

# **APLIKASI PERANGKAT LUNAK SISTEM AKUISISI KECEPATAN REACTION WHEEL (SOFTWARE APPLICATION OF REACTION WHEEL SPEED ACQUISITION SYSTEM)**

**Harry Septanto**

**Pusat Teknologi Satelit**

**Lembaga Penerbangan dan Antariksa Nasional**

**Jl. Cagak Satelit km. 0,4 Rancabungur, Bogor 16310**

**e-mail: [harry.septanto@lapan.go.id](mailto:harry.septanto@lapan.go.id)**

**Diterima 29 Desember 2017; Direvisi 14 Februari 2018; Disetujui 20 Februari 2018**

## **ABSTRACT**

Transfer of technology activity through reverse engineering is an effort done by developing country in order to decrease technology gap, including in satellite technology, from developed country. To perform measurement and data acquisition in order to extract design information of a satellite component are the first step in reverse engineering process. This paper reports the result of research on the development of data acquisition software for satellite actuator called reaction wheel speed. In satellite development phase, the wheel speed acquisition system is also possible to be utilized as a testing tool for component level verification. The wheel speed data obtained from the experiment was represented in graphic of transient response. Based on the data analysis, the developed data acquisition system was possible to be utilized to measure dynamic characteristic of the reaction wheel unit in order to extract its design information. Meanwhile, the analysis done has represented basic part of component level verification test. This fact showed that the developed acquisition system was also possible to be used as testing tool in satellite development phase.

*Keywords: software, data acquisition system, reaction wheel, transient response*

## ABSTRAK

Kegiatan alih teknologi melalui *reverse engineering* merupakan upaya yang dilakukan oleh negara berkembang dalam rangka memperkecil jarak ketimpangan dari negara maju dalam hal teknologi, termasuk teknologi satelit. Melakukan pengukuran dan akuisisi data dalam rangka mengurai informasi desain suatu komponen satelit adalah tahap pertama proses *reverse engineering*. Makalah ini melaporkan hasil penelitian mengenai pengembangan perangkat lunak sistem akuisisi data kecepatan *wheel* dari sebuah aktuator satelit yang disebut *reaction wheel unit*. Dalam aspek fase pengembangan satelit, sistem akuisisi data ini dapat digunakan sebagai perangkat pengujian untuk verifikasi di tingkat komponen. Data kecepatan *wheel*, yang diperoleh dari eksperimen yang dilakukan, ditampilkan dengan grafik respon transien. Berdasarkan analisis atas data tersebut, sistem akuisisi data yang dikembangkan menjanjikan untuk digunakan dalam rangka pengukuran karakteristik dinamik *reaction wheel* sebagai upaya mengurai informasinya. Selain itu, analisis yang telah dilakukan merepresentasikan bagian dasar dari tahap verifikasi di tingkat komponen. Hal ini menunjukkan bahwa sistem akuisisi data ini menjanjikan untuk digunakan sebagai perangkat pengujian di dalam fase pengembangan satelit.

Kata kunci: *perangkat lunak, sistem akuisisi data, reaction wheel, respon transien*

### 1 PENDAHULUAN

Perang Dingin (*Cold War*) pasca Perang Dunia (*World War*) II antara Uni Soviet dan Amerika Serikat merupakan era di mana masing-masing pihak berusaha meyakinkan bahwa teknologi mereka adalah yang terbaik di dunia. Perlombaan penguasaan teknologi luar angkasa, yang dikenal dengan istilah *Space Race*, menjadi salah satu dampak yang dikenal pada era itu. Perlombaan itu dimenangkan oleh Uni Soviet setelah berhasil meluncurkan satelit Sputnik 1 pada 4 Oktober 1957 (Mcdougall, 1985). Keberhasilan Uni Soviet tersebut menjadi sebab Amerika Serikat kemudian melakukan banyak eksperimen terkait teknologi satelit secara cepat (House, Gruber, Hunt, & Mecherikunnel, 1986).

Sekitar lima puluh tahun setelah Sputnik 1 diluncurkan, yaitu 10 Januari 2007, satelit kelas mikro LAPAN-TUBSAT, yang dibangun melalui kerja sama antara Indonesia (Lembaga Penerbangan dan Antariksa Nasional atau LAPAN) dan Jerman (*Technische Universität Berlin*), diluncurkan dengan roket peluncur India PSLV-C7. Kegiatan pengembangan satelit oleh LAPAN dilanjutkan dengan satelit LAPAN-A2/LAPAN-ORARI yang telah diluncurkan pada 28 September 2015

dan LAPAN-A3/IPB yang telah diluncurkan pada 22 Juni 2016. Saat ini, LAPAN sedang mengembangkan dua satelit kelas mikro lainnya, yaitu satelit LAPAN-A4 dan satelit LAPAN-A5.

Amanah kemandirian dalam penyelenggaraan kementerian Indonesia di mana pengembangan satelit adalah salah satu aspeknya-menjadikan kegiatan alih teknologi (*transfer of technology*) dalam bidang satelit adalah keniscayaan. Pustaka (Prabowo, Mayditia, & Yusuf, 2009), (Saifudin, Amin, & Huzain, 2016) dan (Meidiansyah, Rosa, & Ikhsan, 2013) adalah beberapa contoh hasil dari kegiatan alih teknologi di lingkungan Pusat Teknologi Satelit LAPAN.

G. S. Prabowo dkk. (Prabowo *et al.*, 2009) melaporkan hasil rancang-bangun *wheel drive electronic* (WDE) pada sistem pembangkit torsi satelit. Modul WDE tersebut meliputi sistem mikrokontroler, antarmuka untuk komunikasi dengan sensor gyro dan sistem komputer utama satelit atau *on board data handling* dan sistem penggerak motor DC *brushless*.

Di dalam pustaka (Saifudin *et al.*, 2016), M. A. Saifudin dkk. melaporkan hasil pengembangan *star sensor* kelas eksperimental yang diterbangkan dan menjadi bagian dari sistem sensor sikap

(*attitude sensor*) satelit LAPAN-A3/IPB. Perangkat *star sensor* yang telah dikembangkan tersebut terdiri dari sistem optik (kamera dan *baffle*), sistem mikrokontroler pemroses data bintang-termasuk program pengidentifikasi bintang.

T. Meidiansyah dkk. (Meidiansyah *et al.*, 2013) melaporkan hasil pengembangan perangkat lunak yang telah direalisasikan pada Stasiun Bumi (*Ground Station*) LAPAN untuk keperluan penjejakan (*tracking*) dan pengendalian (*tele-command*) satelit LAPAN-A2/LAPAN-ORARI. Perangkat lunak tersebut mengakomodasi berbagai kebutuhan operasi satelit yang meliputi antarmuka radio modem, koreksi frekuensi akibat efek Doppler, pembaharuan parameter orbit (*two-line element* atau TLE) secara *online* dan data *log* operasi satelit.

