ANALISIS SENTIMEN DATA TWEET MENGGUNAKAN MODEL JARINGAN SARAF TIRUAN DENGAN PEMBOBOTAN DELTA TF-IDF

Chindy Amalia[1] , Yuliant Sibaroni[2]

[1,2]Fakultas Informatika, Universitas Telkom, Bandung
[1]chindyamalia@student.telkomuniversity.ac.id, [2]yuliant@telkomuniversity.ac.id

Abstrak

Kritik dan saran masyarakat Indonesia sangat berpengaruh untuk meningkatkan fasilitas dan kinerja pemerintah Indonesia. Salah satu media untuk menampung saran tersebut yaitu twitter dengan mengunggah tweet masyarakat dapat mengungkapkan keluh kesah mereka. Tetapi, dengan tweet yang berjumlah ratusan bahkan ribuan akan menyulitkan pemerintah untuk mengetahui kesimpulan dari seluruh data tweet. Data tweet yang diambil sebagai acuan yaitu data yang berisi tanggapan positif dan negatif dari masyarakat Indonesia. Oleh karena itu, penelitian ini mencoba menganalisis tweet mengenai sentimen masyarakat terhadap rencan perpindahan ibukota Indonesia. Analisis dilakukan dengan melakukan klasifikasi tweet yang berisi sentimen masyarakat terhadap rencana perpindahan ibukota Indonesia. Metode klasifikasi yang digunakan dalam penelitian ini adalah Jaringan Saraf Tiruan (JST) dengan model Multi Layer Perceptron yang dikombinasikan dengan fitur ekstraksi untuk dapat mendeteksi negasi dan pembobotan menggunakan Delta TF-IDF. Hasil pengujian pada aplikasi yang dibangun memperlihatkan bahwa akurasi memberikan tingkat akurasi yang cukup baik yaitu 67,75%.

Kata kunci: Analisis sentimen, Jaringan Saraf Tiruan, Multi Layer Perceptron,TF-IDF, Delta TF-IDF

Abstract

Criticism and suggestions from the Indonesian people are very influential to improve the facilities and performance of the Indonesian government. One of the media to accommodate the suggestion is that Twitter by uploading community tweets can express their complaints. However, with tweets that number in the hundreds or even thousands it will be difficult for the government to find the conclusions from all tweet data. The tweet data taken as a reference is data that contains positive and negative responses from the people of Indonesia. Therefore, this study tries to analyze a tweet about community sentiment towards the planned move of the Indonesian capital. The analysis was carried out by classification of tweets containing public sentiments towards the planned move of the capital Indonesia. The classification method used in this study is the Artificial Neural Network (ANN) with a Multi Layer Perceptron model combined with extraction features to be able to detect negation and weighting using Delta TF-IDF. The test results on the application built show that accuracy gives a fairly good level of accuracy that is 67.75%.

Keywords: sentiment analysis, artificial neural network, multi layer perceptron, TF-IDF, Delta TF-IDF

1. Pendahuluan

Media sosial telah digunakan oleh banyak masyarakat. Semakin banyak masyarakat menggunakan media sosial seperti twitter untuk menyediakan berbagai layanan dan berinteraksi dengan masyarakat lainnya[1]. Twitter adalah situs micro blogging dimana penggunanya dapat meninjau atau mengirimkan pesan berupa segala bentuk aspirasi dengan batasan 280 karakter per pesan yang biasa disebut tweet. Twitter dapat digunakan sebagai media untuk menyampaikan segala aspirasi untuk Pemerintah termasuk kritik atau dukungan terhadap rencana pemindahan Ibukota Indonesia. Twitter dapat diakses melalui website dan aplikasi pada telefon genggam[2].

Opini mining atau sentimen analisis adalah riset komputasional dari opini, sentimen dan emosi yang diekpresikan secara tekstual[3]. Perilaku ini membuat orang-orang menikmati berbagai kegiatan mereka di media social salah satunya twitter, termasuk mengeluh tentang kebijakan-kebijakan pemerintah. Namun akan memakan waktu yang sangat lama dan biaya yang mahal bila sentimen masyarakat di klasifikasikan secara manual[4].

Jaringan Saraf Tiruan dapat digunakan sebagai model jaringan untuk mengenali teks *tweet* resolusi, dengan metode Backpropagation dan *Term frequency – inverse document frequency* (TF-IDF) sebagai pembobotannya. Penggunaan TF-IDF bertujuan untuk membobotkan kata dengan menghitung nilai TF dan menghitung kemunculan sebuah kata di dalam sebuah dokumen dengan jumlah dokumen yang mengandung sebuah term yang sudah terpilih sehingga menghasilkan data vektor berupa matriks. Riset tersebut menunjukan hasil yang diperoleh akurasi model tersebut sebesar 50%, [5]. Selain TF-IDF, Delta-IDF merupakan salah satu pengembangan dari metode TF-IDF[6].

Maka, dalam penelitian ini melakukan analisis pembobotan kata Delta TF-IDF pada model jaringan saraf tiruan untuk untuk mengetahui performansi metode menggunakan model jaringan saraf tiruan, pada kasus ini menggunakan data sentimen masyarakat terhadap rencana pindahnya ibukota Indonesia. Metode Jaringan Saraf Tiruan (JST) dipilih karena metode tersebut merupakan salah satu metode klasifikasi yang meniru cara kerja otak manusia yang dapat menyelesaikan suatu permasalahan dengan cara pembelajaran (learning). Kelebihan JST salah satunya adalah kemampuannya dalam beradaptasi sehingga mampu belajar dari data masukan yang diberikan sehingga dapat memetakan/memodelkan hubungan antara masukan dan keluarannya [7].

Penelitian ini akan membahas mengenai hasil performansi dari dua fitur ekstaksi yang berbeda dari hasil *Data Crawling* dari *Twitter* mengenai rencana pemindahan ibukota Indonesia ke Kalimantan dengan menggunakan metode *Multi Layer Perceptron* dengan algoritma *Backpropagation* dan untuk mengetahui parameter yang dapat mempengaruhi nilai akurasi dari metode *Multi Layer Perceptron* dengan algoritma *Backpropagation.* Terdapat rumusan masalah pada penelitian ini yaitu untuk mengetahui pengaruh dari jumlah *Hidden layer, Learning rate, Dropout,* dan pengacakan data terhadap akurasi yang dihasilkan dalam menentukan opini positif dan negatif terhadap respon masyarakat Indonesia menggunakan metode *Multi Layer Perceptron. B*atasan masalah pada penelitian ini adalah *dataset* terdisi dari 2.000 data yang didapatkan dari *Twitter,* percobaan menggunakan 10 skenario berupa beberapa kombinasi antara *learning rate,dropout, hidden layer,* dan *epoch yang digunakan sebanyak 5 kali*.

