

Implementasi *Chatbot* pada Telegram sebagai *Monitoring Assistant* dengan Analisis Teks Klasifikasi Menggunakan Metode *Support Vector Machine*

Dwi Lestari¹, Lukman Subekti^{1,*}

¹Departemen Teknik Elektro dan Informatika, Sekolah Vokasi, Universitas Gadjah Mada;
lestdwi1003@mail.ugm.ac.id

*Korespondensi: lukmans@ugm.ac.id;

Abstract – *In the digital era of Industry 4.0, there is a strong push for the integration of the world with internet-based processes as the main platform. A chatbot can be defined as a smart program designed to understand the meaning of human language. To provide answers that match user inquiries, a chatbot needs to simulate and classify the intent behind users' conversations. The objective of this research is to develop intent classification on a chatbot using Natural Language Processing (NLP) and the Support Vector Machine (SVM) algorithm. This research developed a chatbot on the Telegram platform with the Support Vector Machine (SVM) algorithm to measure accuracy levels, making the chatbot more precise. This research project successfully implemented the proposed design. The result of this research is a chatbot named "MONA," which can be used as a Monitoring Assistant for Netmonk customers registered in the system. Additionally, this research successfully conducted training on intent labels and texts, enabling the chatbot to provide interactive responses to user questions. Testing was performed using FastAPI integration to transmit data. The results of this testing showed an accuracy of 98.92% with support vector machine, 93.54% with naïve Bayes, and 95.8% with neural networks, making the support vector machine the preferred method in this application.*

Keywords – *Chatbot, Intent, Text Classification, Support Vector Machine, NLP*

Intisari – Dalam dunia industri 4.0 era digital ini sangat mendorong integrasi dunia dengan proses internet sebagai kedudukan utama. *Chatbot* sendiri dapat diartikan sebagai sebuah program ponsel pintar yang didesain untuk memahami maksud dari bahasa manusia, untuk mendapatkan jawaban yang sesuai dengan yang ditanyakan pengguna, *chatbot* perlu melakukan simulasi dan klasifikasi percakapan maksud atau keinginan (*intent*) dari pengguna. Tujuan dari penelitian ini adalah membuat klasifikasi *intent* pada *chatbot* dengan menggunakan teknologi *Natural Language Processing* dan algoritma *Support Vector Machine* (SVM). Penelitian ini membuat pengembangan *chatbot* pada platform Telegram dengan algoritma *Support Vector Machine* (SVM) untuk mengukur tingkat akurasi, sehingga *chatbot* menjadi lebih akurat. Penelitian ini sudah berhasil mengimplementasikan rancangan tersebut. Hasil dari penelitian yang dibuat ini adalah *chatbot* yang bernama "MONA" dapat digunakan sebagai *Monitoring Assistant* pelanggan *Netmonk* yang sudah terdaftar dalam sistem. Selain itu, penelitian ini juga berhasil melakukan *training* pada label dan teks *intent*, sehingga *chatbot* dapat memberikan jawaban yang interaktif dari pertanyaan pengguna, kemudian pengujian ini menggunakan integrasi *FastAPI* dalam mengirimkan data. Pada pengujian ini diperoleh hasil akurasi 98,92% dengan *support vector machine*, 93,54% dengan *naïve bayes*, dan 95,8% dengan *neural network*, sehingga penelitian ini menggunakan metode *support vector machine* dalam pengaplikasiannya.

Kata kunci – *Chatbot, Intent, Teks Klasifikasi, Support Vector Machine, NLP*

I. PENDAHULUAN

Monitoring Network merupakan suatu metode yang digunakan untuk melakukan pengamatan dan *monitoring* pada sistem jaringan sebuah komputer yang berjalan. *Monitoring network* dapat diterapkan pada sebuah *device* atau server yang terhubung ke *network* [1]. Dalam *monitoring* ini dapat memberikan sebuah informasi mengenai kondisi atau keadaan dari *device* atau server yang dimonitor. Informasi dari *monitoring* tersebut dapat dilakukan analisa kondisi server atau *device*. Tetapi yang menjadi permasalahan umum dalam sebuah perusahaan atau industri adalah fleksibilitas dalam proses *monitoring* apabila terjadi permasalahan human eror.

Dunia industri era digital sekarang semakin meningkatkan kecerdasan buatan dalam mengintegrasikan ke dalam kehidupan sehari-hari. Salah satu yang dimanfaatkan di era digital ini adalah layanan tanya jawab secara *online* melalui *chat* pribadi, dengan adanya layanan *chat* secara *real-time* akan sangat memudahkan kebutuhan para pengguna dalam beraktivitas. *Chatbot* merupakan salah satu contoh dari sistem

kecerdasan buatan dan menjadi salah satu contoh interaksi manusia dengan komputer yang sering dikenal [2].

