Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK) p-ISSN: 2355-7699
Vol. 4, No. 2, Juni 2017, hlm. 97-103 e-ISSN: 2528-6579

IMPLEMENTASI METODE K-NEAREST NEIGHBOUR DENGAN PEMBOBOTAN TF.IDF.ICF UNTUK KATEGORISASI IDE KREATIF PADA PERUSAHAAN

Romario Yudo Herlambang[1], Rekyan Regasari Mardi Putri[2], Randy Cahya Wihandika[3]

[123]Fakultas Ilmu Komputer, Universitas Brawijaya
Email: [1]romarioyudo@gmail.com, [2]rekyan.rmp@ub.ac.id, [3]rendicahya@ub.ac.id

(Naskah masuk: 4 Februari 2017, diterima untuk diterbitkan: 7 Mei 2017)

Abstrak

Ide kreatif/inovasi merupakan hal yang dibutuhkan perusahaan dalam pengembangan sebuah individu, kelompok ataupun perusahaan pada teknologi seperti pada masa ini. Pengembangan ide kreatif berpengaruh pada peningkatan kinerja perusahaan. Pada kebanyakan kasus, pengelompokan ide tersebut harus dikelompokkan dengan kecocokan tema yang diusung untuk mempermudah proses pencarian. Oleh sebab itu dibutuhkan suatu sistem yang mampu bekerja secara otomatis untuk mengelompokkan ide tersebut. Kemunginan salah satu teknik pembobotan yang digunakan adalah dengan meggunakan *TF.IDF.ICF*, yang telah mengalami pengembangan dari metode sebelumnya. *TF.IDF.ICF* tidak dapat digunakan sendiri melainkan harus ada metode perhitungan jarak seperti *Cosine Similarity* dan metode klasifikasi lain seperti *KNN* dapat dipakai ke semua atribut. Aplikasi ini nantinya akan diterapkan pada perusahaan PJB Paiton sebagai studi kasus dan ide kreatif yang dikategorikan, dituliskan dalam Bahasa Indonesia. Aplikasi ini akan melakukan beberapa tahap pemrosesan seperti *tokenizing* yaitu pemisahan kalimat menjadi tiap kata, *filtering* yang merupakan penghapusan *stopwords*, *stemming*, *cosine similarity* dan *KNN* yang masing-masing metode digunakan untuk perhitungan jarak dan proses perhitungan klasifikasi Dari hasil pengujian yang telah dilakukan, sistem mampu menghasilkan akurasi terbaik sebesar 93% menggunakan dengan nilai *k* sebesar 1 menggunakan presentase data uji sebanyak 50 akan menghasilkan klasifikasi ideal.

Kata kunci: *ide, kelas, cosine, KNN.*

Abstract

Creative ide is one thing that needed by the company for group development or even the company itself. The development of creative ideas has a big influence on improving corporate performance. On most cases, the clasification of the idea must be grouped based on the similarity of the theme that submitted to simplify the searching process. Therefore we need a system that could work automatically to classify the idea. Probably, one weighting techniques that used is TF.IDF.ICF that already been developed from the method before. TF.IDF.ICF cant be used alone. there must be another method that used before, such as cosine similarity for distance calculation method and KNN for classification method in order TF.IDF.ICF can be used by all atributes. This application will be focused on the PJB company's creative idea and these ideas will be in indonesian language. This application will do a few processing steps such as, tokenizing for breaking sentence into words, filtering which is elimination of stopwords, stemming, cosine similarity, and KNN. each method used for distance calculation and classification calculation process. From the testing result that has been done,the system could produce the best accuracy as big as 93% by using the value of K as big as 1 using the precentage of test data as big 50 produce the ideal classification.

Keywords: *idea, class, cosine, KNN*

1. PENDAHULUAN

PT Pembangkitan Jawa Bali (PJB) adalah anak perusahaaan PT PLN (persero) yang didirikan pada tanggal 3 Oktober 1995, sedangkan Pembangkit Listrik Tenaga Uap (PLTU) Unit Pembangkitan Paiton ini berdiri pada 15 Maret 1993. Pada tanggal 3 Oktober 1995 terdapat rekstrukturisasi pada PT PLN (persero) dan melahirkan dua anak perusahaan yakni PT PLN Pembangkitan Tenaga Listrik Jawa Bali I dan II yang disebut PJB I dan II. Karena PT PJB anak perusahaan PT PLN maka PT PJB harus mengikuti aturan dari PT PLN yang salah satunya adalah kontes ide kreatif.

