

Analisis Algoritma Fuzzy Logic dalam Pengklasifikasian Tugas Akhir

Analysis of Fuzzy Logic Algorithm in Final Task Classification

H Irmayanti

Program Studi Teknik Komputer, Fakultas Teknik dan Ilmu Komputer, Universitas Komputer Indonesia
Jl. Dipati Ukur No. 112 - 116, Bandung, Indonesia 40132

e-mail: haniirmayanti@email.unikom.ac.id

ABSTRACT – This study aims to determine how much the accuracy of the use of Fuzzy Logic Algorithm in the Final Task classification. The method used is Fuzzy K-Nearest Neighbor (KNN). The first step in the process is to do Text Preprocessing to find out what terms will be carried out for the next process, then weighting each term using the TF.IDF algorithm, after weighting, the terms of testing with the training term using Cosine Similarity are calculated, the next step the similarity value will be sorted and then some data will be taken from a number of neighbor values determined to calculate the KNN and see which categories appear more, the last is calculated the accuracy value seen from the comparison of training data and testing data. The final result of accuracy is 76.67%, thus Fuzzy KNN can be used to help the process of grouping research groups.

Keywords – Cosine Similarity; Fuzzy K-Nearest Neighbor; TF.IDF algorithm

ABSTRAK – Penelitian ini bertujuan untuk mengetahui berapa nilai keakuratan penggunaan Algoritma Fuzzy Logic dalam pengklasifikasian Tugas Akhir. Metode yang digunakan adalah Fuzzy K-Nearest Neighbour (KNN). Langkah proses yang dilakukan pertama yaitu melakukan Text Preprocessing untuk mengetahui term apa saja yang akan dilakukan untuk proses selanjutnya, kemudian dilakukan pembobotan setiap term dengan menggunakan algoritma TF.IDF, setelah didapatkan bobot, dihitung kemiripan term testing dengan term training menggunakan Cosine Similarity, Langkah selanjutnya nilai kemiripan akan diurutkan dan kemudian akan diambil beberapa data sejumlah nilai tetangga yang ditentukan untuk menghitung KNN dan melihat kategori mana yang kemunculannya banyak, terakhir dihitung nilai akurasi dilihat dari hasil perbandingan data training dan data testingnya. Hasil akhir akurasinya adalah 76,67%, dengan demikian Fuzzy KNN ini bisa digunakan untuk membantu proses pengelompokan kelompok penelitian.

Kata Kunci - Cosine Similarity; Fuzzy K-Nearest Neighbor; algoritma TF.IDF

1. PENDAHULUAN

Seorang mahasiswa untuk mendapatkan gelar sarjana harus membuat sebuah karya ilmiah yang dikenal sebagai Tugas Akhir (TA). Tugas Akhir (TA) adalah sebuah mata kuliah yang harus ditempuh oleh seorang mahasiswa menjelang akhir studinya. Mata kuliah ini berbentuk proyek mandiri yang dilakukan oleh mahasiswa di bawah bimbingan dosen pembimbing. Karya ilmiah yang dimaksud dapat berupa laporan ditulis sesuai dengan pedoman tugas akhir. Untuk tema penulisan karya ilmiah ini, tentunya disesuaikan dengan bidang studi atau program studinya masing-masing. Ada beberapa kelompok penelitian Tugas Akhir yang dimiliki program studi, yang bisa dipilih oleh mahasiswa.

Namun, terkadang mahasiswa sulit menentukan tulisan yang mereka buat, masuk ke dalam kelompok penelitian yang mana. Oleh karena itu dibutuhkan sebuah sistem untuk membantu proses klasifikasi tugas akhir dengan bantuan sebuah metode untuk proses klasifikasinya, hasil akhirnya nanti akan dianalisa keefektifan proses klasifikasi, apakah lebih efektif menggunakan sistem atau tidak. Maka dari itu tujuan penelitian ini yaitu mengetahui berapa nilai keakuratan penggunaan Algoritma Fuzzy Logic dalam pengklasifikasian Tugas Akhir

