

Implementasi Algoritma Pembelajaran Backpropagation Untuk Klasifikasi Tweet Resolusi

Linggar Juwita Handayani, Diana Purwitasari, dan Dini Adni Navastara
Teknik Informatika, FTIF, Institut Teknologi Sepuluh Nopember (ITS)
Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia
e-mail: diana@if.its.ac.id

Abstrak—Satu Januari mungkin akan datang dan pergi seperti yang terjadi pada tahun-tahun sebelumnya. Sosial media akan dibanjiri resolusi Tahun Baru pengguna. Pengguna secara masal berbagi harapan pribadi untuk meningkatkan kualitas diri menjadi lebih baik. Terbukti dengan kelas-kelas di pusat olahraga menjadi mendadak penuh dan terjual habis. Begitu juga jumlah puntung rokok pada asbak menjadi lebih sedikit di awal tahun. Beragamnya tweet resolusi yang dibagikan pengguna, maka dari itu dilakukan klasifikasi tweet resolusi ke dalam kelas sejenis.

Pada penelitian ini metode yang akan diimplementasikan adalah klasifikasi multi kelas dengan Algoritma Backpropagation, dan optimasi *gradient descent*. Model jaringan yang digunakan adalah Multi-Layer Perceptron dengan *neuron input* sebanyak jumlah fitur, dan *neuron output* adalah enam, di mana masing-masing *neuron* mewakili sebuah kelas. Di antaranya enam kelas tersebut adalah *Career & Education, Finance, Health & Fitness, Personal Growth, Recreation & Leisure, dan Relationship*.

Uji coba penelitian ini menggunakan data set berupa teks kicauan (*tweet*) yang telah dikelompokkan menjadi enam kelas, yang kemudian diimplementasikan pada sistem untuk melihat akurasi yang dihasilkan. Dengan penentuan nilai *learning rate*, jumlah *neuron* pada *hidden layer*, dan jumlah *epoch*, akurasi yang didapat pada tahap pengujian mencapai hingga 50%.

Kata Kunci—*tweet, social media, klasifikasi, backpropagation, resolusi tahun baru.*

I. PENDAHULUAN

TWITTER merupakan salah satu layanan mikroblog yang sedang tren di kalangan pengguna internet. Hingga Mei 2015, Twitter telah memiliki lebih dari 500 juta pengguna, 302 juta di antaranya adalah pengguna aktif [1]. Berbasis teks 140 karakter, lebih dikenal dengan kicauan (*tweet*), Twitter memungkinkan pengguna bertukar informasi dengan cepat, serta kebebasan untuk menuangkan segala hal yang diinginkan dalam teks singkat sebagai alat komunikasi [2]. Jumlah kicauan meningkat pada peristiwa tertentu, seperti peristiwa tahun baru. Pengguna secara masif berkicau tentang resolusi yang ingin dicapai menjelang tahun baru. Twitter yang bersifat *real-time* dapat menjadi sumber informasi bagi penyedia layanan barang dan jasa dalam strategi bisnis.

Teks tweet resolusi akan menjadi bahan yang sangat penting untuk melakukan identifikasi terhadap keinginan atau target pencapaian pengguna di tahun mendatang, di mana pengguna berperan sebagai pelanggan dalam bisnis layanan barang dan jasa. Teks *tweet* resolusi perlu diolah dan diklasifikasikan menjadi beberapa kelas yang telah ditentukan, di antaranya karir dan pendidikan, keuangan, kesehatan dan kebugaran, pengembangan diri, rekreasi dan hiburan, serta hubungan sosial. Data set yang tersedia telah memiliki teks *tweet* beserta kelas hasil klasifikasi secara *manual*. Oleh karena itu, proses klasifikasi dilakukan dengan metode *supervised learning* di mana sistem melakukan training terhadap data yang telah tersedia, dan kemudian bisa dilakukan klasifikasi terhadap data baru untuk memunculkan prediksi pada tahap *testing*.

