

Pembuatan Sistem Akuisisi Data Sederhana Menggunakan Arduino-Python (Studi Kasus: Suhu Fluida pada Tangki Pemanas)

Dita Nurhalimah, Muhammad Iwan Fermi*, Zulfansyah

Laboratorium Pengendalian dan Perancangan Proses
Jurusan Teknik Kimia, Fakultas Teknik, Universitas Riau

E-mail: miwanf@unri.ac.id

ABSTRACT

Data acquisition (DAQ) is a part of control system and almost applied in all aspects, in industrial automation processes. The main objective of this research is to construct the DAQ using Arduino Python-based in developing a graphical user interface (GUI). The system consists of two components, i.e. hardware and software, which are designed on client-runtime module. PyQt5 was chosen as the basis for developing the GUI. The DAQ system testing is carried out on a stirred heating tank with the observations of fluid temperatures and actuation element is electric heaters. The GUI allows users to be able to visualize and store data in real time mode. The performance of the system is verified based on manual observation. It was found that the system could be carried out for interactive manual control. Even so, the created system can be used for data collection process in the process control laboratory in Chemical Engineering Department of Riau University. In the future, it is expected that the performance of the system can be improved.

Keywords: data acquisition, control, Arduino, Python, GUI

1 Pendahuluan

Limpahan data informasi yang dikumpulkan dari instrumen sensor pada operasi pabrik perlu dikelola melalui sistem akuisisi data dan kendali atau yang dikenal dengan sebutan DAQ (Haizad dkk., 2016). Akuisisi data adalah proses dimana fenomena fisik dari dunia nyata ditransformasikan menjadi sinyal listrik yang diukur dan diubah menjadi format digital untuk diproses, dianalisis, dan disimpan oleh komputer. Kendali adalah proses di mana sinyal kendali digital dari perangkat keras sistem diatur ke format sinyal yang digunakan oleh perangkat kendali seperti aktuator dan relay. Perangkat akuisisi data ini akan memantau variabel operasi yang kemudian dapat digunakan untuk meningkatkan efisiensi, serta

memastikan keandalan bahwa proses beroperasi dengan aman. Sistem akuisisi data *real time* bahkan menampilkan pengukuran tanpa penundaan. Dilengkapi dengan kendali, sistem menghasilkan umpan balik secara instan.

Dengan kemajuan teknologi mikrokomputer, sistem DAQ yang ada pada industri dapat diterapkan dengan biaya yang lebih rendah. Mikrokomputer dapat digunakan untuk proses pemantauan dan pengendalian proses; akuisisi, pengolahan dan penyimpanan data dari proses; serta meningkatkan keamanan dalam proses (Papić dkk., 2018). Fungsionalitasnya serupa dengan sistem DAQ komersil yang mungkin tidak tepat digunakan untuk skala kecil seperti hanya untuk kepentingan pembelajaran. Hal ini tentu sangat berguna

