

## PERANCANGAN DAN REALISASI SISTEM KENDALI DAN KOMUNIKASI MULTI CLIENT GROUND STATION UNTUK EDF ROKET

### *DESIGN AND REALISATION CONTROL SYSTEM AND COMMUNICATION MULTI CLIENT GROUND STATION FOR EDF ROCKET*

Adam Aji Nugroho<sup>1</sup>, Burhanuddin<sup>2</sup>, Sarwoko<sup>3</sup>

<sup>1</sup>Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom

<sup>2</sup>Prodi S1 Sistem Komputer, Fakultas Teknik Elektro, Universitas Telkom

<sup>3</sup>Prodi S1 Teknik Elektro, Fakultas Teknik Elektro, Universitas Telkom

[adam.ajinugroho@yahoo.com](mailto:adam.ajinugroho@yahoo.com), [brh@telkomuniversity.ac.id](mailto:brh@telkomuniversity.ac.id), [swk@telkomuniversity.ac.id](mailto:swk@telkomuniversity.ac.id)

**Abstrak** – Roket merupakan peluru kendali atau kendaraan terbang yang mendapatkan dorongan sehingga dapat terbang ke udara. Terdapat berbagai jenis roket diantaranya *Electric Ducted Fan (EDF)* roket yang memakai motor sebagai penggerak dan dikhususkan pada ketinggian rendah. Untuk memantau dan mengendalikan roket diperlukan terminal yang dinamakan *Ground station (GS)*. Terminal ini terletak di darat dan keberadaannya wajib guna memantau misi dan data yang ditangkap oleh roket melalui media gelombang radio. Penelitian ini penulis menyajikan hasil penelitian tersebut dalam bentuk sistem yang terdiri dari alat *transceiver* dan aplikasi. Aplikasi ini terdiri dari *server* dan *client* sehingga mempunyai kemampuan *multi client* agar dalam pengaplikasiannya pengguna dapat memantau roket secara bersama-sama. EDF roket akan berkomunikasi dengan *server* menggunakan RF *transceiver* dan *client* berkomunikasi menggunakan *ethernet* atau *wireless* dengan *server*. EDF roket akan mengirim data sensor. Aplikasi ditulis dengan bahasa C#4.0 dan pada *server* terjadi proses *real-time* sehingga dari sekian banyak data yang diterima akan tidak memberatkan kinerja aplikasi sedangkan pada *client* terdapat *multi threading* agar dengan banyaknya pengolahan data program tidak menjadi lambat. Dari hasil pengukuran didapatkan 10 *client* program masih berjalan dengan baik, pengujian *black box* berjalan lancar dan terjadi delay yang kecil untuk yaitu 6.2 ms untuk 10Hz dan 10.6 untuk 15 Hz.

**Kata kunci** : *ground station, C#, multi client, EDF roket, real-time, sensor*

**Abstract** - A missile or rocket is flying vehicle that get a boost so that it can fly into the air. There are various types of rockets including *Electric ducted fan (EDF)* rocket that uses the motor as a driver and devoted at low altitudes. To monitor and control the rocket called *Ground terminal station (GS)*. The terminal is located on land and its existence is obliged to monitor the mission and the data that is captured by a rocket through the medium of radio waves. With this study the authors present the results of these studies in the form of systems consisting of a *transceiver* and an application tool. The application consists of *server* and *client* so that the *client* has the ability to apply multiple users can monitor the rocket together. EDF rocket will communicate with the *server* using the RF *transceiver* and *client* communicate using *ethernet* or *wirelessly* with the *server*. EDF rocket will send the sensor data. Applications written in C # 4.0 and the *server* process occurs *real-time* so that from the many data received will not be burdensome to the *client* application performance while there are *multi-threading* so that the number of data processing programs do not become slow. From the measurement got concluded with 10 *client* the program still running well, *black box* testing also success and there is a small delay for real time that is 6.2 ms for 10 Hz and 10.6 for 15 Hz.

