MATICS : Jurnal Ilmu Komputer dan Teknologi Informasi ISSN : 1978-161X(p); 2477-2550(e)
Volume 10, No. 2 (2018), pp 30-38 DOI : 10.18860/mat.v10i2.5327
Received : July25th 2018; **Accepted** : August 9th 2018 ; **Avalaible Online** : September 15th 2018

Identifikasi Kemiripan Teks Menggunakan Class Indexing Based dan Cosine Similarity Untuk Klasifikasi Dokumen Pengaduan

Syahroni Wahyu Irianandа[1], Muhammad Aziz Muslim[2], Harry Soekotjo Dachlan[3]

Abstrak – Pada Sistem "LAPOR!" penanganan laporan bergantung pada administrator sistem yang membaca secara manual setiap laporan yang masuk [3]. Hal ini dapat menyebabkan kesalahan dalam menangani keluhan [4], dan jika aliran datanya sangat besar dapat membutuhkan waktu minimal tiga hari, hal ini sensitif terhadap inkonsistensi [3].
Dalam penelitian ini penulis mengusulkan suatu model atau pendekatan yang dapat mengukur dan mengidentifikasi kemiripan dokumen laporan yang dilakukan secara terkomputerisasi yang dapat mengidentifikasi kemiripan antara *Query* dengan *Document*. Dalam penelitian ini penulis mempekerjakan skema pembobotan kata *Class-Based Indexing*, dan *Cosine Similarity* untuk menganalisa kemiripan dokumen. Nilai CoSimTFIDF, CoSimTFICF dan CoSimTFIDFICF kemudian ditetapkan sebagai set fitur untuk proses klasifikasi teks menggunakan metode *K-Nearest Neighbor (K-NN).* Dalam pengujian yang telah dilakukan ditemukan bahwa hasil akurasi optimal dengan preprocessing Stemming dan hasil terbaik dari semua fitur adalah rasio data latih 75% dan data uji 25% pada fitur CoSimTFIDF yaitu 84%. Nilai k = 5 memiliki tingkat akurasi yang tinggi yaitu 84,12%

Kata Kunci– pengaduan, kemiripan teks, class-based indexing, cosine similarity, k-nearest neighbor, lapor!,

I. PENDAHULUAN

Jumlah data laporan pengaduan dan opini publik yang masuk pada *platform* "LAPOR!" (Layanan Pengaduan dan Aspirasi Online Rakyat) dapat menjadi sumber informasi untuk mengukur kinerja pelayanan lembaga pemerintahan [1]. Rata-rata 900 laporan setiap hari, hanya 13% - 14% laporan diproses, sementara sekitar 86% tetap menjadi subyek yang belum diketahui dan diarsipkan. Saluran yang paling banyak digunakan adalah melalui SMS sekitar 80% - 90% laporan [2].

Manuscript received September 22, 2019.This work was supported in part by eknik Elektro of Brawijaya State University , Malang, Indonesia.

Syahroni Wahyu Irianandа is with the Teknik Elektro of Brawijaya State University , Malang, Indonesia (email roniwahyu@student.ub.ac.id)

Muhammad Aziz Muslim was with Teknik Elektro of Brawijaya State University , Malang, Indonesia (e-mail: muh_aziz@ub.ac.id).

Harry Soekotjo Dachlan is is the Teknik Elektro of Brawijaya State University , Malang, Indonesia (email harrysd@ub.ac.id)

Administrator yang terbatas dan angka laporan pengaduan yang cukup tinggi menjadi penyebab utama kurangnya kualitas layanan terutama karakteristik daya tanggap (responsivitas) [2]. Penanganan laporan bergantung pada administrator sistem yang membaca secara manual setiap laporan yang masuk [3]. Hal ini dapat menyebabkan kesalahan dalam menangani keluhan [4], dan jika aliran datanya sangat besar dapat membutuhkan waktu minimal tiga hari, hal ini sensitif terhadap inkonsistensi [3].

Maka dari itu diperlukan solusi terhadap permasalahan analisis laporan pengaduan yang dapat membantu administrator "LAPOR!" dalam melakukan menentukan kategori sehingga analisis *big data* menjadi sangat penting [2].

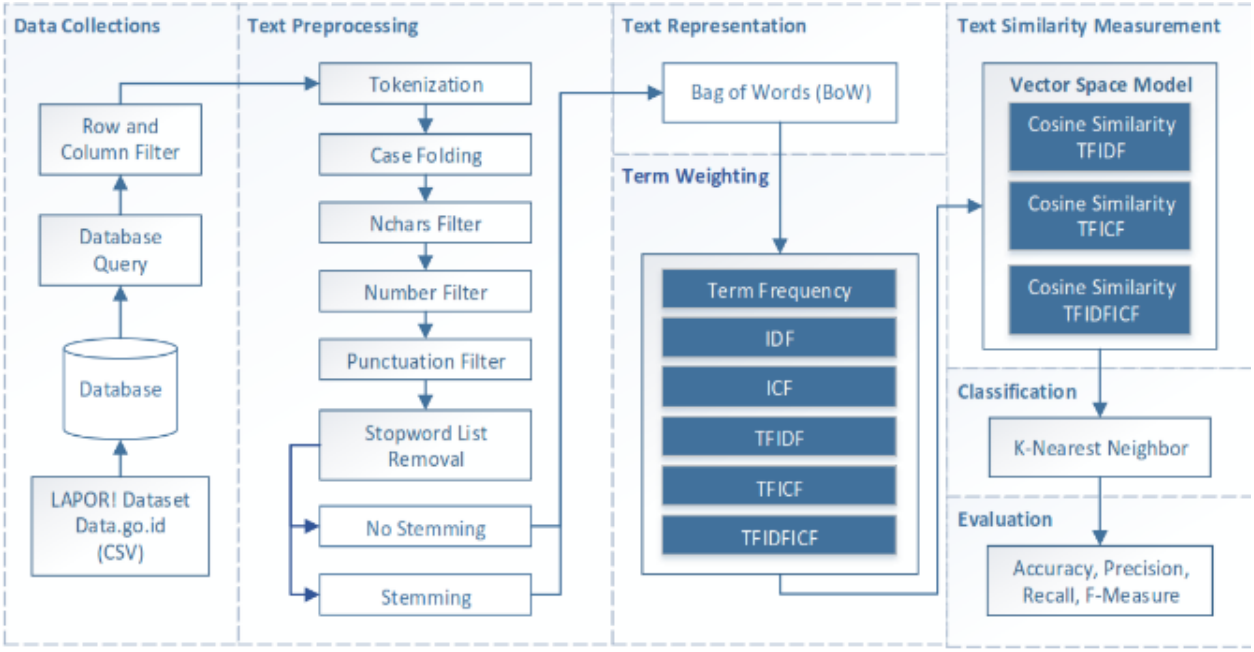
Dalam penelitian ini penulis mengusulkan suatu model atau pendekatan yang dapat mengukur dan mengidentifikasi kemiripan dokumen laporan yang dilakukan secara terkomputerisasi yang dapat mengidentifikasi kemiripan antara *Query* dengan *Document*, berikutnya dilakukan klasifikasi terhadap *Query* untuk memprediksi kelas atau kategorinya berdasarkan nilai kemiripan terbesar atau yang mendekati nilai 1 (satu) antara *Query (Q)* dengan koleksi *Document (D)*. Penulis mempekerjakan skema pembobotan kata *Class-Based Indexing*, kemudian dikomparasi dengan skema pembobotan lainnya yaitu TFIDF dan TFICF. Nilai bobot TFIDF, TFICF dan TFIDFICF kemudian dikonversi ke dalam koordinat kartesius dan dihitung kemiripannya menggunakan fungsi *Cosine Similarity* untuk menganalisa kemiripan dokumen teks dengan cara mendapatkan kemiripannya dengan cara mengukurnya dalam bentuk jarak vektor kemiripan. Nilai *Cosine Similarity* dari ketiga skema pembobotan tersebut (CoSimTFIDF, CoSimTFICF, CoSimTFIDFICF) kemudian ditetapkan sebagai set fitur untuk proses klasifikasi. Berikutnya dilakukan proses klasifikasi teks menggunakan metode *K-Nearest Neighbor (K-NN)* untuk klasifikasi dokumen dan memprediksi kategori dokumen baru berdasarkan fitur-fitur tersebut.

