

# Klasifikasi Sentimen pada Twitter dalam Membantu Pemilihan Kandidat Karyawan dengan Menggunakan Convolutional Neural Network dan Fasttext Embeddings

Farhan Alfariqi<sup>1</sup>, Warih Maharani<sup>2</sup>, Jati Hiliamsyah Husen<sup>3</sup>

<sup>1,2,3</sup>Fakultas Informatika, Universitas Telkom, Bandung

<sup>1</sup>farhanalfa@students.telkomuniversity.ac.id, <sup>2</sup>wmaharani@telkomuniversity.ac.id,

<sup>3</sup>jatihusen@telkomuniversity.ac.id

---

## Abstrak

Karyawan merupakan elemen utama yang dapat meningkatkan nilai dari suatu perusahaan. Menemukan karyawan yang berkualitas umumnya memerlukan proses yang cukup rumit. Proses penerimaan karyawan menjadi tahapan awal yang berperan penting dalam mewujudkan hal tersebut. Proses penerimaan yang buruk akan berdampak pada tingginya angka resign karyawan yang biasanya disebut dengan nilai turnover. Permasalahan tersebut dapat diatasi salah satunya dengan cara mengamati kandidat karyawan melalui aktifitas sosial media. Berdasarkan hasil analisis tersebut, perusahaan dapat secara langsung mengetahui sikap dan sifat dari tiap calon karyawan, sehingga dapat menjadi pertimbangan dalam memilih karyawan. Data yang digunakan dalam penelitian diambil dari media sosial Twitter sebanyak 3583 komentar yang dibagi menjadi tiga sentimen yaitu positif, netral dan negatif. Komentar yang diperoleh direduksi dengan cara menghapus kata, angka maupun simbol yang tidak memiliki informasi dan makna yang penting. Kumpulan kata hasil reduksi diubah menjadi bentuk vektor dengan menggunakan fasttext. Hasil dari fasttext menjadi input pelatihan model Convolutional Neural Network.

Berdasarkan hasil pengujian yang dilakukan pada 20 calon karyawan, kombinasi fasttext dan Convolutional Neural Network dengan beberapa aturan hyperparameter mampu memberikan nilai akurasi terbaik sebesar 84.01%. Nilai akurasi yang dihasilkan tersebut dapat menjadi patokan bahwa model yang dibangun mampu membantu pihak perusahaan dalam memilih calon karyawan.

**Kata kunci :** sentimen, karyawan, seleksi, fasttext, CNN

---

## Abstract

Employees are the main element that can increase the value of a company. Finding qualified employees generally requires a somewhat complicated process. The recruitment process is the initial stage that plays an essential role in realizing this. The imperfect admission process will impact the high employee resign rate, which usually refers to the turnover value. One of the problems can overcome by observing employee candidates through social media activities. Based on the results of the analysis, the company can directly know each employee's attitudes, so that it can consider in selecting employees.

The data used in the study were taken from Twitter social media as many as 3583 comments divide into three sentiments, namely positive, neutral, and negative. Observations obtained reduce by removing words, numbers, or symbols that do not have valuable information and meaning. The deduction words are converted into vector shapes using fasttext. The results of fasttext become training input for the Convolutional Neural Network model.

Based on the results of tests conducted on 20 prospective employees, the combination of fasttext and Convolution Neural Network with several hyperparameter rules can provide the best accuracy value of 84.01%. The resulting accuracy can be a benchmark that the model built is ready to help the company select prospective employees.

**Keywords:** sentiment, employees, selection, fasttext, CNN

---

## 1. Pendahuluan

### Latar Belakang

Karyawan merupakan faktor yang sangat sentral dalam suatu perusahaan. Karyawan akan menjadi elemen utama dalam meningkatkan produktivitas dari perusahaan tersebut. Hal ini menuntut setiap perusahaan mempunyai karyawan yang berkualitas dan berkompeten dalam mencapai tujuan perusahaan. Faktanya, mendapatkan individu dengan talenta yang berkualitas tinggi sangatlah langka dan kebanyakan perusahaan kesulitan mendapatkannya [1].

Kekurangan karyawan yang mumpuni merupakan masalah serius bagi pertumbuhan perusahaan di masa depan. Kegagalan yang ditimbulkan akan menyebabkan tingginya persentase pengunduran diri karyawan yang sering disebut dengan nilai *turnover*.

Beragam metode mulai dari ujian psikotes hingga kuisioner dilakukan demi mendapatkan karyawan yang berkualitas [1]. Perkembangan teknologi juga turut serta memberikan solusi baru bagi pihak *Human Resource*, yaitu salah satunya menganalisis data diri kandidat secara 'sembunyi' atau *underground research* melalui media sosial [2]. Analisis tersebut dilakukan dengan cara mengklasifikasi sentimen tiap kandidat dalam memberikan opini terhadap suatu permasalahan, khususnya *twitter* sebagai media sosial *micro-blogging* dengan *traffic* paling tinggi<sup>1</sup>. Hasil klasifikasi tersebut secara tidak langsung akan memberikan gambaran bagaimana kandidat bereaksi terhadap suatu permasalahan, yang nantinya dapat memberikan informasi apakah kandidat tersebut layak atau tidak berada dalam lingkungan perusahaan [2]. Klasifikasi sentimen merupakan bagian *Natural Language Processing* (NLP) yang bekerja dengan cara memetakan suatu teks kedalam salah satu label sentimen seperti positif, netral maupun negatif [3]. Pengamatan melalui media sosial dinilai cukup efektif karena dapat memberikan gambaran umum mengenai cara kandidat dalam menanggapi opini publik tentang topik-topik tertentu [4]