**Keywords**: *ground station, C#, multi client, EDF rocket, real-time, sensor*

#### I. PENDAHULUAN

Di era modern, teknologi roket sudah mempengaruhi segala macam bidang pekerjaan, peluncuran satelit, militer, perjalanan luar angkasa, *surveillance* dan lainnya. Untuk mendukung kinerja roket perlu didukung *Ground station (GS)* guna memantau roket terutama dalam misi. Area antara roket dan GS adalah udara sehingga pada *unguided area* ini perlu dirancang sebuah GS yang matang dan mampu berkomunikasi jarak jauh dengan roket untuk data yang sekian banyak dikirim oleh roket.

Data yang dikirim roket akan diterima oleh GS dan diolah, hasil data akan dianalisa oleh tim untuk menentukan keputusan lebih lanjut. Sehingga dirasa perlu untuk menambahkan fitur *multi client* pada GS untuk meningkatkan kinerja operasi tim.

Oleh karena itu dirancanglah *multi client monitoring* sistem pada kasus EDF roket yang dilengkapi dari 2 sub sistem yaitu *server* sebagai pusat komunikasi dengan roket dan *client* serta pemrosesan utama data-data dari roket, serta *client* sebagai tatap muka dengan *user*. Sistem ini kemudian dilengkapi dengan *real-time* sistem guna menjamin proses yang dilakukan aplikasi berjalan lancar untuk tiap pengguna.

Adapun batasan masalah yang dibahas dalam penelitian ini meliputi :

- 1) aplikasi *server* dan *client* menggunakan bahasa C#4.0 dengan IDE Visual C# 2010 Express,
- 2) koneksi *client* dan *server* dalam skala LAN,
- 3) menggunakan UDP dalam koneksi antara *server* dan *client*,
- 4) menggunakan komunikasi serial antara RF *transceiver* dan *server*,
- 5) menggunakan *real-time system* pada aplikasi *server*.

## II. TEORI PENUNJANG

### A. Remote Sensing (Penginderaan Jauh) <sup>[1]</sup>

Remote Sensing atau Penginderaan jauh (atau disingkat inderaja) adalah pengukuran atau akuisisi data dari sebuah objek atau fenomena oleh sebuah alat yang tidak secara fisik melakukan kontak dengan objek tersebut atau pengukuran atau akuisisi data dari sebuah objek atau fenomena oleh sebuah alat dari jarak jauh, (misalnya dari pesawat, pesawat luar angkasa, satelit, kapal atau alat lain). Contoh dari penginderaan jauh antara lain satelit pengamatan bumi, satelit cuaca, memonitor janin dengan ultrasonik dan wahana luar angkasa yang memantau planet dari orbit. Inderaja berasal dari bahasa Inggris remote sensing, bahasa Perancis télédétection, bahasa Jerman fernerkundung, bahasa Portugis sensoriamento remota, bahasa Spanyol percepción remote dan bahasa Rusia distantsionnaya. Di masa modern, istilah penginderaan jauh mengacu kepada teknik yang melibatkan instrumen di pesawat atau pesawat luar angkasa dan dibedakan dengan penginderaan lainnya seperti penginderaan medis atau fotogrametri. Walaupun semua hal yang berhubungan dengan astronomi sebenarnya adalah penerapan dari penginderaan jauh (faktanya merupakan penginderaan jauh yang intensif), istilah "penginderaan jauh" umumnya lebih kepada yang berhubungan dengan teresterial dan pengamatan cuaca.

### B. Ground Station<sup>[1]</sup>

Sebuah Ground Station atau stasiun bumi adalah stasiun terminal darat yang dirancang untuk menerima data telemetri (pengukuran besaran dari jarak jauh) melalui sistem transmisi. Stasiun bumi berkomunikasi dengan wahana dengan mengirimkan dan menerima gelombang radio pada frekuensi tertentu yang sesuai dengan kebutuhan. Stasiun bumi harus dapat mentransmisikan gelombang radio untuk wahana atau sebaliknya, yaitu menerima gelombang radio dari wahana menggunakan sambungan telekomunikasi. Sebuah stasiun bumi juga harus mampu menerima dan membaca data telemetri yang dikirim wahana secara realtime.

