
Implementasi Natural Language Processing Dalam Pembuatan Chatbot Pada Program Information Technology Universitas Surabaya

Vincentius Riandaru Prasetyo^{1*}, Njoto Benarkah², Vioni Jannet Chrisintha³

^{1,2,3}Program Studi Teknik Informatika, Universitas Surabaya, Surabaya, Jawa Timur
Email: ^{1*}vincent@staff.ubaya.ac.id, ²benarkah@staff.ubaya.ac.id, ³s160416034@student.ubaya.ac.id

(Naskah masuk: 31 Mei 2021, direvisi: 15 Jun 2021, diterima: 17 Jun 2021)

Abstrak

Program *Information Technology* di Jurusan Teknik Informatika, Universitas Surabaya, merupakan salah satu program yang menggunakan bahasa Inggris sebagai pengantar pada saat perkuliahan berlangsung. Akan tetapi, kurangnya informasi mengenai Program *Information Technology* menyebabkan kurangnya minat calon mahasiswa terhadap program tersebut. Oleh karena itu, penelitian ini bertujuan untuk membuat sebuah aplikasi *chatbot* yang dapat membantu *user* untuk memperoleh informasi-informasi terkait dengan Program *Information Technology* pada Jurusan Teknik Informatika, Universitas Surabaya. *Chatbot* yang dibangun hanya akan memproses pertanyaan dengan bahasa Inggris saja. *Chatbot* yang dibuat pada penelitian ini menggunakan pendekatan *Natural Language Processing* (NLP) untuk memproses pertanyaan yang disampaikan *user* dan untuk mendapatkan kata kunci dari informasi yang diinginkan *user*. Sistem akan melakukan pencarian informasi pada kamus informasi yang ada. Apabila informasi tidak ditemukan, maka sistem akan melakukan proses *crawling* untuk memperoleh informasi yang dibutuhkan *user*. Pada penelitian ini, validasi sistem dilakukan dengan dua metode yaitu *cross validation* dan *user validation*. Berdasarkan validasi dengan metode *cross validation* didapatkan akurasi sebesar 83,33%. *User validation* dilakukan dengan cara meminta 10 *user* untuk melakukan uji coba sistem dan didapatkan akurasi sebesar 76%.

Kata Kunci: *Chatbot, Natural Language Processing, Crawling.*

Implementation of Natural Language Processing in Creating Chatbots at Information Technology Program, University of Surabaya

Abstract

The Information Technology Program at the Department of Informatics Engineering, University of Surabaya, is one of the programs that uses English as a medium during lectures. However, the lack of information about the Information Technology Program causes a lack of interest among prospective students. Therefore, this study aims to create a chatbot application that can help users to obtain information related to the Information Technology Program at the Department of Informatics, University of Surabaya. The chatbot that is built will only process questions in English only. The chatbot made in this study uses a natural language processing (NLP) approach to process user-submitted questions and to get keywords from the information that the user wants to know about. The system will search for information in the existing information dictionary. If the information is not found, the system will perform a crawling process to obtain the information needed by the user. In this study, system validation was carried out by two methods, namely cross validation and user validation. Based on the validation with the cross validation method, the accuracy is 83.33%. User validation is done by asking 10 users to test the system and get an accuracy of 76%.

Keywords: *Chatbot, Natural Language Processing, Crawling.*

I. PENDAHULUAN

Sejalan dengan perkembangan zaman, teknologi telah mengalami berbagai perkembangan pesat. Salah satunya adalah informasi dapat diperoleh dengan lebih mudah. Kecerdasan buatan merupakan salah satu bentuk teknologi yang mengalami perkembangan sangat pesat di era modern ini. Kecerdasan buatan memungkinkan mesin berpikir dan membuat keputusan sendiri, salah satunya adalah teknologi *chatbot*. *Chatbot* atau percakapan dengan robot adalah aplikasi kecerdasan buatan yang dapat mensimulasikan percakapan cerdas antar manusia berdasarkan pengetahuan yang diberikan. *Chatbot* adalah agen cerdas yang dapat meniru kemampuan manusia untuk berkomunikasi melalui pesan teks [1]. Keberadaan *chatbot* telah banyak digunakan di perusahaan seperti pada bank untuk membantu menjawab pertanyaan tentang informasi terkait perbankan dan keluhan pelanggan.

Chatbot pada penelitian ini dirancang untuk membantu dalam menjawab pertanyaan yang terkait dengan Program *Information Technology*, Jurusan Teknik Informatika, Universitas Surabaya. Informasi terkini tentang Program *Information Technology* hanya dapat diakses di halaman <http://teknik.ubaya.ac.id/id/program/information-technology-international-dual-degree.html>, akan tetapi informasi yang diberikan pada halaman ini masih sangat terbatas. Keterbatasan informasi pada halaman tersebut mengakibatkan kurangnya minat dari calon mahasiswa untuk mendaftar pada Program *Information Technology*, Universitas Surabaya. Keterbatasan informasi tersebut sebenarnya dapat diatasi dengan bertanya langsung ke Direktorat *Marketing and Public Relations* (MPR) yang dimiliki Universitas Surabaya melalui telepon, email, ataupun *livechatting*. Akan tetapi, keterbatasan staf yang ada di MPR menyebabkan masalah baru yaitu *human delay*, apabila banyak calon pendaftar ingin bertanya dalam waktu yang bersamaan. Selain itu, adanya *office hour* mengakibatkan calon pendaftar tidak memiliki keleluasaan waktu dalam bertanya. Oleh karena itu, *chatbot* dipilih karena dapat menggantikan peran manusia untuk memberikan jawaban dengan cepat kepada pengguna tanpa harus membaca *skimming* atau *scanning* pada sumber data informasi. Pada penelitian ini, *chatbot* dibangun dengan menggunakan pendekatan *Natural Language Processing*.

