Computatio: Journal of Computer Science and Information Systems, 3/2 (2019), 108-115

SISTEM PENDETEKSI KALIMAT UMPATAN DI MEDIA SOSIAL DENGAN MODEL NEURAL NETWORK

Sahrul[1], Ahmad Fauzan Rahman[2], Muhammad Dzaky Normansyah[3], Ade Irawan[4]
[1,2,3,4] Program Studi Ilmu Komputer, Fakultas Sains dan Komputer, Universitas Pertamina,
Jln. Teuku Nyak Arief, RT.7/RW.8, Jakarta, 12220, Indonesia
E-mail: [1]h2sahrul@gmail.com, [2]ahmadfauzan1198@gmail.com, [3]dzakynormansyah@gmail.com, [4]adeirawan@universitaspertamina.ac.id

Abstrak

Pemerintah dan penyedia layanan media sosial di Indonesia berusaha keras untuk mengatasi maraknya konten negatif di media sosial. Konten negatif yang sering ditemui diantaranya isu suku, agama, ras, dan antargolongan (SARA), cyberbullying, serta body shamming, yang biasanya muncul disertai kalimat-kalimat umpatan. Hal tersebut menjadi sulit untuk diatasi karena jumlah pengguna internet di Indonesia yang sangat besar, sehingga perlu adanya sebuah sistem yang dapat mendeteksinya secara otomatis. Penelitian ini mengusulkan sistem dengan model Neural Network untuk deteksi konten negatif di media sosial dengan cara mempertimbangkan konteks kalimat atau frasa, tidak hanya kata-per-kata. Ada dua model NN yang dianalisis di penelitian ini, yaitu Artificial Neural Network (ANN) dan Recurrent Neural Network (RNN). Model RNN menunjukkan performa yang lebih baik dibandingkan dengan model ANN dengan akurasi training, validasi, dan test masing-masing adalah 94%, 84%, dan 84%.

Kata kunci*—Sistem pendeteksi kalimat umpatan, media sosial, neural network*

Abstract

Governments and social media providers put high effort to tackle massive negative contents in social media. Those contents are mostly containing religion, race, and inter-group issues, cyberbullying, and also body shamming, which usually appears together with offensive languages. It becomes difficult to overcome because of a large number of internet users in Indonesia. Hence, we need a system that can automatically detect the negative contents. This paper utilizes Neural Network (NN) models for not only classifying the words as (non)offensive words but also considering the structure of the sentence to get its context. There are two NN models analyzed in this paper: Artificial Neural Network (ANN) and Recurrent Neural Network (RNN). The computer simulation results show that the RNN has better performances than the ANN with the accuracy of training, validation, and testing 94%, 84%, and 84%, respectively.

Keywords—*Offensive language detection, social media, neural network*

1. PENDAHULUAN

Aktifitas di media sosial telah menjadi bagian dari kehidupan sehari-hari oleh sebagian besar masyarakat Indonesia pada era digital saat ini. Berdasarkan data yang dirilis oleh Hootsuite pada bulan Januari 2019, pengguna media sosial di Indonesia mencapai 150 juta orang dari total

jumlah populasi sebanyak 268.2 juta [1]. Berdasarkan data tersebut, hampir dapat dipastikan bahwa sekitar 56% penduduk Indonesia merupakan pengguna media sosial.

Teknologi internet khususnya media sosial memudahkan setiap kalangan untuk memperoleh informasi dan menyampaikan aspirasi. Namun, kemudahan tersebut tidak serta merta memberi dampak baik kepada seluruh warganet. Interaksi di media sosial dapat memicu keributan yang melibatkan banyak warganet. Hal ini disebabkan karena adanya anggapan bahwa media sosial merupakan wadah kebebasan berpendapat, seperti "*ini akun saya, terserah saya mau ngomong apa!*". Isu-isu yang seringkali menyebabkan keributan di media sosial diantaranya adalah suku, agama, ras, dan antar golongan (SARA), *cyberbullying*, dan *body shaming* yang biasanya muncul bersama kalimat atau frasa umpatan.