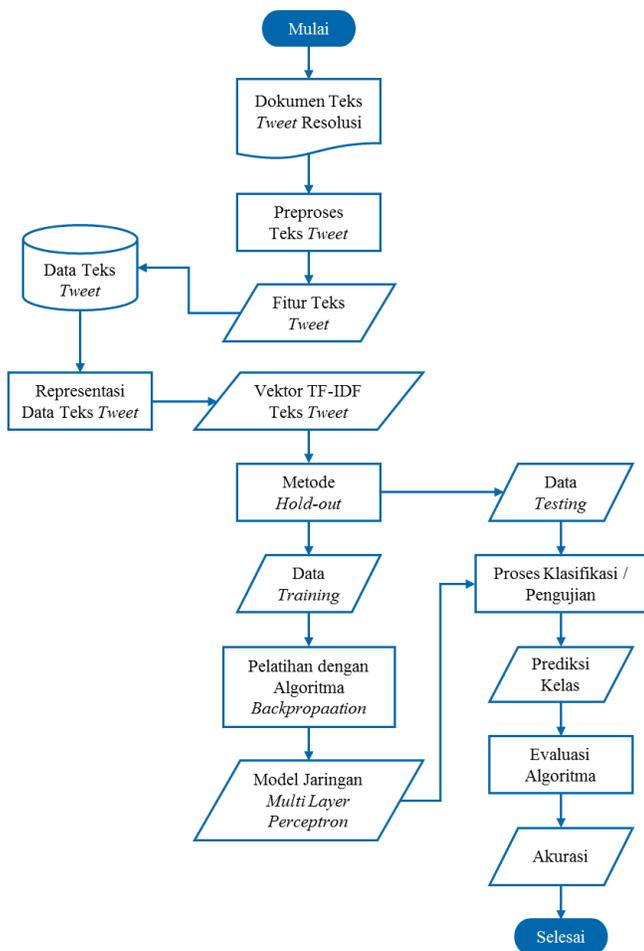
Supervised Learning merupakan suatu pembelajaran yang terawasi di mana jika output yang diharapkan telah diketahui sebelumnya. Beberapa metode populer yang masuk sebagai kategori *supervised learning*, di antaranya adalah *Single-Layer Perceptron* dan *Multi-Layer Perceptron*. Dalam kasus ini akan dipilih *Multi-Layer Perceptron* dengan Algoritma Backpropagation sebagai metode pembelajaran terhadap data pelatihan. *Multi-Layer Perceptron* merupakan metode penerapan *Neural Network (NN)*. *Multi-Layer Perceptron* memiliki kelebihan, yaitu kemampuan untuk mendeteksi atau melakukan analisa untuk permasalahan yang sifatnya sangat kompleks [3], jika dibandingkan dengan *Single-Layer Perceptron*.

Oleh karena itu pada penelitian ini dibangun sebuah sistem yang dapat mengenali teks *tweet* resolusi. Di mana kemudian sistem akan mengklasifikasi teks tersebut ke dalam kelas-kelas yang telah ditentukan, di antaranya karir dan pendidikan, keuangan, kesehatan dan kebugaran, pengembangan diri, rekreasi dan hiburan, serta hubungan sosial.

II. METODE YANG DIUSULKAN

Pada penelitian ini akan dibangun sebuah sistem untuk melakukan klasifikasi tweet resolusi menggunakan Algoritma Backpropagation dengan model Jaringan Saraf Tiruan *Multi-Layer Perceptron*, serta menerapkan metode *supervised learning*. Proses-proses yang terlibat di dalam implementasi sistem ini meliputi tahap praproses teks, tahap pelatihan (*training*), tahap pengujian (*testing*), dan tahap evaluasi kinerja sistem. Tahap praproses teks merupakan tahap pertama yang dilakukan untuk mengolah data masukan sebelum memasuki

tahap utama, yaitu tahap *training*. Praproses teks dilakukan dengan tujuan penyeragaman dan pembersihan elemen-elemen kata yang tidak diperlukan dan tidak bermakna. Pada tahap ini, terdapat tiga subproses, yaitu *case folding* untuk mengubah semua huruf dalam dokumen menjadi huruf kecil (*lower case*), *tokenizing* atau tokenisasi untuk memotong teks berdasarkan setiap kata yang penyusunnya, *filtering* atau penyaringan untuk mengambil kata-kata penting hasil tokenisasi, dan *analyzing* atau analisa untuk menentukan seberapa jauh hubungan antarkata dalam dokumen dan antardokumen. Nilai perhitungan pada tahap praproses teks kemudian akan digunakan sebagai data set untuk tahap *training*, maupun tahap *testing*. Untuk selanjutnya, hasil praproses teks akan disebut sebagai fitur dan banyak data merupakan dokumen. Sebelum masuk pada tahap *training*, data set akan dibagi menjadi bagian, yaitu bagian data *training* dan data *testing*. Diagram alir keseluruhan sistem dapat dilihat pada Gambar 1.



Gambar 1 Diagram Alir Implementasi Sistem

A. Algoritma TF-IDF

Proses penggalian informasi dokumen teks atau lebih khususnya disebut teknologi sistem temu kembali dokumen teks, memiliki 2 tahapan awal yang penting. Selain praproses teks yang telah dijabarkan di atas, proses selanjutnya yang harus dilakukan adalah representasi teks. Tahap representasi teks ini biasa dikenal dengan tahapan pembobotan teks. Telah banyak penelitian-penelitian yang mengusulkan metode-metode baru untuk pembobotan teks, namun metode

pembobotan yang sampai saat ini masih sering digunakan (*popular*) dan hasilnya masih dianggap yang terbaik karena efisien, sederhana, dan akurat adalah Metode *Term Frequency – Inverse Document Frequency*, atau lebih dikenal dengan metode pembobotan Algoritma TF-IDF.

