KATEGORISASI DOKUMEN TEXT MENGGUNAKAN METODE K-NEAREST NEIGHBOR PADA DOKUMEN TUGAS AKHIR UNIVERITAS MUHAMMADIYAH JEMBER

(1) Ja'far Shodiq 1110651085 (2) Lutfi Ali Muharom, S.Si, M.Si
Jurusan Teknik Informatika Universitas Muhammadiyah Jember

Abstrak

Kategorisasi dokumen tugas akhir sangatlah penting untuk dapat memudahkan pengguna dalam pencarian dokumen dengan mudah. Dengan dikategorikannya dokumen-dokumen abstraksi tugas akhir, pembaca dapat menangkap ide utama dari sebuah dokumen tanpa harus membaca keseluruhan dokumen, sehingga dapat menentukan topik tugas akhir sesuai dengan bidang minat yang ada. Pada penilitian ini dilakukan penerapan metode K-Nearest Neighbor (KNN) untuk kategorisasi dokumen tugas akhir dengan mengacu pada abstraksi tugas akhir yang akan melalui proses tahapan *text mining* yang nantinya akan diklasifikasikan oleh KNN. Program aplikasi kategorisasi tugas akhir ini dibangun dengan data latih dari abstraksi tugas akhir Universitas Muhammadiyah Jember yang telah diklasifikasikan sebelumnya dan data uji berasal dari abstraksi tugas akhir yang belum diketahui kategorinya. Aplikasi yang dibuat mampu mengklasifikasikan data abstraksi tugas akhir dengan presentase keberhasilan 82,2% dengan jumlah data latih 85 dan 15 data uji.

Kata Kunci : Kategorisasi dokumen, *K-Nearest Neighbor*, *text mining*

I. PENDAHULUAN

1.1 Latar Belakang Masalah

Tugas Akhir merupakan salah satu syarat wajib untuk mendapat gelar sarjana. Tugas Akhir di beberapa program studi memiliki beberapa kategori, salah satunya pada prodi Teknik Informatika S1 di Universitas Muhammadiyah Jember. Semakin lama dokumen Tugas Akhir semakin bertambah banyak. Banyaknya dokumen yang tersedia mendorong manusia untuk mencari cara untuk mendapatkan informasi dan dokumen yang tepat dalam waktu singkat. Apabila jumlah dokumen yang tersedia sangat besar, proses pencarian secara manual akan menghabiskan waktu dan tenaga. Sehingga akan lebih mudah jika dokumen tersebut sudah diketahui sesuai dengan kategorinya masing-masing. Penyusunan dokumen sesuai dengan kategori yang ada sangat diperlukan untuk menyesuaikan dengan kategori yang terkandung pada dokumen tersebut secara otomatis.

Salah satu proses untuk menyelesaikan masalah tersebut adalah dengan proses *Text Minning*. *Text Mining* merupakan proses mengelompokkan suatu teks ke dalam suatu kategori tertentu. Algoritma kategorisasi teks saat ini telah banyak berkembang, antara lain: *Support Vector Machines (SVM)*, *Naive Bayessian*, *Decision Tree*, dan *K-Nearest Neighbor* (K-NN) (G. Toker and Ö. Kırmemiş). Algoritma K-NN termasuk suatu algoritma yang sederhana, namun cukup efektif dalam melakukan kategorisasi teks. Bukan hanya mudah dan efisien, sifat dari algoritma *K-Nearest Neighbor* sendiri bersifat *self-learning*, dimana algoritma ini dapat mempelajari struktur data yang ada dan mengkategorikan dirinya sendiri (Zhang , 2009).

Penelitian ini akan difokuskan pada kategorisasi Tugas Akhir melalui informasi dalam dokumen abstraksi tugas akhir. Penelitian dilakukan dengan mengklasifikasi data dokumen teks abstraksi ke kategori bidang minat yang ada. Sehingga diharapkan mampu membuat sistem kategorisasi dokumen tugas akhir menggunakan metode *k-nearest neighbor.*

1.2 Rumusan Masalah

Seberapa tinggi kinerja sistem kategorisasi yang didapat dari algoritma *K-Nearest Neighbor* pada kasus pengkategorisasian dokumen tugas akhir yang diwakili oleh abstraksi tugas akhir.

