



## Pemantauan Perhatian Publik terhadap Pandemi COVID-19 melalui Klasifikasi Teks dengan Deep Learning

Novrindah Alvi Hasanah<sup>1</sup>, Nanik Suciati<sup>2</sup>, Diana Purwitasari<sup>3</sup>

<sup>1,2,3</sup> Teknik Informatika, Fakultas Teknologi Elektro dan Informatika Cerdas, ITS Surabaya

<sup>1</sup>alvinovrindah@gmail.com, <sup>2</sup>nanik@if.its.ac.id, <sup>3</sup>diana@if.its.ac.id

### Abstract

*Monitoring public concern in the surrounding environment to certain events is done to address changes in public behavior individually and socially. The results of monitoring public attention can be used as a benchmark for related parties in making the right policies and strategies to deal with changes in public behavior as a result of the COVID-19 pandemic. Monitoring public attention can be done using Twitter social media data because the users of the media are quite high, so that they can represent the aspirations of the general public. However, Twitter data contains varied topics, so a classification process is required to obtain data related to COVID-19. Classification is done by using word embedding variations (Word2Vec and fastText) and deep learning variations (CNN, RNN, and LSTM) to get the classification results with the best accuracy. The percentage of COVID-19 data based on the best accuracy is calculated to determine how high the public's attention is to the COVID-19 pandemic. Experiments were carried out with three scenarios, which were differentiated by the number of data trains. The classification results with the best accuracy are obtained by the combination of fastText and LSTM which shows the highest accuracy of 97.86% and the lowest of 93.63%. The results of monitoring public attention to the time vulnerability between June and October show that the highest public attention to COVID-19 is in June.*

*Keywords: monitoring public concern, twitter, covid-19, word embedding, deep learning*

### Abstrak

Memantau perhatian publik di lingkungan sekitar terhadap suatu kejadian tertentu dilakukan untuk mengatasi perubahan perilaku publik secara individual maupun sosial. Hasil pemantauan perhatian publik dapat dijadikan tolak ukur oleh pihak-pihak terkait dalam membuat suatu kebijakan maupun strategi yang tepat untuk menghadapi perubahan perilaku publik sebagai efek pandemi COVID-19. Pemantauan perhatian publik dapat dilakukan menggunakan data media sosial Twitter karena pengguna media tersebut cukup tinggi, sehingga dapat mewakili aspirasi publik secara umum. Namun, data Twitter mengandung topik yang bervariasi sehingga diperlukan proses klasifikasi untuk mendapatkan data terkait COVID-19. Klasifikasi dilakukan dengan variasi *word embedding* (Word2Vec dan fastText) dan variasi *deep learning* (CNN, RNN, and LSTM) untuk mendapatkan hasil klasifikasi dengan akurasi terbaik. Data COVID-19 hasil klasifikasi berdasarkan akurasi terbaik dihitung persentasenya untuk mengetahui seberapa tinggi perhatian publik terhadap pandemi COVID-19. Percobaan dilakukan dengan tiga skenario yang dibedakan oleh jumlah data train. Hasil klasifikasi dengan akurasi terbaik didapatkan oleh kombinasi fastText dan LSTM yang menunjukkan akurasi tertinggi sebesar 97.86% dan terendah sebesar 93.63%. Hasil pemantauan perhatian publik pada rentan waktu antara Bulan Juni sampai Bulan Oktober menunjukkan bahwa perhatian publik terhadap COVID-19 tertinggi adalah pada Bulan Juni.

Kata kunci: memantau perhatian publik, twitter, covid-19, word embedding, deep learning

### 1. Pendahuluan

Pemantauan perhatian publik di lingkungan sekitar terhadap suatu kejadian tertentu merupakan satu hal yang penting dilakukan. Selama ini, publik dengan mudah merubah perilaku individual maupun sosial saat menghadapi suatu masalah tertentu, termasuk COVID-19. Wabah COVID-19 telah ditetapkan oleh organisasi

kesehatan dunia atau yang disebut dengan *World Health Organization* (WHO) sebagai pandemi global. Pandemi COVID-19 menjadi salah satu topik penting yang menjadi perhatian khusus diseluruh dunia, termasuk di Indonesia khususnya Kota Surabaya. Hasil pemantauan perhatian publik dapat dijadikan tolak ukur oleh pihak-pihak terkait untuk membuat suatu kebijakan maupun

strategi yang tepat dalam menghadapi perubahan perilaku publik sebagai efek menghadapi pandemi [1]. Beberapa penelitian terdahulu menjelaskan pengaruh aktivitas bermedia sosial publik untuk pengambilan kebijakan oleh pihak-pihak terkait. Seperti pada penelitian [2] yang menjelaskan pentingnya sarana dan prasarana untuk membuat kebijakan publik, salah satunya adalah *feedback* publik yang didapatkan melalui media sosial. Penelitian lainnya yaitu penelitian [3] yang menganalisis *feedback* publik mengenai 100 hari pertama sistem pemerintahan baru DKI Jakarta. Hal ini digunakan untuk melihat seberapa puas publik mengenai perubahan sistem pemerintahan baru DKI Jakarta. Penelitian lainnya yaitu [4] menganalisis komunikasi politik melalui media sosial yaitu Twitter sebagai solusi partisipasi masyarakat, sosialisasi komunikatif, dan ruang diskusi yang terbuka. Pada penelitian tersebut objek penelitian merupakan akun Twitter Walikota Bandung saat itu yaitu Ridwan Kamil.

Pemantauan perhatian publik dapat dilakukan melalui data media sosial Twitter. Hal tersebut karena pengguna media sosial di Indonesia cukup tinggi. Sehingga, media sosial dinilai dapat mewakili aspirasi publik secara umum. Selama ini publik cenderung meluapkan isi hati dan berbagi informasi melalui media sosial [5][6]. Akun Twitter terverifikasi menjadi akun populer yang dimanfaatkan publik untuk mendapatkan informasi mengenai berbagai macam masalah termasuk pandemi COVID-19. Salah satu akun Twitter terverifikasi di Surabaya yang sering menjadi tempat rujukan masyarakat untuk mendapatkan berbagai macam informasi di wilayah Surabaya adalah akun Twitter Suara Surabaya. Suara Surabaya merupakan akun radio berbasis Twitter yang selalu meng-*update* kejadian di wilayah Surabaya dan sekitarnya.

Namun, teks Twitter memiliki kendala jika digunakan sebagai dasar untuk memantau perhatian publik pada topik tertentu. Teks Twitter memiliki panjang teks yang terbatas dan mengandung topik yang bervariasi, sehingga sulit digunakan untuk mendeteksi satu topik tertentu misalnya pandemi COVID-19. Kendala tersebut menyebabkan dibutuhkan proses klasifikasi untuk mengetahui seberapa tinggi perhatian publik terhadap topik COVID-19.

