PEMBANGUNAN APLIKASI SIMULASI KONTROL TRAYEKTORI UNTUK PEMANTAUAN PERGERAKAN ROKET KECEPATAN RENDAH

Faddikha Irza Putra[1], Ken Kinanti Purnamasari[2]

[1,2] Universitas Komputer Indonesia
Jalan Dipatiukur No. 112-116, Coblong, Lebakgede, Bandung, Kota Bandung, Jawa Barat 40132
E-mail : faddikha@gmail.com[1], guinev_gs@yahoo.co.nz[2]

ABSTRAK

Roket kecepatan rendah biasanya digunakan sebagai *prototype* penelitian untuk riset roket kecepatan tinggi. Dalam pengembangannya, roket dilengkapi dengan dua sistem, yaitu sistem navigasi atau yang biasa disebut GCS (*Ground Control Station*) yang berguna untuk memonitor keadaan roket dan sistem kontrol penerbangan. Dalam penelitian kali ini, aplikasi GCS digunakan sebagai simulator untuk memvisualisasikan trayektori penerbangan roket yang datanya berasal dari log data uji terbang roket. Log data ini berupa *array* yang berisi hasil pengolahan data sensor akselerometer, giroskop, *magnetometer*, kompas dan gps. Tahap yang dilakukan untuk pengolahan log data uji terbang, yaitu *Parsing*, *Plotting* dan Visualisasi Data. Langkah terakhir dilakukan pengujian visualisasi data dengan kuesioner untuk melihat respon dari responden tentang visualisasi data dari aplikasi GCS yang sudah dibangun sesuai dengan lima aspek menurut Nielson (2000), yakni *learnability*, *efficiency*, *memorability*, *errors* dan *satisfaction*. Dari pengujian kuesioner ini, dapat disimpulkan bahwa aplikasi simulasi yang dibangun memiliki nilai penerimaan yang baik oleh responden dalam skala 5 dengan modus 4.

Kata kunci : GCS, Visualisasi Data, Log Data, *Parsing*, *Plotting*

1. PENDAHULUAN

Pengembangan roket kendali generasi sekarang sangatlah pesat, khususnya yang berkaitan dengan kemampuan roket untuk meluncur dengan kecepatan tinggi, manuver yang sulit dan tingkat presisi target yang tinggi. Dari sisi kecepatan, roket terbagi atas dua kategori, yakni roket dengan kecepatan tinggi yang dapat meluncur di atas 300 km/jam dan roket dengan kecepatan rendah yang dapat meluncur di bawah 300 km/jam dan biasanya digunakan sebagai prototype penelitian.

Dalam pengembangannya, roket juga dilengkapi dengan dua sistem yang dibutuhkan dalam riset peroketan, yaitu sistem navigasi atau yang biasa disebut GCS (Ground Control Station) yang berguna untuk memonitor keadaan roket dan sistem kontrol penerbangan [1]. Pergerakan roket secara visual dibutuhkan untuk melihat kecepatan dan sikap roket [2] yang ditampilkan melalui GCS pada perangkat komputer. Selain itu, roket tanpa sistem kontrol akan sangat berbahaya bila dipaksakan terbang karena tidak ada sistem kendali yang mengatur sikap dari roket saat meluncur dan tidak stabil.

Pembangunan aplikasi pemantauan pergerakan roket dengan kecepatan rendah ini terkait dengan penelitian sebelumnya tentang perancangan sistem monitoring dan kontrol stabilitas pada roket EDF [13]. Pada penelitian sebelumnya, aplikasi monitoring yang dibangun sebatas memonitor roket melalui pembacaan data sensor secara real-time dan masih belum bisa memvisualisasikan dan mensimulasikan pergerakan dari roket. Sehingga, pada saat melakukan uji terbang roket, peneliti roket EDF tidak mengetahui dimana posisi dan bagaimana pergerakan dari roket. Hal ini menyebabkan recovery roket setelah diluncurkan terkadang sulit untuk dilakukan karena posisi akhir dari roket tidak diketahui. Recovery sendiri merupakan proses pengambilan kembali wahana roket ke tempat yang aman setelah uji terbang dilakukan.