Penelitian sebelumnya membahas mengenai klasifikasi sentimen dengan membandingkan performa model *Convolutional Neural Network* (CNN) dengan jenis *word embeddings* yang berbeda [4]. CNN dipilih karena memiliki struktur *hidden layer* yang lebih banyak dibandingkan dengan *Neural Network* lain, sehingga dapat menangani permasalahan kompleks dengan lebih baik [5]. Penelitian tersebut juga menjelaskan bahwa penggunaan *fasttext* [5] memiliki hasil yang lebih baik jika dibandingkan dengan *word2vec* [6]. *Fasttext* bekerja dengan prinsip *n-grams* [7] yang memudahkan mesin dalam menemukan bentuk vektor dari kata yang belum dikenali dalam korpus. Kekurangan dari penelitian sebelumnya yaitu lebih berfokus pada pengaruh *word embeddings* yang digunakan, namun tidak menjelaskan mengenai konfigurasi *hyperparameter* yang digunakan.

Penelitian ini menggunakan komentar sebanyak 3582 data berbahasa Indonesia yang diperoleh melalui proses *crawling* secara bertahap. Pemberian label dilakukan secara manual dengan tiga kategori yaitu label positif, netral dan negatif. Data yang digunakan dibagi menjadi data latih dan data uji. Data latih digunakan dalam membangun model prediksi, sedangkan data uji berfungsi sebagai metrik evaluasi untuk mendapatkan nilai *accuracy*, *presicion*, *recall* dan *f1-score* dari model yang dibangun.

### Topik dan Batasannya

Berdasarkan latar belakang yang telah dijelaskan sebelumnya, maka topik yang dibahas pada penelitian ini yaitu menerapkan *fasttext* sebagai *word embeddings* bersama *convolutional neural network* sebagai model klasifikasi kasus komentar berbahasa Indonesia dalam memahami sentimen kandidat karyawan guna membantu pihak *Human Resource* dalam mengambil keputusan mengenai penerimaan karyawan baru.

Adapun batasan masalah pada penelitian ini adalah sebagai berikut :

- a. Data yang digunakan berasal dari Twitter
- b. Data yang digunakan hanya komentar yang berbahasa Indonesia
- c. Tidak mengidentifikasi simbol, url link, hashtag, gambar maupun video
- d. Pemberian label pada data dilakukan secara manual dengan tiga kategori : positif, netral dan negatif

### Tujuan

Penelitian ini bertujuan mengevaluasi kinerja *fasttext* sebagai *word embeddings* dengan model klasifikasi *convolutional neural network* dalam klasifikasi sentimen tiap kandidat karyawan pada *twitter* berbahasa Indonesia untuk menemukan kandidat yang sesuai dan berkompoten.

### Organisasi Tulisan

Urutan penulisan dalam penelitian ini adalah sebagai berikut : bagian 2 menunjukkan studi literatur mengenai teori yang digunakan, bagian 3 menjelaskan sistem yang akan dibangun yaitu proses *preprocessing* data awal menjadi data yang siap dilatih, dan klasifikasi sentimen pada *twitter* dengan menggunakan *fasttext* sebagai *word embeddings* serta *convolutional neural network* sebagai model klasifikasi. Bagian 4 akan menjelaskan mengenai hasil pengujian yang telah dilakukan, serta bagian 5 akan menjelaskan mengenai kesimpulan dan saran untuk penelitian kedepannya.

---

<sup>1</sup><https://www.alexandria.com/siteinfo/twitter.com>

## 2. Studi Terkait

Penelitian terkait yang menggunakan bahasa Indonesia masih sangat sulit ditemukan. Hal tersebut dikarenakan *fasttext* termasuk kedalam metode yang masih baru, sehingga beberapa penelitian umumnya lebih banyak menggunakan *word2vec* sebagai *word embeddings*. Secara umum, penelitian yang menggunakan *fasttext* sebagai *word embeddings* dengan *convolutional neural network* sebagai model klasifikasi banyak dijumpai pada penelitian yang bersifat internasional. Penelitian tersebut menggunakan data yang diambil dari *public dataset* yang tersedia seperti dari MR (*Movie Review*), dan SST (*Stanford Sentiment Treebank*) [8].