Natural Language Processing atau sering disingkat NLP adalah salah satu bidang ilmu komputer, kecerdasan buatan dan bahasa (linguistik) yang berkaitan dengan interaksi antara komputer dengan bahasa alami manusia. Bahasa alami adalah bahasa yang dapat dipahami manusia. Pada prinsipnya, bahasa alami adalah suatu bentuk informasi yang akan disampaikan dari satu pengguna ke pengguna lainnya. Bahasa alami dapat direpresentasikan dalam bentuk suara atau teks [2].

Penelitian yang berkaitan dengan perancangan aplikasi *chatbot* sudah pernah dilakukan sebelumnya. Suryani dan Amalia [3] pada penelitiannya membangun aplikasi *chatbot* yang bertujuan untuk memberikan informasi tentang objek wisata yang ada di Jawa Timur. Pada penelitian tersebut, *chatbot* dibangun menggunakan konsep *Artificial Intelligence Markup Language* (AIML). AIML adalah salah satu turunan dari XML (*Extensible Markup Language*) di mana terdapat

kumpulan pola dan respon yang dapat digunakan untuk pencarian jawaban yang diinginkan pengguna *chatbot*. Berbeda dengan penelitian sebelumnya, Benedictus, et.al. [4] mengembangkan *chatbot* yang berfungsi sebagai *helpdesk* untuk melayani tanya-jawab seputar Sistem Informasi Terpadu yang dimiliki oleh Universitas Sam Ratulangi. *Chatbot* yang dikembangkan pada penelitian tersebut menggunakan algoritma *bigram* untuk mencocokkan pola kalimat tanya yang diberikan oleh pengguna. Algoritma ini akan membagi suatu kalimat secara berpasang-pasangan. Metode *forward chaining* juga digunakan pada penelitian tersebut untuk melakukan penalaran terhadap kondisi-kondisi tertentu pada sebuah pertanyaan, sehingga sistem dapat menentukan jawaban yang sesuai. Penggunaan metode *forward chaining* juga digunakan oleh Dwi R, et.al. [5] pada penelitiannya tentang pengembangan aplikasi *chatbot* yang terintegrasi dengan web CMS pada UKM MINSU. *Chatbot* yang dibangun pada penelitian tersebut bertujuan untuk menyelesaikan permasalahan pada pelayanan *customer service* UKM MINSU yaitu kurangnya respon cepat *customer service* karena harus melayani banyak pelanggan, sehingga dapat menyebabkan hilangnya pelanggan tetap dan berpengaruh terhadap omset penjualan.

Penelitian lainnya yang masih berhubungan dengan *chatbot* juga dilakukan oleh Amalia dan Wibowo [6] untuk peningkatan performa bisnis. Pengembangan *chatbot* dilakukan dengan memanfaatkan *Chatfuel* sebagai *bot builder* karena memiliki *response time* yang cepat dalam menjawab pertanyaan dari banyak pengguna sekaligus. Selain itu, penelitian tersebut juga menggunakan *Facebook Messenger* sebagai *platform chatting* di mana *chatbot* akan diintegrasikan ke dalamnya. Berbeda dengan penelitian sebelumnya, Yuniar dan Purnomo [7] membangun sebuah *chatbot* dengan nama ALITTA yang bertugas sebagai asisten virtual dan pusat informasi pada aplikasi BALITTAS. Dalam membangun *chatbot* ALITTA, Yuniar dan Purnomo memanfaatkan *platform API.AI* yang menyediakan layanan NLP dan NLU (*Natural Language Understanding*). NLU merupakan turunan ilmu dari NLP di mana fungsinya adalah untuk melakukan analisis semantik, sehingga makna dari suatu kalimat dapat dipahami oleh aplikasi *chatbot*.

II. METODOLOGI PENELITIAN

A. *Chatbot*

Chatbot adalah sebuah aplikasi yang memungkinkan percakapan antara mesin dengan manusia menggunakan bahasa alami manusia. *Chatbot* sering digambarkan sebagai salah satu cara untuk mensimulasikan interaksi antara mesin dengan manusia. Untuk memahami bahasa alami manusia, *chatbot* menggunakan pendekatan NLP untuk mempelajari dan memproses setiap kata yang diucapkan oleh manusia berupa pesan teks [8].

