Jurnal Mantik, 6 (3) (2022), ISSN 2685-4236 (Online)

Published by:Institute of Computer Science (IOCS)

Jurnal Mantik

Journal homepage: www.iocscience.org/ejournal/index.php/mantik

Design of a Telegram Chatbot to Control Internet Connection on Computer Laboratory

Wildanil Ghozi[1], Elkaf Rahmawan P.[2], Oki Setiono[3]

[1,2]Faculty of Computer Scinece, Dian Nuswantoro University, Indonesia

[3]Faculty of Helath Science, Dian Nuswantoro University, Indonesia

A R T I C L E I N F O

Article history:

Received Oct 07, 2022
Revised Oct 14, 2022
Accepted Nov 04, 2022

Keywords:

Internet of things
Chatbot
Network management
Internet controller

ABSTRACT

Colleges with majors in computer science have a large number of computer labs to support practicum. Each laboratory room has several practicum classes per day. Moreover, each practicum requires different software and internet connection. Software can be prepared at pre-lecture. But the internet needs to be configured every class starts. A staff is stationed in each room to serve the needs of the practicum. However the staff does not have access rights to the router to configure the internet. To maintain network security, routers can only be managed by the network administrator. This requires staff to ask the network administrator to turn the internet on or off and to increase bandwidth limit. This interaction takes time so that the performance of the laboratory is poor. To solve this problem, we designed an internet controller bot as a bridging system between laboratory staff and routers. This chatbot provides limited router features needed by staff. So that the router's vital features cannot be accessed by staff. This system has been successfully developed so that staff can fulfill requests from lecturers to turn on/off the internet, check bandwidth consumption and increase bandwidth limit quickly.

Corresponding Author:

Wildanil Ghozi,
Faculty of Computer Scinece,
Dian Nuswantoro University,
Perumahan Nuansa Bangetayu D No 26 Semarang, Central Java, 50115, Indonesia.
Email: wildanil.ghozi@dsn.dinus.ac.id

1. INTRODUCTION

Colleges that have majors in information technology and computers, have a lot of computer laboratory needs to support learning. Furthermore, the need for application software and internet connection for each practicum is different. In conditions when the internet is used massively and simultaneously, bandwidth management is needed to ensure that internet speed is shared fairly among each computer(Sujalwo et al., 2011).

The need for the internet is dynamic according to the lesson plan, causing the network to be configured to turn on, turn off, and allocate bandwidth. The right bandwidth allocation can be one of the guarantees of quality of service (Anggrawan et al., 2018). A laboratory staff is assigned to each computer laboratory room to prepare and fulfill practical needs including the internet. However, granting router access to each staff is a violation of security policy because it is not in accordance with security of

management(Clemm, 2007). This creates a gap between the obligations and the authority given to the laboratory staff. To overcome this gap, a system is needed that can be an intermediary between routers and laboratory staff. This system is built to limit access to vital features on the router that are forbidden to laboratory staff. Laboratory staff can turn the internet on or off, monitor the bandwidth usage and set the bandwidth limits easy and fast but still safe.

To realize an easy and fast system, a platform that is close to social interaction and communication is needed, such as short message services or chatting. The high number of messenger application users is also proof that many people are used to this application for daily use. Telegram as a chat application has a better security system than other applications. Moreover, Telegram has a free and open source API that can be used as an intermediary to create a chat as a user interface of our system so that it allows users to communicate with the system through chatbots(Nufusula et al., 2018; Widya et al., 2020).

Various studies on the use of Telegram chatbots as user interfaces for information systems have been carried out, one of them is building an internet of things for smart workspaces that can control electronic devices through the Telegram chatbot. This study makes commands to turn on or off lights and fans, lock or unlock doors, and turn the on or off power plugs(Muslih et al., 2018).

In the office scope, the telegram chatbot is applied to the paperless office (PLO) system, which is a web-based application that has been developed by Universitas Gadjah Mada (UGM) for various units and purposes. Telegram chatbot can be used as an access interface to the PLO system. User authentication security is guaranteed because each user has a unique chat ID(Prastowo et al., 2019).

