

Perancangan *Chatbot* di Universitas Proklamasi 45

Lukman Hakim¹, Sapriani Gustina², Shela Fadila Putri³, Sri Ulfa Faudiah⁴
^{1,2,3,4} Program Studi Teknologi Informasi, Universitas Proklamasi 45
email: lukman@up45.ac.id¹, sa.gustina@up45.ac.id², shelafadila2016@gmail.com³,
ulfafaudiah99@gmail.com⁴

(Received: 26 Mei 2020/ Accepted: 8 Juni 2020 / Published Online: 20 Juni 2020)

Abstrak

Customer Support adalah orang yang bertugas melayani pertanyaan calon mahasiswa secara tepat dan cepat. Namun penggunaan *Customer Support* membutuhkan biaya operasi besar serta memiliki jam kerja yang terbatas. Selain itu, dalam waktu-waktu sibuk terkadang *Customer Support* lambat dalam memberikan jawaban pertanyaan. Untuk meningkatkan efisiensi dan efektivitas kerja *Customer Support* di Universitas Proklamasi 45, pada penelitian ini sistem *Chatbot* akan digunakan untuk pertanyaan sederhana dari calon mahasiswa. Untuk pertanyaan yang lebih kompleks akan tetap diarahkan ke *Customer Support* konvensional oleh sistem. Waterfall adalah model yang digunakan untuk membangun sistem ini, sedangkan algoritma StarSpace dan *Count Vectorizer* untuk mendeteksi *intent* dari sebuah kalimat. Frekuensi kemunculan kata akan menjadi referensi untuk menentukan maksud sebuah kalimat. Evaluasi dilakukan dengan metode *Black-box*, yaitu dengan memasukkan pertanyaan dan mengevaluasi jawaban yang diberikan. Hasil evaluasi menunjukkan bahwa *Chatbot* yang dibangun dapat memberikan jawaban dengan akurasi 97.75%. Sehingga dapat disimpulkan bahwa sistem *Chatbot* dapat menggantikan peran *Customer Support*. Akurasi dapat ditingkatkan dengan menambah variasi data pada *data training*.

Kata kunci: *Chatbot, Natural Language Processing, StarSpace*

Abstract

Customer Support is a person who serves prospective student questions appropriately and quickly. However, the conventional *Customer Support* requires high operating costs and has limited working hours. Moreover, in busy times, sometimes *Customer Support* is slow to respond to questions. In this study, a *Chatbot* system will be used to answer simple questions from prospective students, so it can improve the efficiency and effectiveness of *Customer Support* work at the Proklamasi 45 universities. The more complex questions will still be directed to conventional *Customer Support* by the system. Waterfall is the model used to build this system, while the StarSpace algorithm and *Count Vectorizer* to detect the *intent* of a sentence. The frequency of occurrence of words in a sentence will be referenced to determine the *intent* of a sentence. The testing is performed by *Black-box* method by sending questions and evaluating the responses given by the system. The Evaluation results show that the *Chatbot* can provide answers with an accuracy of 97.75%. So it could be concluded that the *Chatbot* system can substitute the role of *Customer Support*. Accuracy can be improved by adding more data variations to the training data.

Keywords: *Chatbot, Natural Language Processing, StarSpace*

PENDAHULUAN

Setiap tahun ribuan orang dari berbagai daerah datang untuk menimba ilmu di berbagai kampus yang ada di Yogyakarta. Berdasarkan data dari DIKTI, terdapat 184 perguruan tinggi baik negeri maupun swasta di Yogyakarta. Hal ini menyebabkan persaingan menjadi ketat dan mengharuskan perguruan tinggi untuk terus berinovasi dan melakukan perbaikan agar dapat bertahan dan berkembang. Sebagai perguruan tinggi yang telah berdiri

sejak 1965, Universitas Proklamasi 45 terus melakukan berbagai cara untuk menjangkau mahasiswa baru. Salah satunya dengan menggunakan tenaga *tele-sales* dan *Customer Support* yang akan menjawab pertanyaan terkait penerimaan mahasiswa baru.

