

Pengembangan Sistem *Ground Control Station* Berbasis Internet Webservice pada Pesawat Tanpa Awak

Fredy Aga Nugroho^{*1}, Raden Sumiharto², Roghib Muhammad Hujja³

¹Program Studi Elektronika dan Instrumentasi, FMIPA UGM, Yogyakarta

^{2,3}Departemen Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta

e-mail: ^{*1}fredy.aga.n@mail.ugm.ac.id, ²r_sumiharto@ugm.ac.id, ³roghib.muh@ugm.ac.id

Abstrak

Dalam operasional pesawat tanpa awak, tugas *ground control station* sebagai stasiun pemantauan dan komando sehingga operator di darat dapat mengirimkan perintah misi, mengawasi jalannya misi tersebut dan memantau kondisi UAV selama misi berlangsung. Sehingga diperlukan sistem GCS yang mampu terhubung dengan UAV tanpa terbatas dengan jangkauan pemancar kendali.

Penelitian ini mengembangkan sistem GCS menggunakan jaringan internet dan berbasis web server. sistem terdiri dari dua unit, yaitu unit terbang dan unit GCS. Unit terbang terdiri dari Raspberry pi, modem, webcam, modul ADAHRS dan quadrotor dengan controller MultiWii. pada unit GCS terdiri dari Raspberry pi yang tersambung pada jaringan internet dengan kecepatan download 10Mbps dan upload 1.5Mbps.

Sistem GCS ini dapat menampilkan kondisi pesawat, video streaming dan melakukan perintah kendali. Konfigurasi video streaming dengan resolusi 240x144 pixel, bitrate maksimal 256kbps dan framerate 5 fps menghasilkan delay time tidak lebih dari satu detik. Konfigurasi ini berjalan pada kecepatan upload 1.1 Mbps dengan prosentase kompresi bitrate 93.83%. Data kondisi pesawat yang dikirim ke GCS optimal apabila bandwidth internet melebihi bitrate video streaming yang digunakan pada sistem.

Kata kunci—UAV, GCS, Raspberry Pi, Video streaming, FFMPEG, Web Server

Abstract

In unmanned aerial operations, the ground control station duties as a monitoring and command station so that operators on land can send mission orders, monitor the mission's course and monitor the condition of the UAV during the mission. It is necessary to have a GCS system capable of connecting with UAV that not limited with control transmitter range.

This research develops GCS system using internet network and web server based. the system consists of two units, namely flying units and GCS units. The flying unit consists of Raspberry pi, modem, webcam, ADAHRS module and quadrotor with MultiWii controller. on the GCS unit consists of Raspberry pi connected on the internet network with 10Mbps download speed and 1.5Mbps upload.

The GCS system can display aircraft conditions, stream video and perform command controls. Configure streaming video for delay time of no more than one second with 240x144 pixel resolution, 256kbps maximum bitrate and 5 fps framerate. This configuration runs at a 1.1 Mbps upload speed with a percentage of 93.83% bitrate compression. Aircraft condition data sent to GCS is optimal if internet bandwidth exceeds the bitrate of streaming video used on the system

Keywords—UAV, GCS, Raspberry Pi, Video streaming, FFMPEG, Web Server

1. PENDAHULUAN

Dalam Beberapa tahun terakhir, dunia kedirgantaraan Indonesia mengalami kemajuan yang cukup pesat khususnya pada teknologi pesawat tanpa awak. Pesawat tanpa awak atau yang sering disebut sebagai Unmanned Aerial Vehicle (UAV) merupakan sebuah wahana yang dapat terbang tanpa adanya awak atau pilot. Dalam pengoperasiannya, pesawat tanpa awak dikendalikan langsung oleh operator melalui jaringan komunikasi tertentu seperti remote control atau secara otomatis melalui perangkat yang telah ditanamkan program sebelumnya.

Penelitian dan pengembangan sistem UAV sangat luas, apabila dikategorikan akan menjadi beberapa sub bagian *seperti air vehicle, ground control station, payloads, data link dan support equipment* [1]. Dalam operasional UAV, tugas ground control station sebagai stasiun pemantauan dan komando sehingga operator di darat dapat mengirimkan perintah misi, mengawasi jalannya misi tersebut dan memantau kondisi UAV selama misi berlangsung. Ground control station (GCS) dalam melakukan pemantauan kondisi UAV bergantung pada kemampuan komunikasi antara GCS dengan UAV.

Pada penelitian sebelumnya yang telah dilakukan[2,3], ground control station dapat melakukan pemantauan kondisi UAV pada jarak terbatas. Hal ini dikarenakan komunikasi antara GCS dengan UAV memanfaatkan jaringan lokal. Sehingga area cakupannya menjadi terbatas dan GCS tidak dapat diakses oleh operator yang berada di luar area jaringan lokal. Ground control station yang hanya bisa memantau kondisi dan mengendalikan UAV pada jarak terbatas saja tentunya tidak baik, karena untuk melakukan pengawasan diperlukan ground control station yang dapat memberikan pantauan kondisi dan kendali UAV secara terus menerus.

Sehingga diperlukan sistem ground control station yang mampu terhubung dengan UAV tanpa terbatas dengan jangkauan pemancar kendali. Dengan sistem ini di harapkan dapat memberikan kemudahan dalam pengoperasian pesawat tanpa awak.