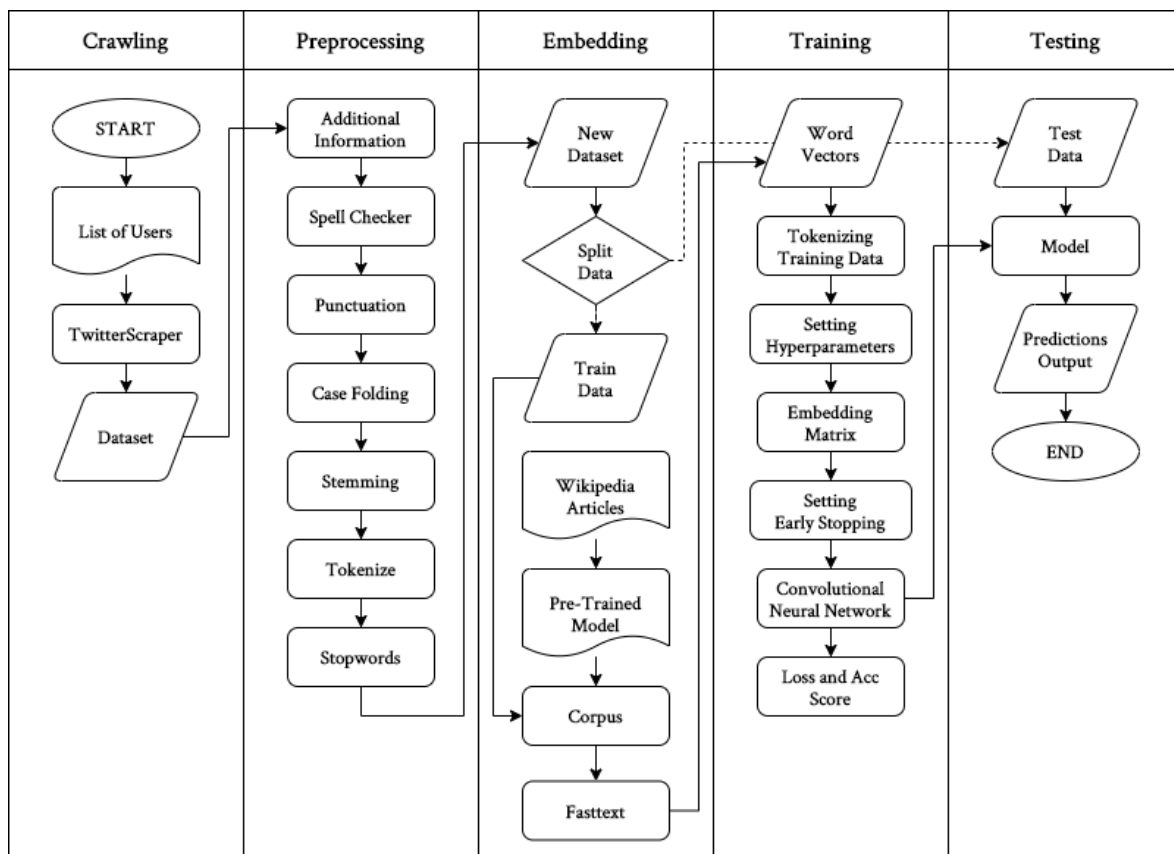
Penelitian yang dilakukan oleh Ouyang [9] terkait Analisis Sentimen dengan *Convolutional Neural Network* dan *Word2Vec Embeddings* [10], mendapatkan hasil akurasi terbaik sebesar 45.4%. Penelitian berikutnya dilakukan oleh Santos [4] terkait Analisis Sentimen dengan *Convolutional Neural Networks* dan *fasttext Embeddings*, menggunakan tiga jenis data yang berbeda yaitu MR, SST-1 dan SST-2 mendapatkan akurasi masing-masing sebesar 85.2%, 65.0%, dan 84.5%. Kedua penelitian tersebut menggunakan *public dataset* yang sama yaitu MR (*Movie Review*) yang diambil dari *rottentomatoes.com*

Berdasarkan kedua penelitian diatas, penggunaan *embeddings* yang berbeda memberikan pengaruh yang sangat signifikan khususnya pada data MR. *Fasttext* mendapatkan akurasi sebesar 85.2% yang jauh mengungguli *word2vec* pada 45.4%. Hasil perbandingan tersebut menunjukkan bahwa *fasttext* dengan kemampuan menganalisis kata berdasarkan struktur subkata pembentuknya mampu memberikan performansi yang lebih baik sebagai *word embeddings* pada model klasifikasi [7].

## 3. Sistem yang Dibangun

### 3.1 Tahapan Perancangan Sistem

Rangkaian proses dalam sistem yang dibangun terdapat pada Gambar 1 berikut :



Gambar 1. Alur Kerja Sistem

### 3.2 Pengumpulan Data

Data yang digunakan merupakan dokumen berbahasa Indonesia. Data diambil dari *Twitter* dengan cara *crawling* (mengumpulkan data berdasarkan kriteria tertentu) menggunakan *library TwitterScraper* yang dikembangkan oleh Ahmet Taspinar<sup>2</sup>. *TwitterScraper* menjadi pilihan dikarenakan memberikan informasi yang lebih detil dibandingkan dengan menggunakan *library* yang disediakan oleh pihak *Twitter*. Total data yang berhasil terkumpul berjumlah sebanyak 3582 *tweet* dalam bentuk tidak terstruktur serta belum memiliki informasi label sentimen.

Data dibagi menjadi dua bagian, yaitu 2900 *tweet* sebagai data latih serta 682 *tweet* sebagai data uji. Data uji digunakan sebagai tolak ukur untuk menghitung keberhasilan dari model klasifikasi yang telah dibangun dengan menggunakan data latih. Berikut merupakan Tabel 1 yang menggambarkan *tweet* yang berhasil diperoleh.

**Tabel 1.** Hasil Proses Crawling

No.	Tweet
1	Selamat jalan kang @didikempotid karya karyamu akan abadi. Saya bersaksi njenengan orang baik, meninggal di bulan baik, innalillahi wa inailaihi roji'un
2	Tak Banyak yang Tahu, Ini Fakta Perjalanan Karier Didi Kempot, Ngutang di Warung Hingga Tidur di Kuburan
3	Begitu parahkah kita? Sampai untuk jujur saja harus diminta-minta & dimohon-mohon? Ya, ampun

### 3.3 Data Preprocessing

Preprocessing dilakukan setelah data diberi label sentimen. Tahapan ini berfungsi untuk menghilangkan beragam masalah yang dapat mengganggu hasil dari model yang dibangun. Proses yang dilakukan yaitu mengubah maupun menghapus angka, huruf, simbol serta kata yang tidak memiliki makna, atau bernilai informasi rendah dengan menggunakan *library PySastrawi*<sup>3</sup>. Contoh proses yang dilakukan yaitu seperti pada Tabel 2 berikut.

