ANDROID MOBILE APPLICATION DEVELOPMENT FOR FIELD VISIT DOCUMENTATION

Risa Herdianto[1], Eko Budi Setiawan[2]

[1,2] Universitas Komputer Indonesia
Jalan Dipatiukur No. 112-116, Coblong, Lebakgede, Bandung, Jawa Barat 40132
E-mail : aputra.zha@email.unikom.ac.id[1], eko@email.unikom.ac.id[2]

ABSTRACT

Documentation is a record of past events, which is in the form of text, images, and others. Documentation is usually used in an accountability report of an event as is done by the Food Department of Majalengka Regency when conducting activities. The resulting documentation is in the form of images or photos. The problem that arises is the lack of information obtained from the photos of the documentation and the photos are often scattered and disorganized, which caused the reporting officer difficulty to finding and knowing the activity information carried out from the photo. In addition, as technology develops, the Food Department of Majalengka Regency also wants to apply technology that can help the officers to analyze nitrogen needs of rice crops, so that nitrogen fertilizer can be done more effectively and efficiently. In this study discuss about the development of android mobile application for helping the Food Department of Majalengka Regency to managing and performing documentation by utilizing the photo tagging feature. Based on the test results performed, it can be concluded that the application already built run well and as expected, both in terms of black box testing and beta testing.

Keywords : android, photo tagging, documentation, nitrogen analysis.

1. INTRODUCTION

Food Department of Majalengka Regency is an executing element of local government that execute regional authority in food sector and responsible for progress of food sector through increasing human resource of agriculture, supporting food security, increasing value of food product, increasing production facilities and realizing natural resource sustainability through development food security in Majalengka District.

One of the activities commonly conducted by the Food Agency of Majalengka Regency is to visit the field, either to conduct extension to the farmers, check the irrigation infrastructure and to check the quality of crops grown by the farmers. When conducting field trips, Food Department staff must carry out documentation as evidence of field visits. One form of documentation is a photo of the activities undertaken. After the field visit activity, the Dinas Pangan staff needs to prepare activity reports, which in the report should attach photos of the activity.

Constraints are often encountered when the employee who makes the report is not the person who took the photo of the activity, which causes often the employee is confused about which photo in accordance with the location of the activity, because not infrequently field visits conducted to several different locations at once on the same day, and that means taking many photos and from several different locations as well. In addition, when there is an event conducted by the Department of Food and the Agricultural Department of Majalengka Regency, photographs of field visits also need to be displayed in the event. However, before printing the photo was previously edited in advance to provide a description of the activities undertaken in the photo. It becomes less efficient because it takes a lot of time to edit one by one of the many photo activities that should be displayed.

Based on the above description, then conducted research to create an application that can help the Food Department in making documentation of field visits using a smartphone with the Android operating system. The android operating system is chosen because it is an operating system with an open source license so that it can be freely developed by everyone to support activities and daily work, including in agriculture.

With this research is expected to assist the Department of Food in conducting documentation process field visits and manage the results of the documentation so as to speed up the work done.

2. RESEARCH CONTENTS

2.1. Geotagging

Geotagging is the process of adding geographic information to various media in the form of metadata. Data usually consists of coordinates such as latitude and longitude, but may even include altitude, distance and place names. Geotagging applied to images is also known as GPS photo

tagging, which is the process of adding GPS position data information (Latitude, Longitude, Altitude) into the metadata of a digital photo.

Photos in digital devices store metadata in EXIF (Exchangeable Image File Format), including time, orientation of the camera, whether using flash lights and other camera details used such as aperture, exposure, and focal length, and so on. EXIF data allows us to search and manage photos effectively with various conditions through multiple attributes [1].

2.2. Reverse Geocoding

The term geocoding generally refers to translating the address that people can read into a location on the map. The process of doing the opposite, ie translating the location on the map into a readable address, is known as Reverse Geocoding. Reverse Geocoding displays the street address of the physical location specified by latitude and longitude [2]

Latlng is the value of latitude and longitude that specifies the desired location in obtaining the closest address anyone can read. Whereas placeId is the place ID of the desired location in obtaining the address that people can read. The place ID is a unique identifier that can be used with other Google APIs. For example, a developer can use placeId returned by the Google Maps Roads API to get address points that have been identified. This value can only be specified if the request includes an API key or a Google Maps API client ID. [3].

