

PREDIKSI KATA KASAR BERBAHASA INDONESIA MENGGUNAKAN MACHINE LEARNING BERBASIS MOBILE INFRASTRUCTURE

Puri Sulistiyawati, Farrikh Alzami^{*}), Dwi Puji Prabowo, Ricardus Anggi Pramunendar, Rama Aria Megantara, Nuanza Purinsyira, dan Enrico Irawan

Faculty of Computer Science, Universitas Dian Nuswantoro, Semarang, Indonesia
Jl. Imam Bonjol 207 Semarang

^{*}E-mail: alzami@dsn.dinus.ac.id

Abstrak

Komentar kasar dan menyinggung dapat dijelaskan sebagai komunikasi yang bertujuan membuat satu atau lebih individu untuk berlaku marah. Oleh karena itu, diperlukan sebuah pendekatan untuk mengetahui apakah kalimat komentar yang akan ditulis merupakan komentar kasar atau bukan. Kemudian, melihat dari keseharian penduduk Indonesia yang tidak terlepas dari smartphone, memberikan peluang untuk memberikan edukasi kepada pengguna smartphone bagaimana mendeteksi komentar kasar. Maka, pengembangan aplikasi berbasis android perlu dikembangkan. Penelitian ini bertujuan mengembangkan aplikasi mobile sentimen analisis deteksi kata kasar menggunakan TF-IDF sebagai fitur ekstraksi dan Naïve Bayes berbasis android flutter yang intuitif. Hasil pengujian menunjukkan nilai training akurasi 98%, training recall 98%, training precision 99%, testing accuracy 84.26%, testing recall 86.81%, dan testing precision 83.15%. Dengan demikian, aplikasi ini telah dapat memberikan prediksi yang baik sesuai harapan.

Kata kunci: hate speech detection, android

Abstract

Harsh and offensive comments can be described as communications aimed at making one or more individuals angry. Therefore, an approach is needed to find out whether the comment sentence to be written is a rude comment or not. Then, looking at the daily lives of Indonesians who cannot be separated from smartphones, it provides an opportunity to provide education to smartphone users how to detect rude comments. Therefore, the development of android-based applications needs to be developed. From this explanation, we developed a sentiment analysis mobile application for hate speech detection using TF-IDF as feature extraction and Naïve Bayes based on intuitive Android Flutter with 98% accuracy values, 98% recall, and 99% precision in training measurement. Then, 84.26% accuracy values, 86.81% recall, and 83.15% precision in testing measurement. As a result, the application can provide acceptable prediction.

Keywords: hate speech detection, android

1. Pendahuluan

Saat ini, dengan kemajuan world wide web, setiap orang dapat berinteraksi dengan orang lain menggunakan media sosial, blog, dan forum online. Karena sangat sedikit moderasi dan setiap orang dapat berkomentar dengan bebas tanpa menyebut nama, komentar yang tidak menyenangkan (ofensif) tidak dapat dihindari dalam percakapan, mengakibatkan banyak orang mengalami trauma mental dan kasus terburuk adalah bunuh diri [1]. Komentar yang menyinggung dapat digambarkan sebagai komunikasi apa pun termasuk perkataan yang mendorong kebencian, pelecehan, penindasan, dan sumpah serapah yang membuat satu atau lebih orang menjadi marah [2]. Dengan pengguna aktif yang besar di world wide web, terutama di media sosial, hampir tidak mungkin untuk mendeteksi komentar yang menyinggung secara manual karena volume teks yang dihasilkan (komentar) yang tinggi

[2]. Oleh karena itu, deteksi dan klasifikasi ujaran kebencian yang kuat sangat dibutuhkan. Analisis sentimen bertujuan mengidentifikasi sikap atau suasana hati orang melalui pemrosesan bahasa alami, analisis teks, dan linguistik komputasi. Dalam beberapa tahun terakhir, pembelajaran mesin telah menjadi alat yang sangat kuat untuk mengklasifikasikan sentimen. Dari literatur, analisis sentiment sudah dipakai untuk deteksi cyberbullying[3], deteksi bahasa kasar [4], ulasan film [5] dan identifikasi cyberhate [6].