### C. C#. NET<sup>[3]</sup>

C# adalah bahasa multi-platform yang sudah menggunakan *type-safe object-oriented language*. Bahasa ini memungkinkan pengembang untuk membuat aplikasi yang aman dan powerful menggunakan *.NET Framework*. Pengembang bisa menciptakan aplikasi Windows *client*, XML Web *services*, *distributed components*, aplikasi *client-server*, aplikasi *database*, dan lainnya. Visual C# menyediakan editor program yang penuh fitur, tampilan yang menarik, debugger terintegrasi, dan masih banyak lagi alat bantu yang bisa digunakan dalam membantu pengembangan aplikasi dengan bahasa C# dan .NET Framework.

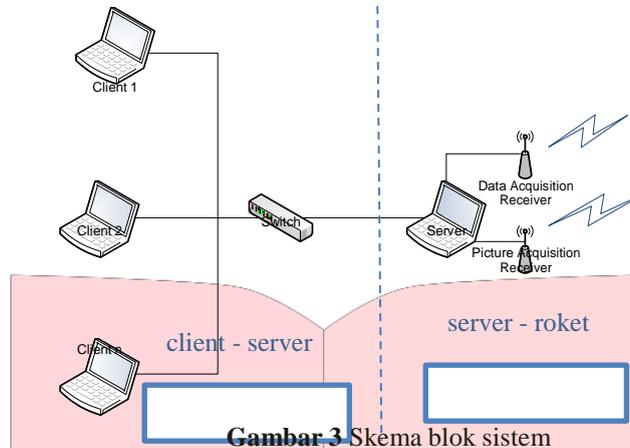
### D. MAVLINK<sup>[4]</sup>

MAVLink atau Micro Air Vehicle Link adalah *protocol* untuk berkomunikasi dengan perangkat kendaraan tak berawak atau yang biasa disebut *unmanned vehicle*(UAV). Protocol ini didesain sebagai paket data yang mempunyai aturan tertentu. MAVLink dirilis pertama kali tahun 2009 oleh Lorenz Meier. Aplikasi yang sering dipakai adalah komunikasi antara Ground Control Station dengan UAV. Tujuan dari diciptakannya protocol ini adalah adanya standarisasi protocol untuk komunikasi antara GCS dan UAV sehingga pengembang Flight Controller dapat mengikuti protocol universal tersebut tanpa mengembangkan sendiri protocol mereka yang kemudian dapat memberatkan pengembang GCS dalam menentukan pembacaan data.

### III. PERANCANGAN & IMPLEMENTASI

#### A. Model dan Desain Sistem

Sistem yang diterapkan pada perancangan ini dapat diklasifikasikan menjadi dua bagian yaitu antara komunikasi roket dengan server dan server dengan client. Berikut arsitektur dari ground station yang dirancang :

