

Rancang Bangun *Virtual Assistant Chatbot* Menggunakan Node.Js pada Layanan Sistem Informasi Akademik

Gusti Putu Mahendra Putra¹, Andi Tenriawaru^{*2}, Gunawan³

^{1,2,3}Program Studi Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Halu Oleo, Kendari

e-mail: ¹gusti.rplcsmkn4kdi@gmail.com, ^{*2}andi.tenriawaru@uho.ac.id, ³gunawan@uho.ac.id

Abstrak

Penelitian ini bertujuan untuk membangun sebuah virtual assistant *chatbot* menggunakan node.js pada layanan sistem informasi akademik. *Chatbot* ini diharapkan dapat menjadi alternatif cara yang lebih efisien untuk mengakses sistem informasi akademik.

Chatbot dibangun menggunakan node.js dan memanfaatkan whatsapp-web.js untuk bisa terhubung ke aplikasi WhatsApp. Pendekatan rule-based system memungkinkan respons yang terstruktur dan konsisten saat pengguna berinteraksi dengan *chatbot*. Integrasi dengan layanan sistem informasi akademik memungkinkan *chatbot* memberikan informasi tentang status mahasiswa, jadwal kuliah, status pembayaran UKT, IPS dan IPK mahasiswa dengan cepat dan tepat.

Pengujian dan pengukuran waktu respon dilakukan untuk mengevaluasi kinerja *chatbot* menggunakan Apache Jmeter. Hasil pengujian menunjukkan bahwa *chatbot* responsif dan dapat diandalkan dalam memberikan respons kepada pengguna.

Kata kunci— chatbot, WhatsApp, Sistem Informasi Akademik.

Abstract

This research aims to build a virtual assistant chatbot using node.js on academic information system services. This chatbot is expected to be an alternative and more efficient way to access the academic information system.

The chatbot is built using node.js and utilizes whatsapp-web.js to connect to the WhatsApp application. The rule-based system approach allows a structured and consistent response when users interact with the chatbot. Integration with academic information system services allows the chatbot to provide information about student status, class schedule, UKT payment status, IPS and student GPA quickly and precisely.

Testing and response time measurements were conducted to evaluate the chatbot performance using Apache Jmeter. The test results show that the chatbot is responsive and reliable in providing responses to users.

Keywords—Academic information system, chatbot, WhatsApp.

1. PENDAHULUAN

Sistem Informasi Akademik (SIKAD) dalam perguruan tinggi, sangat penting dalam memberikan layanan kepada mahasiswa. SIKAD dapat membantu pengelolaan informasi dan administrasi secara efektif dan efisien. Berbagai jenis layanan yang digunakan

untuk melayani kebutuhan akademik seperti menggunakan telepon, chatting dan lain-lain [1]. Namun, saat mengelola SIKAD, masalah seperti waktu respon yang lambat dan kemampuan pengelolaan data yang terbatas seringkali masih terjadi.

Beberapa tahun terakhir, chatbot telah menjadi salah satu solusi untuk membantu

mengatasi kendala dalam pengelolaan SIAKAD. Pengguna dapat berkomunikasi dengan chatbot melalui pesan teks [2]. Dalam memahami dan menjawab pesan, chatbot membutuhkan metode tertentu untuk dapat memberikan respon cepat dan akurat. Rule-based system menjadi metode yang tepat untuk mengidentifikasi kalimat pertanyaan sekaligus menentukan jawabannya [3].

Dengan Rule-based system, chatbot akan lebih mudah dan lebih cepat dibangun karena tidak memerlukan proses pelatihan seperti pendekatan Natural Language Processing (NLP). Untuk proses penggunaannya chatbot biasanya digunakan pada beberapa platform media sosial seperti Facebook, Telegram, LINE, WhatsApp dan masih banyak lagi.

Menurut we are social, WhatsApp masih menjadi media sosial yang paling banyak digunakan oleh masyarakat Indonesia saat ini dengan persentase mencapai 92,1% per Januari 2023. WhatsApp hampir sama dengan aplikasi Short Message Service (SMS) yang biasa digunakan di ponsel lama, tetapi WhatsApp tidak menggunakan pulsa, melainkan data internet serta memiliki banyak fitur yang tidak dimiliki oleh SMS. WhatsApp juga menyediakan client library yang dapat digunakan oleh pengembang untuk mengembangkan chatbot [4].