**Tabel 2.** Hasil Data Preprocessing

Jenis	Input	Output
Additional Information Removal	Semangat teman-teman pengusahaaa, pasti tidak mudah, tapi insya Allah dihitung sebagai pemberat timbangan di akhirat kelak, Aamiin <a href="https://twitter.com/HendySetiono/status">https://twitter.com/HendySetiono/status</a>	Semangat teman-teman pengusahaaa, pasti tidak mudah, tapi insya Allah dihitung sebagai pemberat timbangan di akhirat kelak, Aamiin
Spell Checker	Semangat teman-teman pengusahaaa, pasti tidak mudah, tapi insya Allah dihitung sebagai pemberat timbangan di akhirat kelak, Aamiin	Semangat teman-teman pengusahaaa, pasti tidak mudah, tapi insya Allah dihitung sebagai pemberat timbangan di akhirat kelak, Aamiin
Punctuation Removal	Semangat teman-teman pengusahaaa, pasti tidak mudah, tapi insya Allah dihitung sebagai pemberat timbangan di akhirat kelak, Aamiin	Semangat teman teman pengusahaaa pasti tidak mudah tapi insya Allah dihitung sebagai pemberat timbangan di akhirat kelak Aamiin
Case Folding	Semangat teman teman pengusahaaa pasti tidak mudah tapi insya Allah dihitung sebagai pemberat timbangan di akhirat kelak Aamiin	semangat teman teman pengusahaaa pasti tidak mudah tapi insya allah dihitung sebagai pemberat timbangan di akhirat kelak aamiin
Stemming	semangat teman teman pengusahaaa pasti tidak mudah tapi insya allah dihitung sebagai pemberat timbangan di akhirat kelak aamiin	semangat teman teman pengusahaaa pasti tidak mudah tapi insya allah hitung bagai berat timbang di akhirat kelak aamiin
Stopword Removal	semangat teman teman pengusahaaa pasti tidak mudah tapi insya allah hitung bagai berat timbang di akhirat kelak aamiin	semangat teman teman pengusahaaa tidak mudah insya allah hitung berat timbang akhirat kelak aamiin

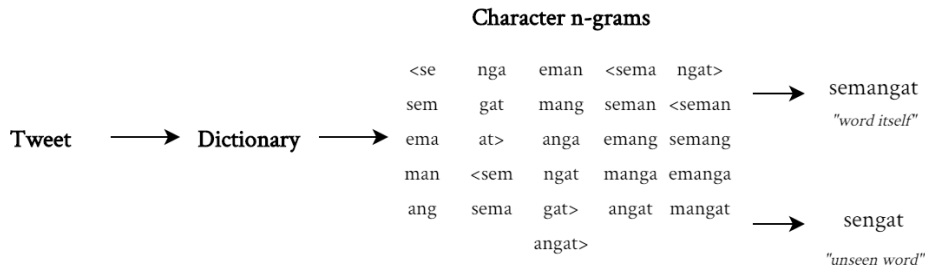
<sup>2</sup><https://github.com/taspinar/twitterscraper>

<sup>3</sup><https://github.com/har07/PySastrawi>

### 3.4 Fasttext Embedding

Proyek *open-source*<sup>4</sup> yang dikembangkan oleh Tim Facebook Research Lab sebagai metode yang efektif dan cepat dalam melakukan vektorisasi kata maupun klasifikasi teks [5] [7]. Dalam merepresentasi kata, *fasttext* memiliki cara kerja yang berbeda dibandingkan *word2vec* [6].

Berdasarkan penelitian dari Bojanowski [7], nilai vektor kata direpresentasikan dengan cara menjumlahkan nilai dari tiap *n-grams*. Kata-kata yang tidak ditemui dalam korpus dapat direpresentasikan dengan baik karena sangat memungkinkan beberapa *n-grams* pembentuk kata tersebut muncul pada *n-grams* yang berada didalam korpus.

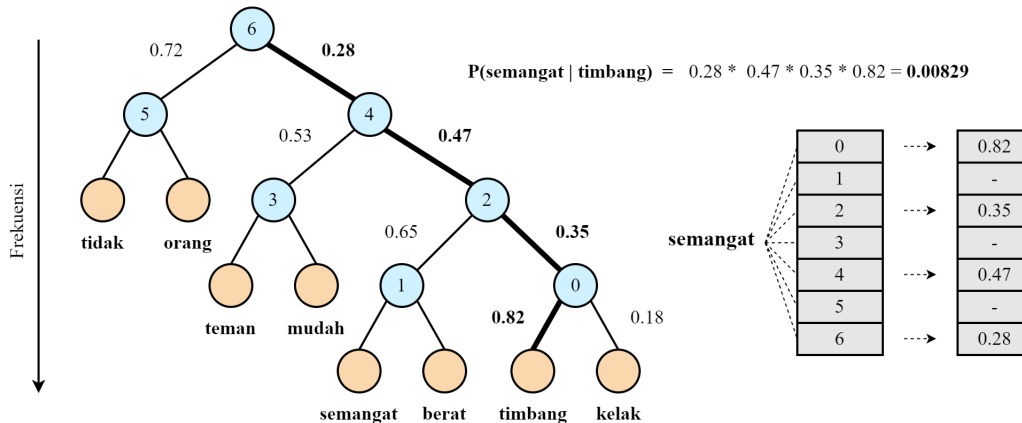


Gambar 2. Ilustrasi Cara Kerja Character n-grams pada Fasttext

Misalkan variabel  $h_w$  menandakan kata yang ingin dicari vektor representasinya, variabel  $G_w$  merupakan kumpulan *n-grams* yang muncul pada kata tersebut, variabel  $z_g$  merupakan representasi vektor tiap *n-grams*, sedangkan  $v_c$  merupakan vektor kata dari *context word*  $c$ . Representasi vektor dari kata tersebut dapat dihitung dengan menggunakan Rumus 1 berikut.

