SCADA Dynamic Application Design Equipped with Multi Connections

Agus Khumaidi[1], M. Basuki R.[1], Hananta Adhi R.[1], R. Y. Adhitya[1], Jossianto Eko P.[1], Arie Indartono[2] and P. N. A. Nugroho[2]

[1]*Department of Marine Elctrical Engineering, Politeknik Perkapalan Negeri Surabaya, Surabaya, Indonesia*

[2]*Department of Marine Engineering, Politeknik Perkapalan Negeri Surabaya, Surabaya, Indonesia*

Keywords: Dynamic Applications, Modbus, IoT, GUI, MQTT, SCADA.

Abstract: Advances in information technology have been widely implemented in the field of industry 4.0. In its development, the role of the automation system becomes very important. The use of automation and data monitoring in the industry aims to help and improve the effectiveness of human work. The need for software that is easy, and can be used on many communication protocols is very high due to the need for supervision, control systems, and data acquisition in the industry. The output of this research is producing Graphical User Interface (GUI) of SCADA software which is equipped with various features, and we called it SCATO (SCADA Teknik Otomasi). In addition, a GUI is needed to display data and configure a multi-connection network between hardware and software to make it easier for users to monitor, control systems, and data acquisition in the industry. The use of Modbus TCP/IP, Modbus RTU, and MQTT communication protocol as the input protocol have an average accuracy rate of 100%. MQTT prioritizes data efficiency and it has a small size that is able to support IoT (Internet of Things) systems and its appropriate to be implemented in industrial plans for remote monitoring. Database purposes are using MySQL, Firebase, and CSV files as input with an average data reading speed of 0.51 seconds. In addition, there is a Graphical User Interface for designing HMI which can be used as an interface to support systems in industry or as modules or teaching materials, especially in the field of Automation Engineering.

1 INTRODUCTION

Advances in information technology have been widely implemented in the industrial world. The use of automation control systems in industry aims to help and improve the effectiveness of human work. Today we are in the middle of the industry 4.0 period where the number 4.0 indicates the Industrial Revolution which has entered the 4th wave. The characteristics of the model from Industry 4.0 are a combination of several of the latest technological developments which include information and communication technology, communication networks, big data, and cloud computing as well as equipment developed to facilitate human interaction with computers (Karacay, 2018). All aspects of industrial computing can communicate with each other by utilizing communication protocols, one of which is Modbus. Communication Protocol is a language to communicate with each other between hardware, and software. Modbus is a two-serial communication protocol method, where communication can occur with the Modbus master and Modbus slave systems (Wenzhu You, 2019). Modbus can be transmitted via serial communication (Modbus RTU), via Ethernet communication (Modbus TCP/IP), via ASCII (Modbus ASCII), and Modbus Plus. Modbus RTU is also equipped with a cyclic redundancy error (CRC) mechanism that functions to reduce errors and ensure data reliability. In practice, Modbus supports many modems and gateways to work (Chandra, 2016).

2 METHODOLOGY

2.1 Research Flow

The initial system design was made in the form of a flow chart as shown in Figure 1. Based on the

Khumaidi, A., R., M., R., H., Adhitya, R., P., J., Indartono, A. and Nugroho, P.
SCADA Dynamic Application Design Equipped with Multi Connections.
DOI: 10.5220/0010958500003260
In *Proceedings of the 4th International Conference on Applied Science and Technology on Engineering Science (iCAST-ES 2021)*, pages 1039-1044
ISBN: 978-989-758-615-6; ISSN: 2975-8246

flowchart diagram, the design of the system is a guide in the design of the system framework and software. To ensure that the entire system that has been designed runs as expected, system testing is carried out. After the system testing is successful, the next stage is making a report book.

2.2 Identification of Problems

In this system, the problem raised is an effort to in order to be able to design dynamic applications with multiple connections as well as a graphical user interface to create Human Machine Interface (HMI). Based on Graphical User Interface (GUI) with efforts to apply Modbus TCP/IP protocols, Modbus RTU and MQTT for module data communication with applications and integrating MySQL, Firebase, and CSV databases for data transactions from server to application.