Pada prinsipnya, kegiatan-kegiatan alih teknologi yang telah dilakukan tersebut dilakukan melalui proses *reverse engineering*, yaitu sebuah proses sistematis dalam rangka mengurai informasi desain atas suatu produk yang telah ada (Geng & Bidanda, 2017). *Reverse engineering* merupakan sebuah strategi manajemen teknologi dalam rangka memperkecil jarak dalam hal teknologi antara negara berkembang (*developing country*) dan negara maju (*developed country*) serta memiliki pangsa pasar yang sesuai dalam bisnis global (Dehaghi & Goodarzi, 2011).

Tahap pertama dari *reverse engineering* adalah pengukuran (*measurement*) dan akuisisi data (*data acquisition*) dari suatu bagian produk yang ada, di mana informasi yang terkumpul kemudian dianalisis dan diinterpretasikan (Wang, 2011). Alih-alih membahas mengenai produk aktuator atau sensor satelit, seperti pada pustaka (Prabowo *et al.*, 2009) dan (Saifudin *et al.*, 2016), atau produk untuk Stasiun Bumi, seperti pustaka (Meidiansyah *et al.*, 2013), makalah ini akan membahas mengenai produk yang dibangun dalam rangka memenuhi tahap pertama dari *reverse engineering*, yaitu pengukuran dan

akuisisi data dari suatu produk yang ada, di mana dalam hal ini yaitu pengukuran dan akuisisi data kecepatan sistem aktuator satelit *reaction wheel* dalam rangka mengetahui karakteristik dinamika sistem tersebut.

Selain sebagai upaya *reverse engineering*, pengujian dalam aspek dinamika kecepatan sistem *reaction wheel* tersebut merupakan tahap yang penting dalam pengembangan satelit. Pengujian ini termasuk ke dalam verifikasi tingkat komponen (*component level verification*) pada fase pengembangan satelit (Eickhoff, 2009).

Aplikasi perangkat lunak berbasis komputer untuk keperluan akuisisi data dari beragam sistem elektro-mekanik telah banyak dilaporkan. Beberapa penerapan aplikasi perangkat lunak pada sistem akuisisi data dilaporkan dalam pustaka (Gupta, Bera, & Mitra, 2010), (Sulistyo, Widodo, Tjokronagoro, & Soesanti, 2013), (Martín, Llopis, Rodríguez, González, & Blanco, 2014) dan (Rozikin, Sukoco, & Saptomo, 2017).

R. Gupta dkk. (Gupta *et al.*, 2010) melaporkan aplikasi perangkat lunak dalam sistem akuisisi sinyal *electrocardiogram* dengan *graphical user interface* berbasis MATLAB. Pemrograman dengan prinsip *event-driven* diterapkan dalam rangka mengotomasi akuisisi dan penyimpanan (*storage*) data serial. Seperti R. Gupta dkk., H. Sulistyo dkk. melaporkan mengenai penggunaan *Data Acquisition Toolbox* dalam MATLAB untuk sistem akuisisi sinyal *electrocardiogram* (Sulistyo *et al.*, 2013).

Di dalam pustaka (Martín *et al.*, 2014), F. J. Ferrero Martín dkk. melaporkan mengenai sistem akuisisi data berbasis perangkat keras (*hardware*) OpenDAQ yang bersifat *open source*. Sistem ini menerapkan bahasa pemrograman Python dan LabVIEW untuk aplikasi perangkat lunaknya.

Pustaka (Rozikin *et al.*, 2017) membahas mengenai sistem akuisisi data pada sistem irigasi otomatis berbasis *wireless sensor network*. Sistem ini

menggunakan aplikasi perangkat lunak Realterm untuk menerima dan menampilkan data jejaring sensor tersebut.

Aspek aplikasi perangkat lunak sistem akuisisi dalam pustaka-pustaka tersebut di atas tidak dibahas secara khusus dan mendalam. Berbeda dengan itu, makalah ini membahas secara khusus dan mendalam mengenai aspek aplikasi perangkat lunak sistem akuisisi data. Pada bagian berikutnya, pemaparan masalah, tujuan dan aspek-sepek rancangan sistem perangkat lunak diuraikan dalam bagian Metodologi. Selanjutnya, pada bagian Realisasi Perangkat Lunak diuraikan hasil penelitian berupa perangkat lunak yang telah dikembangkan. Pada bagian 4, hasil eksperimen dan analisisnya disajikan. Terakhir, makalah ini ditutup dengan bagian Kesimpulan.

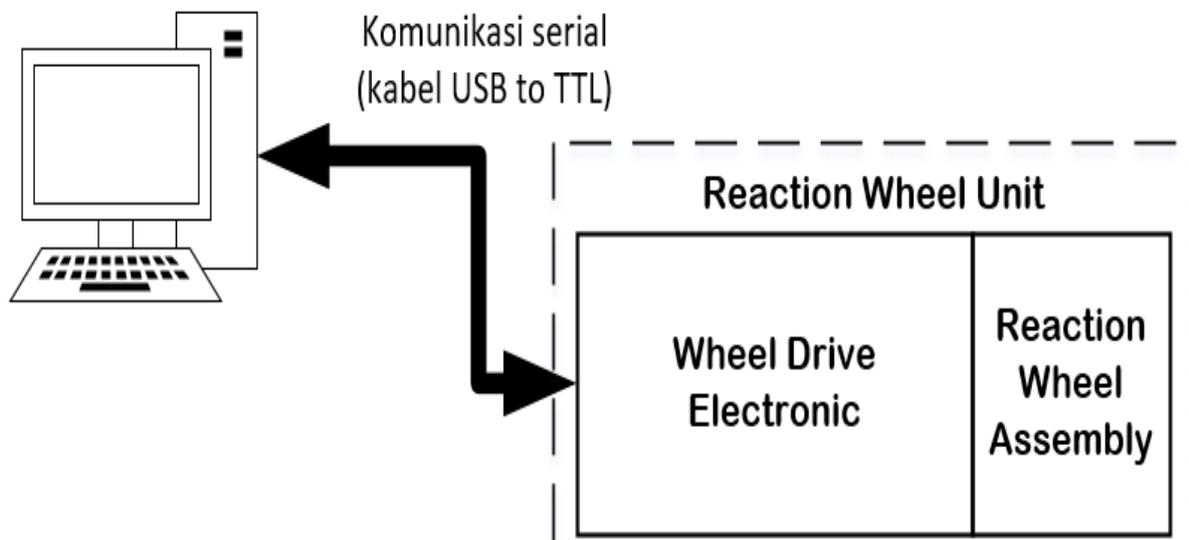
## 2 METODOLOGI

### 2.1 Definisi Masalah dan Tujuan

Dalam menjalankan misinya, misalnya mengarahkan antena untuk keperluan pengiriman data citra ke Stasiun Bumi, satelit memerlukan aktuasi untuk mengendalikan sikap (*attitude control*). Aktuasi kendali tersebut berupa torsi atau momentum angular yang dibangkitkan oleh perangkat elektro-

mekanik yang disebut *reaction wheel unit* (Gambar 2-1). Untuk kemudahan pembaca, pada makalah ini *reaction wheel unit* disebut sebagai sistem *reaction wheel* atau Sistem RW.