Manajemen *Customer Support* memiliki pengaruh signifikan terhadap kepuasan dan loyalitas pelanggan (Rahmat, 2018). Dalam konteks perguruan tinggi, loyalitas dapat diartikan kecenderungan mahasiswa untuk mengajak orang lain untuk bergabung di kampus tempat dia belajar, yang pada akhirnya dapat membantu promosi kampus. Selain itu, pelayanan *Customer Support* yang cepat dan tanggap memiliki nilai korelasi yang kuat dengan peningkatan citra perusahaan (Giastanti et al., 2018). Citra perusahaan yang baik sangat penting dalam menjalin kerja sama, baik dengan perusahaan maupun sesama perguruan tinggi. Perusahaan riset yang berbasis di Amerika Serikat Juniper Research melaporkan, penggunaan *Chatbot* oleh industri retail, perbankan, dan kesehatan tahun ini bisa menghemat biaya operasional hingga US\$ 6 miliar atau sekitar Rp 86,4 triliun dan dapat berkembang hingga mencapai US\$ 11 miliar sampai tahun 2023 (Juniper Research, 2019).

Penggunaan *Customer Support* di Universitas Proklamasi 45 masih memiliki kelemahan yaitu membutuhkan biaya operasional yang besar dan memiliki waktu kerja yang terbatas. Sehingga sering kali pertanyaan dari calon mahasiswa baru tidak segera dijawab, terutama saat hari libur. Padahal saat hari libur sering kali banyak pertanyaan masuk, karena saat itu orang memiliki waktu luang lebih banyak untuk mencari informasi. Sebagian besar pertanyaan yang diajukan adalah pertanyaan yang bersifat umum seperti syarat pendaftaran, biaya kuliah, dan akreditasi program studi. Untuk mengatasi masalah tadi, dapat dibangun suatu sistem *Chatbot* yang dapat menjawab pertanyaan dasar seputar informasi Universitas.

Terdapat beberapa penelitian terkait penerapan *Chatbot* dalam lingkungan bisnis. Aplikasi *Chatbot* diterapkan untuk menangani pemesanan di Usaha Kecil Menengah pemasaran produk susu sapi dengan nama Milkibot (Rahma et al., 2018). Pada penelitian lain, *Chatbot* juga digunakan untuk memberi layanan informasi pada Balai Penelitian Tanaman Pemanis dan Serat (Yuniar & Purnomo, 2019). Pada kedua penelitian tadi, aplikasi *Chatbot* dibuat dengan metode *forward-chaining*, di mana terdapat banyak *rule* untuk setiap kemungkinan tindakan yang dilakukan pengguna. Hal ini tidak efisien untuk model bisnis yang lebih kompleks, karena akan menambah panjang *rule* yang digunakan. Selain *forward-chaining*, aplikasi *Chatbot* juga dapat diterapkan menggunakan *Artificial Intelligence Markup Language* (AIML) AIML adalah salah satu jenis XML yang dirancang khusus untuk menyimpan pola pertanyaan-jawaban. Daftar pola pertanyaan akan disusun berpasangan dengan beberapa kemungkinan jawaban yang akan diberikan (Wallace, 2009). Berdasarkan penelitian sebelumnya (Ranoliya et al., 2017; Rusmarasy et al., 2019; Suryani & Amalia, 2017), terlihat bahwa penggunaan AIML memiliki kelemahan yaitu sulit diterapkan untuk pertanyaan yang rumit, karena alur dialog harus dibangun secara manual.

Berbeda dari penelitian sebelumnya, pada penelitian ini dikembangkan sebuah *Chatbot* dengan menggunakan teknik *Machine Learning*. *Chatbot* yang dikembangkan tidak hanya menjawab pertanyaan berdasarkan pola pertanyaan, tetapi juga dapat mendeteksi entitas yang terdapat dalam kalimat. Selain itu, *Chatbot* yang dibangun juga menggunakan konsep slot. Slot adalah data yang akan dijadikan referensi selama percakapan berlangsung, sehingga *Chatbot* dapat mengenali konteks pembicaraan. Model yang digunakan pada penelitian ini adalah Starspace. Starspace adalah model serbaguna yang dikembangkan Facebook untuk mendeteksi entitas. Diharapkan penelitian ini dapat meningkatkan pelayanan terhadap calon mahasiswa serta mengurangi beban kerja *Customer Support* yang ada di Universitas Proklamasi 45.

METODE

Perangkat lunak *Chatbot* dalam penelitian ini dikembangkan menggunakan metode *waterfall*. Tahapan dalam metode pengembangan *waterfall* seperti tampak pada Gambar 1 adalah sebagai berikut.