Pada penelitian kali ini akan dibangun sebuah aplikasi simulasi trayektori berupa GCS yang akan menampilkan visualisasi trayektori penerbangan roket dari titik awal peluncuran hingga roket mendarat kembali ke bumi dengan masukan berupa log data penerbangan roket dengan harapan aplikasi yang dibangun dapat mempermudah para peneliti roket EDF untuk mengetahui bagaimana pergerakan roket dari titik awal peluncuran dan dapat membantu proses *recovery* roket setelah uji terbang dilakukan.

2. ISI PENELITIAN

2.1. GCS (Ground Control Station)

GCS (*Ground Control Station*) atau GS (*Ground Segment*) adalah perangkat *transmitter-receiver* di stasiun bumi yang dilengkapi dengan perangkat komputer yang berfungsi untuk mengendalikan dan atau memonitor wahana roket dan atau payload yang sedang meluncur [6].



Gambar 1. GCS

2.2. Trayektori

Trayektori adalah lintasan pergerakan suatu benda yang berpindah pada satuan waktu tertentu [7]. Dalam setiap titik pada trayektori terdiri dari nilai posisi, kecepatan, dan *attitude* yang bisa menghasilkan akselerasi atau percepatan. Trayektori bisa berlaku pada benda yang memiliki kecepatan seperti satelit, kendaraan di darat, kapal laut, pesawat, dan lain-lain. Contoh trayektori yang banyak dijumpai dalam dunia pemetaan adalah *flight-path* atau jalur penerbangan pesawat yang diperlukan pada saat pemetaan dengan menggunakan pesawat udara.



Gambar 2. Trayektori dalam bentuk flight-path

2.3. Komunikasi Serial

Komunikasi serial ialah pengiriman data secara serial (data dikirim satu persatu secara berurutan), sehingga komunikasi serial lebih lambat daripada komunikasi paralel [12]. Komunikasi ini hanya membutuhkan satu jalur dan kabel, lebih sedikit dibandingkan dengan komunikasi paralel. Pada prinsipnya komunikasi serial merupakan komunikasi dimana pengiriman data dilakukan per bit, atau dengan kata lain komunikasi serial merupakan salah satu metode komunikasi data dimana hanya satu bit data yang dikirimkan melalui seuntai kabel pada suatu waktu tertentu. Pada dasarnya, komunikasi serial adalah kasus khusus komunikasi paralel dengan nilai n = 1, atau dengan kata lain adalah suatu bentuk komunikasi paralel dengan jumlah kabel hanya satu dan hanya mengirimkan satu bit data secara simultan. Hal ini dapat dibandingkan dengan komunikasi paralel yang sesungguhnya di mana n-bit data dikirimkan bersamaan dengan nilai umumnya $8 \leq n \leq 128$.

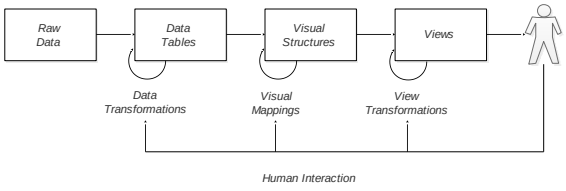
Komunikasi data serial mengenal dua buah metode, yaitu *synchronous* dan *asynchronous*. Metode *sychronous* mengirimkan datanya beberapa *byte* atau karakter (atau disebut blok data atau frame) sebelum meminta konfirmasi apakah data sudah diterima dengan baik atau tidak. Contoh pengunaan metode *synchronous* serial terdapat pada transmisi data keyboard. Sementara itu, metode *asynchronous* data dikirim satu *byte* setiap pengiriman. Biasanya tidak dibutuhkan konfirmasi menerimaan data. Contoh penggunaan *asynchronous* serial adalah pada UART (*Universal Asynchronous Receiver Transmitter*) yang digunakan pada *serial port* (COM) komputer. Dari kedua jenis metode tersebut dapat dipilih dan dilakukan lewat program.