The use of the Telegram chatbot for computer network management has also been carried out. This study focuses on the management of hotspot users to provides features to add users, activate or deactivate users, receive hotspot reports and so on(Suqma et al., 2021). The system has a weakness, there is no user authentication so that everyone who knows this bot can use it. In our study, we have developed an internet controller bot for laboratories that can be used by laboratory staff to manage internet connection. Each staff must register themselves first to be authenticated. A network administrator will be asked for the approval of every staff who registers on the system.

2. RESEARCH METHOD

2.1. System Development Method

The system development method used in this study is the prototyping method. In the prototyping method, there are several activities: 1) Initial investigation, 2) Requirements Definition, 3) System Design, 4) Coding and Testing, 5) Implementation and Maintenance. The life cycle of these activities are illustrated in Figure 1 below:

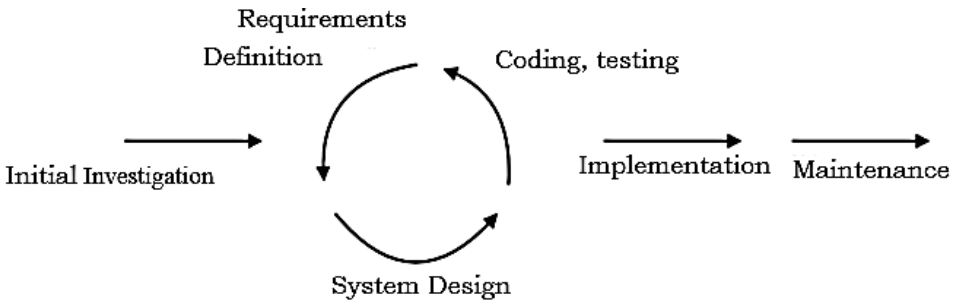

Figure 1. Prototyping Framework(Fathi et al., 2019)

2.2. Telegram Bot API

Telegram is a free messaging application that can run on a variety of different platforms such as mobile, web and desktop computer. Telegram works with cloud-based techniques so that each message is stored on the server. This allows us to use Telegram on several devices at the same time. Telegram is widely used as a chatbot to serve certain needs because Telegram has the advantage of providing an openly accessible Telegram-API for the public(Widya et al., 2020).

Telegram provides a special user as a bot, acting in response to messages it receives with the aim of replacing the human role in carrying out certain activities. The bot is developed by a third party by utilizing the provided bot API to read incoming messages, and send responses(Muhammad et al., 2020; Parlika et al., 2021).

Long polling is a method of communication between the Telegram bot API and third-party software. Long polling is a technique where HTTP requests are made by the client to wait for events from the server. The connection is kept open so that the client gets updates immediately. When the bot receives a message, the system immediately knows the contents of the message and can execute commands according to the algorithm that has been created on the system(Nufusula et al., 2018; Soeroso et al., 2017; Widya et al., 2020).

Telegram bot API allows developers to create a chatbot according to their needs. So that incoming messages can be used as input, process according to the algorithm, and send a reply message as output. The long polling technique requires a local computer to run the system and manage the exchange of data between applications through the API(Rofiq et al., 2017). The long polling method is illustrated in Figure 2.

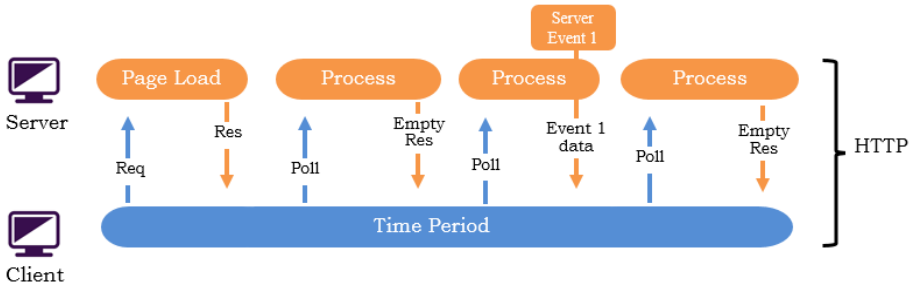

Figure 2. Long Polling Method Illustration(Nufusula et al., 2018)

2.3. Analysis of The Proposed System

The bot that will be developed is a system for controlling the internet. Basically the internet connection can be controlled through the winbox application which directly remotes the router through the local network. In addition, the router can also be controlled remotely using the SSH protocol. This feature will be used by bots to communicate with routers. So the internet controller bot communication flow is described in figure 3.