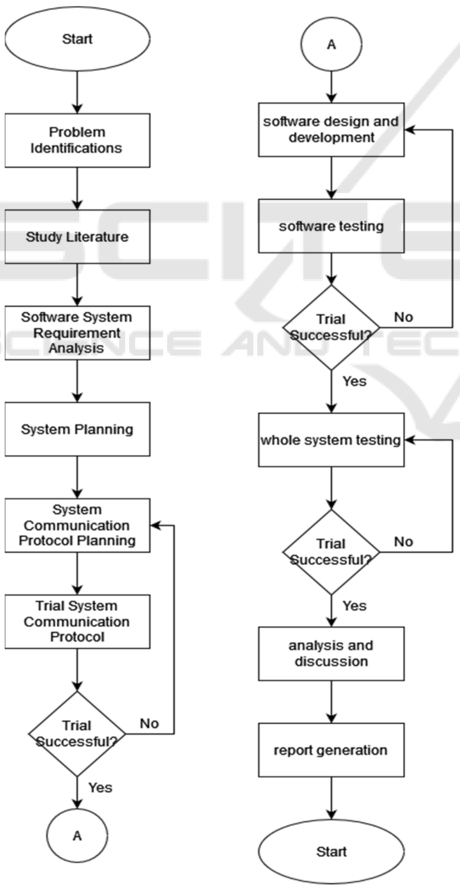


Figure 1: Research Flowchart.

2.3 Study of Literature

At this stage, the authors seek as much information as possible about the concepts that will be used in the study. The search was carried out related to information about the Modbus Communication Protocol (Kelong Wang, 2014), MQTT (Atmoko & Yang, 2018), and MySQL as a Database (Gorskis, 2018) (Fahmi, 2013) (Subiyanto, 2015).

2.4 System Planning

The dynamic application system is running on the ASP .NET MVC framework. Supported by Modbus, MQTT connectivity, and several DBMS for data transactions, including MySQL, Firebase, and CSV. presentation of data on objects such as gauges or charts is done by linking variable names on the tag manager. The system flowchart in this study is shown in Figure 2.

2.5 Object Oriented Programming

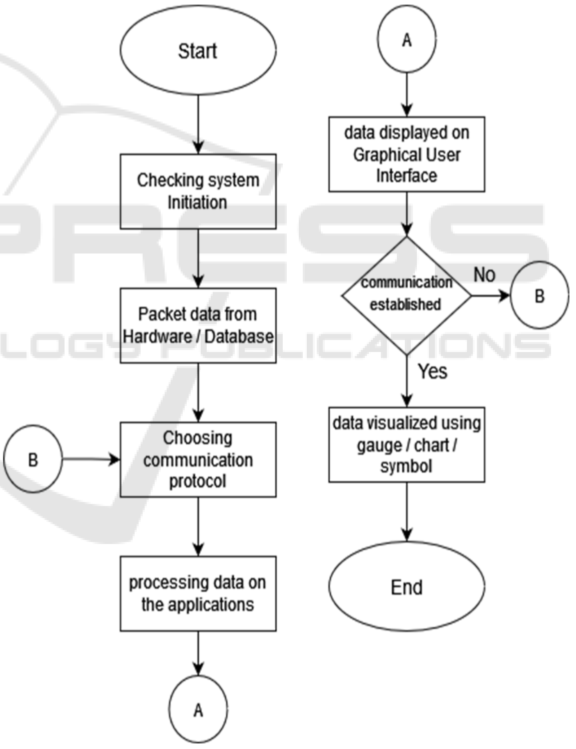

Start
Checking system Initiation
Packet data from Hardware / Database
B
Choosing communication protocol
processing data on the applications
A
A
data displayed on Graphical User Interface
communication established
No
B
Yes
data visualized using gauge / chart / symbol
End

Figure 2: System Flowchart.

In these dynamic applications, the system uses the Object-Oriented Programming as the basis for the preparation of the program structure. The steps taken to detect objects are as shown in Figure 3.