Ada beberapa penelitian terdahulu yang ada hubungannya dengan penelitian ini. Pertama, penelitian yang membahas penggunaan metode *fuzzy inference system* (FIS) mamdani dalam pemilihan peminatan mahasiswa untuk tugas

akhir.[1] Kedua, penelitian yang membahas mengenai Identifikasi Konten E-Government berdasarkan dokumen yang sudah diklasifikasikan menggunakan Naïve Bayes Classification.[2] Ketiga Penelitian ini membahas mengenai penggunaan text mining classification untuk rekomendasi Dosen Pembimbing Tugas Akhir Program Studi Sistem Informasi.[3] Perbedaan penelitian ini dengan penelitian terdahulu diantaranya adalah, pertama yaitu pada penelitian terdahulu menggunakan metode mamdani, sedangkan penelitian ini menggunakan metode K-Nearest Neighbour (KNN), kedua penelitian terdahulu untuk mengidentifikasi konten, sedangkan penelitian ini dilakukan untuk mengetahui seberapa efektif metode ini untuk mengklasifikasikan kelompok penelitian, ketiga penelitian terdahulu text klasifikasi digunakan untuk penentuan dosen pembimbing pada penelitian untuk menentukan kelompok penelitian.

Metode yang digunakan dalam penelitian ini yaitu K-Nearest Neighbour. Fuzzy KNN dipilih karena masalah pemilihan Kelompok Penelitian ini dianggap sebagai masalah klasifikasi yang penuh kesamaran. Sebelum data-data itu dikelompokkan atau diklasifikasikan menggunakan algoritma fuzzy logic, dibutuhkan proses pengenalan text terlebih dahulu yang disebut *text preprocessing*. Dalam proses *text preprocessing* nanti dilakukan beberapa tahapan atau proses seperti, *tokenizing*, *stemming*, *tagging* dan *analysing*. [4] Untuk mendapatkan kata-kata dasar dilakukan proses *stemming*. Hasil dari proses *text preprocessing* tersebut, merupakan poin-poin utama dalam suatu dokumen, yang selanjutnya bisa digunakan untuk proses pengklasifikasian atau *clusterization*

2. METODE DAN BAHAN

Metodenya adalah sebagai berikut:

2.1. Pengumpulan Data

Peneliti melakukan pengumpulan data. Data yang akan digunakan adalah kumpulan Judul Tugas Akhir mahasiswa. Datanya terdapat pada table 1 sebagai berikut :

Tabel 1. Data Awal

No	Data Awal
1	Rancang Bangun Drone Quadrotor Pendeteksi Dan Pemadam Api Dini
2	Pengembangan Alat Pengenal Huruf Dan Angka Dwibahasa Untuk Usia Dini
3	Rancang Bangun Alat Pembuat Pupuk Cair Organik Menggunakan Sendor Teknologi Film Tebal

No	Data Awal
4	Rancang Bangun Alat Pijat Relaksasi Pada Tangan Dan Mendeteksi Tangan Otomatis Berbasis Arduino
5	Pembuatan Sistem Klasifikasi Penyakit Jantung Menggunakan Metode Naive Bayes Berbasis Android
...	...
30	Absensi Sidik Jari <i>Portable</i> Berbasis Website Dan Raspberry

2.2. Text Processing

Text Preprocessing adalah tahapan seleksi dokumen yang akan dipakai untuk analisa berikutnya[3]. Prosesnya adalah sebagai berikut :

- 1) *case folding*
 Tidak semua dokumen teks konsisten dalam penggunaan huruf kapital. Oleh karena itu, peran *Case Folding* dibutuhkan dalam mengkonversi keseluruhan teks dalam dokumen menjadi suatu bentuk standar (biasanya huruf kecil atau *lowercase*).
- 2) *Tokenizing*
 Tahap *Tokenizing* adalah tahap pemotongan string input berdasarkan tiap kata yang menyusunnya. Tokenisasi secara garis besar memecah sekumpulan karakter dalam suatu teks ke dalam satuan kata, bagaimana membedakan karakter-karakter tertentu yang dapat diperlakukan sebagai pemisah kata atau bukan. Sebagai contoh karakter *whitespace*, seperti *enter*, tabulasi, spasi dianggap sebagai pemisah kata.
- 3) *Stemming*
 Membuat indeks supaya dokumen dikenali. Teknik *Stemming* diperlukan selain untuk memperkecil jumlah indeks yang berbeda dari suatu dokumen, juga untuk melakukan pengelompokan kata-kata lain yang memiliki kata dasar dan arti yang serupa namun memiliki bentuk atau *form* yang berbeda karena mendapatkan imbuhan yang berbeda.

