

# Prototype Penggunaan IoT untuk Monitoring Level pada Penampung Air Berbasis ESP8266

Steven Sachio<sup>1</sup>, Agustinus Noertjahyana<sup>2</sup>, Resmana Lim<sup>3</sup>

<sup>1,2</sup>Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Kristen Petra

<sup>3</sup>Program Studi Teknik Elektro Fakultas Teknologi Industri Universitas Kristen Petra

Jl. Siwalankerto 121 – 131 Surabaya 60236

Telp. (031) – 2983455, Fax. (031) – 8417658

E-Mail: Steven.sachio@gmail.com, Resmana@gmail.com, Agust@petra.ac.id

## ABSTRAK

Saat ini, hampir semua komunikasi menggunakan internet, tetapi tidak itu saja, sekarang objek juga bisa berkomunikasi satu dengan yang lain. Konsep ini dinamakan *Internet of Things* (IoT). Things di dalam IoT bisa di artikan barang-barang yang kita gunakan setiap hari. Di dalam jurnal ini barang yang dipakai adalah penampung air atau bak mandi. Mengapa penampung air? Karena, seringkali apabila kita memakai air di dalam penampung air dan level nya berkurang kita harus mengisi nya kembali untuk orang lain yang akan memakai penampung air. Kebanyakan dalam proses pengisian air ini, kita lupa mematikan pompa atau keran air sehingga air di dalam penampung air meluber dan air terbuang percuma. Dari permasalahan tersebut digunakan lah konsep dari *Internet of Things* yang diharapkan untuk dapat memecahkan permasalahan tersebut. Dalam journal ini solusi untuk memecahkan permasalahan tersebut adalah membuat sebuah modul atau perangkat yang dapat me monitoring level pada penampung air. Perangkat atau modul bisa mematikan dan menyalakan pompa atau keran air sehingga air di dalam penampung air tidak meluber dan tidak terbuang percuma. Untuk membuat perangkat atau modul yang dapat menghasilkan level penampung air digunakan sensor ultrasonic. Tetapi itu memunculkan hipotesis: apakah ultrasonic sensor bisa membaca level dari penampung air dan bagaimana akurasi nya?

**Kata Kunci:** *Internet of Things*, IoT, ESP8266, *Microcontroller*, Penampung air, Monitoring

## ABSTRACT

*Nowadays, almost all communication using internet, but is not all, now object also can communicating each other, this concept is calls Internet of Things (IoT). Things in IoT can be everything that we use every day. In this journal, the thing is water container or in Indonesia it calls "bak mandi". Why water container? Because, after we use water in the water container usually the level of the water container is reduced and we must fill it up for other people that use the water container. Usually in the process of filling the water container, we forgot to close the valve or pump so water is overflowing and the water is wasted. This is why we thought using the Internet of Things concept can solve the problem. In our solution is make a device or module that can monitoring level of the water container. That device or module can open and close the pump or valve automatically so water not overflowing and wasted. To make device or module that can generate level of the water container we use ultrasonic sensor. But that bring up hypothesis: could ultrasonic sensor read level of water container and how good the accuracy?*

**Keywords:** *Internet of Things*, IoT, ESP8266, *Microcontroller*, Monitoring, Water Container

## 1. PENDAHULUAN

Internet sudah menjadi kebutuhan masyarakat dunia. Dengan internet, orang dapat berkomunikasi dengan satu sama lain dengan mudah dan cepat. Dengan internet orang yang berada di benua satu dapat berkomunikasi dengan orang yang terdapat di berbeda benua, tetapi internet tidak hanya menghubungkan orang dengan orang lain tetapi objek juga dapat berkomunikasi dengan objek lain. Itulah konsep dari *Internet of Things*.

Konsep dari *Internet of Things* adalah konsep dimana objek dapat memiliki kemampuan untuk mengirimkan data ke dalam jaringan internet tanpa membutuhkan interaksi manusia ke manusia atau manusia ke komputer [5]. Things di dalam *Internet of Things* bisa saja barang yang kita gunakan sehari-hari seperti monitoring implant jantung, sensor yang mengingatkan pengendara jika tekanan ban berkurang, tempat sampah yang dapat menginformasikan kepada user jika tempat sampah sudah penuh dan masih banyak lagi. *Internet of Things* sangat lah dekat hubungan nya dengan komunikasi antara mesin dengan mesin.