Penelitian ini bertujuan untuk mengidentifikasi dan mengevaluasi kemiripan teks menggunakan metode TF-IDF-ICF *(Class Indexing Based)* dan *Cosine Similarity.*

II. KAJIAN PUSTAKA

Suatu kata disebut mirip secara Leksikal, yaitu ketika suatu kata memiliki urutan karakter yang sama. Dan ketika suatu kata memiliki makna yang sama maka disebut mirip secara Semantik [5]. Penelitian [6] dengan memanfaat kan TF.IDF.ICF untuk klasifikasi dokumen pengaduan (e-complaint) mahasiswa menggunakan *Centroid Based Classifier*, dikombinasikan dengan TF.IDF.ICF, *Cosine Similarity* dan *Class Feature Centroid*. [7] Melakukan kategorisasi ide kreatif pada suatu perusahaan menggunakan algoritma K-NN dan TF.IDF.ICF. [8] Mengklasifikasikan pengaduan SambatOnline Kota Malang menggunakan algoritma K-NN, *Cosine Similarity* dan *Chi Square* dari pada TFIDF. [9] Menggunakan algoritma K-NN dan seleksi fitur TFIDF, dan *Categorical Proporsional Difference (CPD)*. Dataset yang sama digunakan [10] dengan mempekerjakan algoritma NW-K-NN, term weighting TFIDF filter N-Gram, dan Unigram pada preprocessing. Pengujian yang dilakukan adalah eksperimen pada proses "Preprocessing" yaitu melalui sub-proses *Stemming* dan tidak menggunakan Stemming. Eksperimen dengan (term weighting) yang berbeda yaitu menggunakan TF-IDF, TF-ICF dan TF-IDF-ICF berikut dengan ekperimen variasi *Cosine Similarity* berdasarkan masing-masing pembobotan kata tersebut. ekperimen dengan variasi jumlah data, dan variasi jumlah data training dan data testing. Kemudian pada hasil akhir akan dievaluasi pengaruh kinerja keduanya.

Metode yang digunakan untuk menganalisa kemiripan antara laporan pengaduan yang baru masuk *(Query)* dengan laporan yang telah diproses administrator *(Document)* adalah *Cosine Similarity*. Hasil *term-weighting* dengan TF-IDF, TF-ICF dan TF-IDF-ICF kemudian dikonversi ke dalam koordinat kartesius dan dihitung menggunakan fungsi *Cosine Similarity* untuk mendapatkan sudut kemiripannya *dan* mengukur jarak vector. Berikutnya dilakukan proses klasifikasi teks berdasarkan fitur *Cosine Similarity* TF-



Gambar 1 Kerangka Konsep Identifikasi Kemiripan Dengan Class-Based Indexing dan Cosine Similarity

Hasil eksperimen [11] menunjukkan bahwa klasifikasi teks dapat digunakan untuk mengevaluasi kualitas layanan dengan data teks dari penanganan keluhan pelanggan (complaint). Metode ini dapat memecahkan masalah evaluasi otomatis dalam manajemen penanganan keluhan pelanggan. [11]

III. METODE PENELITIAN

Secara garis besar kerangka solusi masalah dapat dilihat pada Gambar 1, yang terdiri dari 7 (tujuh) proses utama, yaitu Pengumpulan data *(Data Collecting)*, pra-proses teks *(Text Preprocessing)*, representasi teks *(Text Representation)*, seleksi fitur *(Feature Selection)* mencakup *Term Weighting* pada umumnya (TFIDF) dan *Class Based Indexing* (ICF), klasifikasi, dan evaluasi.

Penelitian ini menggunakan tiga skema pembobotan untuk dikomparasi dan dievaluasi untuk mendapatkan peng pembobotan yang yang memiliki hasil paling optimal. IDF (CoSimTFIDF), TF-ICF (CoSimTFICF), TF-IDF-ICF (CoSimTFIDFICF) yang menggunakan skema pembobotan yang berbeda-beda. Semakin besar nilai ketiga fitur cosine similarity yaitu mendekati nilai 1 (satu), maka semakin mirip suatu *Query (q)* dengan koleksi *Document (d).* Metode K-Nearest Neighbor (KNN) dipilih untuk melakukan klasifikasi unntuk memprediksi kategori Query.

A. Class Based Indexing (ICF)

Skema pembobotan berbasis kategori diusulkan [12]. Penelitian ini memperkenalkan Frekuensi Kategori Terbalik *(Inverse Category Frequency)* dalam skema pembobotan istilah untuk tugas klasifikasi teks. Dua konsep didefinisikan sebagai: *Frekuensi Kategori (CF)* yaitu jumlah kategori di mana istilah *(t)* muncul dan *Frekuensi Kategori Terbalik (ICF)* yang formulanya mirip dengan IDF. [12]. Konsep *Class Based Indexing (ICF)* berikutnya dikembangkan oleh [13], Pada penelitian tersebut dikenalkan Metode pengindeksan

otomatis menggunakan kombinasi berbasis dokumen (IDF) dan kelas/kategori (ICF) yang lebih baik yaitu TFIDFICSdF (Term Frequency Inverse Document Frequency Inverse Class Space Density Frequency), dimana (d) merupakan *density* atau kepadatan anggota term atau kemunculan suatu term dalam kategori atau kelas tertentu. Pada skema pembobotan istilah IDF hanya memperhatikan kemunculan term pada kumpulan dokumen dan melakukan pembobotan berbasis dokumen tanpa memperhatikan kelas/kategori yang merupakan induk dokumen tersebut. Sementara pendekatan menggunakan *Inverse Class Frequency (ICF)* memperhatikan kemunculan term pada kumpulan kategori/kelas. Term yang jarang muncul pada banyak kelas adalah term yang bernilai untuk klasifikasi. Semakin jarang kemunculan term tersebut, maka nilainya akan semakin besar atau mendekati nilai 1 (satu), dan sebaliknya semakin sering kemunculan term tersebut maka nilainya semakin kecil atau mendekati nilai 0 (nol). Kepentingan tiap term diasumsikan memilki proporsi yang berkebalikan dengan jumlah kelas yang mengandung term. Penentuan indeks yang akurat juga bergantung pada kepentingan term terhadap kelas atau kelangkaan term pada keseluruhan kelas *(rare term)*. Sehingga dibutuhkan term weighting berbasis kelas yang dinamakan inverse class frequency (ICF). Namun ICF hanya memperhatikan term yang ada pada kelas tanpa memperhatikan jumlah term dalam dokumen yang menjadi anggota kelas. Formula ICF dihitung dengan formula:

(1)

Dimana C adalah jumlah seluruh kelas/kategori dalam koleksi cf_i adalah jumlah kelas/kategori yang mengandung term t_i

B. Persiapan dan Pengolahan Data

Proses pengumpulan data diawali dengan mengunduh data laporan pengaduan aplikasi "LAPOR!" yang berupa data teks. Data ini didapatkan dengan cara mengunduh data yang telah dipublikasikan pada portal berbagi data publik yaitu http://data.go.id. Kemudian data yang masih dalam bentuk file CSV tersebut di ekspor ke dalam database MySQL, tujuannya adalah untuk mempermudah peneliti melakukan eksperimen.