Salah satu yang dapat digunakan adalah WhatsApp-web.js, dimana client library ini menggunakan Node.js untuk terhubung melalui aplikasi WhatsApp Web browser. WhatsApp-web.js ini dapat digunakan secara gratis dan memiliki dokumentasi yang lengkap.

Oleh karena itu, dalam penelitian ini akan merancang dan mengembangkan chatbot menggunakan Node.js pada layanan sistem informasi akademik. Chatbot yang akan dirancang menggunakan metode rule-based system yang dapat disesuaikan dengan kebutuhan sistem informasi akademik.

Rumusan masalah dari penelitian ini adalah bagaimana membuat virtual assistant chatbot menggunakan Node.js pada layanan sistem informasi akademik?

Adapun tujuan dari penelitian ini adalah menghasilkan virtual assistant chatbot menggunakan Node.js pada layanan sistem informasi akademik.

Secara harfiah, *chat* dapat diartikan sebagai obrolan, namun dalam dunia komputer istilah ini merujuk pada kegiatan komunikasi melalui sarana

baris-baris tulisan yang diketikkan melalui *keyboard*. *Bot* merupakan sebuah program yang mengandung sejumlah data di mana jika diberikan masukan maka akan memberikan jawaban [5].

Chatbot biasanya berjalan pada aplikasi messenger populer, seperti Facebook Messenger, Twitter, WhatsApp, dan Telegram [6]. Terdapat 2 (dua) tipe *chatbot* :

- 1) *Chatbot* yang beroperasi berdasarkan sekumpulan aturan. Hanya dapat merespon berdasarkan command yang spesifik. Jika seseorang menggunakan selain command atau kata yang sudah ditentukan, maka *chatbot* tidak akan mengerti maksud orang tersebut.
- 2) *Chatbot* yang menggunakan *Machine Learning* (ML) dan *Artificial Intelligence* (AI) untuk menyediakan respon terbaik. Biasa dinamakan dengan *AI-powered chatbot*.

Sistem informasi merupakan program perangkat lunak yang dibangun sesuai dengan kebutuhan suatu organisasi untuk mendukung fungsi operasi organisasi dengan tujuan dapat mempermudah pengaksesan informasi yang dibutuhkan oleh organisasi tersebut maupun pihak lain yang berkepentingan [7].

Node.js adalah lingkungan *runtime* JavaScript *open-source* dan *cross-platform*. Ini adalah *platform* yang populer untuk hampir semua jenis proyek. Node.js berjalan dimesin JavaScript V8 sehingga dapat mengeksekusi kode JavaScript di luar *browser* web. Node.js dirancang untuk membangun jaringan aplikasi yang *scalable* sehingga sangat cocok untuk membuat aplikasi jaringan *real-time*, seperti *server* web, *server game*, *server chatting* dan sebagainya [8].

WhatsApp-web.js merupakan *library client* WhatsApp untuk Node.js yang terhubung melalui aplikasi *browser* Web WhatsApp. WhatsApp-web.js memungkinkan pengembang untuk membuat *chatbot* WhatsApp dengan mengirimkan pesan teks, mengirim pesan media, dan melakukan beberapa operasi lainnya.

Rule-based system dibuat untuk memecahkan masalah dengan aturan yang dibuat berdasarkan pengetahuan dari pakar. Aturan tersebut memiliki kondisi (*if*) dan tindakan (*then*). Peraturan-peraturan tersebut akan di masukkan ke dalam mesin aplikasi. Mesin akan mencocokkan dengan pengaturan yang ada dan menentukan aturan yang berhubungan. *Rule-based* mudah untuk digunakan dan dimengerti, namun *rule-based* tidak dapat membuat peraturan baru atau memodifikasi peraturan yang ada dengan

sendirinya karena *rule-based* tidak dirancang untuk dapat belajar [9].

Black box testing merupakan pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Pengujian *black box* ini menitik beratkan pada fungsi sistem. Metode ini digunakan untuk mengetahui apakah aplikasi berfungsi dengan benar atau tidak [10].

Performance testing adalah jenis pengujian untuk memastikan perangkat lunak akan bekerja dengan baik di bawah beban kerja yang diharapkan. Tujuan utamanya bukan untuk mencari *bug*, tapi untuk mengeliminasi *performance bottleneck* [11].