$$h_w = \sum_{g \in G_w} z_g, \tag{1}$$

Gambar 3 berikut merupakan ilustrasi dari cara kerja *Hierarchical Softmax* pada *Fasttext* dengan menerapkan metode *Huffman Tree*. Bekerja dengan cara membangun *binary tree* yang memposisikan kedalaman *leaf node* berdasarkan frekuensi kemunculan kata didalam korpus. Kata yang memiliki frekuensi paling sedikit akan berada pada level kedalaman yang lebih tinggi, begitupun sebaliknya.



Gambar 3. Ilustrasi Cara Kerja Hierarchical Softmax pada Fasttext

**Tabel 3.** Informasi Detil dari fasttext. minn: Nilai Minimum Char n-gram. maxn: Nilai Maksium Char n-gram. dim: Ukuran Vektor Kata. lr: Learning Rate. epoch: Jumlah Epoch. t: Ukuran threshold. ws: Ukuran Context Window. loss: Loss Function

min	maxn	dim	lr	epoch	t	ws	loss
3	6	100	1.0	10	0.0001	5	Hierarchical Softmax

<sup>4</sup><https://github.com/facebookresearch/fastText>

Berdasarkan penelitian dari Kim [8], salah satu cara utama dalam menangani jumlah data yang sedikit yaitu dengan melibatkan *pre-trained model* dari data yang lain. *Pre-trained Model* pada penelitian berikut menggunakan model skipgram dengan melibatkan 100.000 artikel berbahasa Indonesia. Artikel tersebut diambil dari *public dataset* yang tersedia dari *Wikipedia*<sup>5</sup>. Proses *training* memakan waktu selama 481.59 detik, dengan hasil 199.831 kata berukuran 100 dimensi yang nantinya digunakan pada model klasifikasi untuk meningkatkan performansi model.

**Tabel 4.** Informasi Detil dari Data. *c*: Jumlah Klasifikasi.  $l_{max}$ : Panjang Maksimum suatu Kalimat. *V*: Ukuran Korpus.  $V_{pre}$ : Ukuran Korpus Total. *N*: Jumlah Data. Train: Jumlah Data Latih. Test: Jumlah Data Uji

Data	Pre-trained	c	$l_{max}$	V	$V_{pre}$	N	Train	Test
Twitter	Wikipedia	3	42	7495	201428	3582	2900	682

Proses *training* dilakukan dengan menggunakan data dari *Twitter* bersamaan dengan *pre-trained model* yang telah disiapkan. Hasil akhir dari *training* tersebut merupakan vektorisasi kata dengan ukuran 100 dimensi seperti pada Tabel 5 berikut.

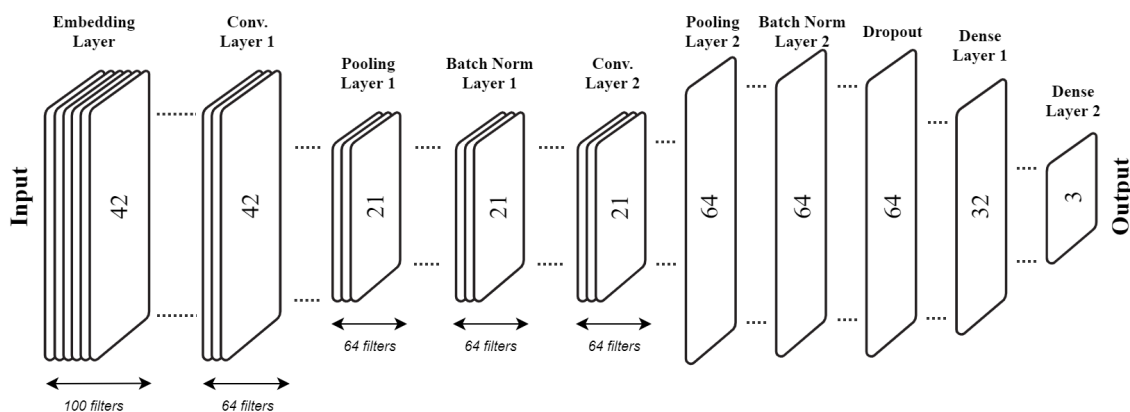
**Tabel 5.** Hasil FastText Embedding

No.	Kata	Dimensi				
		1	2	3	...	100
1	semangat	2.5999	-0.1314	-0.9639	...	-1.1510
2	teman	-0.7549	1.6747	0.5660	...	2.8640
3	aamiin	0.9544	-0.2347	-2.9392	...	-0.1756

### 3.5 Convolutional Neural Network

Motivasi utama dalam penerapan *convolutional neural network* diperoleh dari hasil menjanjikan dari penelitian terkini yang dilakukan oleh Bojanowski [7] dan Kim [8]. Berbeda dengan bidang *Computer Vision* yang menggunakan gambar sebagai *input layer*. Pada bidang NLP, gambar tersebut digantikan dengan dokumen yang berisikan sekumpulan teks. Dokumen tersebut direpresentasikan menjadi bentuk matriks. Tiap baris merepresentasikan kata yang berada dalam tiap kalimat pada dokumen, sedangkan kolom merepresentasikan dimensi dari kata tersebut.

