

Pengembangan Aplikasi Multiplatform untuk Model Sistem Kendali dan Akuisisi Data Berbasis PC

(Development of Multiplatform Application for The Model of PC-Based Control and Data Acquisition System)

Arief Hendra Saptadi

*Program Studi D3 Teknik Telekomunikasi
Sekolah Tinggi Teknologi Telematika TELKOM
Jl. D. I. Panjaitan No. 128 Purwokerto
ariefhs@stttelematikatelkom.ac.id*

Abstrak—Sistem kendali dan akuisisi data berbasis PC memungkinkan komunikasi dua arah antara komputer dengan perangkat akuisisi data. Aplikasi yang berjalan di PC berfungsi untuk menerima data dari perangkat akuisisi data dan memberikan perintah kepadanya. Kendati pun demikian, aplikasi ini pada umumnya hanya berjalan di sistem operasi tertentu dengan antarmuka yang memanfaatkan bentuk-bentuk grafis yang sudah ada. Tujuan dari penelitian ini adalah mengembangkan aplikasi multiplatform untuk model sistem kendali dan akuisisi data berbasis PC dengan bentuk-bentuk grafis yang dirancang sendiri. Seluruh objek grafis dan program utama dikembangkan dengan bahasa pemrograman Processing. Antarmuka aplikasi disusun dari objek-objek grafis sederhana melalui beberapa rutin terpisah. Berdasarkan pengujian yang dilakukan di Windows maupun Linux, perangkat akuisisi data telah dapat mengirimkan data melalui komunikasi serial ke PC. Seluruh modul pada aplikasi telah berjalan dengan baik. Tampilan pada aplikasi telah dapat dikembangkan sebagaimana pada rancangan. Uji fungsionalitas memperlihatkan semua fungsi pada program telah dapat dijalankan. Secara keseluruhan, aplikasi mampu beroperasi dengan baik di lingkungan Windows dan Linux. Untuk membuat objek-objek grafis dengan pendekatan yang lebih terstruktur, seluruh rutin yang ditulis dapat dimasukkan ke dalam suatu berkas pustaka yang terpisah. Pada pengembangan mendatang komunikasi data antara perangkat akuisisi data dengan PC dapat disertai proses *handshake*.

Kata-kata kunci—aplikasi multiplatform, kendali, akuisisi data.

Abstract—PC-based control and data acquisition system had enabled two-way communication between computer and data acquisition (DAQ) device. Application running on a PC serves to receive data from DAQ device and to deliver command to it. Despite of that, this application in general runs only on a particular operating system with an interface that utilizes existing graphical shapes. The purpose of this research is to develop multiplatform application for a model of PC-based control and data acquisition system with self-designed graphical shapes. All graphical objects and main program were developed with Processing programming language. Application interface was composed of simple graphical objects through several separate routines. Based on tests conducted in Windows and Linux, data acquisition device had been able to transmit data via serial communication to the PC. All modules on the application had run well. The appearance of application had been successfully developed as it was previously designed. Functionality test had shown that all functions in the program were able to be operated. Overall, the application had operated well in Windows and Linux environment. To create graphical objects with more structured approach, all written routines can be enclosed in a separate library file. In future development, data communication between data acquisition device and PC can utilize handshake process.

Keywords—multiplatform application, control, data acquisition

I. PENDAHULUAN

Sistem akuisisi data berbasis PC adalah seperangkat komputer yang digunakan untuk mengambil data pengukuran parameter fisik yang kemudian disimpan dan dianalisis lebih lanjut. Sistem ini lazimnya terdiri dari sensor atau transduser, sinyal, rangkaian pengondisi sinyal, perangkat akuisisi data (DAQ) dan komputer (PC). Perangkat akuisisi data memiliki peran penting dalam mengubah sinyal analog yang terukur menjadi sinyal digital melalui ADC (*Analog-to-Digital Converter*). Sedangkan komputer memegang peranan dalam mengambil data pengukuran dari berbagai perangkat akuisisi data yang terhubung kepadanya. Komunikasi antara perangkat akuisisi data dengan komputer secara garis besar diwujudkan dalam dua mode yaitu paralel dan serial. Ada pun untuk mode serial terdapat berbagai macam jenis komunikasi antara lain I²C (*Inter-Integrated Circuit*), SPI (*Serial Peripheral Interface*) dan yang paling sering digunakan adalah USART (*Universal Synchronous Asynchronous Receiver/Transmitter*). Pada masa mendatang, seiring dengan berkembangnya perangkat bergerak, maka media komunikasi untuk akuisisi data akan beralih ke mode nirkabel. Pengecapan waktu (*time stamping*) pada data disertai lokasi pengambilan data yang diambil dari GPS (*Global Positioning System*) juga akan menjadi bagian dalam proses akuisisi data [1].