Klasifikasi teks sebenarnya telah banyak dilakukan dengan berbagai macam metode. Namun, klasifikasi teks menggunakan *deep learning* dinilai lebih baik dari *machine learning* [7]. Hal tersebut dibuktikan dengan penelitian terdahulu yang telah dilakukan seperti penelitian [7][8] yang menunjukkan bahwa nilai *recall*, *f-measure* dan *precision* lebih tinggi dari beberapa penelitian sebelumnya yang menggunakan *machine learning* saat diaplikasikan pada objek yang sama. Berbagai macam metode *deep learning* yang digunakan peneliti terdahulu untuk klasifikasi teks yaitu *Convolutional Neural Network* (CNN), *Recurrent*

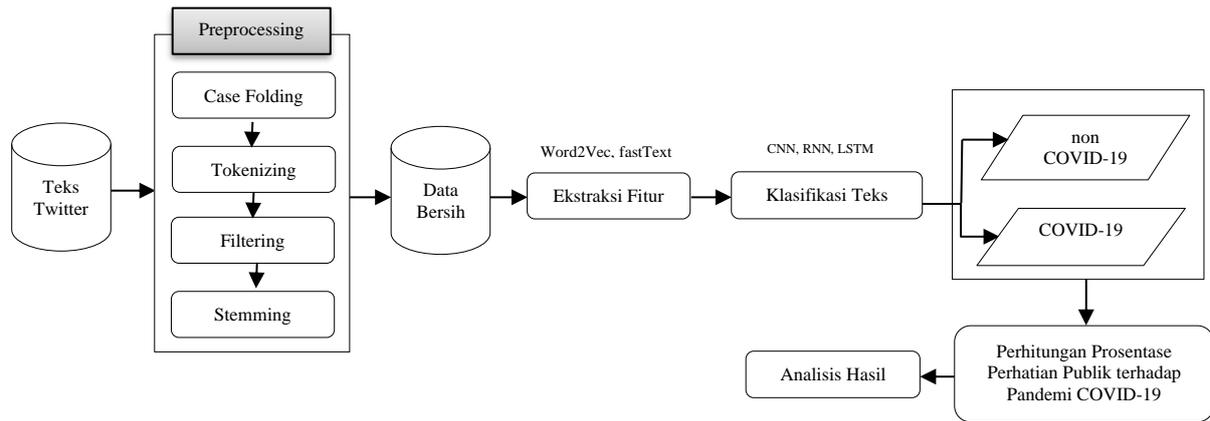
*Neural Network* (RNN) dan *Long-Short Term Memory* (LSTM).

Penelitian [7][9][10] membuktikan bahwa CNN menunjukkan hasil yang baik saat digunakan pada objek teks. CNN sebagian besar memiliki jumlah parameter yang lebih kecil, sehingga CNN dapat dilatih dengan sejumlah kecil data kata [11]. Kelebihan CNN lainnya untuk klasifikasi teks adalah filter konvolusional yang dapat mempelajari fitur yang cocok pada tugas yang diberikan secara otomatis [12].

Namun, penelitian [13] menunjukkan bahwa RNN lebih baik dari CNN. Hal ini dikarenakan RNN dapat menjaga urutan kata [11]. RNN memprediksi kata berikutnya dalam sebuah kalimat karena kemampuan untuk menangkap informasi kontekstual yang baik dalam sekuens [14][15]. RNN memiliki permasalahan yaitu *gradient exploding* atau *vanishing*. Sehingga metode RNN dikembangkan menjadi suatu metode RNN modern yang disebut LSTM. Selain mengatasi masalah pada RNN standar, LSTM juga mempelajari dependensi jangka panjang dari urutan representasi tingkat lebih tinggi yang baik jika dibandingkan dengan model lainnya [16]. Hal tersebut yang menyebabkan LSTM dapat mengatasi masalah yang dialami oleh RNN standar. Penelitian terdahulu yang memanfaatkan LSTM sebagai metode klasifikasi teks adalah penelitian [17][18] yang menunjukkan bahwa LSTM memiliki kinerja yang lebih unggul dari CNN dan RNN standar.

Penelitian ini memanfaatkan *word embedding* sebagai metode ekstraksi fitur yang dikombinasikan dengan metode *deep learning*. Dua metode *word embedding* yang digunakan dalam uji coba yaitu Word2Vec dan fastText. Hal tersebut berdasarkan penelitian [18][10][19] yang memanfaatkan metode tersebut sebagai ekstraksi fitur dan menunjukkan hasil yang unggul jika dibandingkan dengan metode *deep learning* tanpa *word embedding*. Dari berbagai literatur tersebut maka penelitian ini akan membandingkan kinerja variasi *deep learning* (CNN, RNN, dan LSTM) yang dikombinasikan dengan variasi *word embedding* (Word2Vec dan fastText).

Penelitian ini berfokus untuk memantau perhatian publik terhadap pandemi COVID-19 berdasarkan teks Twitter di area Surabaya. Namun, variasi topik pada teks Twitter menyebabkan sulitnya fokus pada satu topik tertentu sehingga membutuhkan proses klasifikasi. Oleh karena itu, penelitian ini mengusulkan pemantauan perhatian publik berdasarkan teks Twitter dengan klasifikasi teks menggunakan variasi *deep learning* dan variasi *word embedding*. Penelitian kami dapat menunjukkan kombinasi metode mana yang memiliki nilai akurasi terbaik. Hasil klasifikasi dengan akurasi terbaik dihitung untuk mengetahui perhatian publik terhadap pandemi COVID-19. Objek penelitian merupakan akun Twitter Suara Surabaya. Pemantauan perhatian publik dilakukan dalam rentan waktu antara Bulan Juni sampai Bulan



Gambar 1. Tahapan Proses Pemantauan Perhatian Publik berdasarkan Hasil Klasifikasi Teks Twitter

Tabel 1. *Preprocessing* Data Teks Twitter Suara Surabaya

Teks Kotor	Output Case Folding	Output Tokenizing	Output Filtering	Output Stemming
Heru Tjahjono Sekretaris Daerah Provinsi Jawa Timur Rabu (29/4/2020) mengabarkan hasil dari evaluasi PSBB hari pertama di sektor dunia kerja.	heru tjahjono sekretaris daerah provinsi jawa timur rabu mengabarkan hasil dari evaluasi psbb hari pertama di sektor dunia kerja	heru tjahjono sekretaris daerah provinsi jawa timur rabu mengabarkan hasil evaluasi psbb hari pertama di sektor dunia kerja	heru tjahjono sekretaris daerah provinsi jawa timur rabu mengabarkan hasil evaluasi psbb hari pertama sektor dunia kerja	heru tjahjono sekretaris daerah provinsi jawa timur rabu kabars hasil evaluasi psbb hari pertama sektor dunia kerja

Oktober. Hasil dari pemantauan perhatian publik dapat dijadikan pihak terkait dalam membangun kebijakan-kebijakan dan strategi tanggap darurat yang tepat untuk menghadapi perubahan perilaku publik sebagai efek pandemi.

## 2. Metode Penelitian

Penelitian ini melakukan beberapa tahap dalam proses pemantauan perhatian publik terhadap pandemi COVID-19 berdasarkan hasil klasifikasi yang ditunjukkan pada Gambar 1. Setiap langkah dijelaskan pada sub-bab berikut.