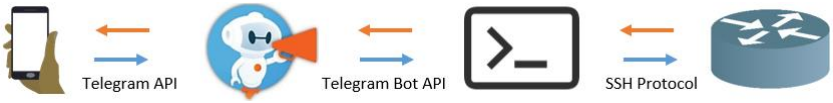

Figure 3. Internet Controller Bot Communication design

To ensure system security, only staff may use the system. user authentication is applied to every incoming message. The bot will check the username if it is registered as a staff to be allowed to run the command. A list of staff usernames can be created by the

network administrator. Anyone can sign up for staff via bot, then the network administrator will decide whether to accept or reject. All features and program flow are described figure 4.

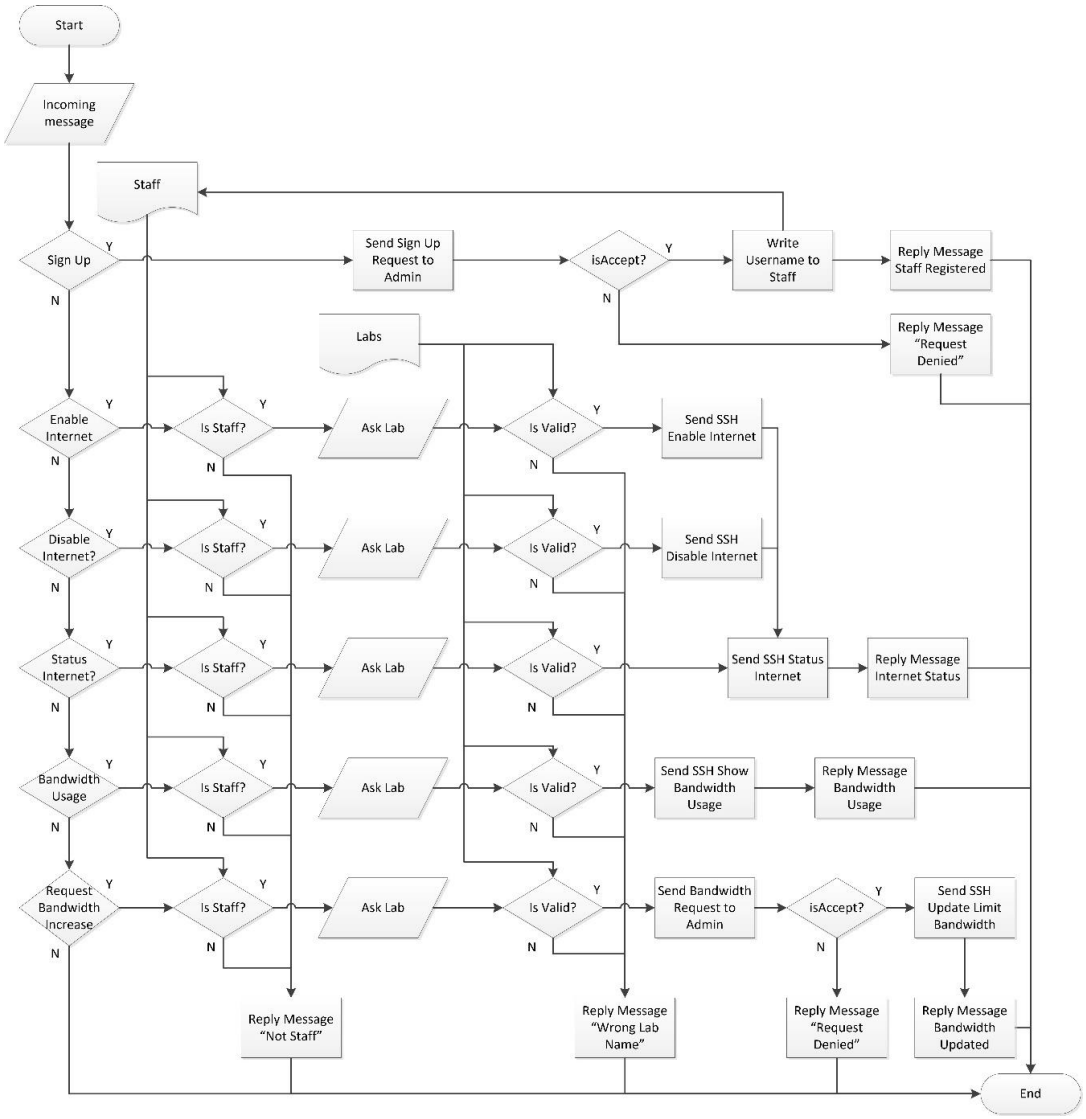

Figure 4. Flow Chart Diagram

The command used to execute each feature on the internet controller bot is listed below:

a. /signup to Sign up for staff
b. /enable to Turn on the internet connection.
c. /disable to Turn off the internet connection.
d. /pnet to Check the status of the internet connection.
e. /bwlab to Check internet bandwidth consumption.
f. /bwup to Request an increase in internet bandwidth to the admin.

The command is case insensitive and listed on the menu button on the chatbot interface so that the use of bots becomes easier. If the user sends an incorrect command then the bot will ignore the message.

2.4. Usability Testing

Usability testing is needed to find out the usefulness of the system that has been developed for users. Then the user is involved to provide a response to the system, then it will produce recommendations for improvement(Setiawati et al., 2018).

This study uses a system usability scale (SUS) as a tool to measure the usefulness of internet controller bot. The System Usability Scale (SUS) was created The System Usability Scale (SUS) was created by John Brooke in 1986 by John Brooke. SUS can be used as a quick measure to find out how people perceive the usability of a computer system. SUS has proven to be a very simple and reliable tool to use in conducting usability evaluations, it can also be used to compare systems(Bangor et al., 2009).

SUS consists of 10 questions with answers strongly disagree to strongly agree. The answer is measured by a score on a scale of 1 to 5. For questions numbered 1,3,5,7, and 9 the contribution score is the value of the selected scale minus 1. For questions number 2,4,6,8, and 10, the contribution score is 5 minus the value of the selected scale. Then multiply the contribution score by 2.5 to get the overall value of the system usability(Aprilia et al., 2015). SUS scores range from 0 to 100 with acceptability ratings described in figure 5.

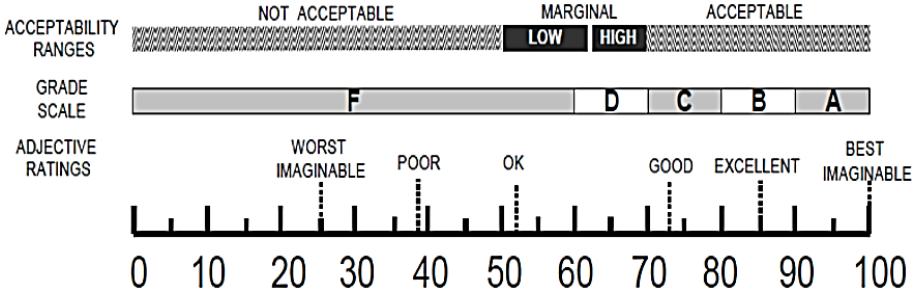

Figure 5. Comparison of Adjective Ratings, Acceptability Scores, and School Grading Scales, Relating to the SUS Average Score(Bangor et al., 2009)

3. RESULTS AND DISCUSSIONS

3.1. Internet Controller Bot

This research has resulted in an innovation that changes the communication flow of internet connection services in the computer laboratory. Previously, a staff member had to contact the network administrator to request internet connection settings. Now staff can manage their own internet via smartphone or computer. This system ensures network security by providing limited access to staff. This system also restricts users to only those permitted by the network administrator. To use the chatbot, the user must install the Telegram Messenger application. To start using it, the user must scan the QR code that has been generated to redirect to a chat with the bot and then tap start.

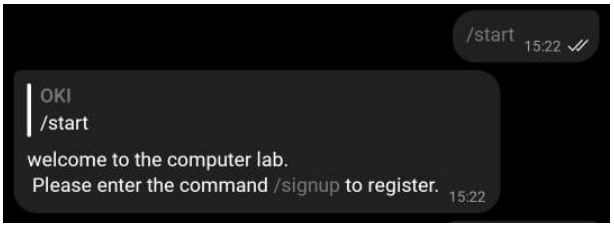

Figure 6. Start Chat with Chatbot

To be able to use the bot, staff must register by pressing menu /signup. The bot will send a registration request to the admin, including the user's username. If the admin recognizes that the user is a computer laboratory staff, then the admin will accept it.

