

Sentimen Analisis Tweet Berbahasa Indonesia dengan *Deep Belief Network*

Ira Zulfa*¹, Edi Winarko²

¹Program Studi S2 Ilmu Komputer, FMIPA UGM, Yogyakarta, Indonesia

²Departemen Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta, Indonesia

e-mail: *¹ira.zulfaa@gmail.com, ²ewinarko@ugm.ac.id

Abstrak

Sentimen analisis adalah riset komputasional dari opini sentiment dan emosi yang diekspresikan secara tekstual. Twitter menjadi perangkat komunikasi paling populer di kalangan pengguna internet. Deep Learning adalah area baru dalam penelitian machine learning, yang telah diperkenalkan dengan tujuan menggerakkan machine learning lebih dekat dengan salah satu tujuan aslinya yaitu artificial intelligence. Tujuan deep learning mengganti manual engineering dengan learning, pada perkembangannya deep learning memiliki himpunan algoritma yang fokus pada pembelajaran representasi data (nonlinear) bertingkat. Deep Belief Network (DBN) merupakan salah satu metode machine learning yang termasuk dalam metode Deep Learning yang merupakan tumpukan atau stack dari beberapa algoritma atau metode dengan feature extraction yang memanfaatkan seluruh resource seoptimal mungkin.

Penelitian ini bertujuan untuk melakukan pengklasifikasian terhadap sentimen positif, negatif, dan netral terhadap data uji dan untuk mengetahui akurasi model klasifikasi dengan menggunakan metode Deep Belief Network ketika diaplikasikan pada klasifikasi tweet untuk menandai kelas sentimen data training tweet berbahasa Indonesia. Dari percobaan yang dilakukan, hasil pengujian pada sistem yang dibangun memperlihatkan bahwa metode terbaik pada data tweet adalah metode DBN yaitu dengan akurasi sebesar 93,31%, ketika dibandingkan dengan metode Naive Bayes yang memiliki akurasi sebesar 79,10%, dan SVM (Support Vector Machine) yang memiliki akurasi sebesar 92,18%.

Kata kunci— *Sentimen Analisis, Twitter, deep belief*

Abstract

Sentiment analysis is a computational research of opinion sentiment and emotion which is expressed in textual mode. Twitter becomes the most popular communication device among internet users. Deep Learning is a new area of machine learning research. It aims to move machine learning closer to its main goal, artificial intelligence. The purpose of deep learning is to change the manual of engineering with learning. At its growth, deep learning has algorithms arrangement that focus on non-linear data representation. One of the machine learning methods is Deep Belief Network (DBN). Deep Belief Network (DBN), which is included in Deep Learning method, is a stack of several algorithms with some extraction features that optimally utilize all resources. This study has two points. First, it aims to classify positive, negative, and neutral sentiments towards the test data. Second, it determines the classification model accuracy by using Deep Belief Network method so it would be able to be applied into the tweet classification, to highlight the sentiment class of training data tweet in Bahasa Indonesia. Based on the experimental result, it can be concluded that the best method in managing tweet data is the DBN method with an accuracy of 93.31%, compared with Naive Bayes method which has an accuracy of 79.10%, and SVM (Support Vector Machine) method with an accuracy of 92.18%.

Keywords— *Sentimen Analisis, Twitter, deep belief*

1. PENDAHULUAN

Keingintahuan pihak korporasi akan sentimen terhadap produk ataupun layanannya biasanya dipacu oleh tingkat persaingan yang semakin tinggi diantara para pelaku pasar. Namun, untuk mengetahui sentimen publik terkadang membutuhkan biaya dan usaha yang tidak mudah. Namun melalui web, perusahaan maupun perorangan dapat mencari jawaban atas keingintahuan tersebut. Opini-opini orang akan berbagai hal dituangkan di web. Yang perlu dilakukan adalah usaha untuk mengumpulkan semua opini tersebut dan mengolahnya menjadi suatu yang dapat menjawab pertanyaan-pertanyaan tadi.

Opini mining atau sentimen analisis adalah riset komputasional dari opini, sentimen dan emosi yang diekspresikan secara tekstual [1]. Sekarang ketika suatu organisasi / perusahaan / perorangan ingin memperoleh opini publik mengenai produk, citra dan layanannya, maka mereka tidak perlu melakukan survei konvensional dan fokus group yang mahal biayanya [2].

Deep Learning adalah area baru dalam penelitian *Machine Learning*, yang telah diperkenalkan dengan tujuan menggerakkan *Machine Learning* lebih dekat dengan salah satu tujuan aslinya yaitu *Artificial Intelligence*. *Deep Learning* adalah tentang belajar beberapa tingkat representasi dan abstraksi yang membantu untuk memahami data seperti gambar, suara, dan teks [3].

Deep Belief Network (DBN) adalah suatu pengembangan dari *Deep Learning* yang merupakan tumpukan atau stack dari beberapa algoritma atau metode yang bertujuan feature extraction yang memanfaatkan seluruh resource seoptimal mungkin. *Deep Learning* mencakup algoritma unsupervised dan supervised learning sehingga dapat memanfaatkan data yang berlabel maupun tidak berlabel. Pendekatan yang sering digunakan untuk mengimplementasikan *Deep Learning* adalah graphical methods atau multi layer representation atau multi layer graphical model seperti *Deep Belief Network* [4].