### 2.1. Persiapan Data

Dataset yang digunakan berupa teks Twitter yang didapatkan dari *timeline* akun Twitter Suara Surabaya mulai dari Bulan Februari 2020 sampai Bulan Oktober 2020. Dataset dikumpulkan dengan teknik *crawling* data menggunakan Twitter API yang disimpan ke dalam format .csv. Data yang berhasil dikumpulkan lebih dari 12000 data yang terdiri dari *tweet* dan *retweet* di *timeline* akun Twitter Suara Surabaya.

### 2.2. Praposes Data

Data yang telah berhasil dikumpulkan di-*preprocessing* untuk menghilangkan *noise* yang tidak dibutuhkan pada proses klasifikasi. Terdapat beberapa tahap *preprocessing* data yang digunakan pada penelitian ini, yaitu:

1. *Case Folding*. Tahap ini merupakan sebuah proses untuk merubah semua huruf kapital menjadi huruf kecil atau *lowercase*. Teks Twitter merupakan teks bebas yang dapat memuat berbagai macam topik dengan berbagai macam bentuk tulisan sesuai dengan apa yang ingin ditulis oleh penggunanya. Sehingga banyak sekali kata yang mengandung huruf besar dan huruf kecil. Selain itu, pada proses *case folding* semua tanda baca seperti koma, titik, tanda seru, tanda tanya dan lain sebagainya serta angka dan semua karakter lain selain huruf dihilangkan.
2. *Tokenizing*. Pada proses ini setiap kata yang ada pada setiap kalimat dipisahkan.
3. *Filtering*. Proses ini adalah suatu proses mengambil kata-kata yang dianggap penting dari hasil *tokenizing* dan membuang *stopword* atau kata sambung seperti

Tabel 2. Label Data Teks Twitter Suara Surabaya

Teks Kotor	Output Preprocessing	Label
Heru Tjahjono Sekretaris Daerah Provinsi Jawa Timur Rabu (29/4/2020) mengabarkan hasil dari evaluasi PSBB hari pertama di sektor dunia kerja.	heru tjahjono sekretaris daerah provinsi jawa timur rabu mengabarkan hasil dari evaluasi psbb hari pertama di sektor dunia kerja	1
Pelaku Usaha Kena Sanksi Kalau Melanggar PSBB	pelaku usaha kena sanksi langgar psbb	1
Kecelakaan di MERR Surabaya, satu orang jadi korban.	kecelakaan merr surabaya satu orang korban	0
Perempuan paruh baya meninggal di rumah kontrakannya di Patmosusastro.	perempuan paruh baya meninggal rumah kontrakan patmosusastro	0

Tabel 3. Skenario Uji Coba Klasifikasi Teks Twitter Suara Surabaya

Skenario	Train Data	Test Data
1	April-Mei	Juni
	Mei-Juni	Juli
	Juni-Juli	Agustus
	Juli-Agustus	September
	Agustus-September	Oktober
2	Maret-April-Mei	Juni
	April-Mei-Juni	Juli
	Mei-Juni-Juli	Agustus
	Juni-Juli-Agustus	September
3	Juli-Agustus-September	Oktober
	Februari-Maret-April-Mei	Juni
	Maret-April-Mei-Juni	Juli
	April-Mei-Juni-Juli	Agustus
	Mei-Juni-Juli-Agustus	September
	Juni-Juli-Agustu-September	Oktober

“yang”, “dan”, “di”, “dari”, “adalah”, dan lain sebagainya.

4. *Stemming*. Proses ini adalah proses mengubah kata berimbuhan menjadi kata dasar. Penelitian ini menggunakan *library sastrawi* karena dataset yang digunakan merupakan dataset berbahasa Indonesia.

Contoh tahap *preprocessing* data dengan dataset yang diambil dari *timeline* akun Suara Surabaya ditunjukkan pada Tabel 1. Tabel tersebut menunjukkan teks asli yang didapatkan dari akun Twitter Suara Surabaya dan diproses melalui tahap *case folding*, *tokenizing*, *filtering*, dan *stemming*. Dataset bersih hasil *preprocessing* dibedakan menjadi data *train* dan data *test*. Dataset yang menjadi data *train* diberi label secara manual untuk digunakan pada proses klasifikasi. Data diberi label dua kelas, data yang berlabel “0” merupakan data non COVID-19 dan data yang berlabel “1” merupakan data COVID-19. Contoh data yang telah diberi label secara manual setelah *preprocessing* data ditunjukkan pada Tabel 2.

Dataset yang telah melalui tahap praproses dibagi ke dalam dua jenis yaitu data *train* dan data *test* yang masing-masing digunakan untuk melatih dan menguji model arsitektur klasifikasi yang dibangun. Eksperimen dilakukan dengan tiga skenario yang dibedakan oleh jumlah data *train*. Hal tersebut digunakan untuk membuktikan pengaruh jumlah data *train* pada proses klasifikasi teks. Skenario tersebut ditunjukkan pada Tabel 3. Pada skenario 1, data *train* menggunakan data

dua bulan, skenario 2 menggunakan data 3 bulan, dan skenario 3 menggunakan data 4 bulan. Sedangkan data *test* yang digunakan adalah data Bulan Juni, Juli, Agustus, September, dan Oktober.

### 2.3. Ekstraksi Fitur dengan *Word Embedding*

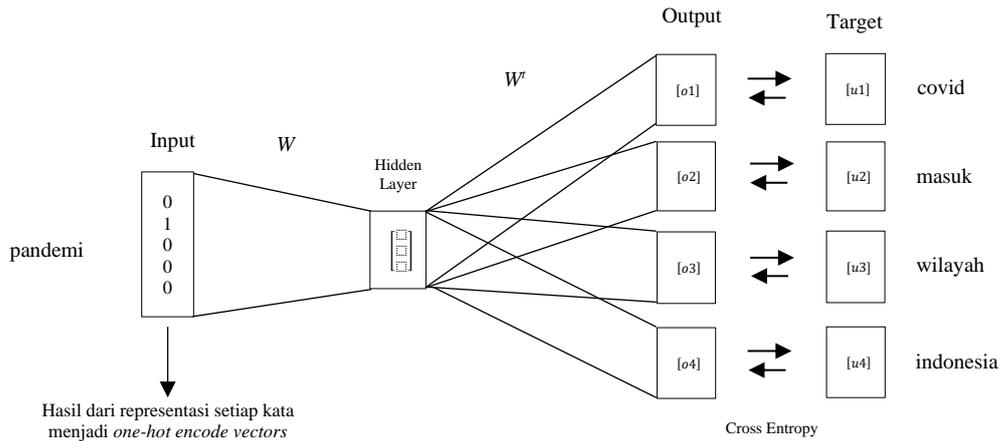
Metode *word embedding* yang digunakan untuk proses ekstraksi fitur adalah *Word2Vec* dan *fastText*. Penjelasan dari setiap metode tersebut dijelaskan pada poin-poin berikut.

