Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN: 2548-964X
Vol. 6, No. 1, Januari 2022, hlm. 148-153 http://j-ptiik.ub.ac.id

Klasifikasi Pertanyaan COVID-19 Bahasa Indonesia menggunakan Naïve Bayes

Glenn Jonathan Satria[1], Putra Pandu Adikara[2], Randy Cahya Wihandika[3]

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: [1]glenn.jonathan.gj@gmail.com, [2]adikara.putra@ub.ac.id, [3]rendicahya@ub.ac.id

Abstrak

Question Answering (QA) adalah sistem yang dapat memberikan jawaban dari pertanyaan yang diberikan oleh pengguna. Dalam QA terdapat satu tugas yang bernama analisis pertanyaan. Analisis pertanyaan berguna untuk memilih tipe pertanyaan apa yang diberikan pengguna melalui *query*. Analisis pertanyaan bisa dicari menggunakan klasifikasi. Penelitian ini menggunakan Naïve Bayes sebagai metode klasifikasi. Selain itu, digunakan beberapa proses dalam pemrosesan bahasa alami seperti ekstraksi fitur kata tanya dan *preprocessing* yang berisi *data cleaning*, *stemming*, *stopword removal*, dan *tokenization*. Tahap selanjutnya adalah membangun model klasifikasi melalui data latih yang berisi 16 kategori pertanyaan. Berdasarkan hasil pengujian dengan 2 skenario yaitu menggunakan *preprocessing* dan tidak menggunakan *preprocessing*, menghasilkan nilai akurasi menggunakan *preprocessing* sebesar 0,58634. Pengujian tanpa menggunakan *preprocessing* menghasilkan nilai akurasi sebesar 0,65060. Penggunaan *preprocessing* dalam klasifikasi pertanyaan berpengaruh negatif karena berhubungan dengan konteks pertanyaan yang diberikan.

Kata kunci: Naïve Bayes, *preprocessing*, Klasifikasi Pertanyaan

Abstract

Question Answering (QA) is a system that provide answer from question given from user. In QA there is one task called question analysis. Question analysis act as type chooser from query user input. Question analysis can be found with classification. This research using Naïve Bayes as classification method. Furthermore, several process used from natural language processing such as question feature extraction and preprocessing contain data cleaning, stemming, stopword removal, and tokenization. Next phase is to build a classification model from training data which contain 16 question categories. Based on test result with 2 scenarios with preprocessing and without preprocessing, we obtained accuracy value of 0,58364 with preprocessing. We also obtained accuracy value of 0,65060 without preprocessing. Application of preprocessing in question classification have a negative impact because it changed the given question context.

Keywords: *Naïve Bayes, preprocessing, Question Classification*

1. PENDAHULUAN

Question Answering (QA) adalah sistem yang otomatis menyediakan jawaban dari pertanyaan yang ditanyakan oleh manusia dalam bahasa alami (Abdelghani, et al., 2015). Tugas dari QA dapat dibagi menjadi 3 bagian berbeda yaitu: analisis pertanyaan, pengambilan dokumen, dan ekstraksi jawaban. Mayoritas *Question Answering System* (QAS) memiliki 3 tugas ini, tetapi bisa juga berbeda tergantung bagaimana QAS dibangun (Abdelghani, et al., 2015).

Persoalan yang sering dihadapi dalam membangun *Question Answering System* menggunakan bahasa alami adalah cara menghubungkan QAS dengan sisi pengguna yang menanyakan beberapa tipe pertanyaan. Contoh, pertanyaan yang sering ditanyakan adalah kata: kapan, dimana, seberapa banyak, siapa, dan apa, kata-kata tersebut merujuk pada waktu/tanggal, tempat, orang, dan kelompok organisasi. Tipe pertanyaan lainnya adalah pertanyaan tentang istilah atau sebuah konsep. Pertanyaan yang menggunakan "kenapa" dan "bagaimana" adalah tipe yang sangat sulit dijawab dan sangat sedikit upaya untuk

menjawab pertanyaan dengan tipe seperti ini.

Salah satu tahapan dalam membangun QAS adalah klasifikasi tipe pertanyaan. Untuk menjawab pertanyaan dengan jawaban yang tepat dibutuhkan pengertian dari tipe informasi yang diberikan oleh pertanyaan. Karena mengetahui tipe pertanyaan dapat memberikan konteks dari data yang dibutuhkan atau jawabannya. Pertanyaan dapat diklasifikasikan melalui tipenya: pertanyaan apa, mengapa, siapa, bagaimana, kapan, dan dimana (Allam & Haggag, 2012).