Pada tahun 2021, terdapat lebih dari 170 juta orang di Indonesia yang menggunakan smartphone untuk menjelajahi media sosial [7]. Jadi, pada titik ini, dengan banyaknya pengguna, sangat sulit untuk mengidentifikasi komentar menyinggung yang dihasilkan oleh beberapa pengguna jahat yang ingin mengganggu stabilitas dan ketertiban wilayah. Dari pengguna smartphone yang

banyak itulah, aplikasi deteksi kata kasar yang dibenamkan pada smartphone layak untuk dikembangkan. Dengan adanya aplikasi tersebut, para pengguna dapat menulis dulu kalimat komentar yang akan dikirim ke media sosial, kemudian aplikasi tersebut akan memberikan notifikasi apakah komentar tersebut termasuk kata kasar atau tidak.

Analisis sentiment untuk mendeteksi kata kasar telah diteliti banyak orang. Namun penelitian-penelitian sebelumnya pada [4]-[6] tidak pernah membahas bagaimana model pembelajaran ini dapat digunakan oleh pengguna akhir (end user). Maka, penelitian ini bermaksud mendeskripsikan bagaimana sebuah model pembelajaran dapat masuk ke tahap produksi dan dapat digunakan oleh end user dengan mudah. Pendekatan yang dipilih adalah dengan ekstraksi fitur TF-IDF dan model pembelajaran berbasis Naïve Bayes.

2. Metode Penelitian

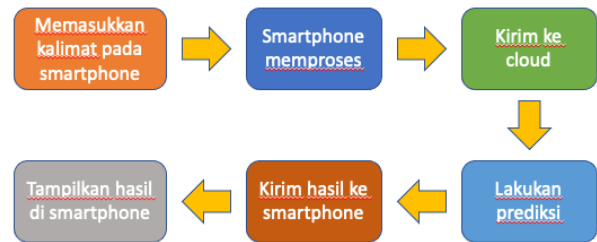
2.1. Tahap Perencanaan Framework

Flutter dikembangkan oleh google yang dapat digunakan untuk mengembangkan aplikasi mobile. Karena kemudahan penggunaannya, maka Flutter merupakan salah satu pilihan pengembang yang ingin membuat aplikasi multi-platform. Dengan kata lain, flutter dapat dimigrasi ke aplikasi android, IOS, web bahkan desktop. Dengan adanya fitur hot-reload, maka proses pengembangan aplikasi menjadi sangat cepat. Didukung dengan pembacaan API untuk mengirim dan menerima data, maka ukuran dari aplikasi menjadi sangat kecil.

Flask adalah kerangka kerja aplikasi web Antarmuka Server Server Gateway WSGI yang ringan yang dibuat untuk memudahkan memulai dan membuatnya mudah bagi pemula yang baru. Dengan kecenderungan meningkatkan aplikasi yang kompleks. Flask memiliki dasar di sekitar Werkzeug dan Jinja2 dan telah menjadi salah satu kerangka kerja aplikasi web Python paling populer.

Untuk pengembangan android, digunakan Flutter untuk mempercepat proses produksi. Disamping itu Flask digunakan sebagai rumah untuk model pembelajaran. Seperti ditunjukkan pada Gambar 1, prosesnya adalah sebagai berikut: 1) pengguna memasukkan kalimat dalam aplikasi android, 2) aplikasi akan mengirimkan pesan dan diterima FLASK menggunakan API, 3) kemudian FLASK memproses data yang diterima dan mengeluarkan prediksi apakah kalimat tersebut merupakan kata kasar atau tidak, 4) FLASK mengirim pesan ke aplikasi android dan menampilkan pesan kepada pengguna. Untuk proses pengiriman dan penerimaan data, dibutuhkan koneksi internet. Alasan model pembelajaran tidak ditaruh di aplikasi android dengan alasan: 1) untuk memperkecil footprint ukuran aplikasi, sehingga smartphone tetap ringan dan tidak cepat panas, dan 2) melindungi kekayaan intelektual jika ada yang berusaha membongkar aplikasi

android tersebut. Sedangkan untuk tahapan pembuatan model prediksi, dapat dilihat pada Gambar 2.



