# Ontology-based Chatbot to Support Monitoring of Server Performance and Security By Rule-base

**Fauzan Ishlakhuddin*[1], Azhari SN[2]**
[1]Master Program in Computer Science, FMIPA UGM, Yogyakarta, Indonesia
[2] Department of Computer Science and Electronics, FMIPA UGM, Yogyakarta, Indonesia
e-mail: ***[1]fauzan.i@mail.ugm.ac.id**, [2]arisn@ugm.ac.id

***Abstrak***

*Server merupakan perangkat lunak atau perangkat keras komputer yang menyediakan fungsi untuk perangkat lain (klien). Umumnya komputer server memiliki sumber daya yang besar sehingga dapat digunakan oleh satu atau lebih klien melalui jaringan dengan izin dan persyaratan tertentu. Oleh sebab itu, dibutuhkan sebuah sistem monitoring yang dapat memantau aktivitas server dan memberitahukan jika terjadi masalah. Penelitian ini berfokus pada pengembangan sistem pemberitahuan dan tanya-jawab untuk menghubungkan antara admin jaringan dengan sistem monitoring melalui chatbot. Chatbot yang dikembangkan mampu mengirim pemberitahuan kepada admin jaringan jika terjadi kesalahan dan mampu menjawab pertanyaan perihal kondisi server. Sistem tanya-jawab yang dikembangkan menerapkan pengolahan bahasa alami (natural language processing) untuk bahasa Indonesia. Proses pemahaman pertanyaan dilakukan dengan mengklasifikasi setiap kata (token) berdasarkan pengetahuan bahasa pada ontologi, kemudian hasil klasifikasi diproses dengan basis aturan sehingga menghasilkan kesimpulan untuk mengambil data monitoring dan disusun menjadi jawaban. Hasil pengujian menunjukkan bahwa sistem yang dikembangkan mampu memberitahukan secara otomatis jika terjadi masalah pada server dan mampu menjawab pertanyaan dengan akurasi sebesar 95%.*

***Kata kunci****—Sistem Tanya-jawab, Rule-base, Pemantauan Server, NLP, Ontologi.*

***Abstract***

*The server is a computer program or a device that provides functionality for other programs or devices, called "clients". Generally, server computers have many resources that can be used by one or more clients through the network with specific permissions and requirements. Therefore, the server needs a monitoring system that can monitor server activity and notify if problems occur. This research focuses on developing a notification and question and answer system to connect the network admin with the monitoring system via chatbot. The developed chatbot can send notifications to the admin if an error occurs and can answer questions about the server's condition. The question and answer system developed implements natural language processing for Indonesian. The process of understanding questions is by classifying each word (token) based on language knowledge stored in the ontology. Then the classification results are processed by rule-base to produce conclusions to take monitoring data and compiled into answers. The test results show that the developed system can auto-notify if any problem in a server, and can answer questions by accuracy 95%.*

***Keywords—****Question Answering, Rule-base, Server Monitoring, NLP, Ontology.*

*Received August 13th,2020; Revised April 29th, 2021; Accepted April 29th, 2021*

## 1. INTRODUCTION

The rapid development of technology currently has a significant impact on changes in the pattern of human life. Generally, the development and research carried out in the field of technology aims to facilitate people in carrying out their work. The server is one of the technological developments in the field of computer network systems that provide services (services) in computer networks. Generally, computer servers have extensive resources and can be used together with specific permissions by clients. There are three types of services available on servers, namely: application servers, data servers, and proxy servers, but generally, a server has several kinds of services at once [1]. Usually, servers are connected to a network so that clients can use existing resources. Therefore a system is needed that can monitor activity on the server and can report monitoring results at any time to the admin. There are many sound monitoring systems today, one of which is the Nagios XI.

Nagios XI is a product from Nagios that has a paid license with various advantages and conveniences offered. Nagios XI is a server monitoring system that can monitor critical servers, connected network equipment, websites, services, and applications [2]. For monitoring notification settings on the Nagios XI, it has provided, but it is still limited via email and SMS, making it difficult for the admin to get information. Therefore it is necessary to develop an active and more accessible notification section, one of which is by developing notifications through chatbots that can send notifications as well as being a question and answer system for answering questions about the results of monitoring.

Chatbots are a conversation agent that allows users to access information and services through natural language dialogues, either text or voice [3]. In practice, chatbots can be built independently or use a chat application that provides services to develop chatbots. Chatbot in having a conversation (understanding and answering chats) requires a specific method, which is commonly referred to as applying natural language processing methods.