Algoritma TF-IDF mempertimbangkan seringnya kemunculan *term* (kata) dalam dokumen dan rasio panjang dokumen tersebut di dalam *corpus* (sekumpulan dokumen teks) [4]. Bobot dari perhitungan TF-IDF inilah yang menggambarkan seberapa pentingnya *term* (kata) dalam sebuah dokumen dan *corpus*.

Frekuensi kemunculan (*term frequency*) merupakan petunjuk sejauh mana *term* tersebut mewakili isi dokumen atau secara formula diartikan sebagai ukuran seringnya kemunculan sebuah *term* dalam sebuah dokumen dan juga dalam seluruh dokumen di dalam *corpus*. Semakin besar kemunculan suatu *term* dalam dokumen akan memberikan nilai kesesuaian yang semakin besar. *Term frequency* ini dihitung menggunakan Persamaan (1) dengan $tf_i(d_j)$ adalah notasi frekuensi kemunculan *term* ke-*i* dalam dokumen ke-*j*.

Faktor kebalikan yang diperhatikan juga dalam pemberian bobot TF-IDF adalah kejarangmunculan *term* (*Inverse Document Frequency*) dalam koleksi atau secara formula diartikan sebagai logaritma dari rasio jumlah seluruh dokumen dalam *corpus* dengan jumlah dokumen yang memiliki *term* yang dimaksud seperti yang dituliskan secara matematis pada Persamaan (2), di mana idf_i adalah frekuensi kemunculan *term* ke-*i* dalam seluruh dokumen atau satu *corpus*. *Term* yang muncul pada sedikit dokumen harus dipandang sebagai *term* yang lebih penting (*uncommon terms*) daripada *term* yang muncul pada banyak dokumen [5].

Bobot TF-IDF sendiri menggabungkan kedua faktor penting di atas, dan secara formula nilai TF-IDF didapatkan dengan mengalikan nilai TF dengan nilai IDF, ditunjukkan pada Persamaan (3) di mana $(tf-idf)_{ij}$ adalah nilai bobot *term* ke-*i* dalam dokumen ke-*j*.

$$tf_i(d_j) = \frac{freq_i(d_j)}{\sum_{i=1}^k freq_i(d_j)} \quad (1)$$

$$idf_i = \log \frac{|D|}{|\{d: t_i \in d\}|} \quad (2)$$

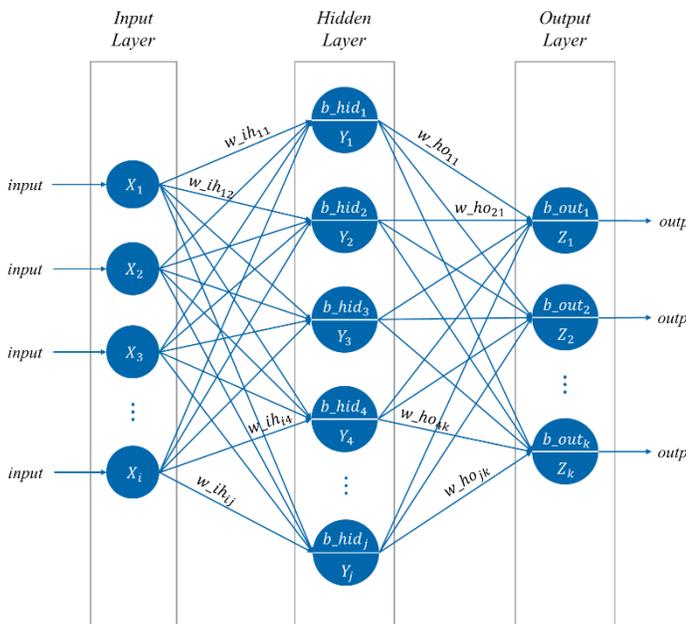
$$(tf-idf)_{ij} = tf_i(d_j) \cdot idf_i \quad (3)$$

Di mana, $freq_i(d_j)$ adalah frekuensi *term* ke-*i* dalam dokumen ke-*j*. $\sum_{i=1}^k freq_i(d_j)$ adalah jumlah *term* pada dokumen ke-*j*. $|D|$ adalah jumlah dokumen dalam *corpus*. $|\{d: t_i \in d\}|$ adalah dokumen yang mengandung *term* ke-*i*.