2.3. Algoritma TF.IDF

Algoritma ini digunakan untuk memberikan bobot pada setiap kata (*term*). *Term* bisa berbentuk frasa, hasil indexing atau kata. Supaya dapat mengetahui konteks dari dokumen kata diberikan indicator *term weight*. Metode ini dilakukan setelah tahapan *stemming* dan *stopword removal* dilakukan. Perhitungan terhadap nilai atau bobot suatu kata (*term*) pada dokumen dilakukan ditahap ini.[5]

1. Term Frequency (TF)

TF (*Term Frequency*) adalah jumlah tampilnya term

dalam dokumen. Bobot akan semakin besar apabila nilai kemunculan TF tinggi.

Pada *Term Frequency* (TF), terdapat beberapa jenis formula yang dapat digunakan :

- TF biner (*binary TF*), hanya memperhatikan apakah suatu kata atau *term* ada atau tidak dalam dokumen, jika ada diberi nilai satu (1), jika tidak diberi nilai nol (0).
- TF murni (*raw TF*), nilai TF diberikan berdasarkan jumlah kemunculan suatu term di dokumen. Contohnya, jika muncul 5 kali maka kata tersebut akan bernilai 5
- TF logaritmik, hal ini untuk menghindari dominansi dokumen yang mengandung sedikit term dalam query, namun mempunyai frekuensi yang tinggi.

$$tf = 1 + \log(tf) \quad (1)$$

- TF normalisasi, menggunakan perbandingan antara frekuensi sebuah kata dengan jumlah keseluruhan kata pada dokumen.

$$tf = 0.5 + 0.5 \times \left(\frac{tf}{\max tf} \right) \quad (2)$$

2. Inverse Document Frequency (IDF)

Menghitung *Inverse Document Frequency* (idf) dapat dilakukan dengan cara :

$$idf_j = \log\left(\frac{D}{df_j}\right) \quad (3)$$

Keterangan :

D : jumlah dokumen dalam df

j : jumlah dokumen yang ada term t_j

Formula untuk TF-IDF adalah perkalian dari perhitungan TF dengan idf :

$$w_{ij} = tf_{ij} \times idf_j \quad (4)$$

$$w_{ij} = tf_{ij} \times \log\left(\frac{D}{df_j}\right) \quad (5)$$

dimana :

w_{ij} : bobot term t_j terhadap dokumen d_i

tf_{ij} : jumlah kemunculan term t_j dalam dokumen d_i

D : jumlah semua dokumen yang ada dalam database

df_j : jumlah dokumen yang mengandung term t_j (minimal ada satu kata yaitu term t_j)

Mengacu pada formula diatas, berapapun nilai tf_{ij} , apabila $D = df_j$ maka hasilnya 0 (nol) untuk perhitungan idf. Oleh karena tambahkan nilai 1 pada idf, perhitungannya adalah berikut ini :

$$w_{ij} = tf_{ij} \times \left(\log\left(\frac{D}{df_j}\right) + 1 \right) \quad (6)$$

C. Cosine Similarity

Cosine similarity merupakan metode untuk menghitung kesamaan antara dua buah objek yang dinyatakan dalam dua buah vector dengan menggunakan *keywords* (kata kunci) dari sebuah dokumen sebagai ukuran.[6]

Formulanya sebagai berikut:

$$\cos\text{Som}(d_j, q_k) = \frac{\sum_{i=1}^n (td_{ij} \times tq_{ik})}{\sqrt{\sum_{i=1}^n td_{ij}^2 \times \sum_{i=1}^n tq_{ik}^2}} \quad (7)$$

Keterangan :

$\cos\text{Sim}(d_j, q_k)$: tingkat kesamaan dokumen dengan query tertentu.