Kontes ide kreatif yang dikonteskan pada PT PLN Tiap Tahun. Maka dari itu PT PJB harus mengirimkan 5 ide kreatif pertahun yang akan ditandingkan, untuk hal itu dalam penerimaan berbagai ide dan solusi kreatif terkait berbagai permasalahan maupun peningkatan efisiensi kerja harus diklasifikasikan sesuai kategorinya. Kategori yang biasa ditandingkan untuk menemukan solusi-solusi masalah adalah T untuk Teknik dan NT untuk Non Teknik. Teknik adalah kategori yang membahas seputar mesin, alat berat, dll, sedangkan untuk Non Teknik adalah kategori yang ruang lingup

bahasannya umum, misal pengadaan jus buah pada pagi hari dan sore sebelum jam kerja habis. Masalah yang muncul adalah tingkat pengkategorian yang dimasukkan oleh peserta adalah terjadinya beberapa kesalahan dalam pengkategorian teks ide kreatif.

Permasalahan ini hampir sama dengan penelitian yang pernah dilakukan oleh Sriram dkk pada tahun 2010, yang melakukan klasifikasi teks dari *twitter* untuk meingkatkan penyaringan informasi. Sehingga, dalam mengatasi permasalahan kategori terhadap pengkategorian ide kreatif, peneliti mengusulkan proses klasifikasi dimana sistem dapat mengolah data teks menjadi kategori-kategori yang diinginkan. Salah satu metode dalam melakukan pemrosesan teks adalah *TF.IDF.ICF*.

TF.IDF.ICF salah satu metode pembobotan dalam mencari tingkat kemiripan teks. Metode ini tediri dari 3 komponen, *TF* untuk menghitung jumlah kata pada suatu dokumen, *ID*F untuk jumlah frekuensi tiap kata pada dokumen dan *ICF* adalah jumlah frequensi tiap kata berdasarkan kelas, lalu ketiganya akan dikalikan untuk membentuk *TF.IDF.ICF*. Berdasarkan konsep yang digunakan tersebut, dikembangkan pemodelan dokumen untuk melakukan pencarian terhadap dokumen yang dibutuhkan. Dokumen akan direpresentasikan sebagai vektor atau biasa disebut *Vector Space Model* dengan cara menghitung besarnya sudut yang terbentuk antara dua vektor dan kemudian diurutkan dari data yang memiliki besar sudut yang terkecil hingga yang terbesar yang menandakan urutan data hasil pengurutan dari yang paling relevan hingga yang tidak relevan yaitu dengan Cosine Similarity. Namun setelah data tersebut selesai dihitung dengan *Cosine Similarity* maka akan diklasifikasikan yang salah satu metode yang dapat diterapkan adalah metode *KNN*.

Berdasarkan latar belakang yang telah dijelaskan, maka pada penelitian ini akan mengimplementasikan *TF.IDF.ICF* pada pembobotan kata dan *KNN* pada klasifikasi untuk pengkategorian ide.

2. LANDASAN TEORI

2.1. *Text Mining*

Klasifikasi ide kreatif menjadi beberapa kategori adalah satu contoh aplikasi dari *text mining*. Beberapa contoh lain aplikasi dari *text mining* adalah *text summarization*, *text categorization*, *document clustering*, *language identification*, *ascribing authorship* (Mustafa. dkk. 2009). Text mining atau dengan sebutan lain seperti intelligent text analysis, text data mining, atau knowledge discovery in text secara sederhana dapat diartikan sebagai proses penemuan pola yang sebelumnya tidak terlihat pada dokumen teks atau sumber tertentu (Manning. dkk. 2009). Dalam pemrosesan teks/*text mining* biasanya awal dilakukannya dengan melakukan Preprocessing, beberapa diantaranya yaitu dengan Tokenizing atau pemisahan suatu dokumen/kalimat menjadi tiap kata/term, Filtering yaitu dengan penghapusan kata penghubung, kata sambung, dll dari hasil Tokenizing tiap dokumen, dan Stemming atau pemisahan imbuhan pada kata menjadi kata dasar. Lalu setelah proses diatas terjadi maka harus ada pembobotan yang dilakukan dengan metode *TF.IDF.ICF* yang lalu dihitung jaraknya dengan *Cosine Similarity* dan pengklasifikasian salah satunya adalah *KNN*.

2.2. *Preprocessing*

Preprocesing adalah proses awal yang dilakukan agar metode pembobotan kata dapat diterapkan, beberapa metode antara lain: *tokenizing*, *filtering* dan *stemming*.

