Available online at http://jurnal.ahmar.id/index.php/asci

Journal of Applied Science, Engineering, Technology, and Education

ISSN 2685-0591 (Online)

Journal of Applied Science, Engineering, Technology, and Education Vol. 2 No. 2 (2020)
https://doi.org/10.35877/454RI.asci2259

Nagios Core Optimization By Utilizing Telegram as Notification of Disturbance

Fahreza[a,*], Muhammad Rifqi[b]

[a]*Faculty of Computer Science, University Mercu Buana, Indonesia, 41515110112@student.mercubuana.ac.id*
[b]*Faculty of Computer Science, University Mercu Buana,Indonesia, m.rifqi@mercubuana.ac.id*

Abstract

Network Monitoring System (NMS) is a system that is highly demanded internet service provider industry in this fast-developing information technology era. The availability of NMS is the best option to restore the service level agreement as a means to compete with other internet service providers' competitors. The occurrence of disturbance in the network is often unnoticed by the network administrator. This may lead to a crucial problem in decreasing network quality as the impact of time-consuming in solving the problem. Through the explanation, the writer tried to anticipate by classifying problems using Pareto, and integrated Nagios with Telegram Messenger as a notification of disturbance. Nagios has many features such as reports, event handler, monitoring resource (CPU load, memory usage, status up / down, up time, data traffic, bandwidth), etc. One of notable feature owned by Nagios is blast notification of disturbance. It is a feature that will function when one of the devices is in trouble. This feature will inform the network administrator or authorized person in a certain divisions as regards the error network. In this case, the problematic device can be categorized according to the parameters made by the network administrator.

Keywords: Network, Monitoring, Nagios Core, Telegram, Blast Notification.

1. Introduction

Network Monitoring System (NMS) is a system that is highly demanded internet service provider industry in this fast-developing information technology era. There are three things to consider in managing complex networks, the structure, management, and effectiveness of the network [1].

The availability of NMS is the best option to restore the service level agreement as a means to compete with other internet service providers' competitors. In order to keep the track of information about the network device such as a server, router, switch, and endpoint devices run as it is, NMS is inevitably right [2].

[*] Corresponding author.
E-mail address: 41515110112@student.mercubuana.ac.id (Fahreza)

The occurrence of disturbance in the network is often unnoticed by the network administrator. This may lead to a crucial problem in decreasing network quality as the impact of time-consuming in solving the problem.

The monitoring absence of system operators toward the running application in NMS in the network operation center leads to a problem. This presumably occurs due to personal or supplementary activity, which urges the system operator to check out the NOC room.

A multinational level company, which works in the internet service provider sector, has been in the same state of such problem. Due to the overdue initial problem-solving in analyzing the cause of certain troubled device gives impact to a reduction on SLA approval according to the agreement.

A relevant application which may help to minimize the effect of the problem is Nagios. Nagios has many features such as reports, event handler, monitoring resource (CPU load, memory, status up / down, uptime, data traffic, bandwidth) and etc. One of notable feature owned by Nagios is blast notification alert. It is a feature that will function when one of the devices is in trouble. This feature will inform the network administrator or authorized person in a certain division as regards the error network.

The error device can be categorized by predetermined parameters that have been set or made by the network operators. The error notification can be transmitted to various ways of communication through e-mail, short message service (SMS), even WhatsApp messenger or Telegram.

2. Methodology

The methods in conducting the research are:

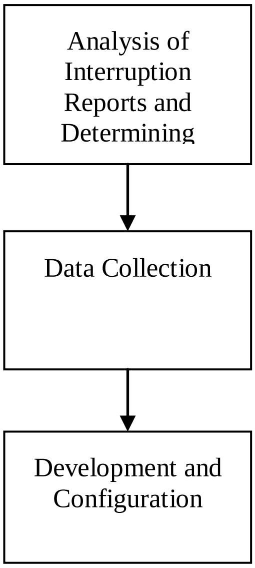

Analysis of Interruption Reports and Determining
Data Collection
Development and Configuration

Fig.1 Research Flow

2.1 Analysis of Interruption Reports and Determining Problems

This research uses the Pareto method to classify the problems in order to maintain SLA approval according to the agreement. By using SLA calculation illustration in a month, if the internet service provider offers 99% SLA and 1% of the SLA is still within the tolerance limit.

SLA Illustration:

If SLA is 99%
SLA = 1 Month x 99%
SLA = 720 Hours x 99%
SLA = 712.8 Hours = 713 Hours

From the calculation above, it can be concluded that 713 hours is a guarantee given by the internet service provider. If there is a problem occur within 7 hours, this is still tolerable. On the other hand, if there is an internet connection interruption over 7 hours, customers can ask for restitution to the internet service provider.

Pareto is chart or image that sort the data classification from left to right following the highest-ranking order to the lowest, hence finding the problems that often occur is able to be resolved immediately (based on the highest to the lowest ranking [3].

The steps of Pareto:

1. Deciding the method or the meaning of data classification, which based on the problem, the cause, deviation types, etc.;
2. Determine the unit for scaling the characteristic order in example currency (rupiahs), frequency, units, and etc;
3. Collect the data according to the specified time interval;
4. Summarize the data and create data category level from the biggest to the smallest;
5. Calculate the used frequency or percentage cumulative; and
6. Drawbar chart shows the relative importance of the disturbance and identify some essential things to observe later.

According to the data that has been acquired, the writer categorizes some important highlight:

A= Worn Out Device
B= Checking Duration on the Field
C= Duration of the Received Information
D= Power Failure
E= Loss of Configuration

Problem Percentage Formula:

$$\text{Percentage of Problem} = \frac{\text{Amount of value problem exist}}{\text{Total the overall value of problems}} \times 100$$

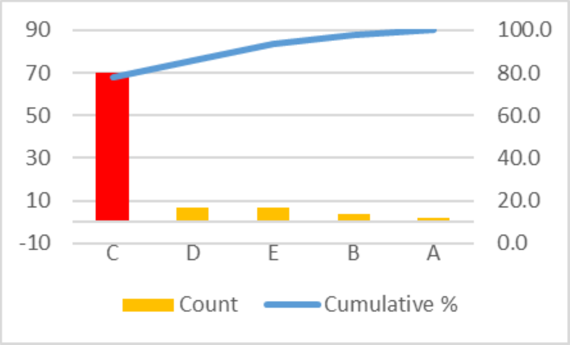


Fig.2 Pareto

As is seen in Figure 2, the most frequent disturbance in August 2019 is the Duration of the Received Information to the operator or the network administer. This problem shows that slow information exchange between the departments affects the decreased SLA [4].