Untuk klasifikasi setiap pertanyaan mengenai COVID-19, peneliti telah mengumpulkan berbagai jenis pertanyaan dari situs tanya jawab seperti Google, Bing, Quora, Yahoo Answer, dan lain-lain. Tujuan dari pengumpulan pertanyaan ini untuk memilah setiap pertanyaan dan mengelompokkan pertanyaan melalui konteks nya. Pertanyaan yang paling umum untuk COVID-19 adalah transmisi, pencegahan, dan efek sosial dari COVID (Wei, et al., 2020).

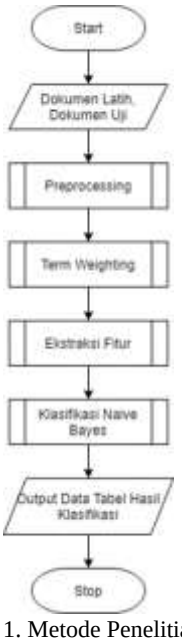
Dalam klasifikasi teks diperlukan sebuah proses untuk membuat data yang akan digunakan menjadi sederhana yaitu *preprocessing*. *Preprocessing* memberikan kualitas data yang lebih baik, seperti menghilangkan angka, tanda baca, kata yang sering muncul seperti: yang, ada, dan, dan sebagainya. Penggunaan *preprocessing* yang salah dapat membuat rendahnya kinerja, sehingga penggunaan *preprocessing* harus dilihat dari urgensi nya (Gharatkar, et al., 2017).

Naïve Bayes adalah algoritme klasifikasi yang sangat populer. Tugas utama dari Naïve Bayes adalah menghitung probabilitas dan statistik untuk memprediksi peluang di masa depan berdasarkan kondisi di masa sebelumnya (Bustami, 2014). Penelitian mengenai pertanyaan medis telah dilakukan oleh Sarrouti (2017) pertanyaan yang didapatkan di kategorikan menjadi 4 kategori berbeda. Dengan menggunakan SVM, Sarrouti berhasil mendapatkan akurasi sebesar 89,40%. Penelitian yang telah dilakukan oleh Yohanes (2018) menunjukkan metode Naïve Bayes menghasilkan akurasi jawaban sebesar 100% dengan *stemming* dan 96,43% tanpa *stemming*.

Pada penelitian ini penulis akan merancang model klasifikasi pertanyaan berbahasa Indonesia menggunakan dataset yang digunakan adalah *dataset* COVID-Q yang diterjemahkan kedalam bahasa Indonesia. Algoritme klasifikasi yang akan digunakan adalah Naïve Bayes. Diharapkan penggunaan algoritme Naïve Bayes pada penelitian ini dapat menghasilkan model klasifikasi pertanyaan yang lebih akurat.

2. METODOLOGI

Pada penelitian ini terdapat tahap-tahap dari penerapan algoritme. Tahap pertama adalah melakukan *preprocessing* dokumen. *Preprocessing* akan menghasilkan kata unik yang dapat dipakai dalam klasifikasi dengan menghilangkan kata-kata yang dapat mengganggu akurasi. Setelah melakukan *preprocessing* dilakukan ekstraksi fitur. Fitur-fitur yang diekstrak adalah kata tanya, kata tanya akan memiliki *bag-of-words* nya sendiri berbeda dari *bag-of-words* kalimat. Fitur yang telah diekstrak sudah memiliki bobotnya masing-masing yang dapat digunakan untuk klasifikasi. Klasifikasi yang digunakan adalah Naïve Bayes dengan membangun model terlebih dahulu dengan data latih dan menghitung *prior* tiap kategori. Model klasifikasi yang telah dilatih dapat digunakan untuk melakukan klasifikasi terhadap data uji. Hasil dari klasifikasi adalah *posterior* tiap kategori, dengan cara membandingkan mana hasil *posterior* yang lebih tinggi maka didapatkan hasil klasifikasi. Tahapan dari strategi ini dapat dilihat pada Gambar 1.