Penelitian ini menggunakan 2 *convolutional layers*, 2 *pooling layers*, serta penambahan 2 *batch normalization layers* yang bersifat opsional. Penelitian dibangun dengan menggunakan *TensorFlow* serta dijalankan pada *Google Colaboratory*<sup>6</sup> dengan menggunakan pengaturan GPU dari *Google*. Arsitektur jaringan yang digunakan dapat dilihat pada Gambar 4 berikut.



**Gambar 4.** Arsitektur CNN dengan input 42 x 100

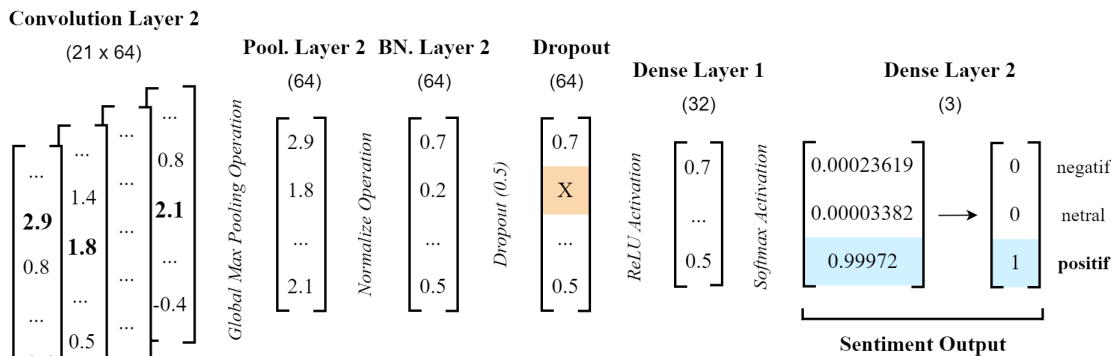
<sup>5</sup><https://dumps.wikimedia.org/idwiki/latest/>

<sup>6</sup><https://colab.research.google.com/notebooks/>



Ketiga operasi yang dijelaskan sebelumnya juga diterapkan pada *layer* kedua seperti pada Gambar 7 dibawah. Perbedaan yang cukup berpengaruh terdapat pada bagian *pooling layer*. Berbeda dengan *layer* pertama yang menggunakan *max pooling layer*, maka pada *layer* kedua berikut menggunakan *global max pooling*, yang bekerja dengan cara mengambil nilai maksimum pada tiap *channel*. Hasil dari proses tersebut akan mereduksi matriks 21 x 64 menjadi bentuk skalar dengan ukuran 64.

Bagian *layer* kedua juga memiliki operasi lain seperti *dropout*. Selama proses *training*, *dropout* akan menjadi bagian yang paling penting dengan memecahkan masalah utama dalam *machine learning*, yaitu *overfitting*. *Dropout* bekerja dengan cara melepaskan *neuron* secara acak selama proses *training* untuk mencegah *co-adaptation* [9]. Hasil dari operasi *dropout* akan dilanjutkan oleh *dense layer*. *Dense layer* umumnya diposisikan diakhir proses *training* yang berfungsi untuk mendefinisikan hasil klasifikasi. Secara berurutan, *dense layer* akan mereduksi ukuran skalar 64 menjadi 32 dan diakhiri dengan ukuran 3 sesuai dengan jumlah klasifikasi sentimen. Dapat dilihat pada Gambar 7 berikut, *dense layer* akan menerapkan *softmax* sebagai fungsi aktivasi untuk mendapatkan probabilitas tiap hasil klasifikasi. Berdasarkan hasil tersebut, dapat diketahui bahwa kalimat "semangat teman teman pengusaha tidak mudah insya allah hitung berat timbang akhirat kelak aamiin" bernilai sentimen positif.



Gambar 7. Ilustrasi Proses Pada Layer 2

### 3.6 Confusion Matrix

*Confusion matrix* [11] merupakan suatu konsep dalam *machine learning* yang menjelaskan mengenai informasi klasifikasi aktual dan prediksi dari suatu sistem klasifikasi. *Confusion matrix* diperlukan untuk mengetahui baik atau tidaknya suatu model dalam memberikan prediksi terhadap suatu data uji yang diberikan.

		Predicted	
		True Positive (TP)	False Negative (FN)
Actual	True Positive (TP)	True Positive (TP)	False Negative (FN)
	False Positive (FP)	False Positive (FP)	True Negative (TN)

Gambar 8. Ilustrasi Confusion Matrix

#### 3.6.1 Precision

*Precision* merupakan tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem. Perhitungan mengenai *precision* dapat dilihat pada Persamaan 3 berikut.

$$Precision = \frac{TP}{TP + FP} * 100 \tag{3}$$

#### 3.6.2 Recall

*Recall* merupakan tingkat keberhasilan sistem dalam menemukan kembali suatu informasi. Perhitungan mengenai *recall* dapat dilihat pada Persamaan 4 berikut.

