Android Application Design as Ground Control Station (GCS) and Waypoint Navigation in Unmanned Aerial Vehicle (UAV)

Mochamad Mobed Bachtiar[1], Fernando Ardilla[2], Abdi Alghifara Felinanda[3]
Department of Informatics and Computer Engineering
Politeknik Elektronika Negeri Surabaya
Surabaya, Indonesia, 60111
mobed@pens.ac.id [1]; nando@pens.ac.id [2]; abdialghi@gmail.com[3];

***Abstract*— The Unmanned Aerial Vehicle (UAV) device consists of several core systems, one of which is the avionics system. This system is part of aircraft control, and the technology developed is a flight controller, communication and Ground Control Station (GCS). Of the three technologies developed in the Avionics system, the development of GCS technology that is used for monitoring UAV conditions and mission input is still using a lot of personal computers so that it is felt to be less efficient for user mobility. Android-based GCS application development has begun to be developed for UAV device developers. From these problems in this study, a solution was made to create an android-based GCS application that can monitor aircraft conditions, remote control controls and can provide waypoint instructions for navigation and storing flight history data. There are 13 types of data that are processed for monitoring, with a range of communications using module telemetry within a maximum radius of 100-150 meters without obstacles or obstructions. Receiving data with an accuracy of 96.12% and a precision level of 0.7%, the average delay obtained is 0.303 s. For maximum video streaming distance that can be reached 100 meters with the fastest streaming speed of 3 Mbps and the longest application can be connected with the flight controller for 60 minutes.**

Keywords— UAV, Android, GCS, Waypoint, Navigation

I. INTRODUCTION

The use of Unmanned Aerial Vehicle (UAV) devices is increasingly being used every day in various fields, ranging from the fields of research, industry, film, agriculture to the general public. Some countries have also begun to develop UAV device products, especially in the avionics system. This system consists of flight controllers, communication networks, and ground control station (GCS) monitoring applications. The use of open-source flight controllers is increasingly being used by developers to meet research needs based on certain fields such as agriculture, disasters, and industry. With the many uses of the flight controller, the developers also have to use GCS that has been provided to monitor UAV devices. In monitoring and providing input instructions, GCS plays an important role to make it easy for users to communicate with UAV devices.

The current GCS application software that is being developed and continues to be developed is a product from Ardupilot, namely Mission Planner and products from Multiwii which also collaborates with EZ-GUI. Most of the GCS applications still run on the Personal Computer (PC) platform. The use of a desktop or PC-based GCS is considered less efficient where mobility will certainly be limited and several other factors such as the still needed power source to turn on a PC which of course does not last long. And for the use of Android-based GCS, for now, it is only used for UAV manufacturers that are closed source or cannot be modified as needed. Developers from within the country themselves still use desktop-based GCS from open-source flight controller developers who collaborate with other parties. So it's time for the development of this GCS application to be developed into the android platform [1].

One of the GCS applications that has been developed by Multiwii still has several disadvantages including still being run on a PC, the absence of location information displayed on the map, and the close communication distance between UAV devices and GCS because of using cables. The design of the User Interface (UI) in the easy-to-understand and informative GCS Android application will play a role in making it easy for users to communicate with UAV devices.

The research will make an android-based GCS application that can display information on the condition of the UAV device through an easy-to-understand UI so that it can maximize the function of the GCS as a UAV monitoring media. The GCS application created will also play a role in providing navigation information to find out the location of the UAV device on the map and be able to configure the flight controller such as setting PID.

II. GENERAL SYSTEM DESIGN

This section discusses about the general system design of this research.

A. System Diagram

The Android GCS application is designed to monitor the conditions of a UAV device. The UAV device used as an example of a vehicle for retrieving sensor information data is the quadrotor UAV model. Quadrotor consists of several parts including mechanical design, then the input device from the remote control, flight controller device, a sensor device, and an output device in the form of a motor. In this study, the focus will be on data communication systems between GCS and quadrotor that use two types of data communication, namely telemetry and UDP protocol[2][3]. quadrotor information that will be sent via communication telemetry is the location monitored with GPS sensors, roll, pitch, yaw, speed, and battery condition. For UDP communication, it will be used to send streaming video data from the camera paired to the quadrotor. After all information data is obtained GCS will also be able to give instructions to carry out a mission in the form of determining the waypoint and changing some parameters on the aircraft control system Fig. 1.