2. METODE PENELITIAN

2.1 Analisis Kebutuhan

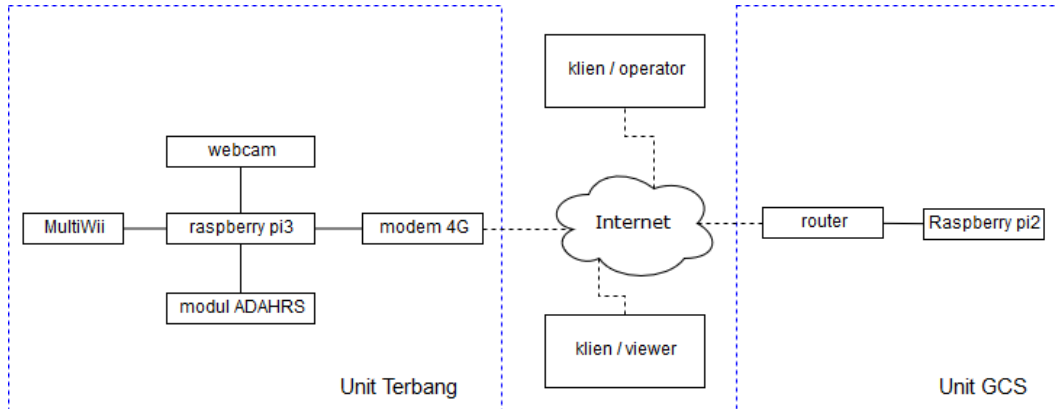
Salah satu kebutuhan pesawat tanpa adalah stasiun pemantauan. Stasiun pemantauan tersebut digunakan untuk memantau keadaan sekitar saat pesawat tanpa awak beroperasi. Untuk itu dibutuhkan suatu sistem stasiun pemantauan yang memenuhi spesifikasi sebagai berikut:

- Panel instrumen penerbangan yang dapat menampilkan keadaan UAV untuk mempermudah pengamatan.
- Panel peta penerbangan yang menampilkan posisi UAV saat penerbangan.
- Panel tampilan visual yang menampilkan keadaan sekitar UAV.
- Penampil data kondisi pesawat pada saat UAV beroperasi.
- Fungsi perintah kendali yang dapat mengendalikan UAV.

2.2 Deskripsi Sistem

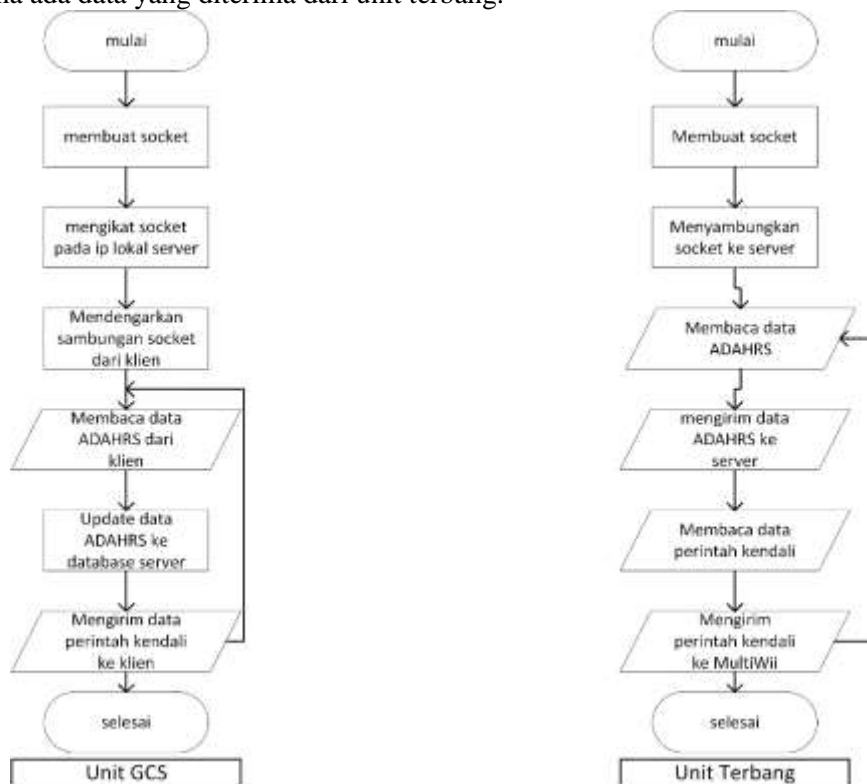
Rancangan keseluruhan sistem dibagi menjadi dua bagian, yaitu rancangan unit terbang dan rancangan unit GCS yang ditunjukkan pada Gambar 1. Unit terbang merupakan unit yang terdapat pada pesawat tanpa awak terdiri dari ADAHRS (*Air Data/ Attitude/ Heading Reference System*) sebagai penyedia data sensor[4], *webcam* digunakan untuk merekam tampilan visual keadaan UAV, SBC (*single board computer*) Raspberry Pi sebagai unit pemroses dan *modem* sebagai modul penghubung ke unit GCS melalui jaringan internet. Modul ADAHRS memiliki fungsi mendeteksi kecepatan laju, ketinggian, posisi dan sikap UAV dengan sensor yang sudah terintegrasi. Raspberry Pi akan menerima data dari modul ADAHRS dan *webcam*, lalu mengirimkan data tersebut ke unit GCS melalui jaringan internet dengan *modem*. Unit GCS

mampu menerima dan menginterpretasikan data dari unit terbang pada antarmuka aplikasi. Paket data diterima, dimasukkan dalam *database*, kemudian dilakukan *parsing* data dan didistribusikan ke setiap panel antarmuka yang membutuhkan.



Gambar 1 Rancangan Sistem Keseluruhan

Proses awal agar kedua unit bisa saling terhubung dengan cara membuat sambungan socket anatar kedua unit. Seperti yang ditunjukkan pada Gambar 2, pada unit GCS diawali dengan membuat socket, kemudian mengikat socket pada ip lokal unit GCS. Setelah itu mendengarkan sambungan dari unit terbang, ketika mendeteksi adanya sambungan socket dari unit terbang kemudian membaca kiriman data ADAHRS dan di-update pada database. Kemudian unit GCS mengirimkan database kendali ke unit terbang. Proses ini secara rutin berulang selama ada data yang diterima dari unit terbang.