Penambahan fitur pengendalian perangkat ke dalam sistem akuisisi data membutuhkan kemampuan beroperasi secara dua arah. Data hasil pengukuran dikirimkan dari perangkat akuisisi data menuju PC, sementara sinyal perintah diberikan dari PC menuju sistem akuisisi data. Dalam kaitan ini, peranan aplikasi yang berjalan di dalam PC menjadi penting. Aplikasi tersebut harus menyediakan antarmuka untuk menjalankan fungsi pengendalian maupun akuisisi data. Dengan adanya berbagai macam sistem operasi untuk PC, maka kemampuan untuk beroperasi secara multiplatform menjadi semakin dibutuhkan.

Ada pun karakteristik multiplatform itu sendiri dapat mengambil berbagai bentuk. Sebuah aplikasi yang dikembangkan untuk sistem akuisisi data nuklir menyediakan antarmuka manusia dengan mesin (*Human Machine Interface*). Aplikasi tersebut dibangun dengan Java sehingga mampu berjalan di berbagai sistem operasi yang mendukungnya. Sedangkan fitur pembuatan grafik dibuat melalui bahasa C dan bahasa Assembly diperlukan untuk mengakses mesin [2]. Kendati pun demikian sistem ini tidak mendukung fitur pengendalian sehingga

hanya beroperasi satu arah. Jika sistem akuisisi data terdiri dari berbagai bagian yang terpisah maka dibutuhkan beberapa macam aplikasi yang menjalankan bermacam jenis komunikasi data. Sistem semacam ini dijumpai dalam metode injeksi berkas cahaya netral untuk reaksi fusi nuklir. Aplikasi kendali dan instrumentasi pada MAST *Neutral Beam Injection System* menggunakan perangkat SCADA (*Supervisory Control and Data Acquisition System*) dengan PLC (*Programmable Logic Controllers*) dan sistem pewaktu berbasis PC [3]. Aplikasi tersebut sebenarnya hanya berjalan di sistem operasi tertentu namun memiliki kemampuan untuk berkomunikasi antar bagian melalui berbagai jenis protokol. Pada penerapan di bidang fisika partikel, sistem akuisisi data menggunakan *framework* aplikasi EUDAQ untuk pengumpulan data dari teleskop. Aplikasi menyediakan antarmuka yang terpusat dan memungkinkan pengguna untuk memantau melalui koneksi TCP/IP [4]. Melalui PC dengan sistem operasi yang berbeda-beda, pengguna juga dapat memberikan perintah ke perangkat pengumpul data, namun sebatas permintaan data, bukan untuk mengoperasikan perangkat secara langsung.

Berdasarkan beberapa penelitian sebelumnya seperti yang sudah dipaparkan tersebut, belum ada aplikasi multiplatform yang memiliki karakteristik: dapat dijalankan pada berbagai sistem operasi berbeda, pengguna tidak hanya mampu menerima data tetapi juga memberikan perintah untuk mengoperasikan perangkat dan tampilan antarmuka sepenuhnya dirancang dari dasar. Tujuan penelitian ini adalah mengembangkan aplikasi multiplatform untuk model sistem kendali dan akuisisi data dengan bentuk-bentuk antarmuka yang sepenuhnya dirancang oleh pengguna.

