

Rancang Bangun Sistem Tanya-jawab Berbasis Aturan STMIK Muhammadiyah Paguyangan Brebes dengan Menggunakan *Telegram Chatbot*

Fauzan Ishlakhuddin^{1*)}, Azhar Basir², Nurlaela³

^{1,2}Jurusan Teknik Informatika, STMIK Muhammadiyah Paguyangan Brebes, Brebes

³Jurusan Sistem Informasi, STMIK Muhammadiyah Paguyangan Brebes, Brebes

^{1,2,3}Jln. Pangeran Diponegoro, Paguyangan, Brebes, 52276, Indonesia

email: ¹fauzan@stmikmpb.ac.id, ²azhar@stmikmpb.ac.id, ³nurlaela005@gmail.com

Abstract — An educational institution should have an academic service system that is easy, accurate and fast in serving documents related to lecture information such as: *Kartu Rencana Studi (KRS)* and *Kartu Hasil Studi (KHS)* which will indirectly support the smooth teaching and learning activities. One of solution is to design an automated service system that can be accessed and served anytime, anywhere, and from anywhere, such as a chatbot. The method used in the design of this chatbot is to use a rule-based method to determine the answer (action) to be taken next. The main emphasis of this research only focuses on information services for the Biro Administrasi Akademik (BAAK). From the results of the adoption of this rule-based chatbot, a fairly high level of accuracy is obtained, which is up to 98.95% success rate (accuracy of answers) from 96 times of testing with different schemes.

Abstrak — Dalam sebuah lembaga pendidikan seyogyanya memiliki sistem layanan akademik yang mudah, akurat dan cepat dalam melayani dokumen-dokumen terkait informasi perkuliahan seperti: *Kartu Rencana Studi (KRS)* dan *Kartu Hasil Studi (KHS)* yang mana secara tidak langsung akan mendukung lancarnya kegiatan belajar-mengajar. Salah satu solusi yang mungkin dapat dilakukan adalah dengan merancang sebuah sistem layanan otomatis yang dapat diakses dan melayani kapan saja, dimana saja, dan darimana saja seperti *chatbot*. Metode yang digunakan dalam rancang-bangun *chatbot* ini adalah dengan menggunakan metode berbasis aturan guna menentukan jawaban (aksi) yang akan dilakukan berikutnya. Penekanan utama dari penelitian ini hanya berfokus pada layanan informasi Biro Administrasi Akademik (BAAK). Dari hasil penerapan *chatbot* berbasis aturan inik didapatkan tingkat akurasi yang cukup tinggi yaitu hingga 98,95% tingkat keberhasilan (ketepatan jawaban) dari 96 kali pengujian dengan sekema yang berbeda-beda.

Kata Kunci — *Chatbot, Sistem Tanya-jawab, Basis Aturan, Question Answer, Rule-based.*

I. PENDAHULUAN

Lembaga pendidikan memiliki peranan penting dalam perkembangan anak bangsa terutama dalam segi keilmuan dan kepribadian, apalagi dalam sebuah institusi perkuliahan yangmana menuntut para siswa didik (mahasiswa) untuk aktif, kreatif, dan memiliki pemikiran yang kritis. Untuk mendukung hal-hal tersebut maka pihak institusi juga perlu memiliki sistem yang memadai dalam segi pelayanan,

terutama ketersediaan layanan informasi yang dapat diakses secara mudah kapanpun dimanapun seperti *Chatbot*.

Chatbot merupakan sebuah sistem sederhana guna memfasilitasi percakapan dengan menggunakan “aturan *pattern matching*” untuk melakukan percakapan dengan pengguna, namun *chatbot* memiliki kemampuan pemahaman bahasa yang terbatas [1]. Dalam penerapannya, *chatbot* dapat dibangun secara mandiri atau menggunakan aplikasi *chatting* yang menyediakan layanan untuk mengembangkan *chatbot* [2]. *Chatbot* dalam melakukan percakapan (memahami dan menjawab *chat*) membutuhkan metode tertentu yang biasa disebut dengan menerapkan metode pengolahan bahasa alami atau *Natural Language Processing*.