Figure 2 on C category, (Duration of the Received Information) has 10 times disturbance with downtime for 675 minutes or 11 hours 25 minutes. In other words, this table shows that the C category exceeds SLA (675 minutes) that guaranteed by the internet service provider. Therefore, the writer limits the discussion on how to overcome the problem in C, meanwhile, the other four categories A, B, D, and E do not exceed the guaranteed period SLA (Downtime within 420 Minutes). The downtime data for each category is presented in Table 1.

Table 1. Data Problems in August 2019

Root Cause	Total Cause	Total Cause Per Minute
C	10	675
D	3	360
B	2	216
E	1	209
A	1	67
Total	17	1527

2.2 Data Collection

The required data for the writer is obtained from direct observation in the form of Escalation Process Flowchart, Existing Topology, and disturbance report from one of the company for a month, start from August 1, 2019, to August 31, 2019, as the data analysis the cause of decreasing SLA.

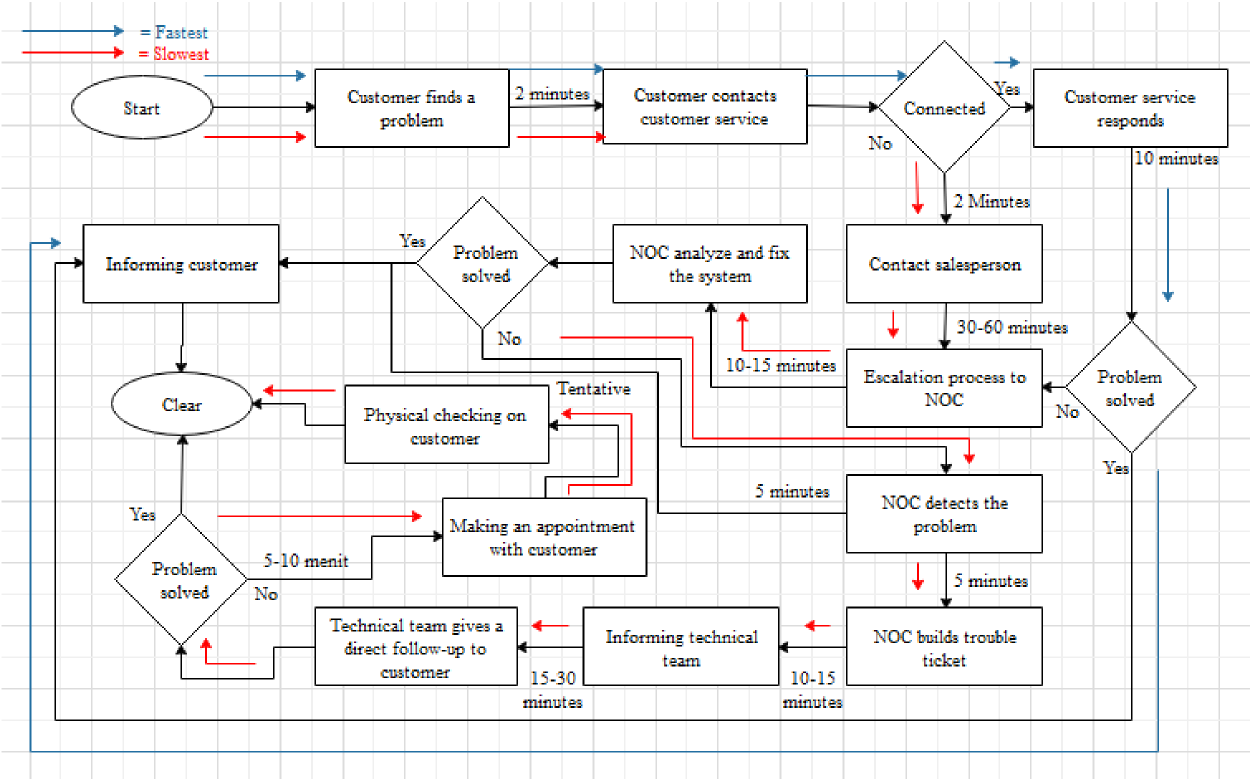


Fig.3 Escalation Process Flowchart

Figure 3 shows the description of the escalation process length significantly affect SLA guaranteed. Due to the ineffective process, only by noticing each process takes 2 to 30 minutes. This may cause wasting more time when the technical team organizes visit period for further checking, time consume needed is more likely to be tentative since it is possible to hold a visitation the next day under the customer agreement.

If the escalation time is calculated excluding tentative visitation time, it spends more than 100 minutes while the customer submits a complaint through customer service. On the contrary, if the customer makes a complaint to the salesperson, it unavoidably spends around 150 minutes only to confirm visitation in order to check the network device.

This condition is getting more deteriorate and ineffective when the administrators are not available or present in the place so that the process of disturbance causal analysis takes more time. Before conducting system development, it is an urgency to take previous existed network topology-related data. The following chart is the existing topology:

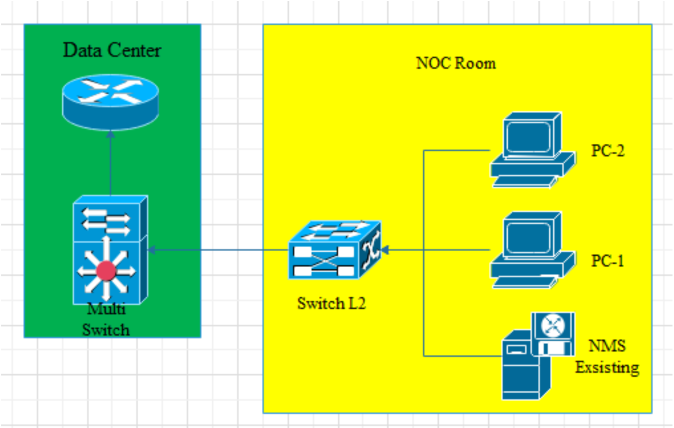


Fig.4 Existing Topology

From Figure 4 it can be inferred that only the NMS server existing in NOC room and required to watch over frequently.