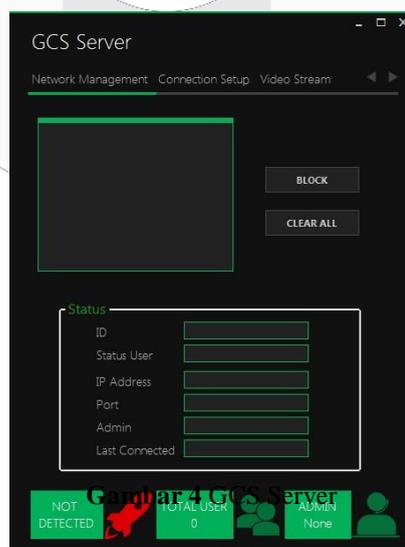


Gambar 3 Skema blok sistem

Pada blok sistem terjadi proses besar yakni data roket diterima kedalam server dengan radio telemetri kemudian data tersebut akan diolah dan disebar kedalam client melalui ethernet ataupun wireless dengan protocol UDP. UDP bersifat connectionless sehingga pengirim hanya mengirim data tanpa ada acknowledgment agar tidak mengganggu proses realtime sistem pada server. Error yang terjadi oleh UDP dapat terminimalisir karena error yang terjadi akan ditumpuk oleh data berikutnya sehingga error tidak akan terlihat. Adapun data yang diterima dari roket dikelompokkan menjadi dua yaitu data gambar dan non gambar. Data non gambar berupa data *flight dynamics* dan data *sensor acquisition*. *Flight dynamics* yaitu parameter dari roket itu sendiri seperti pitch, roll, heading dan lainnya sedangkan data sensor seperti koordinat, ketinggian. Sedangkan data gambar berasal dari kamera pada roket yang dikirim dalam bentuk frame – frame melalui radio yang terpisah dengan radio telemetri non gambar.

#### B. Implementasi Sistem Server (GCS Server)

Pada bagian *server*, aplikasi telah ditulis dengan C# 4.0. Terdapat beberapa *pages* pada aplikasi ini baik yang berupa manajemen maupun *built-in test*. Berikut adalah *pages* yang ada pada *server*.



Gambar 4 GCS Server

- 1) Overlay Status
- 2) Connection Setup

- 3) Network Management
- 4) Network Status
- 5) Video Stream
- 6) Waypoint
- 7) Flight Control

### C. Implementasi Sistem Client (GCS Client)

Pada bagian *client*, aplikasi juga ditulis dengan C# 4.0. Terdapat beberapa *pages* pada aplikasi ini. Beberapa *pages* bersifat hanya menerima data sedangkan lainnya dapat mengirim data ke *server* guna proses validasi ataupun kontrol.



Gambar 5 GCS Client

- 1) Overlay Status
- 2) Instrument Indicator
- 3) MAP
- 4) Textual Data Acquisition
- 5) Visual Data Acquisition
- 6) Initial Setup & Status

## IV. PENGUJIAN DAN ANALISIS

### A. Analisa waktu eksekusi pengiriman pada *server*

Ukuran struktur data yang digunakan berkisar 2500 bytes sehingga dapat diketahui bahwa algoritma yang digunakan memiliki kompleksitas waktu [8]:

$$T_{\min}(n) = n \text{ dan } T_{\max}(n) = 5003 + n$$

Dengan asumsi tidak ada user yang di blokir dan waktu pemrosesan method "BeginSendTo" sama dengan operasi yang lain. Jika asumsi waktu pemrosesan pada PC yang digunakan adalah  $10^{-6}$  detik maka untuk  $n=30$  *client*,

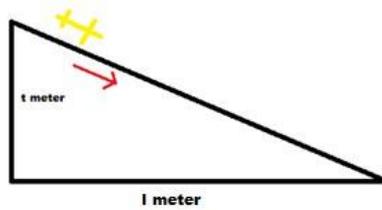
$$T_{\min}(30) = 30 \times 10^{-6} = 3 \times 10^{-5} \text{ s} = 0.03\text{ms}$$

$$T_{\max}(30) = (5003 + 30) \times 10^{-6} = 5033 \times 10^{-6} \text{ s} = 5.033\text{ms}$$

Terdapat perbedaan antara hasil pengukuran dan perhitungan dikarenakan method pengiriman ethernet yang dimiliki C# tidak dapat diketahui source code dalamnya sehingga tidak dapat menghitung kompleksitas waktunya. Kemudian pengaruh beban CPU yang berlebihan ketika 30 aplikasi berjalan bersamaan.

### B. Pengujian aplikasi dengan simulasi terbang

Simulasi terbang dilakukan karena untuk mengantisipasi rusaknya roket karena uji terbang sehingga dilakukanlah rekayasa yang dapat mensimulasikan terbang roket. Hal ini untuk mengetahui apakah ground station masih dapat menerima data dengan baik atau tidak. Terdapat 2 simulasi yaitu menjatuhkan roket dari ketinggian tertentu dan meluncurkan roket dengan bergantung pada suatu tali. Pengujian pertama dilakukan dengan gambar seperti berikut :