Gambar 1. Skema Aplikasi Hate Speech Detection berbasis Android



Gambar 2. Skema pembuatan model Hate Speech Detection berbasis TF-IDF dan Naïve Bayes

Pada Gambar 2, dapat dijelaskan cara skema pembuatan model Hate Speech detection. Tahapan dimulai dengan mengumpulkan dataset, kemudian mengecilkan huruf kapital, tokenizing kalimat menjadi satuan kata kata, kemudian kata-kata yang sudah terpisah diubah menjadi kata dasar, lalu menghapus kata hubung (stop words). Kemudian dari tahapan perencanaan ekstraksi fitur, digunakan TF-IDF. Setelah fitur terbentuk, Naïve Bayes digunakan sebagai model pembelajaran untuk klasifikasi. Parameter akurasi, presisi dan recall diukur untuk mengetahui performa dari model pembelajaran Naïve Bayes.

2.2. Tahap Perencanaan Ekstraksi Fitur

Analisis sentiment menggunakan analisis teks, *natural language processing* (NLP) dan teknik komputasi untuk mengotomatisasi ekstraksi atau klasifikasi sebuah sentiment pada ulasan sentiment [8]. Analisis sebuah opini atau sentiment sangat penting di banyak bidang, seperti: marketing, informasi kebutuhan konsumen, aplikasi, buku, social media dan website [9]. Analisis sentiment juga menjadi area yang sering dikembangkan pada pembuatan keputusan (*decision making*) [10]. Ratusan ribu user sangat tergantung pada ulasan sentiment, ini terbukti pada tahun 2013, sekitar 90% keputusan user atau konsumen tergantung pada ulasan online[11]. Tujuan utama analisis sentiment adalah menganalisis review (ulasan) dan menghitung skor dari sentiment tersebut. Review yang didapat bisa dikelompokkan menjadi review positif, negative atau netral, hal ini disebut

sebagai sentiment polarity [12]. Sentimen review yang saat ini muncul, bisa dikelompokkan menjadi tiga bagian [13], antara lain: (1) sentiment terstruktur, bisa ditemukan di review resmi, seperti di review penelitian atau review bedah buku, hal ini terjadi karena review dilakukan oleh professional; (2) sentiment semi terstruktur, biasanya ditemukan pada diskusi pro dan kontra; (3) sentiment tidak terstruktur, hal ini dapat ditemukan pada penulisan tidak resmi dan bebas yang tidak mengikuti kaidah penulisan yang benar.

Untuk menentukan sebuah review (ulasan) dari sebuah kalimat, terlebih dalam sebuah dokumen, diperlukan ekstraksi fitur untuk mendapatkan feature vectors. Dari feature vectors ini, kemudian akan dilatih menggunakan pembelajaran klasifikasi (classification learning model) untuk mendapatkan hasil polaritas review tersebut. Saat ini, dikenal beberapa metode dalam pembuatan feature vector untuk tipe data dokumen, antara lain: *Bag of Words* (BoW) dan *Term Frequency and Inverse Document Frequency* (TF-IDF). BoW merupakan sebuah pendekatan algoritma yang menghitung seberapa banyak sebuah kata muncul (frekuensi) pada sebuah dokumen. Kelemahan BoW adalah urutan term dan kelangkaan (rareness) term tidak dipertimbangkan. Sedangkan pada TF-IDF, sebuah kata diberikan sebuah weight berupa TF dan IDF score, bukan berupa frekuensi seperti BoW.

Bag of Words (BoW) dapat diibaratkan seperti mesin yang menerima inputan sebuah dokumen dan mengeluarkan output berupa table yang berisi jumlah frekuensi kata yang ada untuk tiap dokumen. Misalkan terdapat tiga buah dokumen dengan kalimat sebagai berikut: (1) Saya suka kacang, (2) Saya alergi kacang dan udang, dan (3) Saya suka udang. Dari tiga kalimat tersebut, didapatkan BoW seperti ditunjukkan pada Tabel 1.

Tabel 1. contoh BoW

	Saya	Suka	kacang	Alergi	Dan	udang
Dok 1	1	1	1	-	-	-
Dok 2	1	-	1	1	1	1
Dok 3	1	1	-	-	-	1

Untuk dokumen yang spesifik, *Term Frequency* (TF) menentukan seberapa penting sebuah kata dilihat dari seberapa sering kata-kata itu muncul pada sebuah dokumen. TF merupakan output dari BoW. Pada TF-IDF, komponen kedua merupakan inverse document frequency (IDF). Pada IDF, sebuah kata dipertimbangkan sebagai penting pada sebuah dokumen jika kata tersebut tidak terlalu sering muncul pada dokumen yang lain. Hal ini bisa dihitung sebagai berikut:

$$idf(Kata) = \log \frac{jd}{jdkata} \quad (1)$$

Dimana jd adalah jumlah dokumen dan $jdkata$ adalah jumlah dokumen yang mengandung kata.