Bukti pemerintah Indonesia serius dalam menangani masalah penggunaan internet di Indonesia adalah adanya pemberlakuan undang-undang (UU) terkait informasi dan transaksi elektronik (ITE), yaitu UU No 19 Tahun 2016 [2]. Namun, masih perlu adanya sinergi antara pemerintah, warganet, maupun pihak penyedia layanan media sosial seperti Twitter, Facebook, Youtube, dan lain sebagainya, untuk menanggulangi maraknya kalimat atau frasa umpatan di media sosial.

Penyedia layanan media sosial berusaha menangani isu yang sama di hampir seluruh negara tempat mereka beroperasional [3]. Upaya yang mereka lakukan sampai saat ini hanya sebatas pada penyediaan saluran pelaporan untuk konten yang merupakan umpatan atau konten lain yang tidak sesuai kebijakan komunitas dan aturan pemerintah Indonesia [4]. Beberapa media sosial seperti Facebook dan Instagram memiliki fitur pendeteksi kata tertentu secara otomatis [5,6]. Namun, metode yang digunakan hanya dapat mendeteksi kata-per-kata tanpa mengetahui konteks kalimat, sehingga metode ini kurang efektif.

Penelitian mengenai pendeteksi umpatan telah dilakukan oleh beberapa peneliti. Metode yang biasa digunakan diantaranya adalah *Naïve Bayes*, *Support Vector Machine*, *Semantic Method* [7,8], dan *Obfuscation Methods* [9]. Namun, semua metode tersebut juga tidak mempertimbangkan konteks kalimat atau frasa yang belum dimengerti oleh komputer dengan baik. Selain itu, metode *Obfuscation Methods* hanya dapat bekerja pada level kata dan belum dikembangkan untuk bahasa Indonesia.

Penelitian ini adalah lanjutan dari [10] yang memanfaatkan teknik untuk menganalisis korelasi pola antar kata dengan model *Artificial Neural Network* (ANN). Misalnya, kalimat "*muka kamu seperti babi hutan!*", digolongkan sebagai umpatan. Namun, ketika kata "*babi*" digunakan dalam kalimat dengan struktur kata yang berbeda, seperti "*babi kamu lucu sekali!*", maka kalimat ini tidak akan dideteksi sebagai umpatan. Model *Recurrent Neural Network* (RNN) yang diusulkan dalam makalah ini memiliki performa yang lebih baik dibandingkan ANN di [10] untuk jumlah data yang lebih banyak dengan cakupan yang luas.

2. METODE PENELITIAN

Penelitian dilakukan dengan dua tahapan, yaitu dimulai dari persiapan data, kemudian analisis dan perancangan model.

2.1 Tahap Persiapan Data

Ada tiga jenis kegiatan yang dilakukan di tahap persiapan data, yaitu pengumpulan data, pengkondisian data, dan pelabelan data. Data yang dikumpulkan di penelitian ini adalah berupa

tweet pengguna Twitter berdasarkan *track-keywords* tertentu yang sudah didefinisikan sebagai kata yang sering diasosiasikan sebagai umpatan. Data tersebut diperoleh dengan menggunakan API Twitter dan diambil secara *stream* dengan pustaka Tweepy pada bahasa pemrograma Python. Data yang berhasil dikumpukan selama tiga hari adalah sebanyak 88,009 data. Data mentah yang merupakan luaran pada kegiatan ini disimpan untuk pengolahan selanjutnya.

Pengkondisian data dilakukan sebagai bagian pra-pemrosesan data teks sebelum diolah dengan model Neural Network (NN). Pra-pemrosesan data yang dilakukan ada tiga, yaitu (1) penghilangan *noise*, misalnya kata *tag html*, *tanda baca*, *angka*, (2) penghilangan *stopwords,* misalnya kata *ada, di, yang*, dan (3) pengembalian setiap kata ke kata dasarnya (*stemming*), misalnya kata *mencari* menjadi *cari*, kata *kesakitan* menjadi *sakit.* Tidak ada normalisasi data pada kata-kata percakapan sehari-hari ke bahasa formal untuk menjaga agar konteks setiap kalimat atau frasa tidak berubah. Luaran pada tahap ini adalah data yang sudah dibersihkan.