Tujuan dari penelitian ini adalah mengetahui parameter yang digunakan untuk mendapatkan model yang optimal berdasarkan metode Jaringan Saraf Tiruan (JST). Serta mengetahui hasil analisis perbandingan pembobotan kata menggunakan metode Delta TF-IDF dan TF-IDF. Berikutnya pada bagian 2 membahas studi terkait pada penelitian ini, bagian 3 membahas perancangan sistem yang dilaksanakan pada penelitian ini, bagian 4 membahas evaluasi dan bagian 5 membahas kesimpulan pada penelitian ini.

2. Studi Terkait

Seiring dengan kemanjuan teknologi, media sosial telah menjadi salah satu kebutuhan masyarakat sehari-hari untuk mengunggah kegiatan, atau mengunggah aspirasi terhadap sekitarnya. Salah satu media sosial yang sering di gunakan oleh masyarakat adalah *twitter*, *Twitter* adalah aplikasi jejaring sosial yang memungkinkan orang untuk menulis teks singkat tentang berbagai topik. *Tweet* dari pengguna disebut sebagai microblogs karena ada batasan 280 karakter yang diberlakukan oleh *Twitter* untuk setiap *tweet*. Pada twitter memungkinkan pengguna menyajikan informasi apa saja hanya dengan beberapa kata, secara opsional pengguna dapat memasukan tautan sumber informasi yang lebih terperinci ke dalam *tweet. Twitter* dapat di akses melalui web atau aplikasi yang ada pada smartphone dengan menggunakan jaringan internet agar bisa terhubung satu sama lain. Tidak jarang *twitter* digunakan sebagai wadah untuk meluapkan aspirasi masyarakat terhadap pemerintah dan tidak sedikit pula memberikan informasi tentang sentimen terhadap postingan orang. Seperti penelitian pada analisis sentimen *twitter* untuk teks berbahasa Indonesia dengan Maximum Entropy dan Support Vector Machine yang dilakukan oleh Noviah [4].

Banyak orang ingin menganalisis terhadap sentimen masyarakat mengenai pemerintahan, yang di maksud analisis sentimen adalah studi komputasi pendapat, sentimen, dan emosi yang diungkapkan dalam teks [8]. Tugas dasar analisis sentimen adalah untuk menemukan ekspresi pada objek yang diberikan dan menentukan teks atau pendapat yang memiliki aspek positif atau negatif [6]. Tujuan dari sentimen analisis adalah memutuskan sikap atau pendapat dari seseorang terhadap suatu objek. Analisis sentimen berguna untuk menganalisis, yang dapat menentukan nilai kesukaan atau ketidaksukaan seseorang terhadap suatu objek. Nilai yang sering digunakan pada sentimen analisis yaitu positif dan negatif. Nilai ini dapat digunakan untuk dijadikan parameter dalam pengambilan keputusan.

Pada tahapan analisis sentimen terdapat proses *text mining*, yang dimaksud *teks mining* adalah teknik untuk melakukan klasifikasi dan menemukan pola yang kasat mata pada objek tertentu agar dapat digunakan untuk suatu tujuan [9]. Tugas utama dari text mining meliputi *Searching*, *Information Extraction*, *Categorization*, *Summarization*, *Prioritization*, *Clustering*, *Information Monitor* dan *Question* & *answers* [10], Tugas dari *text mining* pun merupakan tugas yang jauh lebih kompleks dari data mining karena *text mining* urusannya melibatkan data teks yang tidak terstruktur yang harus diubah menjadi data yang lebih terstruktur dan tahapan proses pada *teks mining* lebih banyak dibanding tahapan proses pada *data mining*. Untuk melakukan analisis pembuat keputusan, maka dibutuhkan data tidak terstruktur yang berjumlah besar dalam bentuk dokumen. Text mining bukanlah sebuah fungsi, akan tetapi kumpulan dari berbagai macam fungsi yang dikombinasikan dan disebut fungsi text mining [11].

Dalam text mining ada beberapa metode dalam melakukan pembobotan salah satunya menggunakan Term Frequency – Inverse Document Frequency (TF-IDF) yang di maksud TF-IDF menurut [12] Term Frequency Inverse Document Frequency (TFIDF) merupakan pembobot yang dilakukan setelah ekstrasi artikel berita. Proses metode TF-IDF adalah menghitung bobot dengan cara integrasi antara term frequency (tf) dan inverse document frequency (idf). Langkah dalam TF-IDF adalah untuk menemukan jumlah kata yang kita ketahui (tf) setelah dikalikan dengan berapa banyak artikel berita dimana suatu kata itu muncul (idf).

Pada saat ini Term Frequency – Inverse Document Frequency (TF-IDF) sudah banyak di kembangkan oleh para peneliti salah satunya adalah Delta Term Frequency – Inverse Document Frequency (Delta TF-IDF). Menurut [13], Delta TF-IDF digunakan sebagai pembanding dalam pencarian dokumen yang masih sering terjadi ketidakseusain data yang diberikan kepada pencari dokumen meski sistem menggunakan komputerisasi dengan TF-

IDF. Dokumen dinilai relevan bila dokumen berisi topik yang sama, atau berhubungan dengan subjek yang diteliti. Pembobotan kata dapat berpengaruh pada tingkat relevansi pencarian dokumen dan mempengaruhi kinerja sistem. Delta TF-IDF merupakan metode dimana TF-IDF dihitung dengan menggunakan dua kelas korpus. Pada kasus analisis sentimen Delta TF-IDF melakukan penilaian TF-IDF suatu term pada kumpulan dokumen yang memiliki label positif dan kumpulan dokumen yang memiliki label negatif [13].