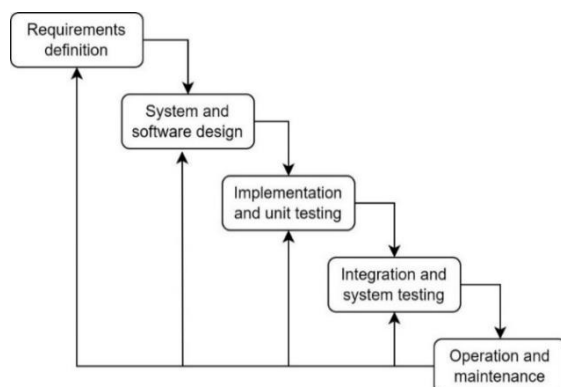
2. METODE PENELITIAN

2.1 Waktu dan Tempat Penelitian

Penelitian ini dilaksanakan pada bulan Mei 2023 hingga Juli 2023. Lokasi untuk melakukan penelitian ini adalah Laboratorium Unit Teknologi Informatika yang berada di Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Halu Oleo, Kendari, Sulawesi Tenggara.

2.2 Prosedur Penelitian

Prosedur penelitian adalah tahapan-tahapan yang dilakukan dalam proses penelitian untuk menyelesaikan sebuah masalah. Prosedur dapat membuat penelitian lebih terarah dan sistematis. Prosedur pada penelitian ini mengadaptasi model pengembangan sistem *waterfall*. Prosedur penelitian dapat dilihat pada Gambar 1.



Gambar 1 Prosedur Penelitian

Berikut adalah penjelasan langkah-langkah prosedur penelitian berdasarkan Gambar 1.

1. *Requirements analysis and definition*, Tahap ini dilakukan untuk menganalisis kebutuhan dan persyaratan pengguna yang diperoleh dari studi pustaka dan observasi mengenai *chatbot* yang akan dibangun.
2. *System and software design*, Pada tahap ini, peneliti akan merancang desain interaksi antara pengguna dan *chatbot*, merancang alur percakapan, memilih platform Node.js, serta membuat skenario pengujian sistem. Desain tersebut akan menjadi acuan dalam tahap berikutnya.
3. *Implementation and unit testing*, Pada tahap ini dilakukan untuk mengimplementasikan desain yang telah dibuat pada tahap sebelumnya, dalam bentuk kode program. Pada tahap ini, peneliti akan mengimplementasikan desain yang telah dibuat dan melakukan unit testing untuk memastikan setiap komponen berfungsi dengan baik.
4. *Integration and system testing*, Pada tahap ini komponen-komponen yang telah dibuat akan diintegrasikan dan diuji sebagai sebuah sistem menggunakan *black box testing*. Hal ini dilakukan untuk memastikan bahwa keseluruhan sistem berfungsi dengan baik sesuai dengan persyaratan yang telah ditentukan sebelumnya.
5. *Operation and maintenance*, Setelah sistem diuji dan berfungsi dengan baik, tahap ini dilakukan untuk menjaga kinerja sistem dan memperbaiki *bug* yang terdeteksi. Peneliti akan melakukan pemeliharaan rutin dan pembaruan sistem secara berkala agar sistem tetap berjalan dengan baik dan memenuhi kebutuhan pengguna.

2.3 Analisis Kebutuhan Sistem

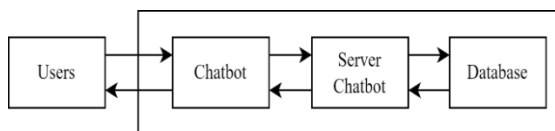
Analisis kebutuhan sistem yang peneliti lakukan yaitu mempertimbangkan berbagai aspek untuk memastikan *chatbot* dapat memenuhi kebutuhan pengguna dengan baik. Dari proses tersebut didapatkan kebutuhan fungsional yang akan dimasukkan ke dalam bot WhatsApp yang dikembangkan. *Chatbot* harus mampu merespon pertanyaan pengguna mengenai status akademik, jadwal perkuliahan, IPS serta IPK mereka.

Integrasi dengan database juga menjadi bagian penting dalam analisis kebutuhan

sistem. Peneliti menggunakan database MySQL sebagai sumber data yang berisi informasi mahasiswa dan entitas terkait. Peneliti mengidentifikasi entitas dan atribut yang perlu diakses oleh chatbot untuk memberikan informasi yang akurat dan relevan kepada pengguna. Peneliti juga memperhatikan performa chatbot dalam analisis kebutuhan. Peneliti akan menguji dan mengukur waktu respons chatbot untuk memastikan kinerjanya yang baik dan waktu respons yang cepat.