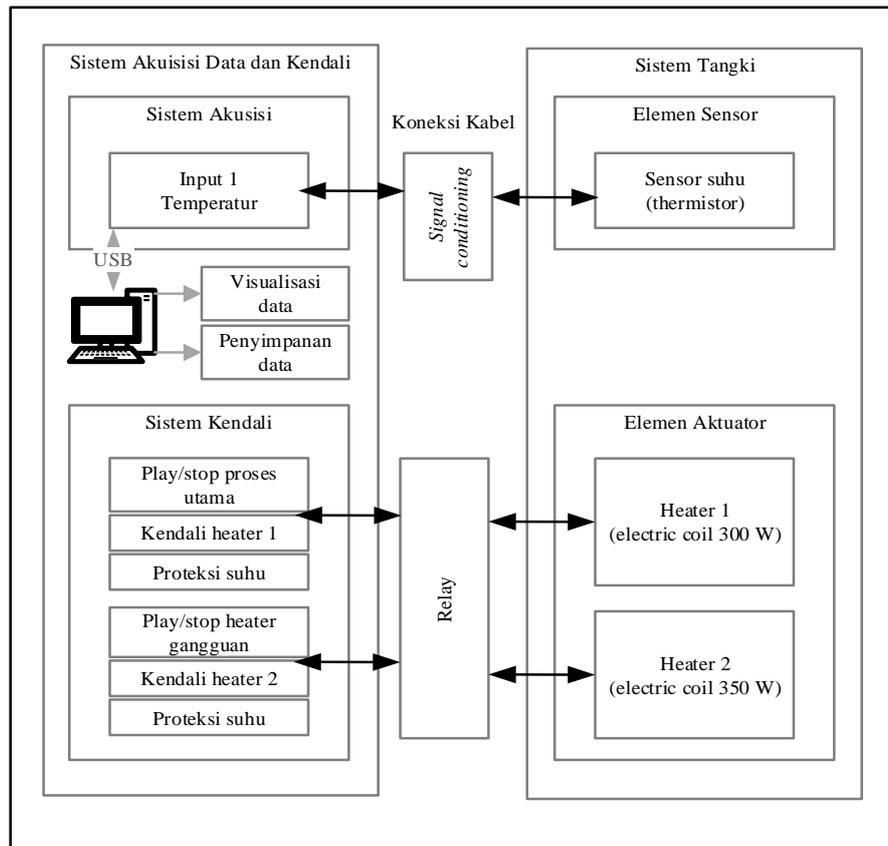
bagi institusi yang menyediakan praktik subjek sistem kendali industri proses, dimana mahasiswa dapat membangun sistem DAQ menggunakan mikrokomputer tersebut dan mempelajari bagaimana prinsip otomasi bekerja pada industri.

Teknologi mikrokomputer yang dimaksud ialah papan dengan sistem tertanam atau *embedded system*, seperti mikrokontroler Arduino (Haizad dkk., 2016) dan

penggunaan *signal conditioning* dapat diminimalisir, selain itu proses pembuatan prototip yang cepat menjadikan sistem DAQ berbasis mikrokomputer ini sesuai diterapkan pada praktik kendali proses dalam lingkungan institusi.

2 Metode Penelitian

2.1 Deskripsi Umum Sistem yang Dibuat



Gambar 2.1 Skema modular sistem DAQ yang diusulkan

mikroprosesor Raspberry (Li dkk., 2015). Sistem ini *low-cost* dikarenakan menggunakan komunitas yang *open source* dalam pengembangan piranti keras maupun lunak, dan memerlukan konsumsi energi yang rendah (Papić dkk., 2018). Mikrokomputer ini dapat digunakan sebagai elemen utama pada sistem DAQ yang menghubungkan bagian fisik seperti sensor dan aktuator terhubung dengan komputer yang menyediakan perangkat lunak sistem DAQ. Sistem tertanam biasanya terintegrasi dengan *Analog-Digital-Converter* atau ADC (Wilmshurst, 2010), sehingga

Sistem pemanas yang dipilih ialah tangki pemanas berpengaduk yang terhubung dengan sistem DAQ dan kendali melalui transmisi kabel seperti yang dilihat pada struktur modular Gambar 2.1. Sistem akuisisi menangani pengukuran nilai fisik yang diperoleh sensor suhu pada Arduino menjadi informasi pada jalur selanjutnya, yakni untuk visualisasi dan penyimpanan. Sistem kendali memuat perintah untuk mengendalikan keadaan *heater* utama dan gangguan. Oleh karena *heater* bertegangan tinggi tidak dapat dikendalikan langsung oleh Arduino yang

beroperasi pada 5V, maka *relay* digunakan sebagai perantara antar kedua komponen. Sistem DAQ dan kendali dapat diakses secara interaktif melalui aplikasi program GUI yang dibuat. Apabila sistem DAQ dan kendali diaktifkan, proses pada sistem tangki akan merespon dan kemudian akan direkam oleh sensor. Sensor mengukur suhu dan terjadi putaran arus data hingga pada GUI diterima perintah untuk menghentikan sistem.