Natural Language Processing (NLP) is a computer science discipline that aims to understand the concepts and intentions of human language. At the same time, humans are quite adept at understanding linguistic syntax and grammar as well as implied (semantic) meanings, while computers have difficulties in processing natural language queries [4]. NLP also handles methods for analyzing, modeling, and understanding human language and the semantics of its sentences [5]. The method that can be used to implement semantics is to use ontology as a medium for representing language knowledge, and rule-base as a method for reasoning question sentences as well as determining answers.

Ontology is used to represent models from the real world with a series of concepts to explain certain domains of knowledge [6]. In the ontology, there is a semantic structure that conveys hidden (implicit) meanings related to the conceptualization of the domains carried by the structure: classes, relationships, functions, and other objects [7].

Rule-base is a form of approach that represents knowledge using a series of rules written in the IF-THEN form [8]. Meanwhile, according to Mendel, rule-based is one of the bases of an expert system that uses human intelligence to solve real-world problems that are represented by rules [9]. Thus, the complexity of the order using the rule-based method will depend on how many rules have been made, and how specific the rules are in making decisions.

## 2. METHODS

### *2.1. Process Logic Server*

The current process logic on the Badan Pemeriksa Keuangan (BPK) server is illustrated in Figure 1.

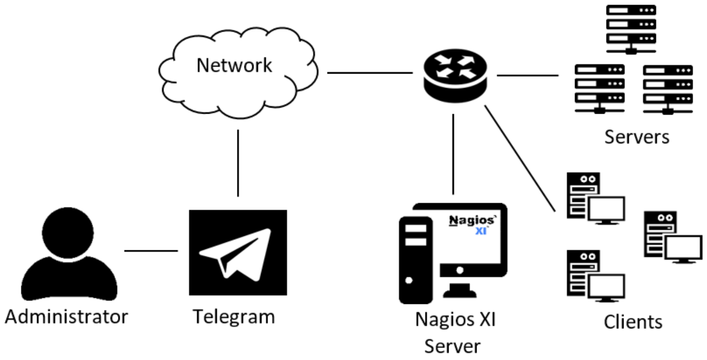


Figure 1 Process logic BPK server

Figure 1 illustrates the process that already exists on the current BPK server, where all server resources and activity have been monitored using the Nagios XI application and have implemented notifications via telegram bots. But, the information is still passive (one way) only from Nagios and have not been able to answer a question. Therefore, this research conducted to develop a notification bot that can be able to answer a question by natural language processing about the server state.

### 2.2. *Chatbot Process Flow Design*

The design of the chatbot process flow as a guide in conducting this research. In general, the chatbot process flow shown in Figure 2

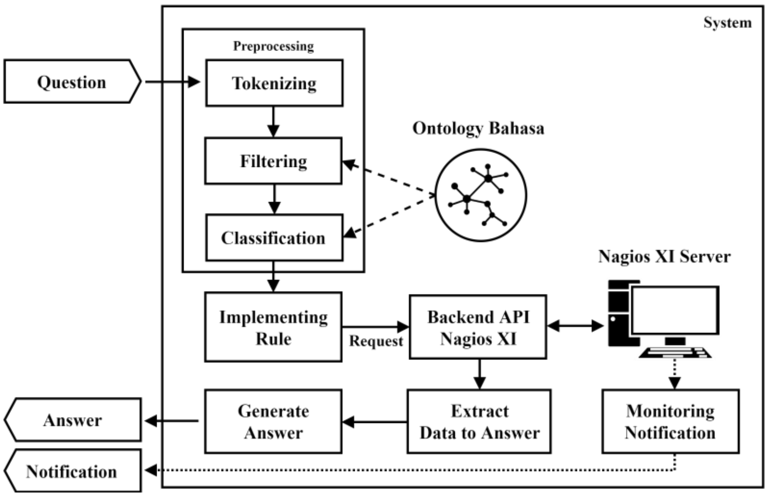


Figure 2 The general overview of chatbot process flow

Figure 2 illustrates the process flow of the chatbot system from questions to answers and automatic notifications. To explain each process from Figure 2 is as described below:

1. Questions is a question sentence in Indonesian.
2. Preprocessing is the processing of question sentence before the main process is carried out (rule-based checking). This process consists of: (a) tokenization, separating sentence by spaces character, then the result called tokens; (b) filtering, remove the stopword in the token; (c) classification, the grouping of tokens in their class according to language ontology.
3. Ontology Bahasa is a representation of Indonesian language knowledge in the server scope.
4. Rule Checking, which is the process of determining the required data to request from the Nagios backend API by processing the tokens by defined rules.
5. The request is the process of requesting data in backend Nagios API's and the return is an XML data format.

