

Designing Consultation Chatbot Using Telegram API and Webhook-based NodeJS Applications

1st Kristian Adi Nugraha*
Faculty of Information Technology
Universitas Kristen Duta Wacana
 Yogyakarta, Indonesia
 adinugraha@ti.ukdw.ac.id

2nd Danny Sebastian
Faculty of Information Technology
Universitas Kristen Duta Wacana
 Yogyakarta, Indonesia
 dannysebastian@staff.ukdw.ac.id

Abstract—Chatbots are a type of technology that has been used in a variety of areas in recent years, especially as the use of online media has become massively. The development of a chatbot application cannot be separated from the usage of server architecture. Because chatbots, in general, require quite complex processing, so the chosen server architecture and components also determining factors in whether or not the chatbot is capable of performing its tasks. This study tries to discuss designing a chatbot using the webhook concept, where the webhook server will only work when there is a message from the user that needs to be replied to. This concept can reduce the server workload, so the server can focus on the chatbot algorithm to be implemented.

Keywords—*chatbot, webhook, consultation*

I. INTRODUCTION

Artificial intelligence (AI) has gained tremendous popularity in recent years, and there are a lot of applications that employ this technology to replace humans' roles for tasks. Computers, mobile phones, robots [1], and everyday objects that have embedded computers inside known as the internet of things [2] are all kinds of AI implementation. For example, a voice and text-based automatic answering system based on natural language processing, include Apple's Siri (voice-based) and chatbots (text-based) which are frequently seen in customer-facing web or mobile applications. Artificial intelligence technology can provide benefits such as resource savings (both human and financial) and time savings, as a result, many companies have begun to shift from traditional to digital technology, aided by artificial intelligence [3].

Many activities, especially since the Covid-19 outbreak, must be doing on the online platform. As a result, many transactions that were previously conducted face-to-face have been forced to transition to an online form. Not only for buying and selling transactions but also in conversation-based services like counseling (psychologist) or consultation (teacher) like

an e-learning system [4]. As a result, this will increase their workload because online media allows people from all over the world to communicate at any time and from any location simultaneously, regardless of distance or time. This is the primary motivation for the development of chatbots, which are digital assistants that can replace individuals while conversing with others.

Chatbots are becoming increasingly popular and are being implemented by a wide range domain of businesses because the instant messaging (chat) application is currently the most commonly used way of communication by the entire world's population [5]. Especially since the existence of smartphones, the gap between mobile and PC applications is getting smaller [6], so people can access all kinds of applications through smartphones. Almost all existing mobile or web apps, like social media, games, or marketplace, allow users to send instant messages in a variety of forms, such as private messages or direct messages, making it simple to integrate chatbots inside them. The chatbot is built on a client-server architecture, with the client is an application that the end-user uses directly, and the server providing a chatbot engine that responds to all messages sent by the client. Database components and algorithms on the server are responsible for processing communications from clients. The contents of the database and the type of algorithm utilized by the server will affect the response given to the client.

The important components to build a chatbot are not only artificial intelligence but also the infrastructure technology used, including the server and the tools inside it [7]. To create a chatbot application, a variety of infrastructure combinations can be used. The goal of this study is to design a chatbot based on the concept of a webhook. In comparison to other instant messaging apps that often using WebSocket technology, webhook apps use a slightly different approach. Webhook is a server-to-server communication that only works when it is triggered by a certain event, meanwhile,

WebSocket is a technology that allows persistent communication on client-server to be occurred [8]. As a result, the webhook shares the workload among various servers based on their responsibilities, ensuring that no server is overloaded. That is very important considering that chatbots require algorithms that require quite a lot of resources, so all of the workloads should not be handled by only one machine. This research will use the Telegram API as the chatbot media platform, with the combination of a custom server (NodeJS-based) to processing input messages.

II. RELATED WORKS

Many earlier studies have covered chatbots and provided an overview of the architecture. Previous research related to chatbots development can be divided into three categories.

A. Service-Oriented Architecture

One of the things to consider while building software is to choose to most suitable software architecture. The software can be developed in many ways depending on the platform [9], such as desktop, web, mobile, or IoT. All data should be centrally located on a server if we want to design software that can be used on multiple platforms, such as instant messaging applications that can be used on mobile devices and desktops as shown in Figure 1. In that case, the most appropriate form of architecture to implement is service-oriented architecture [10].