Chatbot dapat diartikan juga sebagai program komputer yang diatur atau dirancang untuk dapat memahami satu atau lebih dari bahasa manusia dengan menggunakan pemrosesan bahasa alami untuk memudahkan pekerjaan manusia. *Chatbot* juga diartikan sebagai bot pintar, agen interaktif, asisten digital, atau entitas percakapan buatan [3]. Dilihat dari perkembangannya, *chatbot* akan sangat menjanjikan dalam dunia industri terutama dalam menyediakan kebutuhan pengguna dengan cepat dan dapat menanggapi secara khusus pertanyaan pengguna. Ada empat pendekatan kecerdasan buatan (*artificial intelligence/ AI*), yaitu berpikir layaknya manusia (*thinking humanly*), berpikir secara rasional (*thinking rationally*), bertindak layaknya manusia (*acting humanly*) dan bertindak secara rasional (*acting rationally*) [4]. Seiring berjalannya waktu kecerdasan buatan ini akan muncul mesin yang benar-benar mampu berpikir, bertindak, mendengar, melihat, dan berbicara layaknya manusia.

Pemrosesan bahasa alami atau yang disebut *Natural Language Processing* (NLP) merupakan kecerdasan buatan yang bertujuan dalam melakukan proses atau memahami bahasa alami manusia pada komputer [5]. Teknologi *Natural Language Processing* (NLP) ini menjadikan *chatbot* bisa menjadi seperti manusia saat melakukan dialog dengan pengguna. *Chatbot* menjadi salah satu dari model yang di implementasikan oleh teknologi *Natural Language Processing* (NLP) yang banyak dilakukannya penelitian pada beberapa tahun terakhir. Banyak sekali industri yang mengintegrasikan *chatbot* ke dalam proses bisnis untuk mendapatkan suatu manfaat. Penerapan *chatbot* pada proses bisnis ini menjadikan fleksibilitas yang baik, karena dapat digunakan pada berbagai aplikasi. *Chatbot* juga dapat memberikan layanan yang interaktif dan menarik bagi pengguna. Tujuan adanya *chatbot* ini diterapkan yaitu untuk memberikan jawaban yang akurat terhadap pertanyaan-pertanyaan yang di berikan oleh pengguna. Selain itu juga, perusahaan mendapatkan manfaat lain dengan penggunaan *chatbot* ini bisa melihat data yang terkumpul melalui *history* percakapan *chatbot* untuk memahami pengguna dan mengetahui pendapat atau hal menarik yang ditanyakan oleh pengguna.

II. DASAR TEORI

A. Chatbot

Berbagai macam penelitian yang sudah dilakukan untuk *chatbot*, sehingga *chatbot* sendiri memiliki beragam definisi yang berbeda-beda. *Chatbot* adalah sebuah perangkat lunak dengan teknologi bahasa alami yang akan berinteraksi dengan pengguna dalam mencari informasi atau dialog yang berisikan perintah [2]. *Chatbot* di ciptakan bukan semata-mata tidak memiliki fungsi dan tujuan, dengan kata lain fungsi dan tujuan dari adanya *chatbot* untuk menyimulasikan percakapan cerdas dengan seorang manusia. Dialog yang dilakukan untuk mencari informasi akan menyediakan pengguna dengan informasi yang relevan dengan yang di *query* mereka.

Dialog yang berorientasi pada perintah ini dirancang untuk melakukan percakapan dengan hasil yang akan memberikan perintah yang dijalankan. Sehingga dapat disimpulkan bahwa sistem pencarian informasi hanya memberikan informasi mengenai suatu hal sedangkan sistem yang berorientasi perintah akan lebih interaktif dan memperbolehkan pengguna melakukan sebuah kegiatan perintah seperti pemesanan. *Chatbot* akan menganalisis teks yang diterimanya dan memahami apa yang dimaksud dan dibutuhkan oleh pengguna. Jawaban dari *chatbot* sudah tersedia berdasarkan maksud yang telah dipahami oleh sistem.

B. Machine Learning

Machine learning atau biasa dikenal dengan sebutan pembelajaran mesin ini merupakan salah satu ilmu komputer yang dapat bekerja tidak dengan di program secara eksplisit. *Machine learning* adalah kecerdasan buatan (AI) yang dapat mempelajari cara dalam membuat data [6]. *Machine learning* juga dapat diartikan sebagai ilmu yang mempelajari algoritma dan model statistik yang digunakan pada sistem komputer

untuk bisa melakukan pekerjaan tertentu tanpa adanya instruksi yang eksplisit. Cara kerja dari *machine learning* ini sangat bergantung pada pola dan kesimpulan sehingga untuk mendapatkan pola dan kesimpulan tersebut dibutuhkannya algoritma *machine learning* yang menghasilkan model matematika yang didasarkan pada data sampel yang disebut dengan *training data*.