Penggalian data teks sangat bergantung pada data yang digunakan. Pada penelitian ini digunakan kumpulan data (dataset) laporan pengaduan masyarakat. Dataset utama merupakan data platform "LAPOR!" yang telah dipublikasikan. Berikut ini merupakan dataset yang digunakan dalam penelitian ini:

1) Dataset LAPOR!

Dataset utama merupakan data sekunder yang merupakan data aliran laporan pengaduan pada platform "LAPOR!" sejak tahun 2012 hingga Januari 2015. Data ini secara bebas dapat didownload pada situs berbagi data pemerintahan secara terbuka *(Open Government Indonesia)* yaitu data.go.id. Berikut ini merupakan contoh data pengaduan "LAPOR!" seperti pada Tabel 4.1

Tabel 4.1 Tabel data sampel laporan pengaduan "LAPOR!"

JudulLaporan	*IsiLaporan*	*Kategori*
Peserta KKS Belum Mendapat KIP untuk Anak Sekolahnya (Kuningan, Jabar)	38fcob45574000 saya penerima kks.tetapi saya tdk mendapat kartu indonesia pintar padahal anak saya sudah sekolah semua.bagaimana?	Kartu Indonesia Pintar (KIP)

2) Query Database (Input Data)

Query Database merupakan tahap awal daripada penelitian ini. Istilah *query database* yang dimaksud adalah suatu query bahasa database SQL yang digunakan pada server MySQL untuk memilih sekumpulan data. Data ini digunakan sebagai data masukan (input data) pada proses awal penelitian. Peneliti perlu melakukan pembatasan data, dalam rangka efisiensi waktu eksperimen, mengurangi dimensi data yang *irrelevant* dan data *noise*, serta untuk proses pengujian dan evaluasi penelitian yang efektif, maka dalam penelitian ini menggunakan data laporan yang sesuai dengan kriteria berikut ini: a) Data laporan masuk pada tahun 2015, b) Melalui kanal SMS atau aplikasi mobile c) Status laporan "Selesai" d) Memperhatikan prioritas urusan dalam pemerintahan yaitu:
1) Pendidikan 2) Kesehatan 3) Inftrastruktur . Query data dengan kriteria tersebut, menghasilkan jumlah data sekitar 7.134 baris data dan 82 kategori. Kriteria pada poin empat dipilih dengan memperhatikan jumlah data terbanyak pada kategori yang terkait dengan prioritas urusan yang terdapat pada poin empat, Dengan demikian jumlah baris data secara drastis berkurang menjadi 2.825 baris data dan jumlah kategori berkurang dari 82 kategori menjadi 8 kategori seperti yang disajikan pada Tabel 4.2

Tabel 4.2 Query Database Berdasarkan Kategori

Kategori	*Jumlah Data*
Kartu Indonesia Pintar (KIP)	1.389
Infrastruktur	574
Kartu Indonesia Sehat (KIS)	493
Pendidikan	131
Kesehatan	111
BPJS Kesehatan	79
Pendidikan Dasar dan Menengah (Dikdasmen)	35
Pelayanan Kesehatan	13
Total Dokumen	**2.825**

Penelitian ini menggunakan beberapa scenario ekperimen yang salah satunya adalah variasi dataset yang terdapat pada Tabel 4.3. Skenario ini bertujuan untuk untuk menginvestigasi pengaruh jumlah baris data terhadap proses terkait.

Tabel 4.3 Tabel Variasi Dataset

Seri Dataset	*Jumlah Baris Data*
Dataset25	25
Dataset50	50
Dataset75	75
Dataset100	100
Dataset200	200
Dataset300	300
Dataset400	400

Pada variasi dataset tabel 4.3 tersebut setiap anggota data dalam dataset tersebut dipilih secara acak (random sampling) sebanyak Dataset(n) dari dataset "LAPOR!" sesuai dengan kriteria yang telah ditentukan pada tabel 4.2. Pemilihan dataset secara acak dilakukan pada proses awal sebelum proses preprocessing dilakukan. Dataset tersebut memiliki anggota atau jumlah baris data sesuai dengan yang tertera pada nama dataset.

3) Data Partition

Sebagai bentuk simulasi dan eksperimen dalam penelitian ini dilakukan *Data Partition* atau partisi data. Ini dilakukan dengan cara membagi jumlah keseluruhan baris data dalam dataset pada tabel 4.3 menjadi dua bagian yaitu 1) Dataset *Documents (D)* sebesar 90%, 2) Dataset *Query (Q)* sebesar 10%

Setelah dilakukan proses Preprocessing dengan stemming maupun tanpa stemming selanjutnya dataset ini dibagi menjadi dua bagian yaitu 90% untuk dataset dokumen (D) dan 10% digunakan untuk dataset query (Q). Pembagian data juga dilakukan secara acak *(random sampling)* dengan demikian didapatkan anggota data seperti pada tabel berikut.

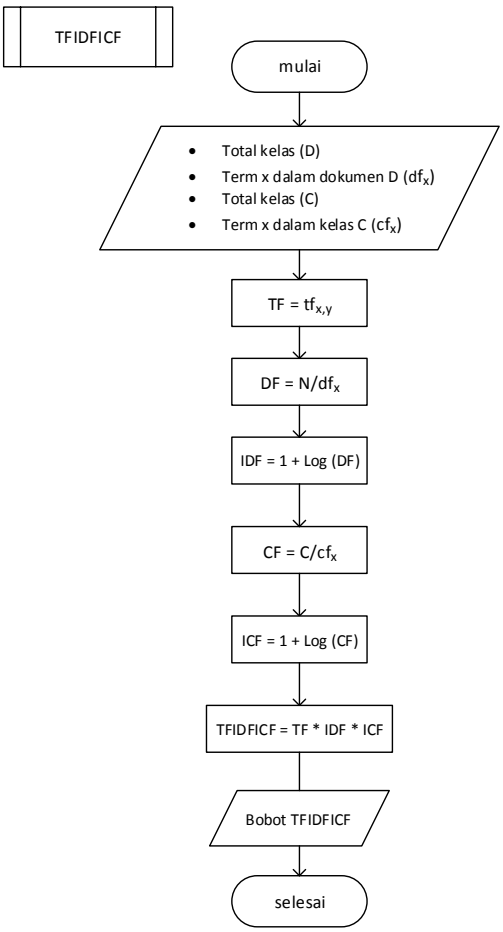
Tabel 4.4 Tabel Partisi Data Document (D) dan Query (Q)

Seri Dataset	*90% (D)*	*10% (Q)*	*Total Data*
Dataset25	22	3	25
Dataset50	45	5	50
Dataset75	67	8	75
Dataset100	90	10	100
Dataset200	180	20	200
Dataset300	270	30	300
Dataset400	360	40	400