2.4 Rancangan Sistem

Rancangan sistem merupakan suatu konsep yang memberikan pandangan umum mengenai arsitektur dan hubungan antar komponen dalam suatu sistem. Hal ini berguna untuk memastikan bahwa sistem yang akan dibangun dapat memenuhi kebutuhan pengguna. Rancangan sistem ini dapat dilihat pada Gambar 2.



Gambar 2 Rancangan Sistem

Tahapan pertama yaitu proses pengiriman *request* oleh users atau pengguna kenomor WhatsApp yang telah terintegrasi dengan *chatbot*. *Request* tersebut berupa pesan yang akan diteruskan ke *server chatbot* untuk diidentifikasi kategori atau topiknya. Jika pesan tersebut *valid* maka akan dilakukan *request* data ke *database* dan akan dikembalikan sesuai dengan data yang diminta. Jika pesan tersebut tidak *valid* maka *chatbot* akan memberikan respon yang sesuai dengan pesan dari pengguna, baik berupa informasi atau tindakan yang diperlukan kepada pengguna.

2.5 Skenario Pengujian Sistem

Skenario pengujian sistem adalah serangkaian langkah konkret yang dirancang untuk menguji fungsionalitas, kinerja dan fitur-fitur sistem secara sistematis. Setiap skenario pengujian sistem terdiri dari langkah-langkah yang ditentukan dengan jelas, termasuk input yang diberikan ke sistem, tindakan yang diambil, dan hasil yang diharapkan. Skenario pengujian sistem dapat dilihat pada Tabel 1.

Tabel 1 Skenario Pengujian Sistem

Perintah	Pengujian		Tujuan
	Sistem	Waktu	
status	Pengguna mengirim pesan berisi perintah "status FIG119026" ke chatbot, di mana " FIG119026" adalah Nomor Induk Mahasiswa (NIM) mahasiswa yang valid.	Mengukur waktu respons chatbot untuk merespons permintaan status mahasiswa.	Memastikan bahwa chatbot dapat memberikan status mahasiswa berdasarkan NIM yang diberikan oleh pengguna.
bayar	Pengguna mengirim pesan berisi perintah "bayar FIG119026" ke chatbot, di mana " FIG119026" adalah NIM mahasiswa yang valid.	Mengukur waktu respons chatbot untuk merespons permintaan bayar mahasiswa.	Memastikan bahwa chatbot dapat memberikan status pembayaran UKT mahasiswa berdasarkan NIM yang diberikan oleh pengguna.
jadwal	Pengguna mengirim pesan berisi perintah "jadwal FIG119026" ke chatbot, di mana " FIG119026" adalah NIM mahasiswa yang valid.	Mengukur waktu respons chatbot untuk merespons permintaan status mahasiswa.	Memastikan bahwa chatbot dapat memberikan jadwal mahasiswa berdasarkan NIM yang diberikan oleh pengguna.
ips	Pengguna mengirim pesan berisi perintah "ips FIG119026 1" ke chatbot, di mana " FIG119026" adalah NIM mahasiswa yang valid dan "1" adalah Semester yang ingin dicek.	Mengukur waktu respons chatbot untuk merespons permintaan ips mahasiswa.	Memastikan bahwa chatbot dapat memberikan ips mahasiswa berdasarkan NIM dan semester yang diberikan oleh pengguna.
ipk	Pengguna mengirim pesan berisi perintah "ipk FIG119026" ke chatbot,	Mengukur waktu respons chatbot untuk	Memastikan bahwa chatbot dapat memberikan ipk

Perintah	Pengujian		Tujuan
	Sistem	Waktu	
	di mana " F1G119026" adalah NIM mahasiswa yang valid.	merespons permintaan ipk mahasiswa.	mahasiswa berdasarkan NIM yang diberikan oleh pengguna.