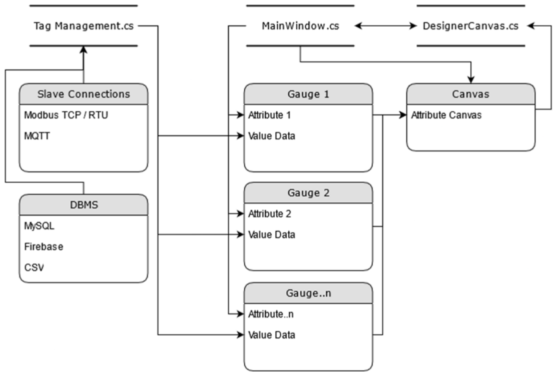


Figure 3: Object Oriented Programming.

2.6 Software Design

The SCATO application has three workspaces, including:

1. The Tag Management tab.
2. Graphic Designer thickness and design applied
3. Configurations

where each workspace tab has a different function as shown in Figures 4, 5, and 6.

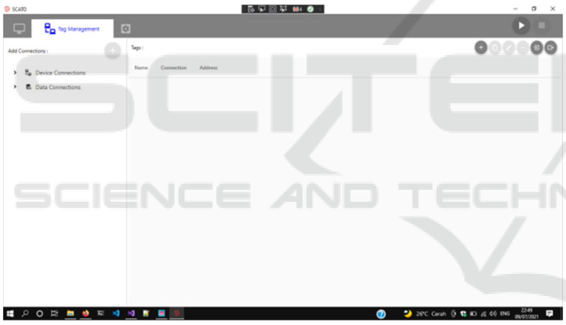


Figure 4: Tag Management Of SCATO.

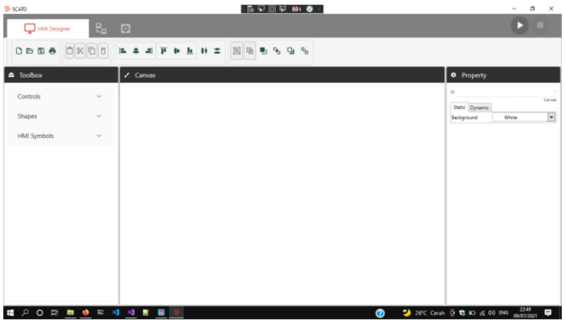
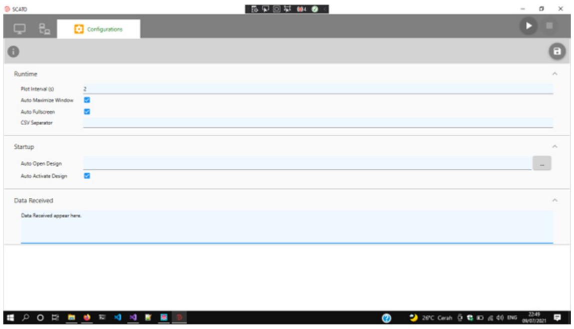


Figure 5: Worksheet Of SCATO.



Runtime

Plot Interval (s) 2

Auto Maximize Window ☑

Auto Fullscreen ☑

CSV Separator

Startup

Auto Open Design

Auto Activate Design ☑

Data Received

Data Received appear here.

Figure 6: Configuration Tab Of SCATO.

3 RESULT

3.1 Modbus Protocol Communications

SCATO dynamic applications use several Modbus communications protocols, including Modbus TCP/IP and Modbus RTU interface. In order to communicate with each other, data is organized using a registry number for each Modbus communication. Every device or device that is connected to each other using Modbus requires Modbus access as a communication protocol that corresponds to each registry number on Modbus access. on Modbus TCP, the parameters used are IP Address, and also on another side, the parameters of Modbus RTU that have been used are serial port and baud rate. The successful connection that has been made using Modbus protocol communications can be seen in Figure 7.

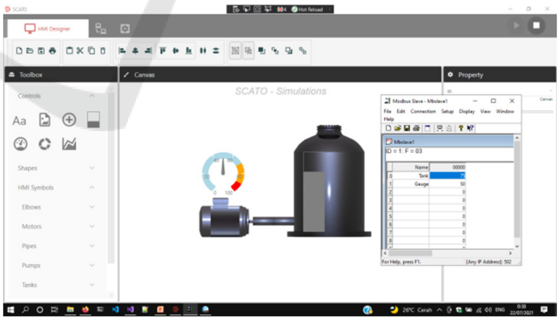
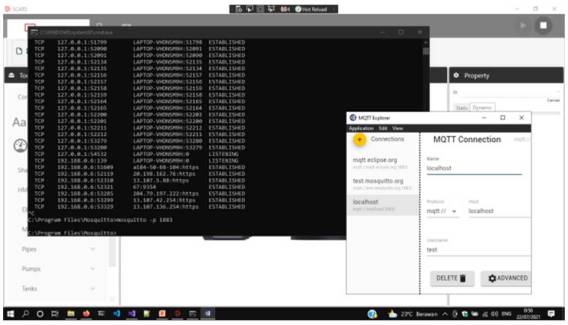


Figure 7: Modbus Protocol.