2.4. Visualisasi Data

Visualisasi data adalah konversi data kedalam format visual (tabel atau grafik) sehingga karakteristik dari data dan relasi diantara item data atau atribut dapat dianalisis atau dilaporkan. Visualisasi data adalah satu dari teknik yang paling baik dan menarik untuk eksplorasi data. Manusia memiliki kemampuan untuk menganalisis sejumlah besar informasi yang dipersentasi secara visual.

Visualisasi data dapat didefinisikan sebagai berbagai jenis cara untuk membuat gambar, diagram atau animasi dengan tujuan mengkomunikasikan sebuah pesan atau informasi. Pada umumnya visualisasi digunakan untuk mengagregasi data dalam jumlah yang sangat besar yang kemudian dipersentasikan dengan berbagai model.

Memonitor pergerakan roket dengan memanfaatkan kapabilitas mata manusia melalui sebuah GCS dapat memberikan sebuah informasi yang lebih bermanfaat dalam mendeteksi anomali gerak roket. Berdasarkan asumsi tersebut, visualisasi data melalui GCS menjadi alternatif alat dalam memonitor pergerak roket.



Gambar 3. Proses visualisasi data

a. *Raw Data* merupakan data mentah yang nantinya akan diolah.

b. *Data Tables* merupakan data mentah ditransformasikan ke dalam tabel data terstruktur yang memiliki beberapa arti direpresentasikan melalui metadata.

c. *Visual Structures* merupakan data yang disusun ditransformasikan ke dalam suatu model geometrik dengan memilih bentuk dasar geometris seperti titik, garis dan poligon dan menetapkan data atribut.

d. *Views* merupakan data geometrik selanjutnya ditransformasikan menjadi gambar. Hal ini dilakukan dengan melihat perubahan skala, translate, zoom dan representasi grafis.

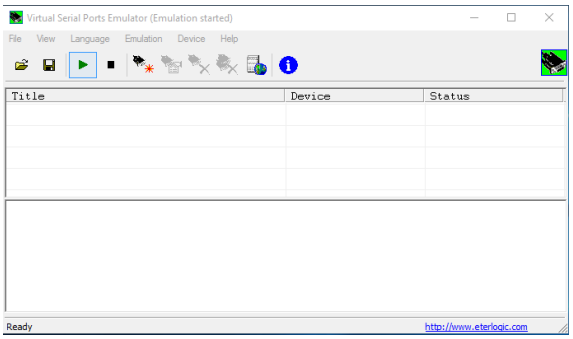
2.5. VSPE

VSPE berfungsi untuk membantu para *software engineer* dan *developer* untuk membuat/men-*debug*/menguji aplikasi-aplikasi yang melalui kabel ataupun menggunakan *port serial*. *Developer* yang mengembangkan *software* dan berhubungan dengan perangkat yang terpasang di komputer melalui kabel atau *port serial* biasanya dibatasi dengan hanya beberapa jumlah *port* yang ada pada komputer. Padahal, semakin banyak port saat melakukan pengujian aplikasi akan semakin baik.

Untuk mengatasi hal tersebut, kita dapat menggunakan VSPE (*Virtual Serial Port Emulator*). Seperti namanya, *software* ini berfungsi untuk membuat *emulator port virtual*. Sehingga, tiap satu komputer pengguna dapat menggunakan hingga 255 *port*. Dengan bantuan *software* ini, proses pembuatan, pengujian dan penelusuran kesalahan akan menjadi lebih mudah.

Tidak seperti port yang sebenarnya yang hanya bisa digunakan untuk satu *software* atau perangkat, pada *port* virtual ini satu port dapat digunakan lebih dari satu aplikasi. Adapun beberapa fitur dari *software* ini, antara lain :

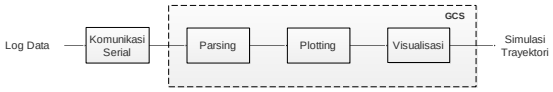
a. Perangkat virtual untuk *connector*, *data splitter* dan perangkat pasangan (*pair*).
b. Menyambungkan berbagai perangkat.
c. Mode perangkat pengguna mencakup *TCP Server*, *TCP Client*, *Serial Redirector*, *UDP Manager* dan *Bridge*.
d. Mendukung pemrograman atau *script* dengan bahasa pemrograman Phyton.
e. Mendukung arsitektur *processor* x86 (32bit) dan x86_64 (64bit).
f. Tersedia API (*Application Programming Interface*) untuk C dan C++ untuk pengembangan aplikasi secara langsung.
g. Tersedia fitur untuk memonitoring penggunaan data.