#### 2.3.1 *Word2Vec*

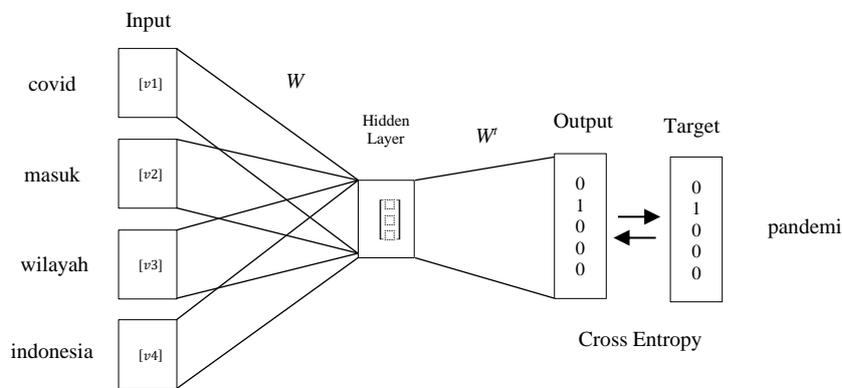
*Word2Vec* adalah salah satu metode *word embedding* pada *package gensim*. Beberapa kasus klasifikasi pada *Word2Vec* menunjukkan bahwa kata yang sama akan memiliki vektor yang sama [20]. *Word2Vec* dinilai efektif digunakan untuk ekstraksi fitur, namun memiliki kelemahan yaitu tidak dapat membedakan tingkat kepentingan suatu kata karena memperlakukan setiap kata secara sama, sehingga menjadi sulit untuk mengekstraksi kata yang memiliki nilai lebih tinggi daripada yang lainnya [21].

Penelitian ini memanfaatkan *pre-train* Bahasa Indonesia. Panjang dimensi yang digunakan adalah 300 dimensi yang berguna untuk menentukan seberapa banyak informasi yang didapatkan. Semakin kecil panjang dimensi maka akan semakin sedikit informasi yang didapatkan karena banyaknya informasi yang terbuang. Hal ini akan mempengaruhi proses klasifikasi. Setiap kata yang ada pada dataset diubah menjadi vektor sesuai dengan *pre-train* *Word2Vec*. Setiap kalimat ditambahkan fungsi *padding* untuk menyelaraskan panjang matriks input setiap kalimat. Hasil matriks digunakan sebagai *input* pada model *deep learning*.

*Word2Vec* menyediakan dua model yaitu *Continuous Bag-Of-Word (CBOW)* dan *Skip-gram*. *Input* pada model *Skip-gram* berupa *current word* yang berguna untuk memprediksi konteks disekitar. *Skip-gram* mempelajari probabilitas dari kata-kata didalam konteks. Pada model *skip-gram*, setiap kata harus diubah menjadi *one-hot encode vectors*. Cara kerja *skip-gram* diilustrasikan pada Gambar 2. Pada ilustrasi tersebut, kata “pandemi” berperan sebagai *input*. Sedangkan yang menjadi target adalah kata “covid” “masuk” “wilayah” “indonesia”. Ketika *output* telah didapatkan, *cross entropy* digunakan untuk memperhitungkan nilai *error*.



Gambar 2. Skip-Gram Model pada Teks Twitter COVID-19



Gambar 3. CBOW Model pada Teks Twitter COVID-19

Sedangkan model CBOW bekerja untuk memprediksi *current word* dengan *input* merupakan konteks disekitarnya. Konteks yang digunakan sama dengan model *skip-gram* yaitu menggunakan *one-hot-encode vectors*. Hal tersebut dikarenakan kinerja model CBOW merupakan kebalikan dari model *skip-gram*. Ilustrasi kinerja model CBOW ditunjukkan pada Gambar 3. Pada ilustrasi tersebut, kata “covid” “masuk” “wilayah” “indonesia” berperan sebagai *input*. Sedangkan target adalah kata “pandemi”. Pada model CBOW juga memperhitungkan nilai *error* dengan menggunakan *cross entropy*.

### 2.3.2 fastText

fastText adalah *toolkit* yang dikembangkan oleh tim peneliti Facebook [22]. fastText merupakan pengembangan dari model Word2Vec yang mempresentasikan setiap kata sebagai kantong karakter *n-gram* [23]. fastText memiliki kelebihan yaitu mampu meningkatkan representasi kata untuk bahasa yang kaya secara morfologis yang mengandung kata kerja dan kata benda dalam berbagai bentuk [24]. fastText dapat meningkatkan kinerja pada tugas-tugas sintaksis secara signifikan tetapi tidak banyak pada semantik [19].

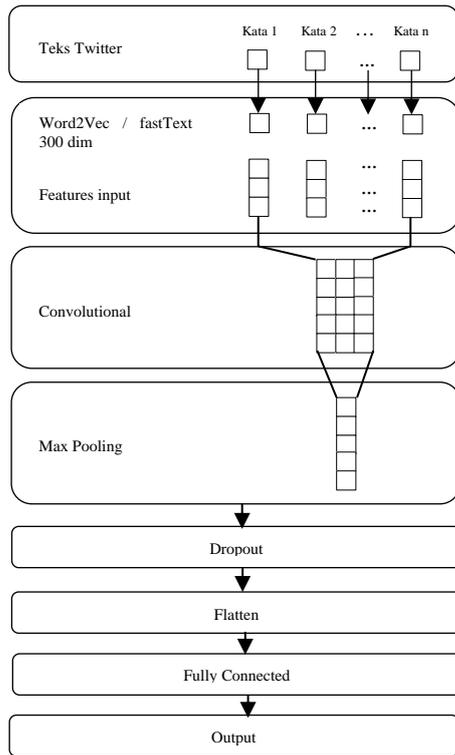
Model fastText yang digunakan adalah model *pre-train Bahasa Indonesia* dengan panjang dimensi sebesar 300 dimensi. *Pre-train word embedding* merupakan sekumpulan *word embedding* yang sudah pernah dilatih untuk domain yang besar pada *case study* tertentu sehingga dapat dimanfaatkan untuk menyelesaikan permasalahan yang sama. Pada dasarnya suatu teks tidak dapat diproses *deep learning* jika bentuknya bukan vektor. Sehingga *word embedding* berfungsi untuk merepresentasikan teks kedalam vektor agar dapat diproses *deep learning*.

## 2.4. Klasifikasi Teks dengan Deep Learning

Proses klasifikasi teks menerapkan variasi *deep learning* yaitu CNN, RNN, dan LSTM. Metode *deep learning* dikombinasikan dengan metode *word embedding* untuk mendapatkan pasangan kombinasi dengan hasil akurasi terbaik pada setiap skenario ujicoba. Setiap metode *word embedding* telah dijelaskan pada sub bab sebelumnya. Sedangkan untuk metode *deep learning* yang digunakan dijelaskan pada setiap poin berikut.

### 2.4.1 CNN (Convolutional Neural Network)

Arsitektur CNN yang digunakan pada penelitian ini ditunjukkan pada Gambar 4. Arsitektur tersebut terdiri



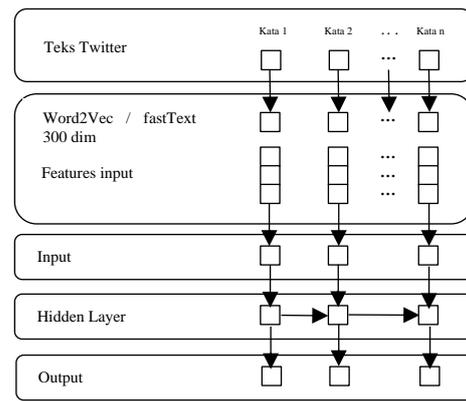
Gambar 4. Arsitektur CNN

dari *input layer*, *convolutional layer*, *max pooling layer* dan *fully connected layer*.