*Internet of Things* biasanya didukung oleh perangkat yang memiliki beberapa sensor seperti Intel Edison dan Intel Galileo, dari *Arduino* ada *Arduino Uno*. Di dalam perangkat data dari sensor dapat dibaca dan diproses lebih lanjut.

Kedepannya *Internet of Things* dipakai untuk mengambil semua jenis data dari sensor-sensor, sehingga data yang terkumpul dapat digunakan untuk menganalisa pasar, menyediakan data untuk perusahaan sebagai referensi ketika menciptakan produk.

Penampung air adalah barang yang kita gunakan sehari hari untuk menampung air sebelum dipakai untuk keperluan mandi, mencuci pakaian dan masih banyak lagi. Tetapi setiap kali penampung air dipakai air di dalam penampung air akan berkurang dan kita harus mengisi nya sehingga orang lain dapat memakai air di dalam penampung air tersebut. Pada proses pengisian air inilah seringkali kita lupa untuk menutup pompa atau keran karena waktu untuk mengisi penampungan air biasanya lama. Saat menunggu ini biasanya kita meninggalkan keran atau pompa untuk mengerjakan hal yang lain sehingga lupa untuk menutup keran atau pompa ketika penampung air penuh dan air terbuang percuma.

## 2. TINJAUAN PUSTAKA

Untuk membuat Prototype Penggunaan *IoT* untuk Monitoring Penampung air Berbasis ESP8266. Harus didasari dengan teori dan penjelasan dari beberapa component sehingga mengerti bagaimana karakteristik dan cara kerja dari proyek ini.

### 2.1 *Internet of Things*

*Internet of Things* terdiri dari 2 kata kunci, *Internet* dan *Things*. *Internet*, memiliki arti *interconnection-networking*, dimana jaringan komputer yang terkoneksi satu dengan yang lain dengan menggunakan protokol TCP/IP (*Transmission Control Protocol/Internet Protocol*).

*Things* di dalam *Internet of Things* merupakan objek yang digunakan sehari hari dimana informasi diambil melalui sensor yang membaca keadaan lingkungan sekitar dengan real time dan tanpa adanya intervensi manusia. Seperti temperatur ruangan dan kelembapan udara.

Jadi *Internet of Things* memiliki arti objek yang dapat menghasilkan data melalui sensor dan data tersebut di kirimkan ke computer atau *server* melalui koneksi internet. *Internet of Things* juga sangat dekat hubungannya dengan komunikasi antara machine ke machine (M2M) yang berkomunikasi tanpa adanya intervensi manusia [4]. *Internet of Things* juga merupakan langkah pertama dari gagasan *Smart World* yang merupakan campuran dari data objek dan smart city [5]. *Internet of Things* memiliki 3 komponen utama:

1. Hardware yang terdiri dari sensor-sensor, akuratur, *microcontroller* dan perangkat komunikasi.
2. Middleware yang terdiri dari penyimpanan data dan analisa data
3. Presentation yang terdiri dari bagaimana data di visualisasikan dengan menghasilkan informasi yang berguna.

### 2.2 ESP8266 *Microcontroller*

*Microcontroller* adalah chip yang memiliki CPU (Central Processing Unit), memori, I / O, dan modul ADC (Analog to Digital Converter) yang digunakan untuk mengubah voltase listrik (sinyal analog) menjadi bit (digital) sehingga bisa dibaca oleh komputer.

ESP8266 *Microcontroller* adalah *microcontroller* yang dirancang oleh Espressif System, ESP8266 dirancang agar Wi-Fi terintegrasi secara langsung, sehingga ESP8266 tidak memerlukan modul Wi-Fi. Jika ESP8266 dibandingkan dengan Arduino, ESP8266 memiliki beberapa kelemahan yang didapat dari jumlah pin analog yang dimiliki ESP8266 dengan Arduino memiliki 5 pin analog sedangkan ESP8266 hanya memiliki 1[8].

ESP8266 biasanya sudah tertanam dengan board development seperti Wemos D1\_R2. ESP8266 sudah tertanam dengan Wemos D1\_R2 dengan form factor yang sama seperti Arduino Uno.