Natural Language Processing (NLP) merupakan disiplin ilmu komputer yang bertujuan untuk memahami konsep dan maksud dari bahasa manusia. Sementara manusia cukup mahir dalam memahami sintak linguistik dan tata bahasa serta makna yang tersirat (semantik), sedangkan komputer memiliki kesulitan dalam pengolahan *query* bahasa alami [3]. NLP dapat digunakan sebagai alat yang memungkinkan komputer untuk mengekstrak entitas dari suatu teks, menerjemahkan informasi antara bahasa, dan dapat menafsirkan bahasa alami atau semantik dari suatu teks [4]. Metode yang dapat digunakan untuk mengimplementasikan semantik adalah dengan memanfaatkan ontologi sebagai media representasi pengetahuan bahasa, dan *rule-based* sebagai metode untuk melakukan penalaran kalimat pertanyaan sekaligus menentukan jawaban.

Rule-based merupakan salah satu bentuk pendekatan yang merepresentasikan pengetahuan menggunakan serangkaian aturan yang ditulis dalam bentuk *IF-THEN* [5]. Sedangkan menurut Sakinnah (2016) *rule-based* merupakan salah satu basis dari sistem pakar yang menggunakan kecerdasan manusia untuk memecahkan masalah dunia nyata yang direpresentasikan dengan aturan. Dengan demikian, kompleksitas sistem yang menggunakan metode basis aturan akan sangat tergantung dengan seberapa banyak aturan yang telah dibuat, dan seberapa akurat aturan dalam mengambil keputusan. Implementasi dari penerapan sistem-tanya jawab berbasis aturan ini menggunakan API Telegram yang dapat digunakan secara gratis dan tanpa batas, juga memiliki kemudahan dalam segi implementasi karena dapat diimplementasikan dalam banyak bahasa pemrograman.

*) penulis korespondensi: Fauzan Ishlakhuddin
Email: fauzan@stmikmpb.ac.id

Masalah yang dihadapi saat ini adalah belum adanya sistem informasi yang dapat melayani permintaan informasi terkait kegiatan pendidikan secara otomatis, seperti permintaan krs, khs, info pmb, info pembayaran, info tunggakan, info umkm, dan lain-lain. Namun dalam penelitian ini penulis membatasi akan lebih fokus pada pelayanan Biro Administrasi Akademik (BAAK) terlebih dahulu agar lingkup penelitian tidak terlalu luas dan sesuai kebutuhan yang paling mendasar.

II. PENELITIAN YANG TERKAIT

Penelitian tentang *chatbot* telah dilakukan oleh Augello et al. (2014) yang mengembangkan sebuah *chatbot* berbasis ontologi agar dapat memasok informasi yang lebih baik guna mengatasi keterbatasan pengetahuan pada *chatbot*. Penelitian ini menerapkan metode *hybrid*, di mana adanya penggabungan metode simbolis dan sub-simbolis. Simbolis digunakan untuk menjawab pertanyaan yang rasional, sedangkan sub-simbolis digunakan untuk menjawab pertanyaan yang sifatnya intuitif dengan pendekatan metode *Latent Semantic Analysis* (LSA). Penelitian ini sudah diterapkan dan diuji, namun tidak dicantumkan keterangan lebih lanjut mengenai hasil pengujian yang telah dilakukan pada jurnal tersebut.

Menurut Quarteroni & Manandhar (2007), *chatbot* merupakan salah satu representasi dari sistem *Question Answering* (QA) yang interaktif. Terdapat beberapa peneliti yang telah menggunakan pendekatan yang bervariasi terkait sistem QA. Bendi (2010) melakukan penelitian dengan menerapkan pendekatan *pattern matching* di dalam membangun sistem QA, sedangkan Suryawan (2013) menggunakan pendekatan *linguistik* pada sistem QA yang dibangun pada penelitiannya. Penelitian lainnya dilakukan oleh Siami (2014) yang melakukan penelitian terkait sistem QA dengan pendekatan *machine learning*, berbeda dengan penelitian yang lain, Muttaqin (2016) membangun pola *reasoning* sistem QA dengan ontologi menggunakan OWL 2.