C. Natural Language Processing

Natural Language Processing (NLP) merupakan salah satu metode dari cabang ilmu *Artificial Intelligence* (AI) yang digunakan untuk memproses bahasa alami menjadi bahasa yang dapat dipahami oleh komputer [5]. Bahasa alami sendiri adalah bahasa yang digunakan oleh manusia dalam sehari-hari untuk bertukar informasi satu sama lain. Tujuan dari NLP ini yaitu melakukan proses atau memahami bahasa alami manusia yang dilakukan oleh komputer yang berhubungan pada kemampuan semantik dan sintaksis [7]. Pada dasarnya komputer tidak dapat mengerti bahasa alami yang digunakan oleh manusia, oleh karena itu diperlukannya algoritma yang dapat mengubah teks supaya dapat dimengerti. Pada dasarnya teknik yang digunakan dalam penerapan NLP ini adalah menggunakan pendekatan analisis sintaksis dan analisis semantik. Analisis sintaksis sendiri merupakan analisis bahasa alami dengan memperhatikan aturan dalam penulisan tata bahasa. Kemudian, untuk analisis semantik merupakan analisis bahasa alami dengan memperhatikan makna atau arti dari setiap kata, penggunaan tanda sampai struktur dari teks tersebut. Penelitian dari NLP sendiri memiliki tujuan utama yaitu membuat mesin komputer mampu memahami atau mengerti makna bahasa alami yang kemudian memberikan respons yang sesuai [8].

D. Support Vector Machine

Support Vector Machine (SVM) adalah suatu metode yang handal dalam menyelesaikan masalah klasifikasi data. Dalam penggunaannya model SVM ini dapat mengolah data menjadi data latih dan data uji. Data latih sendiri dapat digunakan dalam membentuk suatu model *Support Vector Machine* (SVM) sedangkan nilai dari parameter bebasnya dipilih dari data awal [9]. Setelah itu, model dari SVM yang dihasilkan ini akan digunakan untuk mengklasifikasi data uji. Algoritma *Support Vector Machine* (SVM) ini juga dapat diartikan sebagai algoritma yang memiliki tujuan untuk menemukan *hyperplane* maksimal, dimana *hyperplane* sendiri merupakan suatu fungsi yang dapat memisahkan antara dua kelas. Selama proses berlangsung, SVM akan memaksimalkan margin atau jarak antar pola pelatihan dan batas keputusan. Margin sendiri dapat diartikan sebagai jarak terpendek dari sebuah *hyperplane* terhadap satu sisi dari margin itu sama dengan jarak *hyperplane* dengan sisi margin lainnya, dengan catatan pada kedua margin tersebut berada pada posisi yang paralel dengan *hyperplane*.

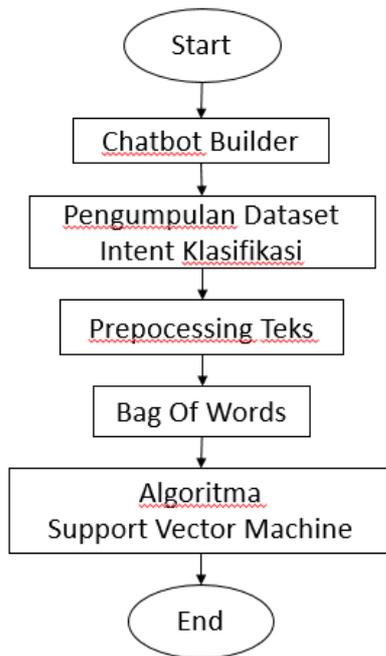
E. Telegram

Telegram merupakan salah satu jenis media untuk komunikasi yang di mana media komunikasi diartikan sebagai sarana dalam mendistribusikan, menyebarkan, dan

menyampaikan informasi [10]. Seiring dengan perkembangan teknologi dan komunikasi maka memunculkan perkembangan terhadap media komunikasi yang digunakan manusia dalam penyampaian pesan dan informasi. Salah satu bentuk perkembangan teknologi komunikasi yaitu media komunikasi *online*. Media *online* dapat diartikan sebagai sarana komunikasi secara *online* atau diakses melalui internet yang terdapat teks, foto, video, dan juga suara. Contoh dari sarana komunikasi *online* seperti email, *mailing list* (*website*, *blog*, whatsapp, line, dan telegram termasuk salah satu kategori media *online*).

III. METODOLOGI

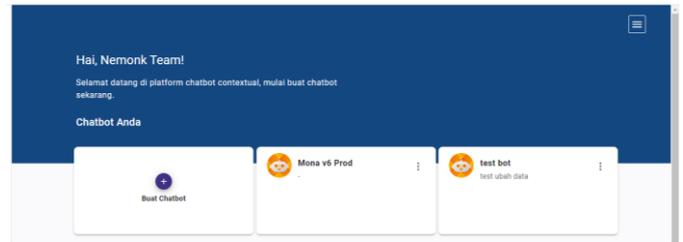
Tahapan penelitian ini dimulai dari membangun *chatbot* pada *chatbot builder* untuk membuat skema dari *chatbot*nya sendiri, kemudian dilakukannya pengumpulan *dataset* pada *intent* klasifikasi sebelum dilakukannya *preprocessing* teks. Setelah data tersebut diolah dan menghasilkan angka perhitungan dari data *preprocessing* tersebut masuklah ke dalam proses *Bag of Words* di mana kalimat atau dokumen dipresentasikan sebagai kantung dari kata-katanya hanya mempertimbangkan pada setiap kata saja dan menghiraukan tata bahasa dari urutan kata. Kemudian baru masuk ke dalam algoritma *support vector machine*. Pada Gambar 1 merupakan alur dari penelitian.