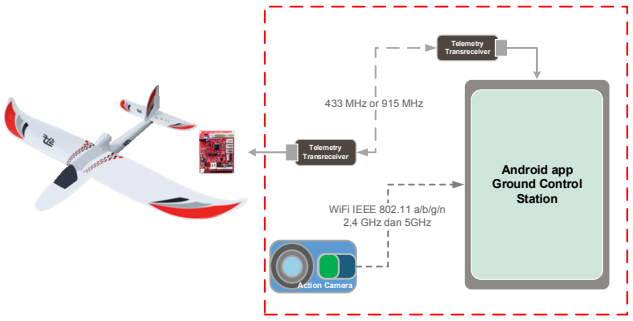

Fig. 1. System Diagram

Fig. 1 is shown in a diagram structure that shows the type of sensor connected to the flight controller. To connect the flight controller with an android smartphone, in this study using the 3DR 433 MHz telemetry module. This module was chosen because it is commonly used in the application of UAV devices and selects the frequency of 433 MHz because according to some sources and some users of the 915 MHz frequency are widely used for television antennas so that it can cause interference. As for video sending communication, in this study using an IP-Camera from Xiaomi that supports access via wireless with a 2.4 GHz frequency and the connection with GCS is to use an outdoor access point with a 2.4 GHz frequency.

B. Hardware of Unmanned Aerial Vehicle Multirotor

This research implemented testing on the Multiwii flight controller. To access Multiwii, another device is needed to be able to use the Multiwii serial protocol. The sensor data accessed is the GY-521 MPU6050, BMP180, and GPS U-Blox NEO sensors. The sensor is a common sensor that is used on UAV devices so they can know the condition of the device.

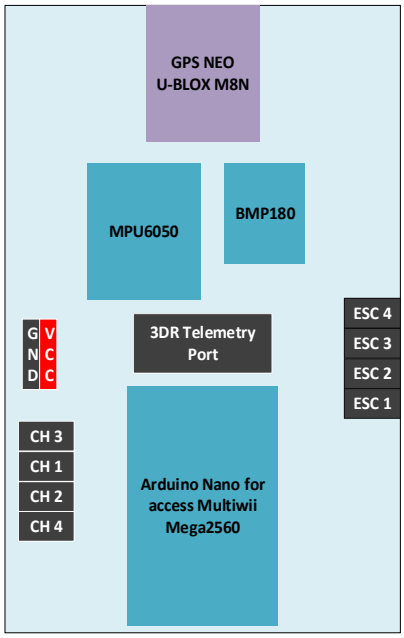

Fig. 2. Multiwii hardware shield details of quadrotor

Fig. 2, it shows the position adjustment scheme of each component device needed. This scheme is then used to create a shield device that will be connected to the Multiwii Arduino Mega2560 flight controller.

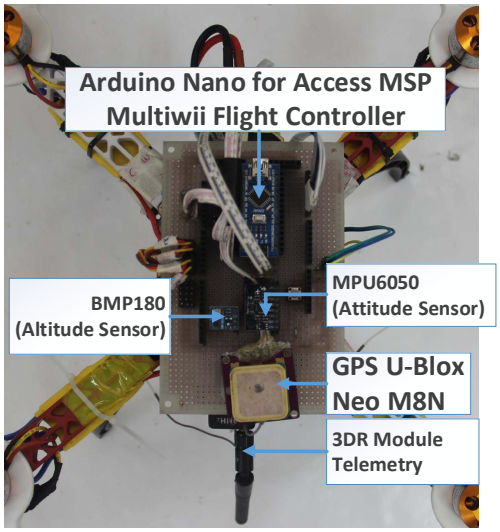

Fig. 3. Overview of the quadrotor

At Fig. 3 shows the shield device that is made according to the scheme in Fig. 2. Sensor devices that will be accessed by Multiwii arduino Mega2560 flight controller for processing and arduino Nano on the shield function to access data that has been processed by flight controller, then sent to the android GCS application through communication from 3DR radio telemetry devices.

TABLE 1
Sensor Device Spesification

Sensor GY-521 MPU6050	
Chipset	MPU-6050
Axis	3-axis accelerometer, 3-axis gyroscope
Voltage	3.3 – 5 Volt DC
Com	I2C up to 400kHz
Sensor BMP180	
Chipset	BMP180
Pressure Range	300hPa to 1100hPa (+9000m to -500m)
Voltage	1.8 – 6 Volt DC
Com	I2C
GPS U-Blox NEOM8N	
Channel	GPS, Glonass, Beidou and Galileo
Com	I2C (SDA, SCL) and UART (TX, RX)
Voltage	3.3 – 5 Volt DC
Anthena	GA25F3-60U

Flight controller functions to read existing sensor data. The sensor will show the condition of the multirotor device. The conditions referred to are rolling roll, pitch movement, yaw movement, altitude position, location based on latitude and longitude coordinates, firmware mode and monitoring of remote control inputs. Fig. 3 shows the hardware configuration when applied to a multirotor vehicle.