Gambar 4. VSPE

2.6. Analisis Sistem

Berikut merupakan arsitektur sistem yang dibangun.



Gambar 5. Arsitektur sistem yang dibangun

Sistem baru yang akan dibangun menggunakan masukan program berupa log data yang didapat dari pengolahan data penerbangan roket. Log data akan dikirimkan secara serial.

2.7. Data Masukan

Data masukkan yang digunakan dalam simulasi trayektori penerbangan roket EDF ini, yaitu *log data* dari uji terbang yang dilakukan dengan format .txt yang dapat dilihat pada lembar lampiran. *Log data* disini merupakan *fixed data* yang diperoleh dari pengolahan data sensor akselerometer, giroskop, *magnetometer* dan gps. *Log data* yang digunakan berupa *array* yang datanya dipisahkan berdasarkan spasi dan dikirim melalui aplikasi komunikasi serial dan selanjutnya pada GCS data tersebut akan di *parsing*, *plotting* dan kemudian divisualisasikan sesuai dengan parameter-parameter yang diperlukan.

```
1 -4,82 -1,61 12,82 1 -10 25 493 694 612 6,99 -21,7 106 1018 -6,85428 107,6039505
```

Gambar 6. Data masukan

Adapun format data masukan dari log data adalah sebagai berikut.

a. Digit pertama merupakan indeks data.
b. Digit kedua hingga keempat merupakan data sensor akselerometer pada masing-masing sumbu X, Y dan Z.
c. Digit kelima hingga ketujuh merupakan data sensor giroskop pada masing-masing sumbu X, Y dan Z.
d. Digit kedelapan hingga kesepuluh merupakan data sensor *magnetometer* pada masing-masing sumbu X, Y dan Z.
e. Digit kesebelas merupakan data *roll.*
f. Digit kedua belas merupakan data *pitch.*
g. Digit ketiga belas merupakan data *heading.*
h. Digit keempat belas merupakan data ketinggian.

i. Digit kelima belas merupakan data *latitude* (garis lintang).
j. Digit keenam belas merupakan data *longitude* (garis bujur).

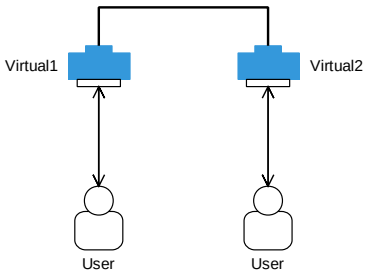
2.8. Analisis Proses

Berikut analisis proses yang dilakukan.

a. Log data sebagai input program dimasukkan ke aplikasi komunikasi serial.
b. Aplikasi komunikasi serial akan mengirimkan log data ke GCS.
c. Pada GCS, log data akan melalui tiga proses, yakni *parsing*, *plotting* dan *visualisasi.*
d. Keluaran dari program merupakan simulasi trayektori penerbangan roket dilengkapi dengan data sensor dan *avionic instrument.*

2.9. Proses Komunikasi Serial

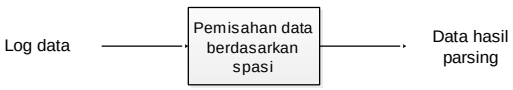
Log data dikirim dengan metode serial setelah dilakukan proses koneksi *pairing virtual port* agar *port virtual* yang di-*pair* dapat berkomunikasi. Komunikasi ini akan menghubungkan antara aplikasi GCS dengan aplikasi komunikasi serial agar data yang berbentuk *array* pada log data dapat dikirim dari aplikasi komunikasi serial ke GCS.