Pada dataset dokumen (D) kolom kategori masih tetap digunakan pada dataset, namun pada dataset query (Q) kolom kategori tersebut tidak dicantumkan. Keduanya digunakan untuk simulasi dan ekperimen, dimana hanya satu anggota dataset (Q) dibandingkan dengan banyak anggota dalam dataset (D). Berdasarkan Tabel 4.4 tersebut, dapat diasumsikan bahwa jika scenario eksperimen menggunakan Dataset25, maka salah satu dari tiga data query (Q) dan hanya dapat dinilai kemiripannya dengan maksimal 22 data dokumen (D). Demikian pula jika diasumsikan scenario eksperimen menggunakan variasi Dataset50, maka salah satu dari 5 data query (Q) hanya dapat dinilai kemiripannya dengan maksimal 45 data dokumen (D). Hal ini juga berlaku untuk scenario pengujian menggunakan variasi dataset lainnya sesuai yang telah ditentukan pada Tabel 4.4

Data ini dapat diunduh dalam bentuk file .csv. kemudian dataset tersebut diimpor ke menjadi database MySQL dengan tujuan agar dapat dikelola dan diolah dengan mudah. Disamping itu KNIME juga dapat mengakses database MySQL.

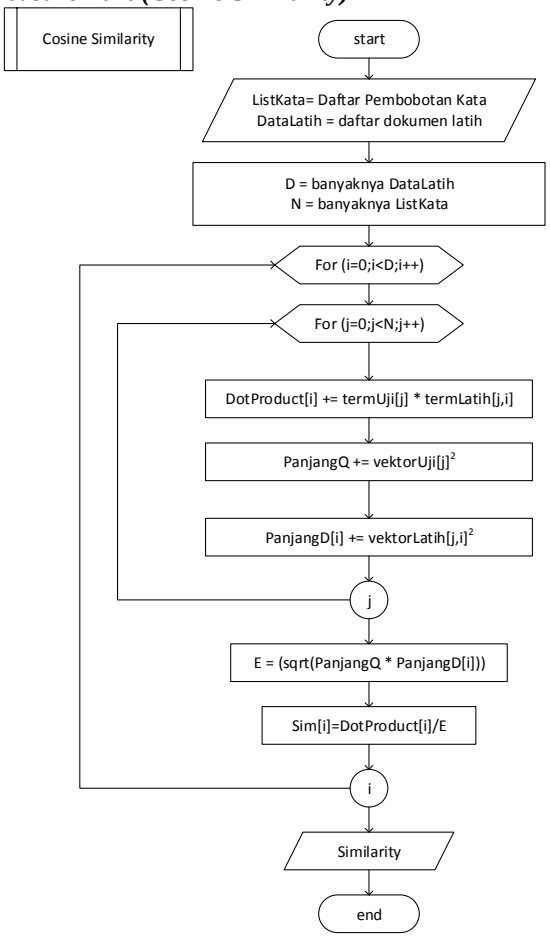
C. Diagram Alir Subsistem Term Weighting



Gambar 2. Diagram Alir Sub Sistem *Term Weighting*

Proses TFIDFICF merupakan gabungan dari proses pembobotan kata terhadap dokumen (TFIDF) dan pembobotan kata terhadap kategori (TFICF) dimana proses ini diawali dengan cara mendapatkan jumlah total dokumen (D), jumlah kata (x) yang terkandung pada dokumen (df_x) dan jumlah total kategori (C), jumlah kata (x) yang terkandung pada kelas atau kategori (cf_x). Setelah didapatkan jumlah total D dan C, selanjutnya adalah mendapatkan Term Frequency (TF), dan Document Frequency (DF), dan Class Frequency (CF). Setelah diketahui nilai TF, DF dan CF berikutnya adalah mendapatkan nilai inverse daripada DF (IDF) dan inverse daripada CF (ICF). Dari kombinasi bobot frekuensi tersebut menghasilkan nilai bobot TFIDF, TFICF, dan TFIDFICF yang akan digunakan untuk proses selanjutnya yaitu normalisasi menggunakan Cosine Similarity.

D. Diagram Alir Sub Proses *Similarity Measurement (Cosine Similarity)*



Gambar 3. Diagram Alir Sub Sistem *Cosine Similarity*

E. Skenario Eksperimen

Tahap perancangan ekperimen ini bertujuan guna memberikan panduan pada proses eksperimen model aliran kerja (workflow) untuk dapat dievaluasi. Berikut ini adalah proses ekperimen pada sistem yang antara lain:

1. Ekperimen dengan variasi preprocessing
2. Ekperimen dengan variasi bobot fitur (term weighting)
3. Ekperimen dengan variasi jumlah dataset
4. Ekperimen dengan variasi rasio jumlah data

IV. HASIL DAN PEMBAHASAN

A. Input Data

Data masukan merupakan data hasil query dari database. Kemudian query tersebut ditentukan sebagai Dataset. Detail tentang variasi dataset terdapat pada Tabel 4.3 di bab sebelumnya,

Tabel 5.1 Sampel Query Database untuk Input Data

ISILAPORAN	KATEGORI
34 QJYG 22!54004 KELUARGA KM MBURAK GINTING BLUM MENDAPAT KIP ATAS NAMA KEVIN	Kartu Indonesia Pintar (KIP)
KEPADA YTH. DINAS PEKERJAAN UMUM TOLONG JALAN RAYA DARI TINGGANG SAMPAI KECAMATAN NGRAHO, TOLONG DI BERI LAMPU PENERANGAN PAK, SERING TERJADI KECELAKAAN MENGINGAT JALANYA SUDAH BAIK, SEBELUMNYA SAYA UCAPKAN TRIMAKASIH {DARI CAK TO NGRAHO}	Infrastruktur

dalam tabel tersebut terdapat perincian Dataset dan jumlah data hasil query database. Tabel 5.1 berikut ini adalah sebagian contoh query database yang digunakan untuk input data dalam penelitian ini. Hasil query memiliki lebih dari 10 kolom, pada proses selanjutnya dilakukan filter kolom dan baris sehingga didapatkan dataset yang siap dilakukan Preprocessing. Kolom yang digunakan untuk input preprocessing adalah kolom ID, IsiLaporan, dan Kategori.

B. Preprocessing

Hasil query database digunakan sebagai input dalam proses ini. Kemudian pada kolom IsiLaporan dilakukan serangkaian proses preprocessing yang secara detail tahapannya dijelaskan pembahasan selanjutnya.

1) Stopword List Removal

2) Stemming

3) Pembobotan Kata (Term Weighting)

Pembobotan kata diperlukan untuk mengetahui acuan nilai yang jelas tentang tingkat kepentingan suatu kata, tingkat keunikan kata, dan tingkat pengaruh kata terhadap dokumen dan kelas/kategori dokumen.

