# Teleoperation Cloud Industrial Robot using XMPP Protocol

**Rachmad Andri Atmoko, Daoguo Yang, Mohamad Ilyas Abas, Afif Zuhri Arfianto, Robbi Rahim**

***Abstract*: *Teleoperation is an operation needed to complete a variety of robot tasks in dangerous and difficult-to-reach environments. In the industrial world, teleoperation robots can be used to help humans complete dangerous tasks such as grasping and lifting in the nuclear industry. Controlling industrial robots can be done remotely via the Internet. However, communication still needs to be improved so that it can achieve low latency, be safe, high reliability, and be interoperable. This study proposes the use of the XMPP protocol to handle data interactions between robots and cloud platform. We modified the Robot Operating System (ROS) package by adding a special node that bridges communication using the XMPP protocol. We compare the performance of MQTT with the HTTP protocol. The test results show that the MQTT latency time is better than HTTP. The results of this study can be used in robot teleoperation operations in various industries that consider aspects of communication quality.***

***Keywords* : *industrial robot, teleoperation, xmpp, internet of things***

## I. INTRODUCTION

Nowadays, the Internet of Things (IoT) is a new paradigm that provides significant changes in various fields of life. This paradigm could enable enormous economic benefit for the industrial sector when all connected objects can be monitored and controlled remotely using an internet network. IoT implementation in this sector called the Industrial Internet of Things (IIoT). It may contribute data explosions on the Internet since various industrial connected equipment supplied extensive data to the cloud [1]. The manufacturing sector provides a higher data supply than in other sectors. Single machines can produce 5000 data every 33 ms, which produces four trillion data per year [2].

The robot is one of the modern machines which used popularly in the industrial environment. It used for various functions such as grasping and lifting[3]. Today, industrial robots bring up a new concept called cloud robotics [4]. Teleoperation is one of the operations used in this concept. This concept uses cloud computing technology to eliminate the limitations of conventional computers that have been used to operate robots. However, communication between robots and cloud platforms is still a technical problem that must be resolved to implement this concept [4]. In the concept of cloud robotics, robots are considered to have weaknesses in the limitations of computing resources, low power, and cannot handle heavy computing tasks [5]. Thus the massive task is transferred to the cloud side, which has high computational resource capabilities. Hence, communication must be run efficiently and quickly with low latency. The following research provides an overview of the development of research on remote robotic control field that has been done. There was research which proposes a robot remote control system based on VPN (Virtual Private Network) and TCP / IP protocols [6]. Other researchers used the UDP protocol for transmitting the data control robot over the wireless Internet, which can achieve a low packet delay and low packet error [7]. Another research made a remote robot control system for inspection and monitoring used the SSH protocol on a snake-like robot via a WIFI network. The iOS-based application was built to overcome security weaknesses, namely, network attacks on IP-based systems [8]. All of the research has not ideal latency for remote robot control applications.

Mostly IoT project implement IoT protocol, such as the application of real-time data acquisition [9], Train anti-collision system [10], intelligent parking space services [11]. All of these studies produce latency times of less than one second.

This study proposes a solution using XMPP protocol for robot control which has advantages over other IoT protocols, namely safe and interoperable [12]. XMPP is recommended to solve interoperability problems in heterogeneous network environments. This protocol present trust connection mechanism. The clients need to know each other before establishing a connection. XMPP protocol is a real-time information transmission specification based on IP technology and Extensible Markup Language (XML) that support publish/subscribe messaging systems. XMPP widely used in chat, entertainment, and application that require real-time communication. [13]. Every client device that uses XMPP has a unique Jabber ID depend on XMPP domain server that used.

**Revised Manuscript Received on September 15, 2019**

**Rachmad Andri Atmoko**, School of Mechanical and Electrical Engineering, Guilin University of Electronic Technology, Guilin, China. - Faculty of Engineering, University of Jember, Jember, Indonesia.
Email: mokoraden@hotmail.com

**Daoguo Yang**, School of Mechanical and Electrical Engineering, Guilin University of Electronic Technology, Guilin, China.
Email: d.g.yang@guet.edu.cn

**Mohamad Ilyas Abas**, Information System Department, Muhammadiyah University of Gorontalo, Gorontalo, Indonesia.
Email: ilyasabas@umgo.ac.id

**Afif Zuhri Arfianto**, Marine Electrical Engineering, Politeknik Perkapalan Negeri Surabaya, Surabaya, Indonesia.
Email: afif@ppns.ac.id

**Robbi Rahim**, Sekolah Tinggi Ilmu Manajemen Sukma, Medan, Indonesia
Email: usurobbi85@zoho.com

## II. METHODOLOGY

The system architecture consists of the robot layer, cloud layer, and application layer. Figure 1 shows an overview of the system architecture.