1.3 Tujuan Penilitian

Mengaplikasikan algoritma *K-Nearest Neigbor* untuk mengukur akurasi dalam mengkategorisasikan dokumen tugas akhir kedalam Bidang Minat dengan mengambil studi kasus untuk program studi Teknik Informatika.

II. TINJAUAN PUSTAKA

2.1 Dokumen Teks

Dokumen adalah sebuah tulisan yang memuat informasi. Biasanya, dokumen ditulis di kertas dan informasinya ditulis memakai tinta baik memakai tangan atau memakai media elektronik (J. Samodra, 2009) . Dokumen teks termasuk kedalam jenis data yang tidak terstruktur. Untuk itu sebelum dilakukan proses kategorisasi teks perlu dilakukan proses transformasi yang dapat mengubah teks-teks menjadi bentuk yang lebih efisien dan lebih siap untuk proses selanjutnya.

2.2 Preprosesing Dokumen

Preprosesing dokumen merupakan tahapan dari *text mining* yang harus dilakukan jika ingin menambang informasi berupa teks. *Text Mining* merupakan suatu proses yang bertujuan untuk menemukan informasi atau tren terbaru yang sebelumnya tidak terungkap, dengan memproses dan menganalisa data dalam jumlah besar (R. Rakhmat Sani). Dalam menganalisa sebagian atau keseluruhan *unstructured text, text mining* mencoba untuk mengasosiasikan satu bagian teks dengan yang lainnya berdasarkan aturan-aturan tertentu. Selain itu *text mining* juga diartikan sebagai kegiatan menambang data yang berupa teks atau dokumen, dengan tujuan mencari kata-kata yang dapat mewakili apa yang ada dalam dokumen sehingga dapat dilakukan analisa keterhubungan antar dokumen (Kusrini, 2009). Tahap-tahap preprosesing yang dilakukan secara umum dalam *text mining* secara umum adalah:

1. Tahap *tokenizing* adalah tahap pemisahan rangkaian *term* menjadi bentuk token atau potongan kata tunggal.
2. Tahap *filtering* adalah tahap mengambil kata-kata penting dari hasil token. Algoritma yang digunakan adalah algoritma stoplist (membuang kata yang kurang penting) atau *wordlist* (menyimpan kata penting).
3. Tahap *stemming* adalah tahap mencari *root* kata dari tiap kata hasil *filtering*. Pada tahap ini dilakukan proses pengembalian berbagai bentukan kata ke dalam suatu representasi yang sama. Tahap ini kebanyakan dipakai untuk teks berbahasa Inggris dan lebih sulit diterapkan pada teks berbahasa Indonesia.
4. Tahap *analyzing* merupakan tahap penentuan seberapa jauh keterhubungan antara kata-kata antar dokumen yang ada. Tahap ini menggunakan algoritma *termfrequency(tf), invers document frequency (idf)* dan kombinasi perkalian antara keduanya (tfxidf). TF-IDF (*Term Frequency-Inverse Document Frequency*) merupakan pembobotan sebuah kata dalam satu dokumen agar dapat diproses lebih lanjut oleh beberapa algoritma lain yang membutuhkan. TF merupakan Term Frequency dari sebuah dokumen. Penggunaan TF-IDF ini akan menggunakan *word count* bonus sebagai hasil dari IDF. Jika IDF sudah didapati, maka akan ditambah dengan 1.

$$W_{dt} = tf_{dt} * IDF_t = log\frac{D}{df_t} \ldots\ldots (2.1)$$

Dimana :

D= dokumen ke-d

t = kata ke-t dari kata kunci

W = bobot dokumen ke-d terhadap kata ke-t

Tf = banyaknya kata yang dicari pada sebuah dokumen

IDF = *Inversed Document Frequency*

D = total dokumen

df = banyak dokumen yang mengandung kata yang dicari

2.3 K-Nearest Neighbor

Algoritma *K-Nearest Neighbor* (k-NN atau KNN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. Data pembelajaran diproyeksikan ke ruang berdimensi banyak, dimana masing-masing dimensi merepresentasikan fitur dari data. Ruang ini dibagi menjadi

bagian-bagian berdasarkan klasifikasi data pembelajaran. Sebuah titik pada ruang ini ditandai kelas c jika kelas c merupakan klasifikasi yang paling banyak ditemui pada k buah tetangga terdekat titk tersebut. Dekat atau jauhnya tetangga biasanya dihitung berdasarkan jarak *Euclidean*.