C. Software Development

Software development is done using Android Studio with the JAVA programming language. The application

was developed to communicate with UAV devices through a radio telemetry module device that is connected via USB-On The Go. Application access permissions that need to be activated on a smartphone are WiFi access, document storage and images and location from a smartphone device[4].

The features that are applied to the application include monitoring conditions that are displayed through UI animations, accessing the camera using the RTSP protocol, and monitoring the position and track path that has been passed by the UAV device. Development uses the default SDK from Android Studio by adding an SDK to access USB-OTG and SDK VLC to access the camera[5][6].

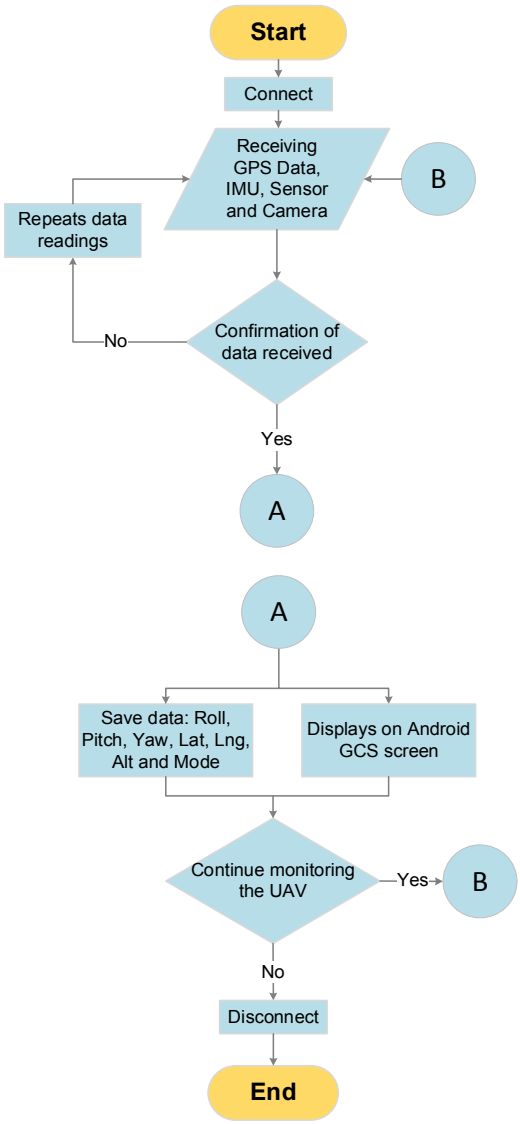

Fig. 4. Flow chart diagram on software system

The flow chart in Fig. 4. a GCS application display design is created that can provide data information in the form of a user interface[7]. The display will be in the form of an animation movement so that users can more clearly know how the UAV is.

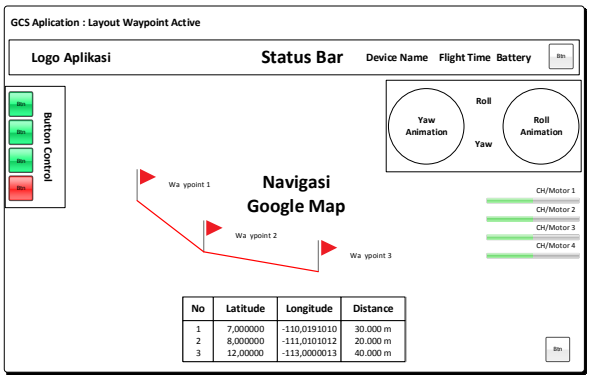

Fig. 5. GCS application UI design

In Fig. 5 a UI design is made that will be applied to the GCS application. There are several sections such as :

- Status bar: contains information about the name of the flight controller device, the active mode on the flight controller, flight time, UAV battery voltage and application logo.
- Button control: contains several buttons that function to connect applications with UAV devices using button connect and button disconnect, and there is a button to activate the waypoint menu and monitoring menu.
- GMap navigation: display a map of navigation locations[8].