B. Algoritma Backpropagation

Backpropagation merupakan salah satu metode pembelajaran yang diterapkan dalam Jaringan Saraf Tiruan. *Backpropagation* melakukan dua tahap komputasi, yaitu tahap umpan maju (*feedforward*) untuk menghitung *error* antara nilai *output* secara aktual dan nilai target, serta tahap propagasi balik (*backpropagation*) untuk menghitung mundur nilai *error* tersebut yang digunakan untuk memperbarui nilai bobot dan bias setiap *neuron* pada jaringan [6].

Algoritma *Backpropagation* adalah salah satu algoritma yang menggunakan metode pembelajaran terawasi (*supervised learning*), serta termasuk jaringan *Multi-Layer Perceptron*. Metode *supervised learning* adalah teknik pembelajaran mesin dengan membuat suatu fungsi dari data *training*. Data *training* terdiri dari pasangan nilai *input* dan *output* yang diharapkan dari *input* yang bersangkutan. Tugas dari *supervised learning* adalah untuk memprediksi nilai fungsi untuk nilai semua *input* yang ada [7]. Sedangkan, *Multi-Layer Perceptron* merupakan salah satu model Jaringan Saraf Tiruan yang memiliki satu atau lebih *hidden layer* yang terletak antara *input layer* dan *output layer*, seperti yang ditunjukkan oleh Gambar 2.



Gambar 2 Arsitektur Jaringan Multi-Layer Perceptron

Algoritma pembelajaran *backpropagation* dibagi menjadi dua fase, yaitu fase propagasi dan fase modifikasi bobot. Fase propagasi terdiri dari propagasi maju dari masukan data *training* untuk menghasilkan keluaran pada *output layer*, dan propagasi mundur di mana nilai keluaran pada *output layer* digunakan untuk menghitung *delta* (selisih nilai target dan nilai keluaran) setiap *neuron* pada *hidden layer* dan *output layer*. Sedangkan, yang termasuk pada fase modifikasi bobot adalah mengalikan hasil *delta* dan nilai masukan untuk mendapat nilai *gradient descent*, serta memperbarui nilai bobot dengan mengurangi nilainya sebanyak persentase dari nilai *gradient descent* [6]. Secara detail, langkah-langkah pengerjaan dan perhitungan Algoritma *Backpropagation* ditunjukkan pada persamaan-persamaan seperti di bawah ini.

- Langkah 0
Inisialisasi bobot secara acak. Inisialisasi bobot bertujuan untuk mengisi nilai *neuron* yang masih kosong pada model jaringan. Untuk menentukan banyak bobot yang perlu diinisialisasi dapat digunakan Persamaan (4).

$$total_{bobot} = ij + jk + j + k \quad (4)$$

Kemudian, inisialisasi nilai *epoch* = 0 dan menentukan nilai maksimum *epoch* yang diinginkan.

- Langkah 1

Kerjakan Langkah 2 selama (*epoch* < *epoch*_{maksimum}). Setiap Langkah 1 selesai dilakukan, perbarui nilai *epoch* menjadi *epoch* = *epoch* + 1.

- Langkah 2
Untuk setiap vektor dokumen yang masuk sebagai data *training* dan disimpan pada variabel *X*, lakukan Langkah 3 hingga Langkah 9. Jadi, pada langkah ini akan mulai dilakukan *training* untuk masing-masing vektor dokumen.
- Langkah 3
Setiap *neuron input* ($X_i, i = 1, 2, \dots, p$) menerima sinyal X_i dari vektor dokumen, kemudian meneruskan sinyal tersebut ke semua *neuron* pada *hidden layer*.
- Langkah 4
Setiap *neuron* pada *hidden layer* ($Y_j, j = 1, 2, \dots, q$) menjumlahkan sinyal-sinyal masukan dari *input layer* yang telah dibobotkan dengan Persamaan (5). Kemudian hitung nilai aktivasi dengan Persamaan (6).

$$Y_{in_j} = b_{hid_j} + \sum_{i=1}^p X_i * w_{ih_{ij}} \quad (5)$$

$$Y_j = \frac{1}{1 + e^{Y_{in_j}}} \quad (6)$$

- Langkah 5
Setiap *neuron* pada *output layer* ($Z_k, k = 1, 2, \dots, r$) menjumlahkan sinyal-sinyal masukan yang telah dibobotkan dengan Persamaan (7). Kemudian hitung sinyal nilai aktivasi seperti Persamaan (8).

$$Z_{in_k} = b_{out_k} + \sum_{i=1}^p X_i w_{ho_{jk}} \quad (7)$$

$$Z_k = \frac{1}{1 + e^{Z_{in_k}}} \quad (8)$$

- Langkah 6
Setiap *neuron* pada *output layer* ($Z_k, k = 1, 2, \dots, r$) menerima target pola yang berhubungan dengan pola *neuron* pada *input layer*. Kemudian, hitung informasi *error* dengan Persamaan (9). Selanjutnya, hitung selisih bobot dengan Persamaan (10), serta selisih bias dengan Persamaan (11).