Gambar 7. Virtual port pairing

2.10. *Parsing* Data

Memisahkan karakter dari *log data* untuk membagi array data yang diterima kedalam *variable* data sensor. Proses ini akan memudahkan kita pada proses selanjutnya, yakni *plotting* data.



Gambar 8. *Parsing* data

2.11. *Plotting* Data

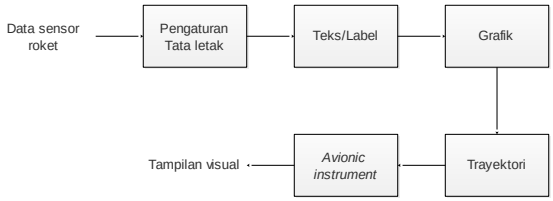
Menempatkan data yang telah diparsing dan menyesuaikan dengan variabel data sensor yang sudah dideklarasikan. Berikut blok diagram *plotting* data.



Gambar 9. *Plotting* data

2.12. Visualisasi

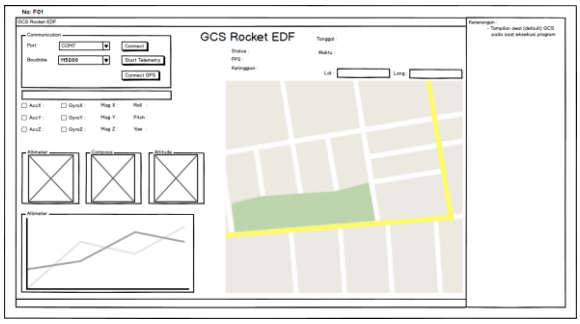
Pemvisualisasian data dilakukan setelah melalui proses *parsing* dan *plotting*. Visualisasi data ditujukan untuk membantu pengguna aplikasi untuk memahami informasi-informasi dari simulasi trayektori yang disajikan dalam bentuk GUI (*Graphical User Interface*).



Gambar 10. Visualisasi data

2.13. Perancangan Antarmuka

Berikut merupakan perancangan antarmuka dari sistem yang dibangun.



Gambar 11. Perancangan antarmuka

Aplikasi GCS yang dibangun akan menampilkan informasi berupa label/teks dari sensor yang digunakan, *avionic instrument* yang terdiri dari komponen *attitude*, kompas dan *altitude*, grafik ketinggian dan *map* yang akan memvisualkan trayektori pergerakan dari roket.

2.14. Implementasi Antarmuka

Berikut implementasi antarmuka dari perancangan antarmuka aplikasi yang dibangun.



Gambar 12. Implementasi antarmuka

2.15. Pengujian Visualisasi Data

Pengujian visualisasi data dilakukan untuk melihat respon pengguna terhadap informasi-informasi dari aplikasi simulasi trayektori yang disajikan dalam bentuk GUI (*Graphical User Interface*). Pengujian ini dilakukan di lingkungan Lab DIRK UNIKOM dalam bentuk kuesioner yang terdiri dari sepuluh buah pertanyaan yang disebarkan pada 20 responden. Pada awalnya, masing-masing responden akan diberikan beberapa *task* atau tugas yang harus dilakukan terlebih dahulu terhadap aplikasi yang akan diuji. Berikut *task* yang harus responden lakukan.

Tabel 1. *Task* pengujian visualisasi data

No.	*Task*/Tugas
1	*Pairing* dua *port* untuk komunikasi data serial dengan aplikasi *VSPE*
2	*Setting* komponen aplikasi serial seperti pemilihan *port*, *baudrate* = 9600, *data bits* = 8, *stop bits* = 1, *parity* = *none*, *handshake* = *none* dan klik tombol *connect*
3	*Setting* komponen aplikasi GCS seperti pemilihan *port* dan *baudrate* = 9600. Kemudian klik tombol *connect*, *start telemetry* dan *connect GPS*
4	Kembali ke aplikasi serial, klik tombol *browse* untuk memasukkan log data pengujian Gasibu atau pengujian Ledeng
5	Log data dikirim melalui aplikasi serial dan data-datanya divisualisasikan pada aplikasi GCS
6	Setelah pengiriman data dari aplikasi serial selesai, klik tombol disconnect GPS, stop telemetry dan disconnect pada aplikasi GCS dan klik tombol X pada pojok kanan atas untuk keluar dari aplikasi
7	Klik tombol disconnect pada aplikasi serial dan klik tombol X pada pojok kanan atas untuk keluar dari aplikasi