III. Building Sub-System

The purpose in designing this GCS is to show the condition of the UAV. This condition is obtained from sensor data sent to the android GCS application and displayed in the form of a user-interface. The design of the application display has been made so that it is then applied to the application. The application has several features, namely monitoring the condition of the movement of the UAV, monitoring by accessing the IP-Camera and monitoring the location and location of the waypoint.

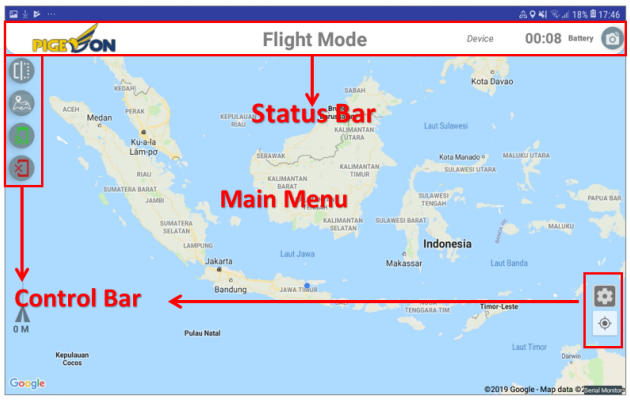

Fig. 6. Main menu of Android GCS app

In Fig. 6 there are several parts including the status bar, main menu, and control bar. The status bar contains information about communication status between the application and the UAV device, such as the type of device connected, the mode that was active on the device,

how long the communication lasted and the name of the application. Then, the main menu is the main area in the application that provides the main collection of information from data that has been received by the application from the devices. While the control bar is a set of buttons that are used to control applications including connection commands and menu switching.

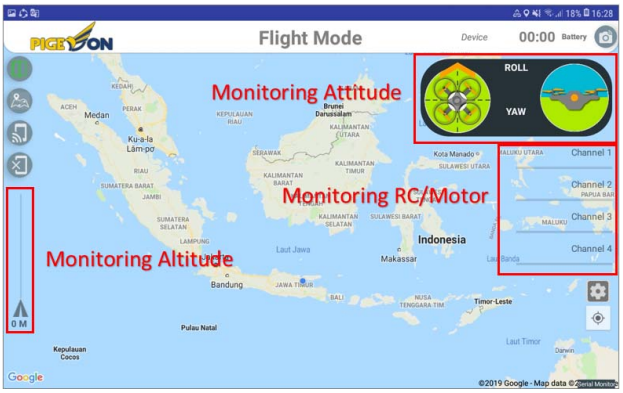

Fig. 7. Monitoring menu of Android GCS app

Fig. 7 is a monitoring menu display on the GCS application has several sections including:

- Monitoring attitude: to find out the attitude of the UAV device. Such as the tilt and direction facing the device. The data obtained in the form of angles in the range of values -180 °: 180 °. The system utilizes the *RotateAnimation* function with DegreeStart = 0 ° as the starting point which will then be reduced or added based on the latest angle.
- RC Monitoring: to find out how much the value of the signal sent by the remote control. The range of signal data obtained is 900: 2000. To show how much the change in signal value, the system uses the *progressBarAnim* function to indicate each change in value. And make sure that the remote control is functioning properly.
- Monitoring altitude: to find out the height of the UAV device. The function used is a *progressBar* that is configured with a range of 0: 100 meters. The *progressBar* display will move up when the data has values> 0 and when it has a value <0 then what changes to show the value is the *textView*.

A. *Communication System*

The communication system architecture design in this study will consist of two types of communication media namely Telemetry in the form of UART communication (TX, RX) using a frequency of 433 MHz for data transfer and a Local Area Network (LAN) in the form of 2.4 GHz wireless frequency for IP-Camera communication. The following is an illustration of the design of a communication system architecture to be applied:

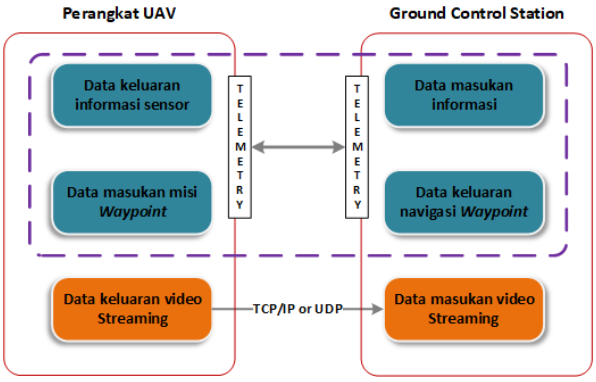

Fig. 8. Communication diagram

In Fig. 8 the communication media between GCS and UAV consists of two types, namely radio telemetry and local network. Radio telemetry will act in half-duplex, which means that both devices can act as transmitters and receivers alternately according to the needs needed by communication on the system.