II. METODE

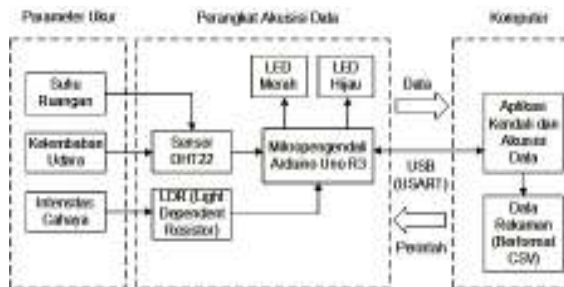
Penelitian ini secara keseluruhan melewati empat tahapan terpisah yang meliputi:

A. Perancangan Sistem

Sistem kendali dan akuisisi data secara keseluruhan terdiri dari tiga bagian yaitu parameter ukur, perangkat akuisisi data dan komputer seperti dalam Gambar 1. Parameter yang diukur dalam sistem ini dicontohkan berupa suhu ruangan, kelembaban udara dan intensitas cahaya. Dua parameter pertama diukur melalui sensor DHT22, sedangkan parameter terakhir melalui LDR (*Light Dependent Resistor*).

Bagian inti dari perangkat akuisisi data adalah mikropengendali Arduino Uno R3. Selain sensor, terdapat dua macam LED yang terhubung dengan mikropengendali. Keduanya berfungsi sebagai indikator untuk menandakan ada atau tidaknya sinyal perintah yang diterima dari komputer. Komunikasi data yang terjadi antara perangkat akuisisi data dan komputer terjadi dalam mode USART melalui satu buah port USB.

Aplikasi kendali dan akuisisi data yang berjalan di komputer berfungsi untuk menerima data hasil pengukuran dan mengirimkan sinyal perintah menuju perangkat akuisisi data. Selain itu, aplikasi juga menghasilkan data rekaman dalam format CSV (*Comma-Separated Value*).



Gambar 1. Sistem Keseluruhan

B. Perancangan Perangkat Keras

Perangkat keras yang digunakan dalam piranti akuisisi data terdiri dari sensor, mikropengendali dan LED. Sensor DHT22 digunakan untuk mengukur suhu ruangan dan kelembaban udara. Sensor ini dipilih karena keluaran dari proses pengukuran sudah dalam bentuk sinyal digital sehingga tidak lagi diperlukan proses ADC dan rangkaian pengondisi sinyal. Sensor dihubungkan ke mikropengendali pada salah satu pin *Digital Input* dan berkomunikasi secara dua arah melalui mode serial satu kabel (*one-wire*) [5]. Ada pun LDR digunakan untuk mendeteksi intensitas cahaya dengan keluaran berupa sinyal analog yang selanjutnya diproses oleh ADC menjadi nilai *integer* dengan rentang dari 0 – 1023 (mode 10 bit). LDR dihubungkan pada pin *Analog Output* yang pertama pada Arduino.

Arduino Uno R3 menggunakan mikropengendali ATmega328P sebagai pemroses utama dan ATmega16U2 untuk pemrograman dan komunikasi serial. ATmega328P memiliki memori program (*Flash*) sebesar 32 KB, EEPROM 1KB dan SRAM 2K [6] yang sudah mencukupi untuk penelitian ini. Mikropengendali tersebut diprogram menggunakan bahasa pemrograman Arduino (<http://arduino.cc>)

melalui aplikasi Arduino IDE versi 1.6.5 yang tersedia untuk lingkungan Windows, Linux dan Mac. Selain untuk memrogram, fitur lain yang digunakan adalah Serial Monitor. Fitur ini memungkinkan untuk memonitor aliran data melalui komunikasi serial secara dua arah [7].

Dua buah LED yang terhubung ke dua pin *Digital Input* dari Arduino berfungsi sebagai indikator. Jika Arduino menerima sinyal perintah komputer untuk melakukan pengukuran, maka LED berwarna hijau akan menyala dan LED merah akan mati. Kondisi nyala dari dua LED tersebut akan berbalik jika Arduino tidak menerima sinyal perintah.

C. Perancangan Aplikasi Multiplatform

Aplikasi kendali dan akuisisi data yang dijalankan di PC dibangun menggunakan perangkat pemrograman Processing IDE versi 2.2.1. Processing (<http://processing.org>) adalah bahasa pemrograman berkonteks grafis yang merupakan derivat dari Java. Bahasa ini dikembangkan oleh Casey Reas dan Ben Frye dari MIT Media Lab pada tahun 2001 dengan lisensi *open source* [8]. Aplikasi yang dikembangkan melalui bahasa ini dapat dijalankan di berbagai sistem operasi yang mendukung Java, sehingga mendukung fitur multiplatform.

