

Pengembangan RTU (Remote Terminal Unit) untuk Sistem Kontrol Jarak Jauh berbasis IP

Rika Sustika
P2 Informatika-LIPI
rika@informatika.lipi.go.id

Oka Mahendra
P2 Informatika-LIPI
oka@informatika.lipi.go.id

Abstrak

Remote Terminal Unit (RTU) adalah bagian dari sistem kontrol yang merupakan antar muka antara objek yang dikontrol dengan master station. Pada saat ini banyak tipe dan jenis RTU yang telah beredar. Pada umumnya perangkat keras dan perangkat lunak RTU telah menjadi satu kesatuan. Pada kegiatan kami, dikembangkan RTU yang berbasis pada embedded controller. Embedded controller memiliki kebebasan dalam penggunaan perangkat lunak sehingga pengembang dapat mengembangkan perangkat lunak di dalamnya sesuai dengan kebutuhan dan keinginan pengembang. Perangkat keras RTU yang kami kembangkan terdiri dari embedded controller, mikrokontroler, dan modul input output. RTU ini juga telah dilengkapi dengan GUI (Graphical User Interface) sehingga apabila ada masalah komunikasi antara RTU dengan master station, sistem masih dapat dimonitor melalui GUI yang telah ada pada RTU. Komunikasi yang dipergunakan untuk berhubungan dengan master station adalah komunikasi berbasis IP (Internet Protocol).

Kata kunci: embedded controller, mikrokontroler, RTU, IP

1. Pendahuluan

Pada umumnya, ada lima komponen utama yang membangun sistem kontrol jarak jauh baik itu berupa *Distributed Control System (DCS)* atau *Supervisory Control And Data Acquisition (SCADA)*, yaitu *master station*, *RTU (Remote Terminal Unit)*, *instruments* di lapangan, link komunikasi, dan perangkat lunak [1]. Tujuan dari sistem kontrol ini adalah mengumpulkan data dari *plant* yang lokasinya berada di tempat yang jauh dari *master station*, mengirimkan data tersebut ke *master station*, menampilkan mereka pada monitor atau *master computer* di ruang kontrol, menyimpan data ke *hard drive* dari *master computer*, dan melakukan kontrol dan monitoring terhadap *instruments* dari ruang kontrol[2].

Remote Terminal Unit (RTU) adalah salah satu bagian dari sistem kontrol jarak jauh yang ditempatkan dekat objek yang dikontrol, dan jauh dari *master station*. RTU

berfungsi untuk mengambil informasi dari sensor-sensor dan peralatan di lapangan, kemudian memformat data yang diperoleh agar bisa ditransmisikan ke *master station* melalui suatu jaringan komunikasi tertentu. RTU juga dapat menerima dan melaksanakan perintah untuk mengubah status peralatan melalui data perintah yang dikirim dari *master station*.

Tulisan ini bertujuan untuk menjelaskan tentang perancangan dan pengembangan RTU yang telah kami lakukan. RTU yang kami kembangkan menggunakan *embedded controller* sebagai komponen utama. *Embedded controller* digunakan dengan alasan operasionalnya yang mudah, tidak tergantung pada sistem operasi tertentu, dan perangkat lunak pada RTU dapat dikembangkan dengan bahasa pemrograman tingkat tinggi. Perangkat keras *embedded controller* sudah mendukung komunikasi berbasis IP sehingga kami dapat mengembangkan perangkat lunak RTU agar RTU dapat menjalin koneksi dengan *master station* melalui jaringan intranet atau internet.

RTU ini diharapkan dapat menjadi alternatif pilihan untuk sistem kontrol jarak jauh pada sistem yang tidak terlalu kompleks.