Domain dalam penelitian QA yang telah dilakukan sebelumnya bermacam-macam, seperti yang dilakukan oleh Bendi (2010) yang melakukan penelitian pada domain film, adapun Suryawan (2013) melakukan penelitian pada domain perpustakaan, ada juga Siami (2014) yang melakukan penelitian pada domain Fiqih, lain lagi dengan Muttaqin (2016) yang melakukan penelitian pada domain data Kabupaten NTB. Dari keseluruhan penelitian tersebut, para peneliti melakukan penelitian sistem QA pada domain yang tertutup (*close domain*).

Hasil pengujian dari penelitian QA yang telah dilakukan bermacam-macam. Hasil yang didapat dari penelitian Bendi (2010) menunjukkan bahwa 96,7% pertanyaan dapat dijawab dengan tepat. Adapun hasil pengujian dari penelitian Suryawan (2013) didapatkan tingkat keberhasilan 80% dalam menjawab pertanyaan dengan baik. Sedangkan hasil pengujian yang didapatkan pada penelitian Siami (2014) memiliki tingkat akurasi sebesar 86,71%. Lain hal dengan penelitian yang dilakukan oleh Muttaqin (2016) yang menggabungkan beberapa ontologi dan mendapatkan hasil yang konsisten, dari hasil pengujian dapat dipertanggungjawabkan atas akurasi dalam menjawab pertanyaan, namun di dalam laporannya tidak menyebutkan tingkat akurasi yang didapatkan.

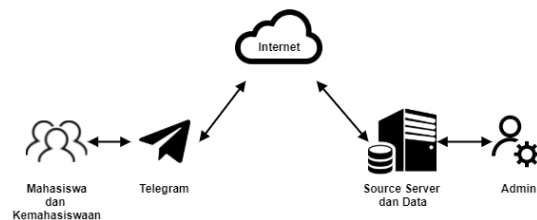
Penelitian terkait lain tentang pengembangan *semantic search* yang dilakukan oleh Winantu (2014) dengan menerapkan metode *Semantic Web Rule Language* (SWRL) untuk membangun aturan pada ontologi untuk *query*. Domain dari penelitian ini membangun sebuah prototipe aplikasi pencarian informasi produk handphone. Pemrosesan bahasa alami yang dilakukan dalam penelitian ini mengacu pada algoritma Tala. Hasil pengujian didapatkan rasio *precision* sebesar 99,55% dan rasio *recall* sebesar 93,67%.

Dalam penelitian ini merupakan pengembangan dari penelitian sebelumnya, yaitu mengembangkan *chatbot* berbasis Telegram dengan menerapkan algoritma Tala seperti yang telah dilakukan oleh Winantu (2014) untuk memahami *keyword* ataupun kalimat pertanyaan. Perbedaan penelitian ini dari penelitian yang telah dilakukan oleh Winantu (2014) terletak pada domain permasalahan pada penerapan.

III. METODE PENELITIAN

A. Kerangka Berfikir

Kerangka berfikir sistem *chatbot* kemahasiswaan yang akan dikembangkan untuk STMIK Paguyangan Brebes sebagaimana yang tertera dalam Gambar 1.

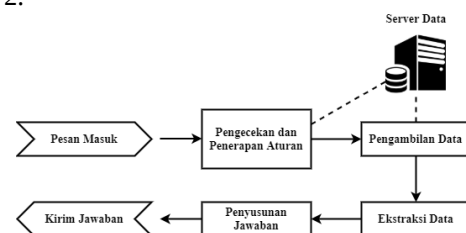


Gbr. 1 Kerangka Berfikir Sistem.

Pada Gambar 1 diatas menggambarkan kerangka berfikir sistem secara garis besar, dimana sistem *chatbot* akan selalu berjalan pada server yang terhubung dengan internet menunggu jika ada pesan masuk dari *Telegram Bot* yang telah ditentukan atau dibuat, sedangkan balasan yang akan dikirim sudah ditentukan dalam sebuah aturan tertentu.