Perancangan aplikasi meliputi dua tahap yaitu pembuatan antarmuka pengguna (*user interface*) dan program utama. Bentuk rancangan antarmuka aplikasi adalah seperti dalam Gambar 2. Untuk keperluan ini, dibentuklah beberapa rutin terpisah yang menghasilkan objek-objek grafis sederhana, seperti kotak (*box*), garis batas (*border*), tombol perintah (*command button*), label teks (*text label*) dan semacamnya. Beberapa objek ini selanjutnya digabungkan untuk menghasilkan tampilan seperti dalam gambar tersebut.



Gambar 2. Rancangan Antarmuka Aplikasi

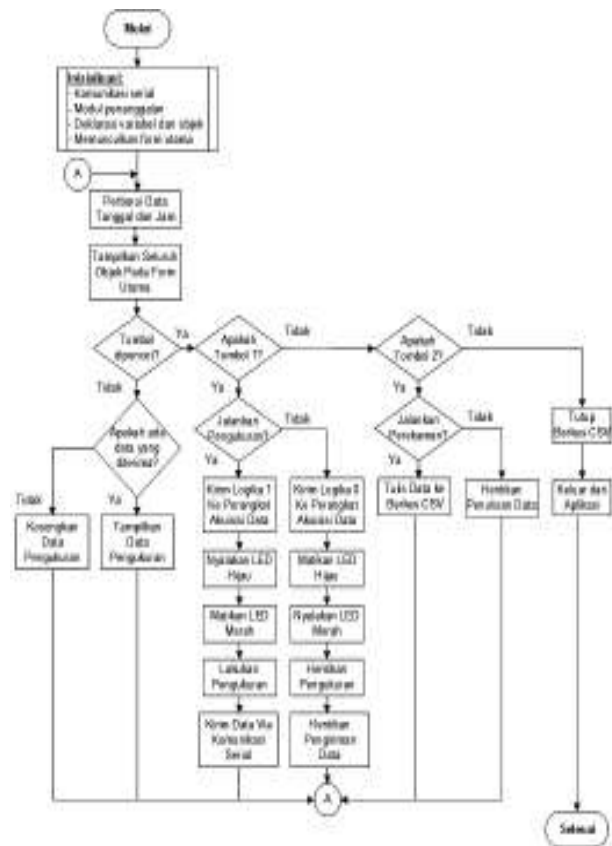
Program utama menggunakan dua buah modul eksternal, masing-masing untuk komunikasi serial dan fungsi penanggalan. Modul komunikasi serial dibutuhkan untuk memilih port, menciptakan objek serial dan membaca aliran data [9] yang dikirimkan dari perangkat akuisisi data. Standarnya, Processing tidak memiliki suatu fungsi untuk mengambil data penanggalan dari sistem komputer dan oleh karenanya memerlukan pemanggilan berkas pustaka Java.

Secara keseluruhan, program utama bertugas untuk menampilkan form utama beserta seluruh objek di dalamnya, menerima data, mengirimkan perintah, menampilkan data hasil pengukuran, merekamnya dan menutup aplikasi. Pengiriman perintah untuk menjalankan pengukuran kepada perangkat akuisisi data akan mengaktifkan dua proses lainnya yaitu pengiriman data hasil pengukuran dan penayangan nilai dari data tersebut. Di sisi Arduino, ada atau tidaknya pengiriman sinyal perintah ditandai dengan kondisi penyalan LED hijau dan merah. Sedangkan jika pengguna atau operator memilih untuk keluar dari aplikasi, maka terdapat juga proses untuk menutup berkas CSV. Bilamana proses ini tidak dijalankan maka berkas akan menjadi tidak dapat dibaca (*unreadable*). Selengkapnya alur yang terjadi di dalam program utama diperlihatkan dalam Gambar 3.

Dalam kedua tahap perancangan aplikasi tersebut, model pengembangan perangkat lunak yang digunakan adalah *Simple Process*. Model ini dipakai karena pengembang hanya terdiri dari satu orang dan aktivitas yang dilakukan berpusat pada penulisan kode program (*coding*). Aktivitas diawali dari pernyataan permasalahan, dilanjutkan dengan penulisan kode program (*code*), kompilasi (*compile*) dan pengujian (*unit test*) [10]. Jika pada kedua tahap terakhir terdapat kesalahan, maka kemudian berlanjut dengan penelusuran kesalahan (*debug*) dan berulang kembali ke penulisan kode. Demikian seterusnya hingga aplikasi siap diluncurkan (*release*).