Teks Twitter hasil representasi vektor kata dengan menggunakan *word embedding* dimasukkan ke dalam *input layer*. Pada penelitian ini panjang maksimum sekuens merupakan panjang urutan dari  $X_{train}$ . Panjang sekuens menyesuaikan banyaknya data *train* yang digunakan. Pada penelitian ini banyaknya data *train* pada setiap skenario berbeda-beda. *Input* akan berupa matriks berukuran  $X_{train}$  dikalikan nilai dimensi *word embedding* yaitu 300 dimensi.

Selanjutnya pada *convolutional layer*, filter yang digunakan adalah 100. Terdapat parameter *kernel\_regularizer* yang menggunakan fungsi *regularizers*. *Regularizer* bekerja sebagai fungsi dari *output* bersih, dan sebagian besar digunakan untuk mengatur *hidden unit*. Pada layer ini menggunakan operasi *non-linear* dengan fungsi aktivasi ReLU yang biasa digunakan pada model *deep learning*. ReLU pada intinya hanya membuat pembatas pada bilangan nol, artinya apabila  $x \leq 0$  maka  $x = 0$  dan apabila  $x > 0$  maka  $x = x$ . Hasil dari proses ini adalah *feature map* yang berisi fitur penting dengan dimensi lebih rendah dari *hidden layer* pertama. Dimensi *feature map* akan terus mengecil untuk *convolutional layer* selanjutnya. Hal ini terjadi jika pada model CNN yang dibangun memiliki lebih dari satu *convolutional layer*.

Sedangkan, *max pooling layer* berfungsi untuk memperoleh informasi penting dari *feature map* pada



Gambar 5. Arsitektur RNN

*convolutional layer*. Pada layer ini terdapat beberapa parameter yang digunakan yaitu *Maxpooling2D*, *sequence\_length*, ukuran *filter* dan *strides*.

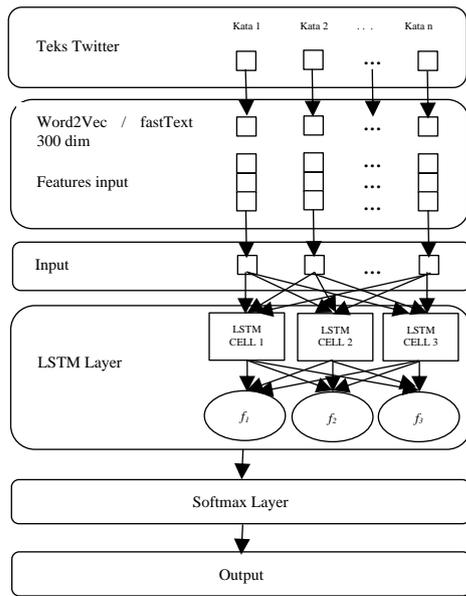
Pada *fully connected layer* terdapat parameter *reshape* yang berfungsi untuk merubah *feature map* menjadi sebuah vektor. Vektor tersebut terhubung dengan *output layer* untuk dilakukan klasifikasi. Pada layer ini parameter aktivasi menggunakan fungsi *sigmoid* karena klasifikasi merupakan klasifikasi biner atau klasifikasi dua kelas. Parameter lainnya yaitu “*loss*” yang menggunakan fungsi *categorical\_crossentropy*.

Terakhir yaitu *code* untuk melatih model CNN yang telah dibangun. Arsitektur CNN dilatih dengan parameter *batch\_size* sebesar 128 dan proses iterasi *epoch* sebesar 50 untuk mendapatkan akurasi yang baik. Fungsi optimasi yang digunakan adalah “*adam*”. Sedangkan untuk parameter *metric* yang berfungsi menyimpan hasil evaluasi berupa “*accuracy*”.

#### 2.4.2 RNN (Recurrent Neural Network)

Model Arsitektur RNN ditunjukkan pada Gambar 5. Arsitektur RNN memproses data *input* dari inputan satu sampai inputan ke- $n$  satu persatu secara sekuensial. Setiap inputan akan diteruskan dari *hidden layer* satu ke *hidden layer* selanjutnya sampai pada hasil *output*. RNN bekerja dengan proses *looping*. Misalnya untuk mencari *output* dari *input* berupa huruf “p”, “a”, “n”, “d”, “e”, “m” maka setiap *input* akan diproses pada *hidden layer* untuk memprediksi *output* yang seharusnya muncul setelah *input* “m”. RNN tidak membuang informasi sebelumnya namun menyimpan dan memproses pada *hidden layer*. Dalam contoh ini, prediksi *output* yang tepat ialah huruf “i”, sehingga akan menjadi kata yang sempurna yaitu “pandemi”.

Tiga layer utama pada RNN adalah *input layer*, *hidden layer*, dan *output layer*. Penelitian ini menggunakan RNN sebesar 100 *unit/cell* dengan *dropout* sebesar 0.4 yang artinya secara *random* tidak mengaktifkan 40% *neuron* pada *testing*. Panjang sekuens menyesuaikan panjang data *train*. Aktivasi yang dipakai adalah fungsi *sigmoid*. Pada fungsi kompilasi, terdapat parameter



Gambar 6. Arsitektur LSTM

$loss=categorical\_crossentropy$ . Sedangkan untuk *optimizer* menggunakan “adam”. Model RNN dilatih dengan *batch\_size* sebesar 128 dan panjang *epoch* 50.

#### 2.4.3 LSTM (Long-Short Term Memory)

Model arsitektur LSTM ditunjukkan pada Gambar 6. Arsitektur LSTM yang dibangun terdiri dari beberapa *layer* yaitu *input layer*, *LSTM layer*, dan *output layer*. Sebenarnya, arsitektur LSTM tidak jauh berbeda dengan RNN. Namun LSTM dapat mengkomputasikan *hidden state* yang dapat menyimpan *long-term dependencies*. Sehingga, LSTM dapat menyempurnakan kinerja RNN yang mengalami permasalahan *gradient exploding* atau *vanishing* pada RNN standar.

Teks Twitter diproses dengan metode *word embedding* yaitu Word2Vec maupun fastText dengan panjang dimensi 300. Hasil representasi vektor kata dimasukkan ke *input layer*. Panjang sekuens menyesuaikan dengan data *train* yang digunakan setiap kali ujicoba dilakukan sesuai skenario yang mengacu pada Tabel 2. Model yang digunakan pada arsitektur LSTM adalah model *sequential*. Model *sequential* ini berfungsi untuk tempat menumpuk lapisan LSTM. *Sequential* merupakan perintah pertama yang digunakan untuk menginisiasi bentuk dasar RNN. Layer LSTM menggunakan ukuran *cell/unit* sepanjang 100 *cell* dan *dropout* dengan *rate* 0.4 yang artinya akan menonaktifkan secara *random* 40% *neuron* selama fase *testing*. Fungsi optimasi menggunakan “adam” dan fungsi *loss* menggunakan *categorical\_crossentropy*. *Output* berupa hasil akurasi dari proses klasifikasi dua kelas yaitu COVID-19 dan non COVID-19. Arsitektur LSTM dilatih dengan panjang *epoch* sebesar 50 kali yang artinya data *training* diulang sebanyak 50 kali agar mendapatkan hasil akurasi maksimal. *batch\_size* menggunakan ukuran 128.