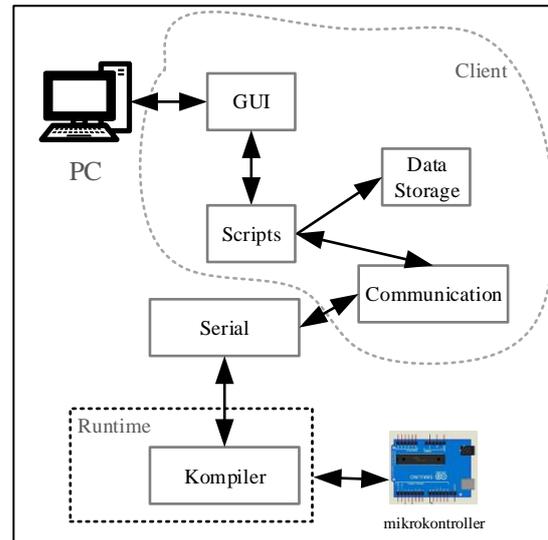
2.2 Komponen Perangkat Keras

Elemen perangkat keras penyusun sistem DAQ dan kendali terdiri dari sensor tipe thermistor, Arduino UNO R3, dan relay mekanik.

2.3 Rancangan Perangkat Lunak

Program yang akan dirancang pada bagian ini terdiri dari kode GUI dan kode kompiler ke Arduino, dengan arsitektur modul *client-runtime* (Gambar 2.2). Kode kompiler dibuat pada Arduino IDE sedangkan GUI dikembangkan dengan Python melalui aplikasi pyCharm IDE. *Plug-in library* yang dibutuhkan pada program GUI ialah pyserial, PyQt5, dan pyqtgraph. Sedangkan untuk program kompiler ialah OneWire dan DallasTemperature.

Pada bagian *client*, program dipaketkan menjadi empat paket utama, yaitu paket GUI, *scripts*, *communication*, dan *data storage*. Paket tersebut dikelompokkan atas dasar: Paket GUI yang berhubungan dengan tampilan layar beserta widget; Paket *scripts* yang berhubungan dengan fungsi non-grafis; Paket *communication* berhubungan dengan penanganan pertukaran data sesuai protokol komunikasi; serta paket *data storage* mengelompokkan penyimpanan data sesuai dengan jenis media penyimpanan.



Gambar 2.2 Modular paket kode pada program perangkat lunak yang akan dibuat

Sementara, prosedur atau fungsi yang akan tersedia pada program kompiler ialah pembacaan data serial yang dikirim dari aplikasi GUI, pembaharuan nilai temperatur, serta pengiriman data yang dibutuhkan oleh GUI kembali ke sambungan serial. Prosedural keamanan pada sistem DAQ juga tersedia untuk menghindari sistem dari *overheat*. Batas suhu maksimum dari studi kasus ialah 60 °C, sehingga apabila suhu sistem telah melebihi nilai ini, maka *heater* (baik utama maupun gangguan) secara otomatis akan dimatikan.

2.4 Uji Coba

Sistem DAQ dan kendali diverifikasi dengan menguji fungsionalitas aplikasi GUI. Pengujian tersebut terdiri dari penerapan mode kendali secara manual dan mode kendali PID. Faktor penentu bahwa sistem DAQ dan kendali telah berfungsi ialah hasil visualisasi dan penyimpanan data yang dilakukan selama sistem DAQ dan kendali diaktifkan.