B. *Data Protocol*

To be able to receive the correct data information, a protocol is needed that contains the order of data to be received. This protocol is needed to prevent errors in receipt of the information received by the GCS application before it is processed and displayed.

TABLE 2
Data Protocol Order

No	Data Order	Description
1	Header = MW/EF	ID of the flight controller used. MW for Multiwii and EF for Efalcon
2	Roll	Roll movement data -180° : 180°
3	Pitch	Pitch movement data -180° : 180°
4	Yaw	Directional data -180° : 180°
5	Latitude	Latitude Coordinate Data
6	Longitude	Longitude coordinate data
7	Altitude	Altitude data
8	Channel 1	Data RC 1000-2000
9	Channel 2	Data RC 1000-2000
10	Channel 3	Data RC 1000-2000
11	Channel 4	Data RC 1000-2000
12	Battery	Battery power
13	Mode	The active mode on the flight controller
14	Tail = \n	End of protocol

In the data transmission protocol sequence, each data is separated by a "," which is also used to calculate how much data is received by the application. The GCS application will only accept data in the order of the protocol if true the data will be sent and separated to be included in each GCS monitoring application function. Otherwise, the monitoring system will not function.

C. *Monitoring System of Android GCS*

Based on the analysis of needs that have been carried out, there are some information and data needed in designing a monitoring system, including:

a. Monitoring the Attitude of UAV Conditions
To find out the attitude of the aircraft, the data needed from the flight controller is the data obtained from the IMU sensor in the form of angular values x, y and z. from the angular value it is necessary to show the attitude of the UAV device, namely, roll the value of the x angle to determine the slope of the plane, pitch the angle value y to indicate the direction of the plane up or down, and y the value of z angle to indicate the direction of the UAV.

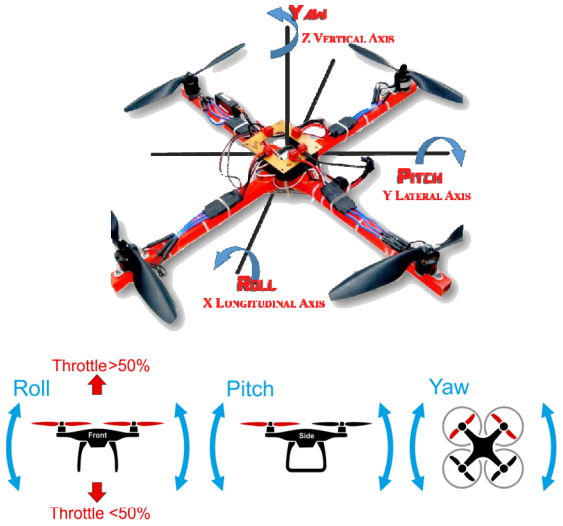

Fig. 9. Attitude of multirotor

Fig. 9 is an example of how to observe the attitude of a UAV device. Where the movement of a roll, pitch, yaw, and throttle can be described.

b. Location monitoring
This monitoring aims to determine the location of the UAV shown in GMap navigation. The data used is latitude and longitude obtained from GPS sensors. Besides that, what is needed is to know the height of the UAV by reading data from the barometer sensor.

c. Monitoring Flight Time
This monitoring aims to determine the length of time communication between UAVs and GCS applications. This is so that users also know how long the UAV flies and is connected.

d. Monitoring *Remote Control*
This monitoring aims to find out the value sent by each channel on the RC. So that the user can know each function on the RC channel used. This data is obtained from an RC receiver that is connected to the flight controller and then passed through telemetry to be sent to the GCS application.

D. Access IP-Camera

IP-Camera devices are connected with applications through 2.4 GHz frequency wireless communication on a local network. Which then to access the camera application needs to enter the IP address that has been given via the RTSP protocol. An example of writing the camera address used is rtsp: // IP-address: 8554 / unicast.