Untuk mendefinisikan jarak antara dua titik yaitu titik pada data latih (x) dan titik pada data testing (y) maka digunakan rumus *Euclidean* seperti yang ditunjukkan pada persamaan 2.2:

$$\mathrm{d}(\mathrm{x},\mathrm{y}) = \sqrt{\sum_{i=I}^{n}(xi - yi)^2} \dots\dots\dots\dots (2.2)$$

dengan d adalah jarak antara titik pada data latih x dan data uji y yang akan diklasifikasi, dimana x=x1,x2,…,x1 dan y=y1,y2,…,y1 dan merupakan dimensi atribut.

Pada fase pembelajaran, algoritma ini hanya melakukan penyimpanan vektor-vektor fitur dan klasifikasi dari data pembelajaran. Pada fase klasifikasi, fitur-fitur yang sama dihitung untuk data test (yang klasifikasinya tidak diketahui). Jarak dari vektor yang baru ini terhadap seluruh vektor data pembelajaran dihitung, dan sejumlah k buah yang paling dekat diambil. Titik yang baru klasifikasinya diprediksikan termasuk pada klasifikasi terbanyak dari titik-titik tersebut.

Nilai k yang terbaik untuk algoritma ini tergantung pada data; secara umumnya, nilai k yang tinggi akan mengurangi efek *noise* pada klasifikasi, tetapi membuat batasan antara setiap klasifikasi menjadi lebih kabur. Nilai k yang bagus dapat dipilih dengan optimasi parameter, misalnya dengan menggunakan *cross-validation*. Kasus khusus di mana klasifikasi diprediksikan berdasarkan data pembelajaran yang paling dekat (dengan kata lain, k = 1) disebut algoritma nearest neighbor.

Ketepatan algoritma k-NN ini sangat dipengaruhi oleh ada atau tidaknya fitur-fitur yang tidak relevan, atau jika bobot fitur tersebut tidak setara dengan relevansinya terhadap klasifikasi. Riset terhadap algoritma ini sebagian besar membahas bagaimana memilih dan memberi bobot terhadap fitur, agar performa klasifikasi menjadi lebih baik.

Algoritma K-NN adalah sebagai berikut :

1. Tentukan k.
2. Hitung jarak antara data baru ke setiap *labeled* data.
3. Tentukan k *labeled* data yang mempunyai jarak yang paling minimal.
4. Klasifikasikan data baru ke dalam *labeled* data yang mayoritas.

Dalam penelitian ini, teknik k-nearest neighbor digunakan untuk menemukan fungsi pemisah (klasifier) yang optimal yang bisa memisahkan dua set data dari dua kelas yang berbeda. Penggunaan teknik machine learning tersebut, karena performansinya yang meyakinkan dalam memprediksi kelas suatu data baru. Untuk menghitung akurasi dapat menggunaka persamaan 2.3 :

$$akurasi = \frac{jumlah\ klasifikasi\ benar}{total\ file\ yang\ diklasifikasikan}\ x\ 100 \dots\dots (2.3)$$

III. METODOLOGI PENELITIAN

3.1 Tahapan Penelitian

Langkah -langkah yang dilakukan dalam penelitian adalah sebagai berikut :

1. Mengumpulkan Studi literatur berupa refrensi yang bersifat teoritis dari buku-buku dan sumber bacaan lain yang dapat mendukung topik.
2. Mengumpulkan data file abstraksi tugas akhir untuk digunakan sebagai data latih dan data uji.
3. Perancangan sistem klasifikasi menggunakan metode *K-Nearest Neighbor*.
4. Melakukan evaluasi hasil uji coba klasifikasi yang dihasilkan oleh sistem serta melakukan perbaikan terhadap sistem apabila ditemukan kekurangan atau kesalahan.

