

Real Time Monitoring and Irrigation Control Using the WebSocket Protocol

1st Sunardi¹, 2nd Aulia Naufal Afif², 3rd Fiftin Noviyanto³
sunardi@mti.uad.ac.id, naufalapip162@gmail.com, fiftin.noviyanto@tif.uad.ac.id

Informatic Engineering Ahmad Dahlan University, Indonesia¹²³

Abstrac. The development of the Internet of Things (IOT) device is now increasingly mushrooming throughout the region, especially the region which is currently incessantly applying the concept of smart city. The development of IOT (Internet of things) architecture is growing very rapidly with the number of IoT devices currently. This is directly proportional to the increasing number of data transmission protocols that can be used for the IOT device communication lines. The problem that arises is that many IoT devices require a real time connection to support the method of sending and receiving data. In this study researchers chose the use of web protocols to solve the problem, because a web protocol that is universal can be used in various devices. The web protocol used is websocket that can be used to support communication in real time. The implementation of the protocol websocket as a web protocol on Arduino devices that is supported by an Ethernet shield as a communication path to the internet with case studies monitoring temperature, soil humidity and controlling the irrigation sluice.

Keywords: Realtime, WebSocket

1 Introduction

The development of the Internet of Things (IOT) device is now increasingly mushrooming throughout the region, especially those that are currently incessantly applying the smart city concept. Internet of things is a global infrastructure for the information society, enabling sophisticated services, connecting physical and virtual objects based on current information exchange technology and the development and communication technology[1]. The IoT device itself is a concept that aims to expand the benefits of internet connectivity that is connected continuously. As for its capabilities such as data sharing, remote control, data monitoring and so on. To be able to do capabilities like on an IoT device, it certainly requires an internet access connectivity to connect from the device to other devices. This is also closely related to the data transmission protocol that will be used to transmit data. The growing number of IoT technologies has led to the creation of various application layer protocols that can be used on IoT. The number of protocols causes different applications used on IoT, this is called the "one device one app"[2]. To solve this problem, a web of things (WOT) concept was conceived. This concept uses a web protocol that is universal so that various devices can be controlled or monitored with just one website application. On IoT devices, there are several devices that require real time connection to run optimally as a data transmission communication medium. One of the web protocols that can support connections in real time is websocket. Web socket is different from http protocol where HTTP protocol requires request

and response every time the communication process occurs. Websocket supports full duplex connections where communication between the server and client requires only one request process and response only[3]. In communicating data transmission over the internet, communication implements an internet connection using Ethernet Shield installed on an Arduino device. Ethernet Shield is chosen because it belongs to a large and stable data transmission power. Based on the background that the author has explained, a conclusion emerged about the system, namely the application of the IoT concept connected to the website application. Where the website uses the websocket protocol to support data communication in real time.

2 Related Research

2.1 Websocket

Websocket is one protocol that runs on the application layer or on layer 7 in the layer layer. websocket supports full duplex communication where communication occurs in two directions between the client and the server is different from the protocol that we know today, HTTP. HTTP is a protocol that runs on port 80 and works with the concept of request and response or what is commonly called half duplex. By using a single TCP connection mechanism, websocket can build one connection path between server and client. Where websocket uses the handshaking method to make an open connection between server and client. The first process is that the client sends the handshake to the server then the server replies to the handshake and creates a single TCP connection between the server and the client. This single TCP connection supports bidirectional communication between server and client until the connection is disconnected by the client or server [4].

2.2 Javascript

Javascript is one of the programming languages used for the world of networking where the programming language is classified as reliable and fast in processing commands due to non-blocking properties. Javascript is also easy in its language because the language used is very similar to English. Javascript is widely used for client side programming and is also used as a widgets program, but javascript development is very fast even now javascript can be used for server side programming where the programming runs on the server side and javascript is also used for network programming like the Node.js javascript framework[5].