**Gambar 6** Visualisasi simulasi terbang pengujian pertama

Dengan  $t = 120$  meter dan  $l = 80$  meter, sehingga lintasan yang dilalui adalah

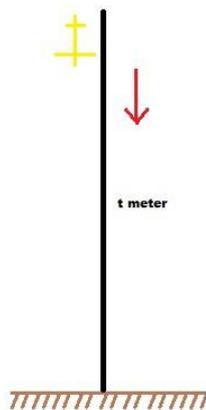
$$(\ ) \sqrt{\quad}$$

Berikut sampel data yang diterima :

**Tabel 1** Sampel data uji simulasi terbang pertama

Pengambilan data ke -	Detik ke -	Latitude	Longitude	Altitude	Roll	Pitch	Yaw
1	1	-6,97369	107,631461	110,24	3,679299	-42,067	21,201
	2	-6,97368	107,63147	89,099	-4,40318	-35,7	62,9970
	3	-6,97339	107,631901	54,09	-4,40318	-35,7286	62,997
	4	-6,97332	107,632011	17,189	12,49789	-35,6864	38,356
	5	-6,97333	107,632043	0,74	9,877608	-21,7798	47,434
2	1	-6,97372	107,631471	100,41	-2,19748	-41,5086	44,2245
	2	-6,97359	107,63165	86,35	-4,77946	-41,048	87,009
	3	-6,97356	107,631699	20,539	4,206283	-26,4682	50,906
	4	-6,97349	107,631786	13,38	5,071861	-10,5334	2,92995
	5	-6,97336	107,631972	0,5	-2,23959	-39,675	76,662
3	1	-6,97369	107,63146	101,87	3,35537	-41,376	24,4187
	2	-6,97368	107,631474	89,3	0,396436	-47,2333	38,085
	3	-6,97352	107,63172	80,96	-5,1181	-11,554	95,5179
	4	-6,97335	107,63196	65,66	3,286732	-27,358	11,672
	5	-6,97333	107,632044	0,03	-2,0036	-21,4864	46,3735
4	1	-6,97371	107,631471	106,67	-2,4164	-30,67	40,1288
	2	-6,97372	107,6314707	69,71	-2,5454	-47,834	46,6867
	3	-6,97356	107,6316985	41,539	4,206283	-26,468	50,9056
	4	-6,97359	107,6317773	9,579	-0,72585	-38,562	75,0096
	5	-6,97343	107,63187	0,93	-1,1461	-41,464	99,789

Pada pengujian kedua dilakukan dengan lintasan yang vertikal dengan sudut  $90^\circ$ . Visualisasi simulasi ditunjukkan seperti gambar berikut :



**Gambar 7** Visualisasi simulasi terbang pengujian kedua

Dengan t = 120 meter, berikut sampel data yang diterima :

**Tabel 2** Sampel data uji simulasi terbang pertama

Pengambilan data ke -	Detik ke -	Latitude	Longitude	Altitude	Roll	Pitch	Yaw
1	1	-6,97367	107,6315	93,669	14,97	-23,52	324,74
	1,5	-6,97356	107,6317	76,58	6,592	-13,21	300,052
	2,5	-6,97355	107,6317	0,049	-4,359	1,014905	177,9
2	1	-6,9737	107,6315	98,32	12,326	-13,7	324,92
	1,5	-6,97358	107,6317	63,34	12,062	-3,431	287,46
	2,5	-6,97353	107,6317	0,36	-141,2	-16,97	149,9

Dari hasil yang diperoleh diketahui bahwa groundspeed rata – rata pada pengujian pertama adalah 28,84 m/s dan vertical speed rata – rata pada pengujian kedua adalah 48 m/s. Dengan kecepatan yang terjadi ground station masih dapat mengolah data dengan baik atau tidak mendapatkan data *loss* pada sampel data.