Gambar 1. Flow alur penelitian

A. Chatbot Builder

Pada proses membuat *chatbot* dibutuhkan sebuah *chatbot builder* untuk memudahkan dalam melakukan *development* pada *chatbot*. *Chatbot builder* ini bertujuan untuk mengetahui seberapa efektif dan interaktif *chatbot* dalam memberikan *respond* kepada pengguna. Hasil pengembangan *chatbot builder* ditampilkan pada Gambar 2.



Gambar 2. Tampilan *chatbot builder*

Pada Gambar 2 di atas merupakan tampilan dari *chatbot builder* yang merupakan hasil dari pengembangan secara mandiri yang digunakan sebagai platform untuk membuat, mengelola, dan juga mengintegrasikan *chatbot*. Pada *chatbot builder* ini mendukung integrasi dengan platform sosial media berupa telegram yang diterapkan sebagai sarana komunikasi *chatbot*.

B. Pengumpulan *Dataset Intent* Klasifikasi

Pada tahapan pengumpulan *dataset* ini bertujuan sebagai pengujian pada algoritma yang digunakan untuk mengetahui nilai akurasi terbaik untuk mengukur seberapa baik model dapat memprediksi variasi dalam data. Selain itu juga dapat melatih model di mana *dataset* yang sudah diberi label seperti *greeting*, *server*, dan *network* ini untuk melatih model pembelajaran mesin agar dapat mengenali *intent* dengan akurasi yang tinggi. *Dataset intent* klasifikasi disajikan pada Tabel 1.

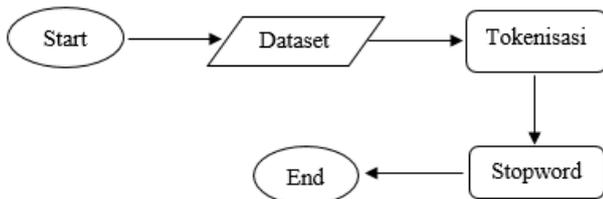
Tabel 1. Contoh *dataset intent* klasifikasi

| Label | Text |
|-----------------|---|
| <i>Greeting</i> | Selamat pagi |
| <i>Greeting</i> | Halo <i>chatbot</i> |
| <i>Greeting</i> | Apakah Anda hadir, Saya butuh bantuan dalam beberapa hal. |
| <i>Greeting</i> | Selamat sore, Bagaimana keadaan Anda hari ini? |
| <i>Server</i> | Bagaimana status server saat ini ? |
| <i>Server</i> | Bagaimana perkembangan terkini dari status server ? |
| <i>Server</i> | Apakah server sedang beroperasi dengan baik |
| <i>Server</i> | Adakah masalah pada server ? |
| <i>Network</i> | Status jaringan saat ini |
| <i>Network</i> | Kondisi perangkat jaringan |
| <i>Network</i> | Bagaimana ketersediaan perangkat jaringan saat ini |

Pada Tabel 1 di atas berisikan mengenai teks dari *intent* label *greeting*, *server*, dan *network* atau jaringan. Pada tabel ini juga berisikan mengenai contoh-contoh dari pertanyaan atau pernyataan yang berkaitan dengan *greeting*, *server* dan juga perangkat jaringan. Dengan adanya *dataset* ini, memudahkan *chatbot* dalam memberikan informasi yang dibutuhkan oleh pengguna.

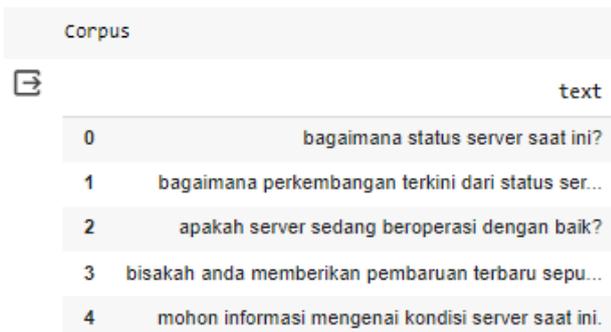
C. Preprocessing Teks

Tahap *preprocessing* ini merupakan tahapan di mana data *input* akan diolah kembali yang kemudian diproses lebih lanjut. Pada tahapan *preprocessing* ini kalimat sebagai masukan yang sudah dikumpulkan kemudian akan diekstrak teksnya untuk melalui beberapa tahapan proses seperti tokenisasi dan *stopwords*. Gambar 3 di bawah ini merupakan alur dari *preprocessing*