2.3. Metadata EXIF

Metadata EXIF (Exchangeable Image File Format) is data stored by a digital camera on an image that contains information about the condition and setting of the digital camera at the time of shooting. EXIF was developed by the Japanese Electronics Industry Development Association (JEIDA) in an effort to simplify and create data exchange standards between image processing software or digital images and hardware such as cameras, and supported by almost all digital cameras [4].

General information that can be obtained from EXIF a digital photo is as follows:

1. What date and time a photo was made.
2. Brand, type of camera, and type of lens used to shoot photos.
3. Resolution of the camera used to take pictures. Whether or not the camera flash feature is used.
4. The location of the photo is taken, if the camera used supports GPS.

2.4. Google Maps

Google Maps is a free service provided by Google and is very popular which is a world map that we can use to see an area. The service also provides an API (Application Programming Interface) that allows other developers to take advantage of this application in their homemade applications.

The user has full access to the map, which allows him to control the view, change the zoom level, and move to a different location. Using Overlay, developers can annotate maps and handle user input [5].

Using the GoogleMaps API, developers can save time and money on building reliable digital map apps, so developers can focus only on the data to be displayed. In other words, developers only create a data while the map to be displayed is owned by Google so that developers do not mess around with making a map of a location, even the world.

To use Google Maps in Android apps, developers need to make sure they check Google APIs as their development targets. Google Maps is not part of the standard Android SDK, so developers need to look for it in the Google API add-on [2].

2.5. Android

Android is a mobile operating system based on a modified version of Linux. Android was originally developed by startup of the same name, Android, Inc. In 2005, as part of a strategy to enter the mobile space, Google bought Android and took over the development work (along with the development team) [2].

The main advantage of adopting Android is offering a unified approach to application development. Developers only need to develop for Android, and their apps can run on a variety of different devices, provided that supported devices use Android [2].

2.6. Open Weather Map

OpenWeatherMap is an online service that provides the latest weather data, including forecast data and updated historical data for developers of web services and mobile apps. For data sources, OpenWeatherMap uses meteorological broadcast services, raw data from airport weather stations, raw data from radar stations, and raw data from other official weather stations.

All data is processed by OpenWeatherMap by trying to provide accurate weather forecasts and weather maps online, such as cloud or rainfall. Beyond that, the service is focused on social aspects by involving weather station owners in linking services and thereby improving the accuracy of weather data. This ideology is inspired by OpenStreetMap and Wikipedia that make information free and available to everyone. OpenWeatherMap uses OpenStreetMap to display the weather maps.

2.7. Design Of TheSystem

The design of the system built in this study consists of the following sections.

2.7.1. System Architecture

In the development of this application has two systems of frontend and backend. Frontend is an application used by field officers to take photo documentation using android mobile device, while for backend is an application used by administrator officer or officer of report maker to perform data processing of user application and photo data of result of documentation. In performing data communication, communication design will be created using API (Application Programming Interface). The API is divided into 3 sections, API for data communication with application server, API for communication with Google servers and API for communication with OpenWeatherMap server. The description of the architecture of the system to be built can be seen on Figure .

Here is an explanation of the architecture of the built system.

1. The user's mobile device makes requests to the server or to the OpenWeatherMap Service via the API on the Internet network.
2. The server receives the request data from the mobile device as well as from the desktop device and determines the requested request type.
3. The server receives the data request and the server retrieves the data in the database.
4. The server receives the data transmission and the server stores the data in the database.
5. The server receives the requested data from the database.
6. The server sends the requested data response in the form of JSON to the mobile device for later processing of the user's mobile device.
7. Service OpenWeatherMap receives requests from mobile devices.
8. Service OpenWeatherMap sends a response in the form of JSON to a mobile device for later processing of the user's mobile device.
9. The user's mobile device receives a response from the server or from OpenWeatherMap.
10. The user's desktop device makes a request to the server on the internet network.
11. The server sends request requests from desktop devices to Google Service via the API on the Internet network.
12. The Google Service receives requests from the server.
13. Google Service sends a response to the server in accordance with its request.
14. The server sends requested data to the desktop device for later processing of the user's desktop device.

