LogicLink : Journal of Artificial Intelligence and Multimedia in Informatics
Vol. 1 No. 1, June 2024, pp. 37 – 49

Peringkasan Dokumen Teks Bilingual Sebagai Reduksi Fitur Untuk Klasifikasi Menggunakan Algoritma K-NN

Rahmawan Bagus Trianto*[1], **Agus Susilo Nugroho**[2]
[1,2]*Universitas An Nuur, Grobogan, Indonesia*
*E-mail : rahmawanbagust@gmail.com**
**Penulis Korespondensi*

Received 31 May 2024; Revised 9 June 2024; Accepted 24 June 2024

Abstrak - Meringkas teks merupakan langkah untuk mengambil intisari dari suatu dokumen teks dengan tidak lebih dari setengahnya. Meringkas teks memiliki peran yang penting dalam mengambil informasi inti dari suatu dokumen dalam bentuk yang lebih ringkas. Meringkas dokumen teks dapat dijadikan sebagai reduksi fitur dalam melakukan klasifikasi dokumen teks karena dapat mengurangi fitur yang dianggap tidak relevan. Dokumen teks diringkas dengan metode *Term Frequency-Inverse Document Frequency* (TF-IDF), kemudian diklasifikasikan menggunakan algoritma *K-Nearest Neighbor* (K-NN). Salah satu kekurangan dari algoritma K-NN adalah tidak maksimal dalam klasifikasi jika nilai *k* tidak sesuai, serta pemilihan metode penghitung jarak yang tidak tepat. Dengan menggunakan pengujian berbagai jumlah nilai *k* dan menggunakan metode pengukuran jarak *Euclidean Distance* dapat meningkatkan akurasi klasifikasi dokumen teks. Peringkasan dokumen teks menggunakan metode TF-IDF yang diusulkan terbukti semakin meningkat ketika dilakukan klasifikasi dengan K-NN. Dari hasil penelitian didapatkan peningkatan akurasi klasifikasi pada compression rate sebesar 50% dengan nilai *k* 6 sampai 8 sebesar 95.33%. Hal ini menunjukkan bahwa peringkasan dokumen teks sebagai reduksi fitur memiliki peran positif dalam proses klasifikasi menggunakan algoritma K-NN.
Kata Kunci: ringkasan, dokumen, TF-IDF, K-NN.

Abstract - Summarizing text is a step to extract the essence of a text document with no more than half. Summarizing text has an important role in extracting the core information from a document in a more concise form. Summarizing text documents can be used as feature reduction in classifying text documents because it can reduce features that are considered irrelevant. Text documents are summarized using the *Term Frequency-Inverse Document Frequency* (TF-IDF) method, then classified using the *K-Nearest Neighbor* (K-NN) algorithm. One of the disadvantages of the K-NN algorithm is that it is not optimal in classification if the k value is not appropriate, as well as the selection of an inappropriate distance calculation method. By testing various k values and using the *Euclidean Distance* distance measurement method, you can increase the accuracy of text document classification. Text document summarization using the proposed TF-IDF method is proven to increase when classification is carried out with K-NN. From the research results, it was found that the classification accuracy at the compression rate increased by 50% with a k value of 6 to 8 of 95.33%. This shows that text document summarization as feature reduction has a positive role in the classification process using the K-NN algorithm.
Keywords: summarization, document, TF-IDF, K-NN.

1. PENDAHULUAN

Teks adalah salah satu jenis data yang tersebar di internet dengan jumlah yang besar yang dengan mudah didapatkan. Menurut *Radicati Group Inc*, melaporkan terdapat sekitar 293 miliar email pada tahun 2019 dan pada tahun 2023 meningkat menjadi lebih dari 347 miliar email yang beredar, belum lagi data teks lain yang semakin banyak tersebar di internet (Hassani et al., 2020). Dengan banyaknya sumber informasi yang tersedia, manusia tidak memungkinkan untuk mengakses dan mendapatkan informasi yang akurat setiap informasi yang ada (Yadav et al., 2015). Dengan adanya peringkasan teks otomatis, maka pengguna akan lebih efisien dan efektif dalam membaca informasi yang tersedia (Wibowo, 2014). Penggunaan bahasa pada suatu teks juga meningkat yang menjadi dampak dari perkembangan data di internet itu sendiri, yang mengakibatkan informasi multi bahasa tidak dapat dihindarkan (Martín-Valdivia et al., 2013). Untuk mendapatkan model peringkasan teks yang optimal, maka digunakan juga data latih yang multi bahasa, dan dalam penelitian ini dibatasi dua bahasa atau *bilingual* dataset, yaitu Bahasa Indonesia dan Bahasa Inggris.

Metode yang sering dipakai peneliti untuk klasifikasi dokumen teks adalah *Support Vector Machine* (SVM) (Amalia & Yustanti, 2021; Nanda et al., 2022), *Naïve Bayes* (Asril et al., 2019; Hidayat & Rizqi, 2020), dan *K-Nearest Neighbor* (Akromunnisa & Hidayat, 2019; Asril et al., 2019; Firdaus et al., 2019). *Naïve Bayes* memiliki keunggulan dengan asumsi independensi yang kuat. Tetapi memiliki kekurangan di tingkat akurasinya dalam situasi riil/kenyataan yang kompleks di mana terdapat keterkaitan antar fitur (Dhande & Patnaik, 2014). *SVM* memiliki kelebihan mampu mengidentifikasi hyperplane terpisah dengan memaksimalkan margin antara dua kelas yang berbeda (Nurhadi, 2015). SVM merupakan algoritma non-parametrik yang kompleksitasnya meningkat secara kuadratik, yang sangat baik dipakai pada data dengan jumlah sedikit dan banyak fitur (Erfani et al., 2015). K-NN memiliki kelebihan dalam menangani data dalam jumlah yang besar dan dengan *noisy* pada data *training*. Akan tetapi K-NN memiliki kekurangan dalam penentuan nilai k yang dapat menghasilkan keluaran yang optimal (Guo et al., 2006). Selain itu, K-NN memiliki kekurangan dalam hal komputasi yang kompleks, penggunaan memori yang besar serta terjadi bias terhadap fitur yang tidak relevan (Mutrofin et al., 2014).