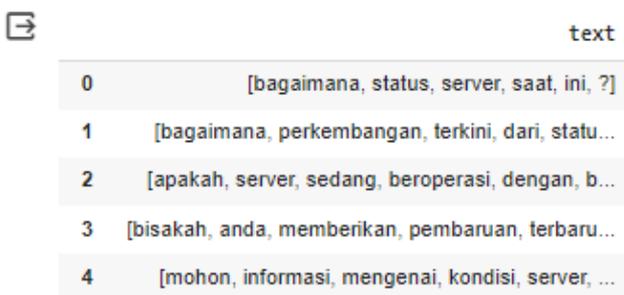


Gambar 3. Flow preprocessing

Pada proses tokenisasi ini merupakan salah satu prosedur dalam pemecahan teks menjadi kata, frasa, atau bagian lainnya yang memiliki makna, yaitu token. Dapat diartikan bahwa tokenisasi merupakan bentuk segmentasi dari teks. Gambar 4 menunjukkan teks sebelum ditokenisasi, sedangkan Gambar 5 menunjukkan teks setelah ditokenisasi.



Gambar 4. Teks sebelum ditokenisasi



Gambar 5. Teks setelah ditokenisasi

Setelah ditokenisasi, data akan melewati proses *Stopwords*. *Stopwords* merupakan kata yang biasa muncul dalam jumlah besar dan juga tidak mempunyai makna yang berarti pada sebuah kalimat. *Stopwords* juga dapat diartikan sebagai kata-kata umum (*common words*) yang biasa muncul dan juga tidak memberikan sebuah informasi penting (diacuhkan). Kata-kata *Stopwords* disajikan pada Tabel 2.

Tabel 2. Contoh kata *stopwords*

| Kata-kata <i>stopwords</i> | | | |
|----------------------------|--------|-------|-----------|
| adalah | adapun | akan | atau |
| bahwa | banyak | bisa | bagaimana |
| cuma | cara | cukup | dan |
| datang | dari | di | hendak |

Pada Tabel 2 merupakan contoh beberapa kata *stopwords* yang kemudian akan dilakukan proses pembuangan *stopwords* dengan tujuan untuk mengetahui suatu kata tersebut merupakan bagian dari *stopwords* atau bukan. Proses pembuangan *stopwords* disajikan pada Tabel 3

Tabel 3. Proses *stopwords removal*

| Jenis | Teks |
|---------------------------------|--|
| Sebelum <i>Stopword Removal</i> | “bagaimana”, “status”, “server”, “saat”, “ini” |
| Setelah <i>Stopword Removal</i> | “bagaimana”, “status”, “server” |

Dapat dilihat pada Tabel 3 terdapat contoh kalimat sebelum di lakukan *stopword removal* dan juga setelah dilakukan *stopword removal*. Beberapa kata yang terdaftar dalam *stopword* tersebut akan hilang secara otomatis setelah dilakukannya *stopword removal*.

D. Bag of Word

Pada tahapan *Bag of Word* (BoW) ini teks, kalimat atau dokumen dipresentasikan sebagai kantung dari kata-katanya dimana hanya mempertimbangkan pada setiap kata saja dan menghiraukan tata bahasa dari urutan kata. Contoh token kata sebelum BoW disajikan pada Tabel 4, sedangkan setelah BoW pada Tabel 5.

Tabel 4. Contoh token kata sebelum BoW

| Token Kata | Kalimat |
|------------|---------------------------------------|
| | Bagaimana status server baik ini saat |

Tabel 5. Contoh token kata setelah BoW

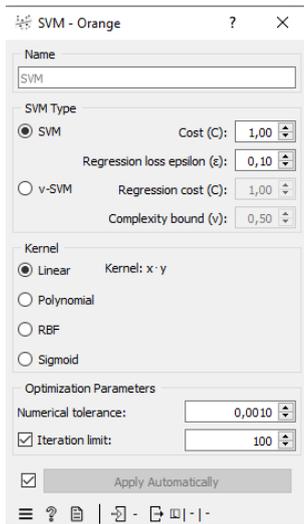
| Token Kata | bagaimana | status | server | baik | ini | saat |
|------------|-----------|--------|--------|------|-----|------|
| BoW | 1 | 1 | 1 | 0 | 1 | 1 |

Pada Tabel 4 di atas ditunjukkan representasi dari kalimat “bagaimana status server saat ini”. Setelah semua kata dalam kalimat di *dataset* sudah diubah ke dalam bentuk *Bag of Words* seperti pada Tabel 5, yaitu berpola angka 0 dan 1 atau bisa juga dengan pola angka 1 dan 2 pada kalimat tersebut akan dijadikan sebagai masukan atau *input* dalam menganalisis dengan algoritma model *support vector machine*.