3.2 MQTT Protocol Communications

The MQTT work system uses the Publish and Subscribe data method. In its application, the SCATO software will connect to a broker and have a certain topic, which in this case study the broker used is ’mosquitto’ with local configurations and the help of the MQTT Explorer application as an interface. Figure 8 shows MQTT Explorer.



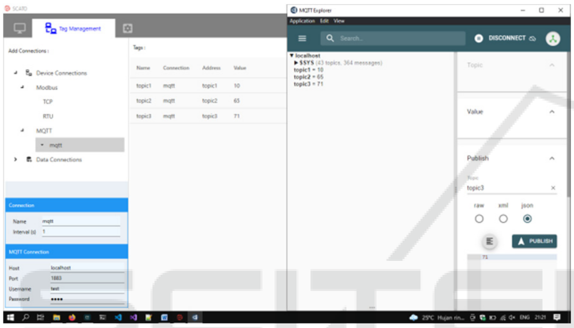
```
TCP    127.0.0.1:51799     LAPTOP-VHDNSM9H:51798  ESTABLISHED
TCP    127.0.0.1:52090     LAPTOP-VHDNSM9H:52091  ESTABLISHED
TCP    127.0.0.1:52091     LAPTOP-VHDNSM9H:52090  ESTABLISHED
TCP    127.0.0.1:52134     LAPTOP-VHDNSM9H:52135  ESTABLISHED
TCP    127.0.0.1:52135     LAPTOP-VHDNSM9H:52134  ESTABLISHED
TCP    127.0.0.1:52156     LAPTOP-VHDNSM9H:52157  ESTABLISHED
TCP    127.0.0.1:52157     LAPTOP-VHDNSM9H:52156  ESTABLISHED
TCP    127.0.0.1:52158     LAPTOP-VHDNSM9H:52159  ESTABLISHED
TCP    127.0.0.1:52159     LAPTOP-VHDNSM9H:52158  ESTABLISHED
TCP    127.0.0.1:52164     LAPTOP-VHDNSM9H:52165  ESTABLISHED
TCP    127.0.0.1:52165     LAPTOP-VHDNSM9H:52164  ESTABLISHED
TCP    127.0.0.1:52200     LAPTOP-VHDNSM9H:52201  ESTABLISHED
TCP    127.0.0.1:52201     LAPTOP-VHDNSM9H:52200  ESTABLISHED
TCP    127.0.0.1:52211     LAPTOP-VHDNSM9H:52212  ESTABLISHED
TCP    127.0.0.1:52212     LAPTOP-VHDNSM9H:52211  ESTABLISHED
TCP    127.0.0.1:53279     LAPTOP-VHDNSM9H:53280  ESTABLISHED
TCP    127.0.0.1:53280     LAPTOP-VHDNSM9H:53279  ESTABLISHED
TCP    127.0.0.1:54532     LAPTOP-VHDNSM9H:0      LISTENING
TCP    192.168.0.6:139     LAPTOP-VHDNSM9H:0      LISTENING
TCP    192.168.0.6:51609   a184-50-68-104:https   ESTABLISHED
TCP    192.168.0.6:52119   20.198.162.76:https    ESTABLISHED
TCP    192.168.0.6:52310   13.107.5.88:https      ESTABLISHED
TCP    192.168.0.6:52321   67:9354                ESTABLISHED
TCP    192.168.0.6:53285   204.79.197.222:https   ESTABLISHED
TCP    192.168.0.6:53299   13.107.42.254:https    ESTABLISHED
TCP    192.168.0.6:53329   13.107.136.254:https   ESTABLISHED
^C
C:\Program Files\Mosquitto>mosquitto -p 1883

C:\Program Files\Mosquitto>
```

Name: localhost
Protocol: mqtt://
Host: localhost
Username: test

Figure 8: Mosquitto and MQTT Explorer.