D. Pengujian

Pengujian yang dilakukan bertujuan untuk memverifikasi perangkat akuisisi data dan aplikasi pada PC. Jika perangkat akuisisi data telah berfungsi dengan baik, maka hal ini dapat diamati dari data yang dikirimkan menuju PC. Sedangkan pengujian untuk perangkat lunak terdiri dari tiga tahap yaitu uji modul, uji tampilan dan uji fungsionalitas. Aplikasi dinyatakan layak untuk dioperasikan jika lolos dari ketiga pengujian tersebut.



Gambar 3. Diagram Alir Aplikasi

Seluruh proses pengujian dilakukan di dua lingkungan sistem operasi yang berbeda, yaitu Windows dan Linux. Untuk Windows, pengujian dilaksanakan di Windows 10 Professional 64 bit sedangkan untuk Linux di dalam Ubuntu 14.40 (*Trusty Tahr*) 64 bit. Komputer untuk pengujian memiliki spesifikasi prosesor Intel Core i5-3337U 1,8 GHz dan RAM 4 GB.

III. HASIL DAN PEMBAHASAN

Berdasarkan uraian sebelumnya tentang proses pengujian yang dilaksanakan, berikut ini adalah hasil-hasil yang diperoleh:

A. Uji Komunikasi Serial

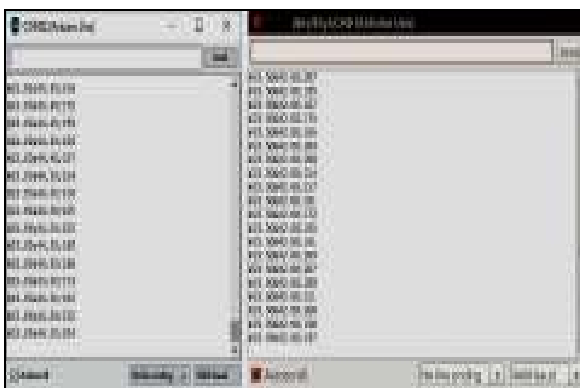
Pengujian ini bertujuan untuk mengetahui apakah perangkat akuisisi data telah dapat mengirimkan data ke komputer dengan baik. Di dalam pengujian ini perangkat tersebut dihubungkan dengan komputer ke salah satu port USB yang tersedia. Setelah terhubung, perangkat lunak Arduino IDE versi 1.6.5 dijalankan. Pengaturan dilakukan untuk memilih port USB yang digunakan. Modul Serial Monitor selanjutnya dibuka dan dilakukan pemilihan

bit rate. Konfigurasi untuk komunikasi serial ini adalah *bit rate* 9600 bps, format 8 bit data dengan satu *stop bit* tanpa paritas. Data yang dikirimkan memiliki format:

#aa.aa&bb.bb;ccc

dengan a menunjukkan nilai suhu ruangan (dalam Celcius), b untuk nilai kelembaban udara (dalam %) dan c adalah nilai intensitas cahaya (berupa 3 digit *integer* tanpa satuan).

Hasil yang diperoleh adalah sebagaimana dalam Gambar 4. Dari tampilan tersebut dapat diketahui bahwa perangkat akuisisi data telah mengirimkan data sesuai format yang ditentukan di lingkungan Windows (kiri) maupun Linux (kanan). Di dalam Windows, port serial yang digunakan adalah COM32 sementara di Linux adalah /dev/ttyACM0.



Gambar 4. Hasil Uji Komunikasi Serial

B. Uji Modul

Pengujian ini memiliki tujuan untuk mengetahui apakah masing-masing rutin di dalam kode program telah mampu menghasilkan keluaran yang diharapkan. Ini mengingat bahwa dari rutin-rutin tersebutlah tampilan aplikasi dan fungsionalitas program utama dibangun. Berdasarkan hasil uji yang dilakukan di lingkungan Windows maupun Linux seperti dalam Tabel 1 berikut dapat diketahui bahwa seluruh modul telah berfungsi dengan baik.