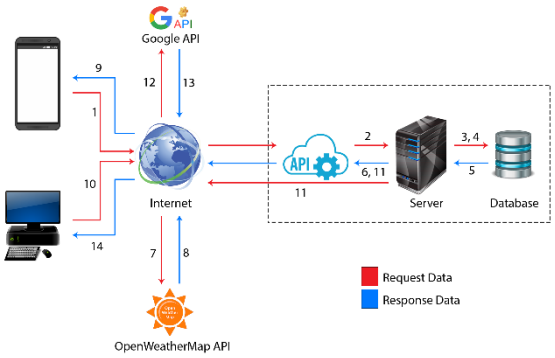


Figure . System Architecture

2.7.2. Documentation Procedure

Broadly speaking, in doing field visit documentation, the application will capture photos of field visits through the camera on the user's smartphone and the photo is then inserted with additional information in the form of GPS data, temperature data from the Open Weather Map based on where the cyclists are located, and information about the activity Which is done, which then additional information is stored into the EXIF data in the photo. Next, the image will be converted into a string format with the base64 encoding to be sent to the server. All data information contained in the photo is then stored in the database on the server. For more details about the procedure documentation process of field visits on the application to be built, can be seen on Figure .

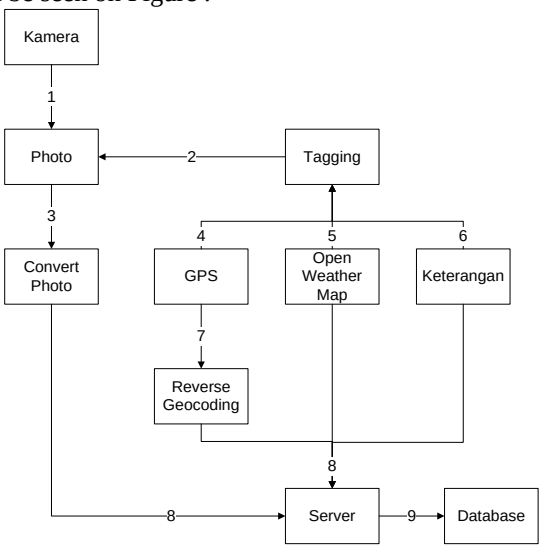


Figure . Documentation Procedure

2.7.3. JSON Data Analysis

The process of data communication between server and mobile application is bridged by API (Application Programming Interface) in the form of JSON data format (JavaScript Object Notation). The mobile application will request the required data to the server, then the server will respond by sending the requested data in the form of JSON data format. There are several modules used for the

communication process in this application, namely OpenWeatherMap module, login module, and documentation module. As detailed, JSON data analysis as follows.

1. OpenWeatherMap

   OpenWeatherMap is used to get temperature information around the location that will be used in the documentation process. The temperature information will be sent by the OpenWeatherMap server in JSON format. The data taken from JSON OpenWeatherMap is temperature (temp), which will be converted from Kelvin to Celsius units. The JSON OpenWeatherMap data structure can be seen in Table .

Table . JSON OpenWeatherMap Data Structure

JSON Data Structure

```
{
  "coord": {"lon": 108.23, "lat": -
6.84},
  "weather": [{"id": 501, "main":
"Rain", "description":
              "moderate rain",
"icon": "10d"}],
  "base": "stations",
  "main": {"temp": 296.819,
"pressure": 999.71, "humidity":96,
          "temp_min":
296.819,"temp_max": 296.819,
          "sea_level": 1024.03,
"grnd_level": 999.71},
  "wind": {"speed": 0.86, "deg":
159.005},
  "rain": {"3h": 5.505},
  "clouds": {"all": 56},
  "dt": 1493462346,
  "sys": {"message": 0.005,
"country": "ID", "sunrise":
          1493419686, "sunset":
1493462435},
  "id": 1636816,
  "name": "Majalengka",
  "cod": 200
}
```

2. Login

   Login is used in the authentication process to enter into the system on a mobile application. The data structure of JSON Login can be seen on Table .

Table . JSON Login Data Structure

JSON Data Structure

```
{
  "error": false,
  "user": {
    "user_id": "1",
    "nip": "111111",
    "username": "xxxxxx",
    "name": "xxxxxx",
    "img_profile": "xxxxxx.jpg",
    "log": "2017-04-25 08:13:42"
  }
}
```

3. Documentation

   Documentation is used to display a list of photo documentation based on the selected documentation album. The data structure of JSON Documentation can be seen on Table .