1. Analisa Kebutuhan

Pada tahap ini dilakukan perumusan spesifikasi sistem meliputi layanan yang diberikan, batasan sistem, serta target yang ingin dicapai. Analisa kebutuhan harus melibatkan pengguna akhir secara aktif

2. Perancangan Perangkat Lunak dan Sistem

Pada tahap ini, dilakukan perancangan arsitektur *software* maupun *hardware* untuk memenuhi kebutuhan yang telah didefinisikan pada tahap sebelumnya.

3. Implementasi dan Pengujian Unit

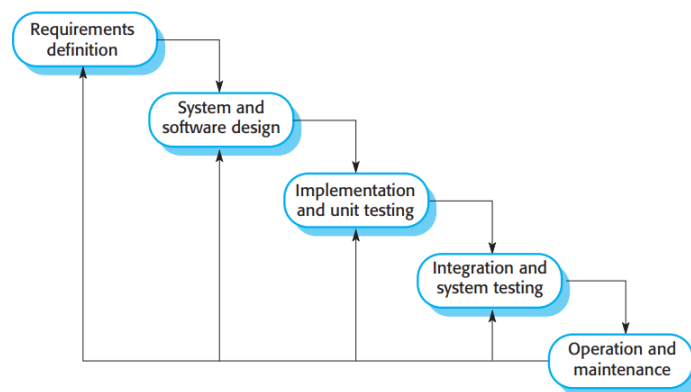
Pada tahap ini, hasil perancangan akan direalisasikan dalam bentuk unit-unit program. Unit-unit tersebut harus diuji untuk memastikan fungsionalitasnya

4. Integrasi dan Pengujian Sistem

Unit-unit program yang telah dihasilkan dari tahap sebelumnya akan diintegrasikan dan dilakukan pengujian secara menyeluruh untuk memastikan bahwa perangkat lunak telah memenuhi kebutuhan fungsional dan non-fungsionalnya.

5. Pengoperasian dan Perawatan

Pada tahap ini sistem akan dipasang pada lingkungan operasional, perbaikan akan dilakukan jika terdapat *error* atau terjadi perubahan fitur sistem.



Gambar 1. Pengembangan Perangkat Lunak *Waterfall* (Sommerville, 2016)

Pada tahapan pengujian, sistem akan diuji metode *Black-box* yaitu dengan memasukkan *input* dan kemudian mengevaluasi *output* yang dihasilkan saat program berjalan. Hasil pengujian akan dianalisis dengan teknik analisa data kuantitatif, yaitu dengan menghitung persentase jawaban benar yang dihasilkan *Chatbot*.

HASIL DAN PEMBAHASAN

Hasil

1. Analisa Kebutuhan

Proses analisis kebutuhan dilakukan untuk mendefinisikan kebutuhan fungsionalitas dan kebutuhan non-fungsionalitas. Proses ini melibatkan pimpinan program studi sebagai pihak yang menguasai *Business Domain* dan mahasiswa sebagai pengguna akhir. Kebutuhan fungsional adalah layanan apa saja yang harus disediakan oleh sistem. Tabel 1 menunjukkan kebutuhan fungsionalitas yang harus dipenuhi perangkat lunak *Chatbot* ini.

Tabel 1. Daftar Kebutuhan Fungsional

Kode FR	Nama Kebutuhan
FR-01	Sistem dapat memberi respons yang benar terhadap pertanyaan yang diberikan
FR-02	Sistem dapat mengarahkan pertanyaan ke <i>Customer Support</i> ketika tidak dapat menjawab pertanyaan yang diajukan
FR-03	Sistem dapat menyimpan data percakapan dengan <i>customer</i>
FR-04	Pengetahuan dalam sistem dapat ditambah atau dikurangi sesuai kebutuhan
FR-05	Sistem dapat dihubungkan dengan aplikasi telegram

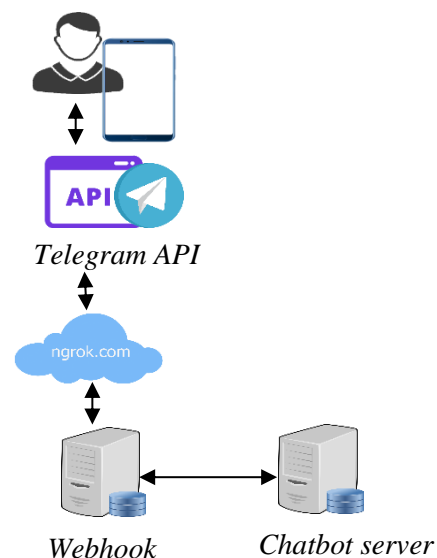
Kebutuhan non-fungsional adalah kebutuhan terkait batasan kualitas layanan yang akan disediakan sistem. Tabel 2 menunjukkan kebutuhan non-fungsionalitas yang harus dipenuhi sistem.