Metode klasifikasi sentimen yang dipilih adalah DBN yaitu salah satu metode *machine learning* untuk pemodelan semantik kalimat yang termasuk dalam metode *Deep learning*. Dalam pemrosesan text data sentimen analisis untuk membedakan dengan penelitian sebelumnya. Hasil dari penelitian ini untuk membandingkan hasil akurasi yang dilakukan oleh Aliandu (2012) dengan metode *Naive Bayesian* dan metode *Support Vektor Machine*. Nantinya dapat dilihat hasil mana yang lebih akurat dari metode-metode yang digunakan.

Sumber data *Twitter* berbahasa Indonesia dipilih karena penelitian dalam bahasa Indonesia belum banyak dilakukan, sedangkan DBN dipilih karena metode ini belum pernah digunakan dalam kasus sentimen analisis sebelumnya dan diharapkan dapat mencapai kinerja yang lebih baik dari metode – metode sebelumnya.

Metode DBN diharapkan bisa digunakan dalam pengimplementasian Tweet untuk keperluan data training dikumpulkan dari beragam data yang dianotasikan kelas sentimennya secara otomatis dengan akun media nasional di *Twitter*. Akun media nasional merupakan teks yang hanya berisi pernyataan atas suatu fakta atau kejadian dan tidak mengekspresikan emosi tertentu. Apakah nantinya karakteristik DBN ini sesuai jika diaplikasikan pada data *Twitter*.

Bedasarkan hal tersebut akan dibuat analisa sentimen tweet berbahasa Indonesia dengan menggunakan metode *Deep Belief Network* dalam membangun model klasifikasi.

2. METODE PENELITIAN

2.1 Inputan Data

2.1.1 Data tweet Berbahasa Indonesia

Data tweet diperoleh dengan melakukan *scraping* pada situs www.search.twitter.com dengan memanfaatkan API twitter. tweet yang diperoleh berisi daftar tweet berbahasa Indonesia.

2.1.2 Data Kata Baku Bahasa Indonesia

Data kata baku diperoleh berdasarkan kamus besar bahasa Indonesia. Total kata baku atau kata dasar bahasa Indonesia yang digunakan dalam penelitian ini berjumlah 828 kata. Data kata baku diperoleh dari inputan secara manual

2.1.3 Data Kata Tidak Baku

Data kata tidak baku diperoleh dari data kotor hasil *scraping* tweet.

2.1.4 Data Stopword

Stopword adalah kata-kata yang tidak memiliki pengaruh dalam proses klasifikasi, seperti: yang, dan, atau, ke, dari, dan lain-lain. Total Kata Umum yang digunakan dalam penelitian ini berjumlah 754 kata. Data Kata Umum diperoleh dari data kotor hasil *scraping* tweet yang diinput secara manual.

2.1.5 Data Simbol

Data Simbol adalah symbol tidak memiliki pengaruh dalam proses klasifikasi, seperti: (., : ; ! - ? " ' () 0-9*).

2.1.6 Data Antonim

Antonim adalah kata-kata yang berlawanan makna dengan kata lain. Data antonim berfungsi sebagai konversi sesudah kata negasi kemudian kata negasi dihilangkan. Misalnya, terdapat kalimat yang mengandung dua kata "*tidak besar*" maka susunan kata setelah dikonversi dengan data antonym menjadi "*kecil*". Total antonim yang digunakan dalam penelitian ini berjumlah 289 kata yang diperoleh dari hasil inputan secara manual pada sistem.

2.1.7 Data POS Tagging

POS Tagging merupakan kata yang telah dilabelisasi kelas kata berdasarkan kamus besar bahasa Indonesia. Contoh : "makanan enak" setelah di *POS Tagging* menjadi "makanan NN// enak JJ//". Total *POS Tagging* yang digunakan dalam penelitian ini berjumlah 48,274 kata. Data *POS Tagging* diperoleh dari link berikut: <http://hikaruyuki.lecture.ub.ac.id/kamus-kata-dasar-dan-stopword-list-bahasa-indonesia/> dan ditambah secara manual berdasarkan pengamatan dalam proses pengembangan sistem ini.