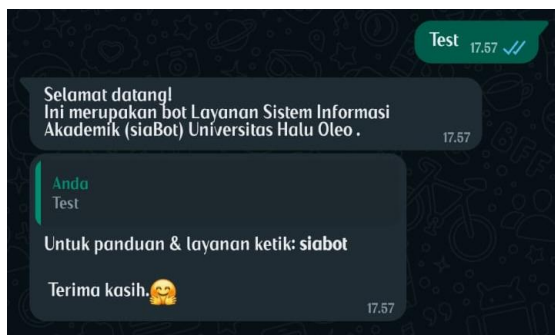
3. HASIL DAN PEMBAHASAN

3.1 Implementasi

Tahap ini, peneliti akan menerapkan hasil perancangan sebelumnya ke dalam bentuk chatbot. Pembuatan chatbot dibagi menjadi unit kecil yang nantinya akan digabungkan dalam tahap selanjutnya. Chatbot diimplementasikan menggunakan bahasa pemrograman Javascript dengan Node.js sebagai runtime-nya, WhatsApp-web.js, xampp dan juga MySQL.

a. Akses Chatbot

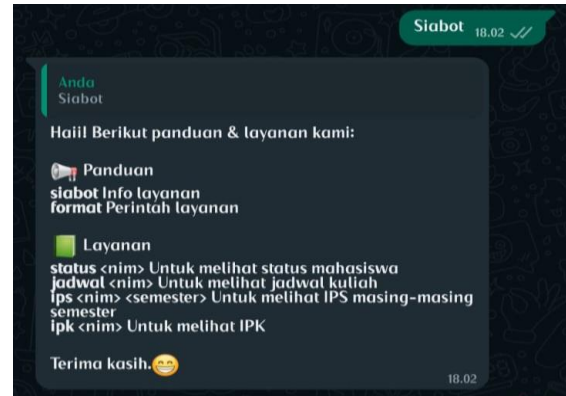
Akses chatbot, saat pengguna pertama kali menggunakan chatbot responnya berupa ucapan selamat datang dan juga memberitahukan bahwa untuk melihat panduan dan layanan chatbot cukup dengan mengetikkan "siabot". Untuk implementasi akses chatbot dapat dilihat pada Gambar 3.



Gambar 3 Akses Layanan Chatbot

b. Akses Layanan SiaBot

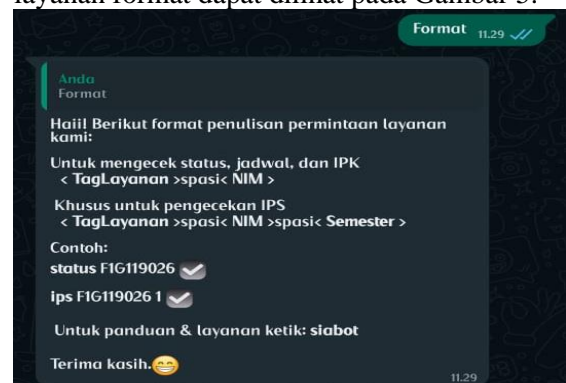
Akses layanan siabot, pada layanan ini terdapat informasi mengenai panduan dan layanan chatbot. Layanan ini berisi beberapa perintah yang dapat digunakan untuk mengakses chatbot. Untuk implementasi layanan siabot dapat dilihat pada Gambar 4.



Gambar 4 Akses Layanan SiaBot

c. Akses Layanan Format

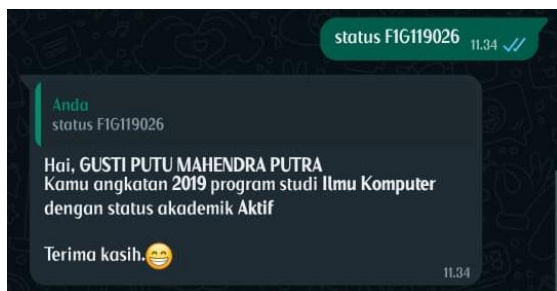
Akses layanan format, pada layanan ini terdapat informasi mengenai format pengetikan layanan yang benar. Untuk implementasi layanan format dapat dilihat pada Gambar 5.



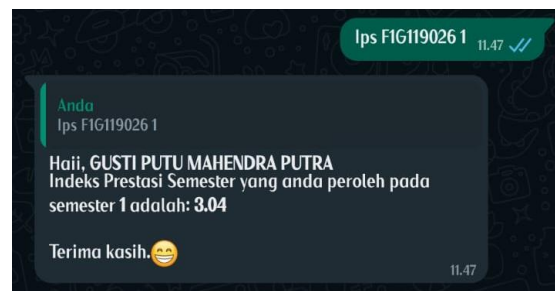
Gambar 5 Akses Layanan Format

d. Akses Layanan Status

Akses layanan status, pada layanan ini terdapat informasi mengenai status dari mahasiswa yang dicek oleh pengguna. Layanan ini berisi nama lengkap, angkatan, program studi dan status akademik mahasiswa. Untuk implementasi layanan status dapat dilihat pada Gambar 6.



