technol. Inf. Syst. Electr. Eng. ICITISEE 2017*, vol. 2018-Janua, pp. 294–298, 2018.

[5] N. Anggraini and M. J. Tursina, "Sentiment Analysis of School Zoning System On Youtube Social Media Using The K-Nearest Neighbor With Levenshtein Distance Algorithm," in *The 7th International Conference on Cyber and IT Service Management (CITSM 2019)*, 2019.

[6] K. Nyodu and K. Sambjo, "Automatic Identification of Arunachal language Using K-Nearest Neighbor Algorithm," in *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, 2018, pp. 213–216.

[7] L. Yang, Q. Yang, Y. Li, and Y. Feng, "K-Nearest Neighbor Model based Short-Term Traffic Flow Prediction Method," in *2019 18th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES)*, 2019, pp. 27–30.

[8] A. Hetami and B. Dwijawisnu, "Perancangan Information Retrieval (IR) Untuk Pencarian Ide Pokok Teks Artikel Berbahasa Inggris dengan Pembobotan Vector Space Model," *J. Ilm. Teknol. dan Inf. ASIA*, vol. 9, no. 1, 2015.

[9] M. Bramer, *Principles of Data Mining*. Springer, 2007.

[10] X. Wu and V. Kumar, *The Top Ten Algorithms in Data Mining*. Minnesota: CRC Press, 2009.

[11] C. Sammut and G. I. Webb, *Encyclopedia of Machine Learning*. New York: Springer, 2010.