Design and Development of Low Cost Coral Monitoring System for Shallow Water Based on Internet of Underwater Things

Abid Famasya Abdillah[1], Muhammad Herwindra Berlian[1], Yohanes Yohanie Fridelin Panduman[1], Muhammad Aditya Wildan Akbar[1], Marlanisa Arifatul Afifah[1], Anang Tjahjono[1], Sritrusta Sukaridhoto[1], Shiori Sasaki[2]

[1]*Electronik Engineering Polytechnic Institute of Surabaya, Surabaya 60111, Indonesia.*

[2]*Multi Database and Multimedia Database Laboratory, Keio University (Shonan Fujisawa Campus), Japan.*

famasya@it.student.pens.ac.id

***Abstract*—Coral monitoring has become major focus to prevent coral bleaching so that various methods have been developed by researchers to analyze coral reefs' health. This paper proposed a low-cost coral monitoring system based for shallow water based on IoUT architecture. The proposed system consists of buoy as base component, controller unit based on Single Board Computer equipped with 5V power source, underwater camera and communication unit to send data into cloud server. This system is able to extend WiFi signal from underwater camera by using coaxial cable and transmit image into cloud server. From experiment, we measured that reliability delay using coaxial cable to capture and save image is 5,6 seconds. Results on cloud server experiment indicate distributed server's performance is 186,6 seconds comparing to 53,6 seconds on single server. Moreover, we also have developed bleaching classification resulting 84,3% correct results and an online representation model of retrieved coral images.**

***Index Terms*—Internet of Underwater Things; Coral Bleaching; Underwater Monitoring; Low Cost System.**

I. INTRODUCTION

In ocean ecosystem, coral reefs take many important roles to support underwater life. It provides feeding, spawning and nursery ground for underwater existences. Moreover, coral reefs also provide coast protection from damaging wave, help nutrient recycling, assist carbon and nitrogen fixing, and habitation for approximately 25% of diverse biotas. As the result, the livelihoods of 500 million people and income worth over $30 billion are at stake [1].

There are many challenges to keep coral reefs healthy, such as global climate change, sea temperature elevation, solar radiation and coral diseases. One of the most influential factor is coral bleaching. Coral bleaching is defined as expulsion of coral's pigment algae, causing coral reef become pale and allows the skeleton to become visible through transparent tissue [2]. When it happens, coral reefs will starve. If bleaching is prolonged, it will cause death to coral. A prevention act is needed to avoid a damage to coral reefs.

Coral monitoring has become a major focus to prevent bleaching. There are numerous approaches to monitor coral to prevent from bleaching. Collecting sample manually, conducting surveys, and map it into color index are no longer considered as effective [3,4,5]. Study of Hedley [5] showed that lack of coordination, ineffective cost and scaling issue still remains as challenges. Using remote sensing to perform monitoring program potentially addresses many of those caveats. Other approaches are also proposed by researchers, such as image analysis [3], computer vision and machine learning [6], multispectral remote sensing [4] and thermal stress analysis [7]. However, those approaches are relatively costly because not every researchers and scientists have adequate access to satellite imaging data [8].

In this paper, we propose a design of automated underwater monitoring system, using low cost, low power to monitor coral reefs on shallow water. By using Internet of Underwater Things (IoUT) architecture, we deliver a prototype that has portability, modularity, low latency data transmission, and integration into Big Data architecture. In particular, the contribution of this paper is the development of low-cost networked embedded system combined with underwater camera and Big Data architecture to provide continuously coral reefs monitoring system.

The remaining of this paper is organized as follows: Section 2 contain previous works from other researchers. Section 3 presents the system design of our device. Section 4 presents the experiment and implementation of the system. Finally, the paper concludes with Section 5.

II. PREVIOUS WORKS

Various methods have been actively developed by researchers to analyze and monitor coral reefs' health and many of them using remote sensing techniques. Many aspects with use of remote sensing are able to evaluate global environment affecting coral bleaching, such as using thermal stress to analyze coral bleaching [7] and multispectral thermal sensors [4]. The advance of this method is likely to continue because it's considered as promising method regarding its capability to respect the synthesis of multiple data products [5]. Similar approach using digital image processing is also proposed by Chow [3] which is able to quantify whiteness level of coral by image data into qualitative data. These research denote that using image processing is capable to assess underwater environment.