Setelah *task* diberikan pada responden, langkah selanjutnya yang dilakukan adalah memberikan kuesioner kepada responden. Kuesioner yang diberikan terdiri dari 10 pertanyaan yang cukup untuk menunjukkan nilai aspek-aspek dari pengujian visualisasi data menurut Nielson (2000). Tiap-tiap pertanyaan dari kuesioner bertujuan untuk menunjukkan tingkat penerimaan user, yang akan dinilai dalam skala nilai 5 (skala Likert). Pertanyaan-pertanyaan yang diberikan dalam kuesioner ini adalah sebagai berikut.

Tabel 2. Pertanyaan kuesioner

No.	Pertanyaan
1.	Apakah aplikasi GCS dapat dikenali dari *interface* awal ?
2.	Apakah Anda mengetahui format data dari log data yang digunakan?
3.	Apakah Anda dapat mengirimkan log data dengan mudah ?
4.	Apakah informasi yang ditampilkan pada GCS sesuai dengan format log data yang digunakan ?
5.	Apakah tampilan teks yang ada mudah dibaca ?
6.	Apakah komponen-komponen pada GCS dapat dipahami ?
7.	Apakah desain warna dari GCS nyaman untuk dilihat ?
8.	Apakah GCS dapat membantu melihat perpindahan dari roket ?
9.	Apakah GCS dapat mempermudah proses *recovery* roket setelah uji peluncuran ?
10.	Apakah Anda dapat mengingat kembali komponen-komponen dan tampilan dari aplikasi GCS ?

Dari sepuluh pertanyaan di atas, masing-masing pertanyaan memiliki bagian tersendiri untuk mewakili tiap aspek dalam pengujian visualisasi data menurut Nielson (2000). Aspek-aspek tersebut antara lain :

a. *Learnability*, menjelaskan tingkat kemudahan pengguna untuk memenuhi *task-task* dasar ketika pertama kali mereka melihat atau menggunakan aplikasi simulasi trayektori.
b. *Efficiency*, menjelaskan tingkat kecepatan pengguna dalam menyelesaikan *task-task* setelah mereka mempelajari hasil perancangan.
c. *Memorability*, menjelaskan tingkat kemudahan pengguna dalam menggunakan rancangan yang baik, setelah beberapa lama tidak menggunakannya.
d. *Errors*, menjelaskan kemungkinan terjadinya *error* yang dilakukan oleh pengguna ketika menggunakan aplikasi simulasi trayektori.
e. *Satisfaction*, menjelaskan tingkat kepuasan pengguna dalam menggunakan aplikasi simulasi trayektori.

Kontribusi pertanyaan kuesioner untuk tiap-tiap aspek pengujian visualisasi data di atas dapat dilihat pada tabel 3 berikut, dimana masing-masing atribut ditunjukkan dengan nomor masing-masing atribut tersebut. Dan dalam pemberian kuisioner, hal ini telah dijelaskan sebelumnya oleh peneliti kepada responden.

Tabel 3. Plot aspek pengujian visualisasi data

No.	Pertanyaan	Aspek Pengujian Visualisasi Data				
		Learnability	*Efficiency*	*Memorability*	*Errors*	*Satisfaction*
1.	Apakah aplikasi GCS dapat dikenali dari *interface* awal ?	■				■
2.	Apakah Anda	■		■		

	mengetahui format data dari log data yang digunakan?					
3.	Apakah Anda dapat mengirimkan log data dengan mudah ?					
4.	Apakah informasi yang ditampilkan pada GCS sesuai dengan format log data yang digunakan ?					
5.	Apakah tampilan teks yang ada mudah dibaca ?					
6.	Apakah komponen-komponen pada GCS dapat dipahami ?					
7.	Apakah desain warna dari GCS nyaman untuk dilihat ?					
8.	Apakah GCS dapat membantu melihat perpindahan dari roket ?					
9.	Apakah GCS dapat mempermudah proses *recovery* roket setelah uji peluncuran ?					
10.	Apakah Anda dapat mengingat kembali komponen-komponen dan tampilan dari aplikasi GCS ?					