$$Recall = \frac{TP}{TP + FN} * 100 \tag{4}$$



### 3.6.3 F1-Score

*F1-Score* merupakan pengukuran performansi sistem yang menggabungkan nilai *precision* dengan *recall*. Perhitungan mengenai *F1-Score* dapat dilihat pada Persamaan 5 berikut.

$$F1 - Score = \frac{2 * Recall * Precision}{Recall + Precision} \quad (5)$$

### 3.6.4 Accuracy

*Accuracy* merupakan perhitungan data yang bernilai *true* dari keseluruhan jumlah data. Perhitungan mengenai *Accuracy* dapat dilihat pada Persamaan 6 berikut.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} * 100 \quad (6)$$

## 4. Evaluasi

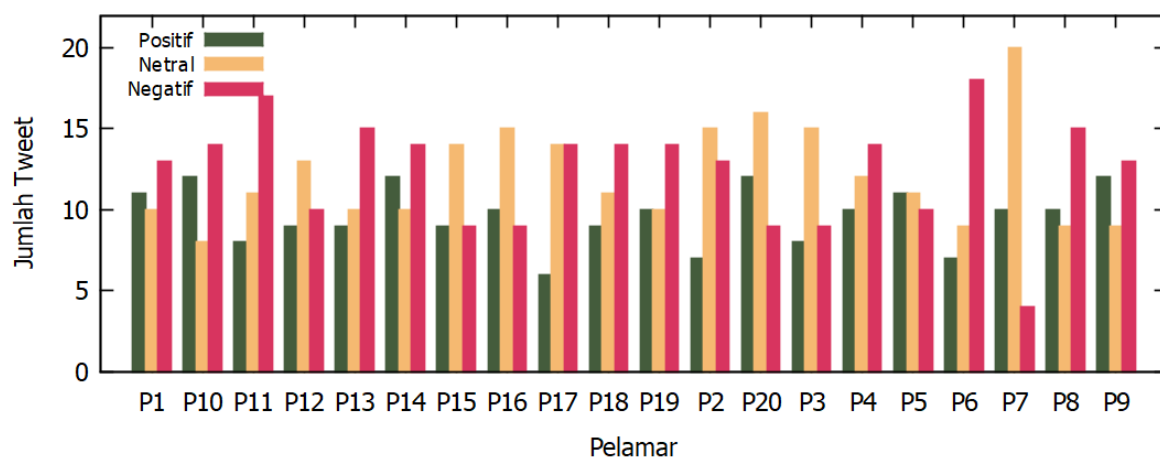
### 4.1 Hasil Pengujian

Tabel 7 berikut, merupakan potongan dari hasil prediksi yang dihasilkan dari data uji yang telah disediakan. Terlihat bahwa pada kelima komentar tersebut mendapatkan prediksi yang cukup tepat. Pembahasan seperti pujian pada komentar pertama merupakan salah satu contoh dari sentimen yang bersifat positif, juga pembahasan seputar penandatanganan petisi pada komentar ketiga umumnya tidak berisikan ujaran yang bersifat reaktif, serta pembahasan seperti ujaran kebencian pada komentar kelima merupakan bukti jelas yang menandakan sentimen negatif. Beberapa komentar seperti komentar kedua dan keempat tidak mendapatkan hasil prediksi yang sesuai. Hasil tersebut dipengaruhi diakibatkan adanya kata (*token*) yang bernilai ekstrim seperti "narkoba" pada komentar kedua yang memiliki makna negatif serta "lindung" yang memiliki makna positif pada komentar kelima.

**Tabel 7.** Hasil Prediksi

No.	Komentar	Label	Prediksi
1	episode doraemon bagus hati hangat ahh nobita doraemoon	Positif	Positif
2	kepala bnn didi kempot teladan nian narkoba	Positif	Negatif
3	komisi lindung anak indonesia kpai audisi pb djarum tandatangani petisi	Netral	Netral
4	rumah dah hapus salur metro tipu	Negatif	Negatif
5	senggol sedikitpun jd langgar karna kiper lindung daerah kotak pinalti	Negatif	Positif

Gambar 9 berikut merupakan hasil visualisasi dari klasifikasi sentimen tiap kandidat karyawan. Visualisasi menampilkan informasi dari 20 data responden dengan masing-masing nilai sentimen tiap komentar yang dimiliki responden tersebut.



**Gambar 9.** Hasil Klasifikasi Sentimen tiap Kandidat Karyawan

## 4.2 Analisis Hasil Pengujian

Berikut merupakan metrik evaluasi hasil pengujian berdasarkan empat jenis pengaturan *hyperparameter* yang berbeda. Informasi dari metrik evaluasi berikut dapat menjadi acuan seberapa baik penelitian yang telah dilakukan.

**Tabel 8.** Hasil Evaluasi. F: Ukuran Filter. BN Layer: Batch Normalization Layer

Hyperparameter	Label	Classification Reports			Accuracy
		Precision	Recall	F1-Score	
7F + BN Layer	Negatif	0.79	0.88	0.83	0.84
	Netral	0.87	0.88	0.87	
	Positif	0.87	0.77	0.81	
7F	Negatif	0.72	0.84	0.78	0.78
	Netral	0.82	0.74	0.78	
	Positif	0.83	0.78	0.80	
9F + BN Layer	Negatif	0.70	0.92	0.79	0.81
	Netral	0.90	0.76	0.83	
	Positif	0.87	0.74	0.80	
9F	Negatif	0.75	0.86	0.80	0.80
	Netral	0.86	0.74	0.79	
	Positif	0.81	0.81	0.81	

Dari Tabel 8 diatas, terlihat bahwa program diuji dengan menggunakan empat jenis *hyperparameter* yang berbeda. *Hyperparameter* yang diujikan yaitu ukuran *filter*, serta keberadaan *batch normalization Layer* untuk normalisasi perhitungan dalam model klasifikasi *convolutional neural network*. Pemilihan ukuran *filter* dipilih berdasarkan jumlah kata yang umumnya muncul dalam suatu kalimat. Suatu frasa kalimat dapat diinterpretasikan maknanya melalui informasi kata yang berdekatan. Pemilihan angka tujuh dan sembilan dijadikan saran *hyperparameter* dikarenakan tujuh maupun sembilan kata yang berdekatan dapat menjadi acuan untuk menilai suatu makna dalam frasa kalimat.