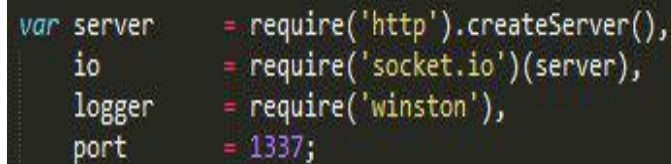
2.3 Node.js

Node.js is a framework of javascript that is widely used to do server programming or network programming. Node.js is built on Chrome's V8 Javascript Engine. Chrome's V8 Javascript Engine is an open source high performance javascript engine written in C ++. Node.js can be used as a programming language framework that organizes events from user input and node.js can be very fast and light in managing I / O models because node.js uses javascript as the programming language so that it is framework. One of the advantages of node.js is the completion of the deadlock process where every process that is done without a thread will occur deadlock but threading programming is relatively inefficient and difficult to

implement. Because node.js is non-blocking, there is no deadlock process. Then the user can easily develop a system with a large scale within the node. Js. (www.nodejs.org/en/about)[6].

2.4 Socket.IO

Socket.io is one of the libraries used in node.js. Socket.io is used to create a system that is real time. Socket.io uses websocket as the application layer mechanism.



```
var server    = require('http').createServer(),
    io        = require('socket.io')(server),
    logger    = require('winston'),
    port      = 1337;
```

Gambar 1. Socket.IO listen port 1337

2.5 Arduino Uno + Ethernet Shield

Arduino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. "Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform; for an extensive list of current, past or outdated boards see the Arduino index of boards. The Arduino Ethernet Shield 2 allows an Arduino board to connect to the internet using the Ethernet library and to read and write an SD card using the SD library[4]



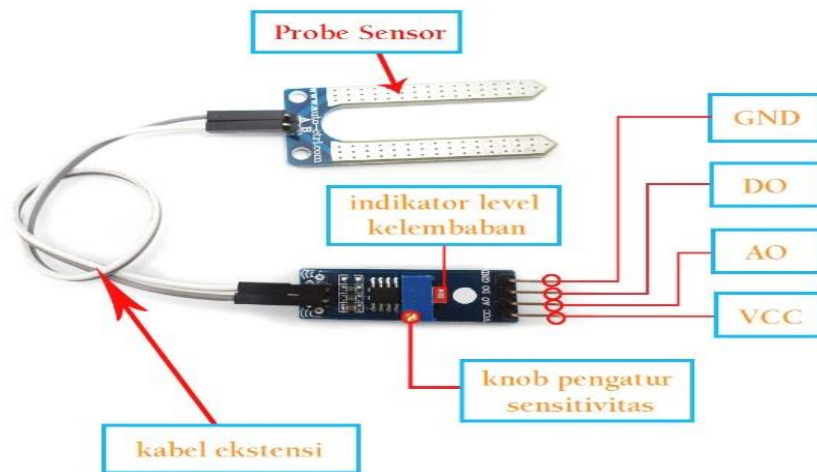
Gambar 2. Arduino with Ethernet Shield

2.6 Sensor Soil Moisture YL-69

The soil moisture sensor of the YL-69 is a humidity sensor that detects soil moisture. A set of humidity sensors type YL-69 consists of YL-69 as a sensor probe and YL-39 as a signal conditioning module. Moisture sensor type YL-69 there is a module which includes IC LM393 which serves to offset offset lace which is lower than 5mV which is stable and precise[7]..

YL-69 moisture sensor can read water content which has 3 conditions, namely:

- 0-300 : dry soil or free air
- 300-700 : moist soil
- 700-950 : wet soil (in water)



Gambar 3. Sensor YL-69

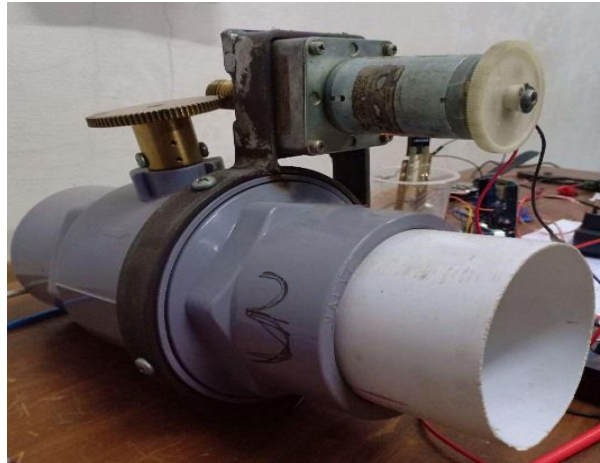
2.7 Sensor Ultrasonic

Ultrasonic sensor is a sensor that works based on the principle of sound wave reflection and is used to detect the presence of an object or certain object in front of the working frequency in the area above the sound wave from 20 kHz to 2 MHz. The ultrasonic sensor consists of two units, namely the transmitter unit and the receiver structure of the transmitter and receiver units. It is very simple that a piezoelectric crystal is connected to the mechanical anchor and is only connected with an alternating voltage vibrating diaphragm that has a working frequency of 20 kHz to 2 MHz. The atomic structure of piezoelectric crystals causes contracting to expand or shrink, a given voltage polarity and this is called the piezoelectric effect on ultrasonic sensors[8].