While the use of remote sensing is extensively used, other researchers studied to build Underwater Wireless Sensor Networks (UWSN) to put sensor nodes and retrieve image data, seawater temperature and GPS tracking [9]. The stationary underwater sensor nodes contain embedded computer with a wireless LAN and GPS sensor in every node of sensors. But considering the information capacity, they didn't use low frequency as the wireless communication unit and separate communication and sensors function.

A global project called Global Reef Record also has been conducted to map coral's condition across the world [6]. They build underwater vehicle that manually directed by diver to capture coral images data. The imagery datasets were annotated automatically using deep neural network called CoralNet to classify and analyze their characteristics.

Technological revolution of Internet of Things (IoT) bring new approach to explore water areas, which is known as Internet of Underwater Things (IoUT). It creates machine-to-machine network and links real life with physical device. IoUT also has some difficulties that must be tackled to create solid architecture such as different communication technologies, network density, tracking technologies, etc. We use proposed architecture presented by Domingo [10] into our low cost system because it has simple and clear layer diversity.

The use of transmission method must be considered as important to transmit data from underwater camera to the cloud server. This remains as challenge since radio waves do not propagate well underwater. Therefore, most communication of underwater rely on acoustic links, even though it has narrow bandwidth capacity [10]. Tsuyoshi et al. use watertight LAN cable to transmit data from underwater camera to floating embedded computer [9]. An experiment from Anguita et al. also represent that UWSN can be used to transport data with the throughput result of 100 kbps [11]. His experiment also indicate that it works not beyond 1,9 meters. We conclude that it cannot be used in our system. To transfer image from Raspberry Pi to cloud server, result from Pérez [12] represent that it's possible to use mobile network to connect into internet. Our low-cost monitoring system is also use this approach to send data into our cloud server.

III. System Design

Figure 1 illustrates an overview of our proposed low-cost monitoring system. It contains four main subsystems: buoy mechanics, underwater camera to take coral picture periodically, a control unit to run image capture program, and communication unit to transmit data into cloud server.

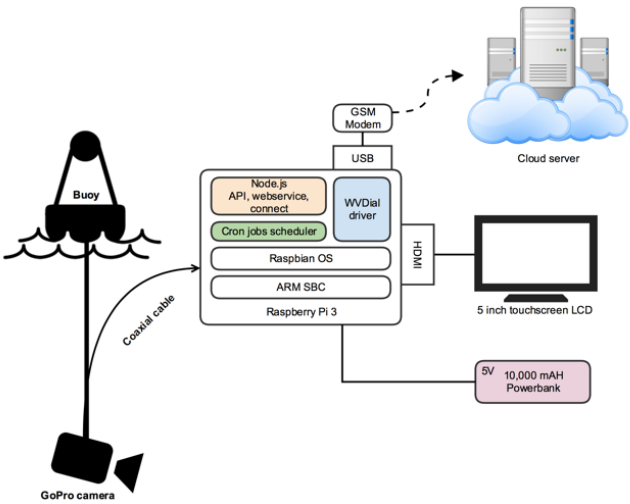

Figure 1: System design overview

A. *Buoy mechanics*

The buoy is the floating component where remaining subsystems are placed. This buoy is intended to keep control unit, power source and communication unit safe from water. It has wheel-shaped form and made of fiberglass to keep air trapped. There is also a small hole in the bottom of the buoy to put underwater rod to stick underwater camera in it.

This buoy is not designed as a fixed model. Therefore, we didn't attach it to the ocean floor to let it movable as seen in Figure 2.

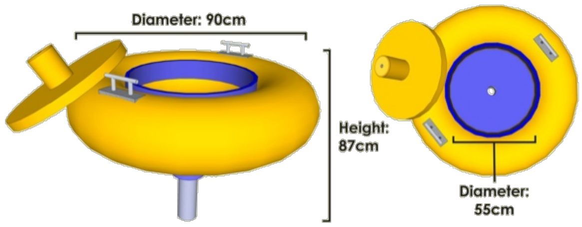

Figure 2: Floating buoy model

B. *Underwater Camera*

Underwater camera is a device attached to underwater rod that used to capture coral's condition. The rod has 3 meters in length because we targeted shallow water area. The device we use here is GoPro Hero 4 with WiFi turned on to communicate with Raspberry Pi. This kind of communication lead to loss of WiFi signal, because water will absorb 2.4 GHz frequency produced by GoPro. To overcome this limitation, many researchers use alternative approach such as using Underwater Acoustic Networks (UAN) or watertight LAN cable. But by using coaxial cable type RG 174/u, we are able to extend WiFi signal to reach Raspberry Pi by glue coaxial cable in end in underwater camera case and in control unit case.