3 Hasil dan Pembahasan

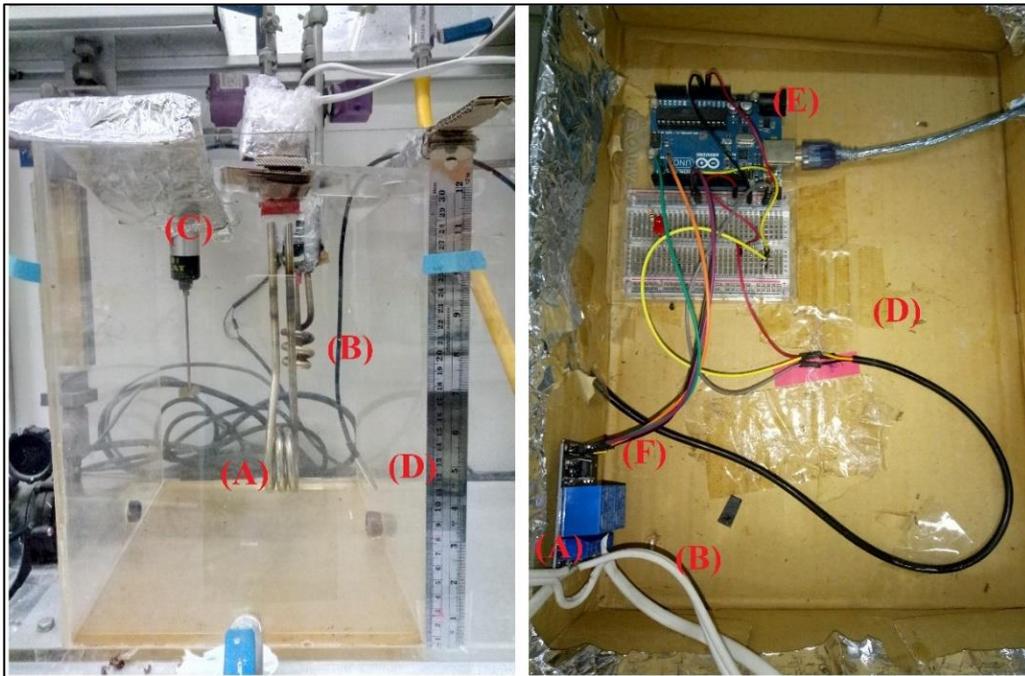
3.1 Sistem yang dihasilkan

Gambar 3.1 menunjukkan tampilan fisik sistem tangki yang digunakan beserta rangkaian elemen perangkat keras pada papan Arduino. Lebih jelasnya, skematik rangkaian tersebut ada pada Gambar 3.2. Pada bagian sensor (Dallas DS18B20), GND dihubungkan dengan ground, VCC dengan 5V, dan DQ dengan pin digital D8 pada Arduino. Pada sambungan ini juga dilengkapi dengan resistor pull-up sebesar

mengubah perubahan resistansi menjadi perubahan voltase (Bolton, 2015). Relay terhubung pada output digital Arduino, yaitu D2 dan D4, yang digunakan untuk mengendalikan pada *heater* utama dan gangguan. Sementara, aliran listrik pada heater dikontakkan secara *normally open* (NO) pada relay, yang mana kontak arus terputus saat tidak ada sinyal dari Arduino.