Gambar 2 Diagram alir komunikasi socket unit GCS dan unit terbang

Proses alir komunikasi data pada unit terbang di mulai dengan membuat socket kemudian melakukan sambungan ke unit GCS. Setelah berhasil, raspberry pi membaca data dari

modul ADAHRS yang berisikan kondisi UAV. Kemudian data ini dikirimkan ke unit GCS, setelah data terkirim akan menerima respon berupa kiriman data kendali dari unit GCS. Data kendali tersebut kemudian dikirimkan ke MultiWii.

2. 3 Rancangan Perangkat Keras

Perangkat keras pada sistem ini terbagi atas 2 bagian, yaitu perangkat keras pada unit terbang dan unit GCS.

2. 3.1 Rancangan Perangkat Keras Unit Terbang

Pada pesawat tanpa awak akan dipasang perangkat keras, antara lain modul ADAHRS, *webcam*, Raspberry Pi3 dan *modem*. Modul ADAHRS ini memiliki dua sensor yang digunakan untuk menentukan sikap pesawat, yaitu GPS (*Global Positioning Unit*), IMU (*Inertial Moment Unit*). GPS berfungsi sebagai pendeteksi ketinggian, kecepatan dan lokasi UAV berdasarkan lokasi geografis yang mengacu pada garis lintang dan garis bujur bumi. IMU memiliki fungsi untuk mendeteksi ketinggian, kecepatan dan rotasi posisi UAV terhadap sumbu x, y, z yang diinterpretasikan sebagai sudut *roll/miring*, *pitch/mengangguk* dan *yaw/menoleh*.

Webcam digunakan sebagai masukan video *streaming*. *Webcam* yang digunakan memiliki resolusi 5 MP dan mampu mengolah video dengan kemampuan maksimal 30 *framepersecond* (fps). Selanjutnya *Webcam* tersebut dihubungkan pada Raspberry Pi dengan menggunakan koneksi USB. Raspberry Pi digunakan sebagai pengolah data video *streaming* yang didapatkan dari *Webcam*[5]. Kemudian mengirimkan data video *streaming* tersebut ke server melalui jaringan internet. Agar sistem dapat terkoneksi ke internet, maka digunakan *modem* yang dipasang pada Raspberry Pi.

Sistem pengiriman perintah kendali dari GCS ke UAV juga dilakukan menggunakan komunikasi jalur internet. Ketika mode kendali aktif, setiap kali klien melakukan kendali akan mengubah *database* kendali pada server, dan unit terbang akan memperbaharui datanya berdasarkan *database* kendali pada server[6]. *Database* kendali ini menjadi acuan bagi unit terbang untuk melakukan tanggapan.

2. 3.2 Rancangan Perangkat Keras Unit GCS

Pada unit GCS terdiri atas Raspberry pi2 yang dijadikan sebagai *server*. Raspberry pi tersebut di hubungkan dengan *router* agar mendapatkan koneksi internet dari ISP. Untuk ISP menggunakan layanan INDIHOME dengan kecepatan *download* 10 Mbps dan *upload* 1.5 Mbps.

2. 4 Rancangan Perangkat Lunak

Pada rancangan perangkat lunak juga di bagi menjadi 2 bagian utama, yaitu perangkat lunak pada unit terbang dan perangkat lunak pada unit GCS.

2. 4.1 Rancangan Perangkat Lunak Unit Terbang

Rancangan perangkat lunak pada unit terbang dibagi menjadi tiga macam fungsi utama, yaitu perangkat lunak yang digunakan untuk mengirimkan data penerbangan (kondisi dan sikap pesawat) dari UAV ke GCS, pengiriman data video *streaming* dari *webcam* yang terpasang pada UAV ke GCS dan yang terakhir mode penerimaan data perintah kendali UAV dari GCS.

Pengiriman data penerbangan berupa paket data ADAHRS yang telah diproses oleh Raspberry Pi kemudian dikirim ke *server* melalui jaringan internet. Proses pengiriman data dari ADAHRS ke Raspberry menggunakan komunikasi serial, sedangkan proses pengiriman paket data ADAHRS dari Raspberry ke *server* GCS menggunakan protokol TCP (*Transmission Control Protocol*)[7]. Paket data tersebut nantinya tersimpan dalam *database mySQL* pada server untuk ditampilkan pada antarmuka *ground control station*.

Untuk melakukan pengiriman data video *streaming* ke *server* di gunakan program *framework* FFMPEG[8]. Penggunaan program agar dapat terhubung dengan *server* dengan cara menambahkan alamat ip public server[9].

2. 4.1 Rancangan Perangkat Lunak Unit GCS

Pada unit GCS agar dapat berperan sebagai server maka di berikan program Apache dan MySQL untuk pengelolaan *database*. Untuk penerimaan data video dari unit terbang dengan menjalankan program FFServer. FFServer merupakan bagian dari program FFmpeg yang berfungsi sebagai penerima data video *streaming*.

Untuk penerimaan data ADAHRS dan pengiriman data kendali dilakukan dengan menggunakan *socket programming*. Unit GCS akan menerima data ADAHRS dan mengirimkan data kendali melalui pemrograman *socket*. Data ADAHRS yang di terima disimpan pada *database* MySQL[10]. Dan unit GCS akan mengirimkan data kendali dengan mengambil data dari *database* kendali MySQL.

2. 5 Rancangan Protokol Kendali dan Video Straming

Pada mode penerimaan perintah kendali seperti ditunjukkan Gambar 3, dapat digunakan ketika sudah diaktifkan dengan cara memasukkan *user* dan *password* pada *landing page* GCS. Setelah berhasil *login*, mode kendali pada antarmuka GCS sudah aktif. Apabila operator menekan tombol tertentu pada *keyboard* akan mengubah *database* kendali yang akan membuat unit terbang melakukan tanggapan aksi. Raspberry mengambil *database* kendali pada *server* menggunakan *socket programming* pada port khusus.