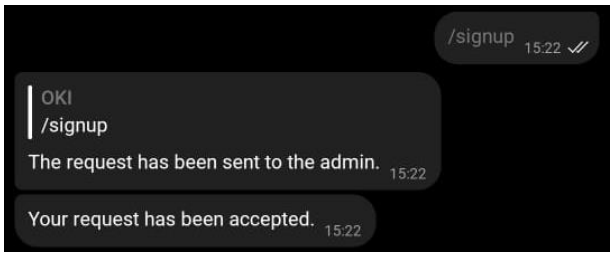

Figure 7. Signup Request by Staff

Figure 8. Signup Request Accepted by Admin

After the username is registered as staff, the user can execute the available commands. For example, the command to turn on the internet is as shown in Figure 9. The internet status on the internet controller bot corresponds to the internet status seen directly on the router as shown in Figure 10.

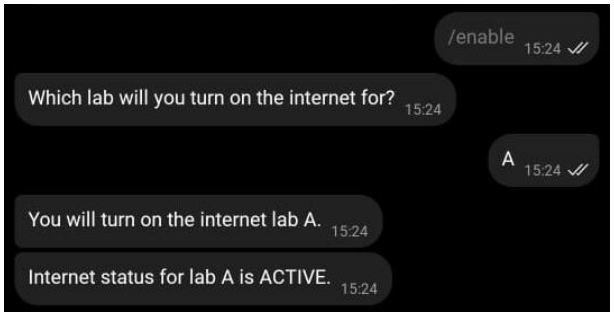

Figure 9. Feature to Enable Internet

```
[botlab@Router-Lab] > ip firewall nat print where comment=A
Flags: X - disabled, I - invalid, D - dynamic
 0    ;;; A
      chain=srcnat action=masquerade src-address=192.168.8.0/26 dst-address-list=!LOCAL-IP
[botlab@Router-Lab] >
```

Figure 10. Internet Status on Lab A Viewed Through The Router

Figure 9 shows the staff executing the command to turn on the internet for lab A, while figure 10 proves that the network controller bot succeeded in sending commands to the router to fulfill the computer laboratory's request.

3.2. System Testing

The internet controller bot has been used by staff within a week. Then the staff was asked to fill out a questionnaire based on their experience using this bot. The following is a questionnaire that researchers use to measure the usability of the system being tested, there are 10 SUS statements described in figure 11.