2.2.1 *Tokenizing*

Proses pemecahan kalimat yang terdapat pada sebuah dokumen menjadi kata. Setiap kata yang didapat biasanya dalam *text mining* disebut dengan *term*/*token*. Metode ini adalah serangkaian metode dalam proses *preprocessing*.

2.2.2 *Filtering*

Filtering atau biasa disebut *stop-word removal* adalah proses pada *preprocessing* yang berguna untuk meghilangkan kata sambung, kata penghubung atau kata umum lainnya. *Filtering* ini berguna untuk mengurangi perhitungan dari kata yang tidak seharusnya tidak mewakili kelas apapun.

2.2.3. Metode *Stemming* Arifin dan Setiono

Algoritma *stemming* akan mengolah tiap kata yang dimasukkan untuk diproses dan menghasilkan kata dalam bentuk dasarnya. Langkah-langkah yang dilakukan dalam algoritma *stemming* bahasa Indonesia Arifin dan Setiono adalah sebagai berikut (Arifin. dkk. 2001) :

Pemeriksaan yang dilakukan pada keseluruan probabilitas bentuk kata. Setiap kata diasumsikan memiliki 2 awalan atau prefika dan 3 akhiran atau sufiks (Arifin. dkk. 2001). Sehingga memiliki bentuk sebagai berikut:

Prefiks 1 + Prefiks 2 + Kata dasar + Sufiks 3 + Sufiks 2 + Sufiks 1

Jika terdapat perbedaan rangkaian kata atau kekosongan pada susunan kata seperti diatas maka cukup ditambahkan x untuk prefix dan xx untuk sufiks pada bagian yang kosong (Arifin. dkk. 2001).

Pemotongan dilakukan dengan catatan AW mewakili awalan, AK adalah akhiran, dan KD untuk kata dasar secara berurutan sebagai berikut :

a. AW I, hasilnya tersimpan pada p1 (prefiks 1)
b. AW II, hasilnya tersimpan pada p2 (prefiks 2)

c. AK I, hasilnya tersimpan pada s1 (sufiks 1)
d. AK II, hasilnya tersimpan pada s2 (sufiks 2)
e. AK III, hasilnya tersimpan pada s3 (sufiks 3)

Pada setiap tahap pemotongan di atas diikuti pemeriksaan pada kamus apakah hasil pemotongan sudah berada dalam bentuk dasar. Jika pemeriksaan berhasil dilakukan maka proses dinyatakan selesai sehingga tidak perlu melanjutkan proses pemotongan imbuhan lainnya (Arifin. dkk. 2001). Contoh pemenggalan kata "membuatkannya" . Berikut Langkahnya :

a. Pengecekan kata dalam kamus
   Ya : Success
   Tidak : lakukan pemotongan AW I
   Kata = buatkannya
b. Pengecekan kata dalam kamus
   Ya : Success
   Tidak : lakukan pemotongan AW II
   Kata = buatkannya
c. Pengecekan kata dalam kamus
   Ya : Success
   Tidak : lakukan pemotongan AK I
   Kata = buatkan
d. Pengecekan kata dalam kamus
   Ya : Success
   Tidak : lakukan pemotongan AK II
   Kata = buat
e. Pengecekan kata dalam kamus
   Ya : Success
   Tidak : lakukan pemotongan AK III. Dalam hal ini AK III tidak ada, sehingga kata tidak diubah.
   Kata = main
f. Pengecekan kata dalam kamus
   Ya : Success
   Tidak : "Kata tidak ditemukan"

Jika sampai pada pemotongan AK III belum juga ditemukan kata yang sama dalam kamus, maka perlu dilakukan kombinasi (Arifin. dkk. 2001). KD yang dihasilkan dikombinasikan dengan imbuhan-imbuhannya dalam 12 konfigurasi seperti dibawah ini:

a. KD
b. KD + AK III
c. KD + AK III + AK II
d. KD + AK III + AK II + AK I
e. AW I + AW II + KD
f. AW I + AW II + KD + AK III
g. AW I + AW II + KD + AK III + AK II
h. AW I + AW II + KD + AK III + AK II + AK I
i. AW II + KD
j. AW II + KD + AK III
k. AW II + KD + AK III + AK II
l. AW II + KD + AK III + AK II + AK I

Sebenarnya kombinasi a, b, c, d, h, dan l sudah diperiksa pada tahap sebelumnya, karena kombinasi ini merupakan hasil pemotongan bertahap (Arifin. dkk. 2001). Dengan demikian, kombinasi yang masih perlu dilakukan yaitu (e, f, g, i, j, dan k).