Cara *chatbot* bekerja dimulai dengan menerima input dari pengguna berupa pesan teks. Sistem kemudian menggunakan NLP untuk memproses input untuk menganalisis, mengidentifikasi, dan menafsirkan makna yang dimaksud oleh

pengguna. Sistem akan memverifikasi input makna dan kondisi dari percakapan yang sedang berlangsung. Setelah sistem memahami artinya, sistem akan mencari data yang dianggap sesuai, kemudian respon berupa jawaban yang ditampilkan kepada pengguna berdasarkan struktur dan bahasa manusia.

B. Natural Language Processing

Natural Language Processing (NLP) adalah kombinasi dari ilmu komputer dan bidang kecerdasan buatan yang terkait dengan linguistik. NLP berkaitan dengan bagaimana mesin memahami bahasa manusia untuk saling berinteraksi. Dengan adanya NLP, komputer dapat belajar dan memahami bahasa manusia, sehingga komputer dapat berkomunikasi dengan manusia. Bahasa manusia adalah hal yang unik karena dibuat khusus untuk menyampaikan suatu makna. Untuk membuat komputer dapat mengerti bahasa manusia adalah tugas yang sulit, karena bahasa manusia memiliki struktur yang kompleks. Selain itu, setiap bahasa memiliki keunikannya sendiri dan mungkin memiliki makna ganda [9]. Sebagai contoh dapat dilihat dari kalimat berikut, “*Look at the dog with one eye*”, di mana kalimat tersebut dapat memiliki arti “melihat anjing dengan satu mata” atau “melihat anjing yang mempunyai mata satu”.

Dua teknik utama untuk memahami NLP adalah *syntactic analysis* (analisis sintaksis) dan *semantic analysis* (analisis semantik). Kedua teknik digunakan untuk memverifikasi struktur bahasa. Analisis sintaksis mengacu pada tata bahasa, sedangkan analisis semantik merujuk pada penafsiran suatu kalimat.

Syntactic analysis (analisis sintaksis) adalah teknik pengaturan pada suatu kalimat sehingga kalimat memiliki tata bahasa yang benar. Analisis sintaksis melibatkan penentuan struktur kalimat seperti subjek, predikat, kata benda, kata kerja, kata ganti, dan sebagainya. Sistem akan dapat membaca input kalimat, yang akan dipecah menjadi kata-kata, dan pada akhirnya menghasilkan deskripsi yang terstruktur. Teknik ini dapat digunakan untuk menyederhanakan kalimat untuk memudahkan pencarian informasi. Selain itu, penggunaan analisis sintaksis juga dapat membantu mendeteksi keberadaan kata atau kalimat baru atau tidak biasa [10].

Sebuah kalimat dapat disebut sebagai kalimat, apabila paling tidak terdiri dari subjek dan predikat, sebagai contoh kalimat “*Andi eat*”. Dengan menggunakan teknik analisis sintaksis, komputer dapat membedakan mana yang termasuk subjek (“*Andi*”) dan predikat (“*eat*”). Kalimat yang terbentuk mungkin saja tidak memiliki makna apapun, karena analisis sintaksis hanya memastikan bahwa struktur dari sebuah kalimat sudah benar.

Analisis semantik adalah teknik yang digunakan untuk memahami makna dan interpretasi dari struktur bahasa. Seseorang bisa memahami perkataan orang lain berdasarkan intuisi dan pengetahuan dari bahasa itu sendiri. Komputer tidak memiliki intuisi dan pengetahuan semacam ini, sehingga mereka membutuhkan metode lain, yaitu semantik. Semantik adalah proses penting karena output semantik yang diharapkan adalah makna yang terkandung pada sebuah input [10]. Analisis semantik memproses teks untuk mengidentifikasi dan

memahami topik yang dimaksud. Semantik juga mempelajari hubungan antara berbagai konsep dalam teks. Sebagai contoh, apabila sebuah teks terdapat kata “*money*” dan “*accounting*”, maka topik yang sedang dibahas berkaitan dengan “*economy*”.

C. Keyword Extraction

Keyword extraction adalah teknik dalam NLP yang menganalisis teks dengan menggunakan *keyword* atau kata kunci yang terkandung dalam teks tersebut. Dengan menggunakan metode ini dapat membantu meringkas teks dan mengidentifikasi subjek teks. Proses yang dilakukan dalam *keyword extraction* adalah tokenisasi kata, penghapusan *stopword*, dan analisis. Tokenisasi adalah teknik pemotongan kalimat menjadi kata-kata. Penghapusan *stopword* adalah teknik untuk menghilangkan kata umum yang terkandung dalam kalimat, seperti “*and*”, “*or*”, “*the*”, dan sebagainya. Tanda baca dan angka juga merupakan bagian dari *stopword*. Analisis dilakukan dalam bentuk *Part-Of-Speech* (POS) *tagging*, di mana setiap kata diberikan label atau tanda sesuai dengan jenis kata, seperti kata kerja, kata benda, kata ganti, dan sebagainya. Pengambilan kata penting hanya menggunakan beberapa jenis kata, seperti kata kerja dan kata benda, karena kalimat lengkap paling tidak terdiri dari subjek dan predikat [11].