After setting up the configurations of MQTT, the data received will be automatically separated and described based on the subscribed topic. How to set up the configuration can be shown in figure 9.



Tags :

Name	Connection	Address	Value
topic1	mqtt	topic1	10
topic2	mqtt	topic2	65
topic3	mqtt	topic3	71

Connection
Name: mqtt
Interval (s): 1

MQTT Connection
Host: localhost
Port: 1883
Username: test
Password: ••••

```
▼ localhost
  ▶ $SYS (43 topics, 364 messages)
  topic1 = 10
  topic2 = 65
  topic3 = 71
```

Topic: topic3
71

Figure 9: MQTT Protocol.

3.3 Database Management System

Some of the databases supported by SCATO include MySQL, Firebase, and CSV files with int, float, double, and Boolean data types. The data will be received and parsed based on each column in the database table automatically. Communication with the database can be made with the help of a connector in the form of a library framework such as SQL Connections and Fire sharp which are available in the NuGet Library Package. The explanation of Database Connection Protocol is shown in Figures 10,11 and 12 sequentially.

3.4 Connection Test Established of SCATO and Controller

In the following communication test, SCATO uses variables to test the connection between the hardware and the SCATO application. In this Communication Test to prove the success of the connection that has been made where the results can be seen in Tables 1, 2, and 3 sequentially.

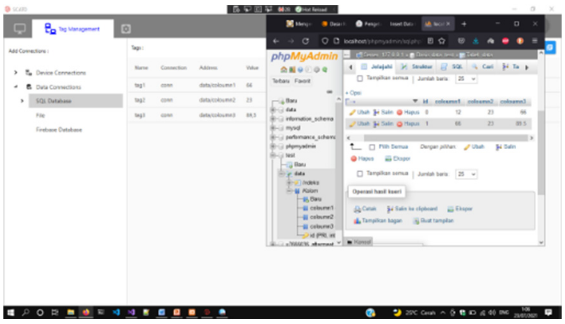
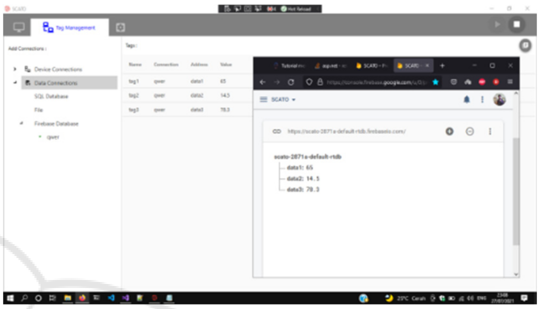


Figure 10: Database Connection Between SCATO and phpmysql.



Tags :

Name	Connection	Address	Value
tag1	qwer	data1	65
tag2	qwer	data2	14.5
tag3	qwer	data3	78.3

https://scato-2871a-default-rtdb.firebaseio.com/

```
scato-2871a-default-rtdb
  ├ data1: 65
  ├ data2: 14.5
  └ data3: 78.3
```

Figure 11: Database Connection Between SCATO and Firebase.

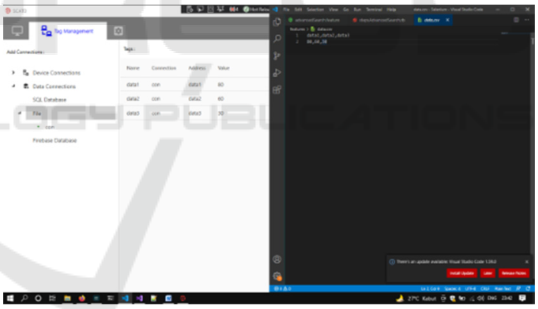


Figure 12: Database Connection Between SCATO and CSV.

Table 1: Modbus TCP Communication Test.

Variable	Input Data	SCATO READING
Data1	28	28
Data2	60	60
Data3	0	0
Data1	28	28
Data2	60	60
Data3	0	0
Data1	29	29
Data2	60	60
Data3	1	1
Data1	28	28
Data2	59	59
Data3	1	1

Table 2: MQTT Communication Test.