B. Rancangan Alur Proses Chatbot

Alur proses sistem *chatbot* ini secara garis besar memiliki tahapan-tahapan yang akan selalu dilalui untuk setiap pesan masuk, adapun tahapan-tahapan prosesnya seperti pada Gambar 2.



Gbr. 2 Alur Proses Chatbot.

Pada Gambar 2 merupakan gambaran secara keseluruhan proses, dari mulai proses pesan masuk hingga mendapatkan balasan, dimana ketika *bot* mendeteksi ada pesan masuk maka dilakukan proses pengecekan dan penerapan aturan yang telah ditentukan yang disimpan dalam server. Jika jawaban yang akan dikirim membutuhkan data tertentu dari database yang sudah ditetapkan maka sistem akan mengambil data sesuai kebutuhan dan merancang menjadi jawaban kemudian sistem mengirim jawaban tersebut sesuai dengan aturan yang ada.

C. Rancangan Aturan

Tahapan ini membahas tentang rancangan aturan yang akan diimplementasikan dalam menentukan jawaban dari setiap pesan yang masuk, dimana aturan ini akan diterapkan untuk setiap pesan masuk (dikirim) ke telegram *chatbot* yang dinami dengan *StmikMpbBot*. Berikut untuk keseluruhan aturan yang akan diterapkan seperti yang ada pada Tabel I.

TABEL I

RANCANGAN ATURAN

No	Nama Aturan	Aturan
1	menu	IF (text == any) menu = SELECT * FROM menu WHERE level = 0 m = bot.send_message(message.chat.id, menu) bot.register_next_step_handler(m, menu_handler) ENDIF
2	menu_handler	IF (text is number) sub_menu = SELECT * FROM menu WHERE level = 1 AND sub = menu_number m = bot.send_message(message.chat.id, sub_menu) bot.register_next_step_handler(m, sub_menu_handler) ELSE bot.send_message(message.chat.id, "Maaf menu yang anda pilih tidak sesuai!") menu(message) ENDIF
3	sub_menu_handler	IF (text is number) rule = SELECT rule FROM menu WHERE menu_id = id_menu exec(rule) ELSE bot.send_message(message.chat.id, "Maaf menu yang anda pilih tidak sesuai!") menu(message) ENDIF
4	tanya_nim	m = bot.send_message(message.chat.id, "Masukkan NIM anda!") bot.register_next_step_handler(m, cek_nim)
5	cek_nim	user = SELECT * FROM user WHERE username = message.text IF (user is not None) IF (text_menu == "krs" OR text_menu == "khs") tanya_semester(m) ELSEIF (text_menu == "komulatif" OR text_menu == "proposal" OR text_menu == "pin") tanya_pin(m) ELSEIF (text_menu == "sanggah_b") sanggah_baak(m) ELSEIF (text_menu == "sanggah_k" OR text_menu == "konfirmasi_k" OR text_menu == "tunggakan") OR text_menu == "pembayaran") sanggah_keuangan(m) ENDIF ELSE bot.send_message(message.chat.id, "NIM tidak terdaftar (salah!)") ENDIF
6	tanya_semester	m = bot.send_message(message.chat.id, "Masukkan semester anda!") bot.register_next_step_handler(m, tanya_pin)
7	tanya_pin	IF (ulang < 3) m = bot.send_message(message.chat.id, "Sebutkan PIN Anda!") bot.register_next_step_handler(m, cek_pin) ulang = ulang+1 ELSE m = bot.send_message(m.chat.id, "Anda telah salah 3 kali, silahkan pilih menu kembali!") menu(message) ENDIF
8	cek_pin	IF (message.text == user.nim) IF (text_menu == "pin") ubah_pin(message) ELSEIF (text_menu == "krs" OR text_menu == "khs" OR text_menu == "komulatif") get_dokumen(message, text_menu, user.semester, user.nim) ELSEIF (text_menu == "tunggakan") sanggah_keuangan(message) ELSEIF (text_menu == "proposal") kumpul_proposal(message) ENDIF