**C. Pengujian aplikasi pada saat terbang manual**

Pengujian ground station dilakukan dengan menerbangkan roket secara manual dengan bantuan remote radio. Berikut hasil jejak roket pada saat terbang manual :



**Gambar 8** Data rekam lintasan yang dilalui roket pengujian

**Tabel 3** Sampel data uji terbang manual pengujian

De tik ke -	Latitude	Longitude	Altitude	Pitch	Roll	Yaw	Thro-ttle	Ground speed	Vertical speed
1	-6,980	107,5697	17,69	5,225	1,83	305,25	56	1,38	9,79
10	-6,981	107,5695	57	24,4115	-68,53	149,59	58	20,19	5,348
20	-6,9807	107,5701	56,63	-23,113	-50,1	260,08	48	13,26	-0,341
35	-6,9822	107,5704	55,41	24,77	-1,994	81,88	56	22,89	-0,299
40	-6,9813	107,5712	31,8	-22,41	-29,89	341,5	47	14,7	-1,65
45	-6,9812	107,5712	28,3	-37,41	-101,4	352,7	48	14,11	-7,069

**V. PENUTUP**

Berdasarkan pengujian dan analisis yang dilakukan, maka dapat disimpulkan beberapa hal sebagai berikut:

1. Ground Station dapat berkomunikasi dengan Flight Controller APM pada roket dengan menggunakan protokol mavlink

2. Fitur pada aplikasi *server* dan *client* berfungsi secara baik pada pengujian black box
3. Perealisasian sistem *telemetry* berjalan dengan baik pada pengujian simulasi terbang maupun terbang manual dengan kecepatan dan jarak yang didapatkan pada masing - masing pengujian
4. Aplikasi real-time pada *server* dengan spesifikasi yang telah disebutkan terdapat delay yang kecil sehingga tidak mempengaruhi pengiriman data ke *client*, yaitu rata – rata 6,2 ms untuk 10Hz dan 10,6 untuk 15 Hz
5. Pengiriman *video* dan *data acquisition* dapat dilakukan dari *server* kepada *client* dengan baik dan secara bersamaan
6. Telah terealisasi aplikasi multi client ground station dengan pengujian 10 *client* berjalan dengan baik yaitu dengan rata – rata throughput 1,0967 Mbit/s dan delay  $9,1604 \times 10^{-7}$  untuk satu bitnya

## REFERENSI

- [1] Wijaya, Wandu, (2015). Perancangan Dan Realisasi EDF Roket Dengan Sistem Kendali Dan Komunikasi Yang Dilengkapi Sistem Penginderaan Jauh Dan Multi Client Ground Station. *Bandung* : Telkom University.
- [2] “*Real-time systems*: Carnegie Mellon University”. [online]. ([http://users.ece.cmu.edu/~koopman/des\\_s99/real\\_time/](http://users.ece.cmu.edu/~koopman/des_s99/real_time/) , diakses September 2014).
- [3] Microsoft. *C# Programmer's Reference*: Electronic reference, [online] (<http://msdn.microsoft.com/en-us/library/>, diakses Juli 2014).
- [4] Mavlink, [online]. (<http://www.qgroundstation.com/mavlink.htm>, diakses Agustus 2014).
- [5] Albahari, Joseph. (2012). *C# 5.0 In a Nutshell*. *Sebastopol*:O'Reilly
- [6] Mavlink,[online].(<https://code.google.com/p/ardupilot-mega/wiki/MAVLink> , diakses 18 Maret 2015).
- [7] MP, [online]. (<https://github.com/diydrones/MissionPlanner-git>, diakses 12 Mei 2015).
- [8] Munir, Rinaldi.(2012). *Matematika Diskrit*. *Bandung*:Informatika
- [9] Zenhadi.(2015). *Pengukuran QoS Streaming Server*.*Surabaya*: Pens
- [10] UDP,[online]. (<https://ilmukomputer.org/2013/01/28/dns-ip-tcp-dan-udp>, diakses 11 April 2015).
- [11] Brassad, Gilles & Paul Bratley.(1988). *Algorithmics, Theory and Practice*. *California*:Prentice Hall
- [12] ITU, [online].<http://www.itu.int/rec/T-REC-G.1010-200111-I/en>, diakses 7 Agustus 2015).