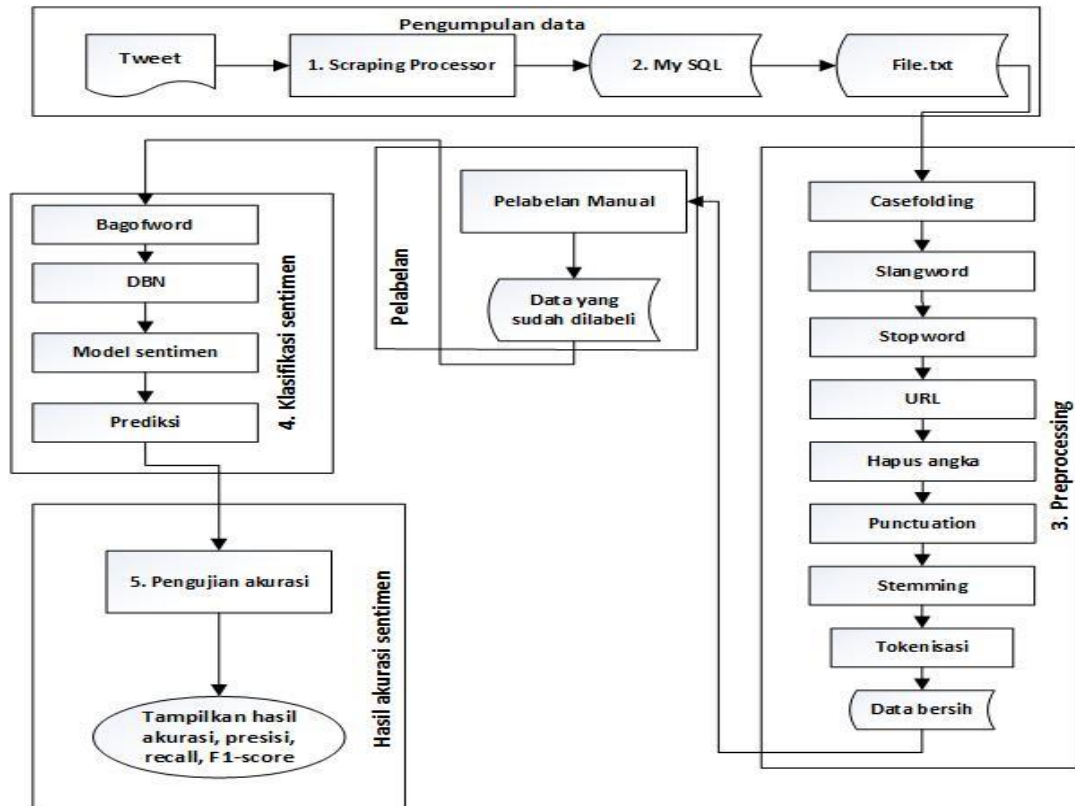
2.2 Arsitektur Sistem

Secara umum arsitektur sistem ini terdiri dari lima bagian diantaranya adalah pengumpulan data, preprocessing, pelabelan, klasifikasi sentimen, hasil akurasi sentimen. Arsitektur sistem yang dirancang seperti terlihat pada Gambar 1.

Tahap pertama dari arsitektur sistem yaitu melakukan pengumpulan data dari server *Twitter* memanfaatkan API *Twitter* dengan cara *scraping*. Selanjutnya data hasil *scraping* tersebut akan mengalami proses *preprocessing* agar data siap digunakan untuk proses klasifikasi. Hal ini dilakukan karena tidak semua data tweet tersebut dapat digunakan. Pada tahap *preprocessing* ini dilakukan pembersihan data tweet yang terdiri dari *case folding*, penghapusan simbol-simbol, tokenisasi, konversi *slangword*, penghapusan *stopword*. Kemudian data tweet hasil *preprocessing* disimpan dalam database dengan tempat yang berbeda dari data tweet kotor.

Kemudian data tweet bersih yang sudah diberi label secara manual akan dirubah menjadi *vector* dengan *bagofword* untuk dilakukan perhitungan dengan metode *deep belief network*,

naïve bayes dan *support vector machine* agar menghasilkan model klasifikasi sentimen. Kemudian tahap terakhir adalah pengujian klasifikasi tweet yang meliputi pengujian akurasi, presisi, *recall*, dan *f1-score* yaitu menghitung keakuratan tweet pada sistem.



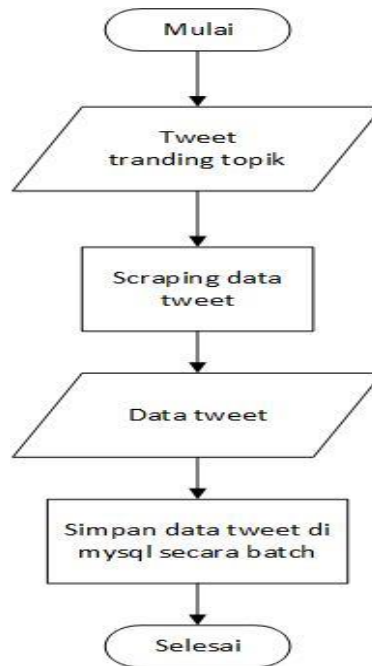
Gambar 1 Arsitektur Sistem

2.3 Pengumpulan Data

Pada tahap ini dilakukan pengumpulan data-data yang dibutuhkan dalam penelitian. Data-data yang dimaksud adalah kumpulan dari tweet terkait tentang *tranding topik* di www.search.twitter.com.