Gambar 3 Diagram fungsi kendali

Untuk proses video *streaming* digunakan protocol http, hal ini dikarenakan program FFServer dapat berjalan dengan menerima data yang menggunakan protocol http. Untuk mengakses video *streaming* dengan mengakses alamat ip dan *port* yang sudah ditentukan untuk proses penampilan video *streaming*. Untuk diagram proses video *streaming* seperti pada Gambar 4.



Gambar 4 Diagram proses video *streaming*

2. 6 Rancangan Antarmuka GCS

Rancangan tampilan antarmuka pada *ground control station* ditunjukkan oleh Gambar 5. Antarmuka ini memiliki 3 panel utama yaitu panel *flight instrument* dan data ADAHRS, panel peta penerbangan dan panel penampil video *streaming*. Panel *flight instrument*, data ADAHRS, panel peta penerbangan mendapatkan data dari mengakses *database* *mysql* yang ter-*update* data ADAHRS. Panel video *streaming* mendengarkan IP *public server* dengan *port* yang ditentukan sebelumnya.



Gambar 5 Rancangan Antarmuka *Ground Control Station*

3. HASIL DAN PEMBAHASAN

3.1 Hasil Pengujian RTT, Download dan Upload dengan ISP yang berbeda

Pengujian ini dilakukan untuk mengetahui ISP mana yang memiliki hasil paling stabil untuk digunakan. Pengujian dengan tiga ISP yang berbeda yaitu B, C dan A. Parameter yang diujikan antara lain RTT, kecepatan download/upload pada jaringan 4G maupun 3G. Setiap parameter dilakukan pengambilan data minimal 10 kali dan lokasi pengujian dilakukan pada koordinat latitude -7.7576599 dan longitude 110.3719617. Pengujian dilakukan dengan jeda setiap 5 menit untuk masing – masing ISP.

RTT (round-trip time) adalah waktu yang diperlukan untuk sebuah sinyal dari pengirim sampai ke penerima dan kembali lagi ke pengirim, sehingga semakin kecil nilai RTT akan semakin baik. Untuk mengetahui nilai RTT dilakukan dengan uji ping.

Tabel 1 Rata – rata nilai *ping*, *download* dan *upload* pada 3 ISP

ISP	3G			4G		
	<i>Ping(ms)</i>	<i>Download</i>	<i>Upload</i>	<i>Ping(ms)</i>	<i>Download</i>	<i>Upload</i>
A	71,14	1,5Mbps	2,94Mbps	40,11	1,48Mbps	9,45Mbps
B	219,98	0,58Mbps	0,7Mbps	66,86	0,92Mbps	0,94Mbps
C	96,24	1,21Mbps	1,39Mbps	63,93	1,53Mbps	8,75Mbps

Dari Tabel 1 didapatkan hasil jika ISP A memiliki hasil paling stabil pada kedua jaringan, sehingga akan digunakan untuk pengujian selanjutnya.

3.2 Hasil Pengujian Pengiriman Data Kondisi Pesawat

Pengujian pengiriman data kondisi pesawat dilakukan antara unit terbang dan unit GCS yang saling terhubung. Pengiriman data melalui *socket* menggunakan protocol TCP. Protokol ini memiliki kelebihan yaitu pada proses pembuatan koneksi TCP yang biasa disebut *three way handshake*.

Proses ini dengan cara *host* pertama mengirimkan sinyal kepada *host* kedua. jika diterima, *host* kedua mengirimkan sinyal balik kepada *host* pertama. Apabila sinyal tersebut diterima *host* pertama, pertukaran data kemudian berlangsung. Proses ini juga berlangsung pada saat mengakhiri koneksi yang dibuat. Sehingga menjamin data yang dikirim diterima dengan baik.

Pengujian pengiriman paket data ini menggunakan modul ADAHRS yang di program untuk mengeluarkan data penerbangan melalui serial *port* USB. Kemudian disambungkan ke raspberry pi pada unit terbang dan dilakukan pengiriman data melalui *socket* ke unit GCS. Pengujian ini dimaksudkan untuk menguji kemampuan GCS dalam menerima paket data yang dikirim unit terbang.

Pengujian dilakukan pada beberapa lokasi area yang berbeda. Pengujian pertama pada area berjaringan 4G dengan koordinat Lat : -7.75761 dan Long : 110.37200 atau area pemukiman padat penduduk. Untuk *ping* 40,11 ms, kecepatan *download* 1,48Mbps dan *upload* 9,45Mbps.

Tabel 2 Hasil pengiriman paket data ADAHR pada area padat penduduk

No	Dikirim	Diterima	Waktu(s)	Paket/dtk	Sukses	Loss
1	182	180	10,4	17	98,90%	1,10%
2	182	179	10,4	17	98,35%	1,65%
3	178	177	10,3	17	99,44%	0,56%
4	180	179	10,4	17	99,44%	0,56%
5	184	183	10,5	17	99,46%	0,54%
Rata – rata				17	99,12%	0,88%

Tampak pada Tabel 2, GCS rata - rata menerima sebanyak 17 paket data per detik. Untuk tingkat keberhasilan data yang dapat diterima secara utuh pada sistem gcs rata – rata 99,12%.

Pada pengujian kedua dilakukan pada area berjaringan 4G dengan koordinat Lat : -7.77130 | Long : 110.37688 atau di Graha Sabha Pramana UGM. Untuk hasil ping pada lokasi tersebut sebesar 43,12 ms, *download* 1,48 Mbps dan *upload* 6,98 Mbps. Dari Tabel 3 dapat

dilihat rata – rata paket ADAHRS yang terkirim 16 paket data per detik dengan prosentase keberhasilan diterima sebesar 98,95%.