Gambar 4. Sensor ultrasonic

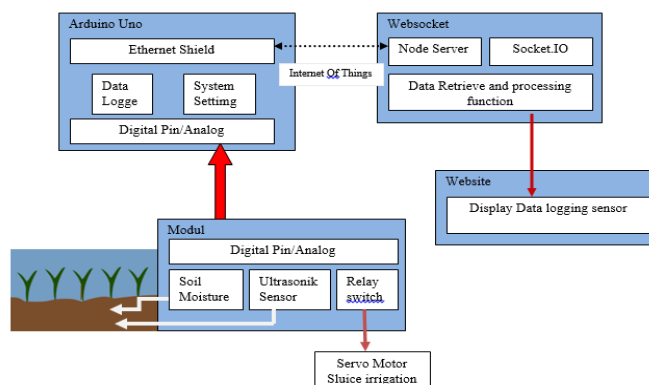
2.8 Motor Penggerak Valve



Gambar 5. Motor Penggerak Valve

3 Research method

The purpose of this study is to create a website-based monitoring and control system. Where the website uses the websocket protocol to determine soil moisture conditions including soil moisture and water level in the field, and can be used to control the opening and closing of the irigas water gate by using the websocket protocol.



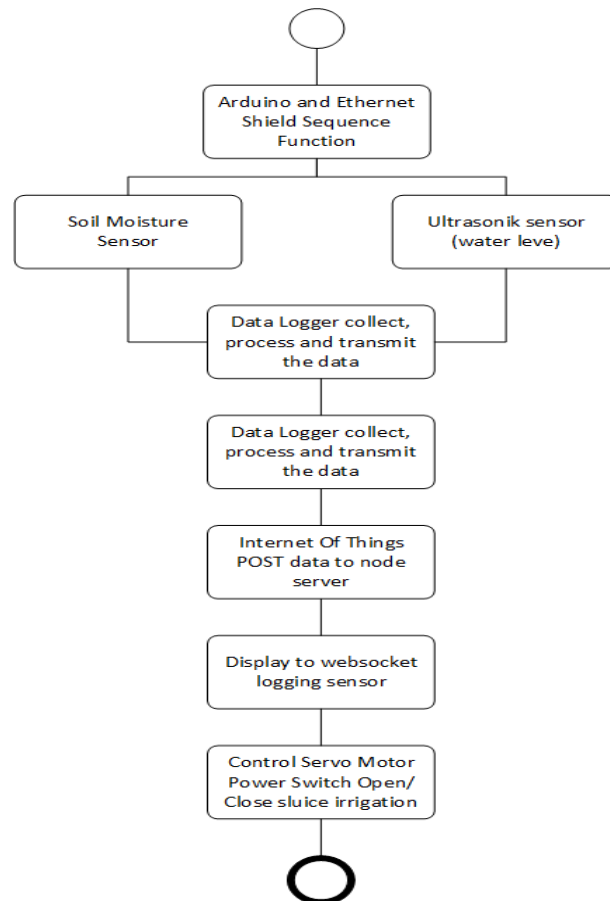
3.1 Feature

Soil moisture sensor measures soil moisture level
 Ultrasonic sensors measure water level / water level in rice fields
 Powering switches are used to adjust open and close irrigation gates

3.2 Functioning Software and Hardware

The above system features can be achieved by using the application of Internet of Things and developing hardware and website software systems with the protocol websocket. Below is a brief description of hardware and software systems used to build monitoring tools and control of rice field irrigation.