In order to handle the problem, Nagios with blast alert by Telegram feature is desired to accelerate network administrator in receiving information faster if there is any troublesome network device (in accordance with predefined parameters by the network administrator) and faster in analyzing and handling to minimize downtime which may influence SLA.

Below is the suggested topology by implementing the data collection that has been gathered as a developing system:

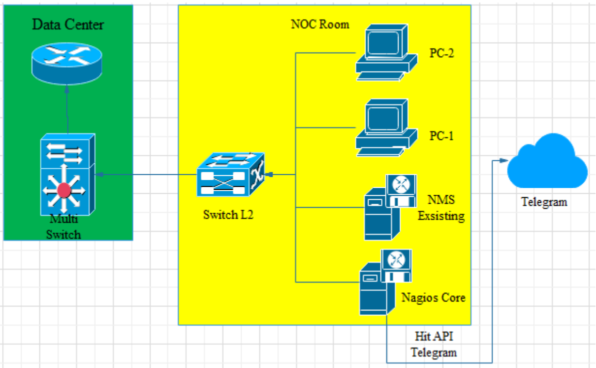


Fig.5 Suggested Topology

From figure 5 suggested by the writer, the network administrator is not necessarily to be in NOC room to keep track of the interfered network devices. By overviewing the suggested topology, it is possible to alter the escalation process.

Furthermore, the writer proposes a specification server needed to install Nagios Server after the writer suggests a new suggested network topology, as it follows table 2.

Table 2. Minimum Specification

Specification
- Hard Drive 20GB; - Memory 2 GB CPU Dual Core, 2,4 Ghz; - Operating System CentOS or Red Hat Enterprise Linux (RHEL) Version 5, 6 or 7.

Source: https://www.nagios.org/

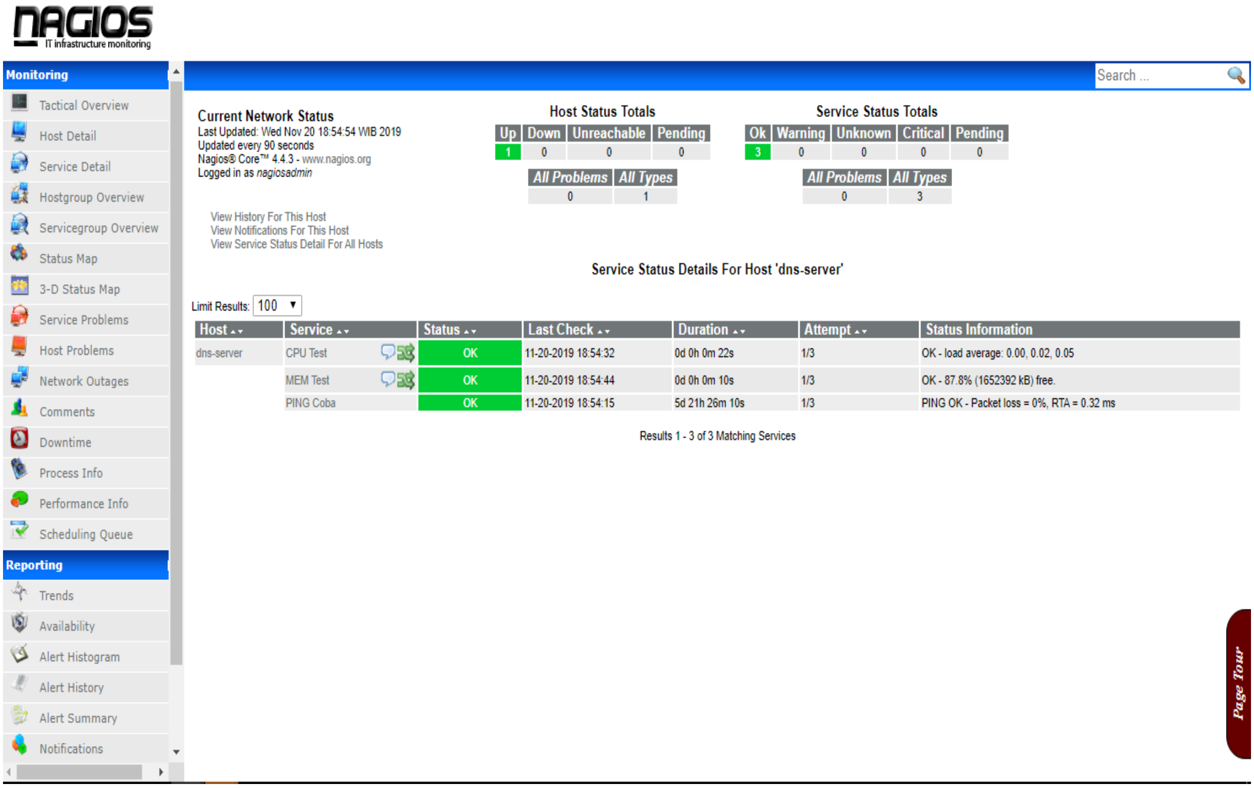
Referring to the point where Nagios is a based on open-source software, is it expected to be able to reduce cost, since this open-source application is free from license fee, enable the user to customized as s/he expected and minimize hijacking licensed application [5].

On the assumption that the server is available according to minimum specification in the figure 2, there are some things need to be considered before making installation and configuration Nagios on the server provided. These requirements are: the server that has been provided by the server must have apache installed as webserver whose job is to display the web GUI (Graphical User Interface), PHP (Hypertext Pre Processor), NRPE (Nagios Remote Plugins Executor) already installed on the agent want to monitor and other services needed in the future. NRPE is a plugin which provides information NMS required such as Ping (Packet Internet Gohper), SSH (Secure Shell), CPU (Central Processing Unit) Load, and etc. The next step is installation Nagios along with Agent and SNMP (Simple Network Management Protocol).

2.3 Development and Configuration

In this development and configuration method section, the writer explains certain things to consider during Nagios Server configuration, NRPE as well as SNMP configuration.

The important highlight to avoid failure during Nagios installation is, the user is expected to turn off the firewall and Selinux so that the port Nagios will use for monitoring the system will not be restrained from the firewall and Selinux itself. Another following step to be noticed is to allow service Nagios runs after booting finish (startup process) so that it is not necessary to activate the command line manually.