Sistem *bus* satelit yang diterapkan pada satelit LAPAN selama ini, yaitu LAPAN-A1, A2, dan A3, memiliki tiga Sistem RW yang ditempatkan di ketiga sumbu satelit. Setiap Sistem RW tersebut terdiri dari dua bagian yaitu *Reaction Wheel Assembly* (RWA) dan *Wheel Drive Electronic* (WDE). Pada prinsipnya, RWA terdiri dari motor arus searah tanpa sikat atau *Brushless Direct Current* (BLDC) dan massa beban (*load mass*) dengan momen inersia tertentu yang dipasang pada poros (*shaft*) motor-bagian ini disebut *wheel*. Sementara itu, WDE merupakan komponen elektronik yang menggerakkan komponen motor BLDC pada RWA sehingga *wheel* berputar sesuai dengan torsi atau momentum angular yang diinginkan. Selain itu, setiap Sistem RW tersebut dilengkapi dengan sebuah sensor gyro dalam rangka menjalankan aksi kendali *closed-loop* kecepatan dan sudut rotasi masing-masing sumbu satelit. Informasi mengenai diagram Sistem RW yang dipaparkan di atas dapat diperoleh pada Gambar 2-1 pada pustaka (Butz & Renner, 1996).



Gambar 2-1: Skema *hardware* sistem akuisisi data kecepatan rotasi *wheel*

Selain yang telah disebutkan, Sistem RW tersebut juga memiliki kemampuan aksi kendali *open-loop* yang meliputi kendali kecepatan rotasi motor BLDC atau kecepatan *wheel* di mana kinerja kendali *open-loop* ini pada prinsipnya akan berimplikasi pada kinerja *closed-loop* tadi. Perlu diperhatikan bahwa terminologi *open*-dan *closed-loop* tersebut adalah dalam sudut pandang sistem satelit di mana Sistem RW sebagai aktuator, gyro sebagai sensor dan satelit sebagai kendalian (*plant*); dalam sudut pandang motor sebagai *plant*, pengendalian kecepatan *wheel* dapat terjadi secara *closed-loop* dengan memanfaatkan sensor Hall dalam mengukur kecepatan rotasi motor aktualnya.

Dengan memperhatikan paparan di atas, perancangan sistem kendali *open-loop* ini merupakan tahap pertama yang harus dilakukan sebelum kemudian merancang sistem kendali *closed-loop*-nya. Selanjutnya, memastikan bahwa sistem kendali yang memiliki kinerja yang dibutuhkan sangat penting untuk menjamin sistem kendali yang dirancang dapat mendukung operasi misi satelit.

Selain merujuk syarat ideal dalam formulasi teoretis, studi komparasi dengan perangkat yang sudah praktis bekerja (*flight proven*) mengambil bagian penting dalam upaya *reverse engineering*, dalam hal ini adalah Sistem RW. Untuk itu, perangkat untuk mengukur kinerja Sistem RW yang ada tersebut harus dibangun. Penelitian yang dilakukan ini bertujuan untuk merealisasikan perangkat lunak aplikasi desktop (*desktop application software*) untuk keperluan pengukuran kinerja Sistem RW, yaitu sistem akuisisi kecepatan *wheel*. Di samping itu, penelitian ini juga merupakan upaya penyediaan perangkat pengujian kinerja Sistem RW pada fase verifikasi tingkat komponen.

Perangkat keras (*hardware*) yang digunakan pada sistem akuisisi tersebut terdiri dari komputer, Sistem RW dan kabel konverter USB ke TTL untuk pengiriman perintah dan penerimaan

data kecepatan rotasi *wheel*. Selain itu, pada Sistem RW itu sendiri, terdapat beberapa koneksi atau pengkabelan antara WDE dan RWA yang pembahasannya di luar lingkup makalah ini. Skema *hardware* tersebut ditampilkan pada Gambar 2-1.

## 2.2 Aspek-aspek dalam Rancang-Bangun Perangkat Lunak

Sistem RW yang digunakan dalam penelitian ini merupakan Sistem RW yang berasal dari vendor. Beberapa aspek yang diperhatikan dalam rancang-bangun sistem akuisisi data kecepatan *wheel* Sistem RW tersebut meliputi pemahaman format data dan pemberian perintah (*command*); penentuan fitur-fitur yang harus diakomodasi pada perangkat akuisisi data; dan penentuan bahasa pemrograman.

Dua tahap umum yang harus direalisasikan pada pemrograman dalam rangka melakukan akuisisi kecepatan *wheel* yaitu mengirimkan perintah (*command*) ke WDE untuk memutar *wheel* dengan kecepatan tertentu, lalu mengirimkan perintah ke WDE secara berulang untuk mengukur (*telemetry*) kecepatan *wheel*.

Perintah yang harus dikirimkan (*transmitted*) ke WDE merupakan paket byte tertentu berupa array, di mana byte pertama (*header*) merepresentasikan jenis perintah dan *byte* terakhir merupakan byte penanda akhir paket *byte* (*tail*). Pada perintah untuk mengeset kecepatan *wheel*, nilai kecepatan bertipe *Single* yang telah dikonversi menjadi empat *byte* dibawa di antara *header* dan *tail* tersebut. Sementara, pada perintah *telemetry*, nilai-nilai *byte* di antara *header* dan *tail* diabaikan.