Pemeriksaan 12 kombinasi dibutuhkan, sebab adanya fenomena *overstemming* pada algoritma pemotongan imbuhan. Kelemahannya berdampak pada pemotongan bagian kata yang sebenarnya adalah milik kata dasar itu sendiri dan kebetulan mirip dengan salah satu jenis imbuhan yang sudah ada. Dengan 12 kombinasi itu, pemotongan yang sudah terlanjur tersebut dapat dikembalikan sesuai strukturnya (Arifin. dkk. 2001).

2.3. TF.IDF.ICF

Nilai *Term Frequenzy (TF)* didapat dari jumlah setiap term dalam setiap dokumen yang ada.

Inverse *Document Frequency (IDF)* adalah proses yang dilakukan saat pembobotan kata. Bertujuan untuk mencari bobot pada setiap kata. *IDF* didefinisikan sebagai

$$Idf_i = \boldsymbol{log}\left(\frac{\boldsymbol{n_d}}{\boldsymbol{df_i}}\right) \quad (1)$$

Keterangan :

Idf_i = Inverse document frequency untuk tiap term/kata.

log = Operasi matematika yang merupakan kebalikan dari operasi pangkat.

n_d = Jumlah dokumen yang ada pada data training.

df_i = Jumlah dokumen yang memiliki kata tersebut.

Inverse Class Frequency (ICF) adalah proses yang dilakukan saat pembobotan kata. Bertujuan untuk mencari bobot pada setiap kata. *ICF* didefinisikan sebagai

$$Icf_i = \boldsymbol{log}\left(\frac{\boldsymbol{n_c}}{\boldsymbol{cf_i}}\right) \quad (2)$$

Keterangan :

Icf_i = *Inverse class frequency* untuk tiap term/kata.

log = Operasi matematika yang merupakan kebalikan dari operasi pangkat.

n_c = Jumlah kelas yang ada pada data training.

cf_i = Jumlah kelas yang memiliki kata tersebut.

dimana *cfi* merupakan frekuensi dokumen dari term i atau sama dengan jumlah dokumen yang mengandung term i dan n adalah total dokumen di dalam database. *Log* digunakan untuk memperkecil pengaruh relative untuk tf_{ij}.

Bobot wij dihitung menggunakan ukuran *TF.IDF.ICF* (*term frequency-inversed document frequency-inversed class frequency*) didefinisikan sebagai

$$w_{ij} = tf_{ij} \; x \; idf_i \; x \; icf_i \quad (3)$$

Keterangan :

w_{ij} = Hasil perkalian dari tf_{ij}, idf_i dan icf_i.

tf_{ij} = Frekuensi kemunculan kata/term pada dokumen.

idf_i = *Inverse document frequency* untuk tiap *term*/kata.

icf_i = *Inverse class frequency* untuk tiap *term*/kata.

2.4 VSM

Vector Space Model (VSM) mempresentasikan setiap dokumen yang terdapat pada *TF.IDF.ICF* ke dalam vektor multidimensi. Dimensi dari vektor sesuai dengan jumlah setiap dokumen dan jumlah jarak antara dokumen dengan tiap *query* tersebut untuk dibentuk pada suatu ruang vektor.

2.5 *Cosine Similarity*

Salah satu ukuran kemiripan teks yang popular digunakan pada VSM untuk pencarian dokumen adalah *cosine similarity* (Krzysztof J. Cios. Dkk. 2007).

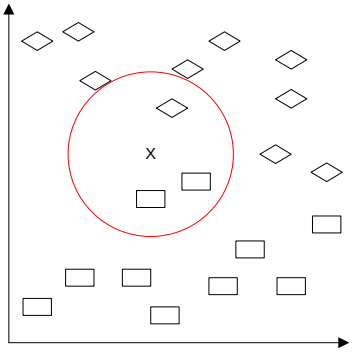
Konsep dari *cosine similarity* yaitu menghitung nilai cosinus sudut antara dua vektor yaitu jika diberikan dokumen yang diwakili oleh vektor d_j dan query q, dan term t yang diekstrak dari database, maka nilai *cosine similarity* didefinisikan sebagai

$$cos(\vec{d}_j, \vec{d}_k) = \vec{d}_j \cdot \vec{d}_k \quad (4)$$

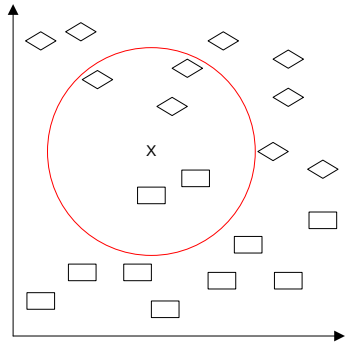
Sudut yang merentang antara vektor d_j dan query q akan menghasilkan sudut yang jika semakin kecil sudut diantara kedua vektor d_j dan query q, maka akan semakin tinggi derajat kesamaan. Cosinus dari sudut tersebut merupakan koefisien yang dapat mewakili kemiripan antara vektor d_j dan query q.