Meringkas teks merupakan langkah untuk mengambil intisari dari suatu dokumen teks dengan tidak lebih dari setengah dari total teks yang ada (Wibowo, 2014). Dengan asumsi setiap kata dalam dokumen teks adalah fitur, maka dengan meringkas teks dapat mengurangi jumlah dari fitur yang ada. *Term Frequency-Inverse Document Frequency* (TF-IDF) merupakan salah satu metode yang dipakai dalam peringkasan teks otomatis. Dengan fungsi dari TF-IDF untuk peringkasan teks otomatis, maka metode ini dapat digunakan untuk mengatasi salah satu kekurangan dari algoritma SVM yang memiliki masalah dalam pemilihan fitur, yaitu dengan mengurangi jumlah fitur yang tidak relevan yang dapat mengakibatkan tingginya dimensi data. Di mana dengan tingginya dimensi data akan menyebabkan kesalahan dalam generalisasi klasifikasi[6].

Dengan peringkasan dokumen teks dan berkurangnya fitur serta menggunakan algoritma K-NN sebagai algoritma pengklasifikasinya, dapat meningkatkan akurasi klasifikasi. Selain itu, dengan memanfaatkan peringkasan dokumen, dapat digunakan sebagai pengambil intisarinya, sehingga mempermudah pembaca dalam menangkap pokok pikiran dari dokumen.

1.1. Tinjauan Studi

Masalah penelitian pada (Firdaus et al., 2019) adalah pengklasifikasian dokumen secara manual memerlukan waktu dan tenaga yang lebih banyak, terlebih pada dokumen yang berukuran besar. Diperlukan sistem yang untuk mengklasifikasikan dokumen secara otomatis

dengan algoritma K-NN dan pengukuran jaraknya menggunakan *cosine similarity*. Dengan menggunakan *cosine similarity* dan algoritma klasifikasi K-NN menunjukkan akurasi yang sangat baik, yaitu di atas 97%.

Pada penelitian (Asril et al., 2019) dilakukan klasifikasi untuk penentuan dosen pembimbing berdasarkan tema tugas akhir. Dengan menggunakan algoritma *Naïve Bayes* dan K-NN didapatkan akurasi terbaiknya masing-masing 86.11% dan 91.67%. Pada penelitian tersebut K-NN memiliki akurasi terbaik.

Pada penelitian (Akromunnisa & Hidayat, 2019) dilakukan klasifikasi dokumen digital tugas akhir mahasiswa. Sebelum diklasifikasikan dokumen akan dilihat bagian terpenting dari abstrak yang selanjutnya dikelompokkan dengan beberapa algoritma, yaitu SVM, *Naïve Bayes*, K-NN, *Decision Tree* dan juga *Artificial Neural Network* (ANN). Dari hasil penelitian tersebut didapatkan algoritma K-NN sebagai yang terbaik dalam keakuratan klasifikasi.

Penelitian (Hidayat & Rizqi, 2020) klasifikasi dokumen teks berupa berita menggunakan algoritma *Naïve Bayes*. *Enhance Confix Stripping Stemming* digunakan pada penelitian ini pada tahap *preprocessing*, yaitu sebelum dilakukan klasifikasi untuk mendapatkan kata dasar dari imbuhan-imbuhan yang ada pada suatu kata. Pada penelitian ini menghasilkan tingkat akurasi rata-rata sebesar 95%.

Penelitian (Amalia & Yustanti, 2021) juga memanfaatkan *text mining* berupa klasifikasi buku, jurnal sampai majalah secara digital pada perpustakaan. Tujuan dari penelitian ini adalah untuk mengelompokkan buku agar memudahkan pengguna perpustakaan dalam menemukan buku. Algoritma yang dipakai adalah *Support Vector Machine* (SVM). Hasil dari penelitian ini dengan menggunakan SVM dalam klasifikasi memiliki akurasi yang cukup rendah, yaitu hanya sebesar 69.24%, presisi sebesar 71% dan recall sebesar 61% serta *f1-score* sebesar 64%.

Penelitian (Nanda et al., 2022) melakukan pengelompokan berita secara otomatis dengan menggunakan algoritma SVM. Hal ini memudahkan penulis berita tanpa harus menafsirkan berita untuk masuk pada kategori tertentu. Dengan menggunakan algoritma SVM menghasilkan akurasi sebesar 88%.

Masalah penelitian pada (Zhang et al., 2015) adalah pembobotan fitur untuk digunakan sebagai pengelompokan data teks memiliki kelemahan, seperti keterbatasan dalam meningkatkan performa pengelompokan data teks dan membutuhkan waktu yang lama untuk mendapatkan modelnya. Membandingkan penggunaan dua metode pembobotan fitur berbasis rasio-gain, dan decision tree untuk mendapatkan performa terbaik. Hasil penelitian menunjukkan peningkatan performa, lebih efektif, efisien dan lebih cepat dibandingkan dengan penelitian sebelumnya.

Masalah penelitian pada (Abdel Fattah, 2015) adalah sulit dalam menentukan pengetahuan semantik yang tepat pada term berdasarkan kategori dalam kumpulan term yang besar. Pendekatan statistikal dinilai lebih mudah, sehingga diajukan beberapa metode term weighting untuk diperbandingkan (TF-IDF, TF-ICF, MI, OR, WLLR, CHI, CDall_c, CD_c, log_CD). Klasifikasi diperbandingkan antara SVM, PNN dan Gaussian mixture model, serta kombinasi multiple classifier. Dari hasil penelitian dihasilkan metode CDall_c menghasilkan akurasi tertinggi, untuk masing-masing dataset, (Moview Review 92.3%, Books 89.1%, Dvd 90.7%, Electronic 91.1%, Kitchen 92.8%).

Masalah penelitian pada (Babar & Patil, 2015) adalah proses ekstraksi data teks dalam mereview masih bergantung pada upaya manual yang lambat, mahal dan tingkat kesalahan manusia yang tinggi. Metode yang digunakan adalah Fuzzy Logic dan Latent Semantic Analysis. Hasil menunjukkan metode yang diusulkan lebih baik dalam presisi (90.775%), recall (44.363%) dan f-measurement (67.569%) dibandingkan dengan fuzzy logic, yaitu presisi 86.91%, recall 41.64%, dan f-measurement 64.66%.