Paket byte perintah atau *array-byte command* tersebut dikirimkan dengan prinsip *echo*, di mana setiap byte yang dikirim oleh sistem akuisisi ke WDE akan dikirim balik oleh WDE. WDE memproses *command* setelah *tail* dari *array-byte command* dibaca dan format paket *command* benar. WDE akan

memberikan suatu respon *byte* apabila salah satu atau kedua kondisi tersebut tidak terpenuhi. WDE akan memproses *command* untuk mengeset kecepatan *wheel* dengan segera memutar *wheel* ke nilai kecepatan yang diset. Sementara pada *command* untuk *telemetry*, WDE memproses *command* dengan mengirimkan sejumlah *byte* yang diakhiri dengan *byte tail* dari *array-byte command*.

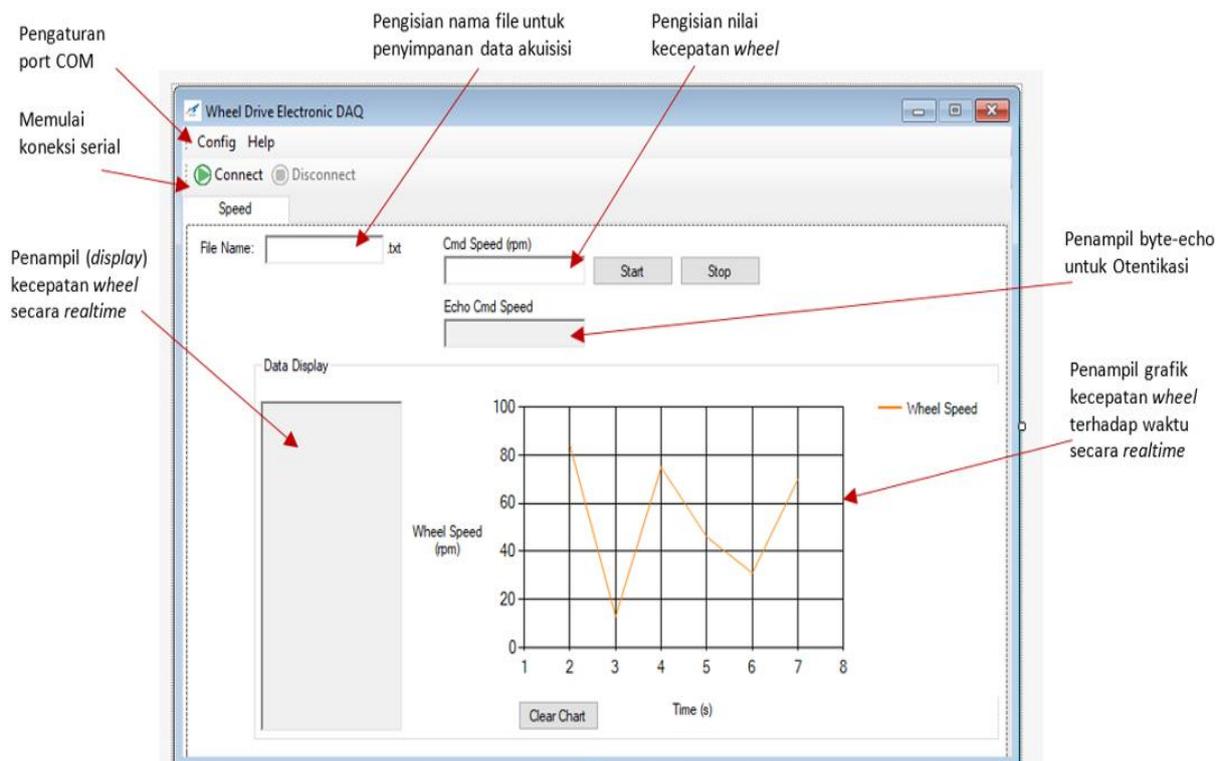
Dengan pertimbangan kondisi pengiriman perintah di atas, maka, untuk keperluan *user* melakukan otentikasi langsung secara visual, perangkat sistem akuisisi data tersebut perlu mengakomodasi tampilan yang menyatakan bahwa *array-byte* perintah yang dikirimkan adalah benar. Selain fitur (*feature*) itu, yang penting untuk direalisasikan dalam perangkat sistem akuisisi tersebut meliputi adanya tampilan grafik secara waktu-nyata (*realtime*) untuk observasi secara langsung dan penyimpanan (*save to file*) atas data hasil akuisisi untuk keperluan analisis dinamika sistem lebih lanjut.

Bahasa pemrograman yang digunakan untuk merealisasikan

perangkat lunak sistem akuisisi data kecepatan rotasi *wheel* adalah Visual Basic.NET pada Visual Studio Community 2015 yang tidak berbayar. Antar-muka pengguna (*user interface*) direalisasikan dengan menggunakan *Windows Forms Application*. Selain itu, misalnya, prinsip *event-driven* dalam rangka komunikasi dengan WDE (proses pengiriman perintah dan penerimaan data) diterapkan dengan bahasa pemrograman tersebut.

### 3 REALISASI PERANGKAT LUNAK

Hasil penelitian yang dilakukan ini adalah perangkat lunak sistem akuisisi data kecepatan *wheel* yang memiliki antar-muka pengguna (*user interface* atau UI). Realisasi UI perangkat lunak sistem akuisisi tersebut dalam “*View Designer*” Visual Studio ditampilkan pada Gambar 3-1. UI sistem akuisisi tersebut meliputi menu konfigurasi komunikasi serial untuk pengaturan port COM, dengan nilai *baudrate*, banyaknya *data bits*, status *parity bit* dan banyaknya *stop bit* yang telah diketahui dan menjadi bagian dari kode program.



Gambar 3-1: Tampilan antar-muka pengguna (*user interface*) dalam menu “*View Designer*” di Visual Studio

UI sistem akuisi ini memiliki dua kotak-teks (*textbox*) pengisian, yaitu “*File Name*” untuk pengaturan nama file penyimpanan data akuisisi dan “*Cmd Speed (rpm)*” untuk pengaturan perintah nilai kecepatan rotasi *wheel* dalam rpm. Selain dua kotak-teks pengisian, pada UI tersebut ada kotak-teks penampil (*display*) untuk otentikasi nilai *byte* secara langsung oleh *user* bahwa perintah yang dikirimkan benar, yaitu “*Echo Cmd Speed*”. Sementara di dalam “*Data Display*” ada dua macam penampil data kecepatan *wheel* secara *realtime*, dalam bentuk teks pada kotak-teks dan dalam bentuk grafik-garis solid oranye pada penampil grafik dalam Gambar 3-1 hanya berupa ilustrasi.