Sebagai contoh, dapat dilihat pada tabel 2 dengan menggunakan contoh BoW

Tabel 2. contoh TF-IDF

	Saya	Suka	Kacang	Alergi	Dan	Udang
Dok 1	0	0.18	0.18	-	-	-
Dok 2	0	-	0.18	0.48	0.48	0.18
Dok 3	0	0.18	-	-	-	0.18

Tabel 2 menunjukkan bahwa pada dokumen 1 penekanannya adalah 'suka' 'Kacang', pada dokumen 2 adalah 'alergi' 'dan', pada dokumen 3 adalah 'suka' 'Udang'. Kata 'dan' secara umum akan dihapus menggunakan stop words sebelum dilakukan ekstraksi fitur.

Pada pemrosesan dokumen, skema representasi teks biasanya menggunakan vector space model (VSM) yang sering digunakan untuk pembobotan kata (word-weighting). Hasil yang diterima dari VSM merupakan dokumen yang relevan. VSM yang digunakan disini menggunakan kata kunci atau frase, yang secara umum dikenal sebagai unigram, bigram, trigram dan n-gram [14]–[16]. Untuk lebih mudahnya, N-gram merupakan urutan dari kata-kata N. misalkan terdapat sebuah kalimat: "pelayanan hotel tidak terlalu baik", maka jika dibuat n-gram didapatkan sebagai berikut:

1. Unigram: 'pelayanan', 'hotel', 'tidak', 'terlalu', 'baik'
2. Bigram: 'pelayanan hotel', 'hotel tidak', 'tidak terlalu', 'terlalu baik'
3. Trigram: 'pelayanan hotel tidak', 'hotel tidak terlalu', 'tidak terlalu baik'

Metode N-gram membuat keputusan dengan membandingkan nilai ini dengan ratio similaritas, yang didefinisikan sebagai ratio identic N-gram dibandingkan dengan jumlah total N-grams. Rasio similaritas dapat dihitung dengan cara [17]:

$$ratio = \frac{\delta}{\min(\alpha, \beta)} \quad (2)$$

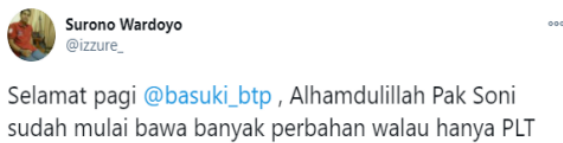
Dimana δ adalah jumlah n-gram yang identik, sementara α dan β masing-masing adalah jumlah n-gram untuk $kata_A$ dan $kata_B$. Disini $kata_A$ merupakan kata pertama dan $kata_B$ merupakan kata kedua yang digunakan sebagai pembanding karakter n-grams. Dari tahap ini, penulis memilih TF-IDF sebagai metode ekstraksi fitur.

2.3. Tahap Pembelajaran Prediksi

Metode penelitian yang dilakukan menggunakan tipe eksperimen, dimana tahapan penelitian tersebut adalah: pengumpulan dataset, preprocessing pada dataset, ekstraksi fitur, pengembangan model pembelajaran dan pengembangan aplikasi prototip berbasis flask ke tahap produksi yang bertujuan agar model pembelajaran dapat digunakan oleh pengguna akhir dengan mudah.

Tahap pertama adalah pengumpulan dataset, Dataset yang berkaitan dengan kata kasar yang digunakan berasal dari data yang digunakan pada penelitian [18]-[22].

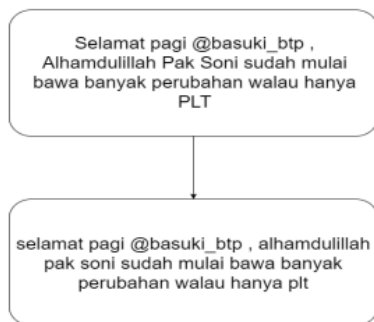
Tahap berikutnya adalah text preprocessing merupakan tahap awal dari proses text mining yang mencakup seluruh proses mempersiapkan teks menjadi data yang akan digunakan. Proses yang dikerjakan adalah *case folding* dan tokenizing. Berikut contoh tweet yang akan digunakan sebagai masukan untuk diproses seperti Gambar 3.