2.3.1 Scraping

Pengumpulan data pada penelitian ini dilakukan dengan cara *scraping* pada situs www.search.twitter.com. Proses *scraping* dilakukan dengan memanfaatkan *library* Python Tweepy <http://tweepy.readthedocs.io/en/v3.5.0/api.html#tweepy-api-twitter-api-wrapper> dan Twitter API. *Scraping* tweet dilakukan secara otomatis untuk mengambil data *tweet* dengan jangka waktu data yang bisa ditentukan. Sebelum dapat melakukan *scraping*, sistem ini akan melakukan pengecekan *key consumer* dan *access token* agar sistem dapat mengambil data dari twitter apabila data token ada terdaftar dalam sistem *twitter* maka sistem dapat mengambil data *twitter* pada wilayah Indonesia. Data *tweet* bahasa Indonesia dengan query yang dicari. Sistem ini menyimpan data tweet kedalam *database*, jika data *tweet* yang diambil telah tersimpan didalam *database* maka sistem tidak menyimpan data tersebut sehingga tidak terdapat tweet yang sama pada data *tweet*. Data *tweet* terkumpul secara otomatis tersimpan dalam *database* mysql. Dari mysql data akan diekspor ke *file .txt*. Langkah-langkah proses *scraping* tweet seperti terlihat pada Gambar 2.



Gambar 2 Proses scraping tweet

2.4 Preprocessing

Preprocessing sangat menentukan dalam proses penentuan sentimen dengan *deep belief network* dan klasifikasi tweet menjadi lebih akurat. *Preprocessing* juga digunakan untuk mendapatkan data bersih. Tahap *preprocessing* terdiri dari beberapa proses yang akan dibahas satu per satu secara detail, antara lain:

2.4.1 Case Folding

Tahap ini berfungsi untuk merubah karakter huruf di dalam komentar menjadi karakter huruf kecil semua.

2.4.2 Penghapusan Simbol-Symbol

Penghapusan simbol-simbol berfungsi untuk menghapus karakter khusus dalam komentar seperti tanda baca (seperti: koma (,), titik(.), tanda tanya (?), tanda seru (!) dan sebagainya), angka numerik (0 - 9), dan karakter lainnya (seperti: \$, %, *, dan sebagainya).

2.4.3 Tokenisasi

Tokenisasi berfungsi untuk memecah komentar menjadi satuan kata. Proses tokenisasi dilakukan dengan melihat setiap spasi yang ada dalam komentar maka berdasarkan spasi tersebut kata-kata dapat dipecah.

2.4.4 Konversi Slangword

Konversi *slangword* merupakan proses mengubah terhadap kata tidak baku ke kata baku. Tahap ini dilakukan dengan menggunakan bantuan kamus *slangword* dan padanannya dalam kata-kata baku. Tahapan ini akan memeriksa kata yang terdapat dalam kamus *slangword* atau

tidak. Jika kata tidak baku terdapat dalam kamus *slangword* maka kata tidak baku akan dirubah ke kata baku yang terdapat didalam kamus *slangword*.

2.4.5 Penghapusan Stopword

Tahap ini berfungsi untuk menghilangkan kata-kata yang tidak penting dalam proses klasifikasi dan penentuan alasan, seperti kata: yang, tetapi, atau, ke, di, dengan, dan sebagainya.

2.5 Perancangan Training

Perancangan proses training dilakukan dengan menggunakan metode *deep belief network* dan *support vector machine*, *Naïve bayes* sebagai metode pembandingan.

2.5.1 Perancangan *deep belief network*

Perancangan *deep belief network* dibagi menjadi 2 tahapan yaitu tahap pembelajaran dan tahap klasifikasi. Proses tahapan perhitungan *deep belief network* pada tahap pembelajaran adalah sebagai berikut :

- Menghitung jumlah seluruh tweet (data) dalam data *training*.
- Menghitung jumlah tweet pada masing-masing kelas.
- Lakukan pelatihan pada RBM pertama pada model dengan data input $x = h^0$ atau dapat dikatakan ini merupakan lapisan terlihat v pada RBM pertama untuk mendapatkan parameter w^1 .
- Tetapkan parameter w^1 dan gunakan lapisan pertama untuk memperoleh representasi fitur yang akan digunakan sebagai data untuk lapisan kedua dengan melakukan sampling h^1 dari $Q(h^1|u) = P(h^1|u, w^1)$.
- Tetapkan parameter w^2 yang didefinisikan oleh fitur lapisan kedua dan gunakan sampel h^2 dari $Q(h^2|h^1) = P(h^2|h^1, w^2)$. sebagai data latih pada lapisan ketiga.

Lakukan langkah diatas secara berulang untuk lapisan berikutnya. Berikut adalah tahapan rinci dari CD-1 untuk training RBM [3]:

- Inisialisasi bobot dan bias, laju pembelajaran dan tetapkan maksil epoch.
- Ketika kondisi berhenti tidak terpenuhi atau epoch maksimal tidak terpenuhi lakukan langkah ke-3 sampai ke-8 secara berulang-ulang.
- Gunakan data training x sebagai neuron-neuron terlihat pada lapisan terlihat v

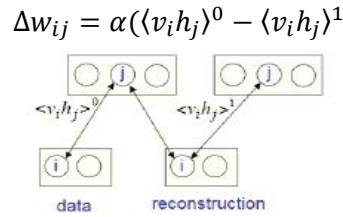
$$p(h_j = 1, v; w) = \varphi\left(\sum_{j=1} w_{ij}v_i + a_j\right) \quad (1)$$