Current Network Status
Last Updated: Wed Nov 20 18:54:54 WIB 2019
Updated every 90 seconds
Nagios® Core™ 4.4.3 - www.nagios.org
Logged in as *nagiosadmin*

View History For This Host
View Notifications For This Host
View Service Status Detail For All Hosts

Host Status Totals

Up	Down	Unreachable	Pending
1	0	0	0

All Problems	*All Types*
0	1

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
3	0	0	0	0

All Problems	*All Types*
0	3

Service Status Details For Host 'dns-server'

Limit Results: 100

Host	Service	Status	Last Check	Duration	Attempt	Status Information
dns-server	CPU Test	OK	11-20-2019 18:54:32	0d 0h 0m 22s	1/3	OK - load average: 0.00, 0.02, 0.05
	MEM Test	OK	11-20-2019 18:54:44	0d 0h 0m 10s	1/3	OK - 87.8% (1652392 kB) free.
	PING Coba	OK	11-20-2019 18:54:15	5d 21h 26m 10s	1/3	PING OK - Packet loss = 0%, RTA = 0.32 ms

Results 1 - 3 of 3 Matching Services

Fig. 6 Dashboard Nagios Core

Figure 6 shows the display of installed GUI Nagios and rendered host, services and additional features in which Nagios Core will monitor. Iso Nagios Core or installer is available on Nagios official website (https://www.nagios.org/).

After making sure NRPE and SNMP configured prudently, the nest action to perform is Telegram Messenger. Nagios Server will be integrated with Telegram, as a notification of interruption with the workflows shown in figure 7.

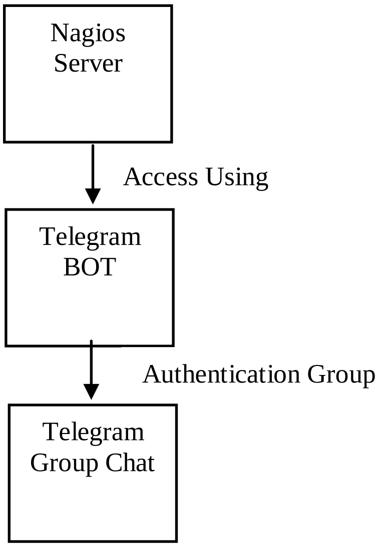
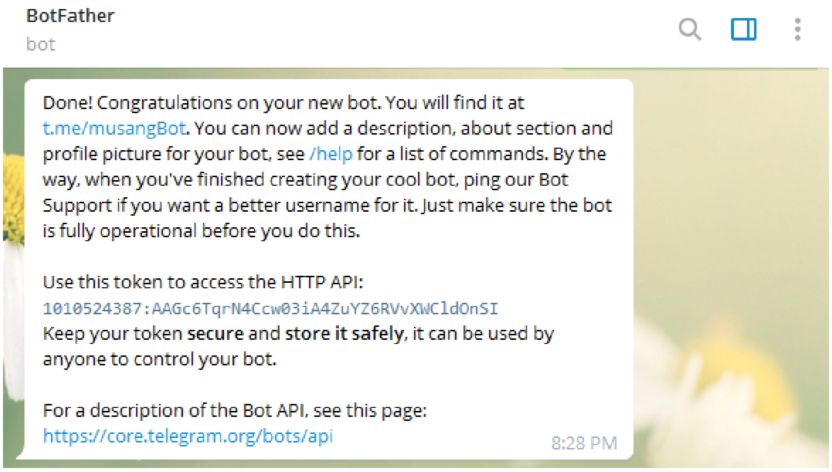


Fig.7 Disturbance Workflow

Referring to notification of interruption workflow, the initial action to overcome this condition is to make BOT (a computer program to run automatic command) to receive a token as a tool command BOT as a notification alert sender. BOT Telegram can be performed from BOT Father.



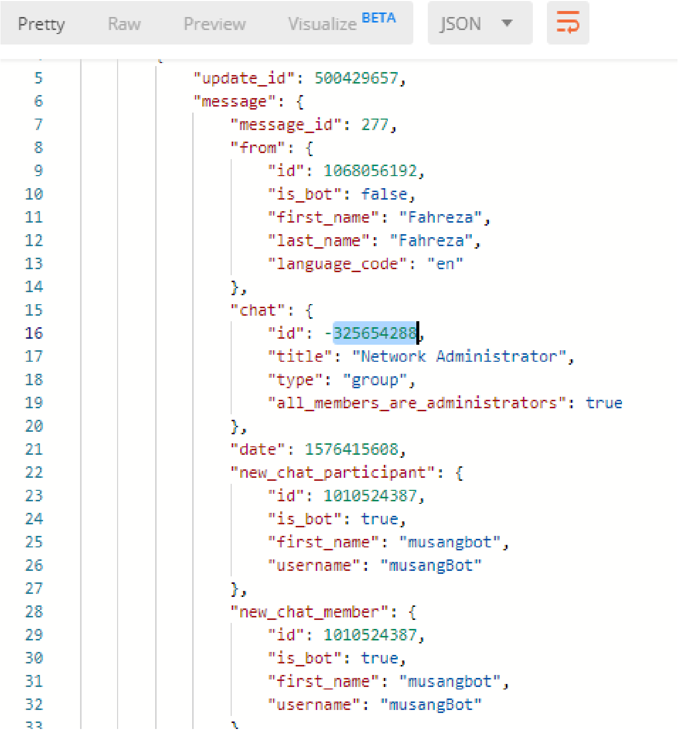
Done! Congratulations on your new bot. You will find it at t.me/musangBot. You can now add a description, about section and profile picture for your bot, see /help for a list of commands. By the way, when you've finished creating your cool bot, ping our Bot Support if you want a better username for it. Just make sure the bot is fully operational before you do this.

Use this token to access the HTTP API:
1010524387:AAGc6TqrN4Ccw03iA4ZuYZ6RVvXWCldOnSI
Keep your token **secure** and **store it safely**, it can be used by anyone to control your bot.