2.6 KNN

Algoritma *K-Nearest Neighbour* (KNN) adalah sebuah metode klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. Data pembelajaran dideskripsikan dengan atribut *numeric* n-dimensi. Setiap data pembelajaran merepresentasikan sebuah titik, dalam ruang n-dimensi. Jika sebuah data query yang labelnya tidak diketahui dimasukkan, maka *K-Nearest Neighbour*nya akan mencari k buah data pembelajaran yang jaraknya paling dekat dengan data *query* dalam ruang n-dimensi. Jarak antar data *query* dengan data pembelajaran dihitung dengan cara mengukur jarak antara titik yang merepresentasikan data *query* dengan semua titik yang merepresentasikan data pembelajaran. Pada kasus ini menggunakan rumus *Cosine Similarity* sebagai rumus perhitungan jarak.



Gambar 1. Cara kerja metode KNN dengan k adalah 3



Gambar 2. Cara kerja metode KNN dengan k adalah 5

Seperti pada Gambar 1 dan Gambar 2 X adalah data latih yang sebagai pusat untuk mencari kemiripan sedangkan ▭ dan ◇ adalah data yang dicari kemiripan dengan data X, pada Gambar 1 dijelaskan bahwa jika menggunakan k = 3, maka kemiripan dari data X adalah yang ada pada lingkaran merah, hampir sama dengan Gambar 1, pada Gambar 2. didalam lingkaran adalah data yang memiliki kemiripan dengan data X namun dengan k = 5.

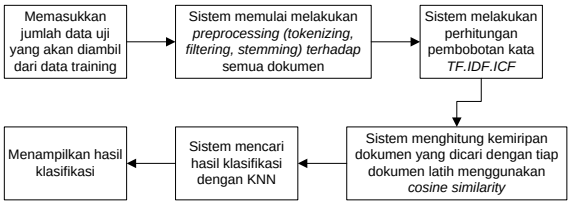
2.6 Standar Deviasi

Standar deviasi yaitu perbedaan nilai sampel terhadap rata-rata. Nilai sampel yakni sedikit dari jumlah keseluruhan objek yang diamati. Didefinisikan sebagai

$$\sqrt{\frac{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}{n(n-1)}} \quad (2.5)$$

3. PERANCANGAN

Penelitian ini merupakan salah satu bentuk penelitian implementatif, dimana penelitian ini akan menghasilkan program yang akan membantu mengklasifikasikan ide yang mirip.



Gambar 3. Diagram Blok Program

4. HASIL UJI COBA DAN ANALISA

Pada bab ini membahas tentang tahapan pengujian dan analisis dari hasil implementasi algoritma *TF.IDF.ICF* untuk pengklasifikasian ide kreatif menjadi kelas Teknik dan Non Teknik.

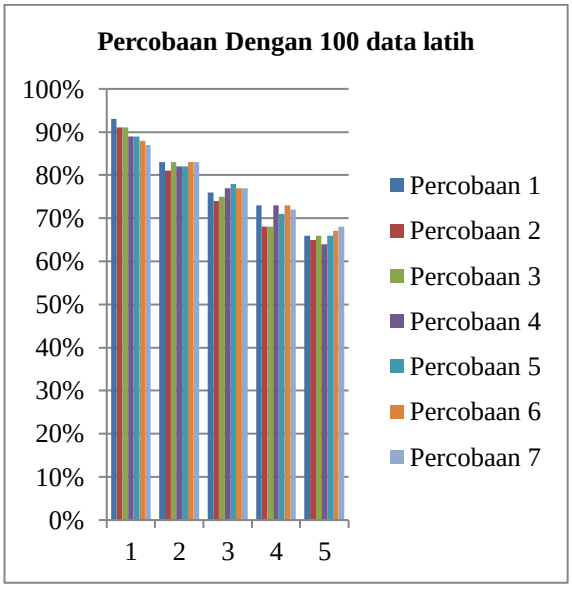
4.1 Hasil dan Analisis Pengaruh Variasi K Dan Jumlah Data Latih