Pelabelan data perlu dilakukan karena algoritma yang digunakan merupakan jenis *supervised learning*. Pelabelan data dilakukan berdasarkan preferensi pribadi untuk menilai apakah *tweet* tertentu merupakan umpatan (dengan label 1) atau bukan (dengan label 0). *Over sampling* perlu dilakukan jika jumlah data antara umpatan dan bukan umpatan tidak seimbang. Teknik *over sampling* yang dipakai pada penelitian ini adalah *synthetic minority over-sampling technique* (SMOTE) [11].

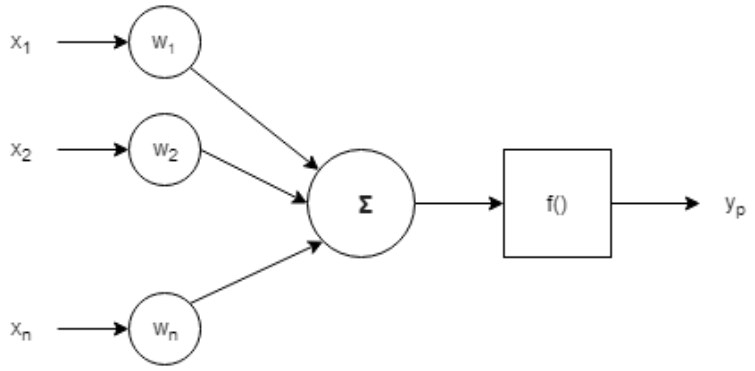
Setiap kata diindeks berdasarkan jumlah kemunculan pada keseluruhan data dengan hanya mengambil satu kata dari kata-kata yang sama. Kata dengan jumlah kemunculan yang paling banyak ditempatkan di indeks awal, kemudian diikuti oleh kata-kata berikutnya yang terurut hingga kata yang jumlah kemunculannya paling sedikit. Setiap data direpresentasikan sesuai urutan kata pada kalimat atau frasa tersebut.

2.2 Tahap Analisis dan Perancangan Model

Tahap analisis dan perancangan model dilakukan setelah data disaring dan dibersihkan di tahapan persiapan data. Terdapat dua jenis model NN yang digunakan di penelitian ini, yaitu ANN dan RNN.

2.2.1 Artificial Neural Network (ANN)

Model ANN memiliki arsitektur jaringan yang lebih sederhana, hanya terdiri dari tiga elemen dasar, yaitu *neuron*s, lapisan (*layers*), dan fungsi aktivasi, serta dua proses utama, yaitu *forward propagate* dan *backward propagate*. Arsitektur model ANN ditunjukkan pada Gambar 1.



Gambar 1 Arsitektur Model ANN

Fungsi aktivasi yang digunakan pada proses *forward propagate* di penelitian ini adalah fungsi sigmoid. Persamaan fungsi aktivasi sigmoid ditunjukkan oleh persamaan (1) sebagai berikut:

$$\sigma(z) = \frac{1}{1+e^{-z}} \text{ dan } 0 \leq \sigma(z) \leq 1, \quad (1)$$

dengan

$$z = Wx + b. \quad (2)$$

W, x, dan b masing-masing adalah nilai faktor pemberat, vektor kata, dan nilai bias. Sedangkan nilai luaran diperoleh dengan persamaan (3) sebagai berikut:

$$y = \sigma(Wx + b) \text{ dan } 0 \leq y \leq 1, \quad (3)$$

dengan y dan x berturut-turut adalah nilai luaran dan masukan disetiap lapisan. Fungsi yang digunakan untuk menghitung *loss* adalah *cross-entropy loss*, seperti yang ditunjukkan oleh persamaan (4) berikut ini:

$$L(y_a, y_p) = -\frac{1}{m}\sum_{i=0}^{m}\left[{y_p}^{(i)} \log\left({y_a}^{(i)}\right) + \left(1 - {y_p}^{(i)}\right) \log\left(1 - {y_a}^{(i)}\right)\right], \quad (4)$$

dengan L, m, y_p dan y_a masing-masing merupakan nilai *error*, jumlah seluruh data, nilai prediksi, dan nilai aktual untuk data ke i. Pada proses *backward propagate*, fungsi yang digunakan adalah fungsi turunan dari fungsi aktivasi terhadap faktor pemberat. *Backward propagate* dimulai dari lapisan paling akhir sampai lapisan paling awal dengan proses pembaharuan nilai faktor pemberat menggunakan teknik *gradient descent*. Persamaan *gradient descent* dan turunan fungsi *loss* terhadap faktor pemberat ditunjukkan oleh persamaan (5) dan (6) berikut ini:

$$W' = W - \propto \frac{dL}{dW} \quad (5)$$

dengan

$$\frac{dL}{dW} = (y - \sigma(z)) \cdot x \quad (6)$$

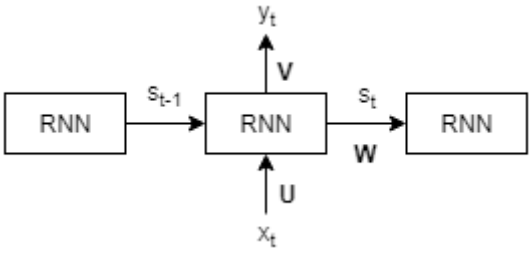
Model ANN pada penelitian ini digunakan untuk memprediksi apakah sebuah kalimat atau frasa merupakan umpatan atau bukan umpatan. Jumlah lapisan yang digunakan adalah sebanyak 4 lapis dengan jumlah *neurons* masing-masing, dari lapisan masukan hingga lapisan terakhir, adalah 20, 1024, 1024, dan 1. Dengan demikian, setiap data akan diambil per 20 kata. Nilai *hyperparameters* pada model ANN yang diperoleh dari proses *parameter tuning* dapat dilihat di Tabel 1.

Tabel 1 Nilai *Hyperparameter* Model ANN

Parameter	Nilai
Jumlah lapisan	4
Jumlah *neurons*	(20, 1024, 1024, 1)
Learning rate	0.001
Jumlah *epochs*	100 (*Early stopping*)
Batch size	64
Fungsi aktivasi	Sigmoid
Fungsi *loss*	*Cross-entropy loss*

2.2.2 Recurrent Neural Network (RNN)

Model kedua yang digunakan pada penelitian ini adalah model RNN jenis *long short term memory* (LSTM). Model ini merupakan jenis NN yang cocok untuk data yang memiliki urutan, seperti data *time series*, teks, dan DNA, karena setiap datanya saling terhubung satu sama lain. Misalnya, pada data *time series*, luaran pada waktu *t* tidak hanya dipengaruhi oleh masukan pada waktu t saja, tetapi juga dipengaruhi oleh masukan pada $t-1$, $t-2$, $t-3$, dan seterusnya. Arsitektur model RNN ditunjukkan pada Gambar 2.



Gambar 2 Model RNN pada Waktu t

Sama halnya dengan model ANN, model RNN juga terdiri dari proses *forward propagate* dan *backward propagate*. Proses *forward propagate* sesuai dengan persamaan (6) berikut ini:

$$s_t = \sigma(W_{ss}x_{t-1} + W_{sx}s_t + b_s), \tag{6}$$

dengan

$$y_t = W_{yx}s_t + b_y \tag{7}$$

Fungsi *loss*-nya didefinisikans sesuai dengan persamaan (8) berikut ini:

$$L(y_a, y_p) = -\frac{1}{T_x}\sum_{t=0}^{T_x} {y_a}^{(t)} \log\left({y_p}^{(t)}\right), \tag{8}$$

dengan T_x merupakan panjang kata pada data masukan x.