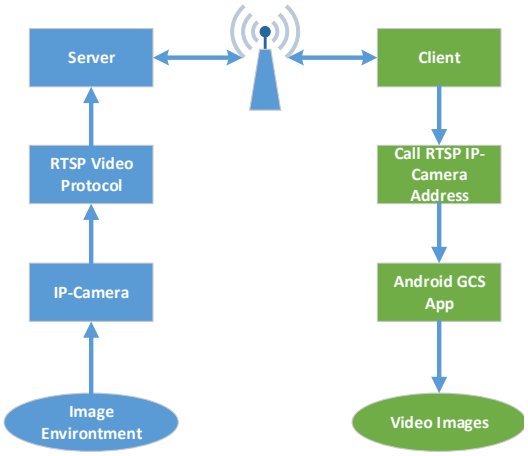

Fig. 10. Video stream system

Fig. 10 shows the flow of the video streaming system. Starting from the IP-Camera which records images from the surrounding environment then the encoding process is carried out with the RTSP protocol and then sent to the server. From the GCS application server, the client decodes by calling the RTSP protocol IP address so that the recorded environment can be displayed on the GCS application frame screen.

IV. Experimental Result

In this section, we will show the results of testing on the android GCS application. Where testing will prove that the application can communicate with the flight controller. Also, the data received will be displayed through UI animations so that users are expected to be able to understand information regarding the condition of the UAV. The data that will be displayed is the latest data update and also the speed of data updates received by the android GCS application.

Fig. 11. Testing data protocol

In Fig. 11 the first experiment, for testing data reception, the GCS application can receive data based on the data protocol arrangement which is then processed in the monitoring function to be displayed. Fig. 7 shows the results of data received from the flight controller. The amount of data sent is 13 data separated by a "," sign. The result of this test is that the GCS application can receive updated data according to what was sent. Problems will only appear if the data order is damaged so that it cannot be displayed.

After the data protocol is received correctly, then the data is displayed through the user interface on the application. Data roll and yaw are displayed using rotation

animation than for altitude data and RC monitoring using the slider bar.

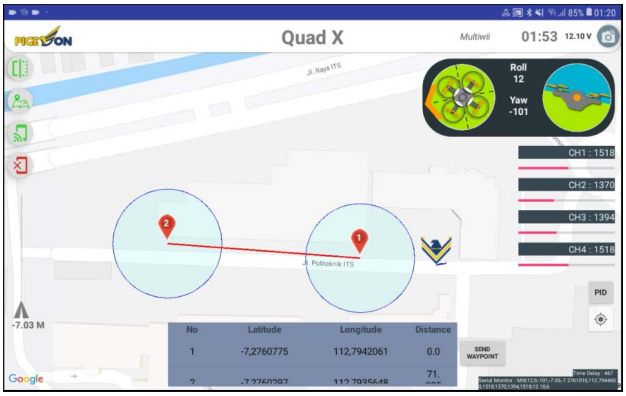

Fig. 12. User interface layout from Android GCS App for monitoring

Fig. 12 it can be noted that the movement conditions of the UAV can be seen. Where the direction of direction, slope, position and height are displayed by the user interface. Also, each data will be stored as a datalog file on the storage media from the smartphone, precisely in the document folder.

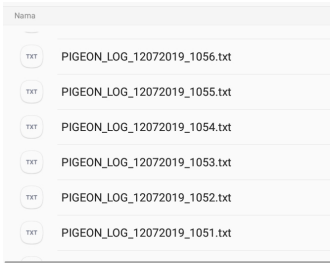

Fig. 13. Datalog on smartphone

In Fig. 13 the datalog serves to store the track record of each movement of the UAV in the form of a sensor data list. Data recorded is obtained every time a data transfer update occurs and file naming has a PIGEON_LOG_DDMMYY_HHMM arrangement in the form of a * txt file. And after one minute, a new datalog document will be created every minute. The stored data is ID, roll data, pitch data, yaw data, altitude data, latitude coordinates, longitude coordinates and data speed every time it is updated.

Fig. 14. IP-Camera access

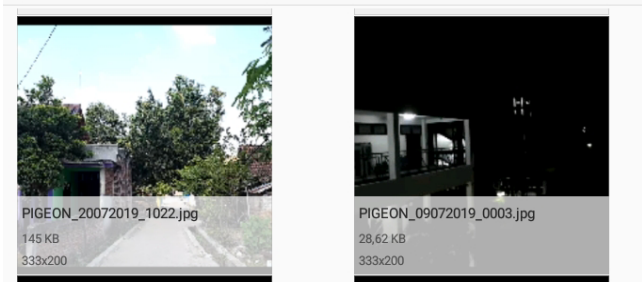

Fig. 15. IP-Camera access

In Fig. 14 and Fig. 15, it shows that applications can access IP-Camera using the RTSP protocol and H.264 format video compression. After connecting the recorded video can also be saved after pressing the capture button located next to the play button.