For a description of the Bot API, see this page:
https://core.telegram.org/bots/api
8:28 PM

Fig. 8. BOT Token

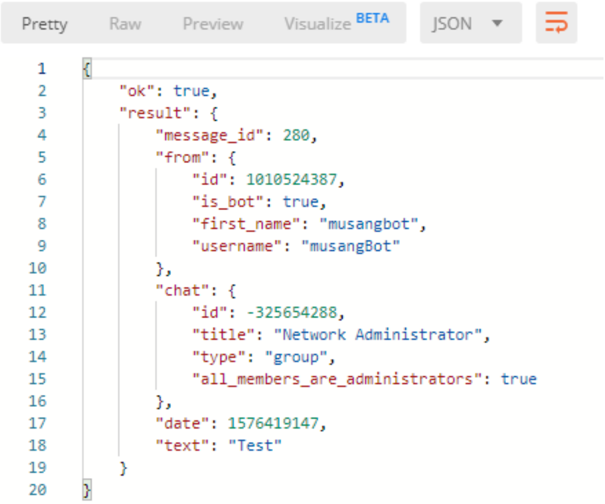
After successfully creating BOT as shown in figure 8, the next following action is to create a group that has been installed BOT, consisting of networks administrators or people from Network division. This is aimed to obtain the ID group for BOT determinant the desired group chat.



```
        "update_id": 500429657,
        "message": {
            "message_id": 277,
            "from": {
                "id": 1068056192,
                "is_bot": false,
                "first_name": "Fahreza",
                "last_name": "Fahreza",
                "language_code": "en"
            },
            "chat": {
                "id": -325654288,
                "title": "Network Administrator",
                "type": "group",
                "all_members_are_administrators": true
            },
            "date": 1576415608,
            "new_chat_participant": {
                "id": 1010524387,
                "is_bot": true,
                "first_name": "musangbot",
                "username": "musangBot"
            },
            "new_chat_member": {
                "id": 1010524387,
                "is_bot": true,
                "first_name": "musangbot",
                "username": "musangBot"
```

Fig. 9 ID Group

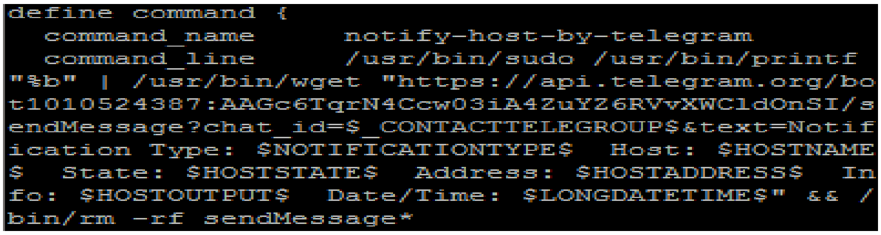
Right after receiving token (1010524387) and ID group (-325654288), the writer did a trial through postman using HIT API Telegram containing parameter of the token and ID group as shown in figure 10.



```
{
    "ok": true,
    "result": {
        "message_id": 280,
        "from": {
            "id": 1010524387,
            "is_bot": true,
            "first_name": "musangbot",
            "username": "musangBot"
        },
        "chat": {
            "id": -325654288,
            "title": "Network Administrator",
            "type": "group",
            "all_members_are_administrators": true
        },
        "date": 1576419147,
        "text": "Test"
    }
}
```

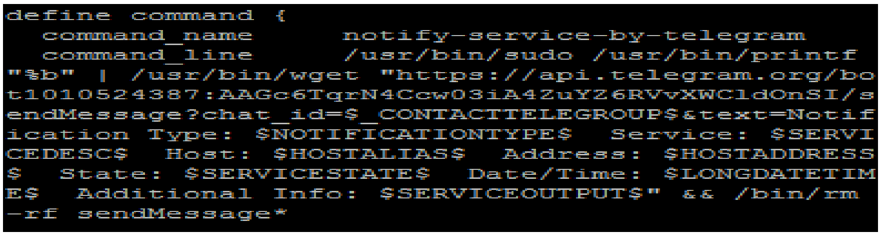
Fig. 10 Hit API Telegram

Eventually, the writer configured on Nagios Server side by turning on notify disturbance for hosts and services desired to be monitored, as shown in figure 11 and figure 12.



```
define command {
   command_name      notify-host-by-telegram
   command_line      /usr/bin/sudo /usr/bin/printf
"%b" | /usr/bin/wget "https://api.telegram.org/bo
t1010524387:AAGc6TqrN4Ccw03iA42uYZ6RVvXWCldOnSI/s
endMessage?chat_id=$_CONTACTTELEGROUP$&text=Notif
ication Type: $NOTIFICATIONTYPE$  Host: $HOSTNAME
$  State: $HOSTSTATE$  Address: $HOSTADDRESS$  In
fo: $HOSTOUTPUT$  Date/Time: $LONGDATETIME$" && /
bin/rm -rf sendMessage*
```

Fig. 11 Notify Host



```
define command {
   command_name      notify-service-by-telegram
   command_line      /usr/bin/sudo /usr/bin/printf
"%b" | /usr/bin/wget "https://api.telegram.org/bo
t1010524387:AAGc6TqrN4Ccw03iA42uYZ6RVvXWCldOnSI/s
endMessage?chat_id=$_CONTACTTELEGROUP$&text=Notif
ication Type: $NOTIFICATIONTYPE$  Service: $SERVI
CEDESC$  Host: $HOSTALIAS$  Address: $HOSTADDRESS
$  State: $SERVICESTATE$  Date/Time: $LONGDATETIM
E$  Additional Info: $SERVICEOUTPUT$" && /bin/rm
-rf sendMessage*
```

Fig. 12 Notify Services

From figure 11, the configuration performs order for Nagios to run blast alert if there is an inactive host. Meanwhile, figure 11 shows the configuration runs the command for Nagios performs blast alert if there is monitored service undergo any disturbance. In the command line, a link from API telegram is detected. To get the API, the user can access on api.telegram.org.

3. Result dan Discussion

In this section, the writer did a trial toward NRPE and SNMP to obtain the average time needed in the process of sending disturbance notification through telegram using a formula to find the Mean value in stress test data selection method and shut down the Nagios monitored device. The Mean value from data collection is gathered from adding the numbers in the data set and dividing the number of individual numbers [6]. In statistics, this term is called mean arithmetic and symbolized with M, the formula is shown as follows:

$$M = \frac{\sum X}{N}$$

with:
M = Mean (Average Value)
$\sum X$ = Sum of Value
N = Number of Individuals

Before doing the trial toward NRPE and SNMP, the writer executed a test through a standard network protocol. The test is done using ICMP protocol or Ping in general term.