Pada uji coba ini, dilakukan pengujian terhadap pengaruh data latih terhadap akurasi. Data yang diuji adalah data ide kreatif PT PJB UP Paiton tahun 2014 yang 50 data menjadi data uji yang dipilih secara acak pada awal percobaan sehingga data uji yang digunakan selalu dan sisa dari data uji akan menjadi data latih yang kesemua data latih. Setiap percobaan dengan jumlah data latih yang sama akan diambil jarak 10 data latih dengan percobaan sebelumnya atau melakukan irisan terhadap percobaan sebelumnya. Percobaan dilakukan sebanyak 27 kali dengan catatan untuk setiap percobaan menggunakan jumlah k sebanyak 1-5. Jumlah data latih berbeda, mulai dari 100 data latih dengan 7 kali percobaan, 80 data latih dengan 9 kali percobaan dan 60 data latih dengan 11 kali percobaan. Seperti dijelaskan pada Tabel 1.

Tabel 1. Tabel percobaan

Percobaan ke-	Jumlah data latih	Akurasi dalam persentase				
		Jumlah k				
		1	2	3	4	5
1	100	93	83	76	73	66
2		91	81	74	68	65
3		91	83	75	68	66
4		89	82	77	73	64
5		89	82	78	71	66
6		88	83	77	73	67
7		87	83	77	72	68

Uji pengaruh data latih yang pertama dilakukan dengan menggunakan 100 data latih dengan 7 percobaan awal dengan menggeser id sebanyak 10 id tiap kali percobaan pada Tabel 1 yang hasilnya bisa diilustrasikan dalam Gambar 4.



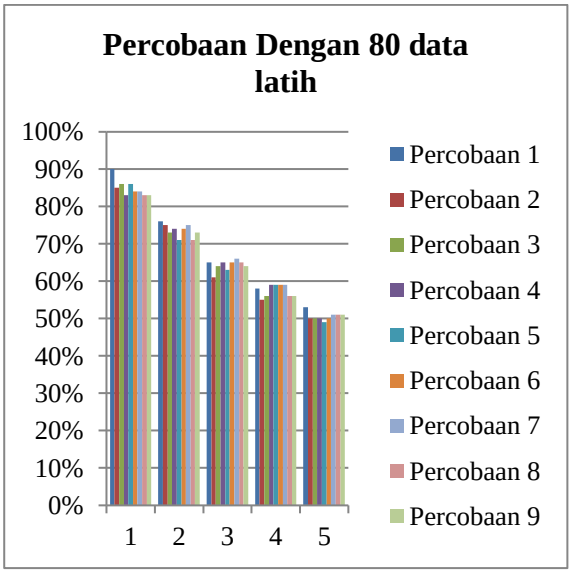
Gambar 4. Hasil pengujian dengan jumlah data latih sebanyak 100

Pada Gambar 4, percobaan dilakukan sebanyak 7 kali dengan menggunakan k mulai dari 1 – 5 pada setiap percobaan. Hasil uji coba pertama digambarkan dengan menggunakan garis Percobaan 1 dan hasil percobaan kedua digambarkan dengan menggunakan garis Percobaan 2 dan begitu seterusnya. Dengan semakin besar jumlah k maka akan semakin menurunkan tingkat akurasi. Dengan 100 data latih variasi data terlihat kecil dengan jarak akurasi (maks-min) terjauh adalah 6% dengan k = 1 dan k = 4. Pada k = 2 terjadi jarak akurasi(maks-min) terkecil yaitu 2%.

Tabel 2. Tabel percobaan

Percobaan ke-	Jumlah data latih	Akurasi dalam persentase				
		Jumlah k				
		1	2	3	4	5
1	80	90	76	65	58	53
2		85	75	61	55	50
3		86	73	64	56	50
4		83	74	65	59	50
5		86	71	63	59	49
6		84	74	65	59	50
7		84	75	66	59	51
8		83	71	65	56	51
9		83	73	64	56	51

Pada uji coba kedua dengan menggunakan 80 data latih dengan 9 percobaan mulai dari percobaan 1 - 9 pada Tabel 2 yang hasilnya bisa diilustrasikan dalam Gambar 5.