No	Nama Aturan	Aturan
		ELSE bot.send_message(message.chat.id, "Pin salah, silahkan coba lagi!") tanya_pin(message) ENDIF
9	sanggah_baak	baak_id = SELECT tele_id FROM user WHERE username = "baak_user_id" bot.forward_message(baak_id, message.chat.id, message.message_id) bot.send_message(message.chat.id, "Pesan telah diteruskan ke bagian BAAK")
10	sanggah_keuangan	keuangan_id = SELECT tele_id FROM user WHERE username = "keuangan_user_id" bot.forward_message(keuangan_id, message.chat.id, message.message_id) bot.send_message(message.chat.id, "Pesan telah diteruskan ke bagian Keuangan")
11	get_dokumen	hasil = "SELECT * FROM file WHERE nim = user.nim AND jenis = text_menu AND semester = user.semester file = open(file_and_name_location, "rb") bot.send_document(message.chat.id, file, menu(message))
12	kumpul_proposal	save_f = save_directory + "/" + u.nim + "_" + m.document.file_name file_id_info = bot.get_file(m.document.file_id) downloaded_file = file_id_info.file_path bot.download_file(file_id_info.file_path) with open(save_f, "wb") as new_file: new_file.write(download_file) bot.send_message(message.chat.id, "Proposal berhasil disimpan dan disampaikan!") bot.forward_message(keuangan_id, message.chat.id, message.message_id) bot.send_message(message.chat.id, "Pesan telah diteruskan ke bagian Kemahasiswaan")
13	ubah_pin	m = bot.send_message(message.chat.id, "Masukkan PIN baru!") bot.register_next_step_handler(m, konfirmasi_pin)
14	konfirmasi_pin	m = bot.send_message(message.chat.id, "Masukkan kembali PIN baru anda!") user.new_pin = message.text bot.register_next_step_handler(m, simpan_pin)
15	simpan_pin	IF (user.new_pin == message.text) query = UPDATE users SET password = user.new_pin WHERE username = user.nim bot.send_message(message.chat.id, "PIN berhasil diperbarui!") menu(message) ELSE bot.send_message(message.chat.id, "Maaf pin baru tidak cocok, silahkan ulang kebalik!") ubah_pin(message) ENDIF

Pada Tabel 1 di atas menjabarkan *rule* (aturan) yang digunakan dalam penentuan jawaban dari setiap menu yang dipilih, dimana ada 12 aturan yang akan diimplementasikan dan saling berkaitan antara satu dengan yang lain karena sifatnya *redirect* (pengalihan). Untuk aturan no. 1 merupakan menu utama yang akan selalu ditampilkan jika ada pesan masuk atau jika ada kesalahan sehingga user bisa memilih menu Kembali. Untuk aturan no. 2 merupakan sub menu dari setiap menu utama yang akan mengarahkan ke aturan-aturan yang telah ditentukan melalui database yang ada. Sedangkan untuk aturan yang lain merupakan aturan yang diberlakukan dari setiap sub menu yang ada.

D. Penerapan Rancangan Aturan

Penerapan rancangan aturan dalam penelitian ini digambarkan menggunakan flowchart diagram seperti pada Gambar 3 di bawah.

Penjelasan dari potongan baris kode pada Gambar 5 secara terangkum adalah sebagai berikut: (a) Baris ke-1 merupakan perintah untuk membuat fungsi; (b) Baris ke-2 merupakan perintah untuk pemanggilan global variable dalam Python, yang mana global variable merupakan variable yang didefinisikan di luar fungsi; (c) Baris ke-3 hingga akhir merupakan perintah untuk logika dari mencari menu, mengembalikan variable menu dll.