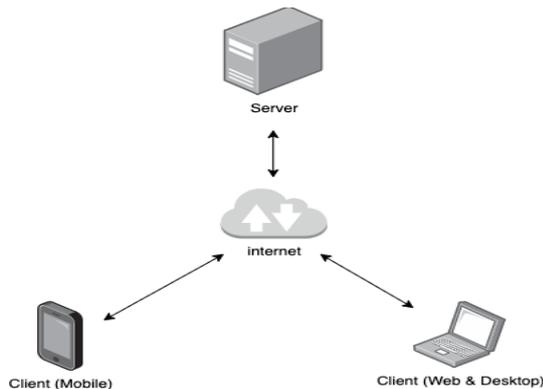


Fig. 1. Client-server architecture

Service-oriented architecture is the extension of client-server architecture where the server acts as a data center that provides access services in the form of services or better known as Application Programming Interface (API) [11]. The main advantage of implementing a service-oriented architecture is that it allows developers to design apps for many platforms using only one data center server, so it will greatly save time and cost in developing applications. One of the disadvantages of service-oriented architecture is that it has a more challenging security mechanism than other systems [12]. Anyone with access to the server can utilize it because it is based on public service. This necessitates additional security measures, such as the

use of encryption methods, tokens, and other measures, to protect the data on the server. However, if the services did not involve sensitive data or services, additional protection is not necessary.

B. Webhook

Webhook is a web-based application development paradigm, where the web application will be triggered when there is a desired event [13]. As in the example shown in Fig 2, a webhook is a mechanism through which a server (Telegram API) will call service on another server (Custom NodeJS server) when it receives a certain request from the client. In the example Figure 2, the Telegram API server is in charge of processing client chat sessions and delivering client requests to the NodeJS-based chatbot server; hence, each server has its responsibility to ensure that it is not overloaded.

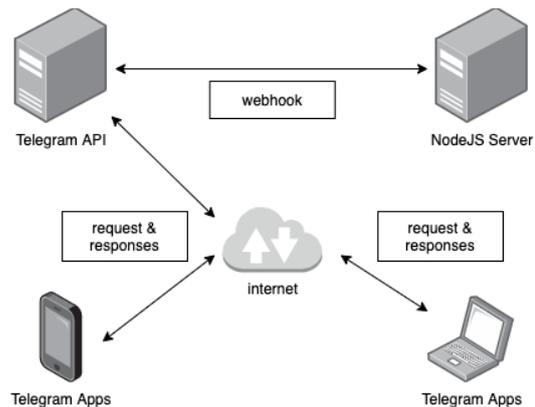


Fig. 2. Webhook application diagram

C. Telegram API

Telegram API is a service owned by Telegram that allows developers to develop applications using the Telegram platform for free. The Telegram API was often used by developers to create a variety of applications, including a bridge (middleware) between Telegram and other instant messaging systems [14]. With this middleware application, the users of the third-party application can communicate with the Telegram users. Another application that makes use of the Telegram API determines whether or not a driver is sleeping. The goal of this software is to prevent car accidents caused by drowsy driving [15]. The Telegram API offers huge potential for future development and the creation of applications that will make people's life easier.

III. DESIGN AND IMPLEMENTATION

This section discusses the chatbot application design process from the requirements gathering stage to the implementation stage. Data is the most important component to prepare while developing an application, in this case, the data is kept in a table-based relational database. For the topic of consultation, the conversation is carried out by two people in the form of questions and

answers. The most basic data requirements are fulfilled by having at least two tables: the user's table, which holds data on users who have sent messages, and the messages table, which stores the contents of user-chatbot dialogues. The user has been confirmed to only converse with the chatbot and no other users, only one id user in the messages table needed to keep. The basic table form of the database can be seen in Figure 3.