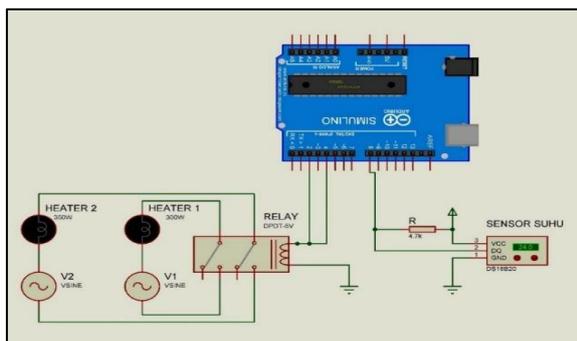
3.1.1 GUI

Gambar 3.3 merupakan tampilan



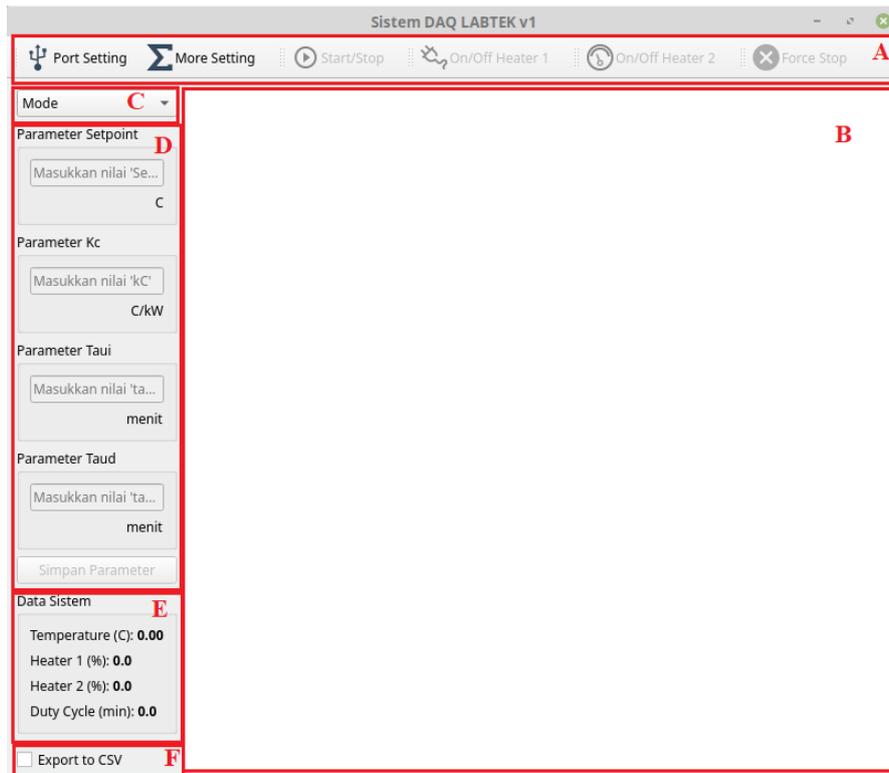
Gambar 3.1 (kiri) Tampak depan sistem tangki yang dibangun; (kanan) Tampilan sambungan elemen perangkat keras DAQ. (A) *Heater* utama; (B) *Heater* gangguan; (C) Pengaduk; (D) Sensor suhu; (E) Arduino; (F) *Relay*

4,7k Ω . Resistor ini berfungsi sebagai signal conditioning yaitu voltage divider untuk



Gambar 3.2 Skematik Rangkaian Elemen Sistem DAQ

layar GUI saat dijalankan. *Toolbar* berisi tombol untuk pengaturan serta perintah-perintah yang utama. Tombol *start/stop* dapat diakses apabila mode kendali dipilih. Selain itu, area plot (B) juga akan kosong sebelum pengaturan ini dilakukan. Bagian D tidak dapat diakses apabila mode kendali yang dipilih bukan kendali PID. Mode penyimpanan akan *disable* saat sistem akuisisi data dimulai. Pada mode kendali PID, output ditranslasikan menjadi lama waktu yang diperlukan untuk menghidupkan *heater* dari total siklus kerja yang diterapkan, dikarenakan relay menggunakan logika hidup/mati. Oleh



Gambar 3.3 Tampilan GUI saat dibuka pertama kali. (A) *toolbar*; (B) daerah *plotting*; (C) pengaturan mode kendali; (D) pengaturan parameter PID; (E) tampilan data dari Arduino; (F) pengaturan penyimpanan data

sebabnya, pengaturan *duty cycle* disediakan pada GUI.