Adapun penjelasan mengenai pertanyaan-pertanyaan di atas terhadap hubungannya pada masing-masing atribut, telah dijelaskan oleh peneliti kepada responden, dengan maksud agar responden dapat mengisi kuisioner dengan benar dan sesuai dengan yang diharapkan.

Setelah dilakukan penyebaran kuisioner yang diberikan kepada responden, selanjutnya dilakukan rekap terhadap hasil kuisioner yang telah disebar. Hasil rekap pengujian kuesioner visualisasi data selengkapnya dapat dilihat pada halaman lampiran (Lampiran D).

Berikut merupakan hasil pengujian visualisasi data yang dilakukan terhadap 20 responden yang menggambarkan penyebaran jawaban dari masing-masing pertanyaan yang diberikan dalam kuisioner visualisasi data.

a. 84% responden menyatakan aplikasi GCS dapat dikenali dari *interface* awal.
b. 79% responden mengetahui format log data yang digunakan.
c. 82% responden dapat mengirimkan log data dengan mudah.
d. 88% responden menyatakan informasi yang ditampilkan sesuai dengan format log data yang digunakan.
e. 89% responden menyatakan tampilan teks yang ada mudah untuk dibaca.
f. 84% responden menyatakan komponen-komponen pada GCS dapat dipahami.
g. 84% responden menyatakan desain warna dari GCS nyaman untuk dilihat.
h. 90% responden menyatakan GCS dapat membantu melihat perpindahan dari roket.
i. 90% responden menyatakan GCS dapat mempermudah proses *recovery* roket setelah uji peluncuran.
j. 80% responden dapat mengingat kembali komponen-komponen dan tampilan dari aplikasi GCS.

Dari hasil pengujian visualisasi data di atas, dapat disimpulkan bahwa aplikasi simulasi trayektori yang dibangun memiliki nilai penerimaan (*acceptance*) yang baik dari responden dalam skala 5 dengan modus 4.

3. PENUTUP

Berdasarkan hasil pengujian aplikasi simulasi trayektori yang dilakukan, dapat disimpulkan bahwa:

a. Aplikasi yang dibangun dapat membantu para peneliti roket EDF untuk melihat pergerakan dari roket melalui visualisasi trayektori pada komponen *maps* di GCS.
b. Aplikasi yang dibangun dapat mempermudah para peneliti roket EDF untuk melakukan proses *recovery* setelah uji peluncuran roket dilakukan

UCAPAN TERIMA KASIH

Pada bagian ini, penulis ingin mengucapkan terima kasih kepada :

a. Allah SWT. yang selalu memberikan nikmat sehat sehingga penulis dapat

menyelesaikan tugas akhir ini dengan lancar.

b. Kedua orang tua yang telah membesarkan dan mendidik penulis, adik dan keluarga yang selalu memberikan motivasi, bantuan moral maupun moril dan do'a yang tidak putus-putusnya.

c. Bapak Dr. Ir. H. Eddy Soeryanto Soegoto selaku Rektor Universitas Komputer Indonesia.

d. Bapak Prof.Dr.H.Denny Kurniadie, Ir.,M.Sc selaku Dekan Fakultas Teknik dan Ilmu Komputer, Universitas Komputer Indonesia.

e. Bapak Irawan Afrianto, S.T., M.T. selaku Ketua Program Studi Teknik Informatika, Fakultas Teknik dan Ilmu Komputer, Universitas Komputer Indonesia.

f. Ibu Ken Kinanti Purnamasari, S.Kom., M.T., selaku dosen pembimbing yang telah banyak meluangkan waktu dan membimbing penulis serta memberikan saran dan kemudahan kepada penulis untuk menyelesaikan tugas akhir ini.