Gambar 3. Contoh tweet

Case folding merupakan proses pengubahan huruf besar menjadi huruf kecil dan menghilangkan karakter yang tidak valid. Dari contoh tweet pada Gambar 3, akan berubah menjadi seperti Gambar 4 setelah melalui proses case folding. Proses selanjutnya adalah tokenizing, yaitu proses penguraian atau pemecahan deskripsi yang awalnya berupa kalimat menjadi sebuah kata yang disebut token seperti pada Gambar 5.

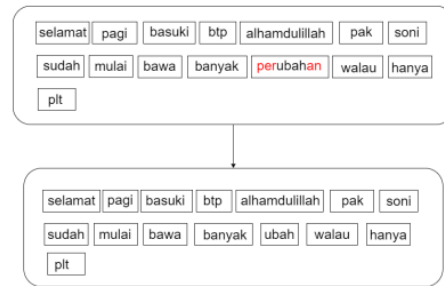
Hasil yang telah diperoleh pada tahap sebelumnya akan digunakan pada tahap text transformation. Pada tahap ini terjadi proses stemming, yaitu merubah kata menjadi kata dasar seperti yang ditunjukkan pada Gambar 6.



Gambar 4. Contoh case folding

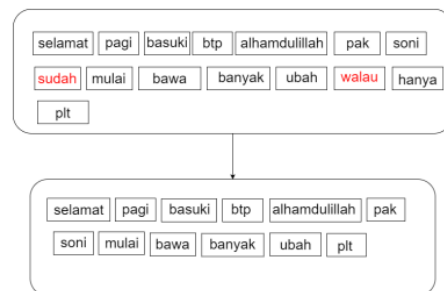


Gambar 5. Contoh tokenizing



Gambar 6. Contoh menjadikan kata dasar

Kemudian dilakukan penghapusan kata stopwords, yaitu penghapusan kata yang umum digunakan tetapi tidak mempunyai dampak yang signifikan dalam kalimat. Atau dengan arti lain, kata tersebut bukan merupakan ciri unik dari suatu dokumen seperti kata sambung dan kata keterangan, contohnya adalah “dari”, “akan”, “seorang”, dan lain sebagainya seperti pada Gambar 7.



Gambar 7. Contoh menghapus stop word

Naïve Bayes digunakan sebagai model pembelajaran dimana merupakan salah satu algoritma yang termasuk kedalam teknik supervised learning yang digunakan untuk mencari nilai probabilitas tertinggi untuk mengklasifikasi data uji pada kategori yang tepat. Dalam Naïve Bayes Classifier, setiap dokumen direpresentasikan dengan pasangan atribut “ $x_1, x_2, x_3, \dots, x_n$ ”, x_1 adalah kata pertama, x_2 adalah kata kedua, x_3 adalah kata ketiga, dan seterusnya. Sedangkan V adalah himpunan kategori opini. Ketika proses klasifikasi, algoritma akan mencari probabilitas dengan nilai tertinggi diantara semua kategori yang diuji, yang mana persamaannya sebagai berikut.

$$V_{MAP} = \operatorname{argmax}_{V_j \in V} \prod_{i=1}^n (P(x_i|V_j)P(V_j)) \quad (3)$$

Dimana V_j merupakan kategori opini, $P(x_i|V_j)$ merupakan probabilitas x_i pada kategori V_j dan $P(V_j)$ merupakan probabilitas dari V_j

Untuk $P(V_j)$ dan $P(x_i|V_j)$ dihitung saat pelatihan, persamaannya adalah sebagai berikut:

$$P(V_j) = \frac{|docs\ j|}{|contoh|} \quad (4)$$

$$P(x_i|V_j) = \frac{n_k + 1}{n + |kosakata|} \quad (5)$$

Dimana $|docs\ j|$ adalah jumlah dokumen pada kategori j , $|contoh|$ adalah jumlah dokumen pada semua kategori, N_k adalah jumlah kemunculan kata x_i pada V_j , N adalah jumlah kata dalam setiap kategori, dan $|kosakata|$ adalah jumlah semua kata dari semua kategori