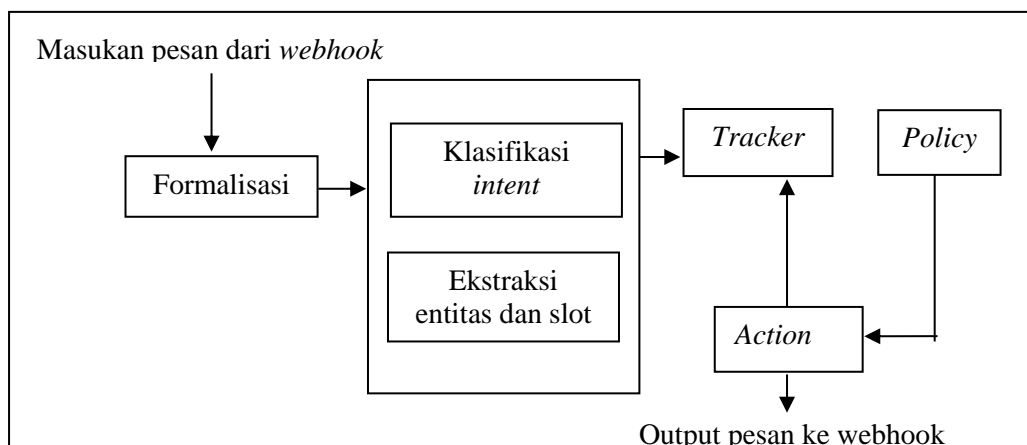
Tabel 2. Daftar Kebutuhan Non-Fungsional

Kode FR	Nama Kebutuhan
NFR-01	Sistem dapat memberi respons paling lama 1 menit setelah pertanyaan diberikan
NFR-02	Sistem memberikan jawaban dalam bahasa sehari-hari

2. Perancangan Perangkat Lunak dan Sistem

Gambar 2 menunjukkan arsitektur sistem *Chatbot* yang akan dibangun. Pengguna akan mengirimkan pesan melalui aplikasi telegram. Pesan tersebut kemudian diteruskan ke sebuah *Webhook*. *Webhook* berfungsi menerima pesan masuk dari Telegram API dan kemudian meneruskan ke *Chatbot server*. Untuk dapat menggunakan Telegram API, sebuah *Webhook* harus terhubung ke *public internet*, karena itu penelitian ini menggunakan sistem ngrok agar *Webhook* yang dibangun dapat terhubung dengan *public internet*. Ngrok adalah sistem yang memungkinkan komputer lokal untuk terhubung dengan *public internet*. Gambar 3 menunjukkan tahapan yang dilakukan oleh server *Chatbot* dalam memroses pesan yang masuk.

Gambar 2. Arsitektur Sistem *Chatbot*



Gambar 3. Arsitektur Sistem *Chatbot*

a. Formalisasi

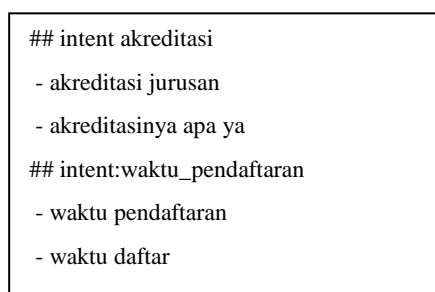
Pesan yang masuk akan diformalkan terlebih dahulu. Hal ini untuk membersihkan pesan dari kata tidak baku maupun singkatan yang sering digunakan oleh pengguna. Contoh formalisasi kata dapat dilihat pada tabel 3 berikut ini.