Sebagai contoh, terdapat sebuah kalimat tanya yaitu “*where is Ubaya?*”. Proses *keyword extraction* diawali dengan memenggal kalimat tersebut sehingga menjadi “*where*”, “*is*”, “*Ubaya*”, “*?*”. Setelah itu, sistem akan melakukan penghapusan *stopword* yaitu kata “*is*” dan “*?*”. Sistem akan memberikan POS *tag* untuk setiap kata yang tersisa, sehingga kata “*where*” memiliki *tag adverb* (ADV) dan “*Ubaya*” memiliki *tag pronoun* (PRON). Pengambilan kata penting hanya mengambil jenis kata *verb*, *noun*, dan *pronoun*, maka sistem akan mengambil kata “*Ubaya*” sebagai *keyword* dari kalimat tanya tersebut. Selain menggunakan metode POS *tagging*, metode lain yang dapat digunakan untuk melakukan *keyword extraction* adalah *term weighting* atau pembobotan *term*.

D. Term weighting

Term weighting atau pembobotan *term* dilakukan berdasarkan hubungan antara kata dan dokumen serta frekuensi kemunculannya. Metode pembobotan yang digunakan adalah TF-IDF (*Term Frequency - Inverse Document Frequency*). *Term Frequency* menghitung jumlah kata yang muncul pada dokumen yang ada. *Inverse Document Frequency* menganggap istilah yang jarang muncul dalam dokumen lebih penting daripada yang sering muncul. Oleh karena itu, semakin sedikit kata yang muncul dalam dokumen, semakin penting kata tersebut dan semakin tinggi nilai IDF [12]. Pembobotan *term* dengan TF-IDF dihitung berdasarkan Persamaan 1.

$$tf.idf = tf \times idf \quad (1)$$

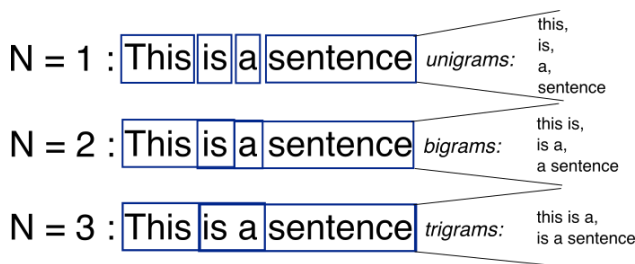
Di mana *tf* adalah *term frequency*, yaitu kemunculan suatu *term/token* pada dokumen tertentu. Sedangkan *idf* adalah *inverse document frequency*, yaitu nilai log basis 10 dari

jumlah N dokumen dibagi nilai df . Variabel df adalah frekuensi dokumen yang merupakan jumlah dokumen yang memiliki $term$ tertentu [13]. Oleh karena itu, IDF dapat dihitung berdasarkan Persamaan 2.

$$idf = \log \frac{N}{df} \tag{2}$$

E. *N-Grams*

N-Grams adalah sebuah metode yang digunakan untuk memecah sebuah kalimat menjadi beberapa bagian, di mana masing-masing bagian terdiri dari N kata. Jumlah N dalam *N-Gram* didasarkan pada jumlah kata yang terkandung dalam kalimat [14]. Ketika jumlah kata dalam kalimat adalah X , jumlah maksimum N yang tersedia untuk *N-Gram* adalah X , dan jumlah kombinasi yang diperoleh setelah mengeksekusi *N-Gram* adalah $X-(N-1)$. *N-Grams* yang sering digunakan adalah *unigrams* ($N=1$), *bigrams* ($N=2$), dan *trigrams* ($N=3$). Ilustrasi untuk metode *N-Grams* dapat dilihat pada Gambar 1 di bawah ini.



Gambar 1. Ilustrasi Metode *N-Grams*

N-grams sering dimanfaatkan pada *language processing*, seperti *spelling correction* (perbaikan ejaan), *word breaking* (pemecahan kata), *predict upcoming words* (prediksi untuk kata selanjutnya), dan *text summarization* (peringkasan teks). Pada penelitian ini, *N-Grams* digunakan untuk mendapatkan alternatif jawaban berdasarkan kata kunci.

F. *Crawling*

Crawling adalah metode pengambilan data yang umumnya berasal dari suatu halaman *web*. Program yang mengimplementasikan metode *crawling* disebut *web crawler* [15]. Proses *crawling* diawali dengan mengakses suatu halaman *web* tertentu, kemudian *server* akan mengirim respon berdasarkan *request* yang diberikan. Respon yang diterima akan berisi kode status, tipe konten, *character encoding*, dan konten *web* itu sendiri. Setelah itu, *crawler* akan membaca respon ini dan mengambil konten yang dibutuhkan.

Crawling dapat dimanfaatkan untuk berbagai kebutuhan, seperti penggunaan terkait mesin pencari (*search engine*) atau pencarian data untuk analisis statistik. Pada penelitian ini, *crawling* digunakan untuk mencari data atau informasi yang berhubungan dengan program *Information Technology*.