Gambar 1. Metode Penelitian

Tahap pertama dari penelitian ini adalah *preprocessing* menggunakan *data cleaning, stopword removal, stemming,* dan *tokenization.* Tahap kedua adalah menghitung *term frequency* untuk menghitung kemunculan *term* dalam

dokumen, setelah dilakukan perhitungan *frequency* fitur kata tanya akan dipisahkan dengan fitur *term* untuk meningkatkan akurasi. Tahap ketiga adalah mencari *prior* tiap kelas yang ada dalam dokumen. Nilai *prior* dihitung dengan Persamaan 1.

$$P(c) = \frac{Nc}{N} \quad (1)$$

Keterangan:
P(c) = *Prior*
Nc = jumlah dokumen dalam kelas c
N = banyak dokumen latih

Tahap selanjutnya adalah menghitung *likelihood*. *Laplace smoothing* digunakan untuk menghindari hasil 0 dalam hasil *term* jika ada yang tidak muncul. *Likelihood* dihitung dengan Persamaan 2.

$$P(w_i|c_j) = \frac{count(w_i,c_j)+1}{(\Sigma_{w\in V} count(w,c_j))+|V|} \quad (2)$$

Keterangan:
P(wi|cj) = *Likelihood*
Count(wi, cj) = Jumlah kata wi dalam cj
$(\Sigma_{w\in V} count(w,c_j))$ = Jumlah kata w dalam cj
|V| = Jumlah seluruh kata dalam dokumen

Tahap selanjutnya adalah mencari *posterior* menggunakan *prior* dan *likelihood*. Nilai *posterior* akan dibandingkan dan nilai tertinggi akan menjadi nilai *posterior*. *Posterior* dihitung dengan Persamaan 3.

$$P(C|X) = \frac{P(X|C)\times P(C)}{P(X)} \quad (3)$$

Keterangan:
X = Data dengan *class* yang belum diketahui
C = Hipotesis data X merupakan suatu class
P(C|X) = Probabilitas C berdasarkan kondisi X (*posterior*)
P(C) = Probabilitas C (*prior*)
P(X|C) = Probabilitas X berdasarkan kondisi C
P(X) = Probabilitas X

Pada tahap pengujian menggunakan *confusion matrix multi class* karena memiliki 16 kelas untuk menghitung akurasi, *precision*, *recall*, dan *f-measure*. *Confusion matrix* akan menampilkan hasil dari klasifikasi.

Confusion matrix dapat digunakan untuk mencari akurasi, *precision*, *recall*, dan *f-measure*. Akurasi adalah presentasi klasifikasi sistem dalam predikisi data. Nilai akurasi dihitung dengan Persamaan 4.

$$Accuracy = \frac{TTP+TTN}{TTP+TTN+TFP+TFN} \quad (4)$$

Precision adalah tingkat ketepatan informasi yang diminta oleh pengguna dengan hasil yang diperoleh. Nilai *precision* dihitung dengan Persamaan 5.

$$Precision = \frac{TTPall}{TTPall+TFPi} \quad (5)$$

Recall adalah tingkat jumlah banyak dan sedikitnya kesesuaian informasi yang didapat dari hasil penelitian. Nilai *recall* dihitung dengan Persamaan 6.

$$Recall = \frac{TTPall}{TTPall+TFNi} \quad (6)$$

F-measure adalah perbandingan rata-rata presisi dan recall yang dibobotkan. Nilai *f-measure* dihitung dengan Persamaan 7.

$$F - measure = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (6)$$

Keterangan:
Total True Positive (TTP) = Total data yang ter klasifikasi dengan benar
Total True Negative (TTN) = Total data yang ter klasifikasi negative dengan benar
Total False Positive (TFP) = Total data yang diprediksi benar, namun salah
Total False Negative (TFN) = Total data yang diprediksi salah, namun benar

3. IMPLEMENTASI

Proses implementasi dilakukan dengan membuat *source code* menggunakan Jupyter Notebook. Proses implementasi meliputi proses pembacaan data latih dan uji, *preprocessing* berisi *data cleaning*, *stopword removal*, *stemming*, *tokenization*, ekstraksi fitur kata tanya, dan klasifikasi Naïve Bayes.

Pembacaan data dilakukan dengan *library* Pandas. Data latih adalah *file excel* berisi 996 pertanyaan dengan kelas nya masing-masing.

Proses kedua adalah *preprocessing* proses ini mencakup *data cleaning* yaitu menghilangkan angka dan tanda baca, *stopword removal* untuk menghilangkan kata-kata.

Proses ketiga adalah ekstraksi fitur kata tanya. Kata tanya yang digunakan dalam data latih terdapat pada Tabel 1.