Apabila sekedar pembobotan saja tidak cukup untuk mendapatkan hasil analisis sentimen maka dari itu pada penelitian ini menggunakan metode klasifikasi Jaringan Saraf Tiruan (JST) untuk melakukan klasifikasi tweet. Metode ini pada umumnya dapat digunakan dalam klasifikasi berbagai permasalahan dengan data non linear. JST memiliki kinerja yang baik dan banyak digunakan pada masalah visi komputer untuk pengenalan pola. Banyak peneliti yang melakukan klasifikasi dalam menyelesaikan penelitiannya dengan menggunakan metode ini. Seperti penelitian pada mengungkapkan pendapat positif atau negatif terhadap tinjauan film dan membandingkan beberapa metode yaitu ANN, HMM, dan SVM yang dilakukan oleh Jian dan tim. Hasil klasifikasi yang diperoleh menunjukkan metode ANN memiliki akurasi lebih tinggi dari pada *SVM* dan *HMM* pada suatu analisis ulasan film [14], dan penelitian pada pengklasifikasian sinyal egg berupa data gelombang dengan memadukan metode fuzzy dan metode modifikasi dari backpropagation yang dilakukan oleh Novitasari [15].

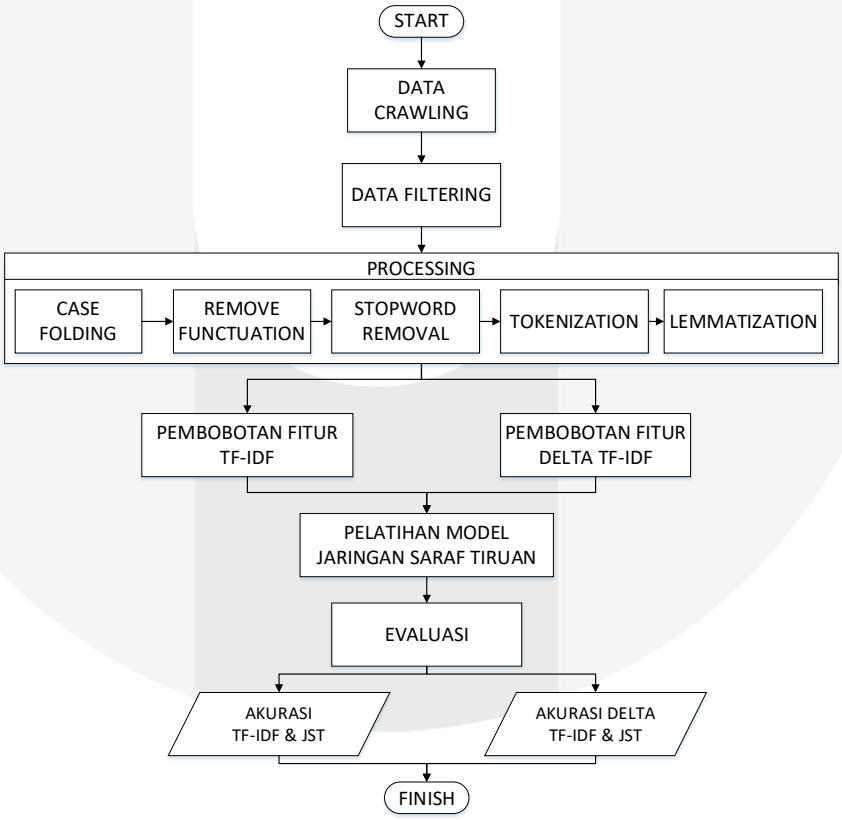
Selain itu metode Jaringan Saraf Tiruan pun digunakan untuk mengklasifikasi permukaan air guna memanajemen kualitas air dengan jaringan saraf tiruan MLP dalam proses klasifikasinya oleh Wechmongkhonkon dan tim [16]. Sebelum Wechmongkhonkon dan tim, ada peneliti lain yang pernah melakukan klasifikasi kualitas air yaitu Meair dan tim yang membedakan penelitian mereka, Meair dan tim cenderung pada model prediksi ketahanan kualitas air berdasarkan variabel permasalahannya[17]. Hal ini menunjukkan bahwa klasifikasi menggunakan metode ANN dinilai masih memberikan hasil yang cukup baik.

Penelitian lain juga menyebutkan bahwa untuk kasus analisis sentimen, metode ANN lebih unggul daripada SVM terutama untuk konteks data tidak seimbang pada 13 tes. Sedangkan SVM dapat mengungguli ANN hanya dengan 2 tes saja, meskipun perbedaan akurasi pada keduanya tidak pernah melebihi 3%. Penelitian lain juga menyebutkan bahwa untuk kasus analisis sentimen, metode ANN lebih unggul daripada SVM terutama untuk konteks data tidak seimbang pada 13 tes. Sedangkan SVM dapat mengungguli ANN hanya dengan 2 tes saja, meskipun perbedaan akurasi pada keduanya tidak pernah melebihi 3% [18].

3. Sistem yang Dibangun

3.1 Diagram Blok Sistem

Pada tahapan ini, akan dijelaskan proses yang terjadi pada sistem. Dapat dilihat pada diagram blok pada Gambar 1:



Gambar 1. Visualisasi Gambaran Umum Sistem Analisis Sentimen

3.2 *Data Crawling*

Data yang didapatkan dari twiiter sebanyak 5000 data yang didapatkan dari beberapa tagar yaitu #GrasaGrusu, #IbuKotaPindah, #IbuKotaBaru, #IbuKitaPindahUntukSiapa dan #PindahIbuKota tetapi pada data-data tersebut terlalu banyak *spam*, iklan, berita dsb. Maka data-data tersebut masuk ke tahap filtering yang dilakukan secara manual, di dapatkan 2000 data dari hasil filtering tersebut. Setelah dilakukan filtering 2000 data tersebut di labeli secara manual dengan label positif 1000 dan negatif 1000 data.

3.3 *Data Filtering*

Data yang dihasilkan dari Data Crawling selanjutnya diolah dengan cara manual untuk menghilangkan kalimat-kalimat yang tidak berhubungan dengan tagar yang telah ditentukan. Kalimat-kalimat yang hilang yaitu kalimat yang memiliki unsur iklan, berita, sara, dan lain-lain. Data Filtering ini juga berguna untuk meningkatkan akurasi agar tidak ada dokumen yang menyimpang

3.4 *Preprocessing*

Preprocessing yaitu tahap dimana data dari dokumen teks yang tidak terstruktur diolah menjadi data yang terstruktur. *Preprocessing* merupakan proses awal pada teks untuk mempersiapkan teks agar dapat menjadi data yang dapat diolah lebih lanjut . Dalam *preprocessing* terdapat beberapa tahapan yaitu sebagai berikut :