Masalah penelitian pada (Fachrurrozi et al., 2013) adalah meringkas teks dalam jumlah yang besar secara manual membutuhkan waktu dan tenaga lebih. Sehingga menghasilkan

ringkasan yang tidak maksimal. Metode yang digunakan adalah TF-ISF. Dari hasil penelitian didapatkan f-measure sebesar 78%, dengan penilaian responden secara manual sebesar 83.3% pada rasio kompresi peringkasan sebesar 30%.

Masalah penelitian pada (Chandani et al., 2015) adalah Klasifikasi sentimen pada data teks memiliki masalah pada banyaknya atribut yang tidak relevan yang ada pada dataset, sehingga mengakibatkan tingkat akurasi yang tidak optimal. Membandingkan beberapa algoritma feature selection (information gain, chi square, forward selection dan backward elimination) pada SVM, ANN dan NB untuk klasifikasi sentimen. Hasil perbandingan beberapa algoritma feature selection yang diterapkan pada algoritma klasifikasi SVM (akurasi 81.10%, AUC 0.904) didapatkan hasil terbaik adalah information gain dengan nilai rata-rata akurasi 84.57% dan AUC 0.899.

1.2 Landasan Teori

Peringkasan teks otomatis adalah suatu langkah untuk mendapatkan gagasan pokok atau inti dari suatu dokumen teks dengan menggunakan komputer (Najibullah, 2015). Dengan peringkasan teks secara otomatis menggunakan komputer dapat mengurangi waktu baca dan meningkatkan efisiensi dan efektifitas dalam mendapatkan informasi yang terkandung dalam dokumen teks. Ringkasan yang baik adalah ringkasan yang dapat menghasilkan informasi inti dari suatu dokumen teks dengan membuang informasi atau kata-kata yang tidak relevan (Ajmal, 2015). Peringkasan teks otomatis menggunakan komputer merupakan salah satu cabang penelitian di bidang *text mining*.

Text mining merupakan salah satu bidang khusus dari *data mining*. *Text mining* merupakan suatu proses menggali informasi yang terdapat pada dokumen atau kumpulan dokumen teks (Nurhadi, 2015). Definisi lain dari *text mining* adalah proses mengambil informasi dari dokumen teks (Lidya et al., 2015). Dari definisi tersebut dapat ditarik tujuan dari dilakukannya *text mining* yaitu untuk mendapatkan informasi yang terkandung dalam dokumen teks. Sehingga data yang digunakan adalah dokumen teks, baik dalam bentuk terstruktur maupun tidak terstruktur. Seperti pada *data mining*, *text mining* juga memiliki fungsi untuk melakukan klasifikasi (*classification*) dan pengelompokan (*clustering*) dokumen teks (Nurhadi, 2015). Secara umum tahapan dalam melakukan peringkasan teks otomatis menggunakan *text mining* adalah dimulai dengan *preprocessing*, *feature extraction*, *sentence selection* (*term weighting*), *summarization* (Babar & Patil, 2015; Hidayatullah, 2015). Sedangkan untuk klasifikasi data teks secara umum tahapannya adalah *preprocessing*, *feature extraction*, *classification* (Hidayatullah, 2015; Wijaya & Santoso, 2016).

Term Frequency-Inverse Document Frequency terdiri dari dua tahap perhitungan, yaitu *Term Frequency* dan *Inverse Document Frequency*. *Term Frequency* (TF) merupakan sebuah formula yang sederhana namun efektif dalam mengevaluasi bobot suatu term dengan cara menghitung jumlah kemunculan dari term dalam suatu dokumen (García Adeva et al., 2014). Sedangkan *Inverse Document Frequency* (IDF) memperhatikan kemunculan term dalam kumpulan dokumen (Wijaya & Santoso, 2016). IDF dapat dihitung dengan logaritma dari pembagian antara jumlah dokumen dengan frekuensi dokumen yang memuat suatu *term*. Sehingga TF-IDF dapat dihitung menggunakan perkalian antara TF dengan IDF (Luthfiarta et al., 2013). Persamaan TF, IDF dan TF-IDF dapat dilihat pada persamaan 1, 2 dan 3 sebagai berikut (Abdel Fattah, 2015).

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \qquad (1)$$

Dimana $n_{i,j}$ adalah jumlah kemunculan *term* ke-i pada dokumen ke-j. Sedangkan penyebutnya adalah total dari jumlah kemunculan semua term dalam dokumen ke-j.

$$idf_i = \log \frac{|D|}{|\{d: t_i \epsilon d\}|}$$, atau bisa juga, $$idf_i = \log \frac{N}{n_j}$$ (2)

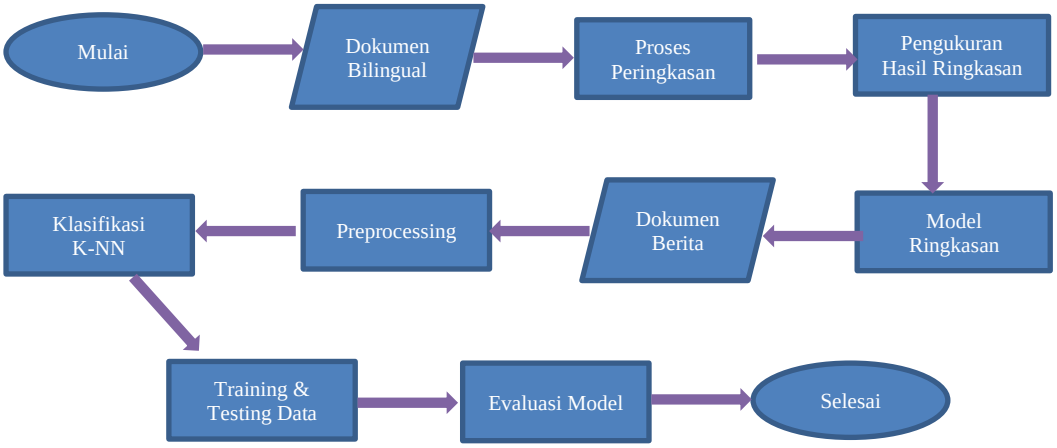
Dimana |D| adalah jumlah dokumen dalam korpus dan $|\{d : t_i \epsilon d\}|$ adalah jumlah dokumen di mana *term* ke-i muncul. Atau pada persamaan yang lain N merupakan jumlah dokumen dataset, dan n_j merupakan jumlah dokumen di mana term j muncul.

$$tfidf_{i,j} = tf_{i,j} \times idf_i$$ (3)

2. METODE PENELITIAN

2.1 Metodologi Yang Diusulkan

Metodologi yang diusulkan pada penelitian ini dapat dilihat pada gambar berikut.