- 1) Arduino as a microcontroller board
- 2) Ethernet Shield as communication via the internet
- 3) YL-69 as a soil moisture sensor (soil moisture)
- 4) Ultrasonic sensor as a water level detector in the rice field
- 5) Relay as a power switch for servo motors to open and close the irrigation door
- 6) Nodejs as a websocket proxy server webserver
- 7) Socket.IO as a support framework for websocket



Gambar6. Functional flowchart

4 Result and analysis

4.1 Implement Server.js API interfacing script

Implementasi coding pada server.js dengan menggunakan socket.IO yang kemudian dijalankan pada server Ubuntu google compute engine

```
var server = require('http').createServer(),
    io = require('socket.io')(server),
    logger = require('winston'),
    port = 1337;

logger.remove(logger.transports.Console);
logger.add(logger.transports.Console, { colorize: true, timestamp: true });
logger.info('SocketIO > listening on port ' + port);

io.on('connection', function (socket) {
    var nb = 0;

    logger.info('SocketIO > Connected socket ' + socket.id);

    socket.on('broadcast', function (message) {
        ++nb;
        logger.info('ElephantIO broadcast > ' + JSON.stringify(message));

        io.sockets.emit("broadcast", message);
    });

    socket.on('disconnect', function () {
        logger.info('SocketIO : Received ' + nb + ' messages');
        logger.info('SocketIO > Disconnected socket ' + socket.id);
    });
});

server.listen(port);
```

Gambar 7. Script server.js

Endpoint script for the Arduino Ethernet Shield post method

```
<?php
require ('./vendor/autoload.php');
use ElephantIO\Client;
use ElephantIO\Engine\SocketIO\Version2X;

$sensor_data= array("temp"=>$_POST["temp1"],"ultrasonik"=>$_POST["ultra1"],"soil"=>$_POST["soil_moisture"]);
var_dump($sensor_data);

$client = new Client(new Version2X('35.240.163.237:1337'));

$client->initialize();
// send message to connected clients
$client->emit("broadcast",$sensor_data);
// $client->emit('broadcast', ['type' => 'notification', 'text' => $data2]);
$client->close();
```

Gambar 8. Emit_test.php

4.2 Data Upload using API

After the data is uploaded using the emit test endpoint the data will be received by server.js which can then be viewed using the logging console cloud conectivity

```
- info: SocketIO > Disconnected socket 5802rtpgluw511ffAACR
- info: SocketIO > Connected socket -p6qHP4TEf0zA0zAACs
- info: ElephantIO broadcast > {"temp": "27", "ultrasonik": "45", "soil": "20"}
- info: SocketIO : Received 1 messages
- info: SocketIO > Disconnected socket -p6qHP4TEf0zA0zAACs
- info: SocketIO > Connected socket D0Wk5dPlak4RS1scAACT
- info: ElephantIO broadcast > {"temp": "27", "ultrasonik": "45", "soil": "21"}
- info: SocketIO : Received 1 messages
- info: SocketIO > Disconnected socket D0Wk5dPlak4RS1scAACT
- info: SocketIO > Connected socket w3cjuDUABdgWIsE1AACU
- info: ElephantIO broadcast > {"temp": "27", "ultrasonik": "45", "soil": "21"}
- info: SocketIO : Received 1 messages
- info: SocketIO > Disconnected socket w3cjuDUABdgWIsE1AACU
- info: SocketIO > Connected socket cRM5QoHVuHeDGe-gAACV
- info: ElephantIO broadcast > {"temp": "27", "ultrasonik": "45", "soil": "21"}
- info: SocketIO : Received 1 messages
- info: SocketIO > Disconnected socket cRM5QoHVuHeDGe-gAACV
- info: SocketIO > Connected socket sFPHIp-1cCJVrjgXAACW
- info: ElephantIO broadcast > {"temp": "27", "ultrasonik": "45", "soil": "21"}
- info: SocketIO : Received 1 messages
- info: SocketIO > Disconnected socket sFPHIp-1cCJVrjgXAACW
- info: SocketIO > Connected socket PF4V23n2CgDvstTAACX
- info: ElephantIO broadcast > {"temp": "27", "ultrasonik": "45", "soil": "21"}
```