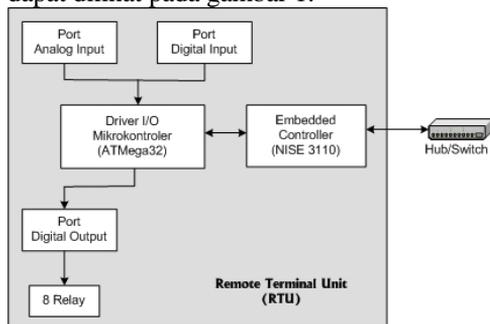
2. Perancangan

2.1 Perancangan perangkat keras

Hasil yang diharapkan dari kegiatan penelitian yang kami lakukan adalah sebuah RTU yang minimal memiliki:

- CPU (*Central Processing Unit*)
- memori (*volatile* dan *non volatile*)
- catu daya
- port komunikasi serial dan TCP/IP
- watch dog timer*
- real time clock*
- modul input output yang terdiri dari 8 *analog input*, 8 *digital input*, dan 8 *digital output*.

Untuk memenuhi spesifikasi RTU yang diharapkan tersebut, kami buat perancangan perangkat keras, dengan blok diagram yang dapat dilihat pada gambar 1.



Gambar 1 Diagram blok perangkat keras

Komponen perangkat keras utama yang digunakan adalah sebuah *embedded controller* dengan jenis NISE 3110 yang didalamnya telah termasuk CPU, catu daya, port komunikasi serial dan TCP/IP, *watch dog timer*, *real time clock*, dan lain-lain. Agar *embedded controller* dapat berkomunikasi dengan modul-modul I/O, dibuat sebuah *driver I/O* menggunakan mikrokontroler ATMega32. Pin I/O pada mikrokontroler ATMega32 sudah dapat digunakan sebagai *digital input* dan *digital output*. Modul *analog input* ditambahkan untuk menyesuaikan data analog (tegangan

atau arus) dari sensor atau instrumen menjadi level tegangan yang dapat dibaca oleh ADC yang ada dalam mikrokontroler ATMega32 (0-5Volt).

2.2 Perancangan perangkat lunak

Perangkat keras akan berfungsi jika ada perangkat lunak yang tepat didalamnya. Perangkat lunak yang dikembangkan dalam RTU terdiri dari dua bagian, yaitu perangkat lunak pada mikrokontroler dan perangkat lunak pada *embedded controller*.

2.2.1 Perangkat lunak di mikrokontroler

Perangkat lunak pada mikrokontroler dibuat dengan tujuan agar mikrokontroler ATMega32 dapat berfungsi sebagai *driver I/O* yang menghubungkan *embedded controller* dengan modul input output. Fungsi lengkap dari *driver I/O* adalah:

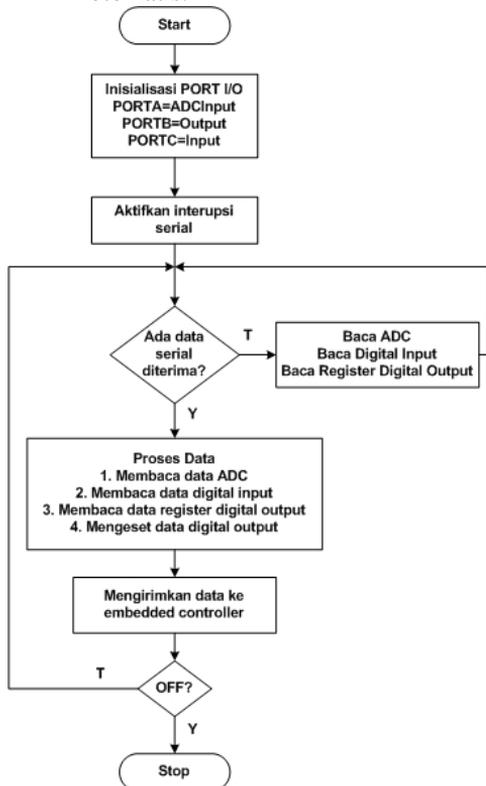
- Membaca data dari analog input. Hal ini dilakukan dengan *Analog To Digital Converter (ADC)* 10 bit yang berada di dalam IC ATMega32.
- Membaca data dari *digital input*.
- Membaca data dari register *digital output*.
- Mengatur nilai *digital output*.
- Berkomunikasi dengan *embedded controller* melalui kabel serial RS-232 dengan fungsi:
 - o mengaktifkan *interrupt serial receive* sehingga mikrokontroler dapat menerima data serial kapan saja
 - o memproses *command* dari *embedded controller*
 - o mengirimkan data yang sesuai kepada *embedded controller* sesuai dengan *command* yang diberikan.