1. *Case Folding*
   Pada tahap ini, mengubah penggunaan huruf besar menjadi huruf kecil atau *lowercase*, apabila tidak dilakukan *case folding* maka data tidak akan konsisten dan mengalami akurasi yang buruk karena bisa jadi kata yang sama tetapi memiliki huruf kecil dan besar yang berbeda, akan diperlakukan berbeda. Contohnya sebagai berikut :
   - Sebelum : Saya dan keluarga setuju ibukota dipindahkan dari Jakarta ke Kalimantan !
   - Sesudah : saya dan keluarga setuju ibukota dipindahkan dari jakarta ke kalimantan !
2. *Remove Punctuation*
   Pada tahap ini, menghilangkan tanda baca dalam teks. Contohnya sebagai berikut :
   - Sebelum : saya dan keluarga setuju ibukota dipindahkan dari jakarta ke kalimantan !
   - Sesudah : saya dan keluarga setuju ibukota dipindahkan dari jakarta ke Kalimantan
3. *Tokenization*
   Tokenization merupakan tahap yang dilakukan setelah *Remove Punctuation*, dimana pada tahap ini merupakan proses memisahkan kalimat menjadi persuku kata atau pemotongan *String input* berdasarkan tiap kata yang menyusunnya. *Tokenization* juga akan memecah sekumpulan karakter dalam suatu teks ke dalam suatu kata.
4. *Stopword Removal*
   Stopword Removal merupakan proses penghilangan kata yang sering muncul pada dokumen, karena tidak memberikan ciri yang signifikan untuk membedakan dokumen satu dengan yang lain. Contohnya adalah "dan" atau "sebuah". Karena pada faktanya kata yang mencirikan suatu dokumen biasanya frekuensi kemunculannya jarang. Sehingga untuk mempersingkat proses klasifikasi, kita dapat menghilangkan S*topword* tersebut. *Stopword* yang digunakan berasal dari *library Sastrawi.*
5. *Lemmatization*
   Lemmatization adalah proses untuk menemukan bentuk dasar dari sebuah kata atau pengubah menjadi kata dasar, *library* yang digunakan untuk mendapatkan bentuk dasar tersebut adalah *library Sastrawi* yang dibuat oleh Universitas Indonesia.

3.5 *Pembobotan Fitur Term frequency – inverse document frequency (TF-IDF)*

Pada tahap ini, dokumen dataset menggunakan proses pembobotan TF-IDF. *Term frequency – inverse document frequency* atau biasa sering disebut TF-IDF adalah metode pembobotan kata dengan menghitung nilai TF dan juga menghitung kemunculan sebuah kata pada koleksi dokumen teks secara keseluruhan [20].Metode ini menggabungkan 2 konsep perhitungan bobot yaitu frekuensi kemunculan sebuah kata di dalam sebuah dokumen tertentu dan inverse frekuensi dokumen yang mengandung kata tersebut [21]. Inverse document frequency (IDF) adalah jumlah dokumen yang mengandung sebuah term didasarkan pada seluruh dokumen yang ada pada dataset. Berikut rumus yang digunakan pada *inverse document frequency (IDF)* :

$$idf = log(\frac{n}{df_i}) \quad (1)$$

Keterangan :

- idf = Invers Document Frequency (IDF)
- n = Banyaknya Dokumen
- dfi = Banyaknya Dokumen yang memiliki term t

Kemudian untuk proses pembobotan dari term yang ada menggunakan rumus sebagai berikut :

$$w_{t.d} = tf_{t.d} . log \frac{n}{dfi} \quad (2)$$

Keterangan:

- $w_{t.d}$ = Nilai delta TF_IDF untuk term t pada dokumen d
- $tf_{t.d}$ = Jumlah term t yang muncul pada dokumen d
- $log \frac{n}{dfi}$ = Invers Document Frequency (IDF)
- n = Banyaknya Dokumen
- dfi = Banyaknya Dokumen yang memiliki term t

Contoh Perhitungan:

d1 : saya setuju ibukota pindah
d2 : pindahnya ibukota bukan urgensi
d3 : setuju, semoga dilancarkan walaupun memakan biaya
d4 : pemindahan akan memakan biaya

Tabel 1. Perhitungan Term frequency - inverse document frequency (TF-IDF)

term	*tf*				*df*	$\frac{n}{dfi}$	$log \frac{n}{dfi}$	$log \frac{n}{dfi} + 1$	$w = tf * log \frac{n}{dfi} + 1$			
	d1	**d2**	**d3**	**d4**					**d1**	**d2**	**d3**	**d4**
setuju	1	0	1	0	2	2	0.30103	1.30103	1.30103	0	1.30103	0
ibukota	1	1	0	0	2	2	0.30103	1.30103	1.30103	0	0	0
pindah	0	1	0	1	2	2	0.30103	1.30103	0	1.30103	0	1.30103
urgensi	0	1	0	0	1	4	0.60206	1.60205999	0	1.60206	0	0
semoga	0	0	1	0	1	4	0.60206	1.60205999	0	0	1.60206	0
lancar	0	0	1	0	1	4	0.60206	1.60205999	0	0	1.60206	0
makan	0	0	1	1	2	2	0.30103	1.30103	0	0	1.30103	1.30103
biaya	0	0	1	1	2	2	0.30103	1.30103	0	0	1.30103	1.30103

3.6 *Delta Term frequency – inverse document frequency*

Delta Term Frequency - Inverse Document Frequency (Delta TF-IDF) merupakan metode dimana TF-IDF dihitung dengan menggunakan dua kelas korpus. Pada kasus analisis sentimen delta TF-IDF melakukan penilaian TF-IDF suatu term pada kumpulan dokumen yang memiliki label positif dan kumpulan dokumen yang memiliki label negative [13]. Rumus Delta TF-IDF dapat didefinisikan sebagai berikut :

$$W_{(t,d)} = C_{(t,d)} * log\left(\frac{|P|}{P_t}\right) - C_{(t,d)} * log\left(\frac{|N|}{N_t}\right) \quad (3)$$

Keterangan :