Tabel 3. Daftar kebutuhan fungsional

Bentuk tidak baku	Bentuk baku
Yg	yang
Bs	bisa
Pndftrn, pendafrn, Menggnkn, memakai, pakai, pake	pendaftaran menggunakan
Yg	yang
Bs	bisa
Pndftrn, pendafrn, Menggnkn, memakai, pakai, pake	pendaftaran menggunakan

b. Klasifikasi *Intent*

Klasifikasi *intent* adalah proses untuk mengetahui tujuan dari pesan secara umum. Pada penelitian ini algoritma yang digunakan untuk klasifikasi *intent* adalah model starspace yang dikembangkan oleh Facebook. Starspace merepresentasikan entitas dan komponen pembentuknya pada vektor yang sama (Wu et al., 2018). Gambar 4 menunjukkan contoh sederhana *data training* untuk pelabelan *intent*.



Gambar 4. Pelabelan *Intent*

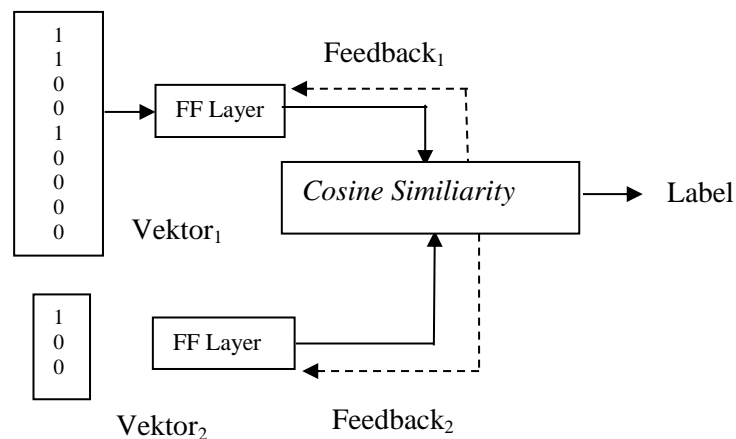
Setiap kalimat akan dimodelkan menggunakan metode *Count Vectorizer*. *Count Vectorizer* adalah metode untuk mengubah susunan kata dalam sebuah kalimat menjadi vektor dengan menghitung frekuensi kemunculan kata dalam sebuah kalimat (Eshan & Hasan,

2018). Gambar 5 menunjukkan proses transformasi dari kalimat menjadi vektor. Terdapat dua vektor untuk tiap kalimat, yaitu vektor yang merepresentasikan keberadaan unsur kata pada *vocabulary* dalam kalimat, dan vektor yang merepresentasikan keberadaan *intent* pada sebuah kalimat.

Vocabulary = [akreditasi, jurusan, apakah, sudah, apa, untuk, ya, syarat, pendaftaran]
 Intent = [akreditasi, waktu_pendaftaran, biaya_pendaftaran]
 Kalimat = akreditasi jurusannya apa ya?
 Vektor₁=[1,1,0,0,1,0,0,0,0]
 Vektor₂=[1,0,0]

Gambar 5. Pemrosesan Kalimat dengan *CountVectorizer*

Vektor kata dan vektor *intent* memiliki dimensi yang berbeda. Biasanya vektor kata memiliki dimensi jauh lebih besar, karena menampung setiap kata yang ada. Untuk dapat dibandingkan menggunakan fungsi *cosine similarity*, kedua vektor tersebut harus memiliki dimensi yang sama. Sehingga dibutuhkan sebuah *feed-forward layer* (*FF Layer*) untuk menyamakan dimensi kedua vektor tersebut. Hasil perhitungan *cosine similarity* adalah keputusan *intent* apakah yang terdapat pada suatu kalimat. Proses ini juga menghasilkan umpan balik untuk memperbaiki bobot dari *FF Layer*.



Gambar 6. Proses Klasifikasi *Intent*

c. Ekstraksi Entitas dan Slot

Ekstraksi entitas dilakukan untuk mengambil informasi tertentu dalam sebuah kalimat. Pada penelitian ini ekstraksi entitas dilakukan dengan metode *Named Entity Recognition Conditional Random Field* (NER-CRF). Pada metode ini proses *training* dilakukan secara manual dengan menandai entitas-entitas tertentu dalam kalimat. Selain mengekstrak entitas, sistem juga mengekstrak slot. Slot adalah data yang akan digunakan untuk mengenali konteks percakapan. Data slot hanya disimpan selama sesi percakapan berlangsung. Gambar 7 menunjukkan contoh *data training* untuk ekstraksi entitas dan slot. Pada contoh tersebut sistem akan dilatih untuk mengenali entitas jurusan. Jadi saat terdapat kalimat dengan unsur kata “teknik perminyakan” dan “teknik mesin”, maka sistem akan mendeteksi unsur kata tersebut sebagai nama jurusan.

```
##intent:syarat_pendaftaran
- untuk jurusan[teknik perminyakan](jurusan) syaratnya apa ya ?
- untuk jurusan [teknik mesin](jurusan) syaratnya apa ya
```