Table 3. Ping Average of Value

Test	**Disbanded Device Period**	**Telegram Disturbance Info**	**Range**
1st	2019-12-10 17:10:14	2019-12-10 17:12:04	0:01:50
2nd	2019-12-11 12:20:23	2019-12-11 12:21:24	0:01:01
3rd	2019-12-11 19:21:42	2019-12-11 19:23:03	0:01:21
4th	2019-12-12 10:08:51	2019-12-12 10:10:11	0:01:20
5th	2019-12-12 14:14:25	2019-12-12 14:15:48	0:01:23
6th	2019-12-13 13:49:18	2019-12-13 13:51:27	0:02:09
7th	2019-12-14 01:32:56	2019-12-14 01:34:07	0:01:11
8th	2019-12-14 09:21:13	2019-12-14 09:22:32	0:01:19
9th	2019-12-16 11:41:37	2019-12-16 11:43:06	0:01:29
10th	2019-12-16 22:16:03	2019-12-16 22:17:14	0:01:11
	Average		0:01:25

The result of ICMP protocol or Ping test is conducted by disbanding the device and it results in 85 seconds average number for the information reach network administration. The value is presented in Table 3.

3.1 NRPE Test

The table 4 is the result of functional NRPE trial toward device server monitored by Nagios.

Table 4. Average Value of CPU Load via NRPE

Test	Disbanded Device Period	Telegram Disturbance Info	Range
1st	2019-12-10 15:46:21	2019-12-10 15:47:36	0:01:15
2nd	2019-12-11 12:25:01	2019-12-11 12:27:31	0:02:30
3rd	2019-12-11 19:30:15	2019-12-11 19:32:19	0:02:04
4th	2019-12-12 10:17:54	2019-12-12 10:19:37	0:01:43
5th	2019-12-12 14:39:42	2019-12-12 14:42:01	0:02:19

Test	Disbanded Device Period	Telegram Disturbance Info	Range
6th	2019-12-13 13:56:09	2019-12-13 13:58:24	0:02:15
7th	2019-12-14 01:40:26	2019-12-14 01:42:21	0:01:55
8th	2019-12-14 09:37:29	2019-12-14 09:39:38	0:02:09
9th	2019-12-16 12:03:12	2019-12-16 12:06:16	0:03:04
10th	2019-12-16 22:47:02	2019-12-16 22:48:54	0:01:52
	Average		0:02:07

The result of functional CPU load via NRPE test using the stress test method obtains average value 127 seconds to reach the network administration. The value result is presented in Table 4.

Table 5. Memory Usage Average Value via NRPE

Test	Disbanded Device Period	Telegram Disturbance Info	Range
1st	2019-12-10 16:13:22	2019-12-10 16:14:32	0:01:10
2nd	2019-12-11 12:33:02	2019-12-11 12:35:07	0:02:05
3rd	2019-12-11 19:51:48	2019-12-11 19:52:58	0:01:10
4th	2019-12-12 10:46:09	2019-12-12 10:47:51	0:01:42
5th	2019-12-12 14:57:35	2019-12-12 14:59:47	0:02:12
6th	2019-12-13 14:17:36	2019-12-13 14:18:53	0:01:17
7th	2019-12-14 01:52:48	2019-12-14 01:54:08	0:01:20
8th	2019-12-14 09:48:31	2019-12-14 09:49:55	0:01:24
9th	2019-12-16 12:28:19	2019-12-16 12:30:58	0:02:39
10th	2019-12-16 23:09:03	2019-12-16 23:11:24	0:02:21
	Average		0:01:44

Functional Memory Usage via NRPE test result using stress test method acquires 104 seconds of average time for the network administrator to receive the information. The result is presented in Table 5.

3.2 SNMP Test

This section indicates the test result of functional SNMP toward the network devices monitored by Nagios.

Table 6. Port Status Average Value via SNMP

Test	Disbanded Device Period	Telegram Disturbance Info	Range
1st	2019-12-10 16:30:01	2019-12-10 16:31:51	0:01:50
2nd	2019-12-11 12:45:08	2019-12-11 12:46:42	0:01:34
3rd	2019-12-11 20:10:39	2019-12-11 20:12:04	0:01:25
4th	2019-12-12 11:32:34	2019-12-12 11:34:47	0:02:13
5th	2019-12-12 15:21:28	2019-12-12 15:23:23	0:01:55
6th	2019-12-13 14:36:12	2019-12-13 14:38:05	0:01:53
7th	2019-12-14 02:00:04	2019-12-14 02:01:37	0:01:33

8th	2019-12-14 10:00:12	2019-12-14 10:02:32	0:02:20
9th	2019-12-16 12:50:13	2019-12-16 12:51:32	0:01:19
10th	2019-12-16 23:25:37	2019-12-16 23:27:05	0:01:28
	Average		0:01:45

Functional test result for Port Status via SNMP by disbanding the device shows 105 seconds average time to connect with network administrator. The value is presented in Table 6.

Table 7. Up Time Device Average Value via SNMP

Test	**Disbanded Device Period**	**Telegram Disturbance Info**	**Range**
1st	2019-12-10 16:37:12	2019-12-10 16:39:21	0:02:09
2nd	2019-12-11 12:52:49	2019-12-11 12:55:35	0:02:46
3rd	2019-12-11 20:17:52	2019-12-11 20:20:02	0:02:10
4th	2019-12-12 11:43:45	2019-12-12 11:45:07	0:01:22
5th	2019-12-12 15:32:57	2019-12-12 15:35:07	0:02:10
6th	2019-12-13 14:36:12	2019-12-13 14:38:05	0:01:53
7th	2019-12-14 02:09:48	2019-12-14 02:11:32	0:01:44
8th	2019-12-14 10:24:56	2019-12-14 10:26:16	0:01:20
9th	2019-12-16 13:01:13	2019-12-16 13:03:17	0:02:04
10th	2019-12-16 23:37:37	2019-12-16 23:39:56	0:02:19
	Average		0:02:00

The functional test result of Up Time Devices via SNMP by turning off the device results in 120 seconds average value to access the network administrator. The result is shown in Table 7.