Table . JSON Documentation Data Structure

JSON Data Structure

```
{
  "error": false,
  "documentation": [
    {
      "doc_id": "1",
      "img_path":
".../public/images/xxx.jpg",
      "title": "xxxxxx",
      "description": "xxxxxx",
      "latitude": "- 6.111111",
      "longitude": "108.111111",
      "location": "xxxxxxxx",
      "temp": "30",
      "post_date": "2017-04-28
06:08:00",
      "name": "xxxxxx"
    },
  ]
}
```

2.7.4. Functional Requirements Analysis

Functional requirements analysis describes the process of activities that will be applied in a system and explain the needs of the system to run properly and in accordance with needs.

Analysis of functional requirements in the development of this system is divided into 2 (two) namely the analysis of functional requirements of the system on the mobile platform and analysis of system functional requirements on the website platform.

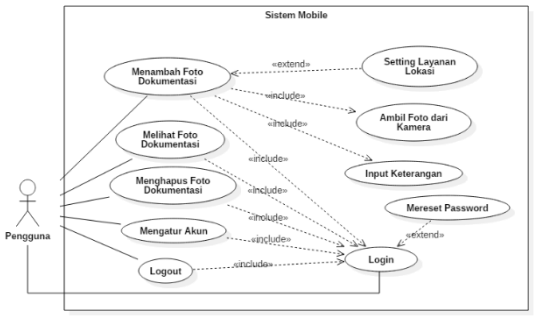


Figure . Use Case Diagram Mobile

Figure 3 above shows the use case diagram contained in the mobile device, where the user concerned is a field officer in charge of doing photo documentation using a smartphone device.

Figure 4 shows the use case diagram contained in the website tool, where the corresponding user is a report officer assigned as administrator with a role that has the authority to use functionality at the user interface level of the website, perform photo management documentation and management of user data.

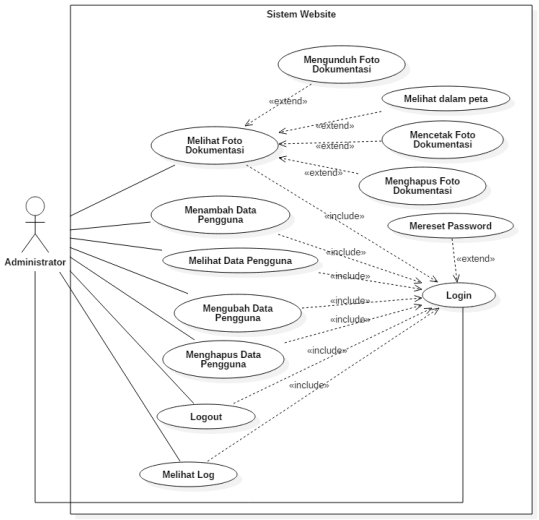


Figure . Use Case Diagram Website

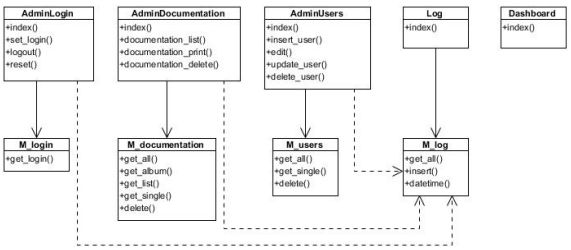


Figure . Class Diagram Website

Figure 5 above shows a class diagram of the website system, which consists of a controller and a model that handles all activities or activities performed by the administrator.

2.8. Results and Discussion

This app is tested by using an Android-based smartphone with the specs on Table . Testing is done with the aim to know the success of the system in dealing with functional that has been predetermined.

This application is designed in accordance with the instructions of google design materials to give a good look, but still simple to provide convenience to the user in understanding the use of this application.

Table . Mobile Device Specifications

Type	Samsung Galaxy J3
Display	5.0 inch, Resolution 720 x 1280 pixels
Operating System	Android v 5.1 Lollipop
Chipset	Chipset Exynos 3475 Quad, CPU Quad-core 1.3 GHz Cortex-A7
Internal Memory	8 GB
RAM memory	1,5 GB
Main Camera	Rear camera 8 MP, Autofocus, LED flash
GPS	A-GPS

2.8.1. Interface

Some interface views of the field visit documentation application made in this study are shown in Figure 6 through Figure 11.