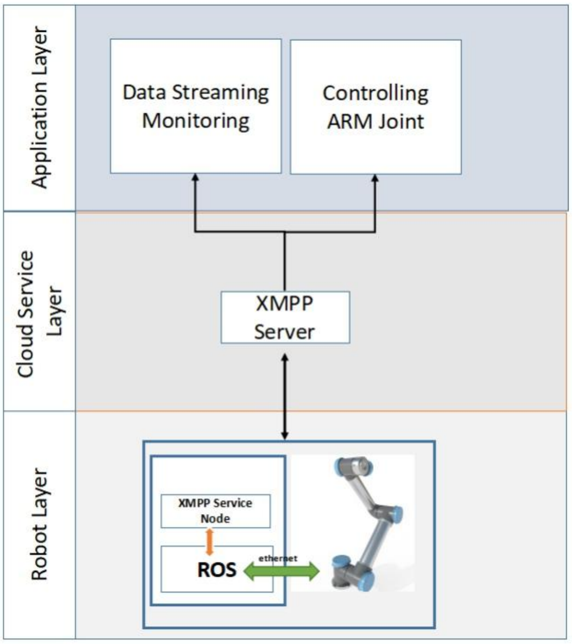


**Figure 1. System architecture**

The details of each layer and steps are explained in the following sections.

### 2.1 Robot Layer

We used a personal computer-based on Ubuntu Linux and the Robot Operating System (ROS) as a framework that can be integrated directly with industrial robots. ROS is a framework software released by Willow Garage which is used popularly by academia and industry to control or simulate industrial robots [14]. ROS has a particular library called ROS Industrial which supports several types of industrial robots used in the industrial environment. This library extends the advanced capabilities of ROS software to handling various robot control scheme in the manufacturing area [15]. Gazebo and Moveit is simulation and motion planning feature in ROS environment.

In this study, we used universal robots which one of the industrial robots used popularly by academia and industry. Figure 2 shows the structure of universal robot version 5, which consists of 6 joints, namely base joint, shoulder joint, elbow joint, wrist one joint, wrist two joint, and wrist three joints. ROS provided logically modified packages called nodes. ROS internal communication architecture used the publish/subscribe scheme. In this scheme, each node publishes or subscribes to the specific topic inside ROS. We used Rospy python library for handled communication in the ROS environment.

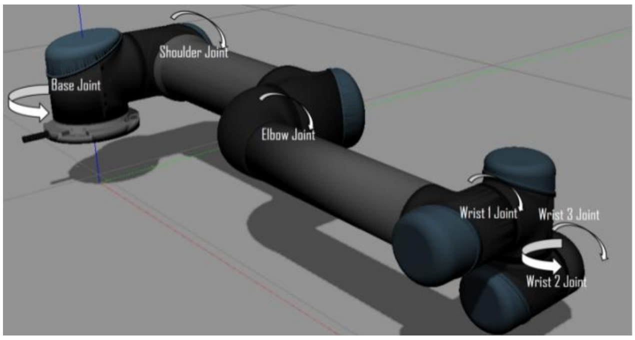


**Figure 2. UR5 structure**

XMPP also use publish/subscribe scheme for establishing a connection. The robot functions as a publisher that sends real-time robot arm movement data to the XMPP server. This robot also becomes a subscriber that receives control commands sent from the XMPP server. To start the connection, the user (publisher/subscriber) need to register an account in XMPP server side. The user consists of robot hardware (robot layer) and an interface application (application layer). Each user communicates with another explicit user that has been registered to enable secure end-to-end communication.

In this study, we used an industrial robot simulation using Gazebo, which had included in the ROS package. We made an XMPP service node for getting data position and rotation of the joint robot. Figure 3 explains the communication flow that occurs in the architecture that we have designed.