Tabel 5.11 Sampel Pembobotan Kata (Term Weighting)

Term	TF	IDF	ICF	TF*IDF	TF*ICF	TF*IDF*ICF
keluarga	0.100	0.821	0.699	0.082	0.070	0.057
ginting	0.100	1.959	0.954	0.196	0.095	0.187
dapat	0.100	0.788	0.477	0.079	0.048	0.038
kerja	0.042	1.322	0.699	0.055	0.029	0.039

TF *(term frequency)* yang merupakan nilai atau banyaknya kata/term dalam koleksi dokumen. Rumus yang digunakan antara lain untuk TF menggunakan persamaan (2.1), IDF menggunakan persamaan (2.3), ICF (4.1), TDIDF (2.4),TFICF (2.1) dan (4.1), TFIDFICF merupakan hasil perkalian dari persamaan (2.4) dan (4.1). Tabel 5.11 merupakan tabel contoh pembobotan kata sebagai berikut

4) Hasil Pengukuran Cosine Similarity

Pada sub bab ini didiskusikan bagaimana pengukuran cosine similarity antara Query (Q) dengan Documents (D). Sebagaimana telah diketahui bahwa Cosine Similarity ini merupakan perkalian product (Dot Product) dari kedua vektor Q dan vektor D. Rumus cosine similarity menggunakan persamaan (2.7). Pada penelitian ini terdapat beberapa tahapan dalam pengukuran Cosine Similarity yang antara lain: a) Menghitung bobot kata (term weighting) pada *Documents* (D) untuk setiap skema pembobotan yaitu TFIDF, TFICF, dan TFIDFICF seperti yang tertera pada tabel 5.11. b) Membentuk vektor Q dan D untuk setiap skema pembobotan kata seperti yang tertera pada tabel 5.12 sampai dengan tabel 5.17 c) Menentukan data uji atau *query* (Q) yang ingin dilakukan pengujian dengan semua *documents* (D). Contoh data uji (Q) tertera pada tabel 5.18 d) Menghitung bobot kata (term weighting) Query (Q) seperti pada tabel 5.19 e) Perkalian dot vector Q dan vector D *(inner product)* seperti pada tabel 5.20 dan 5.21 f) Menghitung panjang vector Q *(magnitude)* tertera pada tabel 5.22 g) Menghitung panjang vector D (magnitude) seperti pada tabel 5.23, dan 5.24 h) Menghitung *Cross Product* |Q| dan |D| pada tabel 5.25

a) Bobot TFIDF Query (Q) dan Documents (D)

Tabel berikut ini merupakan representasi bobot TFIDF antara *query (Q)* dalam hal ini adalah *Query0* dengan *documents (D)* dari proses inner product dimana setiap nilai term *Query0* dikalikan dengan setiap nilai term *D0* sampai *Dn.*

Tabel 5.20 Bobot TFIDF Query dan Documents

Term	Query0	D0	D1	D2	D3	D4	...
anak	0.098	0.043	0.000	0.114	0.043	0.105	...
bantu	0.183	0.000	0.000	0.000	0.000	0.000	...
dapat	0.113	0.000	0.088	0.000	0.049	0.121	...
kip	0.094	0.041	0.146	0.109	0.000	0.000	...
sekolah	0.123	0.000	0.000	0.000	0.000	0.000	...
…	…	…	…	…	…	…	...

Nilai TFIDF setiap dokumen didapatkan dari vector dokumen seperti pada tabel 5.12, sedangkan nilai TFIDF query didapatkan dari vector query seperti pada tabel 5.15. Berikut ini adalah contoh perhitungan inner product antar Query0 dengan D0 dan Query0 dengan D6

Nilai TFIDF setiap dokumen didapatkan dari vector dokumen seperti pada tabel 5.12, sedangkan nilai TFIDF query didapatkan dari vector query seperti pada tabel 5.15. Berikut ini adalah contoh perhitungan inner product antar Query0 dengan D0 dan Query0 dengan D6

$Query0D0 = (Query0_{anak} * D0_{anak}) + (Query0_{bantu} * D0_{bantu}) + (Query0_{dapat} * D0_{dapat}) + (Query0_{kip} * D0_{kip}) + (Query0_{sekolah} * D0_{sekolah}) = (0{,}098 * 0{,}043) + (0{,}183 * 0) + (0{,}113 * 0) + (0{,}094 * 0.041) + (0{,}123 * 0) = 0{,}004 + 0{,}004 = \mathbf{0.008}$

Tabel 5.21 Inner Product Dengan Pembobotan TFIDF

Term	D0	D1	D2	D3	D4	…
anak	**0.004**	0.000	**0.011**	**0.004**	**0.010**	**…**
dapat	0.000	**0.010**	0.000	**0.006**	**0.014**	**…**
kip	**0.004**	**0.014**	**0.010**	0.000	0.000	…
	0.008	0.024	0.021	0.010	0.024	…

b) Panjang Vektor Query Dengan Pembobotan TFIDF

Magnitude merupakan panjang vector, pada sub bab ini didiskusikan hasil perhitungan panjang vector atau magnitude daripada query. Tabel 5.22 adalah panjang vector query yang didapatkan dari hasil kuadrat bobot TFIDF setiap term pada query (Q1).

Tabel 5.22 Perhitungan Panjang Vektor Query

Term	Q0	$Q0^2$
anak	0.098	0.010
bantu	0.183	0.033
dapat	0.113	0.013
kip	0.094	0.009
sekolah	0.123	0.015
…	…	…
		0.145
		0.380

|Q| =

=

= = = **0,380**

c) Panjang Vektor Documents Dengan Pembobotan TFIDF

Perhitungan panjang vector documens (Q) sama halnya dengan perhitungan panjang vector query (Q). Tabel 5.23 merupakan tabel vector dokumen dengan pembobotan TFIDF dilakukan perhitungan panjang vektor sehingga hasilnya dapat dilihat pada

Tabel 5.24 Perhitungan Panjang Vektor Dokumen Dengan Pembobotan TFIDF

Term	$D0^2$	$D1^2$	$D2^2$	$D3^2$	$D4^2$	…
anak	**0.002**	0.000	**0.013**	**0.002**	**0.011**	**…**
bantu	0.000	0.000	0.000	0.000	0.000	**…**
dapat	0.000	**0.008**	0.000	**0.002**	**0.015**	**…**
kip	**0.002**	**0.021**	**0.012**	0.000	0.000	…
sekolah	0.000	0.000	0.000	0.000	0.000	…
…	…	…	…	…	…	…
	0.140	**0.165**	**0.455**	**0.252**	**0.173**	
	0.374	**0.406**	**0.675**	**0.502**	**0.416**	

Setelah dilakukan perhitungan panjang vector Q dan panjang vector D, maka Magnitude |Q| tertera pada tabel 5.22 dan Magnitude |D| pada tabel 5.24 kemudian kedua tabel tersebut dilakukan perhitungan perkalian |Q| * |D| seperti pada contoh perhitungan berikut

$|Q_0D_0| = Q_0 * D_0 = 0{,}380 * 0{,}374 = 0{,}142$

$|Q_0D_1| = Q_0 * D_1 = 0{,}380 * 0.406 = 0{,}154$

$|Q_0D_2| = Q_0 * D_2 = 0{,}380 * 0.675 = 0{,}256$

$|Q_0D_3| = Q_0 * D_3 = 0{,}380 * 0.502 = 0{,}191$

Dan seterusnya menghasilkan nilai magnitude seperti tertera pada tabel 5.25

Tabel 5.25 Perkalian magnitude |Q| dengan |D|

Documents	D0	D1	D2	D3	D4	…
Magnitude	0.142	0.154	0.256	0.191	0.158	…

Tabel 5.26 Hasil Perhitungan Cosine Similarity TFIDF

Documents	D0	D1	D2	D3	D4	…
Inner Product	0.008	0.024	0.021	0.010	0.024	…
Magnitude	0.142	0.154	0.256	0.191	0.158	…
Cosine Similarity	**0.056**	**0.156**	**0.082**	**0.052**	**0.152**	**…**