Implementasi untuk penanganan menu utama dan menampilkan submenu dari setiap menu, untuk model pencariannya menggunakan id menu utama yang disimpan pada level menu 2 (submenu). Adapun hasil implementasinya dapat dilihat pada Gambar 6 di bawah.

```

1 def menu_handler(message):
2     global menus, data
3     menu("salam", 0)
4     try:
5         text = message.text
6         if text.isdigit() and text == "6":
7             msg = bot.reply_to(message, menus[int(text)-1][1])
8             stat = text
9             exec(menus[int(text)-1][0])
10        elif text.isdigit() and text == "5":
11            data = menus[int(text)-1][2]
12            exec(menus[int(text)-1][0])
13        elif text.isdigit():
14            msg = bot.reply_to(message, menu(text, 2))
15            stat = text
16            exec(menus[int(text)-1][0])
17        else:
18            msg = bot.send_message(message.chat.id, menu("salam", 0))
19    except Exception as e:
20        print('Error on line {}'.format(sys.exc_info()[-1].tb_lineno), type(e).__name__, e)
21    bot.reply_to(message, 'Ada kesalahan, silahkan coba lagi')
```

Gbr. 6 Potongan Baris Kode Implementasi Penanganan Menu Utama

Penjelasan dari potongan baris kode pada Gambar 6 secara terangkum adalah sebagai berikut: (a) Baris ke-1 merupakan perintah untuk membuat fungsi; (b) Baris ke-2 merupakan perintah untuk pemanggilan global variable dalam Python, yang mana global variable merupakan variable yang didefinisikan di luar fungsi; (c) Baris ke-3 hingga akhir merupakan perintah untuk logika dari mencari menu, mengembalikan variable menu dll.

C. Pengujian Aturan

Dalam penelitian ini setelah dilakukan percobaan dari setiap menu yang ditampilkan didapatkan bahwa StmikMpbBot mampu menangani menu yang ditampilkan sesuai dengan aturan yang telah dirancang. Untuk beberapa gambaran dari tampilan StmikMpbBot ini dapat dilihat pada Gambar 7 dan 8, sedangkan untuk keseluruhan hasil pengujian dapat dilihat pada bagian pengujian.



Gbr. 7 Contoh Pengujian untuk Menampilkan Menu Utama



Gbr. 8 Contoh Pengujian untuk salah Satu Menu Hingga Hasil

Gambar 7 di atas merupakan contoh mengujian untuk menampilkan menu utama dan submenu dari menu yang dipilih, dimana submenu disesuaikan dengan menu yang dipilih, jika menu yang dipilih tidak memiliki submenu atau langsung eksekusi sebuah proses maka akan langsung eksekusi proses tersebut, sedangkan dalam Gambar 8 diatas merupakan tampilan dari salah satu contoh eksekusi aturan submenu untuk cetak KHS mahasiswa dengan ketentuan permintaan data seperti pada rancangan aturan.

D. Hasil Pengujian

Dari hasil pengujian yang dilakukan didapatkan hasil dengan tingkat akurasi hingga 98.95% dengan kesesuaian proses penanganan dari 96 kasus ujicoba. Rangkuman dari ujicoba tersebut dapat dilihat pada Tabel II di bawah.

TABEL II
RANCANGAN ATURAN

No.	Aturan	Jumlah Pengujian	Ket.
1	Menu dan menu handler	26 kali	(26) ✓ (0) ✗
2	Submenu handler	13 kali	(13) ✓ (0) ✗
3	Tanya dan cek NIM	14 kali	(14) ✓ (1) ✗
4	Tanya semester	2 kali	(2) ✓ (0) ✗
5	Tanya dan cek PIN	18 kali	(18) ✓ (0) ✗
6	Sanggah BAAK	1 kali	(1) ✓ (0) ✗
7	Sanggah Keuangan	4 kali	(4) ✓ (0) ✗
8	Cari dan kirim dokumen	3 kali	(3) ✓ (0) ✗
9	Kumpul dan redirect proposal	1 kali	(1) ✓ (0) ✗
10	Ubah, konfirmasi dan simpan PIN baru	14 kali	(14) ✓ (0) ✗
Total Pengujian		96 kali	(95) ✓ (1) ✗