*Microcontroller* ESP8266 memiliki spesifikasi sebagai berikut [3]:

1. Menggunakan CPU Tensilica L106
2. Memiliki Wi-Fi yang bekerja pada peraturan IEEE 802.11 b / g / n
3. Memiliki ADC 10 bit dengan 1 pin Analog

4. Memiliki 10 input / output digital
5. Dukungan eksternal Memori flash 512kb (kilobyte) - 4MB (Mega Byte)

### 2.3 *Ultrasonic sensor*

Gelombang ultrasonik adalah gelombang suara pada frekuensi di atas 20khz. Gelombang ultrasonik dapat merambat dari media padat, cair, dan gas. Gelombang ultrasonik memiliki kemampuan memantul jika terkena materi padat dan cair. Pada sensor ultrasonik ada dua modul dalam satu rangkaian, yaitu modul *receiver* dan modul *transmitter*. *Transmitter* digunakan untuk menghasilkan dan mengirimkan gelombang ultrasonik ke arah benda yang diamati [6]. Saat gelombang ultrasonik menyentuh atau menabrak benda, gelombang ultrasonik memantul dan ditangkap oleh modul *receiver*. Perbedaan waktu pada modul *transmitter* dan modul *receiver* menentukan seberapa jauh objek tersebut diamati. Setelah mendapatkan perbedaan waktu antara pemancar dan penerima, waktu harus dikonversi menjadi panjang (cm) dengan menggunakan rumus ini (1): [7]

$$\begin{aligned} \text{length(cm)} \\ &= \text{milisecond (ms)} \\ &\div 29 (\text{microseconds per centimeter}) \\ &\div 2 \end{aligned} \tag{1}$$

### 2.4 *Blynk*

*Blynk* adalah aplikasi berbasis Android dan iOS yang digunakan untuk membuat dashboard digital tempat membuat antarmuka grafis melalui widget yang disediakan untuk kebutuhan proyek *IoT (Internet of Things)* [2]. *Blynk* dapat bekerja pada beberapa mikrokontroler seperti ESP8266, Arduino, Raspberry Pi, SparkFun dan lain-lain.

*Blynk* juga menyediakan *server Blynk* yang digunakan untuk membuat sistem *Blynk* di *server* lokal. *Server Blynk* memiliki requirement untuk menginstal java 8 pada sistem operasi yang akan digunakan untuk *server Blynk*. [1]

### 2.5 Relay

Relay adalah saklar listrik untuk memutuskan dan menghubungkan rangkaian listrik dalam kondisi tertentu. Di dalam relay terdapat coil yang digunakan untuk menggerakkan saklar atau kontak dengan gaya magnetik. Jika coil diberi arus listrik maka akan menghasilkan medan magnet sehingga mengangkat saklar di dalam relay.

Jika dibedakan berdasarkan dari contact point nya terdapat 2 jenis:

1. *Normally close(NC)*: saklar di dalam relay akan selalu di posisi menutup jika relay tidak mendapatkan arus listrik.
2. *Normally open(NO)*: saklar di dalam relay akan selalu di posisi terbuka jika relay tidak mendapatkan arus listrik.

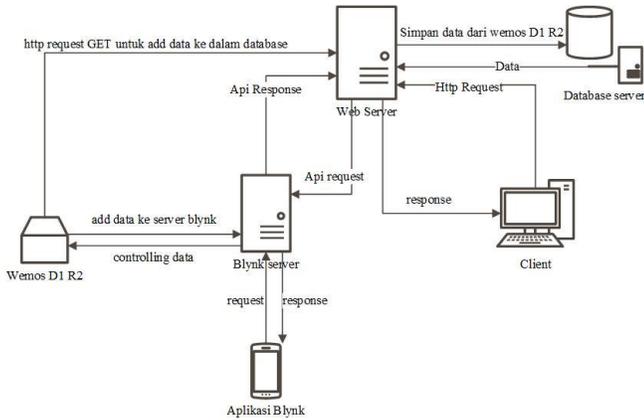
## 3. ANALISA dan DESAIN

### 3.1 Sistem Perangkat Lunak

Sistem software pada proyek ini menggunakan Sistem Informasi Monitoring Penampung Air dan *server Blynk*. Hal ini diperlukan

karena jika hanya menggunakan sistem *Blynk*, sistem *Blynk* hanya terbatas hanya untuk smartphone yang menggunakan aplikasi *Blynk* sehingga kurang fleksibel.