Tabel 1. Kata Tanya

No	Kata Tanya
1	Apa
2	Apakah
3	Bagaimana
4	Kapan
5	Kenapa
6	Mengapa

7	Siapa
8	Berapa
9	Mana

Kata dalam Tabel 1 akan dimasukkan kedalam metriks yang berbeda dengan fitur *term* untuk meningkatkan akurasi. Tahap selanjutnya adalah implementasi Naïve Bayes. Implementasi perhitungan *prior* dilakukan dengan menghitung jumlah data n dalam kelas kemudian jumlah tersebut dibagi dengan banyak data, hasil *prior* disimpan dalam *dictionary*. Selanjutnya, tahap perhitungan *term* tiap kelas yang akan digunakan dalam mencari *likelihood*, implementasi ini dilakukan untuk mencari berapa banyak *term* dalam suatu kelas tertentu.

Implementasi *likelihood* dilakukan dengan rumus *laplace smoothing* untuk menghindari nilai 0 dalam data. Jumlah *term* dalam kelas yang telah dilakukan sebelumnya akan digunakan dalam fungsi ini.

Tahap terakhir adalah perhitungan Naïve Bayes, dalam fungsi ini data uji dimasukkan dan dilakukan *preprocessing*, kemudian hasil *prior* dan *likelihood* yang sebelumnya didapatkan digunakan dalam fungsi ini untuk mencari *posterior*. Hasil *posterior* yang paling tinggi akan digunakan sebagai prediksi data uji.

4. PENGUJIAN DAN ANALISIS

Sebelum melakukan pengujian, dokumen yang berjumlah 1245 akan menggunakan *holdout* dengan rasio 80% data latih dan 20% data uji. *Holdout* akan digunakan dalam penelitian ini karena jumlah data yang banyak dan model klasifikasi yang dibangun dimulai dari awal.

Pengujian akan dilakukan dengan 2 skenario yaitu menggunakan *stopword removal* dan *stemming* dan tidak menggunakan *stopword removal* dan *stemming*. Pengujian 2 skenario digunakan untuk membandingkan pengaruh dari *stopword removal* dan *stemming* dalam klasifikasi pertanyaan.

4.1. Skenario Pengujian Menggunakan *Stopword Removal* dan *Stemming*

Skenario pengujian 1 adalah melakukan klasifikasi menggunakan metode Naïve Bayes dengan melakukan *preprocessing* penuh, yaitu menggunakan *data cleaning*, *stopword removal*, *stemming*, dan tokenisasi. Hasil pengujian dijadikan acuan adalah *micro average* karena *micro average* dapat memberikan nilai yang lebih akurat untuk klasifikasi *multiclass*, *macro average* dipakai untuk menjadi perbandingan. Hasil pengujian skenario 1 ditunjukkan dalam Tabel 2.

Tabel 2. Rata-rata menggunakan *stopword removal* dan *stemming*

	Micro Average	***Macro Average***
Rata-rata Precision	0,58634	0,62778
Rata-rata Recall	0,58634	0,57473
Rata-rata F-measure	0,58634	0,56021

Dari hasil yang ditunjukkan Tabel 2 *micro average* memiliki nilai yang sama karena total FP dan FN dalam *confusion matrix* bernilai sama. Untuk *macro average* cukup menjumlahkan keseluruhan nilai lalu dibagi banyaknya class.

Skenario 2 adalah klasifikasi tanpa menggunakan *stopword removal* dan *stemming*, yang dipakai hanya *data cleaning* dan tokenisasi. Hasil pengujian skenario 2 ditunjukkan dalam Tabel 3.

Tabel 3 Rata-rata tidak menggunakan *stopword removal* dan *stemming*

	Micro Average	***Macro Average***
Rata-rata Precision	0,65060	0,59196
Rata-rata Recall	0,65060	0,56515
Rata-rata F-measure	0,65060	0,56635

Dari kedua tabel didapatkan bahwa akurasi klasifikasi lebih tinggi jika tidak memakai *stopword removal* dan *stemming*. Hal tersebut dikarenakan tidak terjadinya pemotongan kata oleh *stemming*. Klasifikasi pertanyaan memiliki ketergantungan kuat dengan data latihnya karena arti kata memiliki peran penting untuk penentuan kelas. Saat melakukan *stopword removal* dan *stemming* banyak kata yang kehilangan arti membuat pertanyaan tersebut tidak memiliki arti.