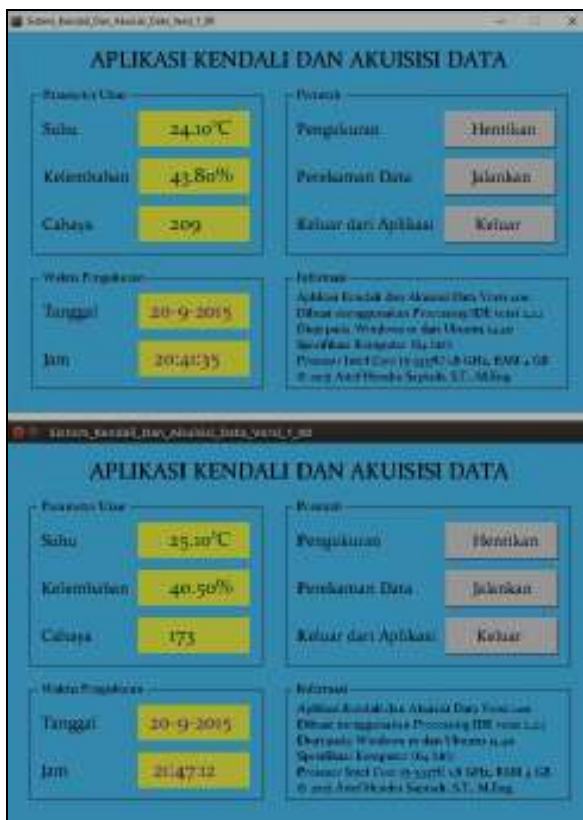
TABEL I
UJI MODUL

No.	Modul	Fungsi	Indikator	Hasil Uji	
				Windows	Linux
1.	time()	Mengambil data jam saat ini dan mengatur format tampilan	Nilai jam tertampil sesuai format dan terbaru setiap saat	OK	OK
2.	border()	Menampilkan garis batas sesuai argumen	Garis batas muncul sesuai argumen	OK	OK
3.	rectangle()	Menciptakan objek segi empat sesuai argumen	Objek segi empat muncul sesuai argumen	OK	OK
4.	label()	Membuat label teks sesuai argumen	Label teks muncul sesuai argumen	OK	OK
5.	cmdButton()	Membuat tombol perintah sesuai argumen	Tombol perintah muncul sesuai argumen	OK	OK
6.	cmdLabel()	Membuat label teks yang berganti-ganti (<i>toggle</i>) sesuai kondisi logika	Label teks muncul dan dapat berubah sesuai kondisi logika	OK	OK
7.	readPort()	Membaca aliran data via komunikasi serial dan mengambil nilai darinya	Data dari komunikasi serial dapat diambil dan dimasukkan ke variabel	OK	OK
8.	checkBox()	Mengecek kondisi logika dari tombol yang ditekan dan menjalankan aksi yang sesuai	Tombol memunculkan aksi yang sesuai ketika ditekan	OK	OK
9.	writePort()	Mengirimkan perintah via komunikasi serial ke perangkat akuisisi data	Perangkat akuisisi data menjalankan perintah sesuai data yang dikirim	OK	OK
10.	displayValue()	Menampilkan nilai sesuai data yang diterima atau mengosongkannya saat tidak ada data	Nilai tertampil atau kosong sesuai keberadaan data	OK	OK

C. Uji Tampilan

Pengujian ini bertujuan mengecek aspek tampilan dari aplikasi, apakah seluruh objek grafis yang membentuk antarmuka pengguna telah dapat ditampilkan sesuai rancangan awal. Tampilan antarmuka secara keseluruhan terbagi ke dalam lima bagian, yaitu form utama, parameter ukur, waktu pengukuran, perintah dan informasi.

Ketika dijalankan, aplikasi memiliki tampilan seperti dalam Gambar 5 di sistem operasi Windows (atas) dan Linux (bawah). Selanjutnya dari aplikasi tersebut dilakukan pengujian di lima bagian sebagaimana dimaksud dengan hasil-hasil seperti tertera dalam Tabel II. Berdasarkan hasil-hasil tersebut dan membandingkannya terhadap rancangan awal seperti pada Gambar 2, maka dapat diketahui bahwa dari sisi tampilan, aplikasi telah lolos pengujian.