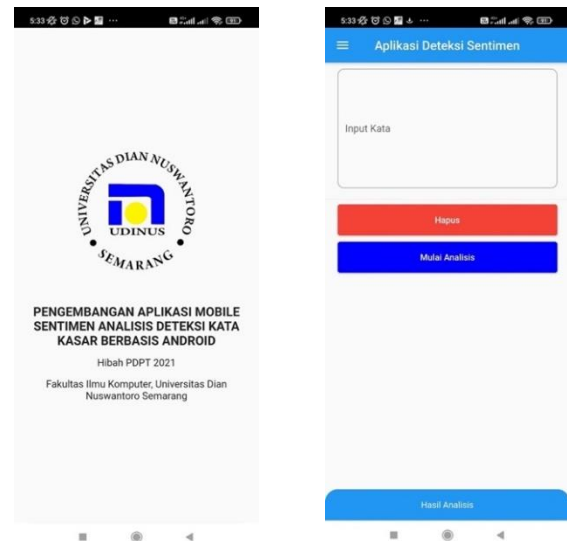
3. Hasil dan Analisis

Pada pengujian proses pembelajaran, proporsi skema training dan testing sebesar 70% dan 30%. Kemudian data training digunakan untuk mencari model dan parameter terbaik dengan metode *brute force 5-fold cross validation*. Setelah ditemukan parameter terbaik, digunakan sebagai dasar framework untuk prediksi. Hasil performa pada Naïve Bayes dirangkum pada Tabel 3.

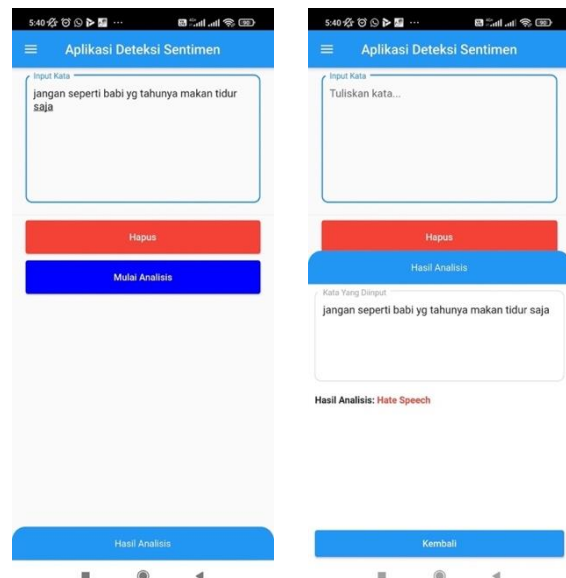
Tabel 3. Performa Klasifikasi

	Training	Testing
Akurasi	98%	84%
Recall	98%	86%
Precision	99%	83%

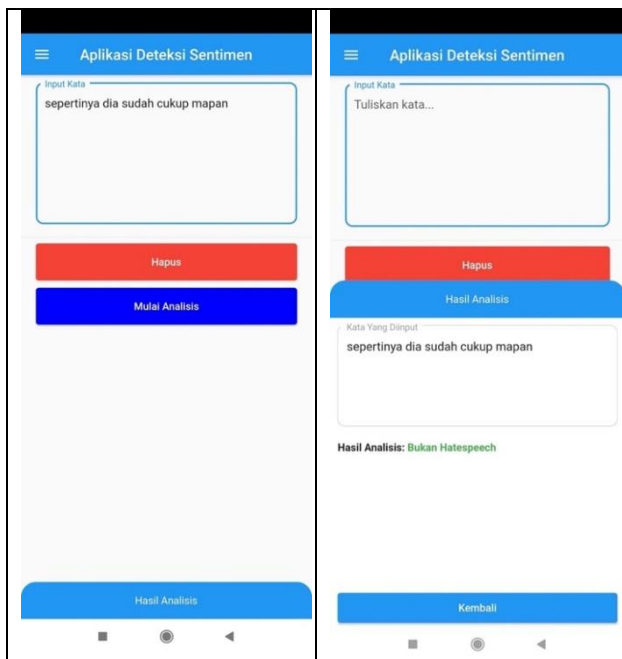
Tabel 3 menunjukkan bahwa akurasi, recall dan precision pada testing lebih rendah daripada training sebesar 10%. Hal ini merupakan tanda bahwa model pembelajaran ini terdampak pada *overfitting* (dimana jika diberikan data uji baru, kemungkinan akan meleset dari prediksi asli). Keadaan ini menunjukkan bahwa diperlukan metode lain untuk ekstraksi fitur dan penanganan fitur lebih baik untuk hate speech detection. Alasan lain adalah data yang dimasukkan masih umum dan tidak terpolarisasi tajam.