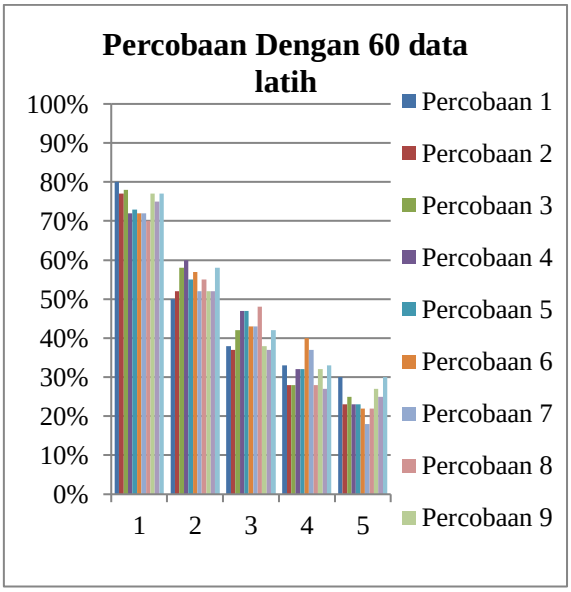
Gambar 5. Hasil pengujian dengan jumlah data latih sebanyak 80

Pada Gambar 5 uji coba dilakukan sebanyak 9 kali dengan menggunakan k mulai dari 1 – 5 pada setiap percobaan. Hasil uji coba kedua digambarkan dengan menggunakan garis Percobaan 1 dan hasil percobaan kedua digambarkan dengan menggunakan garis Percobaan 2 dan begitu seterusnya seperti pada percobaan dengan data latih sebanyak 80. Dengan semakin besar jumlah k maka akan semakin menurunkan tingkat akurasi. Dengan 80 data latih variasi data terlihat lumayan kecil dengan jarak akurasi (maks-min) terjauh adalah 7% dengan k = 1. Pada k = 4 dan 5 terjadi jarak akurasi(maks-min) terkecil yaitu 4%.

Tabel 3. Tabel percobaan

Percobaan ke-	Jumlah data latih	Akurasi dalam persentase				
		Jumlah k				
		1	2	3	4	5
1	60	80	50	38	33	30
2		77	52	37	28	23
3		78	58	42	28	25
4		72	60	47	32	23
5		73	55	47	32	23
6		72	57	43	40	22
7		72	52	43	37	18
8		70	55	48	28	22
9		77	52	38	32	27
10		75	52	37	27	25
11		77	58	42	33	30

Pada uji coba terakhir dengan menggunakan 60 data latih dengan 11 percobaan mulai dari percobaan 1 - 1 pada Tabel 6.3 yang hasilnya bisa diilustrasikan dalam Gambar 6.3.



Gambar 6. Hasil pengujian dengan jumlah data latih sebanyak 60

Pada Gambar 6 uji coba dilakukan sebanyak 11 kali dengan menggunakan k mulai dari 1 – 5 pada setiap percobaan. Hasil uji coba terakhir ini digambarkan dengan menggunakan garis Percobaan 1 dan hasil percobaan kedua digambarkan dengan menggunakan garis Percobaan 2 dan begitu seterusnya seperti pada percobaan dengan data latih sebanyak 60. Dengan semakin besar jumlah k maka akan semakin menurunkan tingkat akurasi. Dengan 60 data latih variasi data terlihat kecil dengan jarak akurasi(maks-min) terjauh adalah 13% dengan k = 4. Pada k = 1 dan 2 terjadi jarak akurasi(maks-min) kecil yaitu 10%.

Berdasarkan 3 pengujian pada Tabel 1, Tabel 2 dan Tabel 3, variabel yang memiliki pengaruh variasi data adalah jumlah data latih, semakin banyak data latih akan semakin kecil jarak variasi data. Mulai dari akurasi pada 60 data latih sampai 100 data latih selalu ada peningkatan akurasi, namun dengan keterbatasan data latih maka yang dapat dijadikan data latih hanya 100 data. Jumlah k juga berpengaruh terhadap akurasi yaitu jumlah k berbanding terbalik dengan akurasi. Semakin banyak jumlah k, maka akurasi semakin kecil. Untuk itu pengunaan klasifikasi KNN pada kasus ini sebaiknya tidak digunakan, hanya perlu 1NN saja sudah dapat diambil kesimpulan pasti mengenai klasifikasinya.