td_{ij} : term ke-i dalam vector untuk dokumen ke-j

tq_{ik} : term ke-i dalam vector untuk query ke-k

n : jumlah term yang unik dalam data set

Berikut langkah-langkah perhitungan manual algoritma *Cosine Similarity* [7]:

- Ditentukan terlebih dahulu masing-masing query, yaitu query dari jawaban (D), *query* dari key jawaban (Q) dan gabungan keduanya (Queries).
- Ketiga *query* tersebut dihilangkan *stoplist* atau simbol-simbol yang tidak mempengaruhi penilaian, seperti tanda titik, tanda koma, tanda seru, dan sebagainya.
- Ketiga *query* tersebut dihilangkan *stopwords* atau kata-kata umum yang lazim digunakan dalam suatu *query*, seperti "dan", "jika", "di", "namun", "tetapi", dan sebagainya.
- Dihitung nilai *term frequency query* jawaban dan *query key* jawaban terhadap *queries*. Jadi perhitungan *term* di *query* jawaban dan *query key* jawaban merujuk pada *term* yang terdapat dalam *queries*.
- Dihitung nilai *document frequency* (n) atau banyaknya *file* (N) yang memiliki suatu *term* untuk tiap *term* dalam *queries*.
- Dihitung nilai *inverse document frequency* dengan rumus :

$$\log\left(\frac{N}{n}\right) + 1 \quad (8)$$

- Dikalikan nilai *term frequency* dengan nilai *inverse document frequency* tiap term dalam Q maupun D.
- Dihitung hasil perkalian skalar masing-masing query jawaban terhadap *query key* jawaban. Hasil perkalian dari setiap jawaban dengan *query* dijumlahkan (sesuai pembilang pada rumus di atas).
- Dihitung hasil perkalian vektor tiap *query key* jawaban dan *query* jawaban.

10. Dihitung nilai *Cosine similarity* (nilai vektor beda antara D terhadap Q) dengan rumus :

$$sim(d, q) = \frac{\sum_{k=1}^i (weight_{dk} \cdot weight_{qk})}{\sqrt{\sum_{k=1}^i (weight_{dk}^2 \cdot weight_{qk}^2)}} \quad (9)$$

2.4. Fuzzy K-Nearest Neighbour

Fuzzy K-NN berisi penggabungan teknik Fuzzy dan K-NN. Dalam Metode ini akan diprediksi kelas yang diikuti secara tegas oleh data uji berdasarkan perbandingan K terdekat. Fuzzy ini akan menggunakan nilai keanggotaan menjadi jarak vektor dari keanggotaan tetangga di kelas dengan KNN. KNN digunakan untuk menghilangkan keraguan dalam klasifikasi [8] Berikut ini formula yang digunakan untuk menghitung sebuah proses sebelum menghitung KNN :

$$u_{ij} = \begin{cases} 0,51 + \left(\frac{n_j}{n}\right) * 0,49, & \text{jika } j = 1 \\ \left(\frac{n_j}{n}\right) * 0,49, & \text{jika } j \neq 1 \end{cases} \quad (10)$$

keterangan :

n_j = Jumlah anggota kelas j pada suatu data latih n

n = Jumlah data latih yang digunakan

j = Kelas data

Selanjutnya menghitung nilai keanggotaan masing-masing kelas dengan persamaan :

$$u_i(x) = \frac{\sum_{j=1}^k u_{ij} \left(\|x - x_j\|^{-2/(m-1)} \right)}{\sum_{j=1}^k \left(\|x - x_j\|^{-2/(m-1)} \right)} \quad (11)$$

keterangan :

u_{ij} = nilai keanggotaan fuzzy pada contoh pengujian (x, x_j)

k = nilai tetangga terdekat

j = variabel data keanggotaan data uji

m = bobot pangkat yang besarnya $m > 1$

3. HASIL DAN PEMBAHASAN

Hasil dan pembahasan dari penelitian ini dapat dilihat dari langkah-langkah penelitian berikut.

3.1. Text Preprocessing

Pada tahapan ini akan dilakukan case folding, tokenizing, filtering dan stemming.

1. Case Folding

Hasil *Case Folding* terdapat pada tabel 2.