Gambar 8. Home Screen dan Input Data



Gambar 9. Input dan hasil prediksi kata kasar



Gambar 10. Input dan hasil prediksi kata normal

Tahap selanjutnya adalah memasang hasil pembelajaran pada Flask, kemudian diunggah ke cloud, agar smartphone bisa meminta layanan untuk melakukan prediksi kata kasar, hasil dapat dilihat pada Gambar 8-10. Pada gambar 8, android flutter akan menampilkan halaman homescreen tentang deskripsi aplikasi. Kemudian akan menampilkan halaman utama yang dapat diisi kata-kata untuk dilakukan analisis prediksi.

Pada Gambar 9 dan 10, diberikan inputan data kasar dan inputan data normal, setelah selesai menulis dan menekan tombol mulai analisis, maka android flutter akan mengirim data tersebut ke cloud, kemudian machine learning yang ada di cloud melakukan prediksi, setelah keluar hasil probabilitasnya, hasil tersebut dikirim ke smartphone untuk ditampilkan hasilnya.

4. Kesimpulan

Aplikasi Deteksi Kata Kasar menggunakan Naïve Bayes berbasis cloud dan smartphone telah dipaparkan. Dengan mendapat hasil testing berupa accuracy 84.26%, recall 86.81%, dan precision 83.15%, dapat disimpulkan bahwa aplikasi berjalan cukup baik, namun masih perlu ditingkatkan performanya agar mendekati sempurna seperti penggunaan ekstraksi fitur yang lain dan metode pembersihan data yang lebih lanjut. Kemudian, aplikasi yang dikembangkan menggunakan Flutter ini, dapat berkomunikasi dengan model pembelajaran menggunakan API, berimbas positif pada berkurangnya waktu pemrosesan data dan ukuran file pada smartphone.

Hal yang dapat dilakukan pada penelitian selanjutnya adalah: 1) menggunakan fitur ekstraksi yang lain untuk

mengetahui tingkat performa yang lebih baik, 2) metode pembersihan data akan lebih dieksplorasi, 3) pembuatan cloud untuk crowd source dataset, maksudnya adalah semua data yang diinput oleh pengguna, akan disimpan di cloud, kemudian akan dilakukan pembelajaran dari dataset yang terkumpul; 4) menggunakan topik lain selain politik, seperti bidang Pendidikan, Kesehatan dan ekonomi.

Ucapan Terima Kasih

Kegiatan penelitian ini terwujud atas hibah Penelitian Dasar Perguruan Tinggi dari Universitas Dian Nuswantoro Semarang dan Center of Excellence Universitas Dian Nuswantoro dalam pendampingan penggunaan sumber daya computer untuk kegiatan penelitian ini.

Referensi

- [1]. D. Smit, "Cyberbullying in South African and American schools: A legal comparative study," *South African J. Educ.*, vol. 35, no. 2, pp. 1–11, May 2015, doi: 10.15700/saje.v35n2a1076.
- [2]. R. Pelle, C. Alcântara, and V. P. Moreira, "A classifier ensemble for offensive text detection," *WebMedia 2018 - Proc. 24th Brazilian Symp. Multimed. Web*, pp. 237–243, 2018, doi: 10.1145/3243082.3243111.
- [3]. R. Zhao, A. Zhou, and K. Mao, "Automatic detection of cyberbullying on social networks based on bullying features," in *Proceedings of the 17th International Conference on Distributed Computing and Networking - ICDCN '16*, 2016, pp. 1–6, doi: 10.1145/2833312.2849567.
- [4]. J. H. Park and P. Fung, "One-step and Two-step Classification for Abusive Language Detection on Twitter," pp. 41–45, Jun. 2017, doi: 10.18653/v1/w17-3006.
- [5]. C. Jefferson, H. Liu, and M. Cocea, "Fuzzy approach for sentiment analysis," in *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, Jul. 2017, pp. 1–6, doi: 10.1109/FUZZ-IEEE.2017.8015577.
- [6]. P. Burnap and M. L. Williams, "Cyber Hate Speech on Twitter: An Application of Machine Classification and Statistical Modeling for Policy and Decision Making," *Policy & Internet*, vol. 7, no. 2, pp. 223–242, Jun. 2015, doi: 10.1002/poi3.85.
- [7]. K. Simon, "DIGITAL 2021: INDONESIA," 2021. <https://datareportal.com/reports/digital-2021-indonesia>.
- [8]. B. Agarwal, N. Mittal, P. Bansal, and S. Garg, "Sentiment Analysis Using Common-Sense and Context Information," *Comput. Intell. Neurosci.*, vol. 2015, pp. 1–9, 2015, doi: 10.1155/2015/715730.
- [9]. F. Elghannam, "Text representation and classification based on bi-gram alphabet," *J. King Saud Univ. - Comput. Inf. Sci.*, Jan. 2019, doi: 10.1016/j.jksuci.2019.01.005.
- [10]. T. Chalothorn and J. Ellman, "Simple approaches of sentiment analysis via ensemble learning," in *Lecture Notes in Electrical Engineering*, vol. 339, 2015, pp. 631–639.
- [11]. L. Peng, G. Cui, M. Zhuang, and C. Li, "Working Paper Series," *Account. Financ.*, vol. 24, no. 1, pp. 75–75, 2009, doi: 10.1111/j.1467-629x.1984.tb00054.x.