Algoritma program pada mikrokontroler dapat dilihat pada gambar 2.

2.2.2 Perangkat lunak pada *embedded controller*

Perangkat lunak pada *embedded controller* dibangun dengan bahasa pemrograman Visual Basic 6. Fungsi dari perangkat lunak pada *embedded controller* adalah:

- Mengatur komunikasi serial RS-232 dengan mikrokontroler.
- Membuka soket komunikasi TCP/IP.
- Menerima data *command* (perintah) dari soket TCP/IP.
- Menjalankan perintah dari soket TCP/IP.
- Menerima data berupa program otomatis yang akan dijalankan, dari soket TCP/IP.
- Menjalankan program yang telah dikirim, sehingga proses yang dikontrol dapat berjalan secara otomatis.



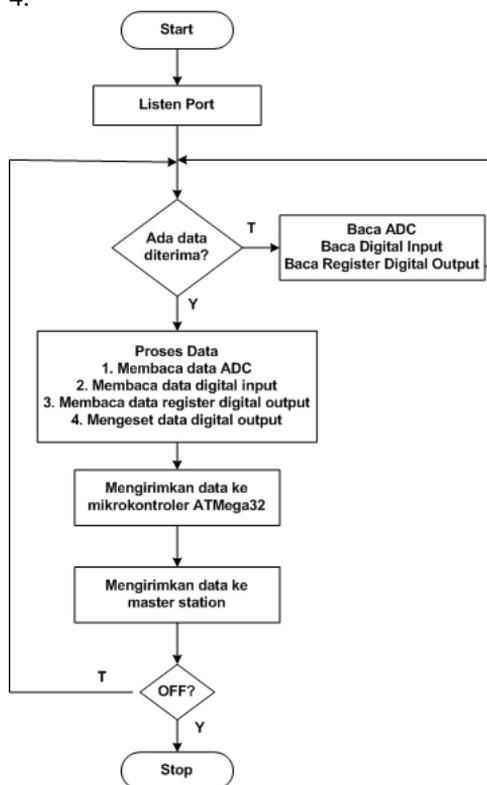
Gambar 2 Algoritma program pada mikrokontroler

Embedded controller menyimpan data dari mikrokontroler ATmega32 ke dalam buffer RAM terlebih dahulu sehingga data dapat dikirimkan segera setelah ada permintaan dari komputer master (*master station*). Algoritma program pada *embedded controller* dapat dilihat pada gambar 3.

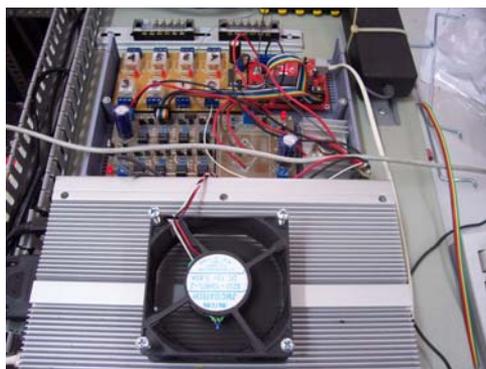
3. Hasil

3.1 Perangkat keras RTU

Implementasi dari perancangan RTU yang telah dilakukan, diperoleh perangkat keras RTU seperti yang terlihat pada gambar 4.



Gambar 3 Algoritma program pada *embedded controller*



Gambar 4 Perangkat keras RTU

Embedded controller terhubung dengan *driver I/O* melalui kabel serial RS-232. RTU

dipasang dekat dengan *plant* yang akan dikontrol, kemudian dihubungkan dengan *switch* melalui kabel jaringan RJ 45. Setelah program pada RTU dijalankan, RTU telah siap untuk diakses oleh *master station* melalui jaringan TCP/IP.