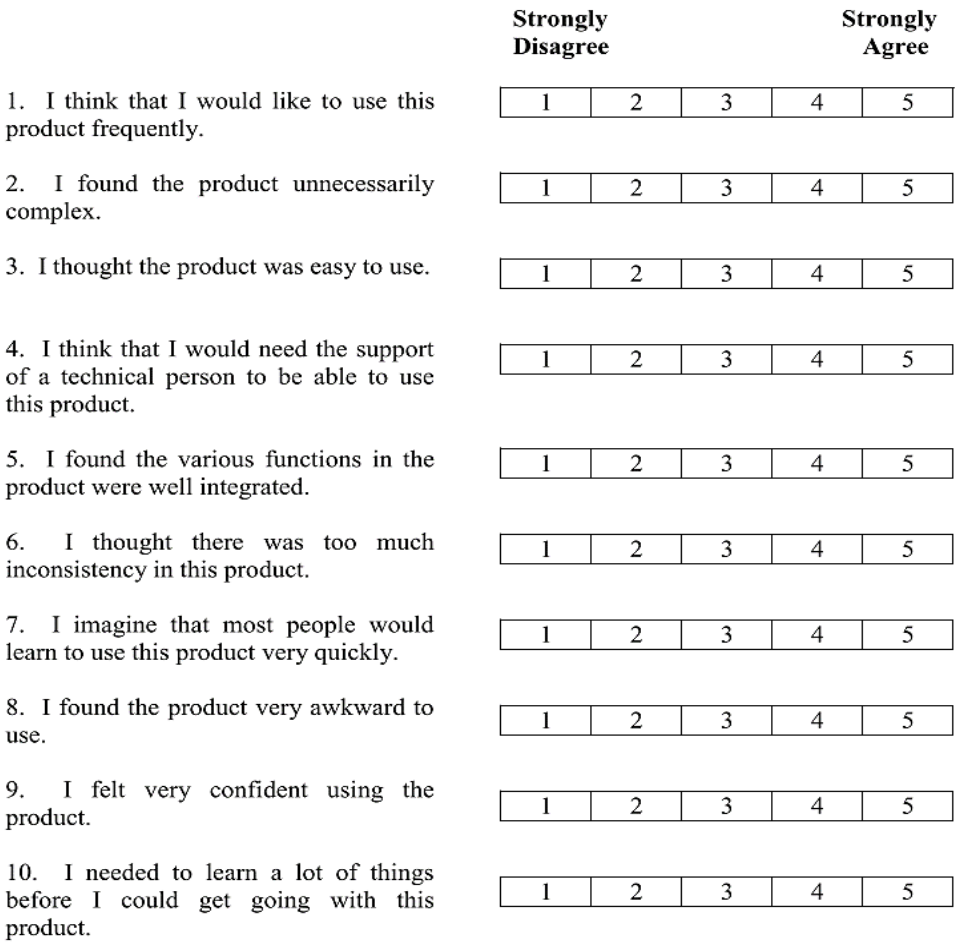

Figure 11. 10 System Usability Scale Statement

16 staff members have filled out the questionnaire anonymously. The results of the questionnaires are summarized in table 1 below.

Table 1. SUS Questionnaire Results

Respondent	SUS Statement										Total	SUS Value
	1	2	3	4	5	6	7	8	9	10		
1	2	2	4	3	4	2	5	1	5	2	30	75
2	4	3	4	2	4	2	4	2	3	4	26	65
3	5	4	4	5	5	2	5	2	5	2	29	72,5
4	5	5	5	4	5	2	3	2	3	4	24	60
5	5	2	5	2	5	2	5	2	5	2	35	87,5
6	5	1	5	1	5	1	5	1	5	1	40	100
7	4	3	4	1	5	2	5	1	4	4	31	77,5
8	4	1	5	2	5	1	5	1	5	1	38	95
9	4	2	5	4	5	1	5	1	5	1	35	87,5
10	5	1	5	3	5	1	4	1	4	5	32	80

11	5	1	5	1	5	1	5	1	5	1	40	100
12	3	1	4	2	4	2	4	2	4	2	30	75
13	5	1	4	1	5	1	5	1	4	1	38	95
14	5	3	5	1	5	1	5	1	5	1	38	95
15	4	2	5	3	4	2	4	2	4	3	29	72,5
16	4	3	5	2	3	1	5	1	5	3	32	80
SUS Final Score												82,34375

The final SUS score from 16 respondents' responses was 82,34375 rounded up to 82. This value is included in the acceptable range according to the comparison of acceptability ranges as described in figure 5. This means that this internet controller bot is suitable for use in computer laboratories.

4. CONCLUSION

In this study, we have analyzed the problem of internet needs in a computer laboratory. We found a gap between the staff's duty to provide internet services and the router access rights that the staff had. To solve this problem, we propose to develop a bot to control the internet connection that can be used directly by the staff safely. The development method used in this research is prototyping. This research succeeded in producing a prototype of the internet controller bot. Beta testing of this bot was conducted in a computer lab for one week. To measure the usefulness of this bot, a system usability scale (SUS) test was conducted on this bot with 16 laboratory staff as respondents. SUS testing has obtained a score of 82, which means the bot is acceptable. For the development of future research, the authors suggest adding new features required by the laboratory. New features can be related to computer networks, the internet, or other facilities in the laboratory to support smart laboratories.

ACKNOWLEDGEMENTS

The author is grateful to the Dian Nuswantoro University computer laboratory which has provided the facilities and infrastructure for this research.

REFERENCES

Anggrawan, A., Santoso, H., Tamando Sihotang, H., Pyanto, D., & Rahmad Hidayat, F. (2018). PENGATURAN BANDWIDTH MANAGEMENT DAN TIME LIMITATION BERBASIS USER MANAJER MIKROTIK. *Jurnal Mantik Penusa*, *2*(2), 22–28.