- Perbaharui lapisan tersembunyi dengan $p(h = 1, v ; w)$ melakukan persamaan (1) kemudian aktivasi neuron-neuron biner stokastik pada lapisan tersembunyi dengan mengambil probabilitas pada persamaan (1) lebih dari nilai random yang tersebar antara 0 dan 1.

$$p(v_i = 1, h, ; w) = \varphi\left(\sum_{j=1} w_{ij} h_j + b_i\right) \quad (2)$$

- Rekonstruksi neuron-neuron terlihat v_i^1 dengan menggunakan hasil dari neuron-neuron biner stokastik pada langkah ke-2 sebagai masukan pada persamaan (2).
- Hitung probabilitas output dari rekonstruksi h_j^1 dengan melakukan perhitungan probabilitas pada persamaan (1), namun hasil rekonstruksi dijadikan sebagai masukan.
- Hitung bobot delta dengan mengikuti persamaan (3) atau dapat melihat Gambar 3 dengan rumus sebagai berikut:

(3)



Gambar 3 Langkah CD-1

- h. Perbaharui bobot dengan mengikuti persamaan (4)

$$w_{ij}(t) = w_{ij}(t - 1) + \Delta w_{ij} \quad (4)$$

2.5.2 Perancangan Support Vector Machine dan Naïve Bayes

Metode pembandingan yang digunakan pada penelitian ini adalah metode *Support Vector Machine* (SVM) dan *Naïve Bayes* (NB). Proses perancangan diawali dengan pemanggilan library sklearn dari Python dengan menggunakan beberapa parameter, antara lain : kernel yang digunakan adalah kernel linear untuk SVM, nilai gamma yang digunakan adalah 1.0, dan nilai coef0 yang digunakan adalah 1.0.

2.6 Algoritma Pelatihan pada DBN

Menggunakan algoritma DBN yang secara *supervised*. Hal tersebut dilakukan karena pada data pelatihan sudah terdapat label yang nantinya juga akan digunakan pada lapisan output jaringan. Meskipun demikian *pre-training* tetap dilakukan dengan mengatur tumpukan RBM yang digunakan dalam DBN menyesuaikan data yaitu dengan Bernoulli (Biner) RBM [5].

2.6.1 Inisialisasi RBM dan Kontruksi DBN

Pada RBM pertama, lapisan terlihat v_1 mendapatkan inputan langsung dari sampel data x . Dari lapisan terlihat v_1 dan lapisan tersembunyi h_1 pada RBM dilakukan pelatihan untuk membentuk bobot w_1 .

Setelah RBM pertama selesai dilatih, dengan mengambil pola aktivitas pada lapisan tersembunyi h_1 RBM pertama, kemudian dapat digunakan sebagai data pada lapisan terlihat v_2 untuk melatih RBM kedua. Dengan menyalin h_1 sebagai v_2 pada RBM kedua, maka bobot w_2 akan dapat dilatih. Proses ini juga dapat dikatakan membentuk tumpukan (stack) RBM. Perhatikan bahwa pada proses ini label tidak terlibat [6].

2.6.2 Pelatihan Setiap Lapis RBM

Pada konstruksi DBN sudah dijelaskan pembelajaran lapis-demi-lapis pada DBN dengan melakukan pelatihan pada setiap lapisan RBM. Pelatihan lapis-demi-lapis ini bertujuan untuk membentuk bobot-bobot yang menghubungkan lapisan input dengan lapisan tersembunyi pertama dan lapisan tersembunyi kedua (Karena terdapat 2 lapisan tersembunyi pada DBN).

Tahapan latihan ini dapat dikatakan sebagai tahapan *pre-training* mengacu pada penelitian yang dilakukan oleh [7], dimana pelatihan setiap lapis. RBM dilakukan secara *Unsupervised Learning* tanpa melibatkan kelas label atau target [8].

2.6.3 Mini-Batch Gradien Descent

Pembaharuan bobot pada RBM mengacu pada aturan pembaharuan Gradient Descent. Pada gradient descent biasa atau batch gradient descent untuk seluruh data akan dikenakan penjumlahan gradien dalam suatu perulangan sebanyak epoch yang diinginkan. Gradient merupakan turunan parsial untuk setiap bobot dan bias. Besarnya gradien dapat memberikan petunjuk tentang berapa banyak bobot atau bias yang harus berubah.

2.6.4 Fine-tuning Backpropagation

Karena DBN dapat digunakan secara supervised (setelah melalui proses pretraining) dengan melibatkan kelas label bobot akan disempurnakan atau proses ini disebut dengan fine-tuning. Fine-tuning dilakukan dengan melatih jaringan menggunakan algoritma Backpropagation [9][10].

Pada proses sebelumnya, hasil pembobotan yang dilakukan pada saat pelatihan setiap lapisan RBM digunakan dalam inialisasi fine-tuning dengan Backpropagation ini. Namun, untuk bobot yang menghubungkan lapisan tersembunyi kedua dengan lapisan output tetap menggunakan inialisasi random.

Dengan demikian, jaringan DBN dapat dianggap seperti MLP dengan dua lapisan tersembunyi dan satu lapisan output yang terhubung secara penuh. Gambar 4. menunjukkan tahapan fine-tuning pada keseluruhan jaringan DBN yang terhubung secara penuh.