#### 2.5. Pemantauan Perhatian Publik

Pemantauan perhatian publik dilakukan melalui perhitungan prosentasi data hasil klasifikasi teks terkait COVID-19 dengan akurasi terbaik. Perhitungan dilakukan menggunakan rumus prosentase seperti yang ditunjukkan pada Rumus 1.

$$PP_t = \frac{\sum tweet\ COVID-19}{\sum semua\ tweet} \times 100\% \quad (1)$$

dengan  $PP_t$  merupakan perhatian publik pada waktu tertentu. Pada penelitian ini pemantauan perhatian publik dilakukan setiap bulan, dimulai dari Bulan Juni sampai Bulan Oktober. Data *tweet* terkait COVID-19 yang didapatkan melalui klasifikasi teks dibagi dengan semua data dan dikalikan 100%. Perhitungan tersebut menunjukkan prosentase *tweet* pada *timeline* akun Twitter Suara Surabaya terkait COVID-19 untuk bulan tertentu.

### 3. Hasil dan Pembahasan

#### 3.1. Klasifikasi Teks Twitter

Pemantauan perhatian publik melalui media sosial dilakukan dengan proses klasifikasi teks. Hal ini disebabkan karena Twitter memiliki topik yang bervariasi. Klasifikasi dilakukan dengan variasi beberapa metode *word embedding* dan *deep learning*. Metode *word embedding* yang digunakan adalah Word2Vec dan fastText. Sedangkan metode *deep learning* yang digunakan adalah CNN, RNN, dan LSTM. Hal ini dilakukan untuk menemukan kombinasi metode terbaik yang dapat diterapkan dalam klasifikasi teks. Eksperimen dilakukan dengan tiga skenario berbeda untuk membuktikan pengaruh jumlah data *train* yang berbeda jika digunakan untuk melatih data *test* yang sama. Skenario 1 menggunakan data *train* dengan data dua bulan. Skenario 2 menggunakan data *train* dengan data tiga bulan. Dan skenario 3 menggunakan data *train* dengan data empat bulan. Setiap data *test* dilatih dengan data *train* dari tiga skenario berbeda. Misalnya, data Bulan Juni dilatih dengan tiga skenario berbeda sesuai dengan data *train* yang telah ditunjukkan Tabel 3 pada bagian metode penelitian. Data *test* lainnya juga dilatih dengan menggunakan tiga skenario yang berbeda. Sehingga, hasil klasifikasi dapat menunjukkan pengaruh perbedaan jumlah data *train* jika digunakan untuk melatih data *test* yang sama.

Hasil klasifikasi ditunjukkan pada Tabel 4. Pada tabel tersebut, kolom “W2V” adalah hasil akurasi yang diperoleh metode *deep learning* yang dikombinasikan dengan metode *word embedding* Word2Vec. Sedangkan kolom “FT” adalah hasil akurasi yang diperoleh metode *deep learning* yang dikombinasikan dengan menggunakan metode *word embedding* fastText. Pada hasil klasifikasi, ketiga skenario menunjukkan bahwa kombinasi metode fastText dan LSTM meraih akurasi terbaik jika dibandingkan dengan hasil dari pasangan

Tabel 4. Akurasi Hasil Klasifikasi Teks Twitter setiap Skenario

Skenario	Train Data	Test Data	CNN		RNN		LSTM	
			W2V (%)	FT (%)	W2V (%)	FT (%)	W2V (%)	FT (%)
1	April-Mei	Juni	89.22	89.88	92.19	94.46	93.44	<b>95.15</b>
	Mei-Juni	Juli	88.37	88.26	91.50	94.26	93.07	<b>94.76</b>
	Juni-Juli	Agustus	86.70	87.07	91.06	93.71	91.86	<b>93.96</b>
	Juli-Agustus	September	87.81	88.37	91.27	94.06	92.05	<b>94.68</b>
	Agustus-September	Oktober	85.31	86.15	90.64	92.96	88.87	<b>93.63</b>
2	Maret-April-Mei	Juni	91.24	91.63	92.86	95.17	94.49	<b>95.93</b>
	April-Mei-Juni	Juli	91.06	90.99	92.92	94.82	94.41	<b>95.81</b>
	Mei-Juni-Juli	Agustus	90.03	90.66	92.60	94.73	94.09	<b>95.33</b>
	Juni-Juli-Agustus	September	89.22	90.02	92.41	94.67	94.05	<b>95.17</b>
	Juli-Agustus-September	Oktober	89.01	88.81	92.10	94.40	92.86	<b>95.14</b>
3	Februari-Maret-April-Mei	Juni	93.29	93.57	96.54	97.53	96.71	<b>97.86</b>
	Maret-April-Mei-Juni	Juli	92.04	92.69	95.10	96.27	95.80	<b>97.09</b>
	April-Mei-Juni-Juli	Agustus	91.46	92.31	94.52	95.82	94.65	<b>96.34</b>
	Mei-Juni-Juli-Agustus	September	91.43	92.23	93.20	95.34	94.51	<b>96.15</b>
	Juni-Juli-Agustus-September	Oktober	90.28	90.79	92.74	94.77	90.28	<b>95.69</b>

Word2Vec dan LSTM serta semua pasangan metode CNN dan RNN. Klasifikasi dengan hasil akurasi terbaik digunakan untuk pemantauan perhatian publik. Pada Tabel 4 kolom LSTM dengan menggunakan metode *word embedding* fastText akan dijadikan untuk pemantauan perhatian publik karena meraih hasil akurasi tertinggi dari semua ujicoba yang telah dilakukan.

Hasil akurasi menunjukkan bahwa fastText lebih unggul dari Word2Vec. fastText merupakan bentuk terbaru dari pengembangan metode Word2Vec [23][22]. fastText dapat melatih dataset dengan ukuran yang besar seperti pada penelitian ini yang memiliki dataset lebih dari 12000 data. fastText juga dapat memberikan representasi kata yang tidak muncul dalam *test* data. Representasi kata fastText yaitu dengan cara mengasumsikan sebuah kata dikomposisikan oleh *n-gram* karakter, panjang *n* dapat berubah. Untuk lebih *simple*-nya, fastText bekerja apabila suatu kata tidak muncul selama *training model*, maka kata tersebut dapat dipecah menjadi *n-gram* agar vektor dari kata dapat keluar.