3.2 Perangkat lunak RTU

Perangkat lunak pada RTU dibuat dalam bentuk *Graphical User Interface* (GUI) agar mudah dalam pengembangan dan mudah untuk perbaikan program jika ada masalah pada RTU. Selama pengembangan, RTU masih memerlukan *peripheral* komputer (layar, *keyboard*, dan *mouse*). Setelah RTU berjalan sesuai dengan yang diharapkan, layar, *keyboard*, maupun *mouse* tidak diperlukan lagi. Apabila program telah dijalankan, RTU dapat beroperasi secara otomatis, dan proses kontrol dan monitoring dilakukan dari *master station* melalui perangkat lunak HMI (*Human Machine Interface*).

Hasil pengembangan perangkat lunak pada *embedded controller* dijelaskan pada uraian berikut ini.

3.2.1 Koneksi *embedded controller* dengan mikrokontroler

Koneksi *embedded controller* dengan mikrokontroler sebagai *driver I/O* menggunakan kabel serial (RS 232). Pada saat program dimulai, koneksi antara *embedded controller* dan mikrokontroler dibuat dengan cara membuka menu **File -> Koneksi**. Selanjutnya user memilih port serial yang digunakan (COM) dan menekan tombol **Connect**. Untuk keperluan *debug*, user dapat menandai pilihan **Cek koneksi modul dengan PC**. Pada kondisi ini perintah RS-232 dapat diberikan langsung kepada mikrokontroler untuk melihat langsung respon mikrokontroler. Respon mikrokontroler terlihat pada *buffer RS-232 input*. Tampilannya dapat dilihat pada gambar 5.

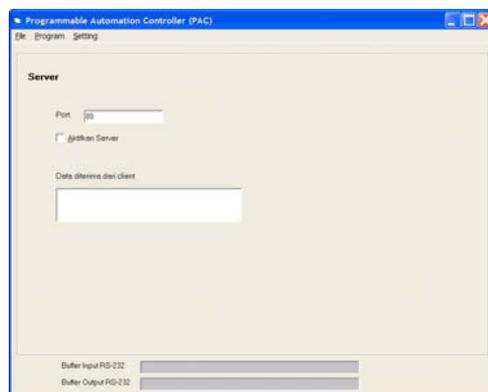
3.2.2 Aktivasi server TCP/IP

Embedded controller dalam kegiatan ini berfungsi sebagai *server TCP/IP* untuk menerima perintah dari *client* (HMI pada

master station) dan merespon *command* yang sesuai. Tampilan aktivasi *server* dapat dilihat pada gambar 6.



Gambar 5 Koneksi *embedded controller* dengan mikrokontroler



Gambar 6 Aktivasi server

Perjanjian *command* yang dikirim dari HMI adalah dimulai dengan tanda '>' dan diakhiri karakter '#'. Daftar perintah yang dapat diberikan melalui TCP/IP oleh *client* kepada *server* dapat dilihat pada tabel 1.

3.2.3 Penyajian data *realtime*

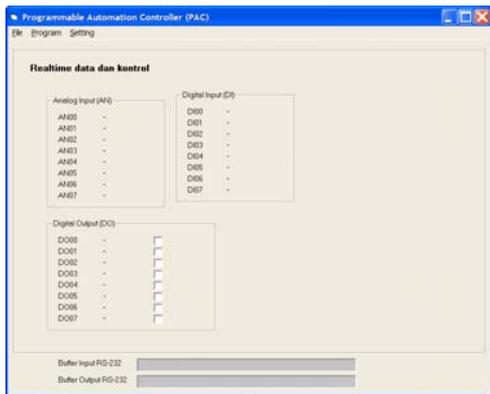
Embedded controller dapat menyajikan data *realtime*, yaitu data analog input, digital input dan digital output. Digital output juga dapat diatur nilainya secara *realtime*. Nilai yang diperoleh *analog input* adalah 4 digit 10 bit (Range 0000 – 1023). Nilai *digital input* dan *digital output* adalah 0 atau 1. Tampilannya dapat dilihat pada gambar 7.