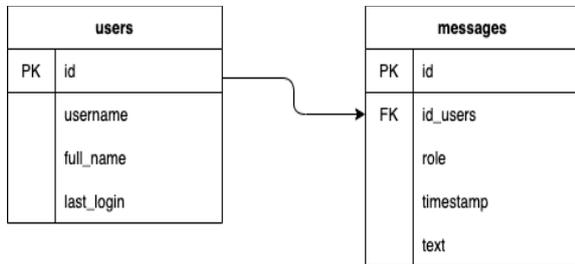


Fig. 3. Database diagram

The timestamp in the messages table plays a critical role in determining whether or not two or more messages belong to the same batch of messages. Checking the time interval between a message and the previous message is one method to determine a new message or is a continuation of the previous messages. If the time gap between the two messages is still less than the given time limit, the two messages are still part of the same series. Because there are various scenarios in which two identical input messages can generate different replies from the chatbot depending on whether the message is a new message or a continuation of the previous message.

The role field contains the data type enumeration client or chatbot, client means the message is a message sent by the client (input), while chatbot means the message is a reply from the chatbot (output). The webhook server is built using NodeJS with the help of the Telegraf package, the token key must be received first from the BotFather account before utilizing the Telegram API, then we can use that token to replace the token id parameter shown in Figure 4.

```
import pkg from 'telegraf';
const {Telegraf} = pkg;
const bot = new Telegraf('token_id');
```

Fig. 4. The token key

There are two basic forms of user input, namely commands and standard text. The command is a message that begins with a backslash sign '\', usually only consists of one word that contains a certain command, while the standard text is a free message like communicating with other users. This may use the function in the bot variable that was initialized previously to implement both types of messages

(command and standard text) using Telegraf that is shown in Figure 5.

```
//for command
bot.command('start', ctx => {
//code
})

//for standard text
bot.on('text', ctx => {
//code
})
```

Fig. 5. The telegraph for command and standard text

The chatbot application was deployed in a local server (localhost) with the help of the pm2 package. The pm2 package has the role to ensure the server is always online even if the terminal that runs the server was terminated. With this method above, we can build a reliable chatbot application with very minimal resources.

IV. RESULTS AND DISCUSSIONS

Chatbot applications can be built well using the Telegram platform, it can process all inputs well and smoothly without any problems. The example results of the chatbot application can be seen in Figure 6.

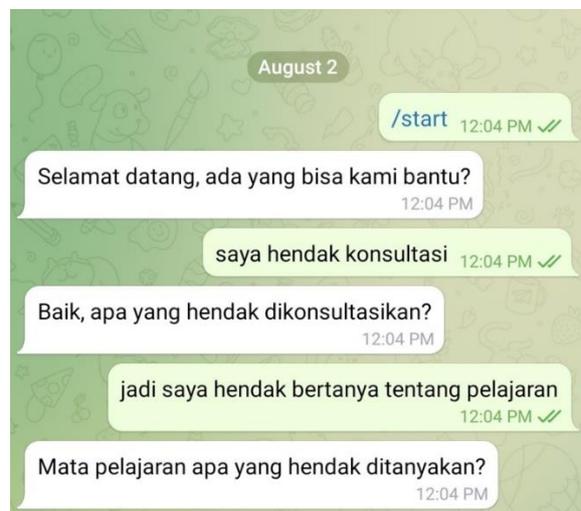


Fig. 6. Chatbot example screenshot

The time interval between the user's request time and the chatbot's response time is calculated during testing. It should be noted that because the chatbot is static and did not have any algorithms, the response will

be delivered immediately from the server, bypassing any processing, because no mathematical calculations or algorithms are executed on the server. Testing is carried out by delivering chatbots messages of varying lengths, ranging from one to ten words in Indonesian, with each word consisting of five to ten characters. Because the chatbot can only receive and respond to text messages, it will not be able to process input other than text.

TABLE I. TIME INTERVAL RESULT

num of word(s)	responses < 1 seconds
1	10
2	10
3	10
4	10
5	10
6	10
7	10
8	10
9	10
10	10

Table 1 shows the test was done with 100 attempts to send a request consisting of 10 types for 1-10 words, where each number was tested 10 times. The delivery time interval tolerance is 1 second, meaning that if the interval since the request is sent until the response is received is less than 1 second, then that time is still acceptable. Overall, the test results shown in Table 1 concluded that all experiments were within tolerance limits.

V. CONCLUSIONS

Based on the results of the tests, it can be seen that the response time required by a webhook-based chatbot is no different from the response time required by other methods. Although data on webhook-based chatbots needs additional steps, the time it takes from the request delivery stage to the chatbot receiving the response is still within tolerance limits, so we could say that it would have a similar performance with ordinary chatbot architecture.