E. Analisis *Support Vector Machine*

Pada *software* Orange ini dapat menggunakan model *Support Vector Machine* (SVM) dalam melakukan pengujian model data. Pada Gambar 6 di bawah merupakan bentuk dari karakteristik pada model *Support Vector Machine* (SVM)

yang akan digunakan dalam mengetahui nilai parameter terbaik. Pada penelitian ini, dapat memilih tipe dari *Support Vector Machine* nya sendiri yaitu ada SVM dan v-SVM (nu). Pada tipe SVM terdapat *cost* dan *regression loss epsilon* di mana fungsi dari *cost* sendiri sebagai evaluasi seberapa baik model *Support Vector Machine* dalam menangani data *training* dan juga seberapa baik model ini dapat memisahkan kelas yang berbeda.



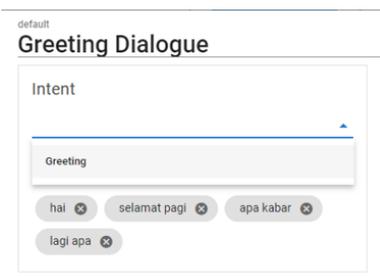
Gambar 6. Karakteristik SVM

IV. HASIL DAN PEMBAHASAN

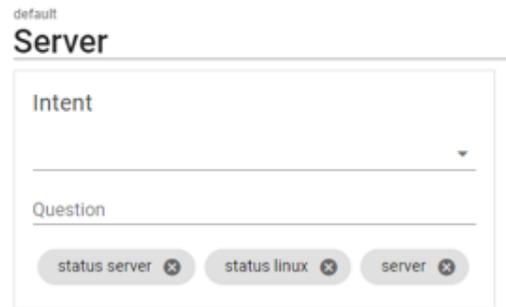
Pada penelitian ini dilakukan dua macam metode pengujian, yang pertama pengujian pada *chatbot builder* dan yang kedua pengujian akurasi pada algoritma yang akan digunakan. Pengujian pada *chatbot builder* ini bertujuan untuk mengetahui seberapa efektif dan interaktif *chatbot* dalam memberikan jawaban kepada pengguna. Sedangkan pada pengujian kedua yaitu pengujian pada algoritma yang digunakan ini bertujuan untuk mengetahui nilai akurasi terbaik untuk mengukur seberapa baik model dapat memprediksi variasi dalam data.

A. Pengujian *Intent* pada *Chatbot Builder*

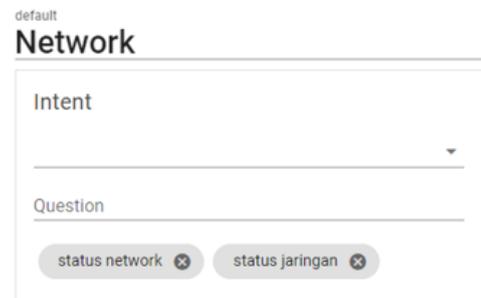
Pengujian *intent* pada *chatbot builder* terdapat *greeting dialogue*, *intent greeting*, dan *intent network Greeting dialogue* disajikan pada Gambar 7, *Intent server* disajikan pada Gambar 8, sedangkan *Intent network* disajikan pada Gambar 9.



Gambar 7. *Intent greeting*



Gambar 8. *Intent server*



Gambar 9. *Intent Network*

Setelah dilakukan masukan satu per satu *question* yang akan digunakan, kemudian *intent* tersebut akan secara otomatis tersimpan dalam sistem *builder*. Seperti pada Gambar 7, Gambar 8, dan Gambar 9 setelah dilakukan penyimpanan, label *intent* dan juga *question* sudah secara langsung tersimpan dalam sistem. Setelah itu akan lebih mudah dalam mengimplementasikan *intent* pada *greeting dialog* pada saat melakukan *builder* di *chatbot*. Selain *question* itu, masih bisa menambahkan *question* lainnya untuk mewakili dari kalimat *intent* lainnya. Hasil sebelum dilakukan *training intent* disajikan pada Gambar 10.



Gambar 10. Hasil sebelum dilakukan *training intent*

Pada gambar 10 terlihat hasil *chatbot* sebelum dilakukan *training*, *chatbot* tidak dapat memahami maksud dari pengguna. Hasil setelah dilakukan *training intent greeting* disajikan pada Gambar 11. Hasil setelah dilakukan *training intent server* disajikan pada Gambar 12. Hasil setelah dilakukan *training intent network* disajikan pada Gambar 13.



Gambar 11. Hasil setelah dilakukan *training intent greeting*



Gambar 12. Hasil setelah dilakukan *training intent server*

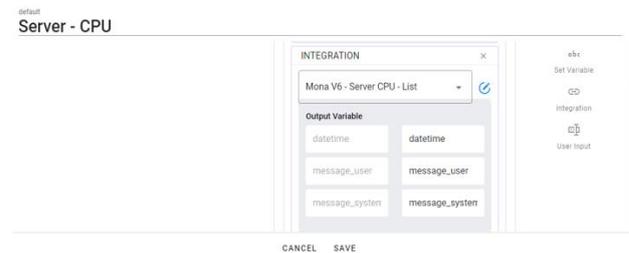


Gambar 13. Hasil setelah dilakukan *training intent network*

Setelah dilakukan *training intent greeting*, *server*, dan *network* seperti pada Gambar 11, Gambar 12, Gambar 13 dapat dilihat bahwa *chatbot* sudah bisa memahami kata-kata dari pengguna sehingga *chatbot* dapat memberikan jawaban yang sesuai.