$$\delta_{out_k} = (t_k - Z_k) f'(Z_{in_k}) \quad (9)$$

$$\Delta w_{ho_{jk}} = \alpha \delta_{out_k} Y_j \quad (10)$$

$$\Delta b_{out_k} = \alpha \delta_{out_k} \quad (11)$$

- Langkah 7
Setiap *neuron* pada *hidden layer* ($Y_j, j = 1, 2, \dots, q$) menjumlahkan perkalian nilai *gradient descent* dengan bobot seperti pada Persamaan (12). Kemudian hitung selisih bobot dan bias dengan Persamaan (13) dan Persamaan (14).

$$\delta_{hid_j} \quad (12)$$

$$= \left(\sum_{k=1}^r \delta_{out_k} w_{ho_{jk}} \right) f'(Y_{in_j})$$

$$\Delta w_{ih_{ij}} = \alpha \delta_{hid_j} Y_j \quad (13)$$

$$\Delta b_{hid_j} = \alpha \delta_{hid_j} \quad (14)$$

- Langkah 8
Setiap *neuron* pada *output layer* ($Z_k, k = 1, 2, \dots, r$) perbarui bias dan bobotnya ($j = 1, 2, \dots, q$) dengan

Persamaan (15) dan Persamaan (16). Untuk setiap *neuron* pada *hidden layer* ($Y_j, j = 1, 2, \dots, p$) perbarui bias dan bobotnya ($i = 1, 2, \dots, p$) dengan Persamaan (17) dan Persamaan (18).

$$w_{ho_{jk}}(baru) = w_{ho_{jk}}(lama) + \Delta w_{ho_{jk}} \quad (15)$$

$$b_{out_k}(baru) = b_{out_k}(lama) + \Delta b_{out_k} \quad (16)$$

$$w_{ih_{ij}}(baru) = w_{ih_{ij}}(lama) + \Delta w_{ih_{ij}} \quad (17)$$

$$b_{hid_j}(baru) = b_{hid_j}(lama) + \Delta b_{hid_j} \quad (18)$$

- Langkah 9

Hitung *MSE* dengan Persamaan (19).

$$MSE = \frac{\sum_{k=1}^r (t_k - Z_k)^2}{i} \quad (19)$$

C. Evaluasi Kinerja Sistem

Evaluasi kinerja sistem ini adalah dengan menggunakan akurasi. Akurasi digunakan untuk mengukur tingkat kualitas keberhasilan klasifikasi *tweet* pada tahap pengujian. Kategori uji coba dinyatakan benar jika hasil prediksi tahap pengujian sama dengan target dari data *ground truth*. Sedangkan kategori uji coba dinyatakan salah jika hasil prediksi tahap pengujian tidak sama dengan target dari data *ground truth*. Persamaan untuk pengukuran akurasi dapat dilihat pada Persamaan (20). Perhitungan akurasi akan dilakukan pada kedua tahap yakni, tahap pelatihan dan tahap pengujian untuk mengetahui bahwa pada tahap pengujian sistem telah belajar dan mengenali pola data dengan baik.

$$akurasi = \frac{uji\ coba\ benar}{uji\ coba\ benar + uji\ coba\ salah} \quad (20)$$

III. UJI COBA DAN PEMBAHASAN

Tabel 1 Informasi Komposisi Berkas 1 dan 2

No	Nama Berkas	Total Dok	Jumlah Dokumen untuk Kelas ke					
			1	2	3	4	5	6
1	Berkas 1	1056	176	176	176	176	176	176
2	Berkas 2	3914	215	176	840	1781	467	435

Tabel 2 Informasi Jumlah Fitur Berkas 1 dan 2

No	Nama Berkas	Total Dokumen	Jumlah Fitur Kata	
			Sebelum	Sesudah
1	Berkas 1	1056	3149	1990
2	Berkas 2	3914	7770	4551

Data uji coba yang digunakan adalah kumpulan data teks *tweet* yang disimpan pada berkas berekstensi *.xls*. Pada pengujian ini, akan digunakan dua jenis berkas. Berkas pertama memiliki 1056 dokumen teks *tweet*, sedangkan berkas kedua adalah memiliki 3914 dokumen teks *tweet*. Masing-masing berkas terdiri dari enam kelas kategori. Di antaranya adalah *Career & Education, Finance, Health & Fitness, Personal Growth, Recreation & Leisure, dan Relationship*. Perbedaan kedua berkas tidak hanya terletak pada jumlah dokumen, tetapi juga jumlah porsi dokumen pada masing-masing kelas. Berkas pertama memiliki ketidakseimbangan pada jumlah dokumen

untuk masing-masing kelas, sedangkan dokumen memiliki jumlah dokumen yang sama untuk masing-masing kelas. Untuk lebih jelas, dapat dilihat pada Tabel 1.