Gambar 7. Data training untuk Ekstraksi Entitas

d. Tracker

Tracker berfungsi untuk menjaga data dalam setiap sesi percakapan. Pada Gambar 8 tampak contoh penggunaan *Tracker* untuk membaca nilai dari slot jalur yang dimasukkan pengguna. Selain menyimpan nilai slot, *Tracker* juga digunakan untuk menyimpan *Action* yang telah dieksekusi.

```
def submit(
    self,dispatcher: CollectingDispatcher, tracker: Tracker,
    domain: Dict[Text, Any], ) -> List[Dict]:
    if tracker.get_slot('jalur') == 'karyawan':
        dispatcher.utter_message(template="utter_tanya_syarat_pendaftaran_karyawan")
    elif tracker.get_slot('jalur') == 'pindahan':
        dispatcher.utter_message(template="utter_tanya_jalur_pindahan")
    else:
        dispatcher.utter_message(template="tanya_syarat_pendaftaran")
```

Gambar 8. Contoh Penggunaan *Tracker*

e. Action

Action adalah sesuatu yang akan dijalankan sebagai respons atas masukan pengguna. *Action* dapat berupa pesan balasan berdasarkan template tertentu seperti tampak pada Gambar 9. Saat *intent* *tanya_kontak* terdeteksi, maka *action* *utter_tanya_kontak* akan dieksekusi. Seperti tampak pada Gambar 10, *action* juga dapat berupa program dalam bahasa python, cara ini dilakukan jika operasi yang dilakukan cukup kompleks.

```
templates:
  utter_default:
    - text: "Pertanyaan akan saya sampaikan ke pihak yang berwenang, nanti akan dibalas
ketika sudah ada jawaban"
  utter_tanya_kontak:
    - text : "Untuk info lebih lanjut bisa hubungi nomer 0859195370754"
  utter_tanya_dosen:
```

Gambar 9. Contoh *Template Action*

```
class PendaftaranForm(FormAction):
    def name(self):
        return "pendaftaran_form"
    @staticmethod
    def required_slots(tracker: Tracker) -> List[Text]:
        return ["jalur"]
```

Gambar 10. Contoh *CustomAction*

f. Policy

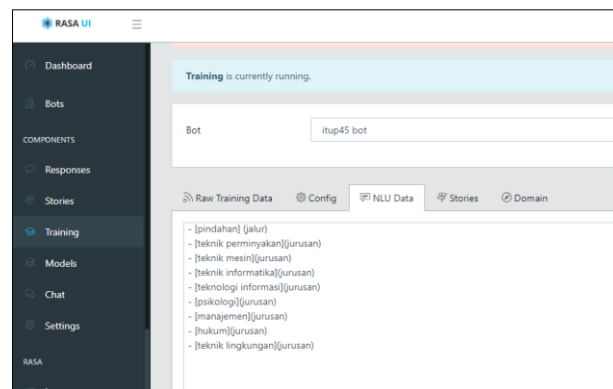
Policy berfungsi untuk memilih *action* yang akan dieksekusi untuk merespons masukan pengguna. Pemilihan ini ditentukan melalui *user story* sebagaimana tampak pada Gambar 11. Saat pengguna mengucapkan salam maka program akan menjalankan *action* yang berisi balasan salam, kemudian memperkenalkan diri dan terakhir adalah menawarkan bantuan. *User story* dapat juga lebih kompleks, misal saat pengguna menanyakan syarat pendaftaran, sistem akan mengaktifkan sebuah *form*, sehingga pengguna harus mengisi terlebih dahulu data slot yang dibutuhkan.

```
* salam_assalamualaikum
- utter_salam_assalamualaikum
- utter_perkenalan
- utter_tawaran_bantuan
* syarat_pendaftaran
- pendaftaran_form
- form{"name": "pendaftaran_form"}
- form{"name": null}
```

Gambar 11. Contoh *User Story*

3. Implementasi

Sistem *Chatbot* dalam penelitian ini dibangun menggunakan bahasa pemrograman PHP dan Python. Bahasa pemrograman PHP digunakan untuk membangun *webhook* serta membuat *user interface* bagi admin untuk mengelola *data training*, *action*, *intent*, maupun slot. Server *Chatbot* dibangun menggunakan bahasa pemrograman Python dengan pustaka Tensorflow dan berjalan di atas *framework* Flask. Gambar 13 menunjukkan tampilan interaksi antara pengguna dengan *Chatbot* melalui aplikasi telegram.