Tabel II di atas menunjukkan bahwa pengujian dilakukan sebanyak 96 kali dalam berbagai bentuk skema yang mungkin terjadi, dan didapat 1 kali kesalahan selama pengujian tersebut. Sehingga jika dimasukkan kedalam rumus perhitungan akurasi adalah sebagai berikut:

$$akurasi = \frac{95}{96} \times 100\% = 98,95\%$$

Gbr. 9 Hasil Perhitungan Akurasi

Kesalahan yang terjadi diakibatkan gagalnya eksekusi pengecekan NIM ketika dilakukan berkali-kali secara berturut-turut, hal ini terjadi dikarenakan pada saat sistem meminta NIM dibuatkan sebuah *variable* sementara guna menyimpan NIM dan informasi lain yang sering dibutuhkan agar mempercepat proses dalam menjawab (eksekusi) proses lain terkait informasi tersebut. Hal ini juga menjadi kelemahan jika terdapat banyak pengguna yang menggunakan atau mengakses *bot* secara bersamaan yang kemungkinan proses sistem akan memakan memory yang berlebih.

V. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan dalam rancang-bangun *telegram chatbot* STMIK Muhammadiyah Paguyangan Brebes berbasis aturan dapat disimpulkan bahwa penerapan aturan ini memiliki akurasi yang tinggi dalam ketepatan eksekusi proses (menjawab permintaan) hingga 98,95% ketepatan jawaban.

Namun dibalik tingginya akurasi tersebut juga memiliki kelemahan dalam segi kecepatan jika aturan yang diterapkan banyak. Juga memiliki kekurangan dalam segi fleksibilitas, karena *bot* hanya akan menganggapi jika permintaan sesuai dengan menu yang disediakan, kecuali jika menambahkan fungsi untuk *Natural Language Processing* sehingga *bot* dapat memahami percakapan bahasa alami.

DAFTAR PUSTAKA

- [1] A. Augello, G. Pilato, G. Vassallo, and S. Gaglio, "Chatbots as Interface to Ontologies," *Adv. Intell. Syst. Comput.*, vol. 260, pp. 285–299, 2014.
- [2] F. Ishlakhuddin, "Chatbot Berbasis Ontologi untuk Mendukung Pemantauan Kinerja dan Keamanan Server dengan Rule-Base," Universitas Gadjah Mada, 2020.
- [3] J. Allen, *Natural Language Understanding (2nd Ed.)*. Redwood City, CA, USA: Benjamin-Cummings Publishing Company, 1995.
- [4] P. Kordjamshidi, J. Hois, M. van Otterlo, and M.-F. Moens, "Learning to Interpret Spatial Natural Language in terms of Qualitative Spatial Relations," *Represent. Sp. Cogn.*, pp. 115–146, 2014.
- [5] M. H. P. Swari, "Sistem Diagnosis Penyakit Gigi dan Mulut menggunakan Kombinasi Case Based Reasoning dan Rule Based Reasoning," Universitas Gadjah Mada, 2014.
- [6] H. R. Sakinnah, "Rule Based System dengan Prediksi Learning Style Berdasarkan Metakognitif," 2016.
- [7] S. Quarteroni and S. Manandhar, "A Chatbot-based Interactive Question Answering System," *Decalog*, pp. 83–90, 2007.
- [8] R. K. J. Bendi, "Sistem question answering sederhana berbasis ontologi sebagai aplikasi web semantik," Universitas Gadjah Mada, 2010.
- [9] I. W. D. Suryawan, "Sistem Question Answering Menggunakan Pendekatan Berbasis Pengetahuan," Universitas Gadjah Mada, 2013.
- [10] M. I. Siami, "Sistem Question Answering Fiqih Berbasis Ontologi dengan Klasifikasi Pertanyaan menggunakan Support Vector Machine," Universitas Gadjah Mada, 2014.
- [11] S. Muttaqin, "Sistem Question Answering Data Kabupaten Di Nusa Tenggara Barat Berbasis Multi-Ontologi," Universitas Gadjah Mada, 2016.
- [12] A. Winantu, "Pengembangan Purwarupa Semantic Search menggunakan Semantic Web Rule Language untuk Pencarian Informasi Produk," Universitas Gadjah Mada, 2014.