Tabel 2. Hasil Case Folding

No	Data Awal	Hasil Case Folding
1	Rancang Bangun Drone Quadrotor Pendeteksi Dan Pemadam Api Dini	rancang bangun drone quadrotor pendeteksi dan pemadam api dini
2	Pengembangan Alat Pengenal Huruf Dan Angka Dwibahasa Untuk Usia Dini	pengembangan alat pengenal huruf dan angka dwibahasa untuk usia dini
3	Rancang Bangun Alat Pembuat Pupuk Cair Organik Menggunakan Sendor Teknologi Film Tebal	rancang bangun alat pembuat pupuk cair organik menggunakan sendor teknologi film tebal
4	Rancang Bangun Alat Pijat Relaksasi Pada Tangan Dan Mendeteksi Tangan Otomatis Berbasis Arduino	rancang bangun alat pijat relaksasi pada tangan dan mendeteksi tangan otomatis berbasis arduino
5	Pembuatan Sistem Klasifikasi Penyakit Jantung Menggunakan Metode Naive Bayes Berbasis Android	pembuatan sistem klasifikasi penyakit jantung menggunakan metode naive bayes berbasis android
...
...
30	Absensi Sidik Jari PorTabel Berbasis Website Dan Raspberry	absensi sidik jari porTabel berbasis website dan raspberry

2. Tokenizing

Hasil Tokenizing terdapat pada tabel 3.

Tabel 3. Hasil Tokenizing

absensi	dwibahasa
alat	edc
algoritma	eigenface
android	elektronik
angka	estimasi
antrian	..
api	..
aplikasi	..
arduino	wajah
autonomous	wall
babakan	web

3. *Filtering*

Hasil *Filtering* terdapat pada tabel 4.

Tabel 4. Hasil *Filtering*

Stopword		
ada	di	dalam
adalah	dimana	berdasarkan
akan	ini	dengan
apa	itu	disertai
atau	yang	kemampuan
dan	oleh	mempergunakan
dari	yaitu	menerapkan

4. *Stemming*

Hasil *Stemming* terdapat pada tabel 5.

Tabel 5. Hasil *Stemming*

Hasil <i>Stemming</i>	
makanan	makan+an
pencarian	pen-cari-an
pendeteksi	pen+deteksi
mendeteksi	men+deteksi
pengadaan	peng+ada+an
menentukan	me+tentu+kan
...	...
...	...
pemotong	pe+potong

A. **Pembobotan**

Selanjutnya dilakukan pembobotan terhadap term hasil text preprocessing. Hasilnya adalah sebagai berikut :

1. TF-IDF

Data pada tabel 6 menunjukkan hasil penentuan nilai TF-IDF :

Tabel 6. Nilai TF-IDF

Term	TF						DF
	D1	D2	D3	D4	...	D30	
absensi	0	0	0	0	...	1	1 2,477
Alat	0	1	1	1	...	0	10 1,477
algoritma	0	0	0	0	...	0	1 2,477
android	0	0	0	0	...	0	2 2,176
Angka	0	1	0	0	...	0	1 2,477
...
website	0	0	0	0	...	1	2 2,176

2. Bobot

Data pada tabel 4 merupakan hasil perhitungan bobot.

Tabel 7. Bobot

Term	Wdt					
	D1	D2	D3	D4	...	D30
absensi	0	0	0	0	...	2,477
Alat	0	1,477	1,477	1,477	...	0
algoritma	0	0	0	0	...	0
android	0	0	0	0	...	0
Angka	0	2,477	0	0	...	0
...
website	0	0	0	0	...	2,176

3.2. **Cosine Similarity**

Langkah *cosine similarity* adalah sebagai berikut :

- Menentukan nilai $Q(\text{Data Testing})$
 Berikut ini tercantum dalam tabel 8 adalah nilai $Q(\text{Data Testing})$:

Tabel 8. Perbandingan Q dengan dokumen Training

Term	Q	D1	D2	D3	D4	...	D30
absensi	0	0	0	0	0	...	6
Alat	0	0	2	2	2	...	0
algoritma	0	0	0	0	0	...	0
android	5	0	0	0	0	...	0
Angka	0	0	6	0	0	...	0
...
website	0	0	0	0	0	...	5