Gambar 4 Fine-tuning secara supervised DBN

3. HASIL DAN PEMBAHASAN

3.1 Pengujian Klasifikasi Tweet

Bagian ini membahas tentang pengujian dari klasifikasi tweet yang telah dibangun. Pengujian klasifikasi tweet dilakukan dengan mengukur akurasi, presisi, *recall*, dan *f1-score* dari hasil perhitungan *deep belief network*, *naïve bayes classifier* dan *support vector machine*.

Adapun total data tweet yang digunakan untuk pengujian klasifikasi (*uji*) adalah 2378 tweet. Total data tweet untuk proses pengujian tersebut dibagi ke dalam delapan trending topik yang telah diinputkan sebelumnya oleh *user*. Rincian data tweet untuk proses pengujian klasifikasi tweet dan pengujian tweet diperlihatkan pada Tabel 2.

Tabel 1 Rincian sentiment tweet pengujian klasifikasi

Tranding Topik	Jumlah Tweet	Sentimen		
		Positif	Negatif	Netral
#AhokDijegal	105	26.00%	19.70%	54.30%
#JakartaKece	201	32.20%	11.60%	53.20%
#Menuju412	45	8.70%	54.30%	37.00%
#SalamBudaya	155	45.50%	12.80%	41.70%
#SidangAhok	550	23.50%	43.40%	33.20%
#SubuhBerjamaah1212	106	34.30%	3.60%	62.00%
Anies	682	36.20%	13.90%	49.90%
Hari Bebas Kendaraan Bermotor	179	9.50%	14.50%	76.00%
Total	2023	26.99%	21.73%	50.91%

Sebelum dilakukan pengujian terhadap klasifikasi tweet, seluruh data tweet dilabeli secara manual sesuai berdasarkan jenis klasifikasinya (sentimennya). Sebagai contoh komentar “*penjara Ahok*” maka akan dilabeli dengan “-” karena memiliki nilai sentimen negatif yang berisi tentang penjarakan Ahok.

Pengujian klasifikasi alasan untuk mengukur akurasi, presisi, *recall*, dan *f1-score* diperoleh dengan membandingkan tiap tweet yang telah dilabeli secara manual dengan hasil perhitungan *deep belief network* yang dilakukan oleh sistem. Jumlah tweet yang sesuai antara hasil perhitungan *deep belief network* oleh sistem dengan pelabelan secara manual, akan mempengaruhi nilai akurasi, presisi, *recall* dan *f1-score* yang diperoleh. Semakin besar jumlah tweet yang sesuai, maka semakin tinggi pula nilai akurasi, presisi, *recall* dan *f1-score* yang didapatkan. Rekapitulasi hasil pengujian klasifikasi perhitungan klasifikasi tweet dengan menggunakan *deep belief network* untuk tiap trending topik diperlihatkan pada Tabel 2.

Tabel 2 Hasil Pengujian Klasifikasi Menggunakan DBN

Tranding Topik	Akurasi	Presisi	Recall	F1-score
#AhokDijegal	49.83%	25%	50%	33%
#JakartaKece	53.20%	28%	53%	37%
#Menuju412	36.96%	14%	37%	20%
#SalamBudaya	41.67%	17%	42%	25%
#SidangAhok	33.17%	11%	33%	17%
#SubuhBerjamaah1212	62.05%	38%	62%	48%
Anies	49.93%	25%	50%	33%
Hari Bebas Kendaraan Bermotor	75.98%	58%	76%	66%
Total	50.35%	27%	50%	35%

Berdasarkan Tabel 3 di atas dapat disimpulkan bahwa hasil akurasi perhitungan klasifikasialasan secara keseluruhan dengan menggunakan *deep belief network* yaitu sebesar 50,35%, presisi 27%, *recall* 50%, sedangkan hasil perhitungan untuk *f1-score* yaitu 35%.

Sedangkan hasil pengujian sistem keseluruhan dari perhitungan klasifikasi dengan menggunakan metode *support vector machine* memperlihatkan bahwa metode *support vector machine* menghasilkan nilai akurasi yang sama besar dengan metode *deep belief network*. Nilai akurasi keseluruhan yang didapatkan dari hasil perhitungan klasifikasi dengan metode *support vector machine* yaitu sebesar 50,35%, nilai presisinya sebesar 27%, nilai *recall* sebesar 50% dan nilai *f1-score* sebesar 35%. Rekapitulasi *confusion matrix* hasil perhitungan klasifikasi menggunakan *support vector machine* dari tiap trending topik ditunjukkan oleh Tabel 3.