g. Ibu Nelly Indriani W., S.Si., M. T. selaku dosen reviewer yang telah memberikan petunjuk, saran serta kemudahaan kepada penulis.

h. Bapak Agus Mulyana, S.Kom., M. T. yang sudah banyak membantu penulis selama menempuh pendidikan di kampus saat pada saat kompetisi.

i. Bapak Dipl Ing. Endri Rachman selaku Ketua DIRK UNIKOM.

j. Bapak dan Ibu dosen serta seluruh staff pegawai di lingkungan Program Studi Teknik Informatika, Universitas Komputer Indonesia yang telah memberikan ilmu untuk menyelesaikan tugas akhir ini dan telah membantu penulis.

k. Kang Lutfi Abdussalam Al Faiz dan Kang Reza Aulia Yulnandi yang telah banyak memberikan saran dan ide-ide untuk penulis.

l. Barudak Lab DIRK UNIKOM dan barudak IF-6 angkatan 2012 yang sudah memberikan semangat dalam menyelesaikan tugas akhir ini.

m. Keluarga IKRAR Universitas Telkom yang telah memberikan tempat bagi penulis untuk berkarya dan berbagi ilmu tentang musik melayu.

n. Keluarga HIPMI Bandung yang telah memberikan semangat bagi penulis untuk menyelesaikan tugas akhir ini.

o. KPMR Bandung yang penulis dan teman-teman perjuangkan hingga saat ini dan penulis berharap organisasi provinsi ini dapat berdiri kembali di Kota Kembang.

DAFTAR PUSTAKA

[1] A. Novel, B. R. Trilaksono and R. A. Sasongko, "Guided Rocket Navigation Design and Implementation on Hardware in Loop Simulation," *2013 3rd International Conference on Instrumentation, Communications, Information Technology, and Biomedical Engineering* , p. 249, 2013.

[2] L. Nugroho, "Sistem Uji Kendali Terbang untuk Roket dan UAV," 2008.

[3] D. Tornqvist, A. Helmersson and F. Gustafsson, "Window Based GPS Integrity Test using Tight GPS/IMU Integration Applied to a Sounding Rocket".

[4] R. Reinaldo, M. Suradam, E. Andri, I. Sugihartono and H. Nasbey, "PENGEMBANGAN PROTOTIP SISTEM PEMANTAU SIKAP DINAMIK ROKET DENGAN VISUALISASI GRAFIK DAN ANIMASI PERGERAKAN 3D SECARA REAL TIME," *Seminar Nasional Fisika*, 2013.

[5] S. B. Prajitno, Metodologi Penelitian Kuantitatif, Bandung.

[6] A. Bintoro and E. Pituwarno, Buku Panduan KOMURINDO 2015, Jakarta: LAPAN, 2015.

[7] A. S. Prasidya and D. P. Laksono , "Sistem Penentuan Posisi dan Navigasi," 2014.

[8] A. Zulharman, RANCANG BANGUN KENDALI KETINGGIAN BALON ATMOSFER BERBASIS MULTIWII, Bandung: Universitas Komputer Indonesia, 2015.

[9] A. Kirk, "Showing changes over time," in *Data Visualization : a successful design process*, Birmingham, Packt Publishing Ltd, 2012, p. 137.

[10] A. Kirk, "Plotting connections and relationships," in *Data Visualization : a successful design process*, Birmingham, Packt Publishing Ltd, 2012, p. 144.

[11] B. Hariyanto, Rekayasa Sistem Berorientasi Objek, Bandung: Informatika, 2004.

[12] Suyadi, "Komunikasi Serial dan Port Serial (COM)," Teknik Informatika UMS, Solo, 2012.

[13] V. F. Kamal Al Islami, Perancangan Sistem Monitoring dan Kontrol Stabilitas pada Roket

EDF, Bandung: Universitas Komputer Indonesia, 2016.

[14] T. W. Oktaviani, Widyawan and B. S. Hantono, "PERANCANGAN USER INTERFACE BERBASIS WEB UNTUK HOME AUTOMATION GATEWAY YANG BERBASIS IQRF TR53B," pp. 271-278, 2014.