Tabel 1 menunjukkan jumlah dokumen yang dimiliki masing-masing berkas. Namun, hal ini belum dapat menentukan jumlah fitur hasil tahap praproses teks. Setelah dilakukan tahap praproses teks, diketahui bahwa berkas ke-1 dengan jumlah dokumen sebanyak 1056 menghasilkan fitur sebanyak 1990 kata, sedangkan berkas ke-2 menghasilkan sebanyak 4551 fitur kata. Sedangkan, Tabel 2 menunjukkan perbedaan jumlah fitur kata sebelum dan sesudah melalui tahap praproses teks. Di mana menunjukkan bahwa tahap ini berpengaruh besar dalam penentuan fitur untuk data set awal sebelum memasuki tahap utama sistem, yakni tahap *training*. Selain itu, juga menunjukkan bahwa jika data dilipatgandakan sebanyak empat kali, belum tentu fitur yang dihasilkan pun empat kali dari jumlah yang lebih kecil. Hal ini bergantung pada jenis kata yang digunakan, terutama penggunaan yang sering dalam suatu dokumen teks.

Tabel 3 Hasil Pengujian dengan Variabel Paling Ideal

No	Data	Fitur	MSE	Akurasi (%)		Lama Train (s)
				Train	Test	
1	1056	1990	0,01	0,98	0,52	52
2	3914	4551	0,06	0,94	0,58	3280

Uji coba dilakukan dengan variabel yang paling ideal untuk mengetahui hasil klasifikasi data, baik pada tahap *training* maupun tahap *testing*, yakni 0,10 untuk *learning rate*, 1% jumlah *neuron*, 10 *epoch*, dan 20% untuk data *testing*.

Tabel 4 Tabel Kebenaran Training Berkas 1

Kelas	Prediksi						Total
	1	2	3	4	5	6	
Target	1	<u>138</u>	0	0	0	0	138
	2	1	<u>138</u>	0	0	0	139
	3	0	0	<u>139</u>	0	0	139
	4	0	1	0	<u>131</u>	0	132
	5	18	0	0	0	<u>136</u>	154
	6	0	0	0	1	0	<u>142</u>
Total	157	139	139	132	136	142	845

Tabel 5 Tabel Kebenaran Testing Berkas 1

Kelas	Prediksi						Total
	1	2	3	4	5	6	
Target	1	<u>19</u>	0	3	5	4	32
	2	1	<u>25</u>	1	1	0	29
	3	3	0	<u>18</u>	2	4	29
	4	2	5	4	<u>11</u>	3	31
	5	11	3	3	11	<u>19</u>	49
	6	4	0	4	11	4	<u>16</u>
Total	40	33	33	41	34	28	209

Dari hasil uji coba, dapat dilihat pada Tabel 3 akurasi masih terhitung cukup rendah, hanya berkisar sekitar 50%. Untuk akurasi tahap *training* dihasilkan nilai yang tinggi karena prediksi dilakukan menggunakan model jaringan yang dihasilkan oleh data itu sendiri. Sedangkan, nilai akurasi tahap *testing* terbilang cukup rendah karena tidak semua fitur yang memiliki nilai pada data *testing* mendapat bobot jaringan yang sesuai saat tahap *training*. Di mana pada data *training* fitur yang dimaksud tidak memiliki nilai pembobotan awal yang tinggi selayaknya seperti pada data *testing*.

Tabel 4 menunjukkan hasil klasifikasi data pada berkas ke-1 dengan 1056 data. Beris pertama Tabel 4 menjelaskan bahwa 132 data dengan target kelas ke-1 diprediksikan dengan tepat ke kelas ke-1 setelah tahap *training*. Masih pada baris ke-2, terdapat satu data yang terjadi misklasifikasi. Data tersebut memiliki target kelas ke-2, yakni *Finance*, namun setelah melalui tahap *training* data diprediksikan sebagai data kelas ke-1, yakni kelas *Career & Education*. Misklasifikasi terbesar terjadi pada kelas ke-5 *Recreation & Leisure* yang diprediksikan sebagai *Career & Growth*, sebanyak 18 data yang salah. Dan, untuk hasil klasifikasi data tahap *testing* dapat dilihat pada Tabel 5.