III. HASIL DAN PEMBAHASAN

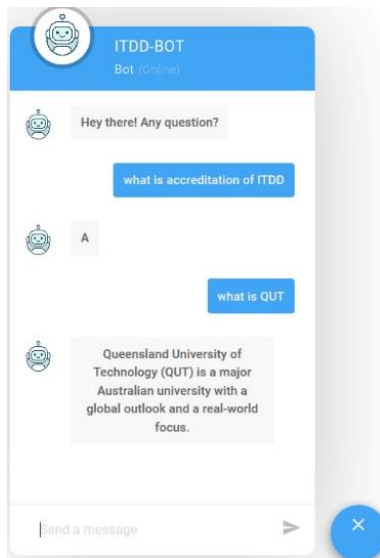
A. Gambaran Kerja Sistem

Proses kerja *chatbot* diawali dengan pengguna menginputkan pertanyaan pada *user interface* yang disediakan. Sistem akan berusaha mengidentifikasi kata tanya yang terdapat pada pertanyaan yang diinputkan *user*. Pertanyaan tersebut akan mengalami tokenisasi terlebih dahulu sebelum dilakukan identifikasi kata tanya. Proses tokenisasi dilakukan dengan memanfaatkan *library* NLTK. Setelah proses tokenisasi, sistem akan melakukan pengecekan pada daftar kata tanya yang sudah disiapkan sebelumnya. Apabila tidak ditemukan kata tanya pada pertanyaan yang diinputkan sebelumnya, maka sistem akan mengirim pesan kepada *user* bahwa format pertanyaan tidak sesuai, sehingga *user* dapat memperbaiki pertanyaan yang diinputkan. Sedangkan apabila kata tanya ditemukan, maka sistem akan melanjutkan ke proses berikutnya yaitu *keyword extraction*.

Proses *keyword extraction* diawali dengan penghapusan *stopword* yang terdapat dalam kalimat tanya yang diinputkan oleh pengguna. Proses penghapusan *stopword* ini juga akan memanfaatkan *library* NLTK, karena *library* tersebut sudah menyediakan daftar *stopword* dalam bahasa Inggris yang siap untuk digunakan. Sistem kemudian akan mencari kata kunci dengan pada kalimat tersebut dengan memanfaatkan *library* *spaCY*.

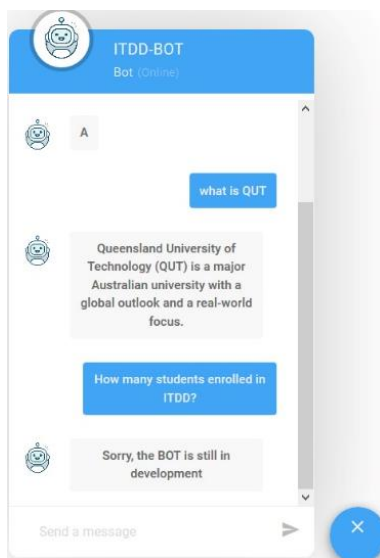
Setelah kata tanya dan kata kunci diperoleh, sistem akan mencari jawaban yang sesuai pada *database* yang ada. Terdapat 30 kombinasi kata tanya, kata kunci, beserta jawabannya pada *database* yang telah disediakan, dimana hal tersebut didapatkan dari hasil wawancara dengan Ketua Jurusan Teknik Informatika, Universitas Surabaya dan perwakilan dari Direktorat MPR, Universitas Surabaya. Apabila pencarian jawaban ditemukan pada *database*, maka sistem akan menampilkan jawaban tersebut kepada *user*. Sedangkan apabila pencarian jawaban tidak ditemukan, maka sistem akan mencari sinonim dari kata kunci dengan memanfaatkan kamus *thesaurus*. Setelah itu, sistem akan mengkombinasikan sinonim yang didapatkan dengan kata tanya. Hasil kombinasi-kombinasi tersebut akan digunakan untuk proses pencarian ulang jawaban pada *database*. Apabila jawaban masih tidak ditemukan, maka sistem akan melakukan *crawling* ke halaman-halaman *web* terkait untuk mencari jawaban yang sesuai.

Proses *crawling* diawali dengan menggunakan kombinasi awal dari kata tanya dan kata kunci hasil *keyword extraction*. Apabila hasil *crawling* awal ini tidak memberikan hasil jawaban, maka akan dilakukan lagi proses *crawling* dengan menggunakan hasil kombinasi-kombinasi sinonim kata kunci dengan kata tanya. Jawaban yang ditemukan berdasarkan hasil *crawling* akan disimpan ke dalam *database*, beserta kombinasi kata tanya dan kata kunci yang digunakan. Hal ini dimaksudkan agar ke depannya proses pencarian jawaban menjadi lebih cepat, apabila ada *user* lain yang menanyakan pertanyaan serupa, karena sistem tidak perlu melakukan proses *crawling* lagi. Sistem akan memberikan pesan kepada *user* untuk menanyakan hal lain, apabila setelah proses *crawling* masih tidak ditemukan jawaban yang cocok. Alur proses kerja