3.2.4 Membuat program untuk otomatisasi sistem

Program untuk menjalankan sistem secara otomatis dapat dilakukan dari

perangkat lunak RTU yang telah dikembangkan, dengan sintaks yang disiapkan sama dengan sintaks standar bahasa C. Pemrograman dilakukan melalui menu **Program -> Tulis**. Misalnya sintaks program untuk menyalakan dan mematikan digital output 0 secara bergantian adalah sebagai berikut:

```
LB ; DO001 ; DELAY ( 1000 ) ;
DO000 ; DELAY ( 1000 ) ;
GOTOLB ;
```



Gambar 7 Penyajian data real time

3.2.5 Pengaturan channel

Secara normal, data *analog input* bernilai dari 0000 sampai 1023, data ini dapat dimodifikasi dengan persamaan garis lurus, dengan persamaan:

$$D_{akhir} = (D_{awal} \times Y) + K, \quad (1)$$

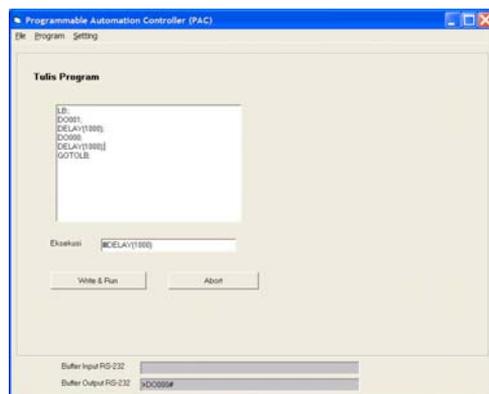
dimana:

D_{akhir} = data akhir

D_{awal} = data awal

Y = gradient

K = konstanta.



Gambar 8 Menulis program untuk RTU

Tabel 1 Kode perintah untuk RTU

No.	Jenis perintah	Kode String
1.	Permintaan data analog input	>ANXX#
2.	Permintaan data digital input	>DIXX#
3.	Setting digital output high	>DOXX1#
4.	Setting digital output low	>DOXX0#
5.	Permintaan seluruh data I/O	>ALL#
6.	Tulis program	>WRITE XXXXX#
7.	Jalankan program	>RUN#
8.	Hentikan program	>STOP#

Program dapat dijalankan dengan menekan tombol **Write & Run** dan dapat dihentikan dengan menekan tombol **Abort**. Tampilannya dapat dilihat pada gambar 8.

Program juga dapat dibuat di HMI, dikirim ke RTU dan kemudian RTU menjalankannya dengan perintah tertentu dari HMI.

Melalui menu **Setting -> Channel**, nilai gradien dan konstanta dapat ditentukan oleh user. Jika gradien = 1 dan konstanta = 0, maka data dari analog input (0000 – 1023) langsung ditampilkan tanpa modifikasi. Tampilannya dapat dilihat pada gambar 9.



Gambar 9 Setting channel

3.2.6 Kalibrasi

Fungsi kalibrasi adalah menentukan gradien dan konstanta agar sensor menunjukkan nilai yang sesuai dengan

standar. Hal ini dapat dilakukan dengan lebih mudah melalui menu **Setting -> Kalibrasi**. Tampilannya dapat dilihat pada gambar 10. Sebagai contoh, untuk sensor suhu termokouple dengan *transmitter* 4 – 20 mA dan jangkauan nilai suhu 0 – 500⁰ C, langkah-langkah kalibrasi adalah sebagai berikut:

- Set Gradien = 1, Konstanta = 0, hubungkan *analog input* dengan kalibrator untuk nilai terendah (Misalnya 4mA). Tulis hasilnya di kotak X1 (Misalnya 196).
- Set Gradien = 1, Konstanta = 0, hubungkan *analog input* dengan kalibrator untuk nilai tertinggi (Misalnya 20mA). Tulis hasilnya di kotak X1 (Misalnya 983).
- Tulis nilai terendah sensor (Misalnya 0⁰C) di kotak Y1.
- Tulis nilai tertinggi sensor (Misalnya 500⁰C) di kotak Y2.
- Pilih *channel* yang sesuai dan klik tombol **Hitung**.