Skema komunikasi antara Sistem Informasi Monitoring Penampung Air dan *server Blynk* dapat dilihat pada gambar 1.



Gambar 1. Skema Komunikasi

Pada gambar 1 *microcontroller* yang digunakan adalah Wemos D1 R2. *Microcontroller* ini berkomunikasi dengan *server Blynk* dan Sistem Informasi Monitoring Penampung Air yang berbasis Web. Wemos D1 R2 mengirimkan data level setiap 4 detik ke Sistem Informasi Monitoring Penampung Air dengan menggunakan protocol GET. Data-data yang dikirimkan melalui protocol GET adalah token *microcontroller*, level dari penampung air, dan status dari valve atau pompa. Setelah mendapat request GET dari *microcontroller*, Sistem Informasi Monitoring Penampung Air akan memeriksa apakah token yang dikirimkan ada di dalam database? jika ada maka data level penampung dan status valve/ pompa akan di simpan di dalam database. Jika tidak ada maka data tidak di simpan di dalam database. Di dalam database setiap data di simpan dengan mencatat waktu data tersebut di tulis.

Wemos D1\_R2 juga mengirim data ke *server Blynk* namun data hanya disimpan ke dalam file yang kurang *flexible*, sehingga diperlukan Sistem Informasi Monitoring Penampung Air. Di samping *Microcontroller* mengirimkan data ke *server Blynk*, *microcontroller* juga menerima data konfigurasi dari *server Blynk*. data-data konfigurasi adalah ketinggian penampungan air, batas maksimal, batas minimal, mode, dan on/off pompa. Pada konfigurasi mode terbagi menjadi dua mode. Mode pertama merupakan mode auto yaitu pump/valve secara otomatis mengisi penampung air ketika level penampung air menyentuh batas minimum, sebaliknya jika level penampung menyentuh batas maksimum pump/ valve akan berhenti. Ketika pada mode semi auto dan on/off pada posisi on, *microcontroller* akan bekerja seperti pada mode auto tetapi jika on/off pada posisi off pump/valve akan tetap mati.

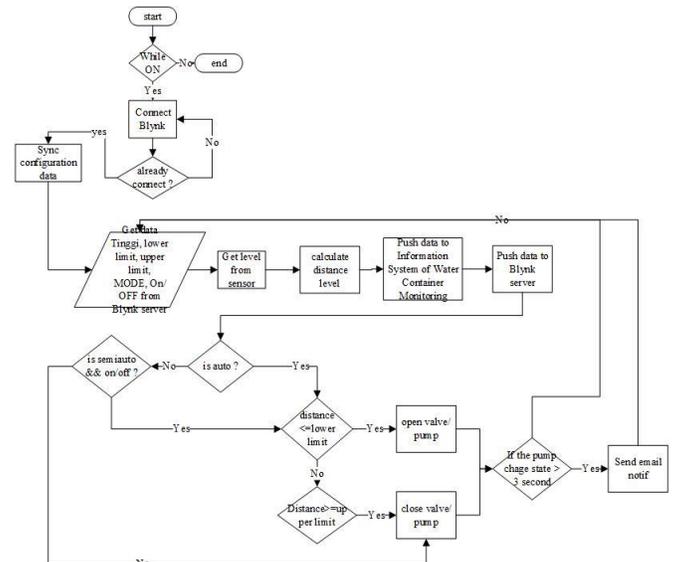
Sistem Informasi Monitoring Penampung Air juga berkomunikasi dengan *server Blynk* untuk merubah konfigurasi dengan menggunakan API (Application Programming Interface) sehingga user dapat merubah konfigurasi dari Sistem Informasi juga dari aplikasi *Blynk*.