Gambar 1. Metodologi Yang Diusulkan

Gambar 1. di atas merupakan metodologi yang diusulkan pada penelitian ini. Dimulai dari pencarian dataset, kemudian proses peringkasan dokumen teks menggunakan TF-IDF untuk melihat efektivitasnya dengan meminta pengguna menilai hasil ringkasan sesuai *compression rate* tertentu. Setelah itu diukur performa hasil ringkasan dokumen tersebut. Model ringkasan yang terbentuk akan digunakan untuk klasifikasi dokumen teks menggunakan algoritma K-NN. Dilanjutkan dengan pengukuran performa model klasifikasi di tahap akhirnya.

2.2 Data

Data yang digunakan pada penelitian ini terbagi menjadi dua jenis, yaitu data untuk peringkasan dokumen bilingual yang didapatkan dari aplikasi yang dikembangkan yang beralamat di https://aglabridgemedia.com/ringkasin yang didapatkan sebanyak 343 data. Data untuk klasifikasi dokumen yang didapatkan dari portal berita https://viva.co.id yang terdiri dari 5 kategori, yaitu travel, otomotif, kuliner, bola dan politik. Masing-masing kategori diambil secara acak sebanyak 30 data berita, sehingga total dokumen untuk klasifikasi sebanyak 150 data.

2.3 Peringkasan Dokumen

Peringkasan dokumen teks diawali dengan tahap *preprocessing*, kemudian melakukan *term weighting* menggunakan metode TF-IDF. Masukan berupa kumpulan dokumen teks berbahasa Indonesia maupun berbahasa Inggris (bilingual), kemudian diujikan untuk klasifikasi menggunakan data lainnya yang terdiri dari 5 kategori, yaitu travel, otomotif, kuliner, bola dan

politik, masing-masing kategori berisi 30 dokumen teks. Secara keseluruhan total dokumen sebanyak 150 dokumen teks.

Pada penelitian ini tahap *preprocessing* akan dilakukan *sentences segmentation*, *tokenizing*, *stopword removal* dan *stemming*. *Sentences segmentation* adalah tahap mengambil kalimat-kalimat dari dokumen dengan tanda pemisah titik, tanda seru dan tanda tanya. Contoh *sentences segmentation* dapat dilihat pada tabel 1 berikut.

Tabel 1. Hasil *Sentences Segmentation*

Dokumen	Hasil *Sentences Segmentation*
Lawan tangguh menanti Manchester City tengah pekan ini. ManCity akan bertandang ke Stade Louis II menghadapi AS Monaco di leg kedua babak 16 besar Liga Champions, Kamis dinihari 16 Maret 2017. Untuk bisa lolos ke perempatfinal, ManCity cukup bermain imbang, atau kalah satu gol saja. Meski demikian, manajer ManCity Pep Guardiola tetap ingin bermain menyerang. Guardiola tidak mau kemenangan 5-3 di leg pertama, menjadi sebab timnya bermain bertahan. Sebab, lawan yang mereka hadapi sangat agresif dalam hal menyerang. "Soal mencetak gol, mereka tim terbaik di dunia. Mereka menyerang dengan lima hingga enam pemain," kata Guardiola seperti dilansir Skysports, Rabu 15 Maret 2017. "Mereka juga memiliki fisik yang kuat dan cepat di ruang pendek," lanjutnya. Jumlah total gol Monaco musim ini menjadi tolak ukur kewaspadaan Guardiola. "Ketika sebuah tim mencetak 123 gol, ini menunjukkan mereka mampu mencetak gol dengan beragam cara," jelas Guardiola. "Jadi, cara terbaik mengatasinya dengan mencetak banyak gol. Kuasai bola dan terus menyerang sebisa mungkin. Ini satu-satunya cara menaklukkan tim seperti ini," lanjutnya. *Sumber : http://www.viva.co.id/bola/read/894329-guardiola-waspadai-amuk-monaco-di-kandang*	• Lawan tangguh menanti Manchester City tengah pekan ini. • ManCity akan bertandang ke Stade Louis II menghadapi AS Monaco di leg kedua babak 16 besar Liga Champions, Kamis dinihari 16 Maret 2017. • Untuk bisa lolos ke perempatfinal, ManCity cukup bermain imbang, atau kalah satu gol saja. • Meski demikian, manajer ManCity Pep Guardiola tetap ingin bermain menyerang. • Guardiola tidak mau kemenangan 5-3 di leg pertama, menjadi sebab timnya bermain bertahan. Sebab, lawan yang mereka hadapi sangat agresif dalam hal menyerang. • "Soal mencetak gol, mereka tim terbaik di dunia. • Mereka menyerang dengan lima hingga enam pemain," kata Guardiola seperti dilansir Skysports, Rabu 15 Maret 2017. • "Mereka juga memiliki fisik yang kuat dan cepat di ruang pendek," lanjutnya. • Jumlah total gol Monaco musim ini menjadi tolak ukur kewaspadaan Guardiola. • "Ketika sebuah tim mencetak 123 gol, ini menunjukkan mereka mampu mencetak gol dengan beragam cara," jelas Guardiola.

	• "Jadi, cara terbaik mengatasinya dengan mencetak banyak gol. • Kuasai bola dan terus menyerang sebisa mungkin. • Ini satu-satunya cara menaklukkan tim seperti ini," lanjutnya.

Tokenizing adalah tahap memecah kalimat menjadi kumpulan kata-kata dengan pemisahnya adalah tanda spasi. Sebagai contoh: "Budi belajar menulis di kelas." dan menghasilkan potongan kata {"Budi", "belajar", "menulis", "di", "kelas"}. Kemudian *stopword removal* adalah tahap membuang kata yang tidak relevan dengan topik, atau kumpulan kata yang dianggap tidak memiliki arti dalam Bahasa Indonesia maupun Bahasa Inggris. Contoh dari kata ini antara lain *di, pada, ke*, *dan*. Selanjutnya tahap *preprocessing* terakhir adalah *stemming*, yaitu mencari kata dasar dari kata-kata yang ada dalam dokumen. Sebagai contoh kata "*menulis*" akan menghasilkan keluaran *stemming* berupa "*tulis*".