Rumus Cosine Similarity menggunakan persamaan (2.7) dimana hasilnya adalah perkalian dot vector Q dan D dibagi dengan cross product antara |Q| dan |D|. Artinya pada proses selanjutnya adalah kalkulasi nilai akhir cosine similarity tersaji pada tabel 2.25 berikut

Sebagai contoh, dapat dilakukan perhitungan Cosine Similarity berbasis TFIDF (CoSimTFIDF) pada D0 seperti pada perhitungan dibawah ini

Selanjutnya berdasarkan pada tabel 5.27 tersebut dilakukan pengurutan (sorting) nilai Cosine Similarity dari nilai terbesar sampai nilai terkecil. Sehingga didapatkan rekomendasi dokumen dengan nilai cosine similarity paling besar yaitu D5 dengan nilai 0.463 atau memiliki kemiripan dokumen **46,3%** terhadap Query (Q0) seperti yang tertera pada tabel 5.28. Pada tabel tersebut disajikan 10 (sepuluh) data ranking teratas

Tabel 5.28 Hasil Kalkulasi Cosine Similarity TFIDF

Document	Magnitude	InnerProduct	CosineSimilarity
D5	**0.156**	**0.072**	**0.463**
D9	0.165	0.064	0.390
D6	0.117	0.044	0.382
D113	0.112	0.039	0.348
D125	0.109	0.036	0.330
…	…	…	…

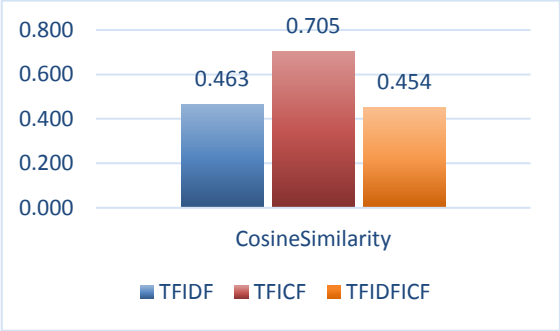
Tabel 5.29 Hasil Kalkulasi Cosine Similarity dengan TFICF

Document	Magnitude	InnerProduct	CosineSimilarity
D5	**0.064**	**0.045**	**0.705**
D9	0.052	0.036	0.692
D113	0.044	0.025	0.555
D6	0.034	0.018	0.536
D51	0.037	0.019	0.524
...	**...**	**...**	**...**

Tabel 5.30 Hasil Kalkulasi Cosine Similarity dengan TFIDFICF

Document	Magnitude	InnerProduct	CosineSimilarity
D5	**0.048**	**0.022**	**0.454**
D113	0.035	0.013	0.371
D9	0.059	0.020	0.337
D6	0.044	0.013	0.304
...	...	...	...

d) Perbandingan Hasil Cosine Similarity Berdasarkan Term Weighting



Gambar 5.4 Grafik Perbandingan Cosine Similarity Berdasarkan Skema Pembobotan

Setelah dilakukan serangkaian perhituangan manual cosine similarity terhadap ketiga pembobotan TFIDF, TFICF, dan TFIDFICF dapat diketahui hasil cosine similarity diantaranya dibandingkan dalam gambar 5.4

e) Eksperimen Variasi Preprocessing

Pada ekperimen ini bertujuan untuk mengevaluasi kinerja term weighting Class Indexing Based (TFIDFCF) dibandingkan dengan kinerja term weighting TFIDF, dan TFICF. Dataset yang digunakan adalah Dataset200, Dengan rasio perbandingan data training 75% dan data testing 25%. Menggunakan enam kategori kemudian diberikan label *(class)* sebagai nama lain *(alias)* ditunjukkan pada Tabel berikut:

Tabel 5.32 Confusion Matrix CoSimTFIDF Dengan Stemming

	C1	C2	C3	C4	C5	C6	TP	FP	TN	FN
C1	18	0	0	0	0	0	18	4	3	0
C2	2	0	0	0	0	0	0	0	23	2
C3	1	0	0	0	0	0	0	0	24	1
C4	0	0	0	3	0	0	3	0	22	0
C5	1	0	0	0	0	0	0	0	24	1
C6	0	0	0	0	0	0	0	0	25	0

Didapatkan Nilai True Positif terbesar adalah pada "C2", berikutya adalah "C3" dan nilai TP terkecil adalah "C1"

Dari tabel 5.32 tersebut dapat dievaluasi Akurasi, Precision, Recall dan F-Measure berdasarkan pada persamaan *Precision* (2.9), *Recall* (2.11), *F-Measure* (2.12) dan *Accuracy* (2.13). Berikutnya perlu dihitung nilai akurasi dari tabel confusion matrix, sehingga dilakukan perhitungan sebagai berikut

Sebagai contoh dapat dilakukan evaluasi terhadap *C1*, sehingga dapat dilakukan perhitungan sel berikut:

Dari serangkaian perhitungan Precision, R dan F-Measure pada setiap label kelas, maka didap daftar evaluasi setiap kelas sebagai berikut seperti tertera pada tabel 5.33

Tabel 5.33 Hasil Evaluasi Setiap Label

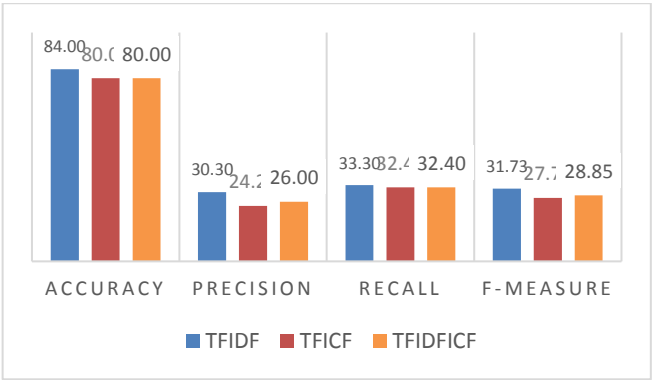
	Precision	Recall	F-Measure
C1	0.82	1.00	0.90
C2	0	0	0
C3	0	0	0
C4	1.00	1.00	1.00
C5	0	0	0
C6	0	0	0
Macro Average	0.30	0.33	0.32

Gambar 5.4 dibentuk berdasarkan nilai cosine similarity terbesar yang terdapat tabel 5.28, tabel 5.29 dan 5.30 didapatkan bahwa hasil rekomendasi dokumen berdasarkan nilai Cosine Similarity dengan skema pembobotan TFIDF, TFICF, dan TFIDFICF adalah dokumen D5 dengan hasil nilai cosine terbesar adalah 0,705 atau 70,5% berdasarkan pembobotan TFICF. Berikut adalah hasil ekperimen berdasarkan pengujian dengan variasi preprocessing stemming dan tanpa stemming. Menggunakan Dataset200 dan rasio data latih 75% dan data uji 25%. Evaluasi yang digunakan adalah Accuracy (A), Precision (P), Recall (R) dan F1-Measure (F1) menggunakan model macro average, ini digunakan mengingat hasil klasifikasi multi kelas *(multi class)*.