Gambar 5. Tampilan Antarmuka Aplikasi

D. Uji Fungsionalitas

Tujuan dari pengujian ini adalah untuk mengetahui apakah fungsi-fungsi yang terdapat di dalam aplikasi telah dapat dijalankan atau tidak. Fungsionalitas yang diuji meliputi kemampuan untuk menerima data dan mengirimkan perintah via

komunikasi serial, merekam data dan menutup aplikasi. Secara khusus, pengujian ini dilakukan pada tiga buah tombol perintah yang terdapat di dalam aplikasi untuk selanjutnya diamati hasilnya. Ada pun tombol untuk melakukan pengukuran dan perekaman data memiliki dua kondisi (*toggle*) yaitu ketika aktif dan non-aktif, sementara tombol terakhir untuk keluar dari aplikasi hanya memiliki satu kondisi yaitu ketika diklik oleh pengguna atau operator.

TABEL II
UJI TAMPILAN

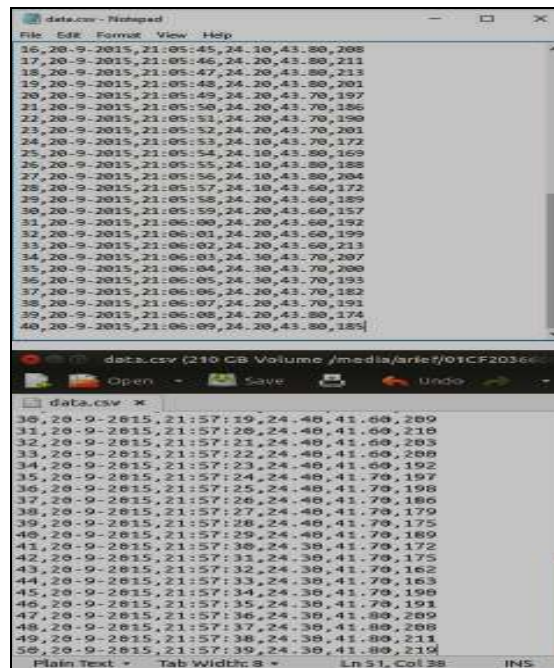
No.	Bagian Tampilan	Indikator	Hasil Uji	
			Windows	Linux
1.	Judul aplikasi dan form utama	Form utama terbentuk dengan ukuran dan pilihan warna sesuai rancangan. Judul aplikasi muncul di form utama.	OK	OK
2.	Parameter Ukur	Format tampilan sesuai rancangan. Nilai suhu, kelembaban dan intensitas cahaya dari perangkat akuisisi data tertampil.	OK	OK
3.	Waktu Pengukuran	Format tampilan sesuai rancangan. Nilai tanggal dan waktu tampil dan berubah mengikuti kondisi saat ini.	OK	OK
4.	Perintah	Format tampilan sesuai rancangan. Aplikasi mampu mendeteksi kondisi penekanan dan posisi cursor mouse. Tombol Jalankan berubah menjadi Hentikan demikian pula sebaliknya saat diklik.	OK	OK
5.	Informasi	Format tampilan sesuai rancangan.	OK	OK

Hasil-hasil pengujian yang diperoleh adalah seperti dalam Tabel III. Sedangkan bentuk rekaman data dalam format CSV diperlihatkan dalam Gambar 6, di lingkungan Windows (atas) dan Linux (bawah). Berdasarkan tabel tersebut dan hasil rekaman data

maka dapat diketahui dari sisi fungsionalitas, aplikasi telah dapat menjalankan seluruh fungsinya dengan baik, di lingkungan Windows maupun Linux.

TABEL III
UJI FUNGSIONALITAS

No.	Fungsio- nalitas	Kondisi	Indikator	Hasil Uji	
				Windows	Linux
1.	Menerima data dari komunikasi serial	Pembacaan data berjalan	Nilai tampil	OK	OK
		Pembacaan data berhenti	Nilai tidak tampil	OK	OK
2.	Mengirimkan perintah via komunikasi serial	Pengiriman perintah berjalan	LED hijau pada perangkat akuisisi data menyala. LED merah mati.	OK	OK
		Pengiriman perintah berhenti	LED merah pada perangkat akuisisi data menyala. LED hijau mati.	OK	OK
3.	Merekam data ke berkas ber-format CSV	Perekaman data selesai dilakukan	Berkas CSV muncul dan terisi data	OK	OK
4.	Keluar dari aplikasi	Diklik	Aplikasi ditutup	OK	OK