Tabel 6 Tabel Kebenaran *Training* Berkas 2

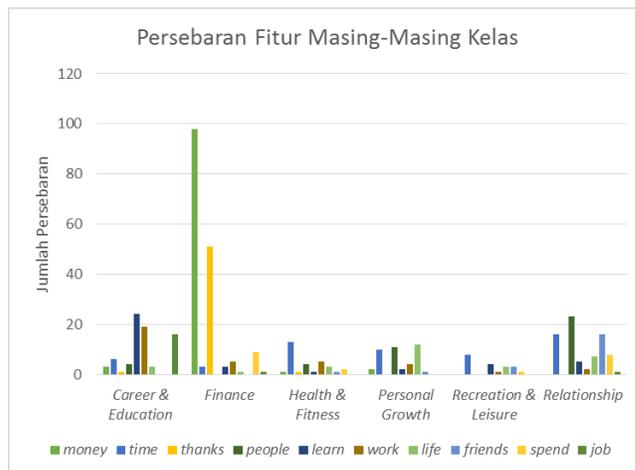
Kelas	Prediksi						Total	
	1	2	3	4	5	6		
Target	1	<u>149</u>	0	1	0	0	2	152
	2	0	<u>131</u>	1	3	0	0	135
	3	1	0	<u>636</u>	10	5	0	652
	4	106	2	7	<u>1375</u>	17	20	1527
	5	1	0	0	7	<u>327</u>	3	338
	6	0	0	0	5	0	<u>323</u>	328
Total	257	133	645	1400	349	348	3132	

Tabel 7 Tabel Kebenaran *Testing* Berkas 2

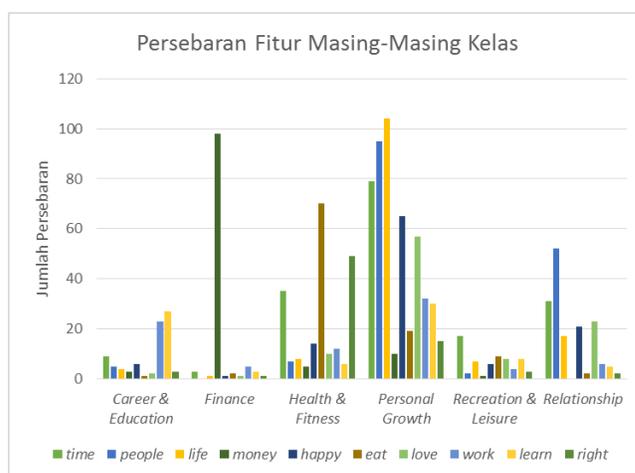
Kelas	Prediksi						Total	
	1	2	3	4	5	6		
Target	1	<u>17</u>	1	7	9	5	0	39
	2	2	<u>27</u>	2	5	0	0	36
	3	6	2	<u>104</u>	21	12	6	151
	4	33	7	52	<u>254</u>	44	43	433
	5	8	1	2	17	<u>29</u>	6	63
	6	2	1	6	15	12	<u>24</u>	60
Total	68	39	173	321	102	79	782	

Selanjutnya akan ditampilkan hasil klasifikasi data untuk berkas ke-2 dengan 3914 data dalam bentuk tabel kebenaran. Untuk mengetahui detail misklasifikasi uji coba berkas ke-2, dapat dilihat pada Tabel 6 untuk tahap *training* dan Tabel 7 untuk tahap *testing*. Jika diperhatikan pada tabel kebenaran pada Tabel 6, data dengan kelas ke-5, yakni *Recreation & Leisure*, merupakan kelas dengan jumlah misklasifikasi data paling tinggi. Di mana hal ini terjadi sama persis dengan uji coba pada 1056 data. Kesalahan klasifikasi pada tahap *training* berpengaruh besar terhadap tahap *testing* dalam memprediksi, karena model jaringan itu sendiri tidak pintar dalam mengenali pola kelas *Personal Growth*, sesuai yang ditunjukkan pada Tabel 7 yang bertanda merah.

Jika diperhatikan pada Tabel 7, kelas ke-4 merupakan kelas dengan hasil prediksi terbanyak. Hal ini mungkin terjadi karena jumlah data pada kelas ke-4 merupakan yang paling banyak dan paling dominan dari kelas-kelas yang lain. Sehingga, ketika sistem melakukan prediksi terdapat kecondongan terhadap kelas ke-4. Hal ini menunjukkan bahwa adanya dominasi data pada kelas tertentu sangat berpengaruh terhadap proses prediksi, yang dapat menyebabkan prediksi lebih condong ke kelas dominan. Selain itu, ketika kelas ke-4 diprediksikan, sistem juga kebingungan dengan model yang disebabkan banyaknya data kelas ke-4.