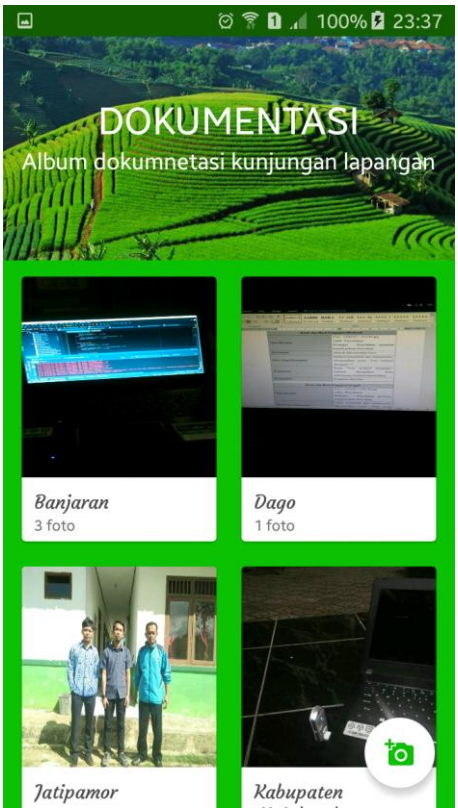


Figure . Documentation Photo Album Interface (Mobile)

Figure 6 above shows the album documentation interface on mobile devices, where the album will be automatically created by the system based on the location name when the user takes the photo of the documentation.

When a user shoots a new documentation at a new location or the location name is not in the album, the system will automatically create a new album, But if the user takes the documentation to a pre-existing location or the location name is in the album, the system will add the new documentation photo to the existing documentation album according to the name of the capture location.

Figure 8 on the side shows the album documentation interface on the website tools used by the administrator. On the website page the user can perform photo documentation management that has been made by the field officer of the mobile device.

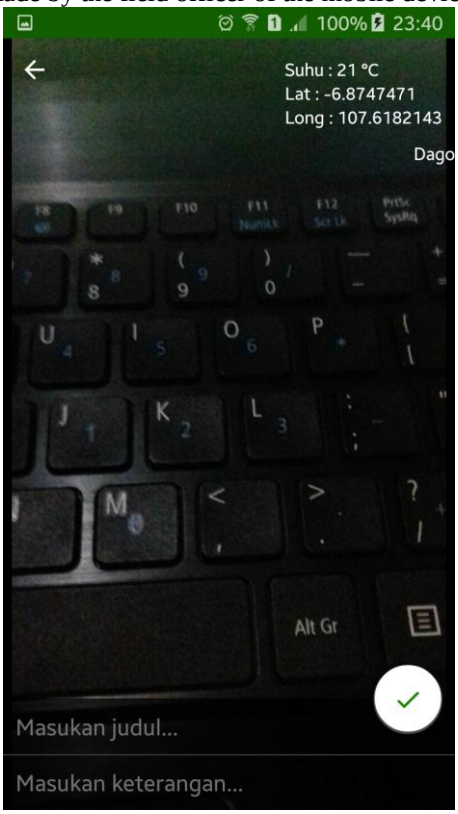


Figure . Add Documentation Photo Interface (Mobile)

Figure 7 above shows the interface when the user takes the photo-taking process of the documentation to add new photo documentation. In the process of adding the photo will display the coordinates of the location where the photo taken, the temperature of the area in that location, and the name of the location.

Before the user can upload a photo of the created documentation, the user must first enter the title and description of the created documentation photograph.

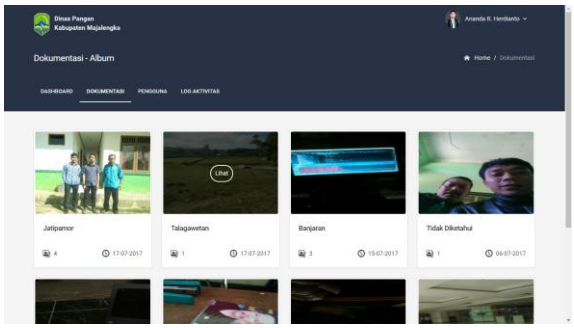


Figure . Documentation Photo Album Interface (Website)

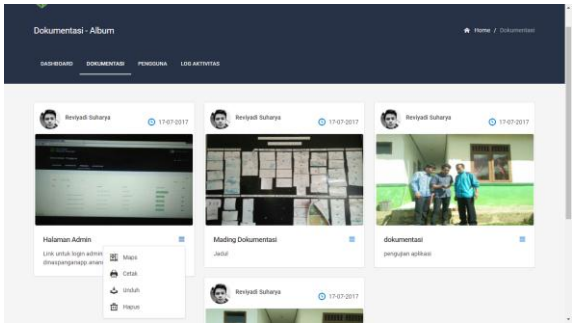


Figure . Interface Documentation List (Website)

In Figure 9 above, the administrator can view a list of photo documentation of the selected album. On this page users can view the location of the photo documentation on the map, print a photo of the documentation, download the documentation photos, and delete the photo documentation.