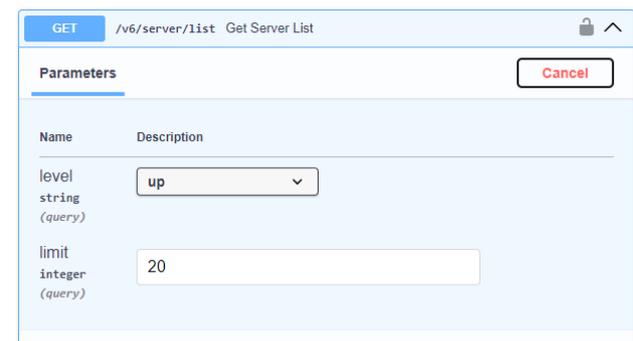
B. Pengujian Integrasi FastAPI

Gambar 14 merupakan integrasi *FastAPI* pada *chatbot*.



Gambar 14. Integrasi *FastAPI* pada *chatbot*

Pada saat melakukan *chatbot builder*, *chatbot* tidak akan bisa memberikan jawaban yang informatif apabila datanya tidak ada. Oleh karena itu, dilakukannya *integration* pada *chatbot*. Pada Gambar 14 merupakan fitur dalam *chatbot builder* dalam melakukan integrasi data dengan menggunakan Integrasi *FastAPI*. Pada *chatbot builder* hanya perlu menentukan id yang digunakan pada integrasi *FastAPI*, sehingga data akan terhubung pada *chatbot builder* sebagai *output*. Parameter *FastAPI* disajikan pada Gambar 15.



Gambar 15. Parameter *FastAPI*

Pada Gambar 15 di atas merupakan bentuk dari *FastAPI* dari *Mona V6 - Server - List*, terdapat parameter yang menjadi salah satu peran penting dari definisi *endpoint API*, selain itu juga dapat membuat *API* yang lebih fleksibel dan dapat menerima berbagai jenis masukan dari pengguna dengan cara yang aman dan terstruktur pastinya. Pada parameter ini terdapat dua parameter yaitu *level* dan *limit*. Parameter *level* dapat digunakan dalam menentukan tingkat atau level detail dari informasi yang diperoleh. Nilai yang terdapat pada parameter *level* ini yaitu “*up*” dan “*down*”. Level “*up*” berarti mengembalikan informasi tentang server yang sedang aktif atau *up*. Kemudian level “*down*” berarti mengembalikan informasi tentang server yang sedang tidak aktif atau *down*.

C. Pengujian Model Dataset

1. Pengujian Nilai Akurasi pada *naive bayes*

Pada pengujian *dataset* dengan model algoritma *naive bayes* ini di dapatkan hasil perhitungan akurasi sebesar 93,5 % ini menunjukkan bahwa tingkat keberhasilan model dalam melakukan prediksi kelas atau label pada *dataset* sebesar 93,5%. Pada hasil nilai tersebut, model

naive bayes berhasil melakukan prediksi dengan tepat sekitar 93,5% dari semua sampel yang ada dalam *dataset*. Hasil nilai akurasi dengan *naive bayes* disajikan pada Gambar 16.

Naive Bayes Accuracy Score -> 93.54838709677419

Gambar 16. Hasil nilai akurasi dengan *naive bayes*

Pada Gambar 16 terlihat proses dalam melatih model klasifikasi *naive bayes* dengan menggunakan data latih dan kemudian dilakukan pengujian performa dengan menggunakan data uji.

2. Pengujian nilai akurasi pada *support vector machine*

Hasil nilai akurasi *support vector machine* disajikan pada Gambar 17.