Pada gambar 2. merupakan gambar yang menunjukkan *flowchart* dari coding *microcontroller*. Di dalam *flowchart* terdapat proses

untuk mendapatkan level dari sensor. Pada proses tersebut dinamakan proses ping, dimana sensor ultrasonic menembakkan gelombang ultrasonic ke permukaan air dan menangkap pantulan gelombang dari permukaan air. Perbedaan waktu saat penembakan gelombang dan penerimaan gelombang harus di convert melalui rumus (2) [7]

$$\begin{aligned} length(cm) &= \text{milisecond (ms)} \\ &\div 29 \text{ (microseconds per centimeter)} \\ &\div 2 \end{aligned} \tag{2}$$

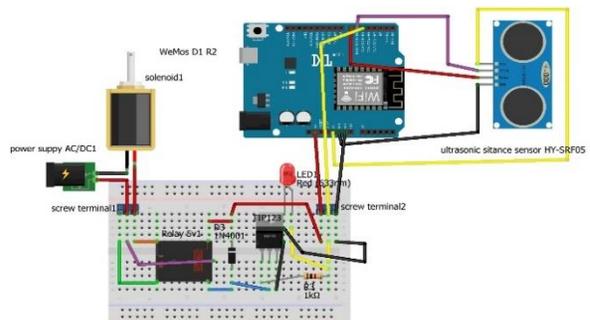
Setelah mendapat level air dari sensor, langkah selanjutnya adalah mengurangkan level dari sensor dengan ketinggian wadah air karena jika wadah air semakin mendekati penuh maka pembacaan sensor akan mengecil, maka untuk mendapatkan pembacaan yang benar harus mengurangi dengan ketinggian penampung air. Selanjutnya, Data di kirim ke Sistem Informasi Monitoring Penampung Air selama 4 detik sekali dan mengirimkan data ke *server Blynk* selama 1 detik sekali.



Gambar 2. Flowchart Microcontroller

### 3.2 Sistem Perangkat Keras

Dalam desain sistem perangkat keras terdiri dari Wemos D1 R2 sebagai *microcontroller*, pump/valve, adaptor, sensor ultrasonic, dan modul relay untuk menggerakkan pompa/pump. Gambar 3. merupakan skema rangkaian dari sistem perangkat keras.

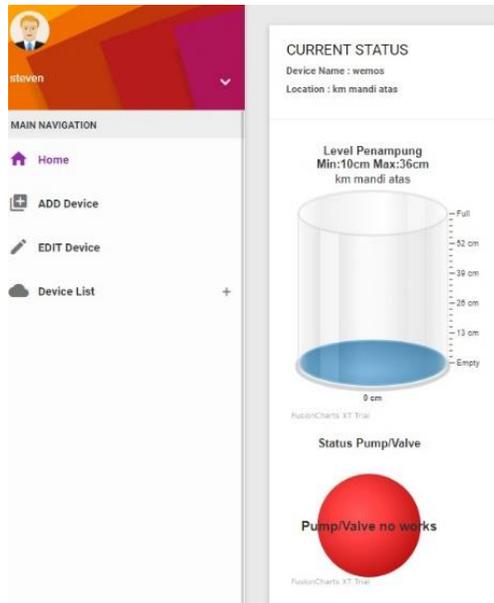


Gambar 3. Skema rangkaian Perangkat Keras

## 4. HASIL

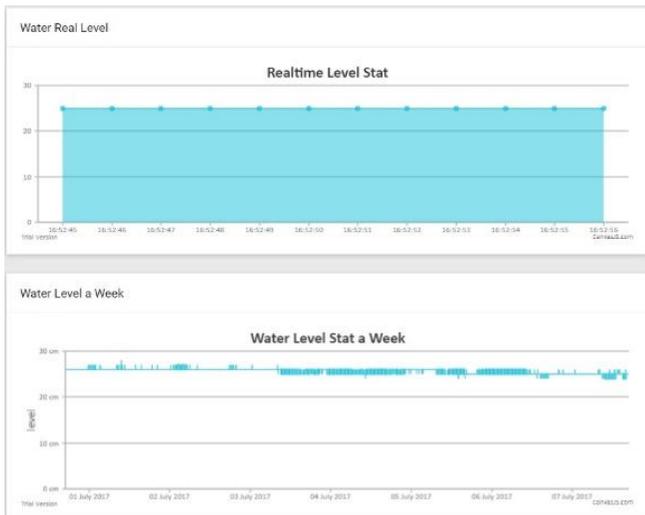
### 4.1 Sistem Perangkat Lunak

Hasil dari Sistem Informasi Monitoring Penampung Air. Gambaran aplikasi secara umum dapat dilihat pada gambar 4.