Table 8. CPU Load Average Value via SNMP

Test	Disbanded Device Period	Telegram Disturbance Info	Range
1st	2019-12-10 17:03:45	2019-12-10 17:05:51	0:02:06
2nd	2019-12-11 13:07:21	2019-12-11 13:10:03	0:02:42
3rd	2019-12-11 20:17:52	2019-12-11 20:20:02	0:02:10
4th	2019-12-12 11:47:41	2019-12-12 11:50:13	0:02:32
5th	2019-12-12 15:32:57	2019-12-12 15:35:07	0:02:10
6th	2019-12-13 14:36:12	2019-12-13 14:38:05	0:01:53
7th	2019-12-14 02:20:37	2019-12-14 02:22:17	0:01:40
8th	2019-12-14 10:39:42	2019-12-14 10:41:22	0:01:40
9th	2019-12-16 13:18:31	2019-12-16 13:19:58	0:01:27
10th	2019-12-16 23:51:59	2019-12-16 23:53:29	0:01:30
	Average		0:01:59

The functional CPU Load via SNMP test result using stress test method indicates that the average value of the system to reach the network administrator is 119 seconds. The result is delivered in Table 8.

From all the six tests conducted by disbanding the device and stress test method as presented above, the result of average time all NRPE and SNMP delivers in Table 9.

Table 9. Average Value and Total of Entire Trial Sub Unit

No	Test Result	Average time
1	Ping Test	0:01:25
2	CPU Load NRPE Test	0:02:07
3	Memory Usage NRPE Test	0:01:44
4	Port Status SNMP Test	0:01:45
5	Up Time Device SNMP Test	0:02:00
6	CPU Load SNMP Test	0:01:59
	Total Period	0:11:00
	Average	0:01:50

By optimizing Nagios Core using telegram as a disturbance notification, signify that the information interfered device will reach the network administrator takes time for ≥ 2 minutes. This condition replaces the escalation process, in which Figure 22 shows the flow before the procedure and the flow after the procedure is in Figure 13.