Variable	Input Data	SCATO READING
Topic1	28	28
Topic2	60	60
Topic3	0	0
Topic1	28	28
Topic2	60	60
Topic3	0	0
Topic1	29	29
Topic2	59	59
Topic3	0	0
Topic1	27	27
Topic2	59	59
Topic3	0	0
Topic1	29	29
Topic2	60	60
Topic3	0	0

Table 3: Mysql Connection Test.

Variable	Input Data	SCATO READING
Tag1	67	67
Tag2	22	22
Tag3	96,5	96,5
Tag1	12	12
Tag2	76	76
Tag3	33,8	33,8
Tag1	60	60
Tag2	12	12
Tag3	32,1	32,1
Tag1	88	88
Tag2	87	87
Tag3	66,5	66,5
Tag1	24	24
Tag2	55	55
Tag3	11,8	11,8

3.5 Connection Test Established of SCATO and Database

The speed of data reading is very influential on application performance where monitoring applications must display data in real-time. the smaller the latency or delay, the better for the application to display data automatically. The results of the reading test can be seen in Tables 4,5, 6 and 7.

Table 4: Modbus TCP/RTU Read Speed Data.

TCP Read	
Variable	Data Reading Speed(s)
Data1	0,52
Data2	0,61
Data3	0,51
RTU Read	
Data1	0,61
Data2	0,58
Data3	0,64

Table 5: MQTT Read Speed Data.

Variable	Data Reading Speed(s)
Topic1	0,56
Topic2	0,64
Topic3	0,47

Table 6: Mysql Read Speed Data.

Variable	Data Reading Speed(s)
Tag1	0,48
Tag2	0,48
Tag3	0,56

Table 7: Firebase/CSV Read Speed Data.

Firebase Read	
Variable	Data Reading Speed(s)
Data1	0,56
Data2	0,56
Data3	0,40
CSV Read	
Data1	0,42
Data2	0,56
Data3	0,62

4 CONCLUSIONS

Based on testing and system analysis that has been done, the result of this research can be concluded that

1. Communication between Modbus TCP-IP and Modbus RTU with the SCATO application runs with a data reading accuracy rate of 100.
2. Communication between MQTT and the SCATO application runs with a data reading accuracy rate of 100.
3. Communication between MySQL, Firebase, and CSV databases can run with a data reading accuracy rate of 100

REFERENCES

Atmoko, R. A. & Yang, D., 2018. *Online Monitoring & Controlling Industrial Arm Robot Using MQTT Protocol.* Bandung, Indonesia, 2018 IEEE International Conference on Robotics, Biomimetics, and Intelligent Computational Systems (Robionetics).

Chandra, S. D., 2016. *Desain Dan Implementasi Protokol Modbus Untuk Sistem Antrian Terintegrasi Pada Pelayanan Surat Izin Mengemudi (SIM) Di Kepolisian Resort.* Undergraduate. Surabaya: Institut Teknologi Sepuluh Nopember.

Fahmi, M., 2013. *ntegrasi Sistem SCADA Web – DCS Berbasis Networking PLC Dilengkapi Sistem Auto Recovery Dan Auto Redundant Dengan Menggunakan Multiple Communication Devices.* Surabaya, Proceeding Seminar Nasional Pendidikan Teknik Elektro.

Gorskis, H., 2018. *SQL query construction from database concepts.* Riga, Latvia, 2018 59th International Scientific Conference on Information Technology and Management Science of Riga Technical University (ITMS).

Karacay, G., 2018. *Talent Development for Industry 4.0.* s.l.:s.n.

Kelong Wang, D. P. L. S., 2014. *Implementation of Modbus Communication Protocol based on ARM Coretx-M0.* Shanghai, China, 2014 IEEE International Conference on System Science and Engineering (ICSSE).

Subiyanto, L., 2015. *MODUL AJAR KODE: A PRAKTIKUM PROGRAMMABLE LOGIC CONTROLLER.* Surabaya: Politeknik Perkapalan Negeri Surabaya.

Wenzhu You, H. G., 2019. *Design and Implementation of Modbus Protocol for Intelligent Building Security.* Xi'an, China, 2019 IEEE 19th International Conference on Communication Technology (ICCT).