6. Backend Nagios XI API is the Nagios XI backend process that shows the developer requests data from monitoring results.
7. Answer extraction is the process of processing query result data into a Python dictionary format so that it is easy to compile answers.
8. Preparation of answers, i.e., processes the results of data extraction so that it is appropriate to answer the questions asked.
9. Answers are sent a correct answer from the question.
10. Warning notice, settings are planted directly in Nagios XI, which will notify you if an error occurs.

### *2.3. Ontology design*

The ontology in this study is to represented language knowledge, and the design stages are carried out following the theory of Noy & McGuinness [10]. The following are the stages in designing a language ontology:

### *2.3.1 Determine concept and domains*

The concept and domain of this study is an ontology to represent language knowledge related to server performance and security in Indonesian.

### *2.3.2 Determine classes*

After defining the domain of the ontology, then determine what classes are needed. For details on the following classes with examples and explanations, see Table 1.

Table 1 Classes and Description in Ontology Bahasa

| No. | Class | Example | Description |
|---|---|---|---|
| 1 | *Keyword* | *Host, service, status, list, information* | To represent keyword information. |
| 2 | *State* | *Up, down, critical, ok, unknown* | To represent information of status words. |
| 3 | *Synonym* | Kondisi, situasi, keadaan | To represent information of synonymous words. |
| 4 | *Stopword* | Itu, pada, yang, di, ?, !, : | To represent information of stopwords. |

### *2.3.3 Determine classes relation*

After determining the class design, the next step is to define the relationships between classes, an overview of the relationships between classes, as in Figure 3.

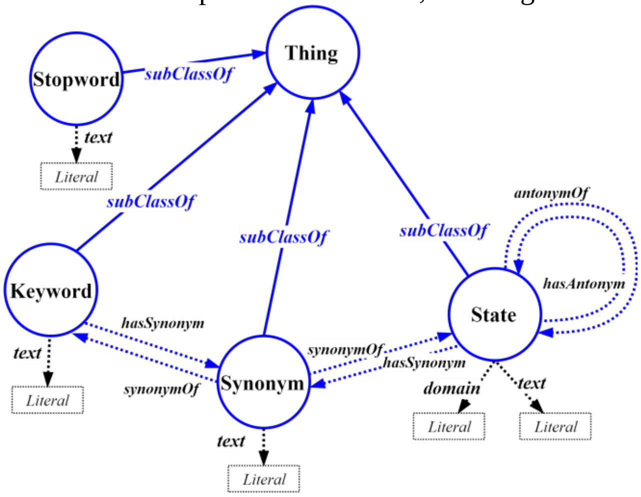


Figure 3 Class Relations Overview

Figure 3 describes the relationships between classes in Table 1 with details of the relationships that exist between classes, where there are six relations or commonly referred to as object properties (OBP), namely: (a) hasSynonym which defines the relationship between Keywords and State with Synonym class; (b) synonymOf is the opposite of hasSynonym; (c) hasAntonym defines the relationship between state and state; (d) antonymOf is the opposite of hasAntonym.

### *2.3.4 Determine data type properties*

After defining the relations between classes or OBP, the next step is to identify Data Type Properties (DTP). In Ontology Bahasa, there are only 2 DTP, namely: Text (to define words of each instance) and domain (to define the area of each instance in class status). A list of DTP in Table 2.

Table 2 Description Data Type Properties in Ontology Bahasa

| No. | Name | Domain | Type |
|---|---|---|---|
| 1 | Text | Keyword, State, Synonym, Stopword | Literal |
| 2 | Domain | State | Literal |

### *2.4. Preprocessing Process*

Preprocessing is designed to filter question messages before rule-based processing takes place. Where in this study, there are three preprocessing stages, namely: tokenizing, filtering, and classification. The three processes are in the flow chart in Figure 4.