Dari semua ujicoba, metode LSTM mengungguli semua metode *deep learning* yang digunakan yaitu RNN dan CNN. Hasil ujicoba ini sama dengan hasil penelitian [17] yang menunjukkan bahwa LSTM lebih baik jika dibandingkan dengan CNN dan RNN. CNN dinilai lebih baik jika diaplikasikan pada objek gambar. Sehingga jika dibandingkan dengan RNN dan LSTM untuk mengatasi objek teks, kinerja CNN tidak cukup baik. LSTM merupakan bentuk *modern* dari RNN. LSTM dapat mengatasi masalah *gradient exploding* atau *vanishing* pada RNN standar. LSTM juga dapat mempelajari dependensi jangka panjang dari urutan representasi tingkat yang lebih tinggi lebih baik daripada model lainnya [16]. Menurut [18][25] metode LSTM lebih unggul bila digunakan untuk model deret waktu

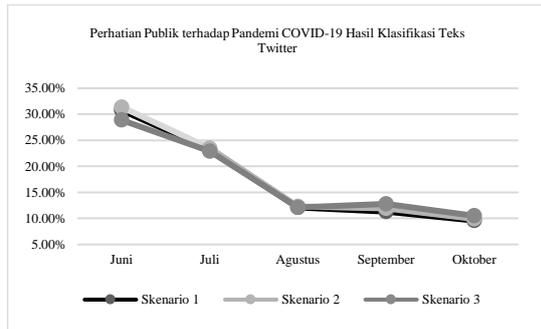
seperti pada teks Twitter. Sehingga, jika LSTM diaplikasikan untuk memproses dataset berupa teks Twitter maka LSTM dinilai sangat cocok.

Hasil akurasi tertinggi dari semua skenario diperoleh skenario 3. Skenario 3 merupakan skenario dengan jumlah data *train* terbanyak. Misalnya, hasil klasifikasi data Bulan Juni menunjukkan hasil akurasi skenario 3 yang diklasifikasi dengan menggunakan LSTM dan fastText sebesar 97.86%. Sedangkan, untuk skenario 1 dan 2 hanya mencapai akurasi sebesar 95.15% dan 95.93%. Hasil akurasi skenario 2 lebih tinggi jika dibandingkan dengan hasil akurasi dengan menggunakan skenario 1. Begitupun untuk hasil klasifikasi data Bulan Juli, Agustus, September, dan Oktober. Hal ini membuktikan bahwa semakin banyak data *train* yang digunakan maka semakin baik pula hasil yang didapatkan.

Namun, pada setiap skenario hasil akurasi yang ditunjukkan pada skenario yang memiliki unsur Bulan September menunjukkan hasil yang lebih rendah. Hal ini disebabkan karena dataset yang berhasil dikumpulkan pada Bulan September dari *timeline* akun Twitter Suara Surabaya hanya sekita 500 data. Berbeda dengan data yang berhasil didapatkan pada bulan lainnya yang mencapai lebih dari 1000 data. Hal tersebut membuktikan semakin banyak data maka kinerja *deep learning* akan semakin baik karena *deep learning* bekerja secara maksimal untuk dataset yang besar.

### 3.2. Pemantauan Perhatian Publik

Pemantauan perhatian publik dilakukan dengan menghitung data *tweet* terkait COVID-19 yang didapatkan melalui klasifikasi teks dengan hasil akurasi terbaik sesuai dengan Rumus 1 yang telah disebutkan pada bab metode penelitian. Prosentase perhatian publik terhadap pandemi COVID-19 melalui klasifikasi teks Twitter menunjukkan bahwa semua skenario



Gambar 7. Perhatian Publik terhadap Pandemi COVID-19 dari Hasil Klasifikasi Teks Twitter

menunjukkan hasil yang hampir sama, yaitu dari Bulan Juni sampai Bulan Oktober mengalami naik-turun seperti yang ditunjukkan pada Gambar 7. Hal tersebut dikarenakan terdapat *tweet* yang mengandung topik diluar bahasan COVID-19. Aktivitas bermedia sosial Twitter oleh masyarakat Kota Surabaya juga menjadi salah satu alasan tinggi rendahnya perhatian publik.

Hasil analisis secara manual pada dataset yang ada menunjukkan bahwa perhatian publik Kota Surabaya fokus pada beberapa topik seperti pandemi COVID-19, kondisi lalu lintas maupun kejadian lainnya yang terjadi pada rentan waktu antara Bulan Juni sampai Bulan Oktober. Hasil pemantauan perhatian publik menunjukkan bahwa perhatian publik terhadap COVID-19 tertinggi terjadi pada Bulan Juni. Jika dilihat pada Gambar 7, penurunan drastis terjadi pada Bulan Agustus. Hal ini dikarenakan pada Bulan Agustus *tweet* lebih banyak terkait perayaan kemerdekaan Republik Indonesia. Sehingga, *tweet* non COVID-19 lebih banyak dari *tweet* terkait COVID-19. Namun, perhatian publik terhadap pandemi COVID-19 mengalami sedikit peningkatan pada Bulan September. Dan mengalami penurunan sedikit pada Bulan Oktober. Hal tersebut dipengaruhi oleh beberapa faktor seperti banyaknya kejadian lain yang terjadi pada bulan tersebut sehingga mengakibatkan fokus publik lebih mengarah ke topik lainnya.

Pada awal pandemi COVID-19 dikonfirmasi masuk ke Indonesia khususnya Kota Surabaya, banyak perubahan perilaku publik yang merugikan satu sama lain seperti terjadinya *panic buying* pada bahan sembako dan lain sebagainya. Sehingga, hasil dari pemantauan perhatian publik setiap bulannya ini dapat digunakan oleh pihak terkait untuk membuat suatu kebijakan dan membangun strategi tanggap darurat yang tepat dalam menghadapi perubahan perilaku publik sebagai efek pandemi. Ujicoba menunjukkan bahwa hasil klasifikasi dengan akurasi tertinggi mencapai 97.86% dan terendah sebesar 93.63%. Sesuai dengan hasil klasifikasi, perhatian publik paling tinggi pada Bulan Juni, sedangkan penurunan perhatian publik terhadap pandemi COVID-19 yang paling drastis terdapat pada Bulan Agustus. Hal tersebut dikarenakan fokus publik terbesar Bulan

Agustus pada perayaan Kemerdekaan Republik Indonesia. Sehingga *tweet* mengenai pandemi COVID-19 lebih sedikit.

#### 4. Kesimpulan

Pemantauan perhatian publik terhadap pandemi COVID-19 dapat dilakukan melalui media sosial Twitter. Namun, teks Twitter memiliki topik yang bervariasi sehingga dibutuhkan proses klasifikasi agar dapat fokus pada satu topik tertentu yaitu pandemi COVID-19. Metode klasifikasi yang digunakan adalah variasi metode *word embedding* (Word2Vec dan fastText) dan variasi *deep learning* (CNN, RNN, LSTM). Percobaan dilakukan dengan tiga skenario yang dibedakan oleh jumlah data *train*. Hasil ujicoba menunjukkan bahwa semakin banyak data *train* maka hasil klasifikasi semakin baik. Arsitektur yang paling cocok adalah kombinasi fastText dan LSTM. Hal ini dikarenakan metode LSTM lebih unggul bila digunakan untuk model deret waktu yang dimiliki oleh teks Twitter. Sedangkan, metode *word embedding* yang paling cocok adalah fastText. Hal tersebut karena kinerja fastText yang menunjukkan apabila suatu kata tidak muncul selama *training model*, maka kata tersebut dapat dipecah menjadi *n-gram* agar vektor dari kata dapat keluar. Hasil pemantauan perhatian publik menunjukkan bahwa pada Bulan Juni perhatian publik terhadap COVID-19 mencapai prosentase tertinggi pada rentan waktu antara Bulan Juni sampai Bulan Oktober.