The user/client sends control commands to drive each joint robot using the XMPP protocol. The XMPP node bridges the exchange of data between the user and the robot. By default, ROS does not provide functions for communicating through the Internet network (Teleoperation). We propose adding XMPP nodes to the internal architecture of ROS so that users can do Teleoperation. The XMPP node is combined with the Moveit commander and move group. The integration of the XMPP node with the Moveit library uses the Python programming language. Move group can directly drive the robot by connecting it to the robot controller. The robot controller is a driver that is used to communicate directly with robot hardware. Besides the move group can also be linked to the Gazebo simulation environment if we want to make simulation without having to connect to real robot hardware.

The position and rotation value of the joint robot was taken by subscribing to the TF topic in the ROS architecture. TF is a ROS package that functions to track multiple coordinate frames over time. Through this package, we can monitor the movement of each joint robot. TF provides data changes in the joint of the robot. Data from the TF topic is sent to the user through the XMPP node in real-time.

### 2.2 Cloud Server Layer

On the server-side, we used a Virtual Private Server that had the following technical specifications on table 1.

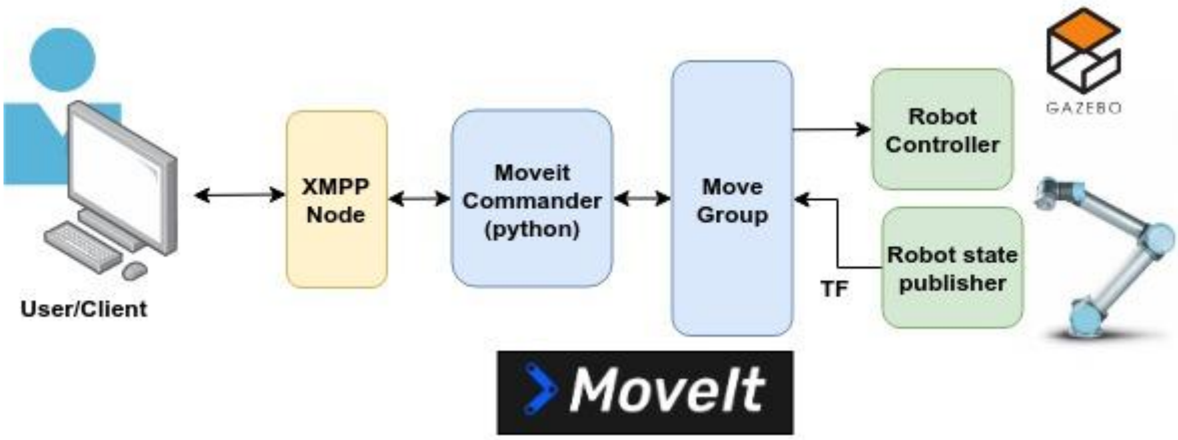


Figure 3. Communication flow of architecture

Table 1. Server Parameter

| Item | Value |
|---|---|
| CPU Core | 1 |
| Memory | 1GB |
| Disk Space | 20GB |
| Bandwidth | 1000GB |
| Server Location | Singapore |

The operating system used Ubuntu 16.04 - 64 Bit. We used an open-source XMPP server called Prosody IM. Prosody was open-source software with an MIT/X11 license and available for Windows, Linux, or OSX.

### 2.3 Application Layer

On this layer, we built a web-based user interface that displays real-time data from the robots. This data consists of the rotation and position value of each joint robot. The application was built using the HTML Javascript programming language and used the NodeJs server. This application was not only to show robot movement data but also to control the angles of each robot joint remotely using the XMPP protocol.

NodeJS was used to build web interface applications. Node-XMPP-client module used to establish connections with XMPP servers. Data sent from the robot layer to the XMPP server then forwarded to the client. Socket.io module used to display data in real-time. Figure 4 shows the part of the application layer, namely the control panel, which consists of six slider buttons. This slider button provides the parameters of the motion angle values for each joint arm so that each joint arm can be controlled online and real-time. Figure 5 shows the simulation results on the gazebo simulator. Figure 6 shows that six joint arms sent the position of each robot joins on the web-based interface application.