Gambar 5. Uji Coba Sistem 2

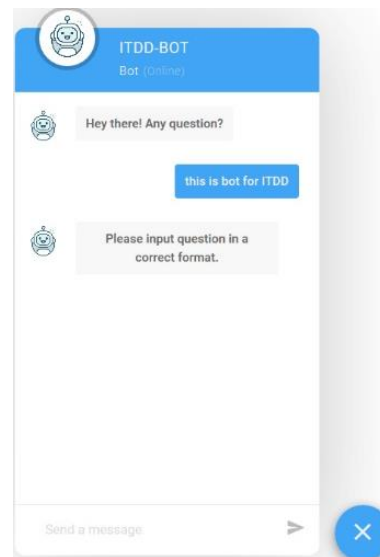
Pada percobaan ketiga, *chatbot* diberikan pertanyaan yaitu “How many students enrolled in ITDD”, di mana pertanyaan dan jawaban yang terkait tidak tersedia di *database* maupun dari hasil *crawling*. Untuk percobaan ketiga ini, *chatbot* akan menampilkan pesan bahwa *chatbot* masih dalam proses pengembangan, seperti yang dapat dilihat pada Gambar 6. Proses pencarian jawaban membutuhkan waktu yang cukup lama yaitu 83,12 detik. Hal ini dikarenakan sistem mencoba untuk mencari jawaban yang sesuai dari berbagai macam sumber yang ada. Jawaban dari pertanyaan tersebut sebenarnya dapat diperoleh pada Sistem Akademik Universitas Surabaya. Akan tetapi, dikarenakan sistem akademik tersebut berisi data-data yang penting dan rahasia, maka *chatbot* tidak diperkenankan untuk melakukan *crawling* data di dalamnya.



Gambar 6. Uji Coba Sistem 3

Uji coba terakhir dilakukan dengan menginputkan pertanyaan dengan format yang salah, yaitu “this is bot for

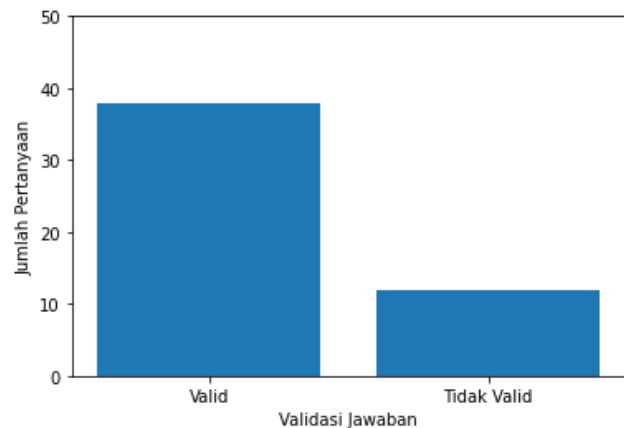
ITDD?”. Sistem tidak menganggap kalimat tersebut sebagai pertanyaan yang valid karena tidak mengandung kata tanya seperti *what, who, where, when, why, dan how*. Oleh karena itu, *chatbot* akan memberikan pesan ke pengguna untuk menginputkan pertanyaan dengan format yang sesuai, seperti yang ditunjukkan pada Gambar 7. Proses ini berlangsung relatif cepat yaitu 5,45 detik karena sistem tidak melakukan proses pencarian jawaban, baik pencarian di *database* maupun dengan proses *crawling*.



Gambar 7. Uji Coba Sistem 4

C. User Validation

User validation adalah salah satu metode validasi, di mana *user* akan mencoba langsung sistem yang telah dibangun dan kemudian menilai apakah *output* yang diberikan oleh sistem sudah sesuai dengan kebutuhan *user*. Pada penelitian ini, validasi dilakukan oleh 10 *user*, di mana setiap *user* akan menginputkan 5 pertanyaan untuk menguji *chatbot* yang telah dibuat. Gambar 8 menunjukkan bahwa akurasi dari sistem *chatbot* yang dibangun yaitu 76%, berdasarkan *user validation* yang dilakukan. Akurasi sistem dihitung dengan cara, jumlah validasi benar dibagi total pertanyaan dan dikali 100%, sehingga akurasi = $(38/50) \times 100\% = 76\%$.



Gambar 8. Hasil User Validation

Berdasarkan hasil *user validation* yang dilakukan, masih ditemukan 12 pertanyaan yang menghasilkan jawaban yang tidak sesuai dengan kebutuhan *user*. Hal ini disebabkan oleh keterbatasan sumber data yang dimiliki, baik yang ada di *database*, maupun sumber-sumber lain saat sistem melakukan *crawling*. Pada hasil *user validation*, masih ditemukan juga pertanyaan-pertanyaan yang memiliki makna sama, tetapi jawaban yang dihasilkan berbeda, sehingga hasil validasi menjadi tidak sesuai. Sebagai contoh, pertanyaan “Who is the head of ITDD program?” dan “Who is the head department of ITDD?”. Pada pertanyaan pertama, sistem memberikan *output* jawaban “Religion Ethics English”, sedangkan untuk pertanyaan kedua menghasilkan jawaban “Budi Hartanto”.