Langkah berikutnya dilanjutkan dengan perhitungan TF-IDF untuk mendapatkan bobot kata dan kalimat, yang akan dijadikan ringkasan. Langkah ini dimulai dengan menghitung nilai TF sesuai dengan persamaan (1). Kemudian menghitung nilai IDF dengan persamaan (2). Dan terakhir mengalikan hasil TF dan IDF dengan menggunakan persamaan (3). Bobot masing-masing *term*/kata akan didapat pada proses ini. Untuk mendapatkan ringkasan maka bobot masing-masing kata pada masing-masing kalimat dijumlahkan, kemudian diurutkan berdasarkan nilai tertingginya.

Pada peringkasan teks otomatis, dikenal dengan adanya *compression rate*. *Compression rate* merupakan nilai kompresi hasil ringkasan yang diharapkan dalam satuan persen (Fachrurrozi et al., 2013). Fungsi dari *compression rate* adalah untuk membatasi hasil ringkasan yang dihitung.

2.4 Pengukuran Peringkasan Dokumen

Dokumen teks yang telah diringkas akan dinilai tingkat kesesuaian dengan isi dari keseluruhan data teks asli. Untuk mengukur ringkasan teks otomatis diperlukan ringkasan manual yang dibuat oleh manusia kemudian diperbandingkan (Lloret & Palomar, 2012). Pada penelitian ini, dibuat aplikasi yang disebar dan diisi oleh pengguna internet yang berada di laman https://aglabridgemedia.com/ringkasin. Pengguna diminta memberikan penilaian dari hasil ringkasan dengan skala likert 1-5, yaitu skala 1 (sangat tidak setuju), 2 (tidak setuju), 3 (netral), 4 (setuju), 5 (sangat setuju). Skala Likert merupakan skala psikometrik yang paling banyak digunakan untuk berbagai penelitian maupun survey (Ghiassi et al., 2013; Rahayu & Shafina, 2022).

2.5 Klasifikasi Dokumen Dengan K-NN

Algoritma *K-Nearest Neighbor* (K-NN) salah satu algoritma *supervised learning* berdasarkan anggota terbanyak dari jumlah *k* terdekatnya. Oleh sebab itu, penentuan nilai *k* menjadi sangat penting. Berdasarkan penjabaran pada studi pustaka, algoritma K-NN merupakan algoritma terbaik berdasarkan tingkat akurasi jika dibandingkan dengan algoritma lainnya. Oleh karena itu, pada penelitian ini menggunakan algoritma K-NN untuk klasifikasi dokumen teks. Selanjutnya, urutan kerja dari algoritma K-NN dijabarkan sebagai berikut:

1. Menentukan nilai *k*, pada penelitian ini menggunakan teknik pengulangan dari *k* = 3 sampai 11.
2. Mencari nilai jarak dataset dengan objek baru.

3. Mengurutkan jarak terdekat berdasarkan hasil langkah 2.
4. Menentukan jarak terdekat sampai pada urutan ke *k*.
5. Mengelompokkan pada kelas yang terpilih.

Pada tahap 2 di atas, diperlukan persamaan *Euclidean Distance* yang dapat dilihat pada persamaan 4 berikut.

$$d_{(x1,x2)} = \sqrt{\sum_{i=1}^{n}(x_{2i} - x_{1i})^2} \qquad (4)$$

Dimana untuk mencari nilai di menggunakan akar kuadrat dari selisih nilai dataset dengan objek yang baru (x2 dan x1).

2.6 Validasi dan Evaluasi Model

Validasi pada penelitian ini menggunakan *k- Fold Cross Validation* dengan nilai *k* sebesar 10. Dari beberapa penelitian yang telah dilakukan, validasi dengan teknik ini memiliki performa rata-rata terbaik, sehingga penelitian ini juga menggunakan teknik yang sama. *10-Fold Cross Validation* berarti dataset akan dipilah menjadi 10 bagian, di mana 9 bagian digunakan untuk data latih dan 1 bagian dipakai sebagai data uji. Proses ini diulang sebanyak 10 kali dan diambil rata-ratanya. Selanjutnya untuk mengevaluasi model yang dihasilkan, digunakan pengukuran akurasi yang dihitung dengan persamaan 5 berikut.

$$Akurasi = \frac{\sum_{i=1}^{c}\frac{TPi + TNi}{TPi + TNi + FPi + FNi}}{c} * 100\% \qquad (5)$$

Dimana:

- c = jumlah label
- TPi = jumlah data positif yang diklasifikasikan benar oleh sistem pada label ke i
- TNi = jumlah data negatif yang diklasifikasikan benar oleh sistem pada label ke i
- FNi = jumlah data negatif yang diklasifikasikan salah oleh sistem pada label ke i
- FPi = jumlah data positif yang diklasifikasikan salah oleh sistem pada label ke i

3. HASIL DAN PEMBAHASAN

Pada penelitian ini peringkasan teks dimanfaatkan sebagai reduksi fitur pada klasifikasi dokumen menggunakan algoritma K-NN. Ringkasan teks yang baik adalah yang mewakili isi dari keseluruhan dokumen teks. Ringkasan teks dilakukan dengan melakukan pemampatan dokumen dengan *compression rate* sebesar 10%, 20%, 30% 40% dan 50% dari total isi dokumen teks. Untuk mengetahui tingkat kesesuaian antara hasil ringkasan dengan dokumen asli dilakukan penilaian oleh pengguna aplikasi yang telah dibuat pada laman https://aglabridgemedia.com/ringkasin dengan memberi nilai pada rentang skala 1 sampai dengan 5.

Langkah awal dimulai dengan tahap preprocessing dataset, dilanjutkan dengan meringkas dokumen sebagai reduksi fiturnya menggunakan TF-IDF. Meringkas teks diawali dengan tahap preprocessing *text mining*, yaitu *tokenizing* (memecah paragraph atau kalimat menjadi kumpulan kata-kata), *stopword removal* (membuang kata-kata yang tidak relevan) dan *stemming*. Setelah didapatkan pembobotan kata, maka akan dihitung totalnya sebagai bobot dari kalimat dan dilakukan pembobotan kata dengan TF-IDF dan dilanjutkan dengan meringkas sesuai *dengan compression rate* yang diusulkan, yaitu 10%, 20%, 30%, 40% dan 50%. Pengukuran hasil ringkasan untuk setiap *compression rate* dapat dilihat pada tabel 1 di bawah ini.