Pengerjaan selanjutnya adalah pemantauan perhatian publik terhadap pandemi COVID-19 dilakukan dengan lebih memfokuskan pada beberapa topik spesifik yang berkaitan dengan COVID-19 berdasarkan hasil klasifikasi dengan akurasi terbaik. Sehingga, proses klasifikasi tidak hanya untuk membagi data COVID-19 dan non COVID-19, namun juga klasifikasi untuk membagi data COVID-19 ke beberapa topik spesifik terkait COVID-19.

#### Daftar Rujukan

- [1] Yan, L., and Pedraza-Martinez, A.J., 2019. Social Media for Disaster Management: Operational Value of the Social Conversation. *Production and Operations Management Society*. 28 (10), pp. 2514–2532
- [2] Fahmi U., and Wibowo, C.P., 2017. Ruang Publik Online: Sebuah Dimensi Baru Dalam Proses Pembuatan Kebijakan? (Studi Kasus Penggunaan Twitter Di Indonesia). *International Seminar: Reconstructing Public Administration Reform to Build World Class Governmentn*. Jakarta, Indonesia, August 2017. Lembaga Administrasi Negara: Indonesia.
- [3] Duli, M.R., 2018. Analisis Konten Interaksi Pengguna Twitter pada Masa 100 Hari Pertama Pemerintahan Baru DKI Jakarta Menggunakan Text Mining Content Analysis on Twitter Users Interaction within First 100 Days of Jakarta's New Government by Using Text Mining. *Jurnal Pekommas*, 3m (2) pp.137-142.
- [4] Abdurrahman, M.S., 2018. Twitter dan Ruang Publik Pemerintahan Lokal yang Partisipatif (Telaah atas Komunikasi Politik Ridwan Kamil Melalui Twitter). *Jurnal Penelitian Komunikasi dan Pembangunan*, 15 (2), pp. 152-168.
- [5] Rathod, T., and Barot, M., 2018. Trend Analysis on Twitter for

- Predicting Public Opinion on Ongoing Events. *International Journal of Computer Applications*, 180 (26), pp. 13–17.
- [6] Vieweg, S., Hughes, A.L., Starbird, K., and Palen, L., 2010. Microblogging During Two Natural Hazards Events: What Twitter May Contribute to Situational Awareness. In: Association for Computing Machinery-SIGCHI, *Conference on Human Factors in Computing Systems - Proceedings*. Atlanta, Georgia, USA 10-15 April 2010, Association for Computing Machinery: United States.
- [7] Boukif, S., Biniz, M., El-Adnani, F., Cherrat, L., and El Moutaouakkil, A.E., 2018. Arabic Text Classification using Deep Learning Technics. *International Journal of Grid and Distributed Computing*, 11 (9), pp. 103–114.
- [8] Calix, R.A., Gupta, R., Gupta, M., and Jiang, K., 2017. Deep Gramulator: Improving Precision in the Classification of Personal Health-Experience Tweets with Deep Learning. In: IEEE Computer Society, *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. Kansas City, MO, USA 13-17 November 2017. IEEE: USA.
- [9] Kim, Y., 2014. Convolutional Neural Networks for Sentence Classification. In: Association for Computational Linguistics, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar 25-29 October 2014. Qatar: Doha.
- [10] Hughes, M., Li, I., Kotoulas, S., and Suzumura, T., 2017. Medical Text Classification using Convolutional Neural Networks. *Studies in Health Technology and Informatics*. April 2017.
- [11] Severyn, A., and Moschitti, A., 2015. UNITN: Training Deep Convolutional Neural Network for Twitter Sentiment Classification. In: Association for Computational Linguistics, *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Denver, Colorado, June 2015. Association for Computational Linguistics: Colorado.
- [12] Kim H., and Jeong, Y., 2019. Sentiment Classification Using Convolutional Neural Networks. *Applied Sciences MDPI*, 9 (11), pp. 1–14.
- [13] Cho K., et al., 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In: Association for Computational Linguistics, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar, October 2014. ACL: Qatar.
- [14] Du, C., and Huang, L., 2018. Text Classification Research with Attention-based Recurrent Neural Networks. *International Journal of Computers Communications & Control*, 13 (1), pp. 50–61.
- [15] Lai, S., Xu, L., Liu, K., and Zhao, J., 2015. Recurrent Convolutional Neural Networks for Text Classification. In: Association for the Advancement of Artificial Intelligence, *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. Austin, Texas USA, 25-30 January 2015. The AAAI Press: California.
- [16] Zhou, C., Sun, C., Liu, Z., and Lau, F.C.M., 2015. A C-LSTM Neural Network for Text Classification. Cornell University, *arXiv: 1511.08630*.
- [17] Tholusuri, A., Anumala, M., Malapolu, B., and Jaya Lakshmi, G., 2019. Sentiment Analysis using LSTM. *International Journal of Engineering and Advanced Technology*, 8 (6), pp. 1338–1340.
- [18] Rao, A., and Spasojevic, N., 2016. Actionable and Political Text Classification using Word Embeddings and LSTM. Cornell University, *arXiv: 1607.02501*.
- [19] Wang, B., Wang, A., Chen, F., Wang, Y., and Kuo, C.C.J., 2019. Evaluating Word Embedding Models: Methods and Experimental Results. *APSIPA Transactions on Signal and Information Processing*, 8, pp. 1–13.
- [20] Mandelbaum A., and Shalev, A., 2016. Word Embeddings and Their Use In Sentence Classification Tasks, *arXiv:1610.08229*, pp. 1–16.
- [21] Lilleberg, J., Zhu, Y., and Zhang, Y., 2015. Support Vector Machines and Word2vec for Text Classification with Semantic Features. *Proceedings of 2015 IEEE 14th International Conference on Cognitive Informatics and Cognitive Computing, ICCI\*CC 2015*. Beijing, China, 6-8 July 2015. IEEE: United States.
- [22] Kuyumcu, B., Aksakalli, C., and Delil, S., 2019. An automated new approach in fast text classification (fastText): A case study for Turkish text classification without pre-processing. In: Association for Computing Machinery, *ICSEB 2019: Proceedings of the 2019 3rd International Conference on Software and e-Business*. Tokyo, Japan, December 2019. Association for Computing Machinery: United States.
- [23] Khattak, F.K., Jebblee, S., Pou-Prom, C., Abdalla, M., Meaney, C., and Rudzicz, F., 2019. A survey of word embeddings for clinical text. *Journal of Biomedical Informatics X*, 4, 100057.
- [24] Dabiri, S. and Heaslip, K., 2018. Developing a Twitter-Based Traffic Event Detection Model using Deep Learning Architectures. *Expert Systems with Applications*, 118, pp. 425–439.
- [25] Zhang, Z., He, Q., Gao, J., and Ni, M., 2017. A Deep Learning Approach for Detecting Traffic Accidents from Social Media Data. *Transportation Research Part C: Emerging Technologies*, 86, pp. 580–596.