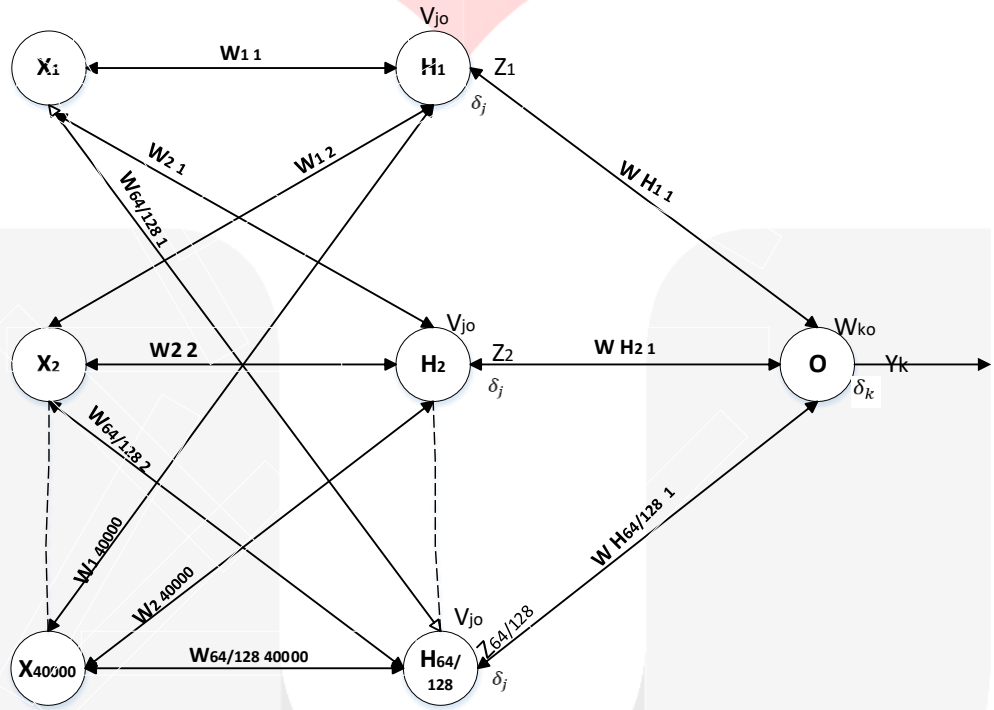
- $w_{t.d}$ = Nilai delta TF_IDF untuk term t pada dokumen d
- $tf_{t.d}$ = Jumlah term t yang muncul pada dokumen d
- $|P|$ = Jumlah dokumen dengan label positif
- P_t = Jumlah dokumen pada label positif yang memiliki term t
- $|N|$ = Jumlah dokumen dengan label negative
- N_t = Jumlah dokumen pada label negative yang memiliki term t

Pada penelitian ini Delta TF-IDF digunakan sebagai pembanding pembobotan yang pada awalnya menggunakan TF-IDF biasa, hal ini dikarenakan delta TF-IDF membandingkan nilai TF-IDF pada dokumen sentimen yang telah dilabeli sehingga dapat memberikan nilai perbedaan antara term positif dan negatif sehingga Delta TF-IDF lebih cocok digunakan dibandingkan TF-IDF [13].

Tabel 2. Perhitungan Delta Term frequency - inverse document frequency (TF-IDF)

term	tf				P_t	N_t	$log\left(\frac{\|P\|}{P_t}\right)$	$log\left(\frac{\|N\|}{N_t}\right)$	$w_{td} = tf_{td} * log\left(\frac{\|P\|}{P_t}\right) - tf_{td} * log\left(\frac{\|N\|}{N_t}\right)$			
	d1	**d2**	**d3**	**d4**					**d1**	**d2**	**d3**	**d4**
setuju	1	0	1	0	2	0	0	0	0	0	0	0
ibukota	1	1	0	0	1	1	0.30103	0.30103	0	0.30103	0	0
pindah	0	1	0	1	0	2	0	0	0	0	0	0
urgensi	0	1	0	0	0	1	0	0.30103	0	0.30103	0	0
semoga	0	0	1	0	1	0	0.30103	0	0	0	0	0
lancar	0	0	1	0	1	0	0.30103	0	0	0	0	0
makan	0	0	1	1	1	1	0.30103	0.30103	0	0	0	0
biaya	0	0	1	1	1	1	0.30103	0.30103	0	0	0	0

3.7 Pelatihan Model Jaringan Saraf Tiruan



Gambar 2. Visualisasi Multi Layer Perceptron

Berdasarkan Gambar 2, Multi Layer Perceptron (MLP) adalah arsitektur jaringan yang memiliki banyak lapisan. MLP memiliki satu atau lebih hidden layer [7], Pada penelitian ini menggunakan 10 skenario dengan *Input Layer* sebanyak 40.000, dengan 2 kombinasi *hidden layer* yaitu 64 dan 128, serta 4 kombinasi *learning rate* yaitu 0.0001, 0.001, 0.01, 0.1. Salah satu algoritma pelatihan MLP adalah Backpropagation (Propagasi Balik). Cara kerja Algoritma Backpropagation adalah mengoreksi kesalahan dan memperbaikinya . Algoritma Backpropagation dilakukan dalam 3 tahap yaitu propagasi maju, propagasi mundur, dan perubahan bobot..

Multi Layer Perceptron (*MLP*) adalah arsitektur jaringan yang memiliki banyak lapisan. *MLP* memiliki satu atau lebih *hidden layer*[7]. Salah satu algoritma pelatihan *MLP* adalah *Backpropagation* (Propagasi Balik). Cara kerja Algoritma *Backpropagation* adalah mengoreksi kesalahan dan memperbaikinya [22]. Algoritma *Backpropagation* dilakukan dalam 3 tahap yaitu propagasi maju, propagasi mundur, dan perubahan bobot. Berikut adalah algoritma *Backpropagation* [7]:

1. Mendefinisikan matriks *Input*, *Hidden*, *Output*.
2. Menginisialisasikan arsitektur jaringan, *Learning rate*, dan nilai bobot melalui nilai acak dengan interval nilai sembarang dengan interval [0, 1].
3. Pelatihan Jaringan
   - Propagasi Maju