Tabel 2. Sebaran Dokumen Teks Yang Telah Diringkas

Compression Rate (%)	Total Data	Skala 1	Skala 2	Skala 3	Skala 4	Skala 5
10	10	3	0	0	0	7
20	20	4	1	2	2	11
30	136	17	11	11	19	75
40	21	2	3	3	2	11
50	45	4	7	2	2	30
60	13	2	1	0	2	8
70	33	4	4	2	3	20
80	11	0	1	2	0	7
90	7	3	2	1	0	1
100	33	13	4	3	0	13
Total Data	**329**	**52**	**34**	**26**	**30**	**183**
Per Rate (%)	**100.00%**	**15.81%**	**10.33%**	**7.90%**	**9.12%**	**55.62%**

Pada tabel 1 di atas, dapat dilihat bahwa untuk setiap skala memiliki prosentase yang beragam. Pada skala 1 (sangat tidak setuju) sebesar 15.81%. Pada skala 2 (tidak setuju) sebesar 10.33%, skala 3 sebesar 7.9%, skala 4 sebesar 9.12% dan skala 5 sebesar 55.62%. Jika dilihat dari skala 3 sampai 5, yang merupakan pendapat netral, setuju dan sangat setuju terhadap kesesuaian hasil ringkasan dengan dokumen aslinya didapat prosentase sebesar 72.64%, dan sisanya (skala 1 dan 2) yang cenderung tidak setuju sebesar 26.14%. Dari hasil ini hasil ringkasan otomatis dengan teknik TF-IDF memberikan hasil yang cukup baik dan dapat mewakili isi keseluruhan dokumen teks.

Dari model peringkasan dokumen teks yang didapatkan di atas, selanjutnya digunakan untuk langkah penyiapan data yang diklasifikasikan. Dataset yang diklasifikasikan terdiri dari 5 kategori, yaitu travel, otomotif, kuliner, bola dan politik. Setiap kategori terdiri dari 30 buah data, sehingga total berjumlah 150 data dokumen teks. Dari data hasil peringkasan pada 150 buah dokumen dengan masing-masing *compression rate* 10%, 20%, 30%, 40% dan 50%, dimasukkan ke dalam tahap *preprocessing* (*case transform, stopword removal, tokenizing, stemming*) kembali untuk memastikan kembali data telah siap diolah.

Data yang telah siap diujikan ke dalam algoritma K-NN dengan percobaan nilai *k* sebesar 3 sampai dengan 11 dan menggunakan pengukur jarak *Euclidean Distance* menggunakan persamaan 4 di atas. Hasil uji coba ini dapat dilihat pada tabel 2 berikut.

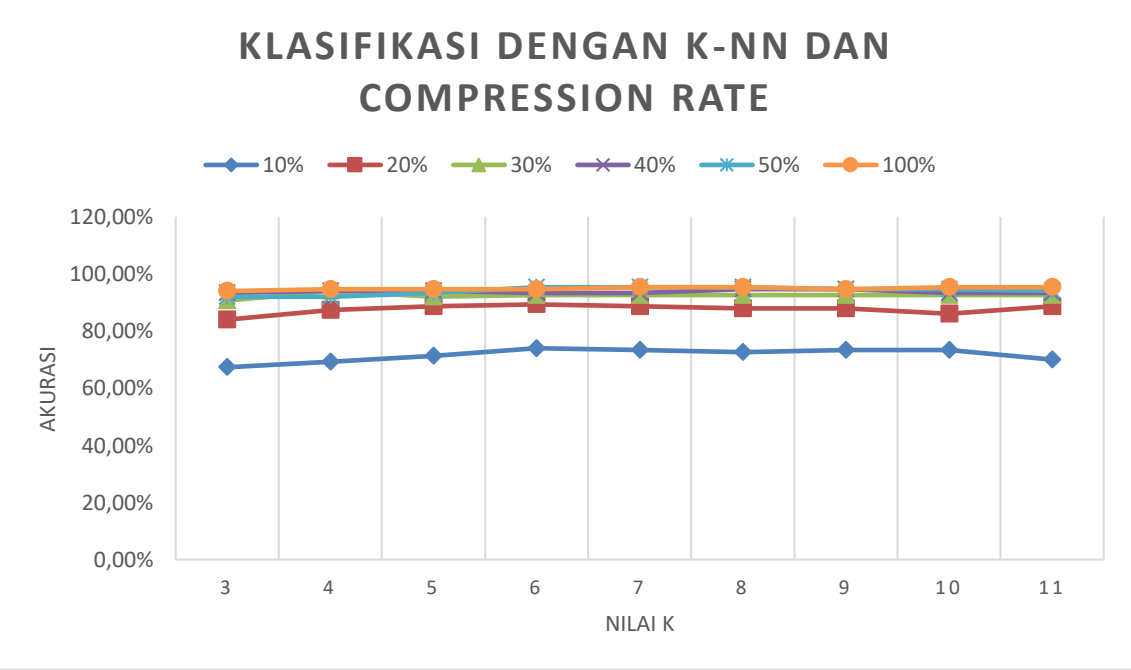
Tabel 3. Hasil Klasifikasi Dokumen Terringkas Menggunakan K-NN

No	K	Akurasi CR 10%	Akurasi CR 20%	Akurasi CR 30%	Akurasi CR 40%	Akurasi CR 50%	Akurasi CR 100%
1	3	67.33%	84.00%	90.67%	93.33%	92.00%	94.00%
2	4	69.33%	87.33%	**94.00%**	94.00%	92.00%	94.67%
3	5	71.33%	88.67%	92.00%	94.00%	93.33%	94.67%
4	6	**74.00%**	**89.33%**	92.67%	93.33%	**95.33%**	94.67%
5	7	73.33%	88.67%	92.67%	93.33%	**95.33%**	**95.33%**
6	8	72.67%	88.00%	92.67%	**94.67%**	**95.33%**	**95.33%**

7	9	73.33%	88.00%	92.67%	**94.67%**	94.67%	94.67%
8	10	73.33%	86.00%	92.67%	93.33%	94.67%	**95.33%**
9	11	70.00%	88.67%	92.67%	93.33%	94.00%	**95.33%**
Rerata		**71.63%**	**87.63%**	**92.52%**	**93.78%**	**94.07%**	**94.89%**

Pada tabel 2 di atas dapat dilihat bahwa nilai akurasi untuk dokumen yang diringkas dengan *compression rate* 10% memiliki akurasi tertinggi hanya sebesar 74 % pada nilai *k*=6. Pada *compression rate* 20% memiliki akurasi tertinggi sebesar 89.33% pada nilai *k*=6. Pada *compression rate* 30% memiliki akurasi tertinggi sebesar 94.00% pada nilai *k*=4. Pada *compression rate* 40% memiliki akurasi tertinggi sebesar 94.67% pada nilai *k*=8 dan *k*=9. Pada *compression rate* 50% memiliki akurasi tertinggi sebesar 95.33% pada nilai *k*=6, 7 dan 8. Sedangkan pada *compression rate* 100% atau dokumen utuh memiliki akurasi tertinggi sebesar 95.33% pada nilai *k*=7, 8, 10 dan 11.