Proses *backward propagate* menggunakan teknik *gradient descent* dengan tiga parameter yang akan diperbaharui sampai ditemukan nilai *loss* yang paling optimal, yaitu *U*, *V*, dan *W*. Nilai *hyperparameters* pada model RNN yang diperoleh dari proses *parameter tuning* dapat dilihat pada Tabel 2.

Tabel 2 Nilai *Hyperparameter* Model RNN

Parameter	**Nilai**
Jumlah lapisan	4
Jumlah *neurons*	(64, 128, 128, 1)
Dropout rate	(0, 0, 0.5, 0)
Learning rate	0.001
Jumlah *epochs*	100 (*Early stopping*)
Batch size	64
Fungsi aktivasi	Sigmoid
Fungsi *loss*	*Cross-entropy loss*

Lapisan pertama artsitektur model RNN menggunakan *Keras Embedding Layer* [12] dengan ukuran 64 yang mengubah bilangan bulat positif berdasarkan indeks menjadi vektor yang berukuran tetap. Misal pada frasa "*anjing kamu lucu sekali*" dan "*muka kamu seperti anjing*" dapat ditulis dengan [0, 1, 2, 3] dan [4, 1, 5, 0] berdasarkan tahap persiapan data pada bagian 2.1.

Contoh proses pembentukan vektor *embedding* pada proses *training* dengan ukuran (6, 2) dapat dilihat di Tabel 3.

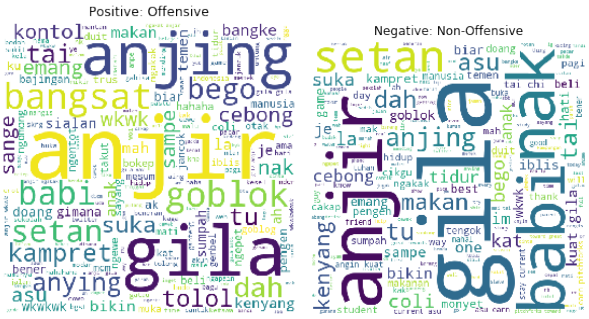
Tabel 3 Proses Pembentukan Vektor *Embedding*

Indeks	***Embedding Weights***
0	[1.2, 3.1]
1	[0.1, 4.2]
2	[1.0, 3.1]
3	[0.3, 2.1]
4	[2.2, 1.4]
5	[0.7, 1.7]
6	[4.1, 2.0]

Nilai *embedding weights* akan terus diperbaharui selama proses *training*. Jadi, frasa kedua dapat direpresentasikan sebagai [[2.2, 1.4], [0.1, 4.2], [0.7, 1.7], [1.2, 3.1]].

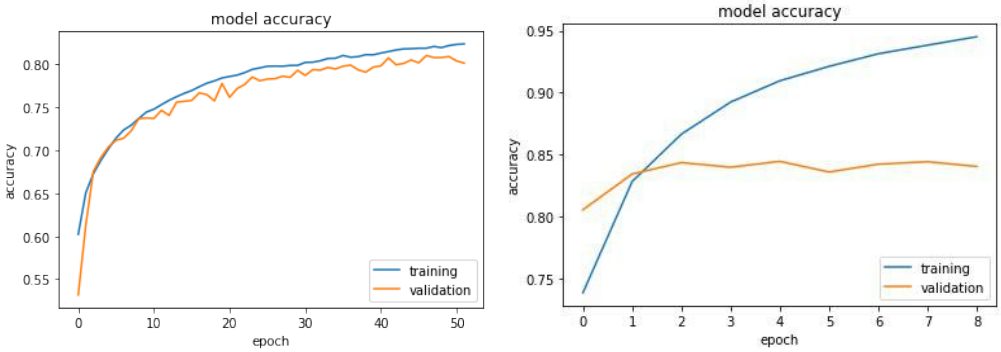
3. HASIL DAN PEMBAHASAN

Gambar 3 menggambarkan kata-kata yang sering muncul atau tidak di kalimat atau frasa umpatan maupun bukan umpatan dengan ukuran hurufnya. Semakin besar ukuran hurufnya, semakin dominan kata tersebut. Dari gambar tersebut terlihat bahwa kalimat atau frasa umpatan maupun bukan umpatan terdiri dari kata-kata yang hampir sama seperti "*anjir*", "*gila*", "*anjing*", dan sebagainya.