Dengan bobot yang telah diacak pada tahap ke dua. Menghitung keluaran dari *Hidden layer* berdasarkan persamaan berikut (menggunakan fungsi aktivasi *Relu*) [23]:

$$Z_{net_j} = v_{jo} + \sum_{i=1}^{n} \quad x_i v_{ji} \tag{3}$$

$$z_j = \left(z_{net_j}\right) = max(0, z_{net_j}) \tag{4}$$

Hasil keluaran *Hidden layer* (z_j) digunakan untuk mendapatkan nilai *Output layer* dengan persamaan berikut (menggunakan fungsi aktivasi *Sigmoid*) [23]:

$$y_{net_k} = w_{ko} + \sum_{j=1}^{p} \quad z_j \, w_{kj} \tag{5}$$

$$y_k = f(y_{net_k}) = \frac{1}{1+e^{-ynet_k}} \tag{6}$$

- Propagasi Mundur
  Melakukan perhitungan factor δ unit *Output* berdasarkan *Error* pada setiap unit *Output* dengan persamaan berikut[23]:

$$\delta_k = (t_k - y_k)\, f'(y\_net_k) = (t_k - y_k) y_k (1 - y_k) \tag{7}$$

Selanjutnya melakukan perhitungan suku perubahan bobot w_{kj} dengan laju percepatan a dengan persamaan berikut[23]:

$$\Delta w_{kj} = a \,.\, \delta_k \,.\, z_j \tag{8}$$

Kemudian melakukan perhitungan δ unit tersembunyi berdasar pada error pada setiap unit tersembunyi dengan persamaan berikut[23]:

$$\delta\_net_j = \sum_{k=1}^{m} \quad \delta_k \, w_{kj} \tag{9}$$

Faktor δ unit tersembunyi dapat dipresentasikan dengan persamaan berikut[23]:

$$\delta_j = \delta\_net_j \, f'(Z_{net_j}) = \delta\_net_j \, z_j \tag{10}$$

Kemudian melakukan perhitungan suku perubahan bobot v_{ji} dengan persamaan berikut[23]:

$$\Delta v_{ji} = a \,.\, \delta_j \,.\, x_i \tag{11}$$

- Perubahan Bobot
  Melakukan perhitungan seluruh perubahan bobot yang menuju unit output dengan persamaan berikut[23]:

$$w_{kj}\,(baru) = w_{kj}\,(lama) + \Delta w_{kj} \tag{11}$$

Melakukan perhitungan seluruh perubahan bobot yang menuju unit tersembunyi dengan persamaan berikut[23]:

$$v_{ji}\,(baru) = v_{ji}\,(lama) + \Delta v_{ji} \tag{12}$$

4. Tahapan diatas adalah untuk satu kali siklus pelatihan. Pelatihan harus diulang-ulang hingga jumlah siklus tertentu.

3.8 Evaluasi

3.8.1 *K-Fold Cross Validation*

K-Fold Cross Validation digunakan untuk memvalidasi model yang telah dibangun. Jenis *Cross Validation* yang digunakan pada penelitian ini adalah *5-Fold Cross Validation* dengan cara membagi *Data testing* dan *Data*

training dengan perbandingan 2:8. Dengan menggunakan *5-Fold Cross Validation* diharapkan dapat menghasilkan nilai akurasi terbaik.

3.8.2 *Confusion Matrix*

Tabel 3. Visualisasi *Confusion Matrix*

#	PREDIKSI	
REALITA	*TP*	*FN*
	FP	*TN*

Berdasasarkan Tabel 1, *Confusion Matrix* dipilih sebagai metode validasi karena data yang diperoleh memiliki jumlah opini positif dan negatif yang tidak seimbang.

Terdapat 4 istilah pada *Confusion Matrix* yaitu:

1. TP adalah *True Positive*, yaitu jumlah data positif yang terklasifikasi dengan benar oleh sistem.
2. TN adalah *True Negative*, yaitu jumlah data negatif yang terklasifikasi dengan benar oleh sistem.
3. FN adalah *False Negative*, yaitu jumlah data negatif namun terklasifikasi salah oleh sistem.
4. FP adalah *False Positive*, yaitu jumlah data positif namun terklasifikasi salah oleh sistem

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} * 100\% \quad (13)$$

$$Presisi = \frac{TP}{FP+TP} * 100\% \quad (14)$$

$$Recall = \frac{TP}{FN+TP} * 100\% \quad (15)$$

Berdasarkan nilai *True Negative (TN), False Positive (FP), False Negative (FN),* dan *True Positive (TP)* dapat diperoleh nilai akurasi, presisi dan *recall*. Nilai akurasi menggambarkan seberapa akurat sistem dapat mengklasifikasikan data secara benar. Dengan kata lain, nilai akurasi merupakan perbandingan antara data yang terklasifikasi benar dengan keseluruhan data. Nilai presisi menggambarkan jumlah data kategori positif yang diklasifikasikan secara benar dibagi dengan total data yang diklasifikasi positif. Sementara itu, *recall* menunjukkan berapa persen data kategori positif yang terklasifikasikan dengan benar oleh sistem[24].

4. Evaluasi

4.1 Hasil Pengujian

Pengujian menggunakan metode *K-Fold Cross Validation* dengan *K* yang berjumlah 5. Setiap *Fold* terdapat 10 skenario. Setiap skenario memiliki jumlah *Learning Rate, Hidden Layer, Drop Out, Shuffle* yang berbeda-beda. Tabel berikut adalah hasil rata-rata dari seluruh fold pada setiap model:

Tabel 4. Hasil Rata-Rata Pengujian TFIDF dan ANN

Model	Learning Rate	Data Shuffling	Hidden Layer	Dropout	Output Layer	Train Accuracy	Test Accuracy	Recall	Precission	F-Measure	Accuracy
ANN_1	0.0001	O	64	0.2	1	50.1245	50.78404	15.09749	66.847042	17.929725	50.78404
ANN_2	0.0001	O	128	0.2	1	48.67544	50.88317	16.99078	79.675737	15.671273	50.88317
ANN_3	0.0001	O	64	X	1	94.99024	59.31945	74.65155	59.200967	63.828879	59.31945
ANN_4	0.0001	O	128	X	1	97.21402	62.41571	61.91481	65.566625	60.119768	62.41571
ANN_5	0.001	O	64	0.2	1	49.83714	57.23192	18.56205	85.998668	28.455411	57.23192
ANN_6	0.001	O	128	0.2	1	51.34911	57.43229	19.10936	87.873732	30.365615	57.43229
ANN_7	0.001	O	64	X	1	99.75012	67.96671	65.18823	68.659135	65.840818	67.96671
ANN_8	0.001	O	128	X	1	99.68766	66.37008	65.07546	69.144992	64.943057	66.37008
ANN_9	0.01	O	64	0.2	1	89.54284	64.47195	36.37443	82.833954	49.952982	64.47195

ANN_10	0.01	O	128	0.2	1	90.05515	66.47207	42.72826	80.283088	55.269546	66.47207
ANN_11	0.01	O	64	X	1	99.75013	67.26608	72.33804	66.600498	67.956044	67.26609
ANN_12	0.01	O	128	X	1	99.86256	69.36571	65.30779	69.637261	67.022603	69.36571
ANN_13	0.1	O	64	0.2	1	89.06812	67.41521	74.67705	64.298215	68.51089	67.41521
ANN_14	0.1	O	128	0.2	1	89.79273	68.96496	71.18188	66.491682	68.122251	68.96496
ANN_15	0.1	O	64	X	1	99.78761	67.91459	42.84999	80.638075	52.335538	67.91459
ANN_16	0.1	O	128	X	1	99.78761	69.01646	58.78961	73.346452	63.543822	69.01646