5. KESIMPULAN DAN SARAN

Proses pengujian menghasilkan *confusion matrix*. Dua skenario digunakan untuk melihat perbedaan hasil pengujian, pertama adalah skenario menggunakan *stopword removal* dan

stemming, kedua adalah skenario tidak menggunakan *stopword removal* dan *stemming*. Pengujian menggunakan *micro average* pada skenario 2 menghasilkan akurasi sebesar 0,65060 yang lebih tinggi jika dibandingkan dengan skenario 1 yang memiliki akurasi sebesar 0,58634. Penggunaan *stopword removal* dan *stemming* menurunkan kinerja klasifikasi pertanyaan karena proses tersebut menghilangkan definisi dari pertanyaan aslinya.

Saran dari penulis berdasarkan penelitian ini adalah menghapus penggunaan *stopword removal* dan *stemming*. Karena dalam kasus klasifikasi pertanyaan, konteks dari pertanyaan mempengaruhi hasil akurasi yang diberikan.

6. DAFTAR PUSTAKA

Abdelghani, B., Bouchiha, D., Doumi, N. & Malki, M., 2015. Question Answering Systems: Survey and Trends. *Procedia Computer Science*, Volume 73, pp. 366-375.

Allam, A. M. N. & Haggag, M. H., 2012. The question answering systems: A survey. *International Journal of Research and Reviews in Information Sciences (IJRRIS*, II(3).

Budiman, I., Faisal, M. R. & Nugrahadi, D. T., 2020. Studi Ekstraksi Fitur Berbasis Vektor Word2Vec pada Pembentukan Fitur Berdimensi Rendah. *Jurnal Komputasi*, 8(1).

Bustami, B., 2014. Penerapan Algoritma Naive Bayes Untuk Mengklasifikasi Data Nasabah Asuransi. *Jurnal Informatika*, VIII(1).

Chicco, D., 2017. Ten quick tips for machine learning in computational biology. *BioData Mining*, Issue 10.

Deppu, S., Pethuru, R. & S, R., 2016. A Framework for Text Analytics using the Bag of Words (BoW) Model for Prediction. *International Journal of Advanced Networking & Application (IJANA)*, pp. 320-323.

Gharatkar, S., Ingle, A., Naik, T. & Save, A., 2017. Review Preprocessing Using Data Cleaning And Stemming Technique. *International Conference on Innovations in Information Embedded and Communication Systems (ICIIECS).*

Grandini, M., Bagli, E. & Visani, G., 2020. Metrics for Multi-Class Classification: an overview. pp. 1-16.

Gurusamy, V. & Kannan, S., 2014. *Preprocessing Techniques for Text Mining.* s.l.:s.n.

Ladani, D. J. & Desai, N. P., 2016. Stopword Identification and Removal Techniques on TC and IR applications: A Survey. *International Conference on Advanced Computing & Communication Systems (ICACCS)*, Issue 6.

Lauda A, Y. & Timur, I. A., 2018. Klasifikasi Pertanyaan Pada Sistem Tanya Jawab Berbahasa Indonesia Menggunakan Naive Bayes Classifier.

Liddy, E. D., 2001. *Natural Language Processing.* New York: Syracuse University.

Liu, A. Y. & Martin, C. E., 2011. Smoothing Multinomial Naïve Bayes in the Presence of Imbalance. *Machine Learning and Data Mining in Pattern Recognition*, Volume 6871, pp. 46-59.

Manliguez, C., 2016. Generalized Confusion Matrix for Multiple Classes.

Mohasseb, A., Bader-El-Den, M. & Cocea, M., 2018. Question categorization and classification using grammar based approach. *Information Processing & Management*, 54(6), pp. 1228-1243.

Moral, C., de Antonio, A., Imbert, R. & Jaime, R., 2014. A survey of stemming algorithms in information retrieval. *Information Research*, 19(1).

Sarrouti, M. & El Alaoui, S. O., 2017. A Machine Learning-based Method for Question Type Classification in Biomedical Question Answering. *Methods of Information in Medicine*, Volume 56.

Sasikumar, U. & Sindhu, L., 2014. A Survey of Natural Language Question Answering System. *International Journal of Computer Application.*

Van-Tu, N. & Anh-Cuong, L., 2016. Improving Question Classifiaction by Feature Extraction and Selection. *Indian Journal of Science and Technology*, 9(17).

Wei, J., Huang, C., Vosoughi, S. & Wei, J., 2020. What Are People Asking About COVID-19? A Question Classification Dataset. *ACL 2020 Workshop NLP-COVID Submission.*