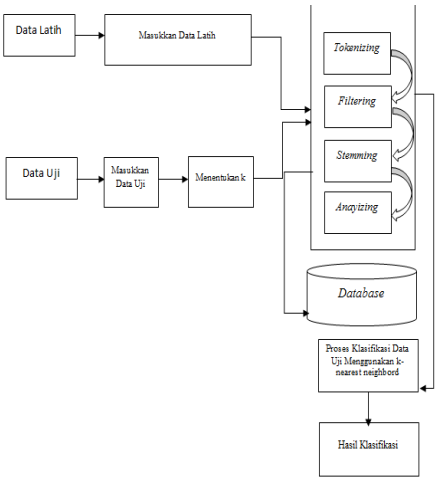
3.2 Deskripsi Data

Data yang digunakan untuk data latih maupun data uji berasal dari dokumen abstraksi tugas akhir program studi Teknik Informatika Universitas Muhammadiyah Jember. Kategori yang digunakan adalah kategori menurut bidang minat dari program studi Teknik Informatika yaitu Jaringan, RPL dan Bisnis Cerdas. Jumlah keseluruhan dokumen berjumlah 100 dokumen tugas akhir yang akan dibagi menjadi 85% (85 dokumen) data latih dan 15% (15 dokumen) data uji.

3.3 Arsitektur Sistem

Klasifikasi teks merupakan proses kategorisasi suatu dokumen teks sesuai dengan karateristik teks tersebut. Dalam prosesnya klasifikasi teks terdiri dari 3 komponen yaitu praproses data, kontruksi pengklasifikasian dan pengkategorian dokumen. Tahapan proses terdiri dari *case folding, tokenizing, filtering, dan stemming* (H. Februariyanti, 2012). Tahap kontruksi pengklasifikasian adalah suatu tahap pembentukan model pengklasifikasi melalui proses pembelajaran terhadap data latih. Sedangkan pengkategorian dokumen adalah uatu tahapan proses testing dari data uji atau data yang akan ditentukan kategorinya berdasarkan model pengklasifikasi yang telah diperoleh.

Arsitektur dapat digambarkan sebagaimana gambar 3.1. Alur prosesnya dapat dijelaskan sebagai berikut :



Gambar 3.1. Arsitektur Sistem

Semua dokumen baik dokumen latih maupun dokumen uji akan dilakukan proses *preprocessing* yang meliputi *case folding* yaitu mengubah semua huruf menjadi kecil, *tokenizing* yaitu pemisahan rangkaian *term* menjadi bentuk token atau potongan kata tunggal, kemudian dilanjutkan pada tahap *filtering* yaitu membuang *term* yang tidak penting dalam koleksi dokumen. Selanjutnya mengubah kembali term menjadi bentuk kata dasar, diakhiri proses *analyzing* yaitu menghitung keterkaitan kata-kata data latih terhadap kata data uji.

Data uji yang telah melalui proses *preprocessing* akan dilakukan pengkategorian dengan membandingkan data uji yang akan dikategorisasi dengan data latih yang telah dilakukan proses *clustering* menggunakan algoritma K-NN.

3.4 Tahapan Uji Coba

Sesuai dengan gambaran umum dari sistem yang akan dibuat dalam penelitian ini, tahapan uji coba akan dilakukan sebagai berikut :

1. Masukkan data latih yaitu judul dan abstraksi tugas akhir. Proses ini akan menyimpan Klasifikasi data latih kedalam database yang nantinya akan dilakukan proses *text minning*.
2. Selanjutnya pada *text mining* akan dilakukan beberapa langkah berikut ini :
    a. *Tokenizing* adalah tahap pemotongan string input berdasarkan tiap kata yang menyusunnya.
    b. *Filtering* adalah tahap mengambil kata-kata penting dari hasil token. Algoritma yang digunakan adalah *stoplist* (membuang kata yang kurang penting) atau *wordlist* (menyimpan kata penting).
    c. *Stemming* adalah tahap mencari *root* kata dari tiap kata hasil *filtering*. Pada tahap ini dilakukan proses pengembalian berbagai bentukan kata ke dalam suatu representasi yang sama.
    d. *Analyzing* adalah tahap penentuan seberapa jauh keterkaitan antar kata-kata dari dokumen yang ada. Tahap ini menghitung keterkaitan kata-kata yang terdapat dalam abstrak data latih terhadap data uji.