3.1.2 Prinsip Kerja Program

GUI dapat dijalankan oleh karena interaksi dari modul *client* dan *runtime* (Gambar 2.2). Pada bagian *runtime* merupakan eksekusi program kompilasi pada papan Arduino. Prosedural pertama ialah pembacaan suhu. Kemudian prosedur translasi paket data yang dikirimkan dari python menjadi informasi untuk menentukan tindakan tertentu. Paket ini memuat informasi berupa nilai output untuk heater utama, nilai periode *duty cycle*, mode kendali yang dipilih, dan ada atau tidak nya gangguan yang dikemas dalam data array tipe byte. Selanjutnya, pengecekan status *overheating* sistem. Prosedur ini harus dilakukan setelah prosedur translasi data. Baru setelahnya, sinyal diberikan pada terminal output. Terakhir, sebelum kembali pada prosedural pertama ialah pengiriman paket data untuk python yang jumlahnya tergantung mode

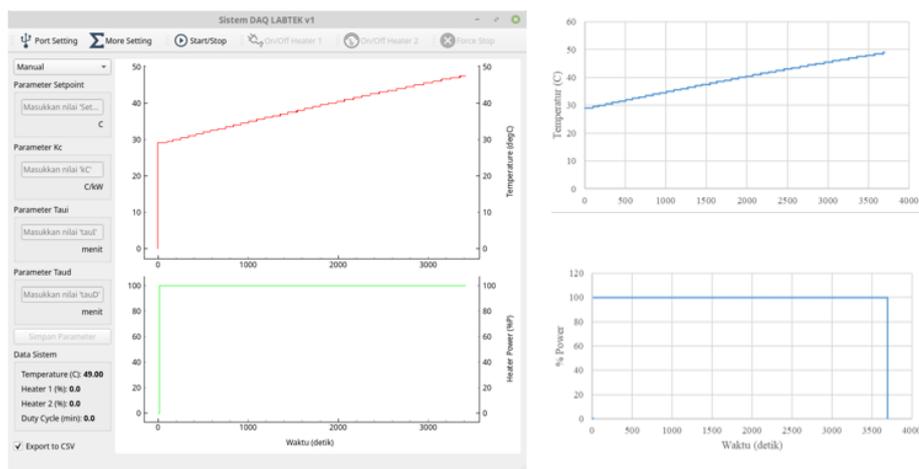
kendali yang dipilih. Pada mode kendali manual, paket data terdiri dari nilai temperatur, output heater pertama, dan output heater kedua, sedangkan pada mode PID, paket data ditambah dengan data waktu siklus terkini heater. Loop selanjutnya dilakukan setelah selang waktu 1 detik.

Verifikasi dilakukan dengan mengamati perubahan suhu sistem selama heater utama dihidupkan. Gambar 3.4 menunjukkan suhu sistem naik dari 29 °C hingga 49 °C. Hasil visualisasi plot data dari GUI sesuai dengan data yang diperoleh dari hasil penyimpanan, hanya saja pada saat sistem dimulai, inialisasi data pada plot GUI bernilai nol karena belum ada yang diperoleh. Setelah itu dilakukan pengamatan dengan memberikan perintah hidup-mati pada heater utama secara bergantian selama sekitar ± 90 detik dalam waktu 400 detik. Diamati bahwa waktu data yang di input baik untuk proses penyimpanan dan display sesuai dengan