Setelah diperoleh nilai gradien dan konstanta, maka *user* dapat memakai nilai ini secara langsung pada menu **Setting -> Channel** atau memodifikasinya.



Gambar 10 Setting kalibrasi

4. Pengujian

RTU yang dikembangkan telah diuji coba dengan cara diakses dari HMI (*master station*) melalui jaringan berbasis IP. Langkah-langkah pengujian adalah sebagai berikut:

- HMI membentuk koneksi dengan RTU melalui jaringan intranet.
- Setelah koneksi terbentuk (status koneksi dapat dilihat pada label HMI), dicoba menyalakan dan mematikan relay yang telah terpasang pada *digital output* RTU melalui *check box* yang tersedia pada HMI.
- Setelah mengendalikan relay secara manual berhasil, langkah berikutnya adalah menulis program pada menu yang tersedia di HMI, mengupload program tersebut ke RTU, kemudian mengirim perintah untuk menjalankannya.

Untuk menguji jalan atau tidaknya sistem yang dikembangkan, dipasang kamera berbasis IP pada RTU. Hasil tangkapan kamera ditampilkan pada *web browser*. Berhasil atau tidak perintah yang dikirimkan dapat dilihat dari nyala mati lampu LED pada relay yang terlihat di *web browser*. Selain itu, dapat dilihat juga dari nilai input output *real time* yang ditampilkan pada layar HMI.

Tampilan HMI yang kami pergunakan untuk uji coba dapat dilihat pada gambar 11 dan gambar 12. Gambar 11 menampilkan menu *check box* untuk menyalakan dan mematikan relay, gambar 12 menampilkan menu *setting* dan *textfield* untuk menuliskan program. Pengiriman *command* ke RTU dilakukan dengan menekan tombol yang sesuai, yang terdapat pada HMI.



Gambar 11 Menu kendali relay manual

Hasil pengujian dapat dilihat pada kondisi input output yang ditampilkan pada *display* HMI, atau dengan menekan tombol

Camera. Uji coba dilakukan beberapa kali, sampai diperoleh respon RTU sesuai dengan program yang dikirimkan dari HMI, dan tidak menunjukkan adanya kesalahan.



Gambar 12 Menu setting dan tulis program otomatis

5. Kesimpulan

RTU yang kami kembangkan telah memenuhi fungsi RTU sebagai antarmuka antara objek yang dikontrol dengan *master station*. Salah satu kelebihan dari RTU ini adalah adanya kebebasan dalam pengembangan perangkat lunak. Perangkat lunak yang telah kami kembangkan saat ini memiliki kelebihan diantaranya adalah RTU telah memiliki GUI sehingga apabila terjadi masalah dalam sistem, misalnya ada masalah komunikasi antara *master station* dengan RTU, perbaikan dapat dilakukan langsung dari RTU, tanpa melibatkan *master station*.

6. Daftar pustaka

- [1] __, *Introduction to SCADA*, <http://www.scribd.com/doc/14120168/Introduction-to-SCADA>, diakses pada 28 Januari 2009.
- [2] __, *A Tutorial on RTU*, <http://www.m-indya.com/tutorial.php?tutorialid=15>, diakses pada 10 Januari 2010.
- [3] M.Ary Heryanto, Wisnu Adi P, *Pemrograman Bahasa C untuk Mikrokontroler ATmega8535*, Penerbit Andi, Yogyakarta, 2008.
- [4] Ardi Winoto, *Belajar Mikrokontroler Atmel AVR Attiny2313*, Penerbit Gava Media, Yogyakarta, 2006.
- [5] __, *Building the Digital Infrastructure*, Nexcom, Taiwan, 2009.
- [6] Rika Sustika, Endang Suryawati, Oka Mahendra, dan Djohar Syamsi, “Supervisory Control Berbasis TCP/IP untuk Otomasi Pilot Plant Sistem Kontinu”, di *Prosiding SRITI*, Yogyakarta, 2009.