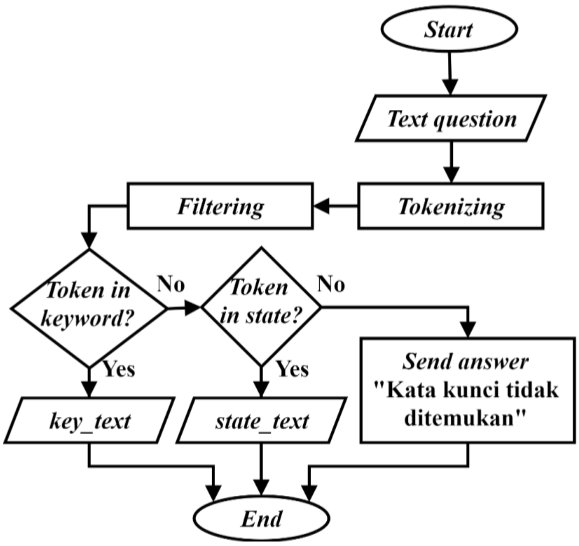


Figure 4 Flow Chart Preprocessing

Figure 4 is a flow chart diagram of the preprocessing process. The flow process starts from the system gets the input of Text questions, next step tokenizing process to break the question text into tokens. The next step is the filtering process to remove the stopword in the tokens, then the collection of tokens is classified according to its group (class) in Ontology Bahasa. The classification results stored in a variable according to the name of the class. If each token in the question sentence does not belong to any class, then a message "Keyword tidak ditemukan" will be sent for the answer.

### *2.5. Rule design*

The draft of this rule is to determine data that will be requested, as well as for deciding what answers will send. The rules are checking process flow, as seen in the flow chart in Figure 5. For the examples of some of the states that have designed in Table 3.

Table 3 List of rules

| No | Name | Rules |
|---|---|---|
| 1 | Check keyword 1 | IF (x is synonym) AND (x is synonym of keyword)<br>IF (keyword value is host)<br>Host is true<br>ELSEIF (keyword value is service)<br>Service is true<br>ELSE<br>Server is true<br>ENDIF<br>ENDIF |
| 2 | Check keyword 2 | IF (x is synonym) AND (x is synonym of keyword)<br>IF (keyword value is count)<br>Count is true<br>ELSEIF (keyword value is list)<br>List is true<br>ELSE<br>Status is true<br>ENDIF<br>ELSE<br>Replay error "Kata kunci tidak ditemukan"<br>ENDIF |
| 3 | Search count | IF (count is true) AND (state is true)<br>IF (host is true) AND (service is true) AND (name is true)<br>Search status of service by state on host by name<br>ELSEIF (host is true) AND (service is true) AND (name is false)<br>Replay error "Nama tidak ditemukan"<br>ELSEIF (host is true) AND (name is true)<br>Search count of host by name and state<br>ELSEIF (host is true) AND (name is false)<br>Search count of host by state<br>ELSEIF (service is true) AND (name is true)<br>Search count of service by name and state<br>ELSEIF (service is true) AND (name is false)<br>Search count of service by state<br>ELSE<br>Search count of host and service by state<br>ENDIF<br>ELSEIF (count is true) AND (state is false)<br>IF (host is true) AND (service is true) AND (name is true)<br>Search count of ervice on host by name<br>ELSEIF (host is true) AND (service is true) AND (name is false)<br>…<br>ENDIF<br>ENDIF |

### *2.6. Answer Extraction*

Answer extraction is the process of retrieving information from the request to the Nagios XI API backend, where the data obtained from the request is in the form of XML data. At this design stage, the XML data that has received is converted to JSON to make it easier to extract, and then the data is taken as needed to answer the questions asked. The following flow chart picture is in Figure 5, while for an example can be seen in Table 4 no. 4, 5, and 6.