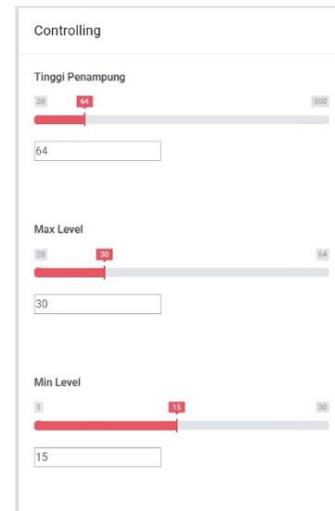


Gambar 4. Tampilan Home

Halaman device list merupakan halaman yang menampilkan data yang di simpan di dalam database dalam bentuk chart dan untuk mengganti konfigurasi. Hal ini dapat dilihat pada gambar 5 dan 6



Gambar 5 Halaman Device list (Chart)



Gambar 6 Halaman Device list (controlling)

### 4.2 Sistem Perangkat Keras

Pada gambar 7. gambar menunjukkan bagaimana sistem perangkat keras saling terhubung satu sama lain. Untuk sensor ultrasonik, sensor ultrasonik memiliki 5 pin vcc, trigger, echo, out, gnd. Pin vcc terhubung dengan output 5 v pada *microcontroller* Wemos D1\_R2, pin trigger terhubung dengan pin D6 pada *microcontroller* Wemos D1\_R2, pin echo yang terhubung dengan pin D7 pada Wemos D1\_R2, pin out tidak digunakan, pin gnd dihubungkan dengan pin ground pada Wemos D1\_R2. Untuk modul relay pemacu menggunakan pin D5 pada Wemos D1\_R2. Vcc dan gnd pada modul relay terhubung sama seperti sensor ultrasonik. Modul relay pin umum dihubungkan dengan kabel pompa dengan konfigurasi yang biasanya dekat. Adaptor digunakan untuk memberi tenaga pada *microcontroller* Wemos D1\_R2.



Gambar 7. Sistem Perangkat Keras

Setelah semua perangkat keras terhubung, selanjutnya di rakit dengan ke model penampung air seperti gambar 8. Sensor ultrasonik dipasang di atas wadah air. *Microcontroller*, adaptor dan modul relay dipasang pada wadah kecil sehingga tidak terkena air. Pompa dipasang dalam model wadah air.



**Gambar 8. Sistem Perangkat Keras & Model Penampung Air**

### 4.3 Pengujian

#### 4.3.1 Pengujian Sensor ultrasonik pada permukaan air statis

Pengujian dilakukan untuk mengukur ketelitian sensor ultrasonik. Pengujian ini dilakukan dengan membaca level air pada kondisi statis pada 0cm, 10 cm, 25 cm, dan 45 cm (maksimal) dengan beberapa timer pada proses ping: tanpa timer, timer 1ms, timer 10ms, 60ms. Data diambil dari 50 data pada database. Hasil data adalah:

**Tabel 1 Perbandingan pembacaan Level Penampung Pada Saat Kosong**

Perbandingan pembacaan Level Penampung Pada Saat Kosong				
	Tanpa timer	Timer 1 ms	Timer 10 ms	Timer 60 ms
Rata-rata kesalahan	55.25 %	26.06 %	0.03 %	0.0 %

**Tabel 2 Perbandingan Pembacaan Level Penampung Pada Saat Isi 10 cm**

Perbandingan Pembacaan Level Penampung Pada Saat Isi 10 cm				
	Tanpa timer	Timer 1 ms	Timer 10 ms	Timer 60 ms
Rata-rata kesalahan	13.2 %	5.8 %	10%	10%

**Tabel 3 Perbandingan Pembacaan Level Penampung Pada Saat Isi 25 cm**

Perbandingan Pembacaan Level Penampung Pada Saat Isi 25 cm				
	Tanpa timer	Timer 1 ms	Timer 10 ms	Timer 60 ms
Rata-rata kesalahan	0.56 %	0.08 %	0.0%	0.0 %

**Tabel 4 Perbandingan Pembacaan Level Penampung Pada Saat Isi 45 cm**

Perbandingan Pembacaan Level Penampung Pada Saat Isi 45 cm				
	Tanpa timer	Timer 1 ms	Timer 10 ms	Timer 60 ms
Rata-rata kesalahan	0.03 %	0.09%	0.0%	0.0 %

#### 4.3.2 Pengujian sensor ultrasonik pada saat bekerja

Pada pengujian data pada saat bekerja dapat ditunjukkan pada tabel 5. Pada pengujian ini setting an konfigurasi pengujian ini adalah 64 cm untuk tinggi penampung, batas maximal 35 cm, batas minimum 15 cm, mode auto.