- [12] K. Khan, B. Baharudin, A. Khan, and A. Ullah, "Mining opinion components from unstructured reviews: A review," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 26, no. 3, pp. 258–275, Sep. 2014, doi: 10.1016/j.jksuci.2014.03.009.
- [13] D. M. E.-D. M. Hussein, "A survey on sentiment analysis challenges," *J. King Saud Univ. - Eng. Sci.*, vol. 30, no. 4, pp. 330–338, Oct. 2018, doi: 10.1016/j.jksues.2016.04.002.
- [14] F. Xie, X. Wu, and X. Zhu, "Efficient sequential pattern mining with wildcards for keyphrase extraction," *Knowledge-Based Syst.*, vol. 115, pp. 27–39, Jan. 2017, doi: 10.1016/j.knosys.2016.10.011.
- [15] L. H. Lee, D. Isa, W. O. Choo, and W. Y. Chue, "High Relevance Keyword Extraction facility for Bayesian text classification on different domains of varying characteristic," *Expert Syst. Appl.*, vol. 39, no. 1, pp. 1147–1155, Jan. 2012, doi: 10.1016/j.eswa.2011.07.116.
- [16] F. Figueiredo, L. Rocha, T. Couto, T. Salles, M. A. Gonçalves, and W. Meira Jr., "Word co-occurrence features for text classification," *Inf. Syst.*, vol. 36, no. 5, pp. 843–858, Jul. 2011, doi: 10.1016/j.is.2011.02.002.
- [17] B. C. Gencosman, H. C. Ozmutlu, and S. Ozmutlu, "Character n-gram application for automatic new topic identification," *Inf. Process. Manag.*, vol. 50, no. 6, pp. 821–856, Nov. 2014, doi: 10.1016/j.ipm.2014.06.005.
- [18] M. S. Saputri, R. Mahendra, and M. Adriani, "Emotion Classification on Indonesian Twitter Dataset," in *2018 International Conference on Asian Language Processing (IALP)*, Nov. 2018, pp. 90–95, doi: 10.1109/IALP.2018.8629262.
- [19] M. O. Ibrohim and I. Budi, "A Dataset and Preliminaries Study for Abusive Language Detection in Indonesian Social Media," *Procedia Comput. Sci.*, vol. 135, pp. 222–229, 2018, doi: 10.1016/j.procs.2018.08.169.
- [20] M. O. Ibrohim and I. Budi, "Multi-label Hate Speech and Abusive Language Detection in Indonesian Twitter," in *Proceedings of the Third Workshop on Abusive Language Online*, 2019, pp. 46–57, doi: 10.18653/v1/W19-3506.
- [21] I. Alfina, R. Mulia, M. I. Fanany, and Y. Ekanata, "Hate speech detection in the Indonesian language: A dataset and preliminary study," in *2017 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, Oct. 2017, pp. 233–238, doi: 10.1109/ICACSIS.2017.8355039.
- [22] R. Hendrawan, Adiwijaya, and S. Al Faraby, "Multilabel Classification of Hate Speech and Abusive Words on Indonesian Twitter Social Media," in *2020 International Conference on Data Science and Its Applications (ICoDSA)*, Aug. 2020, pp. 1–7, doi: 10.1109/ICoDSA50139.2020.9212962.