Pemrograman sistem akuisisi ini direalisasikan dengan sebuah *Public Class* beberapa *Module*, yaitu *Module* yang berkaitan dengan aksi menu konfigurasi port COM “*Config*” dan “*Help*”, *Module* yang berkaitan dengan aksi menu koneksi serial, dan *Module* yang berkaitan dengan aksi penanganan data yang diakuisisi. Namun demikian, pemaparan pada makalah ini hanya difokuskan pada bagian yang menangani secara langsung akuisisi data kecepatan *wheel*, yaitu *Public Class MainForm* dan *Module Parsing* yang melakukan penanganan data yang diakuisisi.

Diagram alir yang ditampilkan pada pada Gambar 3-2 merepresentasikan alur kerja sekaligus posisi realisasi kode program yang menangani setiap langkah pada alur kerja tersebut, yaitu di dalam *Public Class MainForm* atau di dalam *Module Parsing*. Pada gambar tersebut ada tiga warna simbol-yaitu putih, biru, dan kuning-dan dua warna anak panah-biru dan kuning. Arti warna-warna tersebut dijelaskan pada Tabel 3-1 dan Tabel 3-2.

Pada pendefinisian *array-byte command* untuk mengeset kecepatan *wheel*, nilai kecepatan yang akan dikirimkan diambil dari nilai pada kotak teks “*Cmd Speed (rpm)*” yang diisi oleh *user*. Tabel 3-3 menampilkan *pseudocode*

proses tersebut dalam rangka melengkapi diagram alir pada Gambar 3-2.

Tabel 3-1: ARTI WARNA SIMBOL PADA DIAGRAM ALIR GAMBAR 3-2

Warna Simbol	Arti
Biru	Aksi yang direalisasikan dengan kode program
Putih	Aksi yang dilakukan oleh <i>user</i>
Kuning	Aksi dilakukan oleh WDE

Keterangan: Simbol yang dimaksud adalah selain simbol lingkaran bernomor. Simbol lingkaran bernomor menunjukkan bahwa dua anak panah dengan nomor yang sama di dua bagian diagram alir pada Gambar 3-2 saling terhubung.

Tabel 3-2: ARTI WARNA ANAK PANAH PADA DIAGRAM ALIR GAMBAR 3-2

Warna Anak Panah	Arti
Biru	Alur kerja user dan program
Kuning	Komunikasi serial

Tabel 3-3: PSEUDOCODE PENDEFINISIAN ARRAY-BYTE COMMAND UNTUK MENGESET KECEPATAN WHEEL

```

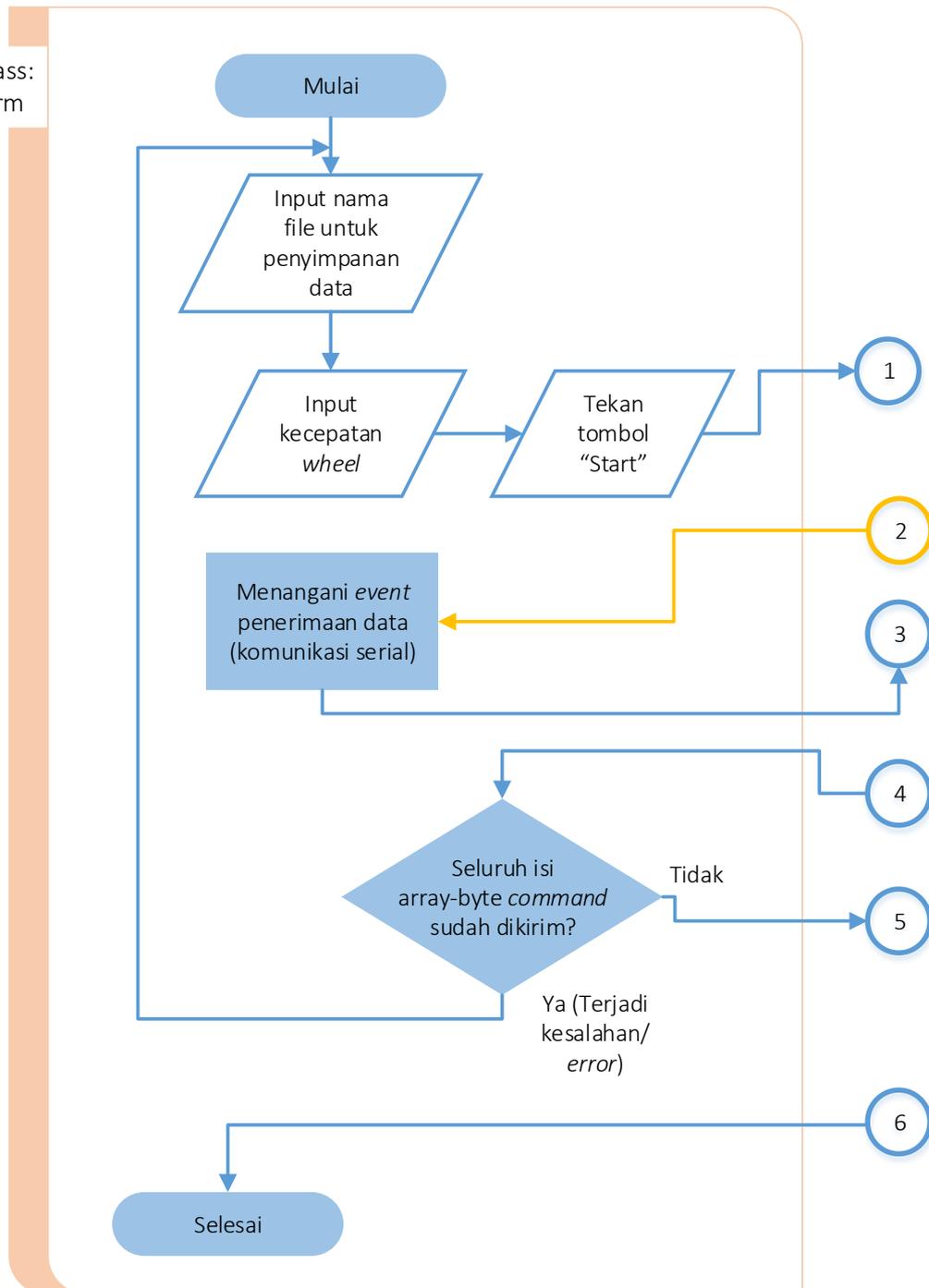
Procedure:
  Deklarasi variabel inp sebagai Single
  Deklarasi variabel inpByte(3) sebagai array Byte
  inp = hasil konversi nilai kecepatan wheel dari kotak teks dari tipe String ke Single
  inpByte = hasil konversi inp ke array Byte
  Membalikkan urutan isi inpByte
  Mendefinisikan array-byte command untuk set kecepatan wheel
End Procedure
    
```