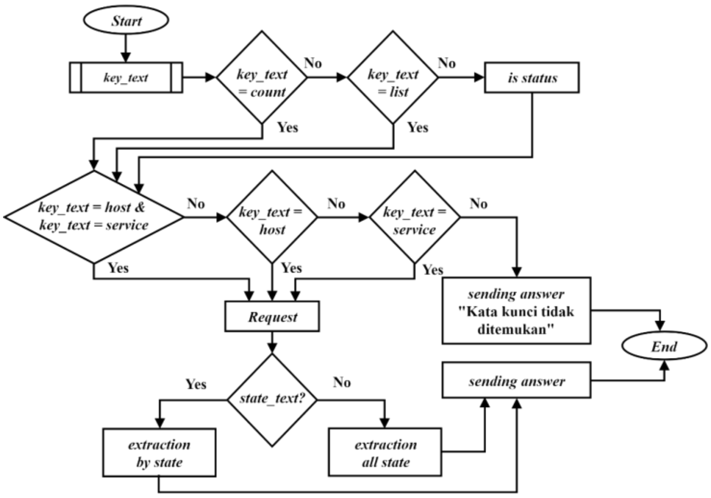


Figure 5 Flow chart of rules, extraction of answers and preparation of answers

Figure 5 illustrates a flow diagram that consists of a draft rule process that starts from the key_text classification process that is separate from this flow diagram, as in Figure 4. The results of the key text checked whether there are words that go into the related rules if there is proceed. Next, if there isn't any, the first stage of the checking process is considered as status. Then check again whether there is the word host and service, host, or service if there is then proceed to request information related to the results of the check if there is not then immediately sent a message if not found the keyword. The request results are processed by extracting the data and checking again whether there is a status text, status text obtained from the classification results. It intended to retrieve only the data needed for the answers taken. After the extraction process, it continued to compile the answers with data from the extraction results and sent it to the sender of the question as an answer.

### *2.7. Design of drafting an answer*

The design of the preparation of answers, in this stage, is to prepare answer from requested data are more easily understood by humans for the answer. Examples of the preparation of the answer can be seen in Table 4 no. 7.

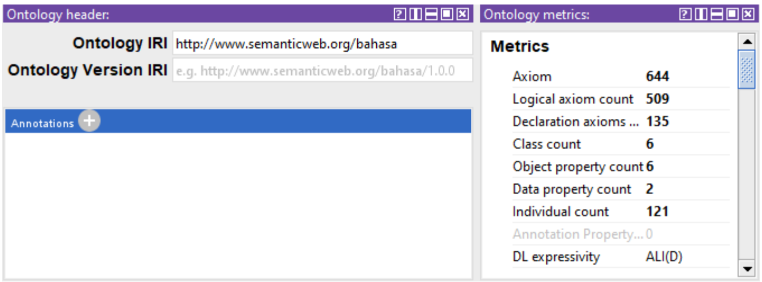
Table 4 Example process from question to answer

| No | Process | Input | Result |
|---|---|---|---|
| 1 | Tokenizing | Apakah ada service yang sedang bermasalah? | [“apakah”, “ada”, “service”, “yang”, “sedang”, “bermasalah”, “?”] |
| 2 | Filtering | [“apakah”, “ada”, “service”, “yang”, “sedang”, “bermasalah”, “?”] | [“apakah”, “service”, “sedang”, “bermasalah”] |
| 3 | Clasification | [“apakah”, “service”, “sedang”, “bermasalah”] | - Question word [“apakah”]<br>- Keyword [“service”]<br>- Status [“problem”]<br>- Time period [“this time”] |
| 4 | Rule-based | - Question word [“apakah”]<br>- Keyword [“service”]<br>- Status [“problem”]<br>- Time period [“this time”] | Service is true<br>Status is true<br>State “problem” |
| 5 | Request | curl(“service”) | XML data |
| 6 | Extrak data for answer | XML data | “service_id”: “218”<br>“host_id”: “214”<br>“name”: “Total Processes”<br>…<br>“current_state”: “1” |
| 7 | Generate answer | “service_id”: “218”<br>“host_id”: “214”<br>“name”: “Total Processes”<br>…<br>“current_state”: “1”<br>… | Berikut status service bermasalah<br>========================<br>ID: 218<br>Nama: Total Process<br>Host: Ozan-Ubuntu<br>Status: CRITICAL |

Table 4 is a description of the whole process in the chatbot system, in which there are seven processes, namely: (a) tokenizing, input in the form of a question sentence, the result is a collection of tokens with the character separator "space"; (b) filtering, input the result of tokenizing, the result is a token that has been filtered from the word stopword; (c) classification, input filtering results, the results are tokens that have been grouped by class; (d) rule-based, input classification results, output in the form of conclusions to request what data is needed to answer; (e) request, input the results of rule-based, output in the form of data about server information according to request; (f) extracting answers, input in the form of a request, output in the form of a Python dictionary information server; (g) preparation of answers, input in the form of the extraction of answers, output answers to be sent to the sender of the question via bot telegram.