Pada dua pertanyaan tersebut, hanya pertanyaan kedua yang menghasilkan jawaban yang sesuai. Hal ini dikarenakan ada kata kunci yang dihasilkan berbeda. Pertanyaan pertama menghasilkan kata kunci “head”, “ITDD”, dan “program”, sedangkan pertanyaan kedua menghasilkan kata kunci “head”, “department”, dan “ITDD”. Perbedaan tersebut terletak pada kata kunci “program” dan “department”, di mana berdasarkan kamus *thesaurus* yang digunakan pada sistem, kedua kata tersebut tidak saling bersinonim satu sama lain.

Selain alasan yang disampaikan di atas, beberapa pertanyaan masih menghasilkan jawaban yang tidak sesuai, karena kombinasi kata kunci yang dihasilkan dari suatu pertanyaan sangat banyak dan beragam. Seperti yang sudah dijelaskan pada sub bab III.A, apabila pencarian jawaban tidak ditemukan dengan kata kunci awal, maka sistem akan mencari sinonim kata kunci tersebut dan melakukan pencarian ulang. Sebagai contoh terdapat pertanyaan berikut, “Where will we live when we’re abroad?”. Pertanyaan tersebut menghasilkan kata kunci “live” dan “abroad”. Kata “live” mempunyai 20 sinonim, sedangkan “abroad” memiliki 3 sinonim, dan apabila sinonim-sinonim tersebut dikombinasikan, maka akan ada 60 kombinasi kata kunci yang bisa digunakan untuk pencarian ulang jawaban. Hal ini menyebabkan kombinasi jawaban yang didapatkan menjadi beragam dan tidak sesuai dengan yang diinginkan oleh *user*.

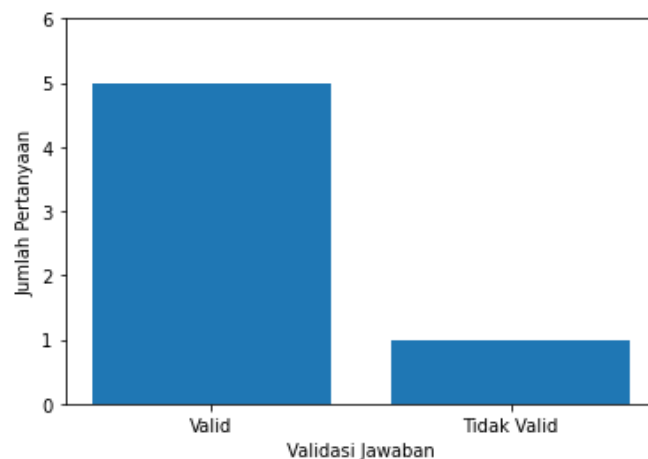
Penyebab lainnya sehingga hasil *user validation* tidak optimal yaitu ada pertanyaan yang sebenarnya sudah memberikan jawaban yang benar, tetapi tidak sesuai dengan kebutuhan *user*. Hal ini dapat dilihat pada pertanyaan berikut “How to apply ITDD”, di mana jawaban dari pertanyaan tersebut adalah “http://daftar.ubaya.ac.id. Click on this link for more information”. Pada kasus tersebut, *user* menginginkan agar semua informasi-informasi yang terdapat pada link <http://daftar.ubaya.ac.id> juga ditampilkan di *chatbot*, sehingga *user* tidak perlu lagi untuk melakukan *scanning* dan *skimming* pada link tersebut untuk memperoleh informasi yang lengkap.

D. Cross Validation

Cross Validation adalah metode validasi yang mengambil sebagian *dataset* untuk dijadikan sebagai *data testing*. Validasi dilakukan dengan cara membandingkan, apakah hasil klasifikasi atau prediksi yang dilakukan sistem sudah sesuai dengan data asli klasifikasi atau tidak [16]. Akurasi sistem

dapat dihitung dengan cara, jumlah validasi benar dibagi dengan total *data testing*, kemudian dikalikan dengan 100%.

Pada penelitian ini, *data testing* didapatkan dengan mengambil 20% data dari total 30 data pertanyaan dan jawaban yang ada di *database* sistem. Pengambilan *data testing* dilakukan secara acak, dan kemudian data tersebut dihapus dari *database* sistem. Setelah itu, pertanyaan-pertanyaan pada *data testing* akan ditanyakan kembali pada sistem *chatbot*. Hasil *cross validation* dari sistem *chatbot* yang dibangun, dapat dilihat pada Gambar 9. Akurasi sistem *chatbot* yang didapatkan berdasarkan hasil *cross validation* adalah 83,33%.



Gambar 9. Hasil Cross Validation

IV. KESIMPULAN

Berdasarkan hasil *user validation* dan *cross validation* yang telah dilakukan, maka dapat disimpulkan bahwa metode NLP dapat diimplementasikan untuk membangun *chatbot* Program Information Technology, Universitas Surabaya. Akurasi yang didapatkan berdasarkan hasil dua validasi tersebut adalah lebih dari 75%, yang artinya metode NLP cukup baik untuk diterapkan pada penelitian ini. Akan tetapi, *chatbot* masih perlu dikembangkan lagi agar akurasi menjadi yang lebih baik. Hal ini dapat dilihat dengan masih ditemukannya beberapa kasus yang membuat hasil validasi kurang tepat, seperti yang sudah dijelaskan pada sub bab sebelumnya.