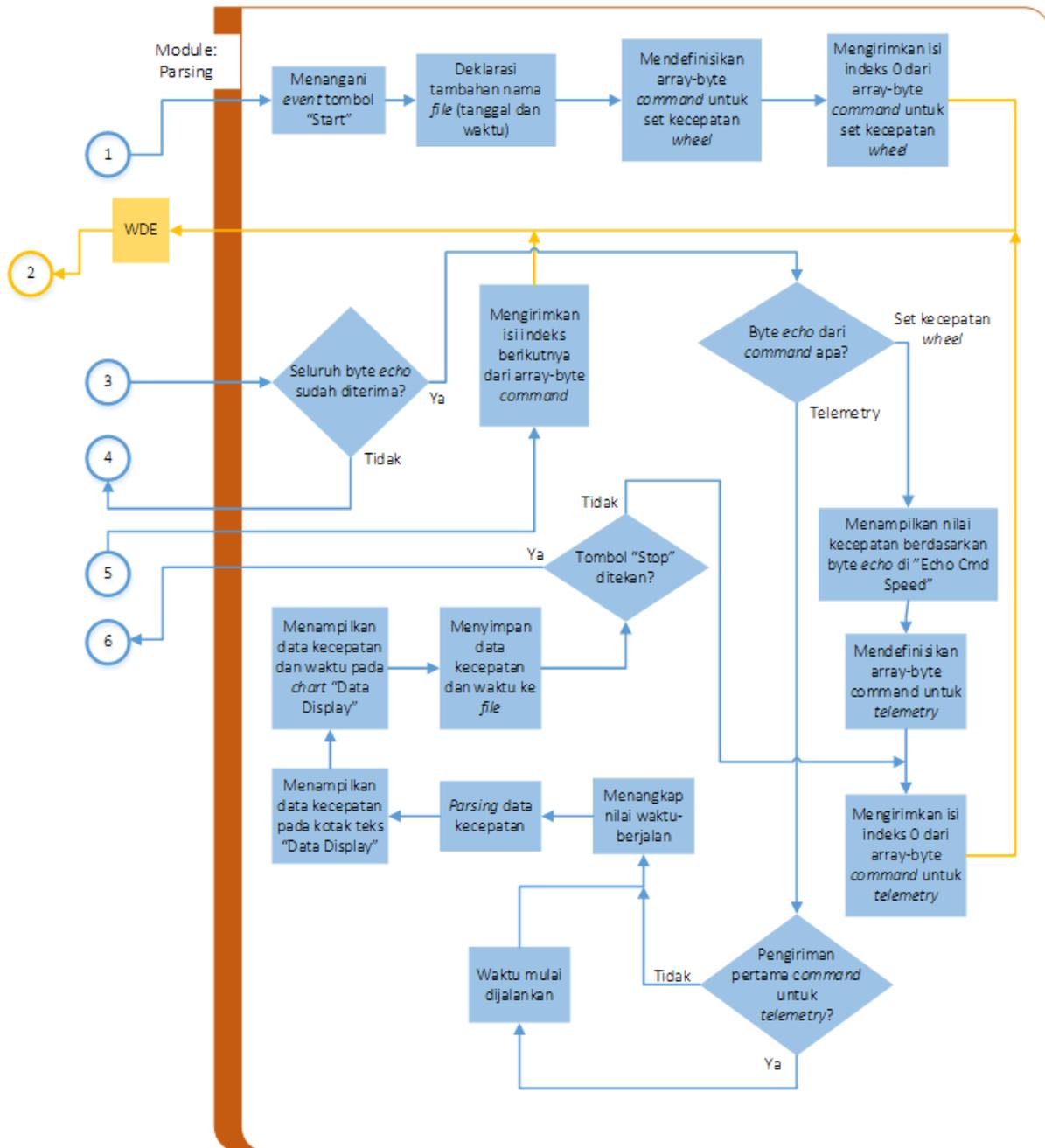
Pada Tabel 3-3, pernyataan untuk membalikkan urutan isi variabel array *inpByte*—yang akan mengisi *array-byte command* antara *byte header* dan *tail*—direalisasikan dalam pemrograman akibat adanya perbedaan *Endianess* antara prosesor atau mikrokontroler WDE dan prosesor komputer di mana perangkat lunak sistem akuisisi kecepatan *wheel* ini dijalankan.

Penyesuaian urutan *byte* juga direalisasikan pada pembacaan *byte-byte echo* di antara *echo byte header* dan *tail*, yaitu pada bagian pemrograman untuk menampilkan kecepatan di “Echo Cmd Speed” dan *parsing* data kecepatan. Pada

bagian *parsing* data kecepatan tersebut, empat *byte* yang merepresentasikan kecepatan dipisahkan dari sejumlah *byte* yang dikirimkan oleh WDE sebagai respon dari *command* untuk *telemetry*.

Public Class:  
MainForm





Gambar 3-2: Alur kerja sistem akuisisi data kecepatan rotasi *wheel* dan posisi realisasi kode program

#### 4 EKSPERIMEN DAN DISKUSI

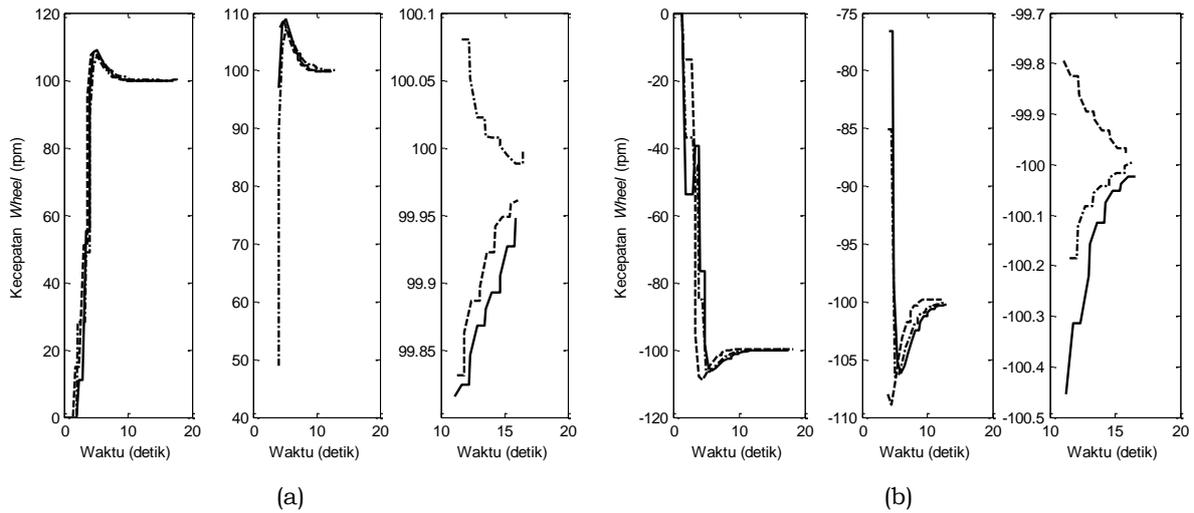
Pembahasan pada bagian ini akan berfokus pada hasil eksperimen perangkat lunak sistem akuisisi kecepatan *wheel*, yaitu data kecepatan *wheel* yang disimpan dalam file dan analisisnya. Sebagian besar isi bagian ini dikutip dari (Septanto, 2016).