### *2.8. Implementation of Ontology*

Implementation of ontology is an advanced stage of the ontology design in the previous chapter. The deployment starts from defining IRI in ontologies like Figure 6, which in this study, IRI from ontologies built is http://www.semanticweb.org/language.



Ontology IRI http://www.semanticweb.org/bahasa

Ontology Version IRI e.g. http://www.semanticweb.org/bahasa/1.0.0

Metrics

| | |
|---|---|
| Axiom | 644 |
| Logical axiom count | 509 |
| Declaration axioms ... | 135 |
| Class count | 6 |
| Object property count | 6 |
| Data property count | 2 |
| Individual count | 121 |
| Annotation Property... | 0 |
| DL expressivity | ALI(D) |

Figure 6 IRI of ontology Bahasa

Then after defining IRI from ontology, it is continued with the class formation with a class reference that has been defined in the ontology design section.

## 3. RESULTS AND DISCUSSION

### *3.1. Research result*

In this study, a chatbot system successfully to built, which in real-time helped inform the server monitoring results by sending a notification message via Telegram to the admin if something happens and can answer questions about the server appropriately. The primary purpose of the chatbot system that built to help inform the results of interactive server monitoring and to be able to process natural language for communication by using rule-based reasoning and ontology-based classification to improve accuracy in answering.

The system built in this study is only used as a bridge to convey information from an existing server monitoring application, where the monitoring application that has been used by the BPK server is Nagios XI. The function of Nagios is to monitor: (a) network services, such as SMTP, POP3, HTTP, NNTP, PING, etc .; and (b) resources, such as processor load, disk usage, memory usage, etc. Nagios also provides various plugins that make it easy, one of which is a backend API that can be used to access multiple monitoring results via URL requests.

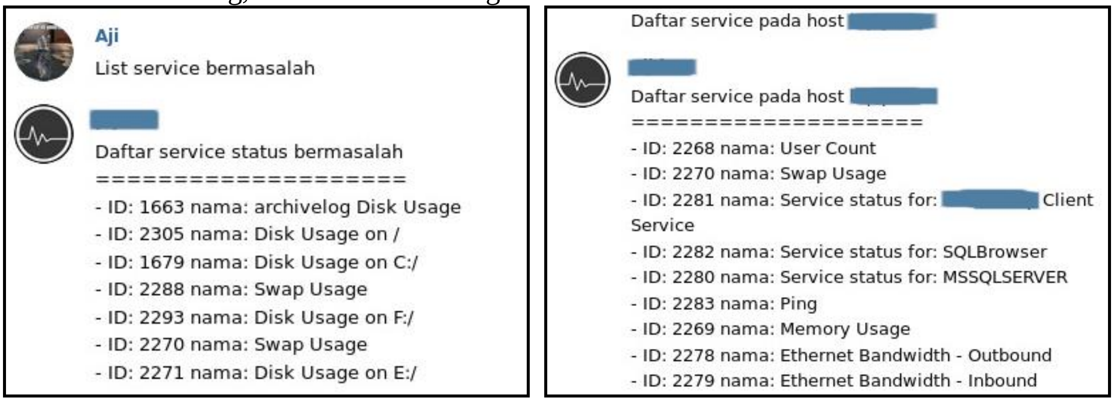
Chatbot server system architecture that is built, this system has two main parts, namely: (a) chatbot system consisting of: tokenizing, filtering, rule-based, request, extracting answers and compiling answers with ontology as media store of language knowledge; and (b) the real-time notification system that is directly carried out by Nagios XI.

The chatbot server system is built using the Python programming language by utilizing several libraries for supporting processes, such as ontology driven, requests, and others, whereas the primary method that applies the rule base is manually created by implementing nested.

How it works from the chatbot server system is by processing the questions that enter through a bot account that has been created using Telegram. The system answers these questions via chat messages according to the sender's user id. The system can only recognize questions with keywords that have been stored on a knowledge base and are limited solely to services and hosts that have been monitored by Nagios XI.

The knowledge base used is stored in an ontology built based on the design of the ontology, which is called the language ontology. In this knowledge base stores all the need to apply to reason using the rule base, words stored in the knowledge base are essential words related to questions about servers that have been classified according to their respective classes.

Chatbot system that is built now has been implemented on the BPK server and has been tested by the CPC server admin, from the results of tests conducted by the parties concerned said they were satisfied. According to their wishes, only suggestions for further research might be able to load more data than just information about the host and service. The following picture is the result of testing, as can be seen in Figure 7.