We also made some adjustment on GoPro default setting regarding burst mode shoot. Instead of using 30 consecutive shoots, we set it into 3 sequence shoots to reduce GoPro's inner post processing time.

C. *Control Unit*

Control unit play the most important role because it handles many parts of the system: a) calling GoPro API to send capture command remotely b) run cron jobs scheduler to capture image automatically c) providing temporary image storage for captured image. The control unit we used is Raspberry Pi 3 model B version that has built in WiFi adapter and powered by 5V DC 10.000 mAH power bank. In the USB port, we attach Huawei E1750 as a communication device to connect to internet. For ease of use we connect HDMI port into touchscreen monitor to monitor program output.

In the software environment, we use Raspbian OS Jessie version as the SBC operating system. For application development, we use Node.js v4, Python 2.7, and Bash shell. The capture application is natively written in JavaScript run by Node.js. We utilized a function from goproh4 library to make HTTP API call to underwater camera. The algorithm works as follows:

```
require goproh4, request, fs
function getFile()
  listMedia() Then
    getLatestImage()
  End
end function

request.get(10.5.5.9)
```

```
.on(response.code == 200)
    setBurstMode() then
        startCapture()
    status = GetGoProStatus()
    if(status == ready)
        getFile() Then
            sendData()
        End
    End If
    Else
        print "not ready"
    End Else
End
```

This system is also using Python to listen physical button to start capture application. To operate continuously, we use cron job scheduler to execute application every particular time.

D. Communication Unit

Communication unit will provide internet connection and allow control unit to send image data into cloud server. The device of communication unit is GSM Modem Huawei E1750 which able to transmit image with throughput around 3,1 Mbps over 3G mobile network.

Control unit will execute command through wvdial driver, which will send command to broadband GSM Modem to connect into internet. The transmitted data will be sent to our Big Data cloud server, which will receive it and save it to be analyzed later.

IV. IMPLEMENTATION AND EXPERIMENTS

In this section, we will describe the implementation of the software and hardware development as well as the performance of the experiments.

After develop whole components individually, we managed them into integrated system as seen in Figure 3. Controller unit will handle image capture run by cron jobs scheduler and the server will process image to count whiteness level.

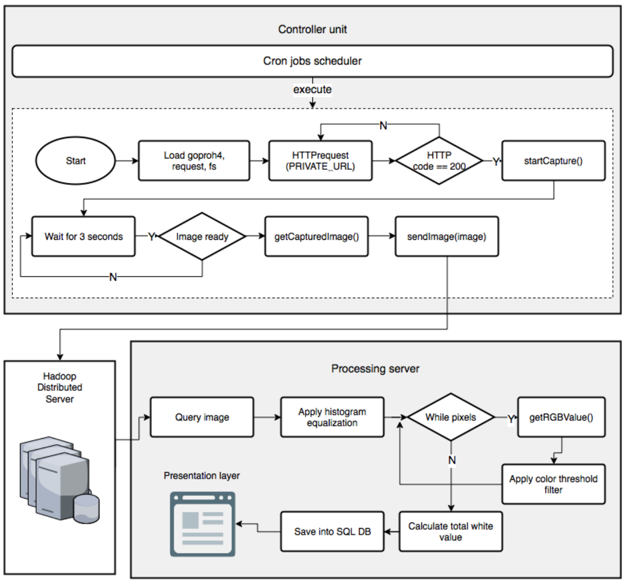

Figure 3: Algorithm of system development

Figure 4 represents controller's component structure of our low cost system. The dimension measurement of case box without its base are 36cm x 23cm x 6cm. When we attach 3m underwater rod in the middle of buoy, it has dimension approximately 90cm x 90cm x 310cm.

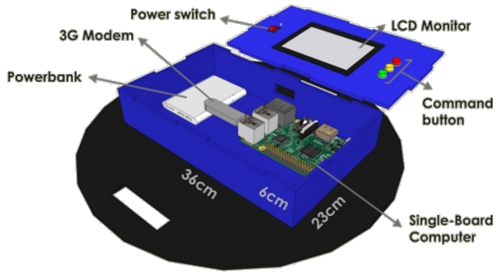

Figure 4: Communication unit components

The cost of our low-cost underwater surveillance system is relatively cheap comparing to another monitoring approach. As described in Table 1 below, total cost of implementation is Rp 11.991.000 or about 900 USD.