Konfigurasi *hyperparameter* terbaik diperoleh dengan melakukan pengujian menggunakan jenis *hyperparameter* yang pertama. Penambahan *Batch Normalization layer* pada jenis pertama terbukti memberikan hasil yang lebih baik dengan selisih akurasi sebanyak 6% dibandingkan tanpa menambahkan *layer* tersebut seperti pada *hyperparameter* yang kedua. Di sisi lain, penambahan ukuran *filter* juga tidak selalu memberikan hasil yang optimal. Berdasarkan tabel tersebut, *hyperparameter* ketiga dengan jumlah filter yang lebih tinggi justru menurunkan nilai akurasi sebesar 3% dibandingkan dengan *hyperparameter* pertama.

Berdasarkan hasil pengujian yang telah dilakukan sebelumnya. *Hyperparameter* pertama menjadi *hyperparameter* terbaik yang diujikan dikarenakan unggul dalam semua nilai metrik evaluasi dibandingkan ketiga jenis lainnya dengan rata-rata nilai *precision* sebesar 84%, *recall* sebesar 84%, dan *f1-score* sebesar 84%

## 5. Kesimpulan

Berdasarkan uraian hasil penelitian dan pembahasan yang telah dijelaskan sebelumnya, maka dapat diambil kesimpulan sebagai berikut : bahwa kinerja *fasttext embeddings* bersama jenis *hyperparameter* yang tepat pada model klasifikasi *convolutional neural network* memberikan hasil yang sangat baik dalam mengklasifikasi sentimen dengan menggunakan data *Twitter* pada tiap kandidat karyawan. Konfigurasi *hyperparameter* dengan ukuran *filter* sebesar 7, serta adanya penambahan *batch normalization layer* berhasil memperoleh tingkat akurasi yang cukup tinggi yaitu sebesar 84.01%. Pengamatan terhadap tujuh kata berdekatan seperti pada nilai ukuran *filter* memberikan hasil yang optimal dibandingkan dengan menggunakan jumlah kata yang lebih tinggi. Selain itu, penambahan *batch normalization layer* juga memberikan hasil yang lebih baik dibandingkan tanpa menggunakan *layer* tersebut. Hal tersebut dikarenakan, *batch normalization layer* dapat mengurangi *overfitting* pada proses *training*, sehingga hasil akurasi yang didapatkan selama proses pengujian dapat menghasilkan nilai yang lebih optimal.

Pada penelitian berikutnya, penulis menyarankan tiga jenis peningkatan. Pertama, jumlah label pada data yang digunakan pada penelitian ini tidak cukup seimbang. Pada penelitian selanjutnya, penulis menyarankan untuk menyiapkan data dengan jumlah label masing-masing yang seimbang. Kedua, penulis menyarankan untuk melibatkan lebih banyak data selama proses *training*, agar program dapat melatih komentar lebih variatif yang bertujuan meningkatkan *accuracy*. Terakhir, penulis menyarankan untuk melibatkan penerjemahan *slang words* pada proses *preprocessing*. Hal ini disebabkan umumnya pengguna *Twitter* lebih banyak berkomunikasi dengan menggunakan bahasa informal, sehingga timbul berbagai macam jenis *slang* yang sulit untuk diproses oleh program.

## Daftar Pustaka

- [1] Hermin Endratno. Talent management dalam meningkatkan kinerja organisasi. *Sustainable Competitive Advantage (SCA)*, 1(1), 2013.
- [2] Emillia Nurdin, Andi Basru Wawo, and Lilis Julia. Pengaruh pengembangan kapasitas sumber daya manusia aparatur pemerintah daerah terhadap kinerja pegawai (studi pada badan pengelola keuangan dan aset daerah kota kendari). *Jurnal Akuntansi dan Keuangan*, 3(1):1–13, 2018.
- [3] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657, 2015.
- [4] Igor Santos, Nadia Nedjah, and Luiza de Macedo Mourelle. Sentiment analysis using convolutional neural network with fasttext embeddings. In *2017 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, pages 1–5. IEEE, 2017.
- [5] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain, April 2017. Association for Computational Linguistics.
- [6] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In Yoshua Bengio and Yann LeCun, editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013.
- [7] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.
- [8] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October 2014. Association for Computational Linguistics.
- [9] Xi Ouyang, Pan Zhou, Cheng Hua Li, and Lijun Liu. Sentiment analysis using convolutional neural network. In *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, pages 2359–2364. IEEE, 2015.
- [10] Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 115–124. Association for Computational Linguistics, 2005.
- [11] Xinyang Deng, Qi Liu, Yong Deng, and Sankaran Mahadevan. An improved method to construct basic probability assignment based on the confusion matrix for classification problem. *Information Sciences*, 340:250–261, 2016.