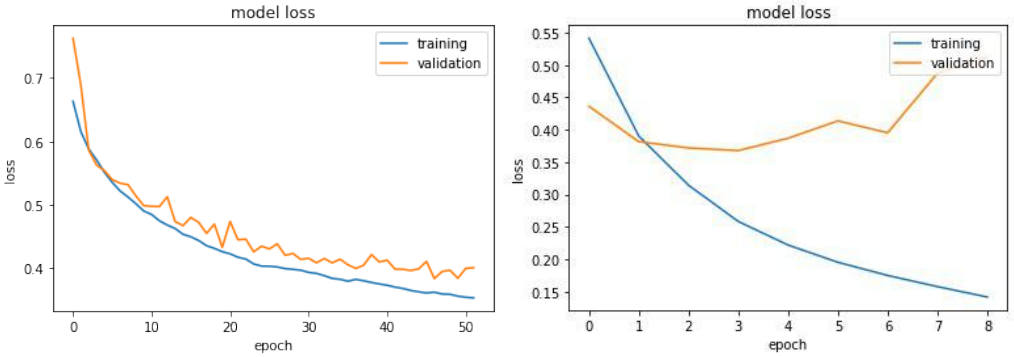
Gambar 3 Kata yang Paling Sering Muncul di Tiap Kategori

Gambar 4 menunjukkan hasil simulasi pengukuran akurasi dari model ANN dan RNN yang diusulkan, yaitu nilai akurasi model RNN cenderung lebih baik dibadingkan dengan model ANN. Namun, model RNN terlihat mengalami *overfit* terhadap *training set* yang ditunjukkan dengan selisih yang cukup besar antara nilai akurasi *training* dengan akurasi pada saat validasi.



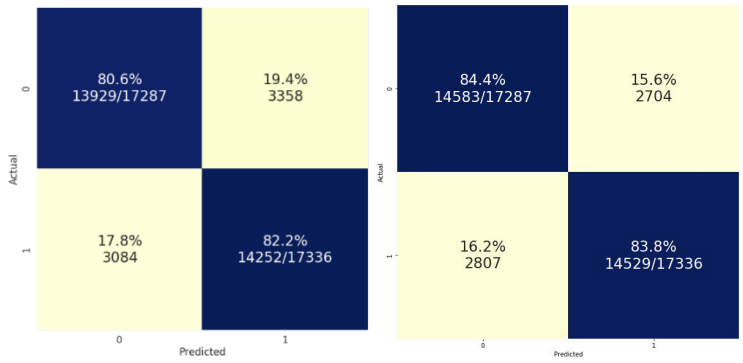
Gambar 4 Perubahan Akurasi Selama Proses *Training*: ANN (kiri) dan RNN (kanan)

Perbandingan nilai *loss* pada Gambar 5 menunjukkan bahwa model RNN lebih baik dibandingkan dengan model ANN. Model RNN juga mengalami konvergensi nilai *loss* yang lebih cepat dengan hanya membutuhkan 8 *epoch*. Namun, model RNN tetap mengalami *overfit* terhadap *training set*.



Gambar 5 Perubahan Loss Selama Proses Training: ANN (kiri) dan RNN (kanan)

Hasil pengujian model pada *unseen data* dapat dilihat pada *confusion matrix* di Gambar 6. Gambar tersebut menunjukkan bahwa model RNN lebih baik dibandingkan dengan model ANN. Model RNN dapat memprediksi dengan benar 83.8% dari keseluruhan *test set* yang dilabeli sebagai umpatan dan 84.4% yang bukan umpatan. Sedangkan model ANN hanya dapat memprediksi dengan benar 82.2% yang dilabeli sebagai umpatan dan 80.6% yang bukan umpatan.