Tabel 5.39 Tabel Skenario Pengujian Variasi Preprocessing

Skenario		Jumlah **D**ata		Hasil Evaluasi		
Term Weighting	***Latih***	***Uji***	***A (%)***	***P (%)***	***R (%)***	***F1 (%)***
TFIDF	***75***	***25***	**84.00**	**30.30**	**33.30**	**31.73**
TFICF	***75***	***25***	80.00	24.20	32.40	27.71
TFIDFICF	***75***	***25***	80.00	26.00	32.40	28.85
TFIDF	***90***	***31***	46.15	17.71	19.82	18.71
TFICF	***90***	***31***	**58.06**	**31.88**	**21.13**	**25.42**
TFICFICF	***90***	***31***	45.16	18.14	17.21	17.66

Rangkuman evaluasi tertera pada tabel 5.39 tersebut merupakan agregasi berdasarkan hasil evaluasi tabel 5.32, 5.33, 5.34, 5.35, 5.36, 5.37 dan 5.38.



Gambar 5.5 Grafik evaluasi pengujian dengan proses stemming (dalam persen)

f) Pengujian Variasi Bobot Fitur

Tabel 5.40 Hasil Pengujian Bobot Fitur TFIDF

Evaluasi	*Jumlah Dataset*					
	25	**50**	**100**	**200**	**300**	**400**
A	50.00	25.00	69.23	**84.00**	60.53	63.33
P	25.00	6.25	18.89	**30.30**	17.36	21.22
R	25.00	25.00	18.89	**33.33**	18.89	21.15
F	40.00	10.00	18.89	**31,70**	18.00	18.47

Tabel 5.41 Hasil Pengujian Bobot Fitur TFICF

Evaluasi	*Jumlah Dataset*					
	25	**50**	**100**	**200**	**300**	**400**
A	50.00	25.00	76.92	**80.00**	71.05	61.67
P	25.00	6.25	12.82	24.17	**28.24**	13.57
R	25.00	25.00	16.67	**32.41**	20.37	19.96
F	**40.00**	10.00	14.49	27.68	19.72	16.00

Tabel 5.42 Hasil Pengujian Bobot Fitur TFIDFICF

Evaluasi	*Jumlah Dataset*					
	25	**50**	**100**	**200**	**300**	**400**
A	50.00	25.00	69.23	**80.00**	60.53	63.33
P	25.00	6.25	12.50	**25.99**	22.80	19.52
R	25.00	25.00	15.00	**32.41**	19.63	21.15
F	**40.00**	10.00	13.64	28.7	18.86	18.86

5. Hasil dan Analisis Proses Klasifikasi KNN

a) Akurasi Proses Klasifikasi Berdasarkan Nilai k

Ekperimen menggunakan Dataset200 dengan proses preprocessing menggunakan Stemming dan fitur CoSimTFIDF.

Tabel 5.43 Akurasi (%) KNN Berdasarkan Nilai k

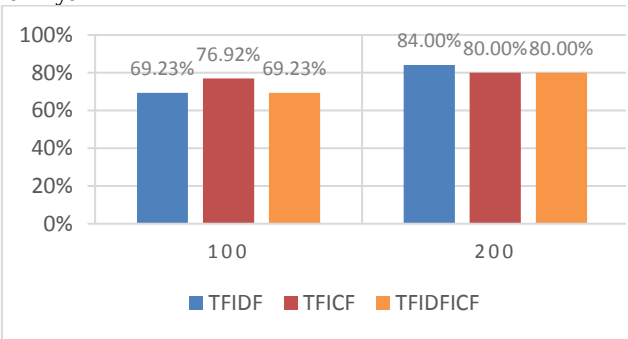
Nilai k / Rasio	1	2	3	4	5
25/75	75.71	80.00	78.57	80.00	80.00
75/25	83.33	83.33	83.33	83.33	83.33
40/60	67.86	80.36	82.14	80.36	82.14
60/40	76.32	**84.21**	**84.21**	**84.21**	**84.21**

b) Hasil Perbandingan Akurasi KNN Berdasarkan Fitur Dan Dataset

Ekperimen dilakukan dengan menentukan dataset yang digunakan yaitu Dataset100 dan Dataset200. Dengan rasio perbanding antara data training 75% dan data testing 25% (75/25). Nilai k yang digunakan adalah k=5. Pada pengujian pertama fitur yang digunakan hanya fitur Cosine Similarity yang berbasis term weighting TFIDF, kemudian pada pengujian berikutnya digunakan fitur Cosine Similarity berbasis TFICF, berikutnya CoSimTFIDFICF. Hasil nilai akurasi dari pengujian tertera pada tabel berikut

Tabel 5.44 Hasil Ekperimen variasi Dataset dan Term Weighting (Dalam %)

Hasil akurasi terbaik yang telah dicapai adalah menggunakan fitur Cosine Similarity berbasis TFIDF (CoSimTFIDF) yaitu 84% pada Dataset200 meningkat 4% dari kedua fitur Cosine Similarity TFICF dan TFIDFICF. Sedangkan pada Dataset100 diperoleh nilai akurasi terbaik dengan menggunakan fitur CoSimTFICF yaitu 76,92% meningkat sekitar 6% dari kedua fitur lainnya



Gambar 5.8 Hasil Perbandingan Akurasi Berdasarkan Dataset dan Fitur

c) Akurasi Pengujian Rasio Data Latih dan Data Uji

Pada eksperimen ini menggunakan Dataset200 dengan variasi proses preprocessing menggunakan stemming dan tanpa stemming. Sebagaimana yang telah ditemukan pada tabel 5.43 dimana nilai K yang memiliki hasil optimal adalah k=5, maka ditetapkan pada pengujian ini klasifikasi K-NN menggunakan nilai k=5. Rasio perbandingan data latih dan data uji yang bervariasi guna mendapatkan hasil yang bervariasi. Berikut ini merupakan hasil pengujian akurasi berdasarkan rasio data dan fitur pada tabel 5.45

Tabel 5.45 Akurasi Pada variasi Term Weighting

RASIO (%)	AKURASI (%)		
	CoSimTFIDF	**CoSimTFICF**	**CoSimTFIDFICF**
25:27	**66.67**	**66.67**	**66.67**
75:25	**84.00**	80.00	80.00
40:60	66.67	**78.33**	71.67
60:40	60.00	**75.00**	70.00
25:27	54.50	**64.86**	60.36
75:25	54.05	**55.41**	**55.41**
40:60	52.81	**60.67**	**60.67**
60:40	55.46	**57.98**	**57.98**

Berdasarkan hasil tersebut ditemukan bahwa hasil akurasi optimal dengan preprocessing Stemming dan hasil terbaik dari semua fitur adalah rasio data latih 75% dan data uji 25% pada fitur Cosine Similarity berbasis term weighting TFIDF yaitu 84%. Kemudian fitur CoSimTFICF dengan rasio data latih 40% dan data uji 60%

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Dalam hasil pengujian yang telah dilaksanakan, ditemukan bahwa skema *term weighting* TFIDF memiliki pengaruh yang signifikan terhadap akurasi klasifikasi. Terutama pada pengujian dengan menggunakan preprocessing stemming hasil akurasinya adalah 35% lebih baik daripada preprocessing tanpa stemming. Pengujian dengan variasi proses stemming menggunakan fitur *Cosine Similarity* berbasis TFIDF (CoSimTFIDF) ini menghasilkan performa algoritma K-NN terbaik yaitu akurasi 84%, kemudian diukur tingkat presisi 30,3%, recall 33,3%, dan f-measure 31,73%. Pengujian tersebut mempekerjakan 75 data latih dan 25 data uji dari total jumlah data anggota dataset sebanyak 200 data (Dataset200). Fitur ini sangat cocok dengan preprocessing stemming.