- Pembobotan Dokumen *Testing* (Q)

Data dalam tabel 9 merupakan hasil pembobotan dokumen *testing* (Q)

Tabel 9. Hasil Pembobotan Dokumen *Testing*(Q)

Term	Q	D1	D2	D3	D4	...	D30
absensi	0	0	0	0	0	...	2,477
Alat	0	0	1,48	1,48	1,48	...	0
algoritma	0	0	0	0	0	...	0
android	2,18	0	0	0	0	...	0
Angka	0	0	2,48	0	0	...	0
...
website	0	0	0	0	0	...	2,176

- Perkalian skalar tiap D terhadap Q

Pada tabel 10 terdapat data perkalian skalar tiap D dengan Q.

Tabel 10. Perkalian Skalar

Term	Q	D1	D2	D3	D4	...	D30
absensi	0	0	0	0	0	...	0
Alat	0	0	0	0	0	...	0
algoritma	0	0	0	0	0	...	0
android	0	0	0	0	0	...	0
Angka	0	0	0	0	0	...	0
...
website	0	0	0	0	0		2,176

d. Perkalian Vektor

Hasil Perkalian Vektor terdapat pada tabel 11.

Tabel 11. Hasil Perkalian Vektor

Term	Q	D1	D2	D3	D4	...	D30
absensi	0	0	0	0	0	...	6
Alat	0	0	2	2	2	...	0
algoritma	0	0	0	0	0	...	0
android	5	0	0	0	0	...	0
Angka	0	0	6	0	0	...	0
...	
website	0	0	0	0	0	...	5
Jumlah	49,5	39	43,8	53,43	53,6	...	33,79
Panjang vector	7,04	6,24	6,61	7,31	7,32	...	5,81

e. Nilai Cosine Similarity

Data pada tabel 12 merupakan data *cosine similarity* dan kemudian data diurutkan tingkat kemiripannya, hasilnya pada tabel 13.

Tabel 12. Cosine Similarity

D1	D2	D3	D4	...	D30
0,000	0,000	0,000	0,031	...	0,040
0%	0%	0%	3,1%	..	4%

Tabel 13. Tingkat Kemiripan

D24	D29	D8	D18	...	D25
0,243	0,197	0,116	0,102	...	0
243%	19,7%	11,6%	10,2%	...	0

3.3. K-Nearest Neighbour (KNN)

Setelah melakukan pembobotan, kemudian dihitung vector untuk mencari nilai kemiripan dokumen, selanjutnya akan dilakukan analisa dengan KNN, dengan cara menentukan jumlah tetangga (k) yang nantinya akan dibandingkan tingkat kemiripannya

dengan dokumen. Sebagai contoh yang diambil adalah k=5. Maka hasilnya terdapat pada tabel 14.

Tabel 14. K-Nearest Neighbour

Dokumen	D24	D29	D9	D18	D15
Class	SC	SC	IK	SC	IK
Cosine Similarity	24,3%	19,7%	11,6%	10,2%	8,7%

Berdasarkan hasil pada tabel 14, kemunculan paling banyak adalah class SC. Berarti Data testing 5 termasuk kelompok penelitian SC (*Soft Computing*).

Akurasi

Selanjutnya ditahap terakhir ini, untuk mengetahui berapa nilai akurasi dari penentuan kelompok keahlian dengan menggunakan Fuzzy KNN, dilakukan perhitungan akurasi pada tabel 15 berikut:

Tabel 15 Hasil Perbandingan Pengelompokan Kelompok Keahlian

NO	Dokumen	CLASS	Hasil KNN	Keterangan
1	D1	IK	IK	Sesuai
2	D2	SC	SC	Sesuai
3	D3	IK	SC	Tidak Sesuai
4	D4	IK	IK	Sesuai
5	D5	SC	SC	Sesuai
6	D6	IK	IK	Sesuai
7	D7	IK	IK	Sesuai
8	D8	IK	IK	Sesuai
9	D9	IK	IK	Sesuai
10	D10	IK	IK	Sesuai
11	D11	IK	IK	Sesuai
12	D12	IK	IK	Sesuai
13	D13	IK	IK	Sesuai
14	D14	IK	IK	Sesuai
15	D15	SC	SC	Sesuai
16	D16	SC	IK	Tidak Sesuai
17	D17	SC	IK	Tidak Sesuai
18	D18	SC	IK	Tidak Sesuai
19	D19	IK	IK	Sesuai
20	D20	IK	IK	Sesuai
21	D21	IK	IK	Sesuai
22	D22	IK	IK	Sesuai
23	D23	SC	IK	Tidak Sesuai
24	D24	SC	SC	Sesuai
25	D25	IK	IK	Sesuai
26	D26	SC	SC	Sesuai
27	D27	SC	SC	Sesuai
28	D28	SC	IK	Tidak Sesuai
29	D29	SC	IK	Tidak Sesuai
30	D30	SC	SC	Sesuai

Berdasarkan Tabel 15 terdapat 23 Dokumen yang sesuai, berarti perhitungannya adalah sebagai berikut :

$$Accuracy = \frac{23}{30} \times 100\% = 76,67\%$$

Jadi nilai keakuratannya adalah 76,67%, oleh karena itu Fuzzy KNN ini bisa digunakan untuk membantu proses pengelompokan kelompok keahlian.

4. KESIMPULAN

Setelah dilakukan analisis terhadap penggunaan Algoritma Fuzzy Logic K-Nearest Neighbor untuk pengklasifikasian Kelompok Keahlian Tugas Akhir, didapatkan hasil akurasi sebesar 76,67%. Maka Algoritma Fuzzy K-Nearest Neighbour dapat digunakan untuk proses klasifikasi Kelompok Keahlian Tugas Akhir.

UCAPAN TERIMA KASIH

Saya berterima kasih kepada pihak Lembaga Penelitian dan Pengabdian Kepada Masyarakat (LPPM) Universitas Komputer Indonesia yang sudah membantu membiayai penelitian ini. Terima kasih juga kepada semua pihak yang sudah membantu dalam proses penelitian ini.

DAFTAR PUSTAKA

- [1] Yulmaini, "Penggunaan Metode Fuzzy Inference System (Fis) Mamdani Dalam Pemilihan Peminatan Mahasiswa Untuk Tugas Akhir," *J. Inform.*, vol. 15, no. June 2015, 2015.
- [2] A. P. Wijaya, H. A. Santoso, J. T. Informatika, U. Dian, and N. Semarang, "Naive Bayes Classification pada Klasifikasi Dokumen Untuk Identifikasi Konten E-Government," vol. 1, no. 1, pp. 48-55, 1978.
- [3] Z. Efendi, "Text Mining Classification Sebagai Rekomendasi Dosen Pembimbing Tugas Akhir Program Studi Sistem Informasi," pp. 18-19, 2017.
- [4] N. L. Ratniasih, M. Sudarma, N. Gunantara, and A. A. U. Sistem, "Penerapan Text Mining dalam Spam Filtering untuk Aplikasi Chat," vol. 16, no. 3, 2017.
- [5] A. H. Harahap, "Implementasi Metode Terms Frequency-Inverse Document Frequency (TF-IDF) dan Maximum Marginal Relevance untuk Monitoring Diskusi Online," vol. 13, no. 2, pp. 151-159, 2016.
- [6] R. T. Wahyuni, D. Prastiyanto, and E. Suprpto, "Penerapan Algoritma Cosine Similarity dan Pembobotan TF-IDF pada Sistem Klasifikasi Dokumen Skripsi," vol. 9, no. 1, 2017.
- [7] P. Tidur, "Deteksi Kesamaan Dokumen Dengan Cosine Similarity," 2017. [Online]. Available:

<http://payahtidur.blogspot.com/2017/09/deteksi-kesamaan-dokumen-dengan-cosine-similarity.html>.

- [8] S. D. Nugraha, R. R. M. Putri, and R. C. Wihandika, "Penerapan Fuzzy K-Nearest Neighbor (FK-NN) Dalam Menentukan Status Gizi Balita," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 1, no. 9, pp. 925-932, 2018.