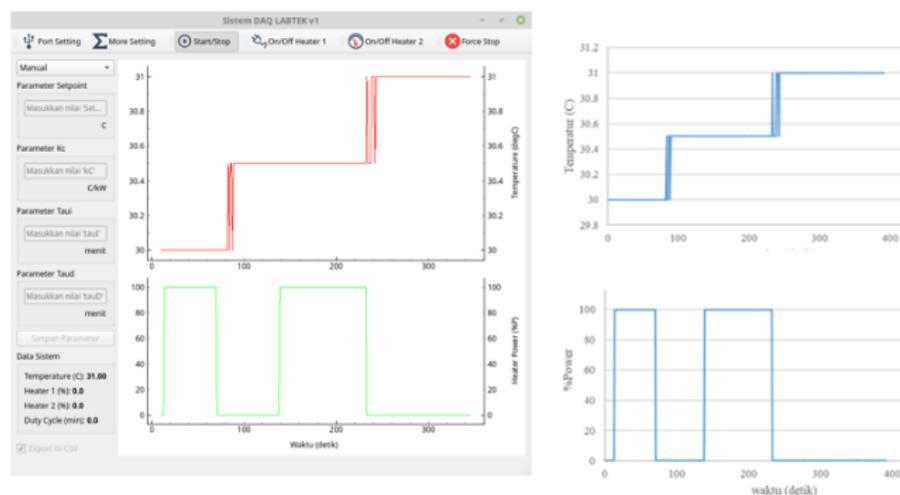
waktu yang terbaca oleh serial. Namun, hal berbeda ditemui pada saat proses graph plotting. Terdapat penundaan waktu plotting data dari waktu yang sebenarnya data serial dibaca. Penundaan waktu tersebut fluktuatif sepanjang perubahan perintah hidup-mati sehingga untuk proses plotting terlihat kurang responsif. Gambar 3.5 menunjukkan visualisasi data tersebut pada GUI (kiri) dan data yang diplot dari file yang disimpan (kanan). Dapat dilihat penundaan plot data pada GUI tidak terlalu nampak.

heater gangguan. Pada awalnya perilaku penundaan waktu *plotting* hampir serupa dengan kasus sebelumnya. Namun setelah 500 detik, proses plot semakin lama memperbaharui nilai yang diperoleh dari proses serial. Pada Gambar 3.6, perilaku ini tidak terlihat, namun bisa dibandingkan dengan data yang diplot dari file penyimpanan. Selisih waktu menjadi kurang lebih 200 detik.

Dari beberapa pengujian, terkadang ditemui data yang dibaca mengalami *noise*, dan biasanya berlangsung pada awal mula



Gambar 3.4 (kiri) Uji akuisisi data selama heater utama dihidupkan; (kanan) Data yang diplot dari hasil penyimpanan



Gambar 3.5 (kiri) Kendali manual dengan perubahan keadaan Q1; (kanan) Data yang diplot dari hasil penyimpanan

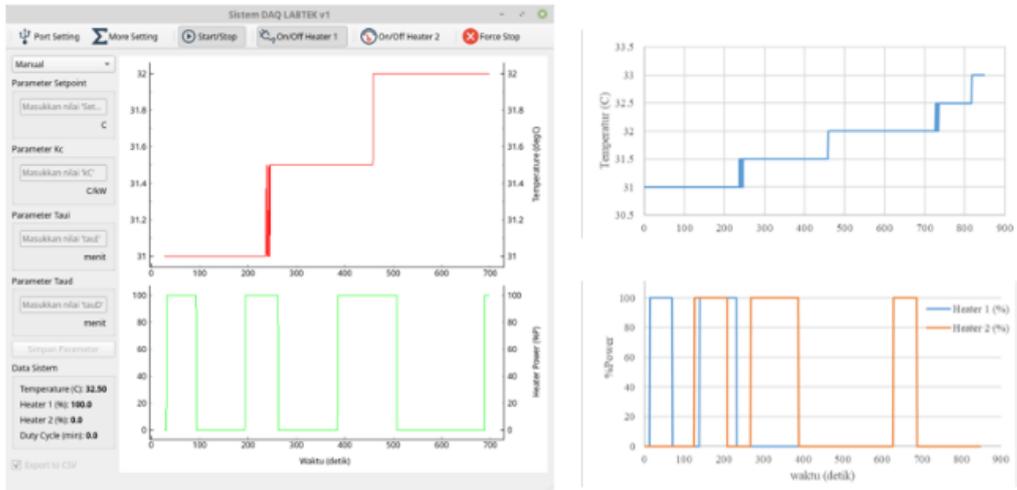
Kemudian, dilakukan pengamatan lain dengan rentang waktu lebih lama, yaitu hampir 900 detik dan perintah hidup-mati juga diberlakukan pada *heater* kedua atau

sistem DAQ dan kendali dimulai. Seperti yang terlihat pada pada Gambar 3.7. Pembacaan suhu tidak konsisten untuk beberapa detik. Isu ini diduga terkait