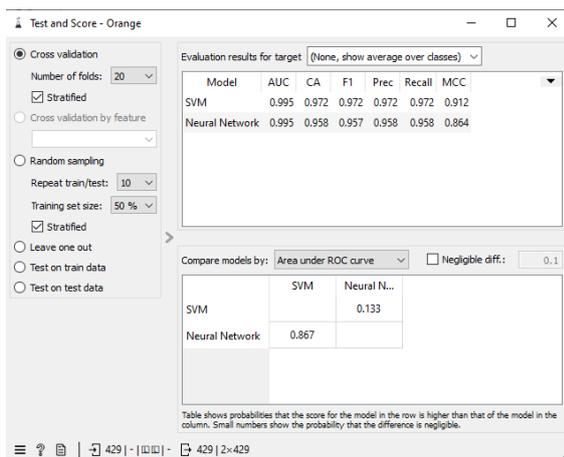
SVM Accuracy Score -> 98.9247311827957

Gambar 17. Hasil nilai akurasi SVM

Pada pengujian nilai akurasi seperti pada Gambar 17 dengan menggunakan klasifikasi algoritma *support vector machine* ini di dapatkan hasil 98,92%. Hasil tersebut menunjukkan tingkat keberhasilan model dalam memprediksi kelas atau label pada sebuah *dataset*. Dengan hasil nilai akurasi sebesar ini, model *support vector machine* berhasil memprediksi dengan tepat sekitar 98,92% dari semua sampel yang ada dalam *dataset*. Akurasi yang tinggi ini menunjukkan kinerja yang baik dari model dalam memahami pola-pola yang ada dalam *dataset* dan juga dalam membuat prediksi yang akurat.

3. Pengujian Model dengan Menggunakan Orange

Pengujian model dengan menggunakan Orange dapat diketahui hasil *test* dan *score* dengan *cross validation*. Hasil *test* dan *score* dengan *cross validation* disajikan pada Gambar 18.



Gambar 18. Hasil *test* dan *score* dengan *cross validation*

Pada pengujian kali ini menggunakan dua buah model algoritma yaitu *support vector machine* dan *neural network*. Kemudian, pada pengujian ini juga menggunakan *dataset* yang berbeda untuk membuktikan

model dan *dataset* mana yang paling baik untuk digunakan. Pada Gambar 18 merupakan hasil dari pengujian dengan menggunakan *cross validation* di Orange. Melalui pengujian ini di dapatkan hasil akurasi pada *neural network* sebesar 0.958.

V. SIMPULAN

Setelah dilakukannya perancangan, pengimplementasian, dan pengujian, penelitian ini dapat diambil kesimpulan, bahwa berhasil melakukan *build* pada *chatbot* dengan menggunakan *chatbot builder* dan berhasil melakukan *training* pada label dan teks *intent*, sehingga *chatbot* dapat memberikan jawaban yang akurat sesuai dengan pertanyaan dari pengguna. Kemudian, berhasil melakukan pengujian terhadap integrasi dengan menggunakan *FastAPI*, sehingga *chatbot* mampu memberikan (*message*) berupa *summary information* kepada pengguna.

Pengujian model dalam mengukur akurasi pada *chatbot* ini di dapatkan hasil yang terbaik adalah dengan menggunakan metode *support vector machine* memperoleh hasil nilai akurasi 98,92%, sedangkan dengan metode *naive bayes* didapatkan hasil akurasi sebesar 93,54%, dan terakhir dengan menggunakan metode *Neural Network* didapatkan hasil akurasi sebesar 95,8 %.

DAFTAR PUSTAKA

- [1] S. Lee, K. Levanti, and H. S. Kim, "Network monitoring: Present and future," *Comput. Netw.*, vol. 65, pp. 84–98, Jun. 2014, doi: 10.1016/j.comnet.2014.03.007.
- [2] T. P. Nagarhalli, V. Vaze, and N. K. Rana, "A Review of Current Trends in the Development of Chatbot Systems," in *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Coimbatore, India: IEEE, Mar. 2020, pp. 706–710. doi: 10.1109/ICACCS48705.2020.9074420.
- [3] E. Adamopoulou and L. Moussiades, "An overview of chatbot technology," *IFIP Advances in Information and Communication Technology*, pp. 373–383, Jun. 2020. doi:10.1007/978-3-030-49186-4_31
- [4] S. J. Russell and P. Norvig, *Artificial Intelligence*. Upper Saddle River, N.J, New Jersey: Pearson Education, 2009.
- [5] M. Zhou, N. Duan, S. Liu, and H.-Y. Shum, "Progress in Neural NLP: Modeling, Learning, and Reasoning," *Engineering*, vol. 6, no. 3, pp. 275–290, Mar. 2020, doi: 10.1016/j.eng.2019.12.014.
- [6] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, Jul. 2015, doi: 10.1126/science.aaa8415.
- [7] C. O. Bilah, "Deteksi Intent pada Teks Bahasa Indonesia Menggunakan Bidirectional Long Short-Term Memory," thesis, 2023
- [8] Y. Kang, Z. Cai, C.-W. Tan, Q. Huang, and H. Liu, "Natural language processing (NLP) in management research: A literature review," *J. Manag. Anal.*, vol. 7, no. 2, pp. 139–172, Apr. 2020, doi: 10.1080/23270012.2020.1756939.
- [9] S. Yue, P. Li, and P. Hao, "SVM classification: Its contents and challenges," *Appl. Math. - J. Chin. Univ.*, vol. 18, no. 3, pp. 332–342, Sep. 2003, doi: 10.1007/s11766-003-0059-5.
- [10] F. Fitriansyah and Aryadillah, "Penggunaan Telegram Sebagai Media Komunikasi Dalam Pembelajaran Online," *Cakrawala*, vol. 20, Sep. 2020. doi:10.31294/jc.v20i2.8935