Figure . Documentation Print Picture

In Figure 10 shows the preview image of the documentation print while the administrator will print the documentation photo.

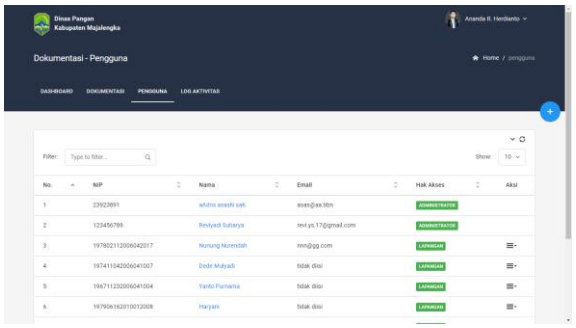


Figure . Manage User Interface

Figure 11 above shows a user data management page, where administrators can add, change or delete user data that can use applications on mobile devices.

2.8.2. System Testing

Testing the system is a stage to perform a series of tests to try a system that has been built with the aim to find errors or deficiencies in the system built. One of the system testing processes built on mobile devices can be seen on Table 5 and Table 6.

Table . Testing Add Documentation Photos with True Input

Cases and Test Results (Correct Data)	
Data Input	Photo : 22062017_161248.jpg Title : Penyuluhan Description : Penyuluhan pertanian kepada poktan desa talaga
Description	All input data filled.
Expected results	Successfully add documentation data. Showing messages "Foto berhasil diunggah!".
Observation	Message "Foto berhasil diunggah!" successfully displayed. Documentation data successfully added.
Conclusion	Testing Accepted

In Table 5 above, is the result of testing the addition of photo documentation with correct data input. The test can be received with the expected results and response from the system accordingly.

Table . Testing Add Documentation Photos with Incorrect Feedback

Case and Test Result (Incorrect Data)	
Data Input	Photo : 22062017_161248.jpg Title : (empty) Description : (empty)
Description	Title and Description on input data not filled.
Expected results	Failed to add documentation data. Showing messages "Masukan judul dan keterangan terlebih dahulu!".
Observation	Message "Masukan judul dan keterangan terlebih dahulu!" successfully displayed. Documentation data failed to add.
Conclusion	Testing Accepted

In Table 6 above, is the result of testing the addition of photo documentation with incorrect data input, where the title and description of the photo is not filled. The test can be received with the expected results and response from the system accordingly.

3. CLOSURE

From the results of testing the system has been done, indicating that the built applications already meet the functional requirements. However, in the process does not close the possibility of errors can occur one day. Functionally the system has been built already can produce the expected output.

Based on beta testing or testing of user satisfaction by interviewing the user system that is the staff of Food Department of Majalengka Regency, it can be concluded that this application has been able to facilitate in knowing the field activities information, and can accelerate the performance of officers in managing the data documentation results.

REFERENCES

[1] J. Kim, S. Lee, J. Won dan Y. Moon, "Photo Cube: An Automatic Management and Search for Photos Using Mobile Smartphones," *Dependable, Autonomic and Secure Computing (DASC)*, pp. 1228-1234, 2011.

[2] W.-M. Lee, Beginning Android 4 Application Development, Indianapolis: John Wiley & Sons, Inc., 2012.

[3] G. Inc., "Reverse Geocoding (Address Lookup)," Google, [Online]. Available: https://developers.google.com/maps/documentation/geocoding/intro#Rever seGeocoding. [Diakses 2017 Maret 2017].

[4] H. Wijayanto, I. Riadi dan Y. Prayudi, "Encryption EXIF Metadata for Protection Photographic Image of Copyright Piracy," *International Journal of Research in Computer and Communication Technology*, vol. V, no. 5, pp. 237-243, 2016.

[5] R. Meirer, Professional Android™ 4 Application Development, Indianapolis: John Wiley & Sons, 2012.

[6] "Penentuan Posisi Dengan GPS Untuk Survei Terumbu Karang," Puslit Oseanografi - LIPI, [Online]. Available: http://www.coremap.or.id/downloads/GPS.pdf. [Diakses 27 Maret 2017].