Tabel 3 Hasil Pengujian Klasifikasi Menggunakan *Support Vector Machine*

Tranding Topik	Akurasi	Presisi	Recall	F1-score
#AhokDijegal	49.83%	25%	50%	33%
#JakartaKece	53.20%	28%	53%	37%
#Menuju412	36.96%	14%	37%	20%
#SalamBudaya	41.67%	17%	42%	25%
#SidangAhok	33.17%	11%	33%	17%
#SubuhBerjamaah1212	62.05%	38%	62%	48%
Anies	49.93%	25%	50%	33%
Hari Bebas Kendaraan Bermotor	75.98%	58%	76%	66%
Total	50.35%	27%	50%	35%

Sedangkan hasil pengujian sistem keseluruhan dari perhitungan klasifikasi dengan menggunakan metode *Naïve Bayes* memperlihatkan bahwa metode *Naïve Bayes* menghasilkan nilai akurasi yang lebih baik dari pada metode *Deep Belief Network* dan *support vector machine*. Nilai akurasi keseluruhan yang didapatkan dari hasil perhitungan klasifikasi dengan metode *Naïve Bayes* yaitu sebesar 80.16%, presisi 82%, *recall* 70%, *f1-score* 78% , *support* 2373% Rekapitulasi hasil perhitungan klasifikasi menggunakan *Naïve Bayes* dari tiap *tranding topik* ditunjukkan oleh Tabel 4.

Tabel 4 Hasil Pengujian Klasifikasi Menggunakan *Naïve Bayes*

Tranding Topik	Akurasi	Presisi	Recall	F1-score
#AhokDijegal	77.16%	78%	77%	76%
#JakartaKece	79.82%	82%	80%	77%
#Menuju412	80.43%	83%	80%	80%
#SalamBudaya	76.28%	81%	76%	74%
#SidangAhok	79.61%	81%	80%	79%
#SubuhBerjamaah1212	91.57%	92%	92%	91%
Anies	76.53%	78%	77%	75%
Hari Bebas Kendaraan Bermotor	79.89%	83%	80%	74%
Total	80.16%	82%	80%	78%

3.2 Perbandingan Hasil Akurasi

Perbandingan hasil akurasi dari perhitungan klasifikasi dengan menggunakan metode *Deep Belief Network*, *support vector machine*, dan metode *Naïve Bayes* diperlihatkan oleh Tabel 5 Perhitungan klasifikasi dengan menggunakan metode *Naïve Bayes* menghasilkan nilai pengujian yang lebih baik secara keseluruhan dari pada menggunakan metode *Deep Belief Network*, dan *support vector machine*.

Tabel 5 Hasil Perbandingan hasil pengujian klasifikasi

Tranding Topik	Akurasi			Presisi			Recall		
	DBN	SVM	NB	DBN	SVM	NB	DBN	SVM	NB
#AhokDijegal	54.33%	54.33%	77.95%	30%	30%	79%	54%	54%	78%
#JakartaKece	53.20%	53.20%	82.40%	28%	28%	84%	53%	53%	82%
#Menuju412	36.96%	36.96%	71.74%	14%	14%	78%	37%	37%	72%
#SalamBudaya	41.67%	41.67%	70.51%	17%	17%	75%	42%	42%	71%
#SidangAhok	33.17%	33.17%	82.36%	11%	11%	83%	33%	33%	82%
#SubuhBerjamaah1212	62.05%	62.05%	89.16%	38%	38%	89%	62%	62%	89%
Anies	49.93%	49.93%	75.93%	25%	25%	78%	50%	50%	76%
Hari Bebas Kendaraan Bermotor	75.98%	75.98%	80.45%	58%	58%	83%	76%	76%	80%
Total	50.91%	50.91%	78.81%	28%	28%	81%	51%	51%	79%

Hasil perbandingan pengujian sistem keseluruhan dari perhitungan klasifikasi total data dengan menggunakan metode *support vector machine* memperlihatkan bahwa metode *support vector machine* menghasilkan nilai akurasi yang sama besar dengan metode *Deep Belief*

Network. Nilai akurasi keseluruhan yang didapatkan dari hasil perhitungan klasifikasi dengan metode *support vector machine* yaitu sebesar 50.91%, presisi 28%, dan *recall* 51%. Rekapitulasi hasil perhitungan klasifikasi menggunakan *support vector machine* dari tiap *tranding topik* ditunjukkan oleh Tabel 5.

Sedangkan hasil pengujian sistem total tweet keseluruhan dari perhitungan klasifikasi dengan menggunakan metode *Deep Belief Network*, *support vector machine*, dan *Naïve Bayes* memperlihatkan bahwa metode *Deep Belief Network* menghasilkan nilai akurasi yang lebih baik dari pada metode *Naïve Bayes* dan *support vector machine* seperti yang diperlihatkan oleh Tabel 6.

Tabel 6 Hasil Perbandingan hasil pengujian klasifikasi total

Metode	Presisi	recall	F1-score	Akurasi
Deep Belief Network	93%	93%	93%	93.31%
Support Vector Machine	92%	92%	92%	92.18%
Naïve Bayes	79%	79%	79%	79.10%

Nilai akurasi keseluruhan yang didapatkan dari hasil perhitungan klasifikasi dengan metode *Deep Belief Network* yaitu sebesar 93.31%, presisi 93%, *recall* 93%, *f1-score* 93%, *support* 2378, *support vector machine* yaitu sebesar 92.18%, presisi 92%, *recall* 92%, *f1-score* 92%, *support* 2378, dan *Naïve Bayes* yaitu sebesar 79.10%, presisi 79%, *recall* 79%, *f1-score* 79%, *support* 2378.

4. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, maka diperoleh kesimpulan sebagai berikut:

1. Hasil pengujian pada sistem yang dibangun memperlihatkan bahwa metode *Deep Belief Network* memberikan hasil pengujian klasifikasi lebih baik dengan akurasi sebesar 93.31%, presisi 93%, *recall* 93%, *f1-score* 93%, dan *support* 2378, sedangkan hasil pengujian pada sistem menggunakan metode *Naive Bayes* memberikan hasil pengujian klasifikasi dengan akurasi sebesar 79.10%, presisi 79%, *recall* 79%, *f1-score* 79%, *support* 2378 dan *Support Vector Machine* sebesar 92.18%, presisi 92%, *recall* 92%, *f1-score* 92%, *support* 2378.
2. Penggunaan metode DBN dengan menggunakan *BagofWord* sebagai fitur ekstraksinya terbukti tidak memberikan akurasi lebih baik, ketika dibandingkan dengan metode *Naive Bayes* dan SVM (*Support Vector Machine*) dengan menggunakan *BagofWord* sebagai fitur ekstraksinya pada aplikasi yang dibangun.

5. SARAN

1. Saran yang dapat diberikan untuk penelitian selanjutnya adalah dapat mengembangkan sentimen analisis ini dengan mengkombinasikan metode *Deep learning* yang lain selain DBN untuk mendapatkan hasil akurasi yang lebih baik lagi.
2. Sistem ini tidak dapat digunakan untuk menentukan sentimen *tweet* dari bahasa selain bahasa Indonesia. Diharapkan pada penelitian selanjutnya bisa dikembangkan suatu multilingual sentimen klasifikasi yaitu sistem yang dapat menentukan sentimen dari berbagai bahasa.

DAFTAR PUSTAKA

- [1] B. Liu, "Sentiment Analysis and Subjectivity," *Handb. Nat. Lang. Process.*, no. 1, pp. 1–38, 2010. <http://www.cs.uic.edu/~liub/FBS/NLP-handbook-sentiment-analysis.pdf> [http://people.sabanciuniv.edu/berrin/proj102/1-BLiu-Sentiment Analysis and Subjectivity-NLPHandbook-2010](http://people.sabanciuniv.edu/berrin/proj102/1-BLiu-Sentiment%20Analysis%20and%20Subjectivity-NLPHandbook-2010) [Online]. Available: <http://www.cs.uic.edu/~liub/FBS/NLP-handbook-sentiment-analysis.pdf> [Accessed: 20-Apr-2017].
- [2] N. D. 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 [Online]. Available: <https://jurnal.ugm.ac.id/ijccs/article/view/3499>. [Accessed: 21-Apr-2017].
- [3] A. Ng, J. Ngiam, C. Y. Foo, Y. Mai, C. Suen, A. Coates, A. Maas, A. Hannun, B. Huval, T. Wang, and Sameep Tandon, "Deep Learning Tutorial," *Univ. Stanford*, 2015 [Online]. Available: <http://ufldl.stanford.edu/tutorial/supervised/ConvolutionalNeuralNetwork/>. [Accessed: 18-Jan-2017].
- [4] Yuming Hua, Junhai Guo, and Hua Zhao, "Deep Belief Networks and deep learning," *Proc. 2015 Int. Conf. Intell. Comput. Internet Things*, pp. 1–4, 2015 [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7111524>. [Accessed: 6-Mar-2017].
- [5] A. A. Al Sallab, R. Baly, and H. Hajj, "Deep Learning Models for Sentiment Analysis in Arabic," *ANLP Work. ...*, no. November, pp. 9–17, 2015 [Online]. Available: <http://www.aclweb.org/anthology/W15-32#page=21>. [Accessed: 22-Feb-2017].
- [6] G. Hinton, "A Practical Guide to Training Restricted Boltzmann Machines A Practical Guide to Training Restricted Boltzmann Machines," *Computer (Long. Beach. Calif.)*, vol. 9, no. 3, p. 1, 2010 [Online]. Available: http://link.springer.com/10.1007/978-3-642-35289-8_32. [Accessed: 10-Jan-2017].
- [7] D. Erhan, A. Courville, and P. Vincent, "Why Does Unsupervised Pre-training Help Deep Learning?," *J. Mach. Learn. Res.*, vol. 11, pp. 625–660, 2010 [Online]. Available: <http://arxiv.org/abs/1206.5538>. [Accessed: 7-Feb-2017].
- [8] G. Tzortzis and A. Likas, "Deep Belief Networks for Spam Filtering," *19th IEEE Int. Conf. Tools with Artif. Intell. 2007*, pp. 306–309, 2007 [Online]. Available: <http://www.annualreviews.org/doi/10.1146/annurev-statistics-010814-020120>. [Accessed: 21-Apr-2017].
- [9] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A Fast Learning Algorithm for Deep Belief Nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006 [Online]. Available: <http://www.mitpressjournals.org/doi/10.1162/neco.2006.18.7.1527>. [Accessed: 15-Mar-2017].
- [10] R. Salakhutdinov, "Learning Deep Generative Models," *Mit.Edu*, pp. 1–84, 2009 [Online]. Available: <http://www.annualreviews.org/doi/10.1146/annurev-statistics-010814-020120>. [Accessed: 3-Jan-2017].