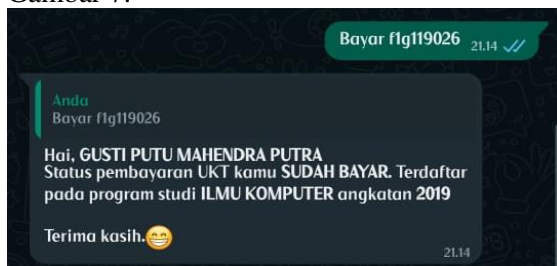
Gambar 6 Akses Layanan Status



Gambar 9 Akses Layanan IPS

e. Akses Layanan Bayar

Pada layanan ini terdapat informasi mengenai status pembayaran UKT dari mahasiswa yang dicek. Layanan ini berisi nama lengkap, status pembayaran UKT, program studi dan angkatan mahasiswa. Untuk implementasi layanan bayar dapat dilihat pada Gambar 7.



Gambar 7 Akses Layanan Bayar

h. Akses Layanan IPK

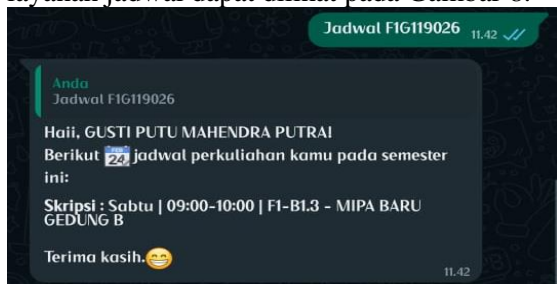
Akses layanan siabot, pada layanan ini terdapat informasi mengenai panduan dan layanan chatbot. Layanan ini berisi beberapa perintah yang dapat digunakan untuk mengakses chatbot. Untuk implementasi layanan siabot dapat dilihat pada Gambar 10.



Gambar 10 Akses Layanan IPK

f. Akses Layanan Jadwal

Pada layanan ini terdapat informasi mengenai jadwal dari mahasiswa yang dicek. Layanan ini berisi nama lengkap dan jadwal perkuliahan mahasiswa. Untuk implementasi layanan jadwal dapat dilihat pada Gambar 8.



Gambar 8 Akses Layanan Jadwal

g. Akses Layanan IPS

Pada layanan ini terdapat informasi mengenai IPS dari mahasiswa yang dicek. Layanan ini berisi nama lengkap, semester dan IPS yang diperoleh. Untuk implementasi layanan IPS dapat dilihat pada Gambar 9.

3.2 Pengujian

Black Box Testing, fokus utamanya adalah untuk menguji fungsionalitas dan perilaku aplikasi tanpa memperhatikan rincian implementasi internal. Pengujian ini hanya berfokus pada input dan output, serta interaksi dengan antarmuka pengguna, tanpa mengetahui cara kerja aplikasi di dalamnya. Pengujian ini bertujuan untuk memastikan bahwa aplikasi berfungsi dengan benar sesuai dengan persyaratan fungsional yang telah ditentukan sebelumnya, dan mengidentifikasi potensi masalah dari perspektif pengguna.

a. Pengujian Perintah Status

Pengujian pada perintah ini yaitu, pengguna memasukkan <Perintah>spasi<NIM> dari salah satu mahasiswa yang ingin dicek. Sebagai contoh pengguna memasukkan "status FIG19026".

b. Pengujian Perintah Bayar

Pengujian pada perintah ini yaitu, pengguna memasukkan <Perintah>spasi<NIM> dari salah satu mahasiswa yang ingin dicek. Sebagai contoh pengguna memasukkan “bayar F1G119026”.

c. Pengujian Perintah Jadwal

Pengujian pada perintah ini yaitu, pengguna memasukkan <Perintah>spasi<NIM> dari salah satu mahasiswa yang ingin dicek. Sebagai contoh pengguna memasukkan “jadwal F1G119026”.