Gambar 6 *Confusion Matrix* pada Performa Model: ANN (kiri) dan RNN (kanan)

4. KESIMPULAN

Model RNN menunjukkan performa prediksi kata umpatan/bukan umpatan yang lebih baik dibandingkan dengan model ANN [10] untuk jumlah data yang lebih banyak dengan cakupan yang luas. Ketidakteraturan cara penulisan di Twitter seperti *grammar*, singkatan, variasi bahasa gaul, dan adanya salah ketik merupakan sebuah tantangan tersendiri untuk memperoleh performa model yang lebih optimal. Oleh karena itu, pengembangan selanjutnya dapat dilakukan dengan pra-pemrosesan data yang lebih baik serta penggunaan arsitektur model RNN yang lebih tahan terhadap *overfitting*.

DAFTAR PUSTAKA

[1] Riyanto, A. D., 2019, (Hootsuite) Indonesian Digital Report 2019, *https://andi.link/hootsuite-we-are-social-indonesian-digital-report-2019/*, diakses tanggal 6 Juli 2019.

[2] Juniman, P. T., 2016, Revisi UU ITE Mulai Berlaku 28 November Esok, *https://www.cnnindonesia.com/teknologi/20161126172041-185-175520/revisi-uu-itemulai-berlaku-28-november-esok/*, diakses tanggal 31 Desember 2018.

[3] Breland, A., 2017, Social media fights back against fake news, *https://thehill.com/policy/technology/335370-social-media-platforms-take-steps-toprotect-users-from-fake-news/*, diakses tanggal 31 Desember 2018.

[4] Dina, S., 2017, Banyak Konten Negatif, Twitter Minta Pengguna Pakai Fitur Report, *https://kominfo.go.id/content/detail/11846/banyak-konten-negatif-twitter-mintapengguna-pakai-fitur-report/0/sorotan_media/*, diakses tanggal 31 Desember 2018.

[5] Agrawal, H., 2018, Facebook Page Feature: Block Words and Profanity Blocklist, *https://www.shoutmeloud.com/facebook-page-profanity-blocklist-spam-words.html*, diakses tanggal 4 Januari 2019.

[6] Carman, A., 2016, Instagram is now letting everyone filter abusive words out of their comments, *https://www.theverge.com/2016/9/12/12887514/instagram-comments-abusive-wordsfilter-section*, diakses tanggal 4 Januari 2019.

[7] Vandermissen, B., 2012, Automated detection of offensive language behavior on social media networking sites, *Disertasi*, Faculteit Ingenieurswetenschappen en Architectuur, Universiteit Gent, Gent, Belgia.

[8] Sood, S. O., Antin, J., dan Churchill, E. F., 2012, Profanity Use in Online Communities, *Proceeding of the SIGCHI Conference on Human Factor in Computing Systems*, Austin, Texas, 5-10 Mei 2012.

[9] Labreiro, G., dan Oliveira, E., 2014, What We Can Learn from Looking at Profanity, *Proceeding on 11th International Conference PROPOR 2014*, Sao Carlos, Brasil, 6-8 Oktober 2014.

[10] Susanty, M., Sahrul, Rahman, A.F., Normansyah, M.D., Irawan, A., 2019, Offensive Language Detection using Artificial Neural Network, *International Conference of Artificial Intelligence and Information Technology (ICAIIT)*, Yogyakarta, Indonesia, Maret 2019.

[11] N. V. Chawla, K. W. Bowyer, L. O.Hall, W. P. Kegelmeyer, 2002, SMOTE: synthetic minority over-sampling technique, *Journal of artificial intelligence research*, 16, 321-357.

[12] C. Francois, et al., 2015, Keras, *https://keras.io/layers/embeddings/*, diakses tanggal 15 September 2019.