Tabel 3 Hasil pengiriman paket data ADAHR pada area GSP UGM

No	Dikirim	Diterima	Waktu(s)	Paket/dtk	Sukses	Loss
1	167	164	10,2	16	98,20%	1,80%
2	169	169	10,4	16	100%	0,00%
3	174	172	10,5	16	98,85%	1,15%
4	170	169	10,5	16	99,41%	0,59%
5	174	171	10,6	16	98,28%	1,72%
Rata – rata				16	98,95%	1,05%

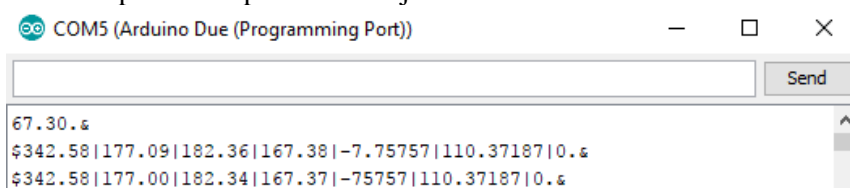
Pada pengujian ketiga dilakukan pada area berjaringan 3G dengan koordinat Lat : - 7.70121 | Long : 110.79819 atau di area pedesaan. Untuk hasil *ping* pada lokasi tersebut sebesar 47,74 ms, *download* 1,3 Mbps dan *upload* 1,1 Mbps. Rata – rata paket ADAHRS yang terkirim 16 paket data per detik dengan prosentase keberhasilan diterima sebesar 98,10% seperti pada Tabel 4.

Tabel 4 Hasil pengiriman paket data ADAHR pada area pedesaan

No	Dikirim	Diterima	Waktu(s)	Paket/dtk	Sukses	Loss
1	161	157	10,5	15	97,52%	2,48%
2	155	152	10,6	14	98,06%	1,94%
3	157	155	10,5	15	98,73%	1,27%
4	123	120	10,2	12	97,56%	2,44%
5	147	145	10,5	14	98,64%	1,36%
Rata – rata				14	98,10%	1,90%

Rata – rata jumlah paket data yang dapat diterima secara utuh oleh GCS dari ketiga lokasi pengujian sebesar 16 paket per detik. Tiap 1 paket memiliki ukuran sebesar 60 byte, sehingga dalam satu detik dihasilkan data sebesar 7680 bit atau 7,68 Kbps. Data kondisi pesawat yang dikirim ke GCS optimal apabila *bandwidth internet (upload)* melebihi *bitrate video streaming* sebesar 7,68 Kbps.

Adanya perbedaan jumlah data yang terkirim dengan yang diterima karena pada program *socket* penerima melakukan *filtering*. Apabila jumlah data yang di pecah tidak sejumlah 9 data, data tidak akan dimasukkan ke dalam *database*. Hal ini dikarenakan awal pengiriman data serial dari modul ADAHRS ke raspberry pada unit terbang ada data yang tidak sempurna/utuh. Terlihat pada Gambar 5 pada awal paket data terjadi error.



Gambar 5 Tampilan data ADAHRS pada Serial Monitor

Hal ini dikarenakan komunikasi serial merupakan transmisi *asynchronous*. Pada transmisi ini, sebelum terjadi komunikasi tidak diadakan sinkronisasi *clock* antara pengirim dan penerima. Sehingga sering terjadi kehilangan data pada awal pengiriman.

3.3 Hasil Pengujian Video Streaming

Pengujian *video streaming* dilakukan untuk mengetahui konfigurasi optimum untuk panel *video streaming* pada sistem GCS. Pengujian dengan melakukan proses *video streaming* pada beberapa lokasi. Dalam pengujian ini sebagai acuan adalah *delay time* pada *video streaming* yang tidak lebih dari satu detik.

Karena *delay time* yang ingin dicapai tidak lebih dari satu detik atau kita anggap satu detik, maka yang menentukan adalah kecepatan *upload* jaringan dari unit terbang ke unit GCS. Semisal kecepatan *upload* 10Mbps, *bitrate video streaming* tidak boleh melebihi kecepatan *upload*. Untuk perhitungan ditunjukkan pada Rumus 1

$$\text{Video bitrate} = \text{resolusi} \times \text{kedalaman warna} \times \text{framerate} \quad (1)$$

Webcam yang digunakan memiliki kedalaman warna *true color*. *True color* berarti memiliki warna RGB. Tingkat kedalaman warna pada RGB adalah 1 *byte* (8 bit) untuk setiap komponen warnanya. Karena RGB terdiri dari 3 komponen warna (RED, GREEN, BLUE), nilai RGB untuk satu *pixel* sebesar 3 *byte* atau 24 bit. Dari Rumus 6.1 dapat dihitung *bitrate* untuk resolusi 320x240 *pixel* dan *framerate* 5 fps yaitu sebesar 9,22 Mbps. Sehingga untuk kecepatan 10Mbps, resolusi tersebut masih mampu untuk menghasilkan *delay time* tidak lebih dari satu detik.

Tabel 5 Hasil perhitungan *bitrate* video pada beberapa resolusi

Resolusi (<i>pixel</i>)	Kedalaman warna	<i>Framerate</i>	<i>Bitrate</i>
240x144	24 bit	5 fps	4,15 Mbps
320x240	24 bit	5 fps	9,22 Mbps
640x480	24 bit	5 fps	36,86 Mbps
800x600	24 bit	5 fps	57,6 Mbps
1024x600	24 bit	5 fps	73,73 Mbps

Pada Tabel 5 dapat dilihat bahwa semakin besar resolusi semakin besar pula *bitrate* yang dihasilkan. Semakin besar *bitrate* semakin baik pula kualitas video yang dihasilkan. Namun dengan *bitrate* yang semakin besar, kebutuhan *bandwidth* jaringan juga semakin besar. Oleh karena itu diperlukan kompresi dalam proses video *streaming*.