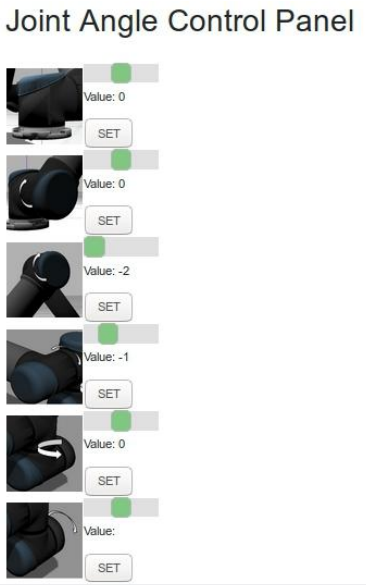


Figure 4. Slider button

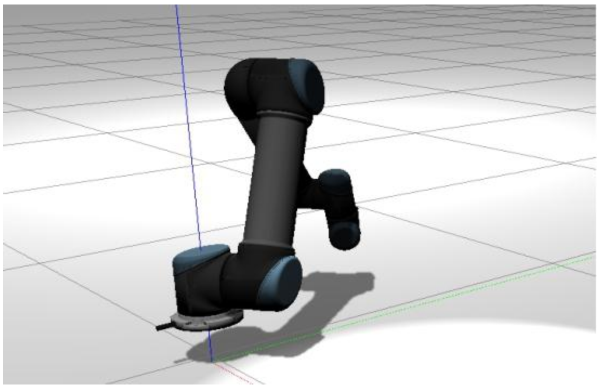

Figure 5. Robot simulation

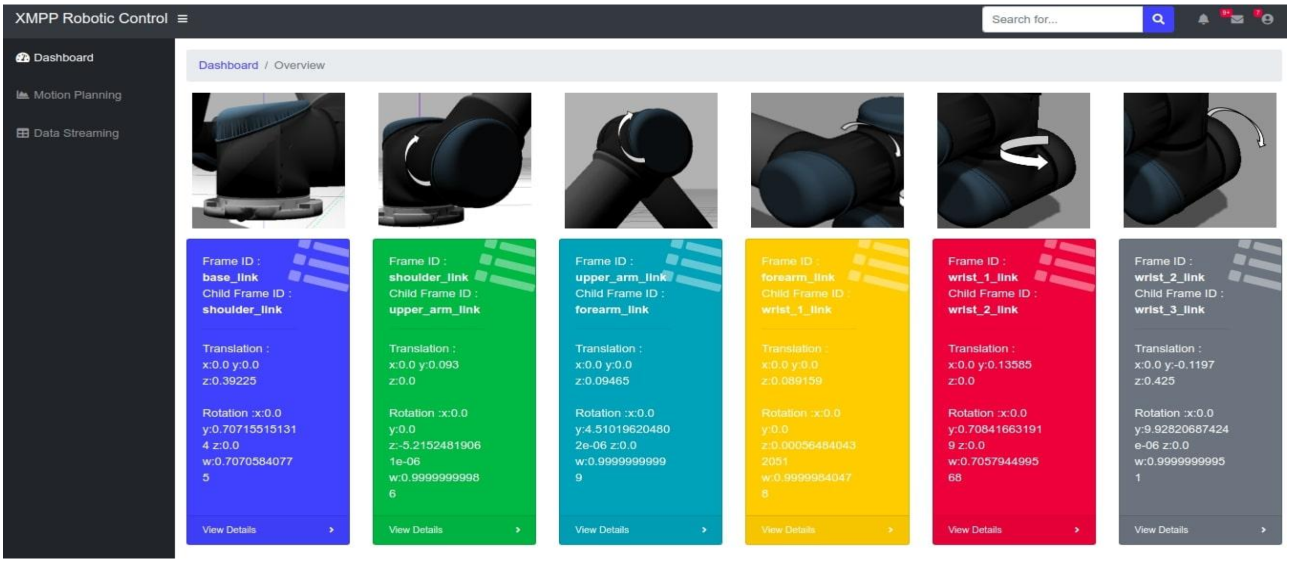


**Figure 6. Web-based interface application**

## III. ANALYSIS

We conducted a communication performance test by looking at the latency time between the time of sending and receiving data. Low latency time is required to support the control and monitor functions. Industry 4.0 concept suppose that real-time communication is a crucial factor to support cyber-physical systems. We make data transmission applications and data reception applications. The data transmission application always records transmission time. The data transmission application sends a message to the XMPP server every 3 seconds, with intervals for 60 minutes. The data reception application functions to receive data sent from the XMPP server. Incoming data is recorded, then the calculation of the difference in time between the delivery time and the time of receipt is done. We get the result in an average latency time using the XMPP protocol is 0.82 seconds. We were also comparing the performance of XMPP with the HTTP protocol. Both are application protocols that run over TCP. HTTP offers a request/response communication scheme to connect the robot layer to the server layer. Table 2 shows the results of the latency comparison between the two protocols. HTTP produces an unsatisfactory latency time compared to XMPP. HTTP generates an average latency time of 2.99 seconds. These results also show the concept of publish/subscribe gives better latency results than the request/response concept.