Gambar 12. Tampilan Admin *Chatbot*



Gambar 13. Tampilan *Chatbot*

4. Pengujian Sistem

Pengujian dilakukan dengan metode *Black-box testing*, yaitu dengan mengirimkan pertanyaan ke sistem chatbot dan kemudian mengevaluasi jawaban yang diberikan. Pengujian ini melibatkan mahasiswa baru untuk menyimulasikan pertanyaan-pertanyaan dari calon pendaftar. Tabel 4 menunjukkan sebagian dari hasil pengujian yang dilakukan. Dari 87 pertanyaan, *Chatbot* berhasil memberikan jawaban benar sebanyak 87 jawaban benar dan 2 jawaban salah atau dengan akurasi 97.75%.

Tabel 4 Hasil Pengujian

No	Pertanyaan	Balasan	Benar/Salah
1	ada nomor telepon yg bisa dihubungi	Untuk info lebih lanjut bisa hubungi nomor 0859195370754	Benar
2	Alamatnya dmn	Alamat kami di Jalan Proklamasi no.1 Babarsari Sleman Yogyakarta	Benar
3	Beasiswa yang tersedia apa sj	Ada 3 jenis beasiswa disini kak. Bidikmisi, proklamasi dan nusantara	Benar
4	Syarat bidik misi apa ya	Syarat bidikmisi adalah calon peserta telah didaftarkan oleh Sekolah asal kedalam sistem bidikmisi	Benar
5	Biaya pendaftarannya berapa	Biaya pendaftaran sebesar Rp. 150.000.	Benar
6	syaratnya apa ya	Syarat bidikmisi adalah calon peserta telah didaftarkan oleh Sekolah asal kedalam sistem bidikmisi	Salah

Pembahasan

Berdasarkan pengamatan selama penelitian, diketahui bahwa akurasi sistem sangat bergantung terhadap variasi *data training* serta pemilihan kombinasi kata. Penggunaan kombinasi kata yang terlalu umum menyebabkan jawaban *Chatbot* menjadi tidak akurat. Misal jika menggunakan kalimat “syarat pendaftaran”, respon dari *Chatbot* akan acak ke topik syarat pendaftaran beasiswa, syarat pendaftaran kelas reguler dan lain lain. Jadi kalimat yang digunakan dalam *data training* harus lebih spesifik, misal “syarat pendaftaran beasiswa bidikmisi”. Sebaliknya pemilihan kalimat yang terlalu spesifik, misal “saya ingin bertanya, kalau mau mendaftar bidikmisi syarat-syaratnya apa saja ya”, juga dapat mengurangi akurasi. Karena metode yang digunakan adalah dengan menghitung frekuensi kemunculan kata. Jadi harus benar-benar dipilih kata yang mempunyai pengaruh signifikan ke topik yang diinginkan. Masalah yang masih timbul adalah ketika pengguna memasukkan pertanyaan yang kurang spesifik, misal “syaratnya apa?”. *Chatbot* akan tetap mengirimkan balasan yang acak, terkadang syarat pendaftaran, syarat beasiswa dan lain lain. Seharusnya *Chatbot* menanyakan lebih detil maksud pertanyaan tersebut, atau dapat juga meneruskan ke *Customer Support*. *Chatbot* juga telah dapat mengenali entitas dalam kalimat, misal jurusan, jenis beasiswa, maupun jalur registrasi dengan minimal 25 *data training* untuk tiap entitas.