Table 1
Development cost calculation

Item	Qty	Price
Raspberry Pi Model B	1	Rp 600.000
Powerbank 10,000 mAH	1	Rp 209.000
Touchscreen LCD	1	Rp 609.000
Monopod (rod)	1	Rp 399.000
Coaxial cable	4m	Rp 44.000
GSM Modem	1	Rp 150.000
Case box	1	Rp 250.000
Underwater camera	1	Rp 6.980.000
Buoy	1	Rp 2.750.000
Total		Rp 11.991.000

For the experiments, we have committed an experiment to deploy our low-cost water monitoring system in Lenggoksono bay, Malang. In this experiment, we are performing benchmark on Big Data server comparing to conventional (single) server, capture transmission delay and simple image analysis.

Our first measurement was calculating image capture and saving time from Raspberry Pi to underwater camera using underwater rod with length of 3 meters. As designed before, we extend the WiFi signal by using coaxial cable type RG 174/u. The experiment took 9 consecutive captures using time Unix command with our Node.js application and download captured image immediately.

Table 2
Comparison of image fetching delay on the air and underwater

No	Over coaxial cable (sec.)	Over the air (sec.)
1	5,5	7,23
2	5,63	8,49
3	5,61	3,19
4	5,62	3,22
5	5,71	7,38
6	5,42	7,98
7	5,44	8,35
8	5,49	8,11
9	5,96	7,66

Table 2 shows that using coaxial cable to extends WiFi signal resulting delay 5,6 seconds in average. For comparison, by using normal WiFi in the air, it took 6,8 seconds in average to do same thing. In this experiment we use 4000px x 3000px image resolution and the size of images varies (about 4 - 6 Mb each).

We also measure the time of image transmission over 3G network to our cloud server. Our goal is comparing response time when we send data to single server than distributed Hadoop server. Computers we used as server were Dell Precision T1700 with Intel Xeon E3-1241 v3 processor, 8GB RAM and 1 Tb SATA hard drive. We simulate conventional single server by using one computer, and distributed server by using 1 master and 3 slave Hadoop nodes.

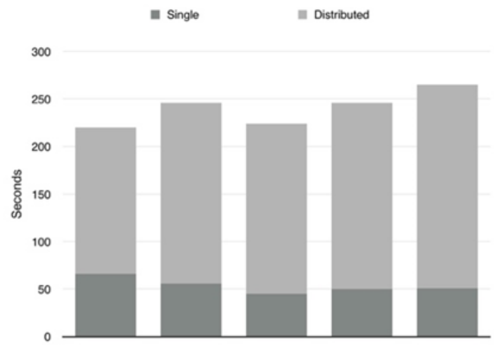

Figure 5: Single and distributed server comparison

Results reported in Figure 5 was single server's responses time were overall faster. While they need average time 53,6 seconds, Hadoop distributed server need 186,6 seconds to save our image. This is because Hadoop server must replicate received image into 3 slave nodes than save it into single server. Table 3 also indicate that scripts execution time in single server are nearly 0 second comparing to Hadoop distributed server that need 130 seconds. However, distributed architecture is still preferable to provide high availability, ability to process unstructured data and high scale data processing in the future [13].

Table 3
Comparison of image fetching delay on the air and underwater

Single server (sec.)	Distributed server (sec.)
8.082389831543E-5	130.25280404091
7.3909759521484E-5	131.46398210526
8.2969665527344E-5	130.60402917862
8.0108642578125E-5	127.8143260479
8.1062316894531E-5	131.66767096519

Next step is analyzing bleached coral by using color detection. We normalize retrieved image by using histogram autolevel equalization, applying RGB threshold to isolate bleaching color, and find bleaching percentage as seen on Figure 6. To count whiteness percentage, we use equation as written below:

$$Whiteness = \frac{\sum Px_b}{\sum Px} \times 100\%$$

where P_{xb} is bleaching pixel detected and P_x is image pixel. After applying the algorithm, we can detect 11,2% and 2,1% bleaching color from two image samples.

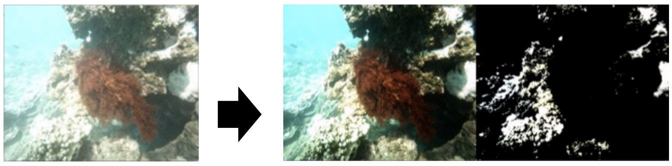

Figure 6: Image bleaching analysis

We also use confusion matrix to measure effectiveness of our detection method with whiteness threshold of 1,5%. Results performed on Figure 7 images denoted that color detection technique could achieve 84,3% correct results compared to manual classification.