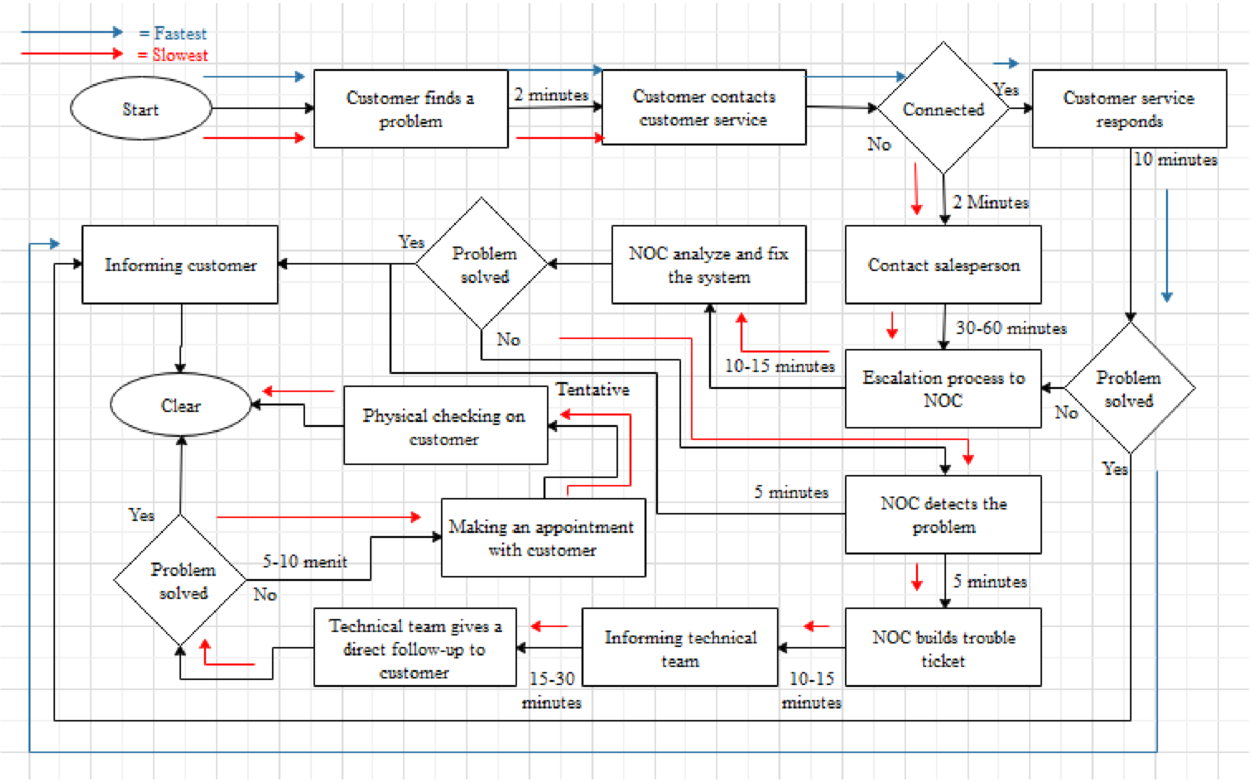


Figure 13. Escalation Process Flowchart before Implementation

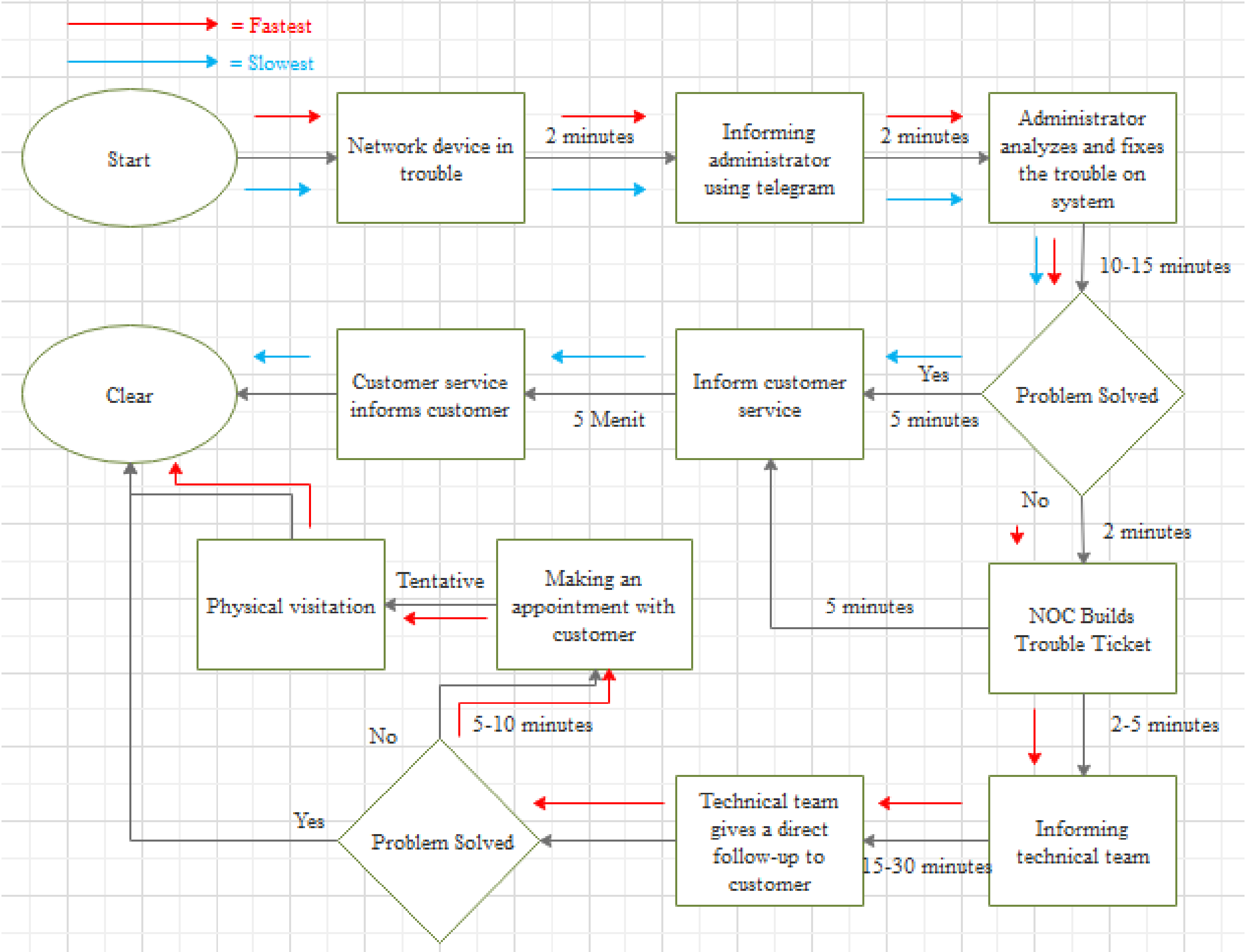

= Fastest
= Slowest
Start
Network device in trouble
2 minutes
Informing administrator using telegram
2 minutes
Administrator analyzes and fixes the trouble on system
10-15 minutes
Problem Solved
Yes
5 minutes
Inform customer service
5 Menit
Customer service informs customer
Clear
No
2 minutes
NOC Builds Trouble Ticket
5 minutes
Making an appointment with customer
Tentative
Physical visitation
2-5 minutes
Informing technical team
15-30 minutes
Technical team gives a direct follow-up to customer
Problem Solved
Yes
No
5-10 minutes

Fig. 14 Escalation Process Flowchart after Implementation

In Figure 14, the escalation process flowchart cuts time up to 84 minutes for technical team contacts the customer to make an appointment for a physical visit.

Table 10. Time Required for the Escalation Process

Escalation Process Flowchart	Before Implementation	After Implementation
The slowest	150 minutes	66 minutes
The fastest	100 minutes	29 minutes

The data in Table 10 presents the time of the escalation process from Figures 13 and 14 to make an appointment for a physical visit. It spends the longest time for 150 minutes and 100 minutes the fastest. After implementing the system proposed by the writer, the longest time to make an appointment for a physical visit is 66 minutes and the fastest time is 29 minutes. As a result, the implemented system is capable to solve the problem in C category. This progress satisfies network administrators to enable controlling the network device without being present in NOC room. However, the implemented system is unable to solve all problems occur in one of the multinational company.

4. Conclusion

Nagios has many features such as reports, event handler, monitoring resource (CPU load, memory usage, status up / down, up time, data traffic, bandwidth), etc. One of notable feature owned by Nagios is blast notification of disturbance. It is a feature that will function when one of the devices is in trouble. This feature will inform the network administrator or authorized person in a certain divisions as regards the error network. In this case, the problematic device can be categorized according to the parameters made by the network administrator.

Finally yet importantly, the writer expects the readers to proceed with this research as a next step in determining a solution that has not completed at the moment. Hopefully, there will be continuation regards with the development of this currently-used system or designing a new system to be integrated as a new sophisticated system that can be the solution to respond to system disturbances.

References

[1] Sholikatin, Y., & Rosyid, N.R. 2017. *Implementasi Fault Management (Manajemen Kesalahan) Pada Network Management System (NMS) Berbasis SNMP*. Jurnal Teknik Informatika Dan Sistem Informasi, 3(2), 354–364. doi: 10.28932/jutisi.v3i2.637

[2] Wulandoro, A., Nurkahfi, G. N., & Fitriyani. 2016. *Desain , Implementasi , Dan Analisis Network Management System (Nms) Berbasis Cacti Design , Implementation , and Analysis of Network Management System (Nms) Based on Cacti*. E-Proceeding of Engineering, 3(1), 1199–1205. Retrieved from https://openlibrary.telkomuniversity.ac.id/pustaka/*file*s/107492/jurnal_eproc/desain-implementasi-dan-analisis-network-management-system-nms-berbasis-cacti.pdf

[3] Parayu, S., & Warman, P. (2017). *ANALISIS FAKTOR PENYEBAB CACAT PENGELASAN PADA PIPA (Study Kasus Pada Pipa Distribusi PDAM Kabupaten Kutai Barat)*. 8(2), 730–736.

[4] Sucipto, S., Prima Sulistyowati, D., & Anggarini, S. 2017. *Quality Control of Mushrooms Canning using Six Sigma Method at Company Y*, Pasuruan, East Java. Industria: Jurnal Teknologi Dan Manajemen Agroindustri, 6(1), 1–7. doi:10.21776/ub.industria.2017.006.01.1

[5] Yanto, J. 2016. *Implementasi Sistem Monitoring Server Menggunakan Nagios*. STTI NIIT I-Tech, (Selisik), 164–169. Retrieved from http://jitter.widyatama.ac.id/index.php/Selisik2016/article/download/126/103

[6] Abidin, Z., & Purbawanto, S. 2015. Pemahaman Siswa Terhadap Pemanfaatan Media Pembelajaran Berbasis Livewire Pada Mata Pelajaran Teknik Listrik Kelas X Jurusan Audio Video Di Smk Negeri 4 Semarang. Edu Elektrika Journal, 4(1), 38–49.