Pada pengujian video *streaming* ini dilakukan pada tiga lokasi yang berbeda, pengujian pertama pada area berjaringan 4G dengan koordinat Lat : -7.75761 dan Long : 110.37200 atau area pemukiman padat penduduk. Pada lokasi pengujian ini memiliki nilai ping 40,11 ms dan kecepatan *upload* sebesar 9,45Mbps. Pengujian dilakukan dengan memvariasi resolusi, sedangkan *bitrate* maksimum dikonfigurasi pada nilai 1024kbps.

Untuk hasil *delay time* rata-rata pada resolusi 1024x600 5,13 detik, resolusi 800x600 3,77 detik, resolusi 640x480 2,55 detik dan resolusi 320x240 0,51 detik. Sedangkan untuk *framerate* semua menghasilkan 5 fps. Lebih lengkapnya dapat dilihat pada Tabel 6.

Tabel 6 Data hasil kompresi video *streaming* pada area padat penduduk

Resolusi	<i>Bitrate</i> webcam	<i>Bitrate</i> maks	% kompresi	<i>Delay time</i>
320x240	9,22 Mbps	1024 kbps	88,89%	0,51 s
640x480	36,86 Mbps	1024 kbps	97,22%	2,55 s
800x600	57,6 Mbps	1024 kbps	98,22%	3,77 s
1024x600	73,73 Mbps	1024 kbps	98,61%	5,13 s

Dapat dilihat bahwa resolusi mempengaruhi besarnya kompresi yang dilakukan serta berimbas pada hasil *delay time* video *streaming*. Semakin besar resolusi video *streaming* maka semakin besar *delay time* yang dihasilkan. Hal ini dikarenakan *encoder* dalam memproses/mengompresi data yang lebih besar akan membutuhkan waktu lebih lama, sehingga menyebabkan *delay time* bertambah.

Pada pengujian kedua dilakukan pada area berjaringan 4G dengan koordinat Lat : -7.77130 | Long : 110.37688 atau di Graha Sabha Pramana UGM. Untuk pengujian dilakukan dengan resolusi 320x240 saja. Untuk hasil *ping* pada lokasi ini sebesar 43,12 ms dan *upload* 6,98 Mbps. Untuk *delay time* video *streaming* dengan maksimal *bitrate* 1024kbps rata – rata 1,8 detik dan *framerate* 5 fps. Karena *delay time* masih lebih dari satu detik, dicoba video *streaming* dengan maksimal *bitrate* 256kbps. Pada pengujian *bitrate* maksimal 256kbps ini didapatkan *delay time* rata – rata sebesar 0,8 detik dan *framerate* 5 fps.

Tabel 7 Data hasil kompresi video *streaming* pada area GSP

Resolusi	<i>Bitrate</i> webcam	<i>Bitrate</i> maks	% kompresi	<i>Delay time</i>
320x240	9,22 Mbps	1024 kbps	88,89%	1,8 s
320x240	9,22 Mbps	256 kbps	97,22%	0,8 s

Bitrate maksimal merupakan banyaknya data dalam satuan bit yang dapat diproses oleh *encoder* dalam satu detik. Semakin tinggi nilai *bitrate* video akan membuat semakin banyak

data yang diproses, berimbas pada *delay time video streaming*. Semakin rendah nilai *bitrate* juga akan menurunkan kualitas video *streaming*. Terlihat pada Tabel 7 pada resolusi 320x240 dengan *bitrate* maksimal diturunkan menjadi 256kbps membuat prosentase kompresi meningkat menjadi 97,22%. Meskipun prosentase kompresi meningkat, tetapi dapat menurunkan *delay time video streaming* menjadi tidak lebih dari satu detik.

Pada pengujian ketiga dilakukan pada area berjarangan 3G dengan koordinat Lat : - 7.70121 | Long : 110.79819 atau di area pedesaan. Pada lokasi ini memiliki nilai *ping* 47,74 ms dan kecepatan *upload* 1,1Mbps. Dengan kecepatan *upload* tersebut, dilakukan pengujian dengan resolusi 320x240 *pixel* dengan maksimal *bitrate* 1024kbps. Dari hasil pengujian didapatkan hasil *delay time video streaming* rata – rata 2,5 detik dan *framerate* 5 fps. Kemudian dilakukan lagi pengujian dengan *bitrate* maksimal 256kbps, didapatkan *delay time* rata – rata sebesar 2,11 detik dan *framerate* 5 fps.

Karena *delay time* masih lebih dari satu detik, dilakukan pengujian lagi dengan menurunkan resolusi dan *bitrate* maksimal video. Resolusi yang digunakan 240x144 *pixel* dan *bitrate* maksimal 256kbps. Dari pengujian ini didapatkan *delay time* rata – rata sebesar 0,84 detik dan *framerate* 5 fps.

Tabel 8 Data hasil kompresi video *streaming* pada area pedesaan

Resolusi	<i>Bitrate</i> webcam	<i>Bitrate</i> maks	% kompresi	<i>Delay time</i>
320x240	9,22 Mbps	1024 kbps	88,89%	2,5 s
320x240	9,22 Mbps	256 kbps	97,22%	2,11 s
240x144	4,15 Mbps	256 kbps	93,83%	0,84 s

Dari pengujian lokasi ketiga ini dapat dilihat meskipun resolusi 240x144 memiliki *bitrate* asli sebesar 4,15 Mbps, masih mampu untuk menghasilkan *delay time video streaming* tidak lebih dari satu detik. Hal ini dikarenakan adanya kompresi video dengan mengkonfigurasi *bitrate* maksimal menjadi 256kbps. Tampak pada Tabel 8 prosentase kompresinya sebesar 93,83%.