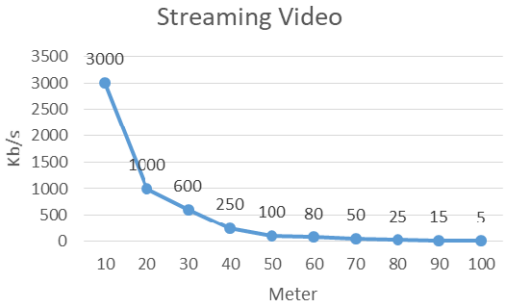

Fig. 16. Video streaming speed chart for distance

The reach of communication applications with IP-Camera as shown in Fig. 16 depends on the radius of reach of wireless connections because both devices are required to be in the same local network area. The farther the distance between devices, the faster the video streaming will be. To prove that the Android GCS application can be connected to a flight controller, there will be several sample graphs of data received by the application with the speed of updating any data changes when received by the application.

TABLE 3
Data Roll

No	Data Measuring Instrument (°)	Data Roll GCS PC (°)	Data Roll GCS App (°)	Time Delay GCS App (s)
1	15	15	15	0,108
2	30	31	31	0,116
3	45	45	45	0,233
4	60	62	62	0,305
5	75	75	75	0,236
6	90	91	91	0,228
7	105	105	105	0,11
8	120	123	123	0,335
9	135	136	136	0,375
10	150	151	151	0,238

No	Data Measuring Instrument (°)	Data Roll GCS PC (°)	Data Roll GCS App (°)	Time Delay GCS App (s)
11	-15	-16	-16	0,235
12	-30	-32	-32	0,336
13	-45	-45	-45	0,165
14	-60	-61	-61	0,282
15	-75	-76	-76	0,336
16	-90	-93	-93	0,375
17	-105	-106	-106	0,11
18	-120	-120	-120	0,212
19	-135	-136	-136	0,215
20	-150	-151	-151	0,165

The data in Table 3 shows the angle comparison of the measuring instrument with the desktop and Android GCS applications. The angle value that is read between the gauge and the GCS desktop application has a difference with the average error range obtained by 1 ° with a precision level of 0.8% and an accuracy of 98.9%. And if you look at the GCS Android application the angle values read tend to be the same as those that have been read on a desktop GCS application, this is of course due to the GCS android accessing data from the same flight controller at the same time. The possibility of differences in reading of values between the two applications will occur when the flight controller moves quickly and causes delays in receiving data on the GCS Android application so that the data appearance update is also late. And for the time being the delay in receiving and processing data on the GCS Android application has an average delay of 0.235 seconds.

TABLE 4
Data Pitch

No	Data Measuring Instrument (°)	Data Pitch GCS PC (°)	Data Pitch GCS App (°)	Time Delay GCS App (s)
1	5	5	5	0,11
2	10	8	8	0,135
3	15	16	16	0,158
4	20	20	20	0,226
5	25	26	26	0,226
6	30	30	30	0,386
7	35	37	37	0,205
8	40	41	41	0,113
9	45	46	46	0,302
10	50	52	52	0,275
11	-5	-6	-6	0,105
12	-10	-10	-10	0,345
13	-15	-16	-16	0,254
14	-20	-22	-22	0,108
15	-25	-27	-27	0,334
16	-30	-31	-31	0,226
17	-35	-36	-36	0,105
18	-40	-41	-41	0,113
19	-45	-47	-47	0,223
20	-50	-53	-53	0,31

The data in Table 4 shows a comparison of the angle of the gauge with the desktop and Android GCS applications. The angle value that is read between the gauge and the GCS desktop application has a difference with the average error range of 1.2 ° with a precision level of 0.66% and an accuracy of 97.8%. The angle values read on the GCS Android application tend to be the same as those read on the desktop GCS application, this is of course due to the android GCS accessing data from the same flight controller at the same time. The difference in data readings in the GCS application is due to delays in receiving and processing data on the application system. And for now, the delay in receiving and processing data on the GCS Android application has an average delay of 0.212 seconds.