ACKNOWLEDGMENT

We thank the Faculty of Information Technology - Universitas Kristen Duta Wacana for providing the sponsorship to do this research.

REFERENCES

- [1] X. Mao, W. Li, C. Lei, J. Jin, F. Duan and S. Chen, "A Brain-Robot Interaction System by Fusing Human and Machine Intelligence," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 27, no. 3, pp. 533-542, 2019.
- [2] I. García-Magariño, R. Muttukrishnan and J. Lloret, "Human-Centric AI for Trustworthy IoT Systems With Explainable Multilayer Perceptrons," *IEEE Access*, vol. 7, pp. 125562-125574, 2019.
- [3] X. I. Quan and J. Sanderson, "Understanding the Artificial Intelligence Business Ecosystem," *IEEE Engineering Management Review*, vol. 46, no. 4, pp. 22-25, 2018.
- [4] E. H. Wu, C. Lin, Y. Ou, C. Liu, W. Wang and C. Chao, "Advantages and Constraints of a Hybrid Model K-12 E-Learning Assistant Chatbot," *IEEE Access*, vol. 8, pp. 77788-77801, 2020.
- [5] B. Srivastava, F. Rossi, S. Usmani and M. Bernagozzi, "Personalized Chatbot Trustworthiness Ratings," *IEEE Transactions on Technology and Society*, vol. 1, no. 4, pp. 184-192, 2020.
- [6] K. A. Nugraha, "The effectiveness of peripheral interaction concept for mobile phone usage while driving," in 2018 International Conference on Information and Communications Technology (ICOIAC), Yogyakarta, Indonesia, 2018.
- [7] A. Balasubramanian, D. Damm, S. Melhem, A. C. Dausman, J. T. L. Levy, Y. Bu, B. Mao and C. S. Cauvel, "Transformers: Delivering Innovative Data Center High performance computing (HPC) and Artificial Intelligence (AI) Intel® Server Systems," in 2019 International Symposium on Systems Engineering (ISSE), Edinburgh, UK, 2019.
- [8] B. Li and S. Yu, "Keyword Mining for Private Protocols Tunneled Over WebSocket," *IEEE Communications Letters*, vol. 20, no. 7, pp. 1337-1340, 2016.
- [9] Restyandito and K. A. Nugraha, "The Effectiveness of Button Size on Mobile Device Based on Hand Dimension," in Proceedings of the International MultiConference of Engineers and Computer Scientists 2017 Vol II, IMECS 2017, Hong Kong, 2017.
- [10] X. Du, J. Xu, W. Cai, C. Zhu and Y. Chen, "OPRC: An Online Personalized Reputation Calculation Model in Service-Oriented Computing Environments," *IEEE Access*, vol. 7, pp. 87760-87768, 2019.
- [11] L. Li, W. Chou, W. Zhou and M. Luo, "Design Patterns and Extensibility of REST API for Networking Applications," *IEEE Transactions on Network and Service Management*, vol. 13, no. 1, pp. 154-167, 2016.
- [12] A. K. Rajaram, B. C. Babu and C. K. K. R., "API based security solutions for communication among web services," in 2013 Fifth International Conference on Advanced Computing (ICoAC), Chennai, India, 2013.
- [13] T. Istiana, R. Indra A, G. Budhi Dharmawan and B. Prakoso, "Pengembangan Sistem Diseminasi Prakiraan Cuaca Menggunakan Aplikasi Bot Telegram dengan Metode Webhook.," *Elektron : Jurnal Ilmiah*, vol. 12, no. 1, pp. 41-47, 2020.
- [14] C. Swandi, K. Nugraha, D. Sebastian and Restyandito, "Middleware Development to Connect Telegram Messenger and Instant Messenger for the Elderly.," in IOP Conference Series: Materials Science and Engineering, Yogyakarta, Indonesia, 2021.
- [15] C. Huda, F. A. Bachtiar and A. A. Supianto, "Reporting Sleepy Driver into Channel Telegram via Telegram Bot," in 2019 International Conference on Sustainable Information Engineering and Technology (SIET), Malang, Indonesia, 2019.