Tabel 5. Hasil Rata-Rata Pengujian Delta TFIDF dan ANN

Model	Learning Rate	Data Shuffling	Hidden Layer	Dropout	Output Layer	Train Accuracy	Test Accuracy	Recall	Precission	F-Measure	Accuracy
ANN_1	0.0001	O	64	0.2	1	50.1245	52.03354	20.11421	68.603989	24.464981	52.03354
ANN_2	0.0001	O	128	0.2	1	48.67544	52.18167	20.1468	84.084034	20.914535	52.18167
ANN_3	0.0001	O	64	X	1	95.62743	58.9192	71.43981	59.449522	62.536764	58.9192
ANN_4	0.0001	O	128	X	1	97.8262	61.26758	58.24223	65.059259	58.04454	61.26758
ANN_5	0.001	O	64	0.2	1	49.79964	58.6803	22.58404	84.591091	33.594397	58.6803
ANN_6	0.001	O	128	0.2	1	51.53659	58.83067	22.0599	88.206512	34.575181	58.83067
ANN_7	0.001	O	64	X	1	99.8001	67.76721	62.39371	69.112602	64.814562	67.76721
ANN_8	0.001	O	128	X	1	99.83758	65.5207	63.43633	68.808235	63.79875	65.5207
ANN_9	0.01	O	64	0.2	1	89.6803	62.92282	32.60086	82.538737	46.09276	62.92282
ANN_10	0.01	O	128	0.2	1	90.09264	63.87369	35.4236	82.130661	48.897433	63.87369
ANN_11	0.01	O	64	X	1	99.81259	66.06671	71.61004	66.060491	66.969462	66.06671
ANN_12	0.01	O	128	X	1	99.87506	68.41571	64.37347	69.128373	66.019612	68.41571
ANN_13	0.1	O	64	0.2	1	89.1306	67.21534	74.13036	63.22622	68.004268	67.21534
ANN_14	0.1	O	128	0.2	1	88.36857	66.41509	73.23761	63.99115	67.223197	66.41509
ANN_15	0.1	O	64	X	1	99.73764	64.01509	33.01365	83.375142	40.792459	64.01509
ANN_16	0.1	O	128	X	1	99.70017	70.01409	62.47787	70.680858	64.870107	70.01409

4.2 Analisis Hasil Pengujian

Nilai akurasi yang didapat dari kedua metode tersebut tidak mendapatkan nilai yang cukup baik karena beberapa faktor, yaitu:

1. Jumlah data yang hanya 2000
2. Kalimat yang digunakan terlalu bebas
3. Pelabelan yang tidak akurat

Walaupun dengan nilai akurasi yang didapatkan kedua metode tersebut tidak mendapatkan nilai akurasi yang memuaskan, tetapi dapat disimpulkan bahwa metode Delta TFIDF mendapatkan nilai akurasi yang lebih tinggi dari metode TFIDF biasa. Skenario ke 12, mendapatkan nilai akurasi tertinggi pada kedua metode pembobotan. Dengan *Learning Rate* 0.01, Jumlah *Hidden Layer* 128, dan tidak menggunakan *Dropout*, mendapatkan nilai akurasi 99.85 pada pembobotan fitur TFIDF dan 99.81% pada pembobotan fitur Delta TFIDF.

Nilai akurasi pada data *testing* pada masing-masing pembobotan fitur memiliki skenario yang berbeda, pada TFIDF, Skenario ke 13 dengan *Learning Rate* 0.1, *Hidden Layer* 64, dan menggunakan *Dropout* mendapatkan hasil 68.56%. Sedangkan pada Delta TFIDF, Skenario 14 dengan *learning rate* 0.1, *hidden layer* 128, dan menggunakan *dropout*, mendapatkan hasil 70.61%. Nilai *Recall* pada masing-masing pembobotan fitur juga memiliki skenario yang berbeda, pada TFIDF, Skenario 14 dengan *Learning Rate* 0.1, H*idden Layer* 128, dan menggunakan *Dropout*, mendapatkan hasil 81.32%. Sedangkan pada Delta TFIDF, Skenario 3 dengan *Learning Rate* 0.0001, *Hidden Layer* 64, dan tidak menggunakan *Dropout*, mendapatkan hasil 74.6%.Nilai Presisi pada masing-masing pembobotan fitur juga memiliki skenario yang berbeda, pad TFIDF, Skenario 6 dengan *Learning Rate* 0.001, *Hidden Layer* 128, dan

Dropout mendapatkan hasil 85.25%. Sedangkan pada Delta TFIDF, Skenario 5 dengan *Learning Rate* 0.001, *Hidden Layer* 64, dan Dropout mendapatkan hasil 86%.

Nilai akurasi pada pada masing-masing pembobotan fitur memiliki skenario yang berbeda, pada TFIDF, Skenario ke 13 dengan *Learning Rate* 0.1, *Hidden Layer* 64, dan menggunakan *Dropout* mendapatkan hasil 68.56%. Sedangkan pada Delta TFIDF, Skenario 14 dengan *Learning Rate* 0.1, *Hidden Layer* 128, dan menggunakan *Dropout*, mendapatkan hasil 70.61%. Dari seluruh hasil pengujian dapat diketahui bahwa jumlah *Hidden layer, Learning rate, Drop out,* dan pengacakan data berpengaruh terhadap nilai akurasi. Besarnya *Hidden layer* tidak menjamin akurasi menjadi lebih baik dibandingkan dengan jumlah *Hidden layer* yang lebih sedikit. Perbandingan opini negatif dan positif pada *dataset* sangat berpengaruh terhadap hasil pengelompokan opini positif dan negatif.

5. Kesimpulan dan Saran

Berdasarkan seluruh hasil pengujian dan analisis yang telah dilakukan pada penelitian ini, maka dapat ditarik kesimpulan dan saran sebagai berikut:

5.1 Kesimpulan

Semakin tinggi nilai *Learning Rate* semakin besar nilai akurasi dapat dilihat dari hasil seluruh Skenario. Pembobotan Delta TFIDF lebih baik dibandingkan dengan TFIDF biasa, terlihat dari hasil akurasi seluruh skenario, Delta TFIDF mendapatkan hasil akurasi tertinggi yaitu 70.6% dan TFIDF sebesar 68.5%. Besarnya *Hidden layer* tidak menjamin akurasi menjadi lebih baik dibandingkan dengan jumlah *Hidden layer* yang lebih sedikit.