4.3 Reading Value from API

To read value from API on website we need to create another script. In that script also we need to provide data.temp, data.ultra, and data.soil. By using socket we can update the sensor data on cloud continuously.

```
<script src="//cdnjs.cloudflare.com/ajax/libs/socket.io/1.7.4/socket.io.min.js"></script>
<script>
var socket = io.connect('//35.240.163.237:1337');
const temperatureDisplay = document.getElementById('temperature');
const humDisplay = document.getElementById('ultra');
const soilDisplay = document.getElementById('soil');

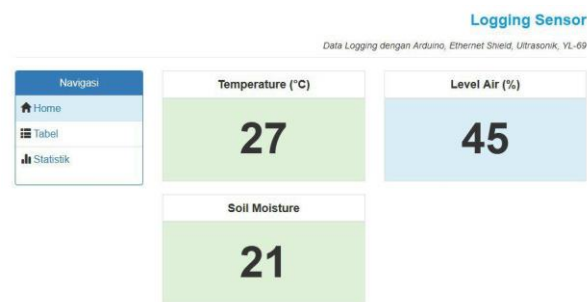
socket.on('connect', function () {
  console.log('connected');

  socket.on('broadcast', function (data) {
    temperature.innerHTML = `${data.temp}`;
    ultra.innerHTML = `${data.ultra}`;
    soil.innerHTML = `${data.soil}`;
  });

  socket.on('disconnect', function () {
    console.log('disconnected');
  });
});
</script>
```

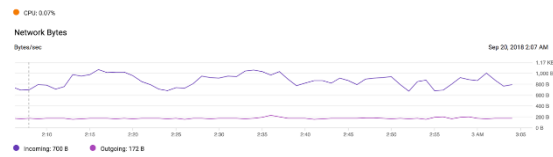
Gambar 9. Reading value

4.4 Display logging sensor on Website

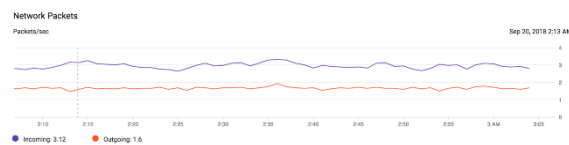


Gambar 10. Logging sensor Website

In this study several tests were carried out, namely the transmission testing of the system and the system reliability testing served the request



Gambar 11. Network Bytes



Gambar 12. Network Packet

Conclusion

The development of a system for smart agriculture can greatly benefit from the knowledge of the soil and water dynamics. Besides the soil moisture monitoring, to assess the plants water needs for its proper and healthy development, it was also important to assess the possibility of natural resources' usage optimization. Experimental tests on a real case scenario at a farm allowed the identification of some limitations, mainly related to energy consumption. Ongoing developments address the storage of historical information on the cloud concerning an agriculture field. The availability of a large amount of valuable data on the cloud enables the creation of intelligent services with very high potential, such as data correlation between different cultures and/or fields, plant disease estimation through the application of machine learning techniques, or determination of the most appropriate land cultivation according to soil conditions. In this research we are able to send data on cloud by using google cloud compute engine socket protocol as server.js as a gateway with realtime connection

References

- [1] Anonim, Overview of the Internet of things. Recommendation ITU-T Y.2060. Overview of the Internet of things. Recommendation ITU-T Y.2060, 2013.
- [2] Hidayat et al., "Implementasi Protokol Websocket Pada Perangkat Non IP Berbasis NRF24L01," vol. 02, no. 06 juni, pp. 2058–2066, 2018.
- [3] Pimentel and B. G. Nickerson, Communicating and Displaying Real-Time Data with WebSocket. Dalam: Programming Web Interface. s.l.:IEEE Computer Society. 2012.
- [4] Candelas, "Experiences of Using Arduino for Laboratory Experiments of Automatic Control and Robotics, IFAC: Alicante.," 2015.
- [5] Durfee, Arduino Microntroller Guide, s.l. University of Minnesota, 2011.
- [6] Hasibuan, A., Mustadi, D. I. E. Y. , Syamsudin, and I. M. A. Rosidi, Design and Implementation of Modular Home Automation Based on Wireless Network, REST API and

WebSocket. Internastional Symposium on Intelligent Signal Processing and Communication Systems (ISPACS),. 2015.

[7] P, M., Kramer and S. Molnar, “Comparison of Web Transfer Protocols. Budapest, High Speed Network Laboratory.,” 1997.

[8] P, M. R., Suharsono and R. Primananda, Sistem Pengamatan Kelembapan Tanah Dengan Sensor dan Mekanisme Publise-Subscribe. Computer Science Minor Thesis, 2016.