**Table 2. Latency result**

| Order Sampling | HTTP | XMPP |
|---|---|---|
| 1 | 2762 | 800 |
| 2 | 3244 | 855 |
| 3 | 2711 | 890 |
| 4 | 2859 | 815 |
| 5 | 3235 | 854 |
| 6 | 2817 | 761 |
| 7 | 2935 | 800 |
| 8 | 2962 | 894 |
| 9 | 2857 | 809 |
| 10 | 2999 | 802 |
| 11 | 3298 | 879 |
| 12 | 2931 | 821 |
| 13 | 3286 | 820 |
| 14 | 3115 | 782 |
| 15 | 2776 | 795 |
| 16 | 2783 | 894 |
| 17 | 3214 | 842 |
| 18 | 3032 | 808 |
| 19 | 3142 | 759 |
| 20 | 2934 | 805 |
| **Latency Average (milisecond)** | **2994.6** | **824.25** |

*Retrieval Number: C5893098319/2019©BEIESP*
*DOI:10.35940/ijrte.C5893.098319*



*Published By:*
*Blue Eyes Intelligence Engineering*
*& Sciences Publication*

## IV. DISCUSSION & CONCLUSION

Cloud industrial robot in the industry 4.0 paradigm requires reliable and real-time communication to represent the real-plant conditions. In this study, we implemented the XMPP protocol for teleoperation robots. XMPP has a mechanism that ensures clients know each other before establishing connections. This protocol provides a better security experience. Herein lies the reliability of the XMPP protocol as a real-time and reliable communication solution. The experiment result gets the average latency time of XMPP less than 1 second. XMPP has the advantage of lower time latency than the scheme offered by the HTTP protocol. This result can support the control system and monitoring industrial arm robots. In the future development needs to be done by applying other protocols to compare the results.

## V. ACKNOWLEDGMENT

This work is supported by the Guangxi Science and Technology Major Project (Grant No. AA17204018).

## REFERENCES

1. Yin S, Kaynak O. Big data for modern industry: challenges and trends [point of view]. Proceedings of the IEEE. 2015 Mar 24;103(2):143-6.
2. Baily JMM and MN. Is Manufacturing "Cool" Again? [Internet]. Brookings. 2013 [cited 2019 Jul 10]. Available from: https://www.brookings.edu/opinions/is-manufacturing-cool-again/
3. Widodo RB, Quita RM, Amrizal S, Gunawan RS, Wada C, Ardiansyah A, Sutrisno I, Rahmat MB, Arfianto AZ, Handoko CR, Santoso AD. Grasping and Attached Mode in Human-Computer Interaction in the Study of Mouse Substitution. In Journal of Physics: Conference Series 2019 Mar (Vol. 1196, No. 1, p. 012034). IOP Publishing.
4. Wan J, Tang S, Yan H, Li D, Wang S, Vasilakos AV. Cloud robotics: Current status and open issues. IEEE Access. 2016 Jun 1;4:2797-807.
5. Kehoe B, Patil S, Abbeel P, Goldberg K. A survey of research on cloud robotics and automation. IEEE Transactions on automation science and engineering. 2015 Jan 12;12(2):398-409.
6. Qin X, Jiang B, Deng X, Zu X, Du Y, Li H. A robot remote control system based on VPN and TCP/IP protocol. In 2008 IEEE International Conference on Mechatronics and Automation 2008 Aug 5 (pp. 285-289). IEEE.
7. Cho DK, Chun SH, Ahn JG, Kwon YS, Eom JH, Kim YI. A UDP-based protocol for mobile robot control over wireless Internet. In 2009 Second International Conference on Robot Communication and Coordination 2009 Mar 31 (pp. 1-4). IEEE.
8. Tian Y, Matsuno T, Ma S. Development of remote robot control system for snake-like robot based on SSH protocol and iOS system. In 2016 IEEE International Conference on Robotics and Biomimetics (ROBIO) 2016 Dec 3 (pp. 100-105). IEEE.
9. Atmoko RA, Riantini R, Hasin MK. IoT real time data acquisition using MQTT protocol. In Journal of Physics: Conference Series 2017 May (Vol. 853, No. 1, p. 012003). IOP Publishing.
10. Mishra B. TMCAS: An MQTT based Collision Avoidance System for Railway networks. In 2018 18th International Conference on Computational Science and Applications (ICCSA) 2018 Jul 2 (pp. 1-6). IEEE.
11. Dhar P, Gupta P. Intelligent parking Cloud services based on IoT using MQTT protocol. In 2016 International conference on automatic control and dynamic optimization techniques (ICACDOT) 2016 Sep 9 (pp. 30-34). IEEE.
12. Kirsche M, Klauck R. Unify to bridge gaps: Bringing XMPP into the Internet of Things. In 2012 IEEE International Conference on Pervasive Computing and Communications Workshops 2012 Mar 19 (pp. 455-458). IEEE.
13. Wang H, Xiong D, Wang P, Liu Y. A lightweight XMPP publish/subscribe scheme for resource-constrained IoT devices. IEEE Access. 2017 Aug 21;5:16393-405.
14. Quigley M, Conley K, Gerkey B, Faust J, Foote T, Leibs J, Wheeler R, Ng AY. ROS: an open-source Robot Operating System. In ICRA workshop on open source software 2009 May 12 (Vol. 3, No. 3.2, p. 5).
15. Mirkhanzadeh B, Shao C, Shakeri A, Sato T, Razo-Razo M, Tacca M, Fumagalli A, Yamanaka N. A two-layer network Orchestrator offering trustworthy connectivity to a ROS-industrial application. In 2017 19th International Conference on Transparent Optical Networks (ICTON) 2017 Jul 2 (pp. 1-4). IEEE.