Aji

List service bermasalah

[redacted]

Daftar service status bermasalah
====================
- ID: 1663 nama: archivelog Disk Usage
- ID: 2305 nama: Disk Usage on /
- ID: 1679 nama: Disk Usage on C:/
- ID: 2288 nama: Swap Usage
- ID: 2293 nama: Disk Usage on F:/
- ID: 2270 nama: Swap Usage
- ID: 2271 nama: Disk Usage on E:/

Daftar service pada host [redacted]

[redacted]

Daftar service pada host [redacted]
====================
- ID: 2268 nama: User Count
- ID: 2270 nama: Swap Usage
- ID: 2281 nama: Service status for: [redacted] Client Service
- ID: 2282 nama: Service status for: SQLBrowser
- ID: 2280 nama: Service status for: MSSQLSERVER
- ID: 2283 nama: Ping
- ID: 2269 nama: Memory Usage
- ID: 2278 nama: Ethernet Bandwidth - Outbound
- ID: 2279 nama: Ethernet Bandwidth - Inbound

Figure 7 Test results on the BPK server

### 3.2. *Test result*

The conclusion from the results of testing the system has obtained an accuracy of 95% with a calculation of 38/40 × 100% = 95%. For example, the description of questions and test results, taken from one of the questions as follows:

| | |
|---|---|
| Example Question | : Ada berapa service bermasalah di host localhost? |
| Answer (+) | : Ada n service bermasalah pada host localhost |
| Answer (-) | : Semua service pada host localhost dalam kondisi baik |
| A question to bot | : Ada berapa service bermasalah di host localhost? |
| Question word | : Ada, berapa |
| Question word by system | : Ada, berapa |
| Keyword | : service, host |
| Keyword by system | : count, service, host |
| Status | : bermasalah |
| Status by system | : problem |
| Service name by system | : - |
| Host name by system | : localhost |
| First answer from system | : Tidak ada service pada host localhost dengan status bermasalah |
| Second answer from system | : Tidak ada service pada host localhost dengan status bermasalah |

## 4. CONCLUSIONS

The conclusion of building an Ontology-Based Chatbot to Support Monitoring of Performance and Server Security by Rule-Base in this research is: Using ontology to represent language knowledge which includes: relations, synonyms, antonym, and word class, allows the system to recognize the types of Indonesian words and can classify according to their class. Using a rule base can interpret input sentences after preprocessing and produce answers that are relevant from the questions. The test results that have been carried out twice with different server conditions obtained stable results with the correctness of the answers reaching 95%.

## REFERENCES

[1] I. Efendi, “Apa Yang di Maksud Dengan Server?”, 2015. [Online]. Available: https://www.it-jurnal.com/apa-yang-di-maksud-dengan-server/. [Accessed: 23-Oct-2017].

[2] A. Nagios, “Nagios Start Up Guide”, *Nagios*, Oct-2017.

[3] A. Følstad, T. Araujo, S. Papadopoulos, and E. L.-C. Law, *Chatbot Research and Design*, vol. 11970. 2019.

[4] A. Kedia and M. Rasu, *Hands-On Python Natural Language Processing*. Birmingham: Packt Publishing, 2020.

[5] S. Vajjala, B. Majumder, A. Gupta, and H. Surana, *Practical Natural Language Processing: A Comprehensive Guide to Building Real-World NLP Systems*. O’Reilly Media, Inc, 2020.

[6] F. T. Admojo, “Sistem Pencarian Informasi Berbasis Ontologi untuk Jalur Pendakian Gunung Menggunakan Query Bahasa Alami dengan Penyajian Peta Interaktif”, Universitas Gadjah Mada, 2015.

[7] G. I. Simari, C. Molinaro, M. Vanina Martinez, T. Lukasiewicz, and L. Predoiu, *Ontology-Based Data Access Leveraging Subjective Reports*. Cham: Springer International Publishing, 2017.

[8] J. M. Mendel, *Uncertain Rule-Based Fuzzy Systems*. Cham: Springer International Publishing, 2017.

[9] H. R. Sakinnah, “Rule Based System dengan Prediksi Learning Style Berdasarkan Metakognitif”, Universitas Gadjah Mada, 2016.

[10] N. F. Noy and D. L. McGuinness, “Ontology Development 101: A Guide to Creating Your First Ontology”, *Stanford Knowl. Syst. Lab.*, p. 25, 2001.