Eksperimen dilakukan dengan parameter sebagai berikut:

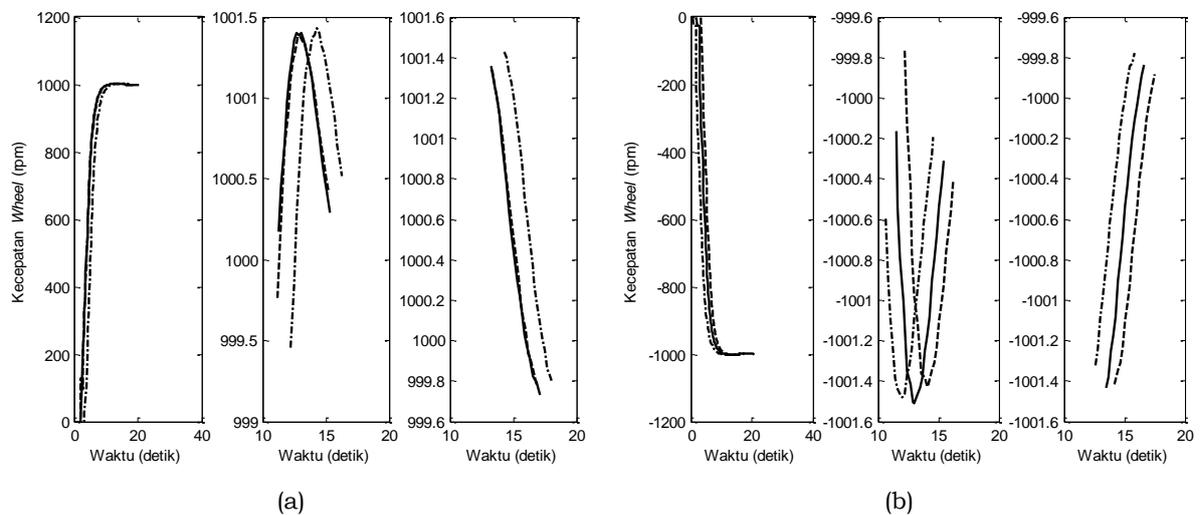
- nilai kecepatan (dalam rpm) yang diset yaitu +100, -100, +500, -500, +5000, dan -5000,

- akuisisi data kecepatan dilakukan sebanyak tiga kali untuk setiap nilai kecepatan yang diset

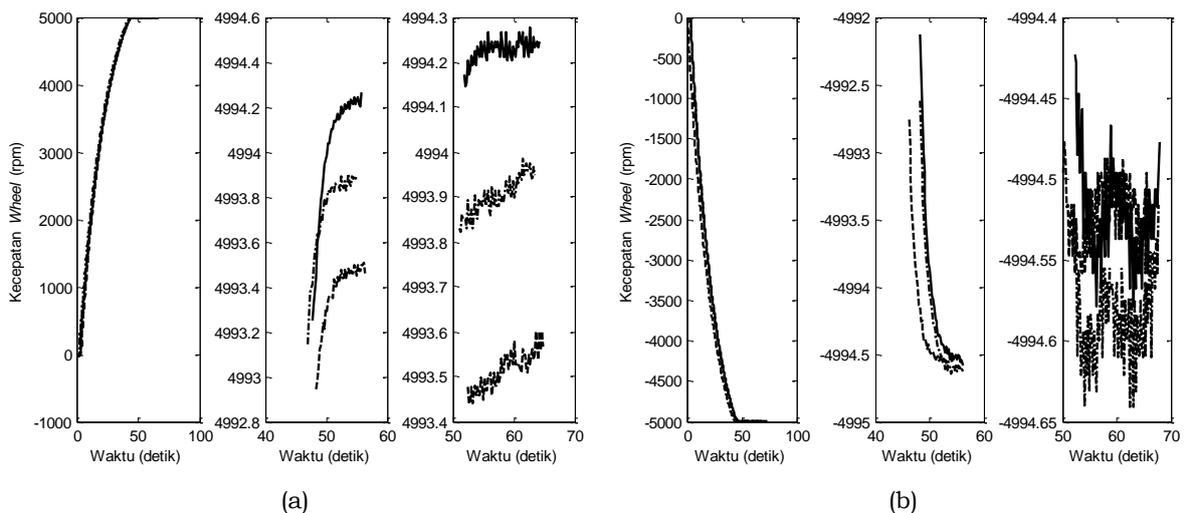
Data kecepatan *wheel* yang disimpan dalam *file* kemudian diplot untuk mengetahui dinamika *wheel* melalui grafik respon transien yang ditampilkan pada Gambar 4-1, Gambar 4-2, dan Gambar 4-3. Setiap grafik tersebut menampilkan hasil akuisisi data kecepatan dengan tiga kali pengulangan, yaitu dengan garis *solid* (-), garis *dash* (--), dan garis *dotted dash* (-.-).



Gambar 4-1: Respon transien kecepatan *wheel* untuk nilai set (a) +100 dan (b) -100



Gambar 4-2: Respon transien kecepatan *wheel* untuk nilai set (a) +500 dan (b) -500



Gambar 4-3: Respon transien kecepatan *wheel* untuk nilai set (a) +5000 dan (b) -5000

Berdasarkan grafik-grafik tersebut, kinerja Sistem RW dalam aspek kecepatan *wheel* dapat diketahui. Semakin besar nilai kecepatan yang diset, nilai *overshoot*

yang terjadi semakin kecil. Gambar 4-1 menampilkan *overshoot*, dengan variasi nilai dari tiga kali pengukuran untuk nilai set kecepatan +100 dan -100 rpm,

sebesar sekitar 6 hingga mendekati 10 rpm. Pada Gambar 4-2, *overshoot* masih terlihat namun lebih kecil. Sementara Pada Gambar 4-3, *overshoot* tidak ada.