3. Masukkan data abstraksi tugas akhir baru sebagai data uji dan tentukan k. Data baru tersebut juga harus melalui tahap *preprocessing* seperti pada data latih.
4. Menentukan kategorisasi data baru sesuai dengan bidang minat yang ada dengan menggunakan algoritma K-NN. Biasanya uji coba menggunakan algoritma K-NN ini harus membandingkan dengan semua data latih.
5. Tahap terakhir adalah tahap pengujian dimana data uji akan diberikan kategori dengan model yang telah dibangun. Sehingga data baru tugas akhir memiliki kategori sesuai dengan bidang minat yang ada yaitu Jaringan, RPL, dan Bisnis Cerdas.

IV. HASIL DAN PEMBAHASAN

4.1 Uji Coba

Data yang diolah pada tugas akhir ini adalah data tugas akhir program studi Teknik Informatika yang mencakup 3 jurusan yaitu Rekayasa Perangkat Lunak (RPL), Bisnis Cerdas dan Jaringan. Data yang diambil berjumlah 100 dokumen yang akan dibagi menjadi 3 skenario. Data latih dimasukkan kedalam database melalui aplikasi sebagai antarmuka yang menjembatani pengguna dan sistem. Sedangkan data uji akan dilakukan pengujian klasifikasi menggunakan aplikasi.

Tabel 4.1 Deskripsi Sekenario Data

Skenario	Jumlah Data	
	Data Latih	Data Uji
K=3	85	15
K=5	85	15
K=7	85	15

4.2 Hasil Pengujian

Pengujian kategorisasi menggunakan metode *k-nearest neighbor* ini menggunakan 3 skenario untuk menghasilkan akurasi yang paling optimal. Dengan menggunakan data sesuai sekenario yang telah disebutkan pada tabel 4.1, berikut hasil yang diperoleh dari 3 skenario tersebut :

1. Hasil Sekenario Pengujian Pertama

   Pada skenario pertama, data yang digunakan berjumlah 85 data latih dan 15 data uji dengan k=3. Data latih yang digunakan untuk menentukan kategori dari data uji hanya 3 data yang memiliki jarak terdekat dari data uji.

   Berikut adalah hasil presentase pengujian skenario pertama untuk data uji :

Tabel 4.3 Hasil Pengujian Skenario 1 dengan k=3

Sumber	Benar	Salah	Total	Presentase
Bisnis Cerdas	4	1	5	80%
Jaringan	4	1	5	80%
RPL	3	2	5	60%
Total	11	4	15	73,3%

1. Hasil Sekenario Pengujian kedua

   Skenario kedua menggunakan 85 data latih dan 15 data uji dengan k=5. Data latih yang digunakan untuk menentukan kategori dari data uji hanya 5 data yang memiliki jarak terdekat dari data uji. Berikut adalah hasil presentase pengujian skenario kedua untuk data uji:

Tabel 4.5 Hasil Pengujian Skenario 2 dengan k=5

Sumber	Benar	Salah	Total	Presentase
Bisnis Cerdas	4	1	5	80%
Jaringan	4	1	5	80%
RPL	4	1	5	80%
Total	11	4	15	80%

2. Hasil Sekenario Pengujian Ketiga

   Skenario ketiga menggunakan 85 data latih dan 15 data uji dengan k=7. Data latih yang digunakan untuk menentukan kategori dari data uji hanya 7 data yang memiliki jarak terdekat dari data uji. Berikut adalah hasil presentase pengujian skenario ketiga untuk data uji:

Tabel 4.7 Hasil Pengujian Skenario 3 dengan k=7

Sumber	Benar	Salah	Total	Presentase
Bisnis Cerdas	5	0	5	100%
Jaringan	5	0	5	100%
RPL	4	1	5	80%
Total	14	1	15	93,3%

4.3 Analisa Hasil Penelitian

Setelah melakukan 3 kali percobaan dengan k yang berbeda, didapatkan bahwa akurasi tertinggi diperoleh dari sekenario 3 dengan k=7. Hasil yang didapatkan sebesar 93,3% dengan kesalahan klasifikasi sebesar 6,7%.

Penentuan k sangat mempengaruhi hasil klasifikasi pemilihan. Jika k yang dipilih sangat kecil maka kategori pada data uji hanya bergantung pada beberapa data latih yang mewakili karateristik dari kategori. Namun banyaknya jumlah data latih juga mempengaruhi tingkat keakuratan dari sistem klasifikasi tersebut.

4.1 Pembahasan

Kategorisasi dokumen menggunakan k-nearest neighbor pada percobaan yang dilakukan dapat memperoleh hasil yang cukup baik. Tingkat keberhasilan yang diperoleh lebih banyak dibanding tingkat kegagalan dalam proses klasifikasi.

Pada percobaan pertama dimana k=3 memperoleh hasil akurasi 73,3%. Kategorisasi RPL memiliki tingkat akurasi yang lebih rendah dibanding bisnis cerdas dan jaringan.

Pada percobaan kedua sistem klasifikasi memperoleh hasil yang lebih baik dengan menggunakan k=5 daripada k=3. Masing-masing memiliki hasil keakuratan yang sama dengan kesalahan hanya 20%.

Pada percobaan ketiga dipilih k=7, hasil yang didapatkan pada proses klasifikasi pada 2 kategori sangat akurat. Kategori tersebut adalah bisnis cerdas dan jaringan. Pada skenario ini menunjukan tingkat akurasi 93,3%.

Akurasi penelitian ini dihitung menggunakan persamaan 2.3. Berikut adalah akurasi yang dihasilkan oleh ketiga skenario :

Tabel 4.8 Akurasi

Skenario	Akurasi
K=3	73,3%
K=5	80%
K=7	93,3%

Dari 3 skenario ini dihasilkan akurasi paling tinggi adalah skenario ketiga yaitu 93,3% dan paling rendah skenario ke pertama yaitu 73,3% . Sehingga kategorisasi menggunakan metode *k-nearest neighbor* memliki akurasi rata-rata sebesar 82,2%

DAFTAR PUSTAKA

G. Toker and Ö. Kırmemiş, "TEXT CATEGORIZATION USING K-NEARESTNEIGHBOR CLASSIFICATION."

H. Februariyanti and E. Zuliarso, "Klasifikasi Dokumen Berita Teks Bahasa Indonesia menggunakan Ontologi," J. Teknol. Inf. Din., vol. 17, no. 1, pp. 14–23, 2012.

H. Muhammad Isa , "Klasifikasi Dokumen Teks Menggunakan Metode Support Vector Mechine Dengan Pemilihan fitur Chi-Square" , Jember, 2016.

J. Samodra, S. Sumpeno, and M. Hariadi, "Klasifikasi Dokumen Teks Berbahasa Indonesia dengan Menggunakan Naïve Bayes," Semin. Nas. Electr. Informatic, IT's Educ., pp. 1–4, 2009

Kusrini, Emha Taufiq Luthfi, "Algoritma Data Mining",Yogyakarta, Andi, 2009.

R. Rakhmat Sani , J. Zeniarja , A. Luthfiarta "Penerapan Algoritma K-Nearest Neighbor pada Information Retrieval dalam Penentuan Topik Referensi Tugas Akhir" Journal of Applied Intelligent System, Vol. 1, No. 2, Juni 2016: 123 – 133

Zhang, X.F, Huang, H.Y, Zhang K.L. 2009. KNN Text Categorization Algorithm Based on Semantic Centre. 2009 International Conference on Information Technology and Computer Science.

.