	Actual: YES	Actual: NO	
Predicted: YES	19	6	25
Predicted: NO	2	24	26
	21	30	

Figure 7: Confusion matrix of image bleaching analysis

For the representation model, we also have developed an online dashboard to presenting our retrieved images. This model loads our coral images from distributed servers and put them into interactive maps. We intend to make our platform open so that everyone can learn from it.

V. CONCLUSION

In this paper, we have presented a coral monitoring system that consists of buoy mechanics, controller unit, underwater camera and GSM modem. Compared to existing approach, our proposed model gives continuously coral monitoring with clear coral image, low price and relatively low image fetching delay. This system also has portable model so it can be moved easily.

We have overcome signal loss problem in underwater WiFi camera with coaxial cable. The capture command was successfully executed and our program was able to download captured image with average time 5,6 seconds. In the server side, Hadoop distributed server took 186,6 seconds in average to retrieve and save image into 3 slave nodes, comparing to 53,6 seconds on single server. However, Hadoop distributed server provides high availability, ability to process unstructured data and high scale data processing. For bleaching classification performance, we achieved 84,3% correct results by using color detection.

For future development, we have a plan to utilize our system with 360-degree camera to capture wider area of corals and build rigid Big Data architecture. Moreover, we also want to implement more advanced image processing analysis and classification to classify bleaching level of retrieved images automatically.

ACKNOWLEDGMENT

The authors would like to thank to EEPIS Robotics Research Center (ER2C) for the preparation and source code modification. This research is funded by KEMRISTEKDIKTI from Kerjasama Luar Negeri scheme with number 0969.01/PL14/PG/2016 and Hibah Program Kreatifitas Mahasiswa 2015.

REFERENCES

[1] Underwater Earth, "Global Coral Bleaching," [Online]. Available: http://www.globalcoralbleaching.org/. [Accessed 2016 June 7].

[2] T. J. Goreau and R. L. Hayes, Ambio-Journal of Human Environment Research and Management 23, 176 (1994).

[3] M. Chow, R. H. Tsang, E. K. Lam and P. A. Jr, Journal of Experimental Marine Biology and Ecology 479, 60 (2016).

[4] J. Hedley, C. Roelfsema, B. Koetz and S. Phinn, Remote Sensing of Environment 120, 145 (2016).

[5] J. D. Hedley, C. M. Roelfsema, I. Chollett, A. R. Harborne, A. R. Harborne, S. J. Weeks, W. J. Skirving, A. E. Strong, C. M. Eakin, T. R. L. Christensen, V. Ticzon, S. Bejarano and a. P. J. Mumby, Remote Sensing 8, 118 (2016).

[6] M. González-Rivero, P. Bongaerts, O. Beijbom, O. Pizarro, A. Friedman, A. Rodriguez-Ramirez, B. Upcroft, D. Laffoley, D. Kline, C. Bailhache, R. Vevers and O. Hoegh-Guldberg, Aquatic Conserv: Mar. Freshw. Ecosyst. Aquatic Conservation: Marine and Freshwater Ecosystems 24, 184 (2014).

[7] S. F. Heron, G. Liu, J. L. Rauenzahn, T. R. L. Christensen, W. J. Skirving, T. F. R. Burgess, C. M. Eakin and J. A. Morgan, Journal of Operational Oceanography 7, 3 (2014).

[8] G. B. Palmerini, Disaster Advances 5, 780 (2012).

[9] T. Suzuki, K. Kato, E. Makihara, T. Kobayashi, H. Kono, K. Sawai, K. Kawabata, F. Takemura, N. Isomura and H. Yamashiro, International Journal of Distributed Sensor Networks 2014, (2014).

[10] M. C. Domingo, "An overview of the internet of underwater things," Journal of Network and Computer Applications 35, 1879 (2012).

[11] D. Anguita, D. Brizzolara, and G. Parodi. SENSORCOMM 2009. Proceedings of the Third International Conference, (2009), 18-23 June; Athens, Greece.

[12] C. A. Pérez, H. N. Hellín, F. S. Valles, A. I. García and R. Torres. Martech 2015. Proceedings of the Sixth International Workshop on Marine Technology, (2015), 15-17 September; Barcelona, Spain.

[13] T. White, Hadoop: The Definitive Guide, O'Reilly, California (2012).