## AUTHORS PROFILE

**Rachmad Andri Atmoko,** received the (Bachelor of Applied Science) BAS degree in automation engineering from the Shipbuilding Institute of Polytechnic Surabaya, Indonesia, in 2013, and M.Eng Degree in Instrumentation Engineering from Sepuluh Nopember Institute of Technology (ITS), Indonesia, in 2015. He is currently pursuing the Ph.D degree in Guilin University of Electronic Technology, Guilin, China. His current research interest includes cloud robotics, cloud manufacturing, and internet of things.

**Daoguo Yang,** received the M.S. degree from Zhejiang University, China, in 1989, and the Ph.D. degree from the Delft University of Technology (TU Delft), The Netherlands, in 2007. After his doctoral research, he was a Postdoctoral Research Fellow with TU Delft for two years. He then joined Philips Semiconductors (NXP Semiconductors, The Netherlands), as a Principal Engineer and the Project Leader. Since 2009, he has been a Professor with the College of Mechanical and Electrical Engineering, Guilin University of Electronic Technology, China. His current research interests include microelectronics and microsystems packaging technologies, LED packaging and system integration, reliability analysis, and numerical and experimental mechanics.

**Mohamad Ilyas Abas,** is a lecturer at the Muhammadiyah University of Gorontalo in the Information Systems study program. He is currently active as a renewable energy researcher by collaborating with USAID SHERA (America and several universities in Indonesia). Is a researcher with a concentration in data mining, Geographic Information System (GIS) and Software Engineering. Also active in social activities by conducting several services including computer application training for village officials, introduction of IT-based systems to villages, empowerment based on local potential in several villages.

**Afif Zuhri Arfianto**, received his Bachelor and Master with "Beasiswa Unggulan" scholarship at multimedia telecommunication study programme, Electrical Engineering Department in ITS (Institut Teknologi Sepuluh Nopember). He has more than 4 years of teaching experience, having previously worked as member of publication center in Politeknik Perkapalan Negeri Surabaya, Indonesia. His main research work focuses on Computer Network, internet of things and Telecommunication Network. Currently, he is working in the area LoRA for internet of things, IoT Access Architecture for Vehicle, and deployment of an IoT network infrastructure, as a localised regional service.

**Robbi Rahim**, received his Master in Protocol Cryptography and soon his PhD degree also focus Protocol Cryptography. He has more than 5 years of teaching experience, having previously worked as head of publication at STIM Sukma Medan, Indonesia. His main research work focuses on Computer Security, Genetic Algorithm, Steganography, Cryptography and Decision Support System. Currently, he is working in the area of Cryptography and Decision Support System, with a specialization in the model of communication in Protocol Cryptography