Data pada tabel 2 di atas dapat divisualisasikan pada gambar 2 berikut ini untuk memudahkan dalam melihat performa peringkasan dokumen teks sebagai reduksi fitur untuk klasifikasi menggunakan algoritma K-NN.



Gambar 2. Klasifikasi hasil ringkasan dokumen teks dengan K-NN

Dari gambar 2 di atas dapat terlihat jelas bahwa semakin kecil *compression rate* maka semakin kecil juga tingkat akurasinya. Begitu juga sebaliknya, semakin besar *compression rate* maka semakin besar juga akurasinya. Hal ini menunjukkan informasi yang dibawa akan dipengaruhi oleh *compression rate* yang digunakan. Kemudian nilai *k* antara 3 sampai 11 pada algoritma K-NN cenderung stabil meskipun terdapat sedikit anomali tingkat akurasinya. *Compression rate* 30% sampai 50% memiliki perbedaan akurasi yang sedikit jika dibandingkan dengan *compression rate* sebesar 10% dan 20%.

Nilai *k* memiliki pengaruh terhadap hasil akurasi pengelompokan menggunakan algoritma K-NN. Nilai *k* antara 6 sampai 11 memiliki rerata akurasi tertinggi untuk masing-masing dataset terringkas. Nilai *k* antara 3 sampai 5 memiliki rerata akutasi terendah walaupun ada anomali, yaitu pada dokumen terringkas 30% dengan nilai *k* = 4 dengan akurasi sebesar 94%. Hal

ini dapat ditarik kesimpulan bahwa semakin kecil nilai *k* maka data yang masuk pada tetangga terdekat masih terhitung sedikit, yang mengakibatkan tidak terpresentasikan kelas yang sesungguhnya pada data uji. Sebaliknya, semakin besar nilai *k* akan memiliki kecenduran semakin banyak data yang masuk pada tetangga terdekat, yang berarti menggambarkan kelas sesungguhnya pada data uji. Selain itu, nilai *k* yang lebih besar juga memiliki kecenderungan menurunkan *noise* pada klasifikasi, sehingga akurasi akan semakin meningkat.

4. KESIMPULAN

Dari hasil penelitian ini dapt disimpulkan tingkat akurasi klasifikasi menggunakan algoritma K-NN antara dokumen dengan *compression rate* sebesar 50% dan dokumen utuh adalah sama besarnya pada nilai *k* tertentu. Hal ini berarti klasifikasi dengan menurunkan 50% fitur (*term*) pada dokumen bisa menyamai tingkat akurasinya dibandingkan dengan dokumen utuh. Dokumen dengan ringkasan sebesar 50% dengan nilai *k* = 6, 7 ataupun 8 bisa digunakan sebagai pengganti dokumen utuh, dimana tingkat akurasinya sama-sama memiliki tingkat akurasi sebesar 95.33% pada nilai *k* = 7, 8, 10 maupun 11. Dengan hasil ini maka teknik peringkasan dokumen teks dengan TF-IDF bisa dimanfaatkan sebagai alternatif reduksi fitur untuk klasifikasi, khususnya pada data dengan kategori tidak terstruktur seperti dokumen teks. Selain itu, dengan memanfaatkan peringkasan dokumen teks ini dapat memudahkan pengguna ataupun pembaca untuk dengan cepat menemukan intisari/informasi tanpa harus membaca keseluruhan dokumen. Hal ini berarti dapat mengurangi waktu baca yang, sehingga meningkatkan efisiensi dan efektifitas dalam hal membaca dokumen teks.

DAFTAR PUSTAKA

Abdel Fattah, M. (2015). New term weighting schemes with combination of multiple classifiers for sentiment analysis. *Neurocomputing*, *167*, 434–442. https://doi.org/10.1016/j.neucom.2015.04.051

Ajmal, E. B. (2015). Summarization of Malayalam Document Using Relevance of Sentences. *International Journal of Latest Research in Engineering and Technology*, *1*(6), 8–13.

Akromunnisa, K., & Hidayat, R. (2019). Klasifikasi Dokumen Tugas Akhir (Skripsi) Menggunakan K-Nearest Neighbor. *JISKA (Jurnal Informatika Sunan Kalijaga)*, *4*(1), 69. https://doi.org/10.14421/jiska.2019.41-07

Amalia, D. H., & Yustanti, W. (2021). Klasifikasi Buku Menggunakan Metode Support Vector Machine pada Digital Library. *Journal of Informatics and Computer Science (JINACS)*, *3*(01), 55–61. https://doi.org/10.26740/jinacs.v3n01.p55-61

Asril, H., Mustakim, M., & Kamila, I. (2019). Klasifikasi Dokumen Tugas Akhir Berbasis Text Mining menggunakan Metode Naïve Bayes Classifier dan K-Nearest Neighbor. *Seminar Nasional Teknologi Informasi Dan Industri*, *November*, 2579–5406.