6.2 Pengujian Standar Deviasi

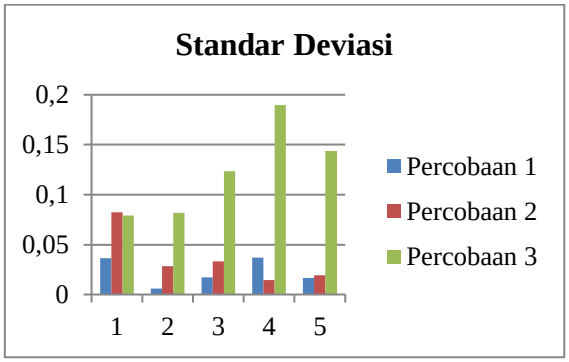
Pengujian standar deviasi dilakukan untuk mengetahui penyebaran data akurasi pada masing-masing k sehingga pada k yang memiliki akurasi terbaik.

Tabel 4. Tabel Standar Deviasi

Percobaan ke-	Jumlah data latih	Standar deviasi * 100				
		Jumlah k				
		1	2	3	4	5

1	100	3,65	0,62	1,72	3,69	1,68
2	80	8,21	2,85	3,30	1,48	1,95
3	60	7,93	8,19	12,37	18,95	14,39

Pada Tabel 4 adalah adalah standar deviasi tiap data latih dengan jumlah data latih sama dari Tabel 1, Tabel 2 dan Tabel 3 berdasar persamaan 5 dengan patokan jumlah k.



Gambar 7. Hasil pengujian standar deviasi

Hasilnya penyebaran data akurasi pada masing-masing k sehingga pada k yang memiliki akurasi terbaik yaitu k = 1, dinyatakan valid karena dengan variasi skenario uji coba, akurasi yang dihasilkan tidak memiliki persebaran data yang tinggi.

Tabel 5. Tabel Standar Deviasi

Percobaan ke-	Jumlah data latih	Standar deviasi dalam persentase				
		Jumlah k				
		1	2	3	4	5
1	100	2,087917	3,177666	4,690372	7,766456	5,92898

Pada Tabel 5 didapatkan standar deviasi dari semua percobaan dengan berpatokan pada jumlah k. Disimpulkan bahwa dengan jumlah k = 1 akan menghasilkan standar deviasi terendah, dan k = 4 adalah standar deviasi tertinggi. Hasil ini memperkuat bahwa dengan 1NN atau KNN dengan k = 1 sudah cukup untuk mendapat hasil maksimal.

5. KESIMPULAN

Berdasarkan hasil uji coba parameter *TF.IDF.ICF* pada permasalahan pengelompokan ide kreatif adalah sebagai berikut :

1. Algoritma *TF.IDF.ICF* dapat menyelesaikan permasalahan pada pengelompokan ide kreatif pada PT PJB UP Paiton, hasil akurasi yang didapat memiliki rata-rata kemiripan yang paling tinggi adalah 90% dengan k = 1.
2. Hasil parameter terbaik pada hasil pengujian ini adalah sebagai berikut.
   - Jumlah k : 1
   - Jumlah data latih : 100
   - Percobaan : 1
   - Jumlah data uji : 50
   - Akurasi rata-rata : 90%
3. Pada bab pengujian dijelaskan dengan k = 1 sudah bisa didapat standar deviasi terkecil, jika jumlah k bertambah standar deviasi akan semakin tingi. Standar deviasi pada variasi data dengan ketentuan sebagai berikut.
   - Jumlah k : 1
   - Jumlah data uji : 50
   - Standar deviasi : 2,087917

6. DAFTAR PUSTAKA

ARIFIN. A. Z. & NOVAN. S. A. 2001. Klasifikasi Dokumen Berita Kejadian Berbahasa Indonesia dengan Algoritma *Single Pass Clustering*. ITS, Surabaya

CIOS, K. J., PEDRYCZ, W., SWINIARSKI, R.W. & KURGAN, L.. 2007. Data *Mining A Knowledge Discovery Approach*. Springer.

MANNING, C., RAGHAVAN, P. & SCHÜTZE, H. 2009. An *Introduction to Information Retrieval. Cambridge University Press.*

MUSTAFA, A., AKBAR, A. & SULTAN, A. 2009. *Knowledge Discovery using Text Mining: A Programmable Implementation on Information Extraction and Categorization. International Journal of Multimedia and Ubiquitous Engineering Vol. 4, No. 2*, April, 2009.

SRIRAM. B., FUHRY, D., DEMIR, E., FERHATOSMANOGLU, H. & DEMIRBAS, M. 2010. *Short Text Classification in Twitter to Improve Information Filtering. International ACM SIGIR Conference on Research and Development in Information Retrieval*.