Aprilia, I. H. N., Santosa, P. I., & Ferdiana, R. (2015). Pengujian Usability Website Menggunakan System Usability Scale Website Usability Testing using System Usability Scale. *Jurnal IPTEK-KOM*, *17*(1), 31–38. Retrieved from https://jurnal.kominfo.go.id/index.php/iptekkom/article/view/428

Bangor, A., Kortum, P., & Miller, J. (2009). Determining what individual SUS scores mean; adding an adjective rating. *Journal of Usability Studies*, *4*(3), 114–123.

Clemm, A. (2007). Network Management Fundamentals. In Journal of Family Practice (Vol. 6, Issue 4). Cisco Press.

Fathi, A., & Sawehli, A. (2019). *Improving Software Security Testing of Software Development Life Cycle (SDLC) For Web-Based Applications By Providing A Quality System (Web-Vs) for Vulnerability Assessment - Maste ...* (Issue December) [ASIA PACIFIC UNIVERSITY OF TECHNOLOGY & INNOVATION (APU)]. doi: 10.13140/RG.2.2.17114.29128

Muhammad, R., Arwani, I., & Rahayudi, B. (2020). Implementasi Plugin Notifikasi Sebagai Media Integrasi Antara E-Learning Moodle dengan BOT Telegram (Studi Kasus : Bimbingan Belajar The Second School). *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *4*(7), 2275–2282.

Muslih, M., Supardi, D., Multipi, E., Nyaman, Y. M., & Rismawan, A. (2018). Developing Smart Workspace Based IoT with Artificial Intelligence Using Telegram Chatbot. *2018 International Conference on Computing, Engineering, and Design (ICCED)*, 230–234. doi:

10.1109/ICCED.2018.00052
Nufusula, R., & Susanto, A. (2018). Rancang Bangun Chat Bot Pada Server Pulsa Mengunakan Telegram Bot API. *JOINS (Journal of Information System)*, *3*(1), 80–88. doi: 10.33633/joins.v3i1.1884
Parlika, R., Khariono, H., Kusuma, H. A., & Setyawan, A. (2021). PEMANFAATAN BOT TELEGRAM SEBAGAI E-LEARNING UJIAN BERBASIS FILE. *JIP (Jurnal Informatika Polinema)*, *7*(4), 65–72. Retrieved from http://jip.polinema.ac.id/ojs3/index.php/jip/article/view/696
Prastowo, B. N., Putro, N. A. S., & Dhewa, O. A. (2019). PLO User Interface based on Telegram Bot. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, *13*(1), 21. doi: 10.22146/ijccs.29089
Rofiq, F. N., & Susanto, A. (2017). Implementasi RESTful Web Service untuk Sistem Penghitungan Suara Secara Cepat pada Pilkada Implementation of Restful Web Servicefor Quick Count System of MayorElection. *Eksplora Informatika*, *6*(2), 159–168.
Setiawati, A., Rahim, A., & Kisbianty, D. (2018). Pengembangan dan Pengujian Aspek Usability pada Sistem Informasi Perpustakaan (Studi Kasus : STIKOM Dinamika Bangsa Jambi). *Processor*, *13*(1), 1173–1188.
Soeroso, H., Zuhri Arfianto, A., Eka Mayangsari, N., & Taali, M. (2017). Penggunaan Bot Telegram Sebagai Announcement System pada Intansi Pendidikan. *Seminar Master PPNS*, *2*(1), 45–48.
Sujalwo, Bana Handaga, & Heru Supriyono. (2011). Manajemen Jaringan Komputer dengan Menggunakan Mikrotik Router. *Jurnal Komunikasi Dan Teknologi Informasi*, *2*(2), 32–43.
Suqma, A. A., & Prihanto, A. (2021). Manajemen User Mikrotik Berbasis Telegram Bot. *Journal of Informatics and Computer Science*, *03*(02), 131–142.
Widya, M. A. A., & Irlangga, P. (2020). Pengembangan Telegram Bot Engine Menggunakan Metode Webhook Dalam Rangka Peningkatan Waktu Layanan E-Government. *Saintekbu*, *12*(2), 13–22. doi: 10.32764/saintekbu.v12i2.884