d. Pengujian Perintah IPS

Pengujian pada perintah ini sedikit berbeda yaitu, pengguna memasukkan <Perintah>spasi<NIM>spasi<Semester> dari salah satu mahasiswa yang ingin dicek. Sebagai

contoh pengguna memasukkan “IPS F1G119026 1”.

e. Pengujian Perintah IPK

Pengujian pada perintah ini sedikit berbeda yaitu, pengguna memasukkan <Perintah>spasi<NIM> dari salah satu mahasiswa yang ingin dicek. Sebagai contoh pengguna memasukkan “IPK F1G119026”.

f. Performance Testing

Pengujian ini dilakukan sebanyak 3 kali dengan penambahan thread (user) mulai dari 100, 500, 1000 sampai dengan 5000 dengan waktu jeda 20 detik. Pengujian ini dilakukan pada 5 perintah, yaitu status, bayar, jadwal, IPS dan IPK. Hasil pengujian waktu respon (millisecond) ini dapat dilihat pada Tabel 2.

Tabel 2 Hasil Pengujian Waktu Respon (m/s)

Perintah	Chatbot WhatsApp				Web			
	Threads (pengguna)				Threads (pengguna)			
	100	500	1000	5000	100	500	1000	5000
Status	8	2	9	672	625	735	1041	81723
Bayar	7	3	9	285	674	979	908	96770
Jadwal	8	4	8	4002	633	714	724	16825
IPS	9	5	8	6266	618	647	733	81072
IPK	10	8	10	6496	Tidak memiliki halaman IPK			

Berdasarkan Tabel hasil perbandingan waktu akses diatas didapatkan bahwa jika jumlah threads(pengguna) meningkat dari 100 sampai 5000, waktu respon pada perintah Status juga meningkat mulai dari 2/ms hingga 672/ms melalui *chatbot* sedangkan melalui web waktu responnya meningkat mulai dari 625/ms hingga 81723/ms. Pada perintah Bayar waktu respon meningkat mulai dari 3/ms hingga 285/ms melalui *chatbot* sedangkan melalui web waktu responnya meningkat mulai dari 674/ms hingga 96770/ms. Pada perintah Jadwal waktu respon meningkat mulai dari 4/ms hingga 4002/ms melalui *chatbot* sedangkan melalui web waktu responnya meningkat mulai dari 633/ms hingga 16825/ms. Pada perintah IPS waktu respon meningkat mulai dari 5/ms hingga 6266/ms melalui *chatbot* sedangkan melalui web waktu responnya meningkat mulai dari 618/ms hingga 81072/ms. Terakhir pada perintah IPK waktu responnya meningkat mulai dari 8/ms hingga 6496/ms melalui

chatbot sedangkan melalui web tidak ada karena tidak memiliki halaman yang menampilkan IPK.

Berdasarkan Tabel hasil perbandingan waktu akses juga didapatkan data jika jumlah threads(pengguna) meningkat mulai dari 100 hingga 5000 waktu respon yang didapatkan oleh *chatbot* tidak meningkat secara berurutan melainkan bervariasi. Pada seluruh perintah *chatbot* mendapatkan waktu respon tercepat di 500 pengguna dan waktu respon terlambat di 5000 pengguna. Berbeda halnya melalui web, peningkatan jumlah threads(pengguna) mempengaruhi waktu respon yang didapatkan. Semakin banyak jumlah threads(pengguna) maka akan semakin lama waktu respon yang didapatkan juga.

Kesimpulannya, waktu pengaksesan data di SIAKAD dengan menggunakan *chatbot* lebih cepat dibandingkan pengaksesan melalui web. Hal ini disebabkan oleh *chatbot* yang langsung mengambil data yang ada pada

database, tanpa perlu memuat berbagai file lain seperti gambar atau teks yang menyusun halaman web. Inilah yang membuat *chatbot* memiliki performa yang baik.

4. KESIMPULAN

Berdasarkan hasil penelitian, maka dapat ditarik kesimpulan bahwa virtual assistant *chatbot* menggunakan node.js pada layanan system informasi akademik berhasil dibangun. Performa pengaksesan data yang dihasilkan melalui *chatbot* WhatsApp sangat baik dibandingkan melalui web, disebabkan oleh *chatbot* WhatsApp yang langsung mengambil data yang ada pada database, tanpa perlu memuat berbagai file lain seperti gambar atau teks yang menyusun halaman web. Dengan adanya *chatbot* ini dapat menjadi virtual assistant yang baik sehingga masalah seperti waktu respon yang lambat dapat teratasi.