5.2 Saran

1. Proses menganalisis data memakan waktu yang sangat lama, sehingga akan lebih baik apabila menggunakan *platform* yang lain sebagai *compiler*. (Contoh: *MatLab*)
2. Akurasi yang dihasilkan pada penelitian ini belum cukup baik, maka disarankan untuk menggunakan lebih banyak data dan lebih bervariasi sehingga hasil yang diperoleh akan lebih baik.
3. Penelitian selanjutnya dapat mencoba menerapkan seleksi fitur pada algoritma klasifikasi ini. Algoritma seleksi fitur yang digunakan pun bisa bermacam-macam. (Contoh: Fast Correlation Based Filter).

Daftar Pustaka

[1] W. He, S. Zha, and L. Li, "Social media competitive analysis and text mining: A case study in the pizza industry," *Int. J. Inf. Manage.*, vol. 33, no. 3, pp. 464–472, 2013.

[2] S. A. Phand and J. A. Phand, "Twitter sentiment classification using stanford NLP," *Proc. - 1st Int. Conf. Intell. Syst. Inf. Manag. ICISIM 2017*, vol. 2017-Janua, pp. 1–5, 2017.

[3] B. Liu, "LIBRO_NLP-handbook-sentiment-analysis," pp. 1–38, 2010.

[4] N. Dwi Putranti and E. Winarko, "Analisis Sentimen Twitter untuk Teks Berbahasa Indonesia dengan Maximum Entropy dan Support Vector Machine," *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 8, no. 1, pp. 91–100, 2014.

[5] L. J. Handayani, D. Purwitasari, and A. Navastara, "Tweet Resolusi," vol. 4, no. 1, 2015.

[6] T. Nasukawa and J. Yi, "Sentiment analysis: Capturing favorability using natural language processing," *Proc. 2nd Int. Conf. Knowl. Capture, K-CAP 2003*, pp. 70–77, 2003.

[7] S. T. M. S. Dr. Suyanto, *Data Mining*. Informatika Bandung, 2017.

[8] J. LING, I. P. E. N. KENCANA, and T. B. OKA, "Analisis Sentimen Menggunakan Metode Naïve Bayes Classifier Dengan Seleksi Fitur Chi Square," *E-Jurnal Mat.*, vol. 3, no. 3, p. 92, 2014.

[9] Indriati and Ridok, "Sentiment analysis For Review Mobile Applications Using Neighbor Method Weighted K-Nearest Neighbor (NWKNN)," *J. Environ. Eng. Sustain. Technol.*, 2016.

[10] A. Mustafa, A. Akbar, and A. Sultan, "Knowledge Discovery using Text Mining: A Programmable Implementation on Information Extraction and Categorization," *Int. J. Multimed. Ubiquitous Eng.*, vol. 4, no. 2, pp. 183–188, 2009.

[11] D. J. Haryanto, L. Muflikhah, and M. A. Fauzi, "Analisis Sentimen Review Barang Berbahasa Indonesia Dengan Metode Support Vector Machine Dan Query Expansion," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 9, pp. 2909–2916, 2018.

[12] S. Asiyah and K. Fithriasari, "Klasifikasi Berita Online Menggunakan Metode Support Vector Machine Dan K-Nearest Neighbor," *J. Sains dan Seni ITS*, vol. 5, no. 2, 2016.

[13] J. Martineau *et al.*, "Delta TFIDF: An Improved Feature Space for Sentiment Analysis," *Proc. Second Int. Conf. Weblogs Soc. Media (ICWSM*, vol. 29, no. May, pp. 490–497, 2008.

[14] J. Zhu, C. Xu, and H. S. Wang, "Sentiment classification using the theory of ANNs," *J. China Univ. Posts Telecommun.*, vol. 17, no. SUPPL. 1, pp. 58–62, 2010.

[15] D. Novitasari, "Klasifikasi Sinyal EEG Menggunakan Metode Fuzzy C-Means Clustering (FCM) Dan

Adaptive Neighborhood Modified Backpropagation (ANMBP)," *Mantik J. Mat.*, vol. 1, p. 31, Nov. 2015.
[16] E. Phil A. and G. Jim, "Application of Artificial Neural Network to Chromosome Classification," *Cytometry*, vol. 14, pp. 627–639, 1993.
[17] H. Maier and G. Dandy, "Neural networks for the prediction and forecasting of water resources variables: A review of modelling issues and applications," *Environ. Model. Softw.*, vol. 15, pp. 101–124, Jan. 2000.
[18] R. Moraes, J. F. Valiati, and W. P. Gavião Neto, "Document-level sentiment classification: An empirical comparison between SVM and ANN," *Expert Syst. Appl.*, vol. 40, no. 2, pp. 621–633, 2013.
[19] P. I. Lesmana, "Analisis sentimen ..., Pekik Indra Lesmana, Fasilkom UI, 2013," 2013.
[20] A. Achmad and A. A. Ilham, "Implementasi Algoritma Term Frequency – Inverse Document Frequency dan Vector Space Model untuk Klasifikasi Dokumen Naskah Dinas," vol. 257, pp. 88–92, 2012.
[21] A. A. Maarif, "Penerapan Algoritma TF-IDF untuk Pencarian Karya Ilmiah," *Dok. Karya Ilm. | Tugas Akhir | Progr. Stud. Tek. Inform. - S1 | Fak. Ilmu Komput. | Univ. Dian Nuswantoro Semarang*, no. 5, p. 4, 2015.
[22] Yusran, "IMPLEMENTASI JARINGAN SYARAF TIRUAN (JST) UNTUK MEMPREDIKSI HASIL NILAI UN MENGGUNAKAN METODE BACKPROPAGATION," 2016.
[23] Y. D. Lestari, "Jaringan syaraf tiruan untuk prediksi penjualan jamur menggunakan algoritma backropagation," *J. ISD*, vol. 2, no. 1, pp. 40–46, 2017.
[24] I. Menarianti, "Klasifikasi Data Mining Dalam Menentukan Pemberian Kredit Bagi Nasabah Koperasi," *Ilm. Teknosains*, vol. 1, no. 1, pp. 36–45, 2015.