Pada pengujian lokasi kedua dan ketiga sama – sama dilakukan pengujian menggunakan resolusi 320x240 dengan *bitrate* maksimal 256kbps. Akan tetapi menghasilkan *delay time video streaming* yang berbeda. Hal ini dikarenakan pada lokasi kedua memiliki kecepatan *upload* 6,98Mbps dan lokasi ketiga 1,1Mbps. Perbedaan kecepatan *upload* inilah yang menyebabkan *delay time* yang berbeda. Sehingga didapatkan konfigurasi optimum video *streaming* dengan menggunakan resolusi 240x144 *pixel* dan *bitrate* maksimal 256kbps pada area dengan kecepatan *upload* 1,1Mbps.

3.4 Hasil Pengujian Fungsi Kendali

Pengujian ini dilakukan untuk mengetahui apakah data yang dikirimkan ke unit terbang sampai dan dapat dijalankan oleh MultiWii. Pengujian dengan menekan tombol L,J,K,H,G dan V pada *keyboard* dan melihat database GCS pada *database* dataperintah. Ketika operator menekan tombol fungsi kendali akan dibaca nilai ASCII tombol dan di *update* pada *database* kendali. *Database* kendali ini secara rutin dikirimkan GCS ke unit terbang.

Tabel 9 Data rcCMD yang dikirim ke MultiWii

Tombol	Nilai throttle pada GCS	Prosentase % throttle	RPM Motor
L ditekan	710	90 %	10508
K ditekan	610	78 %	10049
J ditekan	510	67 %	9877
H ditekan	410	55 %	9309
G ditekan	310	43 %	8731
V ditekan	210	31 %	7029

Pada Tabel 9 menunjukkan data dari GCS sampai pada unit terbang dan menggerakkan motor pada *quadrotor*. Data rcCMD dikirimkan ke MultiWii apabila nilai *throttle* pada GCS lebih dari 200.

3.5 Hasil Pengujian Pemantauan Lebih Dari Satu Pengguna

Pengujian ini dilakukan untuk mengetahui GCS dapat dipantau lebih dari satu pengguna dalam waktu yang bersamaan. Seperti pada Gambar 6, GCS dapat di akses oleh dua klien secara bersamaan.



Gambar 6 GCS diakses bersamaan oleh dua *client*

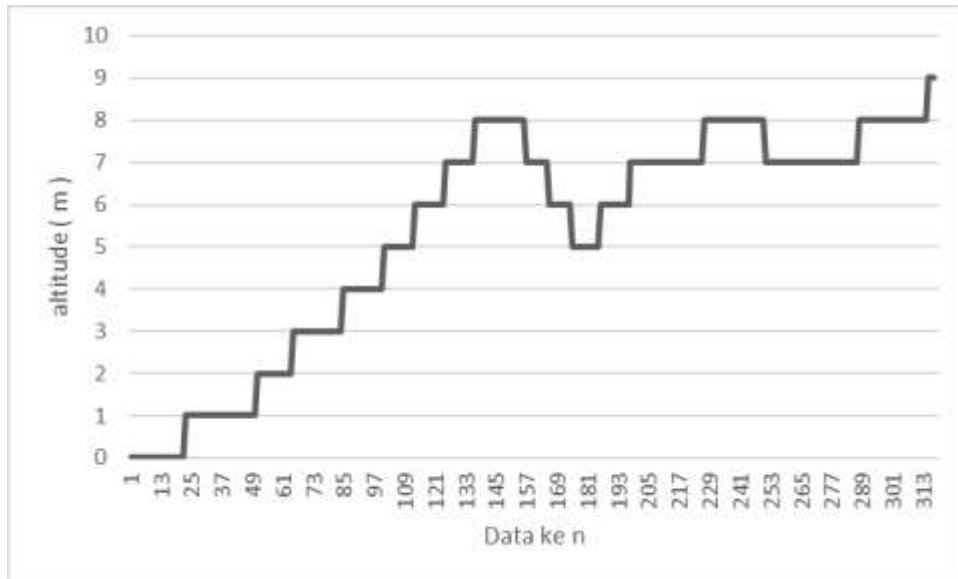
Pada *client* sebelah kiri mengakses GCS melalui jaringan internet dengan mengakses *ip public* indihome yang di *port forwarding* ke *ip server* GCS. Sedangkan pada *client* sebelah kanan terhubung pada jaringan lokal server, sehingga untuk mengakses GCS melalui *ip* 192.168.100.5.

3.6 Hasil pengujian keseluruhan sistem

Pengujian keseluruhan sistem dengan cara menjalankan sistem GCS dengan melakukan penerbangan menggunakan pesawat tanpa awak bertipe *quadrotor*. Pengujian dilakukan pada lokasi dengan koordinat Lat : -7.75761 dan Long : 110.37200 atau area pemukiman padat penduduk. Pengujian dilakukan pukul 08.00 pagi. Pengujian penerbangan dilakukan untuk mengetahui kinerja keseluruhan sistem GCS.

Pada pengujian ini, ketinggian maksimal penerbangan didapatkan setinggi 5 meter. Untuk *flight time* sekitar 5 menit dari normal tanpa muatan unit terbang sekitar 15 menit. Hal ini dikarenakan unit terbang menambah massa pesawat tanpa awak, sehingga daya listrik yang digunakan untuk membuat *quadrotor* terbang semakin besar.

Untuk pengujian penerbangan dengan melakukan lepas landas pada *quadrotor*. Lepas landas dengan menekan tombol L yang memberikan nilai throttle sebesar 1768. Serta dilakukan pengamatan pada antarmuka dan *database* GCS. Pada saat posisi UAV mulai naik, terlihat pada data *altitude* mengalami perubahan kenaikan ketinggian yang dapat dilihat pada Gambar 7.



Gambar 7 Data perubahan *altitude* pada *database*

Gambar 7 menunjukkan adanya perubahan kenaikan ketinggian *quadrotor* dari 0 meter perlahan menuju 9 meter. Sehingga ketinggian terbang berdasarkan data altitude setinggi 9 meter, sedangkan dari pemantauan langsung dilapangan ketinggian maksimal *quadrotor* terbang sekitar 5 meter. Perbedaan ini dikarenakan modul gps yang digunakan memiliki nilai *error* dalam membaca ketinggian.