5. SARAN

Adapun Saran dari penelitian ini adalah Secara umum *chatbot* ini memiliki hasil yang sangat baik, namun masih ada beberapa hal yang perlu ditambahkan ataupun diperbaiki agar dapat berfungsi lebih lagi. Adapun saran yang diberikan untuk penelitian selanjutnya sebagai berikut.

1. Diharapkan *chatbot* yang akan dikembangkan selanjutnya itu mampu memahami langsung permintaan pengguna tanpa menggunakan perintah atau rule dengan memanfaatkan salah satu algoritma NLP.
2. Diharapkan *chatbot* dapat diintegrasikan pada suatu sistem dengan cakupan yang lebih luas.

DAFTAR PUSTAKA

- [1] G. Guntoro, Loneli Costaner, And L. Lisnawita, "Aplikasi *Chatbot* Untuk Layanan Informasi Dan Akademik Kampus Berbasis Artificial Intelligence Markup Language (Aiml)," *Digit. Zo. J. Teknol. Inf. Dan Komun.*, Vol. 11, No. 2, Pp. 291–300, 2020, Doi: 10.31849/Digitalzone.V11i2.5049.
- [2] S. Rubaeah, T. T. W. Ningrum, Z. W. Fandol, And R. A. Setiawan, "Sispac: *Chatbot* Untuk Diagnosis Dan Penanganan Hipertensi," *Pros. Semin. Nas. Din. Inform. 2021*, Pp. 100–105, 2021.
- [3] F. Ishlakhuddin, A. Basir, And Nurlaela, "Rancang Bangun Sistem Tanya-Jawab Berbasis Aturan Stmik Muhammadiyah Paguyangan Brebes Dengan Menggunakan Telegram *Chatbot*," *J. Inform.*, Vol. 5, No. 3, Pp. 100–105, 2020.
- [4] R. Parluka, S. I. Pradika, A. M. Hakim, And K. R. N. Manab, "Bot Whatsapp Sebagai Pemberi Data Statistik Covid-19 Menggunakan Php, Flask, Dan Mysql," *J. Inform. Dan Sist. Inf.*, Vol. 1, No. 2 Se-Articles, Pp. 282–293, 2020, [Online]. Available: [Http://jifosi.upnjatim.ac.id/index.php/jifosi/article/view/101](http://jifosi.upnjatim.ac.id/index.php/jifosi/article/view/101)
- [5] E. Utami And S. Hartati, "Aplikasi Botqa Untuk Meningkatkan Cara Interaksi Manusia Dan Mesin," *Semin. Nas. Apl. Teknol. Inf. ...*, Vol. 2007, No. Snati, Pp. 1–8, 2007, [Online]. Available: [Https://journal.uui.ac.id/snati/article/download/1618/1393](https://journal.uui.ac.id/snati/article/download/1618/1393)
- [6] A. Zubaidi And Ramdani, "Layanan Dan Informasi Akademik Berbasis Bot Telegram Di Program Studi Teknik Informatika Universitas Mataram," *J. Ilm.*, Vol. 1, No. 1, Pp. 103–110, 2019.
- [7] Muhaemin, "Pengembangan Fungsionalitas Sistem Informasi Dengan Pendekatan Kansei Engineering," *Infotronik J. Teknol. Inf. Dan Elektron.*, Pp. 43-47., 2020.
- [8] Putri, D. J. Valentino, And R. Andarsyah, "Cara Praktis Membuat *Chatbot* Whatsapp," 2023.
- [9] F. Hayes-Roth, "Rule-Based Systems," *Commun. Acm*, Vol. 28, No. 9, Pp. 921–932, 1985, Doi: 10.1145/4284.4286.
- [10] R. M. Syaban And H. Bunyamin, "Pengembangan Sistem Informasi Pengelolaan Surat Masuk Dan Surat Keluar Berbasis Web Di Dinas Sosial Tenaga Kerja Dan Transmigrasi Kabupaten Garut Menggunakan," Pp. 301–311, 2015.
- [11] D. I. Permatasari, "Pengujian Aplikasi Menggunakan Metode Load Testing Dengan Apache Jmeter Pada Sistem Informasi Pertanian," *J. Sist. Dan Teknol. Inf.*, Vol. 8, No. 1, P. 135, 2020, Doi: 10.26418/Justin.V8i1.34452.