Untuk pengembangan *chatbot* berikutnya, diharapkan pemrosesan sinonim dari kata kunci dapat berjalan lebih baik lagi, sehingga pencarian jawaban juga dapat berlangsung lebih cepat. Selain itu berdasarkan saran dari *user*, perlu ditambahkan menu atau halaman admin khusus agar bisa menambahkan daftar pertanyaan, kata kunci, dan jawaban dengan lebih mudah, terutama untuk pertanyaan yang tidak ditemukan jawabannya, sehingga *dataset* yang ada menjadi lebih kaya dan beragam. *Chatbot* juga diharapkan untuk dikembangkan agar dapat menerima pertanyaan dalam bahasa Indonesia.

REFERENSI

- [1] Hakim, M.A & Nurhayati, S. (2019). Pembangunan Aplikasi Chatbot Midwify sebagai Media Pendukung Pembelajaran Ilmu Kebidanan Berbasis Android di Stikes Bhakti Kencana Bandung. *Komputika: Jurnal Sistem Komputer*, Vol. 8(1), pp. 45-52.
- [2] Wangsanegara, N.K. & Subaeki, B. (2015). Implementasi Natural Language Processing Dalam Pengukuran Ketepatan Ejaan Yang Disempurnakan (EYD) Pada Abstrak Skripsi Menggunakan Algoritma Fuzzy Logic. *Jurnal Teknik Informatika*, Vol. 8(2), pp. 1-6.
- [3] Suryani, D. & Amalia, E.L. (2017). Aplikasi Chatbot Objek Wisata Jawa Timur Berbasis AIML. *SMARTICS Journal*, Vol. 3(2), pp. 47-54.
- [4] Benedictus, R.R., Wowor, H. & Sambul, A. (2017). Rancang Bangun Chatbot Helpdesk untuk Sistem Informasi Terpadu Universitas Sam Ratulangi. *E-Journal Teknik Informatika*, Vol. 11(1).
- [5] Dwi, A.R., Imamah, F., Andre, Y.M.S. & Ardiansyah. (2018). Aplikasi Chatbot (MILKI BOT) Yang Terintegrasi Dengan Web CMS Untuk Customer Service Pada UKM MINSU. *Jurnal Cendikia*, Vol. XVI, pp. 100-106.
- [6] Amalia, E.L. & Wibowo, D.W. (2019). Rancang Bangun Chatbot Untuk Meningkatkan Performa Bisnis. *Jurnal Ilmiah Teknologi Informasi Asia*, Vol. 13(2), pp. 137-142.
- [7] Yuniar, E. & Purnomo, H. (2019). Implementasi Chatbot "ALITTA" Asisten Virtual Dari BALITTAS Sebagai Pusat Informasi Di BALITTAS. *Jurnal Ilmiah Teknik Informatika*, Vol. 12(1), pp. 24-35.
- [8] Khanna, A., Pandey, B., Vashishta, K., Kalia, K., Pradeepkumar, B., & Das, T. (2015). A Study of Today's A.I. through Chatbots and Rediscovery of Machine Intelligence. *International Journal of u- and e- Service, Science and Technology*, Vol. 8(7), pp. 277-284.
- [9] Lisangan, E.A. (2013). Natural Language Processing Dalam Memproses Informasi Akademik Mahasiswa Universitas Atma Jaya Makassar. *Jurnal TEMATIKA*, Vol. 1(1), pp. 1-9.
- [10] Redd, M.V. & Hanumanthappa. (2014). Semantical and Syntactical Analysis of NLP. *International Journal of Computer Science and Information Technologies*, Vol. 5(3), pp. 3236-3238.
- [11] Siddiqi, S. & Sharan, A. (2015). Keyword and Keyphrase Extraction Techniques: A Literature Review. *International Journal of Computer Applications*, Vol. 109(2), pp. 18-23.
- [12] Qaiser, S. & Ali, R. (2018). Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents. *International Journal of Computer Applications*, Vol. 181(1), pp. 25-29.
- [13] Christian, H., Agus, M.P. & Suhartono, D. (2016). Single Document Automatic Text Summarization Using Term Frequency-inverse Document Frequency (TF-IDF). *ComTech: Computer, Mathematics and Engineering Applications*, Vol. 7(4), pp. 285-294.
- [14] Khan, N.H., Saha, G.C., Sarker, B. & Rahman, M.H. (2014). Checking the Correctness of Bangla Words using N-Gram. *International Journal of Computer Applications*, Vol. 89(11).
- [15] Prasetyo, V.R. (2018). Searching Cheapest Product On Three Different E-Commerce Using K-Means Algorithm. *Proceeding of International Seminar on Intelligent Technology and Its Application (ISITIA)*. Bali, Indonesia.
- [16] Prasetyo, V.R., Hartanto, B. & Mulyono, A.A. (2019). Penentuan Pembimbing Tugas Akhir Mahasiswa Jurusan Teknik Informatika Universitas Surabaya Dengan Metode Dice Coefficient. *Teknika*, Vol. 8(1), pp. 44-51.