Gambar 3 Grafik Persebaran Fitur - Kelas Berkas 1



Gambar 4 Grafik Persebaran Fitur - Kelas Berkas 2

Ditunjukkan pada Gambar 3 merupakan grafik persebaran lima fitur terbanyak yang dari Berkas 1. Dapat dilihat pada grafik, fitur *money* merupakan yang paling dominan pada kelas *Finance*, di mana sesuai dengan pemetaan secara manual. Ketika kata ‘uang’ disebut maka yang terlintas pasti mengenai bidang keuangan. Sama halnya dengan *money*, fitur *eat* merupakan fitur paling mendominasi pada kelas *Health & Fitness*. Untuk fitur *work*, memiliki persebaran yang cukup tinggi pada kelas *Career & Education*, namun juga pada kelas *Personal Growth*. Fitur *learn* yang memiliki makna mempelajari dimiliki sangat tinggi pada kelas *Personal Growth*, di mana sesuai. Namun fitur *learn* juga sesuai jika masuk kelas *Career & Education*. Hal ini yang menjadi penyebab adanya misklasifikasi pada sistem.

Kelemahan menggunakan data *tweet* adalah karena memiliki karakter yang sangat terbatas. Sehingga dengan kata seminimal mungkin, informasi harus tersampaikan. Dan juga, dalam media sosial pengguna tentunya tidak menggunakan kata baku. Sehingga sulit untuk diinterpretasikan menjadi fitur yang benar-benar merepresentasikan masing-masing kelas.

IV. KESIMPULAN

Kesamarataan jumlah dokumen pada masing-masing kelas juga merupakan faktor penting yang dapat mempengaruhi hasil klasifikasi. Dengan jumlah dokumen yang sama pada masing-masing kelas, akan membantu sistem mempelajari dan melakukan prediksi dengan lebih adil. Kelas dengan jumlah dokumen yang lebih banyak dibanding kelas yang lebih sedikit dapat menyebabkan kecondongan sistem pada kelas tertentu. Sehingga, akan mempengaruhi nilai akhir akurasi.

Algoritma *Backpropagation* yang diaplikasikan pada penelitian ini menghasilkan akurasi *testing* tertinggi sebesar 58% dengan kombinasi variabel adalah 0,10 untuk nilai *learning rate*, jumlah *neuron* pada *hidden layer* adalah 1% dari jumlah *neuron* pada *input layer*, *epoch* maksimum adalah 10, dan persentase jumlah data *testing* adalah 20%.

Untuk dilanjutkan ke penelitian selanjutnya adalah melakukan pengembangan metode ekstraksi fitur teks untuk mengurangi jumlah fitur yang terlalu banyak. Kemudian, pengembangan metode pada pembobotan. Pada kasus *tweet*, sebuah kata cenderung hanya muncul beberapa kali, tidak sebanyak dalam paragraf. Perlu adanya pembobotan yang tepat untuk masing-masing fitur. Selain itu juga perlu pengembangan Algoritma *Backpropagation* untuk menangani kasus apabila jumlah dokumen pada masing-masing kelas tidak sama atau tidak rata, yang menyebabkan dominasi atau kecondongan pada kelas tertentu saat proses *training*.

DAFTAR PUSTAKA

- [1] Wikipedia, "Twitter," [Online]. Available: id.wikipedia.org/wiki/Twitter. [Accessed 11 March 2016].
- [2] J. Tenuto, "From Getting in Shape to Finally Meeting One Direction: A Roundup of 2015 New Year's Resolutions," CrowdFlower, [Online]. Available: crowdfunder.com/blog/new-years-resolutions-2015. [Accessed 16 March 2016].
- [3] I. Prasetyo, "Jenis-Jenis Neural Network," Tukar Pengetahuan, [Online]. Available: tukarpengetahuan.com/2015/06/jenis-jenis-neural-network. [Accessed 16 March 2016].
- [4] M. N. Saadah, R. W. Atmagi, D. S. Rahayu and A. Z. Arifin, "Sistem Temu Kembali Dokumen Teks dengan Pembobotan Tf-Idf Dan LCS," *Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember*, 2012.
- [5] O. Karmayasa and I. B. Mahendra, "Implementasi Vector Space Model dan Beberapa Notasi Metode Term Requency Inverse Document Frequency (Tf-Idf) pada Sistem Temu Kembali Infomasi," *Jurnal Elektronik Ilmu Komputer Universitas Udayana*, vol. 1, no. 1, 2012.
- [6] Wikipedia, "Backpropagation," 2013. [Online]. Available: en.wikipedia.org/wiki/Backpropagation. [Accessed 20 Maret 2016].
- [7] N. Z. D. L. Mulyawati, "Machine Learning," 2012. [Online]. Available: nita_zelfiafst09.web.unair.ac.id/artikel_detail-44883-Umum-Machine%20Learning.html. [Accessed 1 April 2016].