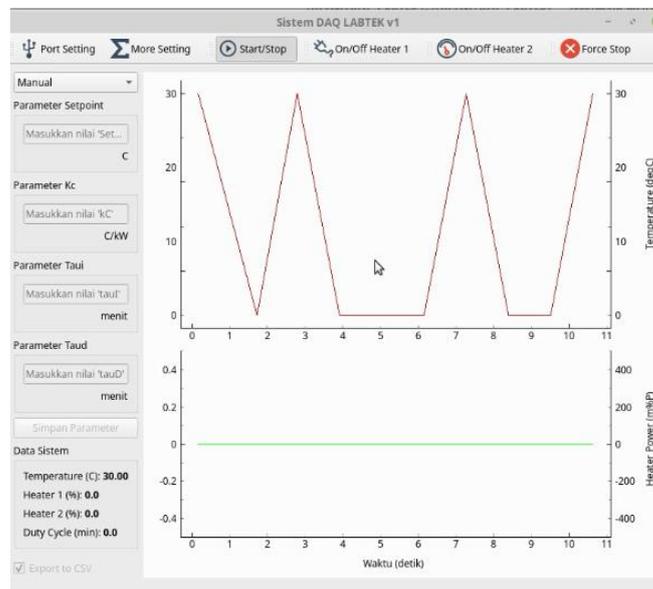
perbedaan waktu pemrosesan pada program kompilasi dan pada aplikasi GUI, yang bergantung kepada performa dari komputer saat digunakan.

4 Kesimpulan

Sistem DAQ menggunakan Arduino dengan aplikasi GUI berbasis Python telah dibuat. Secara garis besar sistem ini,



Gambar 3.6 (kiri) Kendali manual dengan perubahan Q1 & Q2 bergantian; (kanan) Data yang diplot dari hasil penyimpanan



Gambar 3.7 Plot data saat start up

3.2 Keterbatasan dari Sistem yang Dibangun

Sistem DAQ yang dibuat masih perlu untuk terus ditingkatkan. Sistem ini masih bergantung pada sistem operasi yang digunakan untuk menjalankan program GUI. GUI hanya bisa dioperasikan pada OS Linux. Ini dikarenakan struktur multiprocess yang ada pada GUI tidak compatible dengan sistem operasi Windows.

1. Dirancang dengan arsitektur modul *runtime-client*, dengan *runtime* merupakan paket program kompilasi, sedangkan *client* terdiri dari paket-paket yang berhubungan dengan aplikasi GUI
2. Dapat digunakan untuk pengambilan dan penyimpanan data, sedangkan kendali sudah dapat diterapkan untuk

basis operasi manual dengan memberi perintah melalui aplikasi.

5 Daftar Pustaka

- Haizad, M., Ibrahim, R., Adnan, A., Chung, T. D., & Hassan, S. M. (2016). *Development of low-cost real-time data acquisition system for process automation and control*. Paper dipresentasikan pada 2016 2nd IEEE International Symposium on Robotics and Manufacturing Automation (ROMA).
- Li, W. J., Tung, S. C., & Huang, S. M. (2015). *Web-based supervisory control system based on raspberry Pi*. Paper dipresentasikan pada Applied Mechanics and Materials.
- Papić, M., Bundalo, Z., Bundalo, D., Stojanović, R., Kovačević, Ž., Pašalić, D., & Cvijić, B. (2018). Microcomputer based embedded SCADA and RFID systems implemented on LINUX platform. *Microprocessors and Microsystems*, 63, 116-127.
- Wilmshurst, T. (2010). CHAPTER 11 - Data acquisition and manipulation. In T. Wilmshurst (Ed.), *Designing Embedded Systems with PIC Microcontrollers (Second Edition)* (pp. 339-369). Boston: Newnes.