Gambar 6. Hasil Rekaman Data

IV. PENUTUP

A. Kesimpulan

Berdasarkan perancangan yang sudah dilakukan dan hasil-hasil pengujian yang diperoleh, maka dapat ditarik beberapa kesimpulan sebagai berikut:

1. Perangkat akuisisi data telah beroperasi dengan baik dan hal ini dibuktikan oleh tampilan aliran data yang diperoleh via komunikasi serial.
2. Seluruh modul yang menjadi dasar dalam membuat tampilan aplikasi dan program utama telah berfungsi baik.
3. Dari sisi tampilan, semua objek di dalam antarmuka aplikasi telah dapat ditampilkan sesuai rancangan awal.
4. Dari sisi fungsionalitas, seluruh fitur yang diharapkan ada di aplikasi telah dapat dijalankan dengan baik.

B. Saran

Saran yang dapat dipertimbangkan untuk pengembangan aplikasi semacam ini adalah penggunaan paradigma pemrograman berorientasi objek (*Object-Oriented Programming*). Dalam penelitian ini seluruh tampilan maupun fungsionalitas dihasilkan dari berbagai fungsi yang tidak terstruktur dengan baik. Fungsi-fungsi untuk menghasilkan bentuk grafis seperti label teks, kotak

dan semacamnya perlu disusun ke dalam satu berkas pustaka terpisah. Berkas ini selanjutnya dipanggil dari program utama.

Pada pengembangan selanjutnya, komunikasi data serial yang terjadi antara perangkat akuisisi data dengan komputer dapat menggunakan proses *handshake*. Dengan demikian, perangkat akuisisi data tidak mengirimkan data secara terus-menerus melainkan menunggu sinyal permintaan (*request*) dari komputer terlebih dahulu.

DAFTAR PUSTAKA

- [1] Subramaniam, E. T., Kumar, B. P. A & Bhowmik, R. K., (2010), Data Acquisition Systems – Current and future trends, *Proceedings of The DAE Symposium on Nuclear Physics*, Pilani, India.
- [2] Lapolli, A. L., Zahn, G. S., (2013), Development of a multiplatform and multiparameter data acquisition interface, *2013 International Nuclear Atlantic Conference (INAC 2013)*, Recife, Brazil, 24 – 29 November 2013.
- [3] Homfray, D. A., Deakin, K., Khilar, S., King, R., Payne, D., Simmonds, M. R., Tame, C. & Whitehead, B., (2013), Development of multiplatform control and instrumentation communications to increase operational reliability – Application to MAST, *Fusion Engineering and Design*, Vol. 88, Issues 6 – 8, hal. 1219 – 1223.
- [4] Perrey, H., (2014), EUDAQ and EUTELESCOPE – Software Frameworks for Testbeam Data Acquisition and Analysis, *Technology and Instrumentation in Particle Physics 2014*, Amsterdam, Belanda, 2 – 6 Juni 2014.
- [5] Aosong, (2013), *Temperature and humidity module. AM2302 product manual*, Aosong Electronics, Guangzhou, China.
- [6] ATMEL, (2014), *8-bit AVR microcontroller with 4/8/16/32Kbytes In-System Programmable Flash. ATmega48A ATmega48PA ATmega88A ATmega88PA ATmega168A ATmega168PA ATmega328 ATmega328P*, ATMEL Corporation, San Jose, California, USA.
- [7] Schmidt, M., (2011), *Arduino A Quick Start Guide*, Pragmatic Programmers, USA.
- [8] Reas, C., Frye, B., (2015), *Make: Getting Started with Processing. A Hands-On Introduction to Making Interactive Graphics*. Second Edition, Maker Media, Inc. San Fransisco, USA.
- [9] Runberg, D., (2015), *The Sparkfun Guide to Processing. Create Interactive Art with Code*. No Starch Press Inc., San Fransisco, USA.
- [10] Tsui, F., Karam, O. & Bernal, B., (2013), *Essentials of Software Engineering*, Third Edition, Jones & Bartlett Learning, Burlington.