Tampilan video *streaming* dapat berjalan dengan lancar. Untuk panel peta penerbangan dapat menampilkan koordinat lokasi pesawat dengan jeda pembaharuan lokasi setiap empat detik.

4. KESIMPULAN

Penelitian ini berhasil mengimplementasikan sistem GCS berbasis internet web server yang dapat menampilkan kondisi pesawat, video streaming dan melakukan perintah kendali.

Konfigurasi video streaming dengan resolusi 240x144 pixel, bitrate maksimal 256kbps dan framerate 5 fps menghasilkan delay time tidak lebih dari satu detik. Konfigurasi ini berjalan pada kecepatan upload 1.1 Mbps dengan prosentase kompresi bitrate 93.83%. Data kondisi pesawat yang dikirim ke GCS optimal apabila bandwidth internet (upload) melebihi bitrate video streaming sebesar 7,68 Kbps.

5. SARAN

Dari penelitian ini, untuk pengembangan lebih lanjut dapat disarankan untuk pengujian terbang menggunakan pesawat yang memiliki daya angkat lebih tinggi dan daya baterai lebih besar agar flight time lebih lama. Menerapkan fungsi kendali perintah menggunakan joystick atau gamepad. Mencoba menggunakan framework selain FFMPEG agar diketahui apakah framework berpengaruh pada proses video streaming.

DAFTAR PUSTAKA

- [1] E. Torun, "UAV Requirements and Design Consideration," The Research and Technology Organization (RTO) of NATO, pp. B1-8, 2000 [Online]. Available: <http://www.dtic.mil/get-tr-doc/pdf?AD=ADA381871#page=222>. [Accessed: 02-Jun-2017]
- [2] A. A. Farghani, R. Sumiharto, S. B. Wibowo, "Purwarupa Ground Control Station untuk Pengamatan dan Pengendalian Unmanned Aerial Vehicle Bersayap Tetap," *IJEIS (Indonesian J. Electron. Instrum. Syst.*, vol. 3, no. 1, pp. 1-10, 2013 [Online]. Available: <https://journal.ugm.ac.id/ijeis/article/view/3834>. [Accessed: 02-Jun-2017]
- [3] U. W. T. P. Putro, T. K. Priyambodo, and R. Sumiharto, "Pengembangan Antarmuka Stasiun Pemantauan Pesawat Tanpa Awak Menggunakan Aplikasi Android," *IJEIS (Indonesian J. Electron. Instrum. Syst.*, vol. 4, no. 2, pp. 113-122, 2014 [Online]. Available: <https://jurnal.ugm.ac.id/ijeis/article/view/7110>. [Accessed: 01-Jun-2017]
- [4] A. P. Manggala, R. Sumiharto, S. B. Wibowo, "Purwarupa Air Data, Attitude, dan Heading Reference System untuk Unmanned Aerial Vehicle," *IJEIS (Indonesian J. Electron. Instrum. Syst.*, vol. 2, no. 1, pp. 21-32, 2012 [Online]. Available: <https://jurnal.ugm.ac.id/ijeis/article/view/2337>. [Accessed: 04-Jun-2017]
- [5] E. N. Wijatsongko, A. E. Putra, B. N. Prastowo, "Sistem Pemantauan Ruang Dengan Server Raspberry Pi," *IJEIS (Indonesian J. Electron. Instrum. Syst.*, vol. 5, no. 1, pp. 65-76, 2015 [Online]. Available: <https://jurnal.ugm.ac.id/ijeis/article/view/7154>. [Accessed: 04-Jun-2017]
- [6] D. Holandrio, R. Sumiharto, B. A. A. Sumbodo, "Sistem Video Streaming dengan Server Mini Personal Computer (Mini Pc) pada Jaringan Ad-Hoc," *IJEIS (Indonesian J. Electron. Instrum. Syst.*, vol. 2, no. 2, pp. 165-174, 2012 [Online]. Available: <https://journal.ugm.ac.id/ijeis/article/view/2436>. [Accessed: 02-Jul-2017]
- [7] F. R. Hadiputra, T. K. Priyambodo, "Purwarupa Pengendalian Jarak Jauh Pada Mobile Robot Berbasis Web Melalui Jaringan Wireless TCP/IP," *IJEIS (Indonesian J. Electron. Instrum. Syst.*, vol. 6, no. 1, pp. 105-116, 2016 [Online]. Available: <https://jurnal.ugm.ac.id/ijeis/article/view/15247>. [Accessed: 01-Jun-2017]
- [8] Khan, A., Sun, L. and Ifeachor, E., 2009, June. Content clustering based video quality prediction model for MPEG4 video streaming over wireless networks. In *Communications, 2009. ICC'09. IEEE International Conference on* (pp. 1-5). Available: IEEE. <http://ieeexplore.ieee.org/abstract/document/5198850/>. [Accessed: 01-Jun-2017]
- [9] S. S. Henura, T. W. Widodo, "Rancang Bangun Sistem Jaringan Nirkabel untuk Pemantauan Penggunaan Air Pelanggan PDAM," *IJEIS (Indonesian J. Electron. Instrum. Syst.*, vol. 5, no. 2, pp. 155-164, 2015 [Online]. Available: <https://jurnal.ugm.ac.id/ijeis/article/view/7639>. [Accessed: 04-Jun-2017]
- [10] I. F. Priyanta, M. Rivai, R. Dikairono, "Pemetaan Distribusi Gas Polutan Menggunakan Quadcopter Berbasis Autonomous Waypoint Navigation," *Jurnal Teknik ITS*, vol. 5 no. 2, pp. 154-159, 2016 [online]. Available: <http://ejurnal.its.ac.id/index.php/teknik/article/view/16238/0>. [Accessed: 04-Jun-2017]