**Tabel 5 Percobaan kerja penampung dengan konfigurasi tinggi 64cm, batas max 35cm dan batas min 15cm**

percobaan ke-	Pompa menyala (cm)	Pompa mati (cm)
1	15	34
2	14	31
3	14	34
4	15	32
5	14	34
Rata-rata	14.4	33
Rata-Rata Persen Kesalahan	4 %	5.71 %

Pada percobaan sensor ultrasonik saat bekerja sensor ultrasonik membuat kesalahan pada saat pump menyala dan pump mati. Pada saat pump menyala rata-rata error pembacaan sensor ultrasonik dengan level aktual adalah 4 %, namun pada kondisi pump mati rata-rata error pembacaan sensor ultrasonik meningkat menjadi 5.71 %. Hal ini disebabkan karena saat penampung air di isi, permukaan air tidak rata lagi.

#### 4.3.3 Pengujian koneksi Microcontroller ESP8266

Pada bagian ini adalah menguji bagaimana koneksi *microcontroller* (ESP8266) ke router Wi-Fi menggunakan protokol keamanan WPA-PSK, WPA2-PSK, dan WEP.

**Tabel 6 Koneksi Microcontroller dengan Wi-Fi Security Protocol**

Security Protocol	Encryption Protocol	Connection
WPA-PSK	AES	Ok
	TKIP	Ok
WPA2-PSK	AES	Ok
	TKIP	Ok
WEP	-	Ok

## 5. KESIMPULAN

Kesimpulan yang dapat diambil dari penelitian ini adalah sebagai berikut:

1. Dari percobaan antara ping tanpa menggunakan timer dan dengan menggunakan timer. Dapat disimpulkan bahwa semakin lama delay semakin akurat pembacaan sensor ke permukaan air.
2. Dari percobaan penampung air pada saat bekerja dapat disimpulkan bahwa ketika pump dari posisi menyala ke

mati, tingkat kesalahan membaca menjadi meningkat karena arus air membuat permukaan air tidak rata sehingga pantulannya tidak sempurna lagi.

3. Dari percobaan konektivitas dengan Wi-Fi dapat disimpulkan bahwa *Microcontroller* ESP8266 dapat dihubungkan dengan Wi-Fi dengan protokol keamanan WPA-PSK, WPA2-PSK, serta protokol WEP.
4. Dari percobaan sensor ultrasonik dapat digunakan sebagai pemantauan air dengan ambang kesalahan antara 2 cm

## 6. REFERENSI

- [1] BlynkTeam. 2015. Blynk-Server. Retrieved October 21, 2016, URI= <https://github.com/blynkkk/blynk-server>
- [2] BlynkTeam. 2015. Overview of blynk app. Retrieved October 21, 2016, URI=<http://www.blynk.cc/>
- [3] Espressif System IOT Team. 2015. ESP8266EX. Espressif System.
- [4] Garcia, C. G., Llorian, D. M., G-Bustelo, B. P., Lovelle, J. C., dan Garcia-Fernandez, N. 2015. Midgar:Detection of people through computer vision in the Internet of Things scenarios to improve the security in SmartCities, Smart Towns, and Smart Homes. *Future Generation Computer System*.
- [5] Gubbi, J., Buyya, R., Palaniswami, M., dan Marusic, S. 2013. Internet of Things (IoT): A Visison, architectural elements, and Future directions. *Future Generation Computer System*, 1645-1660.
- [6] Jatmiko, S., Mutiara, A. B., dan Indriati, M. 2012. Prototype of Water Level Detection System with Wireless. 52-59.
- [7] Margolis, M. 2012. *Arduino Cookbook*. United States of America: O'Reilly.
- [8] Singh, M., dan Goel, S. 2013. Review of Arduino Microcontroller. 2.