TABLE 5
Data Pitch

No	Data Measuring Instrument (°)	Data Yaw GCS PC (°)	Data Yaw GCS App (°)	Time Delay GCS App (s)
1	10	10	11	0,112
2	20	21	21	0,229
3	30	33	33	0,335
4	40	43	43	0,17
5	50	53	53	0,35
6	60	64	64	0,21
7	70	72	72	0,273
8	80	83	83	0,15
9	90	92	92	0,372
10	100	103	103	0,367
11	-10	-11	-11	0,215
12	-20	-19	-19	0,105
13	-30	-29	-29	0,224
14	-40	-40	-39	0,165
15	-50	-51	-51	0,103
16	-60	-60	-60	0,204
17	-70	-71	-71	0,118
18	-80	-82	-82	0,198
19	-90	-92	-92	0,337

No	Data Measuring Instrument (°)	Data Yaw GCS PC (°)	Data Yaw GCS App (°)	Time Delay GCS App (s)
20	-100	-102	-102	0,217

The data in Table 5 shows the angular comparison of measuring devices with desktop and Android GCS applications. The angle value that is read between the gauge and the GCS desktop application has a difference with the average error range of 1.8 ° with a precision level of 1.16% and an accuracy of 95.6%. The angle values read on the GCS Android application tend to be the same as those read on the desktop GCS application, this is of course due to the android GCS accessing data from the same flight controller at the same time. There are some sample data from the GCS Android application with the GCS desktop application, this is influenced by the delay in receiving and processing data when displayed on the GCS Android application. And for the time being the delay in receiving and processing data on the GCS Android application has an average delay of 0.227 seconds.

V. CONCLUSION

Applications can be connected to the Multiwii flight controller as evidenced by the data acquisition and sending of both devices in half duplex. To access the telemetry module, an Android smartphone device that has USB Host and On The Go features is needed. From the test data obtained by GCS has an accuracy of 96.12% with a precision level of 0.7% so it can be concluded that the GCS Android application can read accurately. In addition, the application can display navigation by showing the location point based on latitude data and longitude data. Whereas for IP-Camera which is integrated with the GCS application through the local WiFi network, it still has a low coverage where in the range of 0-100 meters. And the directional angle transmits the access point signal by 60 °.

REFERENCES

[1] Hayajneh Mohammad, Melega Marco, Marconi Lorenzo, *"Design of Autonomous Smarthpone Based Quadrotor, and Implementation of Navigation and Guidance System,"* The Hashemite University, Zarqa, University of Bologna, Bologna, 2018.

[2] Espen Oland, Rune Schlanbusch, Raymond Kristiansen, *"Underactuated Waypoint Tracking of a Fixed-Wing UAV*,"* Narvik University College, Narvik, Norway, 2013.

[3] Martiningtyas Handayani, Ariesta & Nur Rifa'i, Isnan. (2018). *"Sistem Ground Control Station Berbasis Mobile Untuk Pengamatan Dan Pengendalian UAV"*. Jurnal Nasional Teknologi Terapan (JNTT). 2. 121. 10.22146/jntt.39204.

[4] Yasenjiang. Aimaiti, Ubaid M. Al-Saggaf , Khalid Munawar, 2014, Android Based Control Softwarefor Small Unmanned AerialVehicles, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 03, Issue 04 (April 2014),

[5] Z. Lu, F. Nagata and K. Watanabe, "Development of iOS application handlers for quadrotor UAV remote control and monitoring," *2017 IEEE International Conference on Mechatronics and Automation (ICMA)*, Takamatsu, 2017, pp. 513-518. doi: 10.1109/ICMA.2017.8015870

[6] Aga Nugroho, Fredy & Sumiharto, Raden & Muhammad Hujja, Roghib. (2018). Pengembangan Sistem Ground Control Station Berbasis Internet Webserver pada Pesawat Tanpa Awak. IJEIS (Indonesian Journal of Electronics and Instrumentation Systems). 8. 1. 10.22146/ijeis.30126.

[7] Farghani Akbar Ali, Sumiharto Raden, Wibowo Bekti Set*Yaw*an, *"Purwarupa Ground Control Station Untuk Pengamatan dan Pengendalian Unmanned Aerial Vehicle Bersayap Tetap,"* Universitas Gajah Mada, Yogyakarta, 2013.

[8] M. L. Baidhowi, F. Ardilla and M. M. Bachtiar, "Build and evaluate library on naisduino B board for implementation on mobile robot," 2017 International Electronics Symposium on Engineering Technology and Applications (IES-ETA), Surabaya, 2017, pp. 263-269. doi: 10.1109/ELECSYM.2017.8240414