Algoritma pengklasifikasi diketahui menggunakan metode klasifikasi *K-Nearest Neighbor*. Berdasarkan bukti berupa data hasil pengujian berdasarkan nilai k untuk mengukur akurasi. Hasil yang terbaik untuk 100 data latih dan uji dengan variasi rasio data latih dan data uji yang berbeda-beda adalah nilai k=5. Nilai akurasi dengan rasio data latih 75% dan data uji 25% adalah 83,3%, rasio 25:75 (dalam persen) adalah 80%, rasio 40:60 adalah 82,14%. Dan nilai akurasi terbaik diperoleh dengan rasio 60:40 yaitu 84,21%. Berdasarkan hasil tersebut ditemukan bahwa hasil akurasi optimal dengan preprocessing Stemming dan hasil terbaik dari semua fitur adalah rasio data latih 75% dan data uji 25% pada fitur Cosine Similarity berbasis term weighting TFIDF yaitu 84%. Kemudian fitur CoSimTFICF dengan rasio data latih 40% dan data uji 60%

Metode K-NN digunakan sebagai metode klasifikasi dalam penelitian ini, seperti pada sifat yang dimiliki K-NN metode ini akan bertambah akurasinya jika data training atau pola dokumen laporan yang dimiliki semakin bervariasi. Selain itu metode ini memiliki konsistensi yang kuat, dengan cara mencari kasus dengan menghitung kedekatan antara *query* (q)

dengan *documents* (q) berdasarkan nilai k yang dimiliki. Hasil percobaan nilai k = 1,2,3,4, dan 5 dapat diketahui dari 200 dokumen laporan dengan 75 data latih dan 25 data uji maka diperoleh nilai akurasi yang hampir sama. Nilai k = 5 memiliki tingkat akurasi yang tinggi yaitu 84,12% , jadi dalam penelitian ini penjurusan menggunakan metode K-NN ditetapkan nilai k yang dipakai adalah 5.

B. Saran

Beberapa hal yang dapat dikembangkan untuk penelitian selanjutnya dalam lingkup yang sama antara lain: **1)** Sebaiknya dilakukan penambahan variasi preprocessing yaitu stopword list berbeda bahasa misalkan Bahasa Sunda, Basaha Jawa, Bahasa Slang/gaul dan sebagainya. Variasi preprocessing berikutnya adalah dengan menambahkan filter seperti pengenalan sinonim (synonym recognizion), POS Tagger, NER (Name Entity Recognition) dan lain sebagainya. Sesuai dengan sifat yang dimiliki metode K-NN yaitu semakin banyak pola kasus maka akurasi klasifikasi akan bertambah. 2) Sebaiknya dipekerjakan teknik klastering seperti K-Means untuk pengelompokan data dan digunakan teknik-teknik untuk mengoptimasi nilai k pada algoritma K-NN. 3) Sebaiknya juga dapat digunakan teknik Cross Validation untuk mendapatkan rasio data latih dan data uji K-NN yang proporsional. 4) Sebaiknya digunakan teknik SVD (Singular Value Decomposition) untuk mengurangi dimensi data. 5) Penelitian ini menggunakan algoritma Machine Learning terawasi (supervised) berbasis Instance Based, dapat dikembangkan algoritma Machine Learning terawasi yang lain misalnya SVM. 6) Penelitian ini dilakukan dengan menggunakan data sekunder . Kasus kemiripan teks atau dokumen sering dimanfaatkan untuk plagiasi, sentiment analysis, text emotion detection, spam detection, opinion mining, essay scoring, dan sebagainya. 7) Sebaiknya dilakukan penelitian lebih lanjut untuk tentang data pengaduan "LAPOR!" dan skema pembobotan kata yang lain

VI. DAFTAR PUSTAKA

[1] A. Sofyan And S. Santosa, "Text Mining Untuk Klasifikasi Pengaduan Pada Sistem Lapor Menggunakan Metode C4.5 Berbasis Forward Selection," *Cyberku J.*, Vol. 12, No. 1, Pp. 8–8, 2016.

[2] I. Surjandari, "Application of Text Mining for Classification of Textual Reports: A Study of Indonesia's National Complaint Handling System," in *6th International Conference on Industrial Engineering and Operations Management (IEOM 2016)*, Kuala Lumpur, Malaysia.

[3] A. Fauzan and M. L. Khodra, "Automatic multilabel categorization using learning to rank framework for complaint text on Bandung government," in *2014 International Conference of Advanced Informatics: Concept, Theory and Application (ICAICTA)*, 2014, pp. 28–33.

[4] S. Tjandra, A. A. P. Warsito, and J. P. Sugiono, "Determining citizen complaints to the appropriate government departments using KNN algorithm," in *2015 13th International Conference on ICT and Knowledge Engineering (ICT Knowledge Engineering 2015)*, 2015, pp. 1–4.

[5] W. H. Gomaa and A. A. Fahmy, "A Survey of Text Similarity Approaches," *Int. J. Comput. Appl.*, vol. 68, no. 13, pp. 13–18, 2013.

[6] M. A. Rosid, G. Gunawan, and E. Pramana, "Centroid Based Classifier With TF – IDF – ICF for Classfication of Student's Complaint at Appliation E-Complaint in Muhammadiyah University of Sidoarjo," *J. Electr. Electron. Eng.-UMSIDA*, vol. 1, no. 1, pp. 17–24, Feb. 2016.

[7] R. R. M. Putri, R. Y. Herlambang, and R. C. Wihandika, "Implementasi Metode K-Nearest Neighbour Dengan Pembobotan TF.IDF.ICF Untuk Kategorisasi Ide Kreatif Pada Perusahaan," *J. Teknol. Inf. Dan Ilmu Komput.*, vol. 4, no. 2, pp. 97–103, May 2017.

[8] C. F. Suharno, M. A. Fauzi, and R. S. Perdana, "Klasifikasi Teks Bahasa Indonesia Pada Dokumen Pengaduan Sambat Online Menggunakan Metode K-Nearest Neighbors (K-NN) dan Chi-Square," *J. Pengemb. Teknol. Inf. Dan Ilmu Komput. Vol 1 No 10 2017*, Jul. 2017.

[9] N. H. A. Sari, M. A. Fauzi, and P. P. Adikara, "Klasifikasi Dokumen Sambat Online Menggunakan Metode K-Nearest Neighbor dan Features Selection Berbasis Categorical Proportional Difference," *J. Pengemb. Teknol. Inf. Dan Ilmu Komput. Vol 2 No 8 2018*, Oct. 2017.

[10] A. A. Prasanti, M. A. Fauzi, and M. T. Furqon, "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 2018*, Aug. 2017.

[11] S. Dong and Z. Wang, "Evaluating service quality in insurance customer complaint handling throught text categorization," in *2015 International Conference on Logistics, Informatics and Service Sciences (LISS)*, 2015, pp. 1–5.

[12] D. Wang and H. Zhang, "Inverse-Category-Frequency based supervised term weighting scheme for text categorization," *J. Inf. Sci. Eng.*, vol. 29, no. 2, pp. 209–225, Dec. 2010.

[13] F. Ren and M. G. Sohrab, "Class-indexing-based term weighting for automatic text classification," *Inf. Sci.*, vol. 236, pp. 109–125, Jul. 2013.