Chatbot telah dapat meneruskan pertanyaan ke *Customer Support* ketika mendapat pertanyaan diluar pengetahuannya. Performa *Chatbot* yang dibangun juga sudah baik, karena rata rata dapat memberikan jawaban kurang dari 5 detik. Namun untuk pesan pertama biasanya akan mendapat jawaban setelah 10 detik. Hal itu karena sistem harus membuat *thread* baru saat pertama kali pesan masuk. Sistem *Chatbot* ini juga dilengkapi dengan halaman admin, sehingga proses pengelolaan *Chatbot* dapat dilakukan dengan. Daftar pertanyaan yang diberikan pengguna juga telah tersimpan dan dapat digunakan untuk *data training*, sehingga akurasi *Chatbot* dapat ditingkatkan.

SIMPULAN

Berdasarkan hasil dan pembahasan yang telah dijabarkan di atas, dapat disimpulkan bahwa *Chatbot* dapat digunakan untuk mengotomasi kerja *Customer Support.*, karena akurasi yang dihasilkan sudah mencapai 97.75% dan waktu respons yang cukup cepat. *Chatbot* yang dibangun sangat bergantung pada jumlah *data training*, karena itu jumlah *data training* harus ditingkatkan sehingga akurasi yang dihasilkan lebih baik. Untuk ke depannya, sistem ini dapat dihubungkan dengan *graph database*, sehingga manajemen pengetahuan dalam sistem menjadi lebih baik.

REFERENSI

- Eka Yuniar, & Heri Purnomo. (2019). Implementasi Chatbot “Alitta” Asisten Virtual Dari Balittas Sebagai Pusat Informasi Di Balittas. *Antivirus: Jurnal Ilmiah Teknik Informatika*, 13(1), 24–35
- Eshan, S. C., & Hasan, M. S. (2018). An application of machine learning to detect abusive Bengali text, *2017 20th International Conference of Computer and Information Technology (ICCIT)* (pp. 1-6), Dhaka:IEEE.
- Giastanti, M., Hamim, H., & Pratiwi, N. M. I. (2018). Pengaruh Pelayanan Customer Service Terhadap Peningkatan Citra Perusahaan: Studi Korelasional Pada Pt. Ace Hardware Indonesia, Tbk. Di Lenmarc Mall Surabaya. *Representamen*, 3(02),1-7.
- Juniper Research. (2019). *Bank Cost Savings Via Chatbots to Reach \$7.3 Billion By 2023, As Automated Customer Experience Evolves*. Diambil dari <https://www.juniperresearch.com/press/press-releases/bank-cost-savings-via-chatbots-reach-7-3bn-2023>
- Rahma, A. D., Imamah, F., Andre, Y. M., & Ardiansyah. (2018). Aplikasi Chatbot (Milki Bot) Yang Terintegrasi Dengan Web CMS Untuk Customer Service Pada UKM MINSU. *Jurnal Cendikia*, XVI, 100–106.
- Rahmat, A. (2018). Pengaruh Customer Relationship Management Terhadap Kepuasan Dan Loyalitas Pelanggan (Survei pada Pelanggan Service Kendaraan AUTO2000 Kediri Suharmadji). *Jurnal Administrasi Bisnis*, 64(1), 153-160.
- Ranoliya, B. R., Raghuwanshi, N., & Singh, S. (2017). Chatbot for university related FAQs. *2017 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2017, 2017-Janua*, 1525–1530. <https://doi.org/10.1109/ICACCI.2017.8126057> Ranoliya, B. R., Raghuwanshi, N., & Singh, S. (2017). Chatbot for university related FAQs. *2017 International Conference on Advances in Computing, Communications and Informatics* (pp. 1525–1530), Udipi:IEEE.
- Rusmarasy, B., Priyambadha, B., & Pradana, F. (2019). Pengembangan Chat Bot pada CoMa untuk Memberikan Motivasi Kepada Pengguna Menggunakan AIML. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 3(5), 4484–4490.
- Sommerville, I. (2016). *Software engineering (10th edition)*. In Pearson Education Limited. Essex. England.
- Suryani, D., & Amalia, E. L. (2017). Aplikasi Chatbot Objek Wisata Jawa Timur Berbasis AIML. *SMARTICS Journal*, 3(2), 47–54.
- Wallace, R. S. (2009). The anatomy of A.L.I.C.E. In *Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer* (pp. 181–210). Dordrecht:Springer.
- Wu, L., Fisch, A., Chopra, S., Adams, K., Bordes, A., & Weston, J. (2018). StarSpace: Embed all the things! *32nd AAAI Conference on Artificial Intelligence* (pp. 5569–5577), California:AAAI.