Sementara pada aspek galat (*error*) terhadap nilai kecepatan yang diset, semakin besar nilai kecepatan yang diset, galat yang terjadi semakin besar. Gambar 4-3 menampilkan bahwa galat yang terjadi adalah sekitar 6 rpm. Gambar 4-2 dan Gambar 4-1 menampilkan nilai galat yang semakin kecil, yaitu, secara berurutan, sekitar 1 dan 0,1 rpm.

Terakhir, seperti telah disebutkan di atas, akuisisi data kecepatan dilakukan sebanyak tiga kali untuk setiap nilai kecepatan yang diset. Hasil pengukuran menunjukkan bahwa untuk setiap nilai kecepatan yang diset, tidak ada yang menunjukkan bahwa ketiga akuisisi data yang dilakukan menghasilkan respon transien yang sama persis. Dalam sudut pandang pemodelan sistem dinamik, hal ini kemungkinan besar akibat hadirnya elemen nonlinier, baik itu pada motor BLDC maupun pada komponen elektroniknya. Namun demikian, perbedaan hasil pengukuran yang terjadi tidak berbeda secara signifikan.

## 5 KESIMPULAN

Perangkat lunak sistem akuisisi data kecepatan rotasi *wheel* sebagai upaya melakukan *reverse engineering* dan penyediaan perangkat pengujian untuk verifikasi tingkat komponen telah direalisasikan. Eksperimen telah dilakukan dan hasil eksperimen berupa data kecepatan *wheel* yang disimpan dalam *file* telah disajikan. Variabel *overshoot* dan *error* kecepatan yang dibahas dalam analisis menunjukkan bahwa perangkat lunak yang direalisasikan menjanjikan untuk digunakan dalam rangka mengetahui dinamika Sistem RW sebagai upaya mengurai informasi desainnya. Di samping itu, analisis yang dilakukan telah merepresentasikan pengukuran kinerja Sistem RW, sebagai bagian dari verifikasi di tingkat komponen.

Penelitian selanjutnya yang menarik untuk dilakukan setidaknya ada dua, yaitu membuat *hardware in the loop simulator* yang menggabungkan aktuator Sistem RW dan simulator sensor bintang (*star tracker sensor*) (Ubay & Saifudin, 2014) dan pengembangan fitur perangkat lunak ini sehingga perhitungan kinerja seperti *overshoot* dan *error* langsung ditampilkan pada perangkat lunak tersebut.

## UCAPAN TERIMA KASIH

Penulis sangat berterima kasih atas kerjasama dalam Tim Eksperimen *Attitude Control System 2016*. Penulis juga berterima kasih kepada Abdul Karim, MT yang telah mendukung tersedianya fasilitas penelitian ini. Untuk berbagai masukan dari *blind reviewer* dalam rangka membuat tulisan ini menjadi lebih baik, penulis ucapkan terima kasih.

## DAFTAR RUJUKAN

- Butz, P., & Renner, U., 1996. *TUBSAT-C, a Microsat-bus for Earth Observation Payloads*. In 3rd International Symposium of Small Satellites Systems and Services.
- Dehaghi, M. R., & Goodarzi, M., 2011. *Reverse Engineering – A Way of Technology Transfer in Developing Countries Like Iran*. In International Journal of e-Education, e-Business, e-Management and e-Learning (Vol. 1, 347–353).
- Eickhoff, J., 2009. *Simulation Tools for System Analysis and Verification*. In *Simulating Spacecraft Systems* (23–54). Springer-Verlag Berlin Heidelberg. <http://doi.org/10.1007/978-3-642-01276-1>.
- Geng, Z., & Bidanda, B., 2017. *Review of Reverse Engineering Systems–Current State of the Art*. *Virtual and Physical Prototyping*, 0(0), 1–12. <http://doi.org/10.1080/17452759.2017.1302787>.
- Gupta, R., Bera, J. N., & Mitra, M., 2010. *Development of an Embedded System and MATLAB-Based GUI for Online Acquisition and Analysis of ECG Signal*. *Measurement*, 43(9), 1119–1126.

- <http://doi.org/10.1016/j.measurement.2010.05.003>.
- House, F. B., Gruber, A., Hunt, G. E., & Mecherikunnel, A. T., 1986. *History of Satellite Missions and Measurements of the Earth Radiation Budget (1957-1984)*. *Reviews of Geophysics*, 24(2), 357–377.
- Martín, F. J. F., Llopis, M. V., Rodríguez, J. C. C., González, J. R. B., & Blanco, J. M., 2014. *Low-Cost Open-Source Multifunction Data Acquisition System for Accurate Measurements*, 55, 265–271. <http://doi.org/10.1016/j.measurement.2014.05.010>.
- Mcdougall, W. A., 1985. *Sputnik, the Space Race, and the Cold War*. *Bulletin of the Atomic Scientists*, 41(5), 20–25. <http://doi.org/10.1080/00963402.1985.11455962>
- Meidiansyah, T., Rosa, W., & Ikhsan, M. Y., 2013. *Pengembangan Software Stasiun Bumi LAPAN-ORARI/A2*. In *Pengembangan Teknologi Satelit di Indonesia: Sistem, Subsistem dan Misi Operasi* (63–78).
- Prabowo, G. S., Mayditia, H., & Yusuf, A., 2009. *Analisis Desain Wheel Drive Electronic (WDE)-LAPAN untuk Pengembangan Satelit LAPAN-A2*. *Jurnal Teknologi Dirgantara*, 7(1), 1–10.
- Rozikin, C., Sukoco, H., & Saptomo, S. K., 2017. *Sistem Akuisisi Data Multi Node untuk Irigasi Otomatis Berbasis Wireless Sensor Network*. *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi*, 6(1), 43–50.
- Saifudin, M. A., Amin, D. El, & Huzain, M. F., 2016. *Development of Experimental Star Sensor for LAPAN-A3/IPB Microsatellite*. In *Advances in Aerospace Science and Technology in Indonesian Vol. I* (8–16).
- Septanto, H., 2016. *Laporan Teknis Bulan November*. Pusat Teknologi Satelit LAPAN.
- Sulistyo, H., Widodo, T. S., Tjokronagoro, M., & Soesanti, I., 2013. *Akuisisi Data dan Pengolahan Isyarat Elektrokardiograf Menggunakan Modul USB Dataq DI-148U*. *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi*, 2(4), 68–74.
- Ubay, M. S. N., & Saifudin, M. A., 2014. *Analysis Software Star Simulator for Star Sensor Testing*. *Jurnal Teknologi Dirgantara*, 12(2), 92–101.
- Wang, W., 2011. *Reverse Engineering: Technology of Reinvention*. CRC Press.