Babar, S. A., & Patil, P. D. (2015). Improving Performance of Text Summarization. *Procedia Computer Science*, *46*(Icict 2014), 354–363. https://doi.org/10.1016/j.procs.2015.02.031

Chandani, V., Wahono, R. S., & Purwanto, . (2015). Komparasi Algoritma Klasifikasi Machine Learning Dan Feature Selection pada Analisis Sentimen Review Film. *Journal of Intelligent Systems*, *1*(1), 55–59. http://journal.ilmukomputer.org/index.php/jis/article/view/10

Dhande, L. L., & Patnaik, P. G. K. (2014). Analyzing Sentiment of Movie Review Data using Naive Bayes Neural Classifier. *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)*, *3*(4), 313–320. www.ijettcs.org

Erfani, S. M., Rajasegarar, S., Karunasekera, S., & Leckie, C. (2015). High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning. *Pattern Recognition*, *58*, 121–134. https://doi.org/10.1016/j.patcog.2016.03.028

Fachrurrozi, M., Yusliani, N., & Yoanita, R. U. (2013). Frequent Term based Text Summarization for Bahasa Indonesia. *International Conference on Innovations in Engineering and Technology*, 30–32. https://doi.org/10.15242/IIE.E1213550

Firdaus, F., Pasnur, P., & Wabdillah, W. (2019). Implementasi Cosine Similarity untuk Peningkatan Akurasi Pengukuran Kesamaan Dokumen pada Klasifikasi Dokumen Berita dengan K Nearest Neighbour. *Inspiration: Jurnal Teknologi Informasi Dan Komunikasi*, *9*(1), 69. https://doi.org/10.35585/inspir.v9i1.2496

García Adeva, J. J., Pikatza Atxa, J. M., Ubeda Carrillo, M., & Ansuategi Zengotitabengoa, E. (2014). Automatic text classification to support systematic reviews in medicine. *Expert Systems with Applications*, *41*(4), 1498–1508. https://doi.org/10.1016/j.eswa.2013.08.047

Ghiassi, M., Skinner, J., & Zimbra, D. (2013). Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network. *Expert Systems with Applications*, *40*(16), 6266–6282. https://doi.org/10.1016/j.eswa.2013.05.057

Guo, G., Wang, H., Bell, D., Bi, Y., & Greer, K. (2006). Using kNN model for automatic text categorization. *Soft Computing*, *10*(5), 423–430. https://doi.org/10.1007/s00500-005-0503-y

Hassani, H., Beneki, C., Unger, S., Mazinani, M. T., & Yeganegi, M. R. (2020). Text mining in big data analytics. *Big Data and Cognitive Computing*, *4*(1), 1–34. https://doi.org/10.3390/bdcc4010001

Hidayat, E. Y., & Rizqi, M. A. (2020). Klasifikasi Dokumen Berita Menggunakan Algoritma Enhanced Confix Stripping Stemmer dan Naïve Bayes Classifier. *Jurnal Nasional Teknologi Dan Sistem Informasi*, *6*(2), 90–99. https://doi.org/10.25077/teknosi.v6i2.2020.90-99

Hidayatullah, A. F. (2015). The Influence of Stemming on Indonesian Tweet Sentiment Analysis. *Proceeding of International Conference on Electrical Engineering, Computer Science and Informatics*, *August*, 19–20.

Lidya, S. K., Sitompul, O. S., & Efendi, S. (2015). Sentiment Analysis Pada Teks Bahasa Indonesia Menggunakan Support Vector Machine (Svm). *Seminar Nasional Teknologi Dan Komunikasi 2015*, *2015*(Sentika), 1–8.

Lloret, E., & Palomar, M. (2012). Text summarisation in progress: A literature review. *Artificial Intelligence Review*, *37*(1), 1–41. https://doi.org/10.1007/s10462-011-9216-z

Luthfiarta, A., Zeniarja, J., & Salam, A. (2013). Algoritma Latent Semantic Analysis (LSA) Pada Peringkas Dokumen Otomatis Untuk Proses Clustering Dokumen. *Seminar Nasional Teknologi Informasi & Komunikasi Terapan 2013 (SEMANTIK 2013)*, *2013*(November), 13–18.

Martín-Valdivia, M. T., Martínez-Cámara, E., Perea-Ortega, J. M., & Ureña-López, L. A. (2013). Sentiment polarity detection in Spanish reviews combining supervised and unsupervised approaches. *Expert Systems with Applications*, *40*(10), 3934–3942. https://doi.org/10.1016/j.eswa.2012.12.084

Mutrofin, S., Izzah, A., Kurniawardhani, A., & Masrur, M. (2014). Milton. *Jurnal Gamma*, *10*(1), 130–134. https://doi.org/10.1017/9781316534946.021

Najibullah, A. (2015). Indonesian Text Summarization based on Naïve Bayes Method. *International Seminar and Conference 2015 : The Golden Triangle (Indonesia-India-Tiongkok)*, 67–78.

Nanda, R., Haerani, E., Gusti, S. K., & Ramadhani, S. (2022). Klasifikasi Berita Menggunakan Metode Support Vector Machine. *Jurnal Nasional Komputasi Dan Teknologi Informasi (JNKTI)*, *5*(2), 269–278. https://doi.org/10.32672/jnkti.v5i2.4193

Nurhadi, A. (2015). Klasifikasi Konten Berita Digital Bahasa Indonesia Menggunakan Support Vector Machines (SVM) Berbasis Particle Swarm Optimization (PSO). *Jurnal Bianglala Informatika*, *3*(2), 1–9.

Rahayu, W. I., & Shafina, M. R. (2022). Aplikasi Analisis Kelayakan Sistem Untuk Pengukuran

UsabilityDengan Menerapkan Metode Use Questionnaire. *Jurnal Teknik Informatika*, *14*(3), 152.

Wibowo, E. D. (2014). Text Feature Weighting for Summarization of Documents Bahasa Indonesia by Using Binary Logistic. *International Journal of Computer Science and Telecommunications*, *5*(7).

Wijaya, A. P., & Santoso, H. A. (2016). Naïve Bayes Classification Pada Klasifikasi Dokumen Untuk Identifikasi Konten E-Government. *Journal of Applied Intelligent Systems*, *1*(1), 48–55.

Yadav, J., Winata, F., Rainarli, E., Informatika, T., Indonesia, U. K., Wibowo, E. D., Vinodhini, G., Varghese, R., Jayasree, M., Trstenjak, B., Mikac, S., Donko, D., Tian, F., Wu, F., Chao, K.-M., Zheng, Q., Shah, N., Lan, T., Yue, J., … Abdel Fattah, M. (2015). Automatic text classification to support systematic reviews in medicine. *Expert Systems with Applications*, *3*(1), 1498–1508. https://doi.org/10.1016/j.eswa.2013.08.047

Zhang, L., Jiang, L., Li, C., & Kong, G. (2015). Two feature weighting approaches for naive Bayes text classifiers. *Knowledge-Based Systems*